University of Silesia
Faculty of Computer and Materials Science
Institute of Computer Science

mgr Beata Zielosko

PhD dissertation

Construction and Optimization of Partial Decision Rules

Supervisor:
prof. dr hab. Mikhail Ju. Moshkov

Sosnowiec, 2008

Wyrażam zgodę na udostępnienie mojej dysertacji doktorskiej dla celów naukowo-badawczych.

Data . . . . Podpis autora pracy . . . .

Słowa kluczowe: partial decision rules, greedy algorithm, rough set theory

Oświadczenie autora pracy

Świadoma odpowiedzialności prawnej oświadczam, że niniejsza dysertacja doktorska została napisana przeze mnie samodzielnie i nie zawiera treści uzyskanych w sposób niezgodny z obowiązującymi przepisami.

Data . . . . Podpis autora pracy . . . .

Abstract

In thesis we study greedy algorithms for construction and optimization of partial (approximate) decision rules.

The study of partial decision rules is based on the study of partial covers. We prove that under some natural assumptions on the class NP, the greedy algorithm is close (from the point of view of precision) to the best polynomial approximate algorithms for minimization of the length of partial decision rules and minimization the total weight of attributes in partial decision rule.

Based on an information received during the greedy algorithm work, it is possible to obtain nontrivial lower and upper bounds on the minimal complexity of partial decision rules. Theoretical and experimental results show that these bounds can be used in practice.

We obtain a new bound on the precision of greedy algorithm for partial decision rule construction that does not depend on the number of rows in decision table.

Under some assumptions on the number of rows and number of columns in decision tables we prove that, for the most part of binary decision tables exist short irreducible partial decision rules.

Theoretical and experimental results confirm the following 0.5-hypothesis for decision rules: for the most part of decision tables greedy algorithm during partial decision rules construction chooses an attribute, that separates at least one-half of unseparated rows which should be separated. It means that greedy algorithm constructs often short partial rules with relatively high accuracy.

Results of experiments with decision tables from UCI Repository of Machine Learning Databases show that, the accuracy of classifiers based on partial decision rules is often better than the accuracy of classifiers based on exact decision rules.

Streszczenie

Tematyka pracy związana jest z badaniem algorytmów zachłannych dla konstruowania i optymalizacji częściowych (przybliżonych) reguł decyzyjnych.

Przedstawione w pracy badania dotyczące częściowych reguł decyzyjnych opierają się na wynikach badań uzyskanych dla problemu częściowego pokrycia zbioru.

Zostało udowodnione, że biorąc pod uwagę pewne założenia dotyczące klasy NP, algorytm zachłanny pozwala uzyskać wyniki, bliskie wynikom uzyskiwanym przez najlepsze przybliżone wielomianowe algorytmy, dla minimalizacji długości częściowych reguł decyzyjnych oraz minimalizacji całkowitej wagi atrybutów tworzących częściową regułę decyzyjną.

Na podstawie danych uzyskanych podczas pracy algorytmu zachłannego, dokonano oszacowania najlepszych górnych i dolnych granic minimalnej złożoności częściowych reguł decyzyjnych. Teoretyczne i eksperymentalne wyniki badań pokazały możliwość wykorzystania tych granic w praktycznych zastosowaniach.

Dokonano także oszacowania granicy dokładności algorytmu zachłannego dla generowania częściowych reguł decyzyjnych, która nie zależy od liczby wierszy w rozważanej tablicy decyzyjnej.

Biorąc pod uwagę pewne założenia dotyczące liczby wierszy i kolumn w tablicach decyzyjnych udowodniono, że dla większości binarnych tablic decyzyjnych istnieją tylko krótkie, nieredukowalne częściowe reguły decyzyjne.

Wyniki przeprowadzonych eksperymentów pozwoliły potwierdzić 0.5-hipotezę: dla większości tablic decyzyjnych algorytm zachłanny w każdej iteracji, podczas generowania częściowej reguły wybiera atrybut, który pozwala oddzielić przynajmniej 50% wierszy jeszcze nie oddzielonych.

W przypadku klasyfikacji okazało się, że dokładność klasyfikatorów opartych na częściowych regułach decyzyjnych jest często lepsza, niż dokładność klasyfikatorów opartych na dokładnych regułach decyzyjnych.

Acknowledgements

I would like to express my gratitude to my supervisor Professor Mikhail Ju. Moshkov for his support during the last few years. Dear Professor I am grateful that You appeared on my scientific way. Thank You for Your guadience in my research work, patience and stimulated disscussions.

I would like to thank my family for forbearance and help in difficulties,

and other people who contributed to my thesis for various support of my work.

Contents

Introduction

The subject matter of this thesis is connected with the following two problems of data mining and knowledge discovery:

1. Representation of knowledge, contained in a decision table, in a form which is convenient for understanding. The length of knowledge description is crucial in this case.
2. Prediction of the value of decision attribute for a new object. The accuracy of prediction is the most important aspect of this problem.

These two aims (short description and high accuracy) seem to be incompatible. However, it is known that classifiers with shorter description are often more precise. This dissertation is one more confirmation of this fact.

In this thesis, we study one of the main notions of rough set theory: the notion of decision rule (local reduct) [45, 47, 48, 49, 56, 69].

Let T be a table with n rows labeled with nonnegative integers (decisions) and m columns labeled with conditional attributes $f_1, \ldots, f_m$. This table is filled by nonnegative integers (values of attributes). The table T is called a decision table. We say that an attribute f_i separates rows r_1 and r_2 of T if these rows have different values at the intersection with the column f_i. Two rows are called different if at least one attribute f_i separates these rows.

Let $r = (b_1, \ldots, b_m)$ be a row of T labeled with a decision d. By $U(T, r)$ we denote the set of rows from T which are different from r and are labeled with decisions different from d. Let $\alpha \in \mathbb{R}$ and $0 \leq \alpha < 1$. A decision rule

$$(f_{i_1} = b_{i_1}) \wedge \ldots \wedge (f_{i_t} = b_{i_t}) \rightarrow d$$

is called an *α-decision rule* for T and r if attributes $f_{i_1}, \ldots, f_{i_t}$ separate from r at least $(1 - \alpha)|U(T, r)|$ rows from $U(T, r)$.

Exact decision rules are widely used in rough set theory both for construction of classifiers and as a way of knowledge representation [55]. In particular, the presence of decision rules with small number of attributes can simplify the understanding of relationships among decision and conditional attributes. Note that notions similar to the notion of decision rule are studied deeply also in test theory [6, 10, 65, 73, 76, 77], where the notion of control test is not far from the notion of decision rule, and in logical analysis of data (LAD) [1, 9], where pattern is an analog of decision rule.

The main idea of the thesis is the following: instead of exact decision rules we can use partial (approximate) rules. Exact decision rules can be overfitted, i.e., dependent essentially on the noise or adjusted too much to the existing examples. If decision rules are considered as a way of knowledge representation, then instead of an exact decision rule with many attributes, it is more appropriate to work with a partial decision rule containing smaller number of attributes that separate from given row almost all other rows with different decisions.

The considered idea is not new. For years, in rough set theory partial reducts and partial decision rules (partial local reducts) are studied intensively by H.S. Nguyen, A. Skowron, D. Ślęzak, Z. Pawlak, J. Wróblewski and others [2, 42, 43, 46, 47, 60, 61, 62, 63, 64, 71]. There is a number of approaches to the definition of approximate reducts [62]. In [43, 61, 62, 63] it was proved that for each of the considered approaches the problem of partial reduct minimization (construction of a partial reduct with minimal cardinality) is NP-hard. The approach considered in [43] is similar to the approach studied in this dissertation (see also [61, 63]). More detailed discussion of partial decision rules considered in this thesis can be found in Chap. 5. Approximate reducts are also investigated by W. Ziarko, M. Quafafou and others in the extensions of rough set model such as variable precision rough sets (VPRS) [78] and alpha rough set theory (α-RST) [51].

There are different measures of the quality of decision rules: the length of rule, the total weight of attributes in decision rule, the support of decision rule, etc. We are concentrate here on minimization of the length of rules (which allows us to design more precise classifiers or obtain more compact representation of knowledge contained in decision tables) or on minimization of the total weight of rules (which allows us to minimize time complexity or cost, or risk of classifier work).

There are different approaches to construction of decision rules: brute-force approach which is applicable to tables with relatively small number of attributes, genetic algorithms [64, 72], simulated annealing [15], Boolean reasoning [42, 49, 56], ant colony optimization [23], algorithms based on decision tree construction [4, 14, 25, 52], different kinds of greedy algorithms [40, 42, 59].

Each method can have different modifications. For example, as in the case of decision trees, we can use greedy algorithms based on Gini index, entropy, etc., for construction of decision rules.

In thesis we study mainly greedy algorithms for construction of rules. Of course, these algorithms are not new, and were used by different authors [16]. Our choice is connected with mathematical results obtained for greedy algorithms. In particular, we prove that, under some natural assumptions on the class NP, greedy algorithms are not far from the best polynomial algorithms for decision rules optimization.

The most important feature of this thesis is a serious mathematical analysis of problems of partial decision rule construction, which is closely connected with results of experiments. In many cases, experimental results led to important and unexpected new statements, and mathematical analysis allowed us to choose new directions of research in a well-grounded way.

The study of partial decision rules is based on the study of partial covers. Let $A = \{a_1, \ldots, a_n\}$ be a nonempty finite set and $S = \{B_1, \ldots, B_m\}$ be a family of subsets of A such that $B_1 \cup \ldots \cup B_m = A$. Let $\alpha \in \mathbb{R}$ and $0 \leq \alpha < 1$. A subfamily $Q = \{B_{i_1}, \ldots, B_{i_t}\}$ of the family S is called an *α-cover* for (A, S) if $|B_{i_1} \cup \ldots \cup B_{i_t}| \geq (1 - \alpha)|A|$.

There exists simple reduction of the problem of construction of a 0-cover with minimal cardinality to the problem of construction of a 0-decision rule with minimal length. There exists also the opposite reduction which is simple too. The similar situation is with partial covers and partial rules (where $\alpha > 0$). This fact allows us to use various mathematical results obtained for the set cover problem by J. Cheriyan and R. Ravi [7], V. Chvátal [8], U. Feige [11], D.S. Johnson [16], R.M. Karp [17], M.J. Kearns [18], L. Lovász [24], R.G. Nigmatullin [44], R. Raz and S. Safra [53], and P. Slavík [57, 58] for analysis of partial rules. In addition, we use a technique created by D. Ślęzak [61, 63] for the proof of NP-hardness of partial reduct optimization.

Known and new (obtained in this thesis) results for covers and partial covers will be useful for wider spectrum of problems considered in rough set theory, for example, for the investigation of (i) reducts and rules for information systems, (ii) reducts and rules for decision tables with missing values, (iii) subsystems of a given decision rule system which “cover" the same set of rows, etc.

The thesis contains five chapters.

In Chap. 1, we prove that, under some natural assumptions on the class NP, the greedy algorithm is close (from the point of view of precision) to the best polynomial approximate algorithms for partial cover optimization.

Information about the greedy algorithm work can be used for obtaining lower and upper bounds on the minimal cardinality of partial covers. We fix some kind of information, and find the best lower and upper bounds depending on this information. Theoretical and experimental (see also Chap. 4) results show that the obtained lower bound is nontrivial and can be used in practice.

We obtain a new bound on the precision of greedy algorithm for partial cover construction that does not depend on the cardinality of covered set, and prove that this bound is, in some sense, unimprovable.

We prove that for the most part of set cover problems there exist exact (and, consequently, partial) covers with small cardinality. Results of experiments with randomly generated set cover problems allow us to formulate the following informal 0.5-hypothesis: for the most part of set cover problems, during each step the greedy algorithm chooses a subset which covers at least one-half of uncovered elements. We prove that, under some assumptions, the 0.5-hypothesis is true.

The most part of results obtained for partial covers is generalized to the case of partial decision rules.

In particular, we show that, under some natural assumptions on the class NP, greedy algorithm is close to the best polynomial approximate algorithms for the minimization of the length of partial decision rules.

Based on an information received during the greedy algorithm work, it is possible to obtain nontrivial lower and upper bounds on the minimal length of partial decision rules.

For the most part of randomly generated binary decision tables, the greedy algorithm constructs simple partial decision rules with relatively high accuracy. In particular, experimental and theoretical results confirm the following 0.5-hypothesis for decision rules: in the most part of cases, greedy algorithm chooses an attribute that separates at least one-half of unseparated rows which should be separated.

In Chap. 2, we study the case, where each subset used for covering has its own weight, and we should minimize the total weight of subsets in partial cover. The same situation is with partial decision rules: each conditional attribute has its own weight, and we should minimize the total weight of attributes in partial decision rule. The weight of attribute characterizes time complexity, cost or risk (as in medical or technical diagnosis) of attribute value computation. The most part of results obtained in Chap. 1 is generalized to the case of arbitrary natural weights.

We generalize usual greedy algorithm with weights, and consider greedy algorithm with two thresholds. The first threshold gives the exactness of constructed partial cover, and the second one is an interior parameter of the considered algorithm. We prove that, for the most part of set cover problems there exists a weight function and values of thresholds such that, the weight of partial cover constructed by greedy algorithm with two thresholds is less than the weight of partial cover constructed by usual greedy algorithm. The same situation is with partial decision rules. Based on greedy algorithm with two thresholds we create new polynomial time approximate algorithms for minimization of total weights of partial covers and decision rules.

Results of massive experiments with randomly generated set cover problems and binary decision tables show that the new algorithms can be used in practice.

In Chap. 3, we consider binary decision tables with m conditional attributes, in which the number of rows is equal to $\lfloor m^{\alpha} \rfloor$, where α is a positive real number, and partial decision rules that can leave unseparated from a given row at most $5\left\lceil (\log_2 m)^{\beta} \right\rceil$ different rows with different decisions, where β is a real number such that $\beta \geq 1$.

We show that for almost all such tables for any row with minor decision (minor decision is a decision which is attached to at most one-half of rows of decision table) the length of each irreducible partial decision rule is not far from $\alpha \log_2 m$ and the number of irreducible partial decision rules is not far from $m^{\alpha \log_2 m}$.

Based on these results, we prove that there is no algorithm which for almost all decision tables for each row with minor decision constructs the set of irreducible partial decision rules and has for these tables polynomial time complexity depending on the length of input. However, there exists an algorithm which for almost all decision tables for each row with minor decision constructs the set of irreducible partial decision rules and has for these tables polynomial time complexity depending on the length of input and the length of output.

Chapter 4 is devoted to consideration of results of experiments with decision tables from UCI Repository of Machine Learning Databases [41]. The aim of the first group of experiments is to verify 0.5-hypothesis for real-life decision tables. We made experiments with 23 decision tables. Results of 20 experiments confirm 0.5-hypothesis: under the construction of partial decision rule, during each step the greedy algorithm chooses an attribute which separates from r at least one-half of unseparated rows that are different from r and have other decisions. It means that the greedy algorithm can often construct short partial decision rules with relatively high “accuracy". In particular, for the cases, where 0.5-hypothesis is true, the greedy algorithm constructs a partial decision rule with seven attributes only which separate from a given row at least 99% of different rows with different decisions. Such short partial decision rules are easy for understanding.

The aim of the second group of experiments is the comparison of accuracy of classifiers based on exact and partial decision rules. The considered approach to construction of classifiers is the following: for a given decision table and each row we construct a (partial) decision rules using greedy algorithm. By removing some attributes from this (partial) decision rule we obtain an irreducible (partial) decision rule. The obtained system of rules jointly with simple procedure of voting can be considered as a classifier [19, 20, 55]. We made experiments with 21 decision tables using test-and-train method. In 11 cases, we found partial decision rules for which the accuracy of the constructed classifiers is better than the accuracy of classifiers based on exact decision rules. We made also experiments with 17 decision tables using cross-validation method. In 9 cases, we found partial decision rules for which the accuracy of the constructed classifiers is better than the accuracy of classifiers based on exact decision rules.

In Chap. 5, we consider an universal attribute reduction problem. Let T be a decision table and $\mathcal{P}$ be a subset of pairs of discernible rows (objects) of T. Then a reduct for T relative to $\mathcal{P}$ is a minimal (relative to inclusion) subset of conditional attributes which separate all pairs from $\mathcal{P}$. Reducts for information systems, usual decision and local reducts (decision rules) for decision tables, decision and local reducts, which are based on the generalized decision, can be represented in such a form. We study not only exact, but also partial reducts. Moreover, we consider a scenario of the work with real data tables that can contain continuous variables, discrete variables with large number of values, and variables with missing values.

Based on results from Chap. 1, we obtain bounds on precision of greedy algorithm for partial super-reduct construction. We prove that, under some natural assumptions on the class NP, the greedy algorithm is close to the best (from the point of view of precision) polynomial approximate algorithms for minimization of cardinality of partial super-reducts. We show that based on an information received during the greedy algorithm work it is possible to obtain a nontrivial lower bound on minimal cardinality of partial reduct. We obtain also a bound on precision of greedy algorithm which does not depend on the cardinality of the set $\mathcal{P}$.

Experimental and theoretical results obtained in this thesis show that the use of partial decision rules instead of exact ones can allow us to obtain more compact description of knowledge contained in decision tables, and to design more precise classifiers. This is a reason to use partial decision rules in data mining and knowledge discovery for knowledge representation and for prediction.

The results obtained in this thesis can be useful for researchers in such areas as machine learning, data mining and knowledge discovery, especially for those who are working in rough set theory, test theory and logical analysis of data.

An essential part of software used in experiments described in Chaps. 1 and 4 will be accessible soon in RSES – Rough Set Exploration System [54] (Institute of Mathematics, Warsaw University, head of project – Professor Andrzej Skowron).

1

Partial Covers and Decision Rules

In this chapter, we consider theoretical and experimental results on partial decision rules. These investigations are based on the study of partial covers.

Based on the technique created by Ślęzak in [61, 63], we generalize well known results of Feige [11], and Raz and Safra [53] on the precision of approximate polynomial algorithms for exact cover minimization (construction of an exact cover with minimal cardinality) to the case of partial covers. From obtained results and results of Slavík [57, 58] on the precision of greedy algorithm for partial cover construction it follows that, under some natural assumptions on the class NP, the greedy algorithm for partial cover construction is close (from the point of view of precision) to the best polynomial approximate algorithms for partial cover minimization.

An information about the greedy algorithm work can be used for obtaining lower and upper bounds on the minimal cardinality of partial covers. We fix some kind of information, and find the best lower and upper bounds depending on this information.

We obtain a new bound on the precision of greedy algorithm for partial cover construction which does not depend on the cardinality of covered set. This bound generalizes the bound obtained by Cheriyan and Ravi [7] and improves the bound obtained by Moshkov [27]. Based on the results of Slavík [57, 58] on the precision of greedy algorithm for partial cover construction, we prove that obtained bound is, in some sense, unimprovable.

We prove that for the most part of set cover problems there exist exact (and, consequently, partial) covers with small cardinality. Experimental results allows us to formulate the following informal 0.5-hypothesis for covers: for the most part of set cover problems, during each step the greedy algorithm chooses a subset which covers at least one-half of uncovered elements. We prove that, under some assumption, the 0.5-hypothesis for covers is true.

The most part of results obtained for partial covers is generalized to the case of partial decision rules. In particular, we show that

- Under some natural assumptions on the class NP, greedy algorithm is close to the best polynomial approximate algorithms for the minimization of the length of partial decision rules.
- Based on an information received during the greedy algorithm work, it is possible to obtain nontrivial lower and upper bounds on the minimal length of partial decision rules.
- For the most part of randomly generated binary decision tables, greedy algorithm constructs simple partial decision rules with relatively high accuracy. In particular, experimental and theoretical results confirm the 0.5-hypothesis for decision rules.

This chapter is based on papers [31, 38, 79].

The chapter consists of three sections. In Sect. 1.1, partial covers are studied. In Sect. 1.2, partial decision rules are considered. Section 1.3 contains short conclusions.

1.1 Partial Covers

This section consists of six subsections. In Sect. 1.1.1, main notions are described. In Sect. 1.1.2, known results are considered. In Sect. 1.1.3, polynomial approximate algorithms for partial cover minimization (construction of partial cover with minimal cardinality) are studied. In Sect. 1.1.4, upper and lower bounds on minimal cardinality of partial covers based on an information about greedy algorithm work are investigated. In Sect. 1.1.5, an upper bound on cardinality of partial cover constructed by the greedy algorithm is considered. In Sect. 1.1.6, exact and partial covers for the most part of set cover problems are discussed from theoretical and experimental points of view.

1.1.1 Main Notions

Let $A = \{a_1, \ldots, a_n\}$ be a nonempty finite set and $S = \{B_i\}_{i \in \{1,\ldots,m\}} = \{B_1, \ldots, B_m\}$ be a family of subsets of A such that $B_1 \cup \ldots \cup B_m = A$. We assume that S can contain equal subsets of A. The pair (A, S) is called a *set cover problem.*

Let I be a subset of $\{1, \ldots, m\}$. The family $P = \{B_i\}_{i \in I}$ is called a *subfamily* of S. The number $|I|$ is called the *cardinality* of P and is denoted by $|P|$. Let $P = \{B_i\}_{i \in I}$ and $Q = \{B_i\}_{i \in J}$ be subfamilies of S. The notation $P \subseteq Q$ means that $I \subseteq J$. Let $P \cup Q = \{B_i\}_{i \in I \cup J}$, $P \cap Q = \{B_i\}_{i \in I \cap J}$, and $P \setminus Q = \{B_i\}_{i \in I \setminus J}$.

A subfamily $Q = \{B_{i_1}, \ldots, B_{i_t}\}$ of the family S is called a *partial cover for* (A, S). Let $\alpha \in \mathbb{R}$ and $0 \le \alpha < 1$. The subfamily Q is called an α*-cover for* (A, S) if $|B_{i_1} \cup \ldots \cup B_{i_t}| \ge (1 - \alpha)|A|$. For example, 0.01-cover means that we should cover at least 99% of elements from A. Note that a 0-cover is an exact cover. By $C_{\min}(\alpha) =$

$C_{\min}(\alpha, A, S)$ we denote the minimal cardinality of α-cover for (A, S). The notation $C_{\min}(\alpha)$ will be used in cases, where A and S are known.

Let us consider a greedy algorithm with threshold α which constructs an α-cover for (A, S) (see Algorithm 1).

Algorithm 1: Greedy algorithm for partial cover construction

```
Input  : Set cover problem $(A, S)$ with $S = \{B_1, \ldots, B_m\}$, and real number $\alpha$, $0 \le \alpha < 1$.
Output: $\alpha$-cover for $(A, S)$.
$Q \longleftarrow \emptyset$;
while $Q$ is not an $\alpha$-cover for $(A, S)$ do
    select $B_i \in S$ with minimal index $i$ such that $B_i$ covers the maximal number of elements from $A$
    uncovered by subsets from $Q$;
    $Q \longleftarrow Q \cup \{B_i\}$;
end
return $Q$;
```

By $C_{\text{greedy}}(\alpha) = C_{\text{greedy}}(\alpha, A, S)$ we denote the cardinality of constructed α-cover for (A, S).

1.1.2 Known Results

First, we consider some known results for exact covers, where $\alpha = 0$.

Theorem 1.1. (Nigmatullin [44])

$$C_{\text{greedy}}(0) \le C_{\min}(0)(1 + \ln |A| - \ln C_{\min}(0)) \ .$$

Theorem 1.2. (Johnson [16], Lovász [24])

$$C_{\text{greedy}}(0) \le C_{\min}(0)(1 + \ln(\max_{B_i \in S} |B_i|)) \le C_{\min}(0)(1 + \ln |A|) \ .$$

More exact bounds (depending only on $|A|$) were obtained by Slavík [57, 58].

Theorem 1.3. (Slavík [57, 58]) *If $|A| \ge 2$, then $C_{\text{greedy}}(0) < C_{\min}(0)(\ln |A| - \ln \ln |A| + 0.78)$.*

Theorem 1.4. (Slavík [57, 58]) *For any natural $m \ge 2$, there exists a set cover problem (A, S) such that $|A| = m$ and $C_{\text{greedy}}(0) > C_{\min}(0)(\ln |A| - \ln \ln |A| - 0.31)$.*

There are some results on exact and approximate polynomial algorithms for cover minimization.

Theorem 1.5. (Karp [17]) *The problem of construction of 0-cover with minimal cardinality is NP-hard.*

Theorem 1.6. (Feige [11]) *If $NP \not\subseteq DTIME(n^{O(\log\log n)})$, then for any ε, $0 < \varepsilon < 1$, there is no polynomial algorithm that for a given set cover problem (A, S) constructs a 0-cover for (A, S) which cardinality is at most $(1-\varepsilon)C_{\min}(0)\ln|A|$.*

Theorem 1.7. (Raz and Safra [53]) *If $P \neq NP$, then there exists $\gamma > 0$ such that there is no polynomial algorithm that for a given set cover problem (A, S) constructs a 0-cover for (A, S) which cardinality is at most $\gamma C_{\min}(0)\ln|A|$.*

Note that some results on the minimal exact covers for almost all set cover problems from some classes were obtained by Vercellis [66]. Kuzjurin in [22] investigated the behavior of greedy algorithm during the construction of exact covers for almost all problems from some classes of set cover problems such that each element from A belongs to the same number of subsets from S.

We now consider some known results for partial covers, where $\alpha \geq 0$.

Theorem 1.8. (Slavík [57, 58]) *Let $0 \leq \alpha < 1$ and $\lceil(1-\alpha)|A|\rceil \geq 2$. Then $C_{\text{greedy}}(\alpha) < C_{\min}(\alpha)(\ln\lceil(1-\alpha)|A|\rceil - \ln\ln\lceil(1-\alpha)|A|\rceil + 0.78)$.*

Theorem 1.9. (Slavík [57, 58]) *Let $0 \leq \alpha < 1$. Then for any natural $t \geq 2$ there exists a set cover problem (A, S) such that $\lceil(1-\alpha)|A|\rceil = t$ and $C_{\text{greedy}}(\alpha) > C_{\min}(\alpha)(\ln\lceil(1-\alpha)|A|\rceil - \ln\ln\lceil(1-\alpha)|A|\rceil - 0.31)$.*

Theorem 1.10. (Slavík [58]) *Let $0 \leq \alpha < 1$. Then $C_{\text{greedy}}(\alpha) \leq C_{\min}(\alpha)(1 + \ln(\max_{B_i \in S}|B_i|))$.*

There are some bounds on $C_{\text{greedy}}(\alpha)$ which does not depend on $|A|$. Note that in the next two theorems we consider the case, where $\alpha > 0$.

Theorem 1.11. (Cheriyan and Ravi [7]) *Let $0 < \alpha < 1$. Then $C_{\text{greedy}}(\alpha) \leq C_{\min}(0)\ln(1/\alpha) + 1$.*

This bound was rediscovered by Moshkov in [26] and generalized in [27].

Theorem 1.12. (Moshkov [27]) *Let $0 < \beta \leq \alpha < 1$. Then $C_{\text{greedy}}(\alpha) \leq C_{\min}(\alpha - \beta)\ln(1/\beta) + 1$.*

There is a result on exact polynomial algorithms for partial cover minimization.

Theorem 1.13. (Ślęzak [61, 63]) *Let $0 \leq \alpha < 1$. Then the problem of construction of α-cover with minimal cardinality is NP-hard.*

1.1.3 Polynomial Approximate Algorithms

In this subsection, using technique created by Ślęzak in [61, 63], we generalize the results of Feige, Raz and Safra (Theorems 1.6 and 1.7) to the case of partial covers.

When we say about a polynomial algorithm for set cover problems (A, S), it means that the time complexity of the considered algorithm is bounded from above by a polynomial depending on $|A|$ and $|S|$.

When we say about an algorithm, that for a given set cover problem (A, S) constructs an α-cover which cardinality is at most $f(A, S)C_{\min}(\alpha, A, S)$, we assume that in the case $f(A, S) < 1$ the considered algorithm constructs an α-cover for (A, S) which cardinality is equal to $C_{\min}(\alpha, A, S)$.

We consider an arbitrary set cover problem (A, S) with $S = \{B_1, \ldots, B_m\}$. Let $\alpha \in \mathbb{R}$ and $0 < \alpha < 1$. We correspond to (A, S) and α a set cover problem (A_α, S_α). Let $n(\alpha) = \lfloor |A|\alpha/(1-\alpha) \rfloor$ and $b_1, \ldots, b_{n(\alpha)}$ be elements which do not belong to the set A. Then $A_\alpha = A \cup \{b_1, \ldots, b_{n(\alpha)}\}$ and $S_\alpha = \{B_1, \ldots, B_m, B_{m+1}, \ldots, B_{m+n(\alpha)}\}$, where $B_{m+1} = \{b_1\}, \ldots, B_{m+n(\alpha)} = \{b_{n(\alpha)}\}$.

It is clear that there exists a polynomial algorithm which for a given set cover problem (A, S) and number α constructs the set cover problem (A_α, S_α).

Lemma 1.14. *Let $Q \subseteq S$ be a 0-cover for (A, S) and α be a real number such that $0 < \alpha < 1$. Then Q is an α-cover for (A_α, S_α).*

Proof. It is clear that $|A_\alpha| = |A| + n(\alpha)$. One can show that

$$|A| - 1 < (1-\alpha)|A_\alpha| \leq |A| \; . \tag{1.1}$$

It is clear that subsets from Q cover exactly $|A|$ elements from A_α. From (1.1) we conclude that Q is an α-cover for (A_α, S_α). □

Lemma 1.15. *Let $Q_\alpha \subseteq S_\alpha$ be an α-cover for (A_α, S_α). Then there exists $Q \subseteq S$ which is a 0-cover for (A, S) and for which $|Q| \leq |Q_\alpha|$. There exists a polynomial algorithm which for a given Q_α constructs corresponding Q.*

Proof. Let $Q_\alpha = Q^0 \cup Q^1$, where $Q^0 \subseteq S$ and $Q^1 \subseteq S_\alpha \setminus S$. If Q^0 covers all elements of the set A, then in the capacity of Q we can choose the set Q^0. Let Q^0 cover not all elements from A, A' be the set of uncovered elements from A, and $|A'| = m$. Taking into account that Q_α covers at least $(1-\alpha)|A_\alpha|$ elements from A_α and using (1.1) we conclude that Q_α covers greater than $|A| - 1$ elements. Thus, Q_α covers at least $|A|$ elements. It is clear that each subset from $S_\alpha \setminus S$ covers exactly one element. Therefore, $|Q^1| \geq m$. One can show that there exists a polynomial algorithm which finds $t \leq m$ subsets $B_{i_1}, \ldots, B_{i_t}$ from S covering all elements from A'. Set $Q = Q^0 \cup \{B_{i_1}, \ldots, B_{i_t}\}$. It is clear that Q is a 0-cover for (A, S), and $|Q| \leq |Q_\alpha|$. □

Corollary 1.16. *Let* $\alpha \in \mathbb{R}$ *and* $0 < \alpha < 1$. *Then*

$$C_{\min}(0, A, S) = C_{\min}(\alpha, A_\alpha, S_\alpha) \ .$$

Proof. From Lemma 1.14 it follows that $C_{\min}(\alpha, A_\alpha, S_\alpha) \leq C_{\min}(0, A, S)$. From Lemma 1.15 it follows that $C_{\min}(0, A, S) \leq C_{\min}(\alpha, A_\alpha, S_\alpha)$. □

Lemma 1.17. *Let* α, b *and* δ *be real numbers such that* $0 < \alpha < 1$, $b > 0$ *and* $\delta > 0$, *and let there exist a polynomial algorithm* $\mathcal{A}$ *that, for a given set cover problem* (A, S), *constructs an* α*-cover which cardinality is at most* $b \ln |A| C_{\min}(\alpha, A, S)$. *Then there exists a polynomial algorithm* $\mathcal{B}$ *that, for a given set cover problem* (A, S), *constructs a* 0*-cover which cardinality is at most* $(b + \delta) \ln |A| C_{\min}(0, A, S)$.

Proof. Let us describe the work of the algorithm $\mathcal{B}$. Let $\beta = 1 + \alpha/(1 - \alpha)$ and $a = \max\{1/b, b \ln \beta / \delta\}$. If $\ln |A| \leq a$, then, in polynomial time, we construct all subfamilies of S, which cardinality is at most $|A|$, and find among them a 0-cover for (A, S) with minimal cardinality. It is clear that the cardinality of this 0-cover is equal to $C_{\min}(0, A, S)$.

Let $\ln |A| > a$. Then $b \ln |A| > 1$, $(b + \delta) \ln |A| > 1$ and

$$\delta \ln |A| > b \ln \beta \ . \tag{1.2}$$

In polynomial time, we construct the problem (A_α, S_α), and apply to this problem the polynomial algorithm $\mathcal{A}$. As a result, we obtain an α-cover Q_α for (A_α, S_α) such that $|Q_\alpha| \leq b \ln |A_\alpha| C_{\min}(\alpha, A_\alpha, S_\alpha)$.

It is clear that $|A_\alpha| \leq |A| \beta$. By Corollary 1.16, $C_{\min}(\alpha, A_\alpha, S_\alpha) = C_{\min}(0, A, S)$. Therefore, $|Q_\alpha| \leq b(\ln |A| + \ln \beta) C_{\min}(0, A, S)$.

From (1.2) we obtain $b(\ln |A| + \ln \beta) = (b+\delta) \ln |A| - \delta \ln |A| + b \ln \beta \leq (b+\delta) \ln |A|$. Therefore, $|Q_\alpha| \leq (b + \delta) \ln |A| C_{\min}(0, A, S)$. From Lemma 1.15 we conclude that, in polynomial time, we can construct a 0-cover Q for (A, S) such that $|Q| \leq (b + \delta) \ln |A| C_{\min}(0, A, S)$. □

We now generalize Theorem 1.6 to the case of partial covers.

Theorem 1.18. *Let* $\alpha \in \mathbb{R}$ *and* $0 \leq \alpha < 1$. *If* $NP \not\subseteq DTIME(n^{O(\log \log n)})$, *then for any* ε, $0 < \varepsilon < 1$, *there is no polynomial algorithm that for a given set cover problem* (A, S) *constructs an* α*-cover for* (A, S) *which cardinality is at most* $(1 - \varepsilon) C_{\min}(\alpha, A, S) \ln |A|$.

Proof. If $\alpha = 0$, then the statement of the theorem coincides with Theorem 1.6. Let $\alpha > 0$. Let us assume that the considered statement does not hold: let $NP \not\subseteq$

$DTIME(n^{O(\log\log n)})$ and for some ε, $0 < \varepsilon < 1$, there exist a polynomial algorithm $\mathcal{A}$ that, for a given set cover problem (A, S), constructs an α-cover for (A, S) which cardinality is at most $(1 - \varepsilon)C_{\min}(\alpha, A, S) \ln |A|$.

Applying Lemma 1.17 with parameters $b = (1 - \varepsilon)$ and $\delta = \varepsilon/2$ we conclude that, under the assumption $NP \not\subseteq DTIME(n^{O(\log\log n)})$, there exists a polynomial algorithm $\mathcal{B}$ that, for a given set cover problem (A, S), constructs a 0-cover for (A, S) which cardinality is at most $(1 - \varepsilon/2)C_{\min}(0, A, S) \ln |A|$. The last statement contradicts Theorem 1.6. □

From Theorem 1.10 it follows that $C_{\text{greedy}}(\alpha) \leq C_{\min}(\alpha)(1 + \ln |A|)$. From this inequality and from Theorem 1.18 it follows that, under the assumption $NP \not\subseteq DTIME(n^{O(\log\log n)})$, the greedy algorithm is close to the best polynomial approximate algorithms for partial cover minimization.

We now generalize Theorem 1.7 to the case of partial covers.

Theorem 1.19. *Let $\alpha \in \mathbb{R}$ and $0 \leq \alpha < 1$. If $P \neq NP$, then there exists $\varrho > 0$ such that there is no polynomial algorithm that for a given set cover problem (A, S) constructs an α-cover for (A, S) which cardinality is at most $\varrho C_{\min}(\alpha, A, S) \ln |A|$.*

Proof. If $\alpha = 0$, then the statement of the theorem coincides with Theorem 1.7. Let $\alpha > 0$. We will now show that in the capacity of ϱ we can take the number $\gamma/2$, where γ is the constant from Theorem 1.7. Let us assume the contrary: let $P \neq NP$, and a polynomial algorithm $\mathcal{A}$ exist that, for a given set cover problem (A, S), constructs an α-cover for (A, S) which cardinality is at most $(\gamma/2)C_{\min}(\alpha, A, S) \ln |A|$.

Applying Lemma 1.17 with parameters $b = \gamma/2$ and $\delta = \gamma/2$ we conclude that, under the assumption $P \neq NP$, there exists a polynomial algorithm $\mathcal{B}$ that, for a given set cover problem (A, S), constructs a 0-cover for (A, S) which cardinality is at most $\gamma C_{\min}(0, A, S) \ln |A|$. The last statement contradicts Theorem 1.7. □

1.1.4 Bounds on $C_{\min}(\alpha)$ Based on Information About Greedy Algorithm Work

Using information on the greedy algorithm work we can obtain bounds on $C_{\min}(\alpha)$. We consider now two simple examples. It is clear that $C_{\min}(\alpha) \leq C_{\text{greedy}}(\alpha)$. From Theorem 1.10 it follows that $C_{\text{greedy}}(\alpha) \leq C_{\min}(\alpha)(1 + \ln |A|)$. Therefore, $C_{\min}(\alpha) \geq C_{\text{greedy}}(\alpha)/(1 + \ln |A|)$. Another lower bounds on $C_{\min}(\alpha)$ can be obtained based on Theorems 1.8 and 1.12.

In this subsection, we fix some information on the greedy algorithm work, and find the best upper and lower bounds on $C_{\min}(\alpha)$ depending on this information.

Information on Greedy Algorithm Work

Let us assume that (A, S) is a set cover problem and α is a real number such that $0 \le \alpha < 1$. We now apply the greedy algorithm with threshold α to the problem (A, S). Let us assume that during the construction of α-cover the greedy algorithm chooses consequently subsets $B_{j_1}, \ldots, B_{j_t}$. Set $B_{j_0} = \emptyset$ and for $i = 1, \ldots, t$ set $\delta_i = |B_{j_i} \setminus (B_{j_0} \cup \ldots \cup B_{j_{i-1}})|$.

Write $\Delta(\alpha, A, S) = (\delta_1, \ldots, \delta_t)$. As information on the greedy algorithm work we will use the tuple $\Delta(\alpha, A, S)$ and numbers $|A|$ and α. Note that $\delta_1 = \max\{|B_i| : B_i \in S\}$ and $t = C_{\text{greedy}}(\alpha, A, S)$. Let us denote by P_{SC} the set of set cover problems and $D_{SC} = \{(\alpha, |A|, \Delta(\alpha, A, S)) : \alpha \in \mathbb{R}, 0 \le \alpha < 1, (A, S) \in P_{SC}\}$.

Lemma 1.20. *A tuple $(\alpha, n, (\delta_1, \ldots, \delta_t))$ belongs to the set D_{SC} if and only if α is a real number such that $0 \le \alpha < 1$, and $n, \delta_1, \ldots, \delta_t$ are natural numbers such that $\delta_1 \ge \ldots \ge \delta_t$, $\sum_{i=1}^{t-1} \delta_i < (1-\alpha)n$ and $(1-\alpha)n \le \sum_{i=1}^{t} \delta_i \le n$.*

Proof. Let $(\alpha, n, (\delta_1, \ldots, \delta_t)) \in D_{SC}$ and $(\alpha, n, (\delta_1, \ldots, \delta_t)) = (\alpha, |A|, \Delta(\alpha, A, S))$. It is clear that α is a real number, $0 \le \alpha < 1$, and $n, \delta_1, \ldots, \delta_t$ are natural numbers. From the definition of greedy algorithm it follows that $\delta_1 \ge \ldots \ge \delta_t$. Taking into account that α is the threshold for the greedy algorithm we obtain $\sum_{i=1}^{t-1} \delta_i < (1-\alpha)n$ and $(1-\alpha)n \le \sum_{i=1}^{t} \delta_i \le n$.

Let $(\alpha, n, (\delta_1, \ldots, \delta_t))$ be a tuple for which α is a real number such that $0 \le \alpha < 1$, and $n, \delta_1, \ldots, \delta_t$ are natural numbers such that $\delta_1 \ge \ldots \ge \delta_t$, $\sum_{i=1}^{t-1} \delta_i < (1-\alpha)n$ and $(1-\alpha)n \le \sum_{i=1}^{t} \delta_i \le n$. We define a set cover problem (A, S) in the following way: $A = \{a_1, \ldots, a_n\}$ and $S = \{\{a_1, \ldots, a_{\delta_1}\}, \ldots, \{a_{\delta_1+\ldots+\delta_{t-1}+1}, \ldots, a_{\delta_1+\ldots+\delta_t}\}, \{a_{\delta_1+\ldots+\delta_t+1}\}, \ldots, \{a_n\}\}$ (for simplicity, we omit here notation $B_1 = \{a_1, \ldots, a_{\delta_1}\}, \ldots$). It is not difficult to show that $\Delta(\alpha, A, S) = (\delta_1, \ldots, \delta_t)$. Thus, $(\alpha, n, (\delta_1, \ldots, \delta_t)) \in D_{SC}$. □

The Best Upper Bound on $C_{\min}(\alpha)$

We define a function $\mathcal{U}_{SC} : D_{SC} \to \mathbb{N}$. Let $(\alpha, n, (\delta_1, \ldots, \delta_t)) \in D_{SC}$. Then $\mathcal{U}_{SC}(\alpha, n, (\delta_1, \ldots, \delta_t)) = \max\{C_{\min}(\alpha, A, S) : (A, S) \in P_{SC}, |A| = n, \Delta(\alpha, A, S) = (\delta_1, \ldots, \delta_t)\}$. It is clear that

$$C_{\min}(\alpha, A, S) \le \mathcal{U}_{SC}(\alpha, |A|, \Delta(\alpha, A, S))$$

is the best upper bound on $C_{\min}(\alpha)$ depending on α, $|A|$ and $\Delta(\alpha, A, S)$.

Theorem 1.21. *Let $(\alpha, n, (\delta_1, \ldots, \delta_t)) \in D_{SC}$. Then $\mathcal{U}_{SC}(\alpha, n, (\delta_1, \ldots, \delta_t)) = t$.*

Proof. Let us consider an arbitrary set cover problem (A, S) such that $|A| = n$ and $\Delta(\alpha, A, S) = (\delta_1, \ldots, \delta_t)$. It is clear that $C_{\min}(\alpha, A, S) \leq C_{\text{greedy}}(\alpha, A, S)$. Since $C_{\text{greedy}}(\alpha, A, S) = t$, we have $\mathcal{U}(\alpha, n, (\delta_1, \ldots, \delta_t)) \leq t$.

We now consider the set cover problem (A, S): $A = \{a_1, \ldots, a_n\}$ and $S = \{\{a_1, \ldots, a_{\delta_1}\}, \ldots, \{a_{\delta_1+\ldots+\delta_{t-1}+1}, \ldots, a_{\delta_1+\ldots+\delta_t}\}, \{a_{\delta_1+\ldots+\delta_t+1}\}, \ldots, \{a_n\}\}$ (we omit here notation $B_1 = \{a_1, \ldots, a_{\delta_1}\}, \ldots$). It is clear that $|A| = n$. Lemma 1.20 now shows that $\Delta(\alpha, A, S) = (\delta_1, \ldots, \delta_t)$. Taking into account that all subsets from S are pairwise disjoint it is not difficult to prove that $C_{\min}(\alpha, A, S) = C_{\text{greedy}}(\alpha, A, S) = t$. Therefore, $\mathcal{U}_{SC}(\alpha, n, (\delta_1, \ldots, \delta_t)) \geq t$. □

Thus, $C_{\min}(\alpha, A, S) \leq C_{\text{greedy}}(\alpha, A, S)$ is the best upper bound on $C_{\min}(\alpha)$ depending on α, $|A|$ and $\Delta(\alpha, A, S)$.

The Best Lower Bound on $C_{\min}(\alpha)$

We define a function $\mathcal{L}_{SC} : D_{SC} \to \mathbb{N}$. Let $(\alpha, n, (\delta_1, \ldots, \delta_t)) \in D_{SC}$. Then $\mathcal{L}_{SC}(\alpha, n, (\delta_1, \ldots, \delta_t)) = \min\{C_{\min}(\alpha, A, S) : (A, S) \in P_{SC}, |A| = n, \Delta(\alpha, A, S) = (\delta_1, \ldots, \delta_t)\}$. It is clear that

$$C_{\min}(\alpha, A, S) \geq \mathcal{L}_{SC}(\alpha, |A|, \Delta(\alpha, A, S))$$

is the best lower bound on $C_{\min}(\alpha)$ depending on α, $|A|$ and $\Delta(\alpha, A, S)$. For $(\alpha, n, (\delta_1, \ldots, \delta_t)) \in D_{SC}$ and $\delta_0 = 0$, set

$$l(\alpha, n, (\delta_1, \ldots, \delta_t)) = \max\left\{\left\lceil \frac{\lceil (1-\alpha)n \rceil - (\delta_0 + \ldots + \delta_i)}{\delta_{i+1}} \right\rceil : i = 0, \ldots, t-1\right\}.$$

Theorem 1.22. *Let $(\alpha, n, (\delta_1, \ldots, \delta_t)) \in D_{SC}$. Then $\mathcal{L}_{SC}(\alpha, n, (\delta_1, \ldots, \delta_t)) = l(\alpha, n, (\delta_1, \ldots, \delta_t))$.*

Proof. Let us consider an arbitrary set cover problem (A, S) such that $|A| = n$ and $\Delta(\alpha, A, S) = (\delta_1, \ldots, \delta_t)$. Set $p = C_{\min}(\alpha, A, S)$. It is clear that there exist p subsets from S which cover a subset V of the set A such that $|V| \geq \lceil (1-\alpha)n \rceil$.

Let $i \in \{0, \ldots, t-1\}$. After i steps of the greedy algorithm work, at least $\lceil (1-\alpha)n \rceil - (\delta_0 + \ldots + \delta_i)$ elements from the set V are uncovered. Therefore, in the family S there is a subset which can cover at least $(\lceil (1-\alpha)n \rceil - (\delta_0 + \ldots + \delta_i))/p$ of uncovered elements. Thus, $\delta_{i+1} \geq (\lceil (1-\alpha)n \rceil - (\delta_0 + \ldots + \delta_i))/p$ and $p \geq (\lceil (1-\alpha)n \rceil - (\delta_0 + \ldots + \delta_i))/\delta_{i+1}$. Since p is a natural number, we have $p \geq \lceil (\lceil (1-\alpha)n \rceil - (\delta_0 + \ldots + \delta_i))/\delta_{i+1} \rceil$. Taking into account that i is an arbitrary number from $\{0, \ldots, t-1\}$ we obtain $C_{\min}(\alpha, A, S) \geq l(\alpha, n, (\delta_1, \ldots, \delta_t))$. Thus, $\mathcal{L}_{SC}(\alpha, n, (\delta_1, \ldots, \delta_t)) \geq l(\alpha, n, (\delta_1, \ldots, \delta_t))$.

Let us show that $\mathcal{L}_{SC}(\alpha, n, (\delta_1, \ldots, \delta_t)) \leq l(\alpha, n, (\delta_1, \ldots, \delta_t))$.

Write $d = l(\alpha, n, (\delta_1, \ldots, \delta_t))$, $r = \lceil (1-\alpha)n \rceil$ and $q = n - (\delta_1 + \ldots + \delta_t)$. Let us consider the following set cover problem (A, S): $A = \{a_1, \ldots, a_n\}$ and $S = \{B_1, \ldots, B_t, B_{t+1}, \ldots, B_{t+q}, B_{t+q+1}, \ldots, B_{t+q+d}\}$, where $B_1 = \{a_1, \ldots, a_{\delta_1}\}$, ..., $B_t = \{a_{\delta_1+\ldots+\delta_{t-1}+1}, \ldots, a_{\delta_1+\ldots+\delta_t}\}$, $B_{t+1} = \{a_{\delta_1+\ldots+\delta_t+1}\}$, ..., $B_{t+q} = \{a_n\}$. Let $D = \{a_1, \ldots, a_r\}$. For $j = 1, \ldots, d$, the set B_{t+q+j} includes all elements from the set D of the kind $a_{r-id-j+1}$, $i = 0, 1, 2, \ldots$, and only such elements.

It is clear that subsets $B_{t+q+1}, \ldots, B_{t+q+d}$ form an α-cover for (A, S). Therefore, $C_{\min}(\alpha, A, S) \leq d$.

We prove by induction on $j = 1, \ldots, t$ that, during the step number j, the greedy algorithm chooses the subset B_j from S. From Lemma 1.20 it follows that $\delta_1 \geq \ldots \geq \delta_t$.

Let us consider the first step of greedy algorithm. It is clear that the cardinality of B_1 is equal to δ_1, and δ_1 is greater than or equal to the cardinality of each of sets $B_2, \ldots, B_{t+q}$. Let us show that δ_1 is greater than or equal to the cardinality of each of sets $B_{t+q+1}, \ldots, B_{t+q+d}$. We have $\lceil r/\delta_1 \rceil \leq d$. Therefore, $r/\delta_1 \leq d$ and $r/d \leq \delta_1$. Let $r = sd + a$, where s is a nonnegative integer and $a \in \{0, 1, \ldots, d-1\}$. Then the cardinality of each of the sets $B_{t+q+1}, \ldots, B_{t+q+d}$ is equal to s if $a = 0$, and is at most $s+1$ if $a > 0$. From the inequality $r/d \leq \delta_1$ it follows that $\delta_1 \geq s$ if $a = 0$, and $\delta_1 \geq s+1$ if $a > 0$. So at the first step the greedy algorithm chooses the set B_1.

Let us assume that during j steps, $1 \leq j \leq t-1$, the greedy algorithm chooses the sets $B_1, \ldots, B_j$. Let us consider the step number $j+1$. It is clear that B_{j+1} covers δ_{j+1} uncovered elements. One can show that each set from $B_{j+2}, \ldots, B_{t+q}$ covers at most δ_{j+1} uncovered elements. Set $u = r - (\delta_1 + \ldots + \delta_j)$. Let $u = sd + a$, where s is a nonnegative integer and $a \in \{0, 1, \ldots, d-1\}$. One can show that each set from $B_{t+q+1}, \ldots, B_{t+q+d}$ covers at most s uncovered elements if $a = 0$, and at most $s+1$ uncovered elements if $a > 0$. It is clear that $\lceil u/\delta_{j+1} \rceil \leq d$. Therefore, $u/\delta_{j+1} \leq d$ and $u/d \leq \delta_{j+1}$. Hence, $\delta_{j+1} \geq s$ if $a = 0$, and $\delta_{j+1} \geq s+1$ if $a > 0$. So at the step number $j+1$ the greedy algorithm chooses the set B_{j+1}.

Since greedy algorithm chooses subsets $B_1, \ldots, B_t$, we have $\Delta(\alpha, A, S) = (\delta_1, \ldots, \delta_t)$. Therefore, $C_{\min}(\alpha) \geq d$. As it was proved earlier, $C_{\min}(\alpha) \leq d$. Hence, $C_{\min}(\alpha) = d$ and $\mathcal{L}_{SC}(\alpha, n, (\delta_1, \ldots, \delta_t)) \leq l(\alpha, n, (\delta_1, \ldots, \delta_t))$. Therefore, $\mathcal{L}_{SC}(\alpha, n, (\delta_1, \ldots, \delta_m)) = l(\alpha, n, (\delta_1, \ldots, \delta_t))$. □

So $C_{\min}(\alpha, A, S) \geq l(\alpha, |A|, \Delta(\alpha, A, S))$ is the best lower bound on $C_{\min}(\alpha)$ depending on α, $|A|$ and $\Delta(\alpha, A, S)$.

Properties of the Best Lower Bound on $C_{\min}(\alpha)$

Let us assume that (A,S) is a set cover problem and α is a real number such that $0 \le \alpha < 1$. Let

$$l_{SC}(\alpha) = l_{SC}(\alpha, A, S) = l(\alpha, |A|, \Delta(\alpha, A, S)) .$$

Lemma 1.23. *Let $\alpha_1, \alpha_2 \in \mathbb{R}$ and $0 \le \alpha_1 < \alpha_2 < 1$. Then $l_{SC}(\alpha_1) \ge l_{SC}(\alpha_2)$.*

Proof. Let $\Delta(\alpha_1, A, S) = (\delta_1, \ldots, \delta_{t_1})$ and $\Delta(\alpha_2, A, S) = (\delta_1, \ldots, \delta_{t_2})$. We have $t_1 \ge t_2$. Let $\delta_0 = 0$, $j \in \{0, \ldots, t_2 - 1\}$ and

$$\left\lceil \frac{\lceil |A|(1-\alpha_2) \rceil - (\delta_0 + \ldots + \delta_j)}{\delta_{j+1}} \right\rceil = l_{SC}(\alpha_2) .$$

It is clear that $l_{SC}(\alpha_1) \ge \lceil (\lceil |A|(1-\alpha_1) \rceil - (\delta_0 + \ldots + \delta_j))/\delta_{j+1} \rceil \ge l_{SC}(\alpha_2)$. □

Corollary 1.24. $l_{SC}(0) = \max\{l_{SC}(\alpha) : 0 \le \alpha < 1\}$.

The value $l_{SC}(\alpha)$ can be used for obtaining upper bounds on the cardinality of partial covers constructed by the greedy algorithm.

Theorem 1.25. *Let α and β be real numbers such that $0 < \beta \le \alpha < 1$. Then $C_{\text{greedy}}(\alpha) < l_{SC}(\alpha - \beta) \ln((1 - \alpha + \beta)/\beta) + 1$.*

Proof. Let $\Delta(\alpha - \beta, A, S) = (\delta_1, \ldots, \delta_t)$, $\delta_0 = 0$, $M = (1 - \alpha + \beta)|A|$ and $l = l_{SC}(\alpha - \beta)$. We have $l \ge 1$ and

$$l \ge \max\left\{ \frac{M - (\delta_0 + \ldots + \delta_i)}{\delta_{i+1}} : i = 0, \ldots, t-1 \right\} .$$

Therefore, for $i = 0, \ldots, t-1$, $(M - (\delta_0 + \ldots + \delta_i))/\delta_{i+1} \le l$ and

$$\frac{M - (\delta_0 + \ldots + \delta_i)}{l} \le \delta_{i+1} . \quad (1.3)$$

Let us assume that $l = 1$. Then $\delta_1 \ge M$ and $C_{\text{greedy}}(\alpha) = 1$. It is clear that $l_{SC}(\alpha - \beta) \ln((1 - \alpha + \beta)/\beta) > 0$. Therefore, if $l = 1$, then the statement of the theorem holds. Let $l \ge 2$. Let us show that for $j = 1, \ldots, t$,

$$M - (\delta_0 + \ldots + \delta_j) \le M \left(1 - \frac{1}{l}\right)^j . \quad (1.4)$$

For $i = 0$, from (1.3) it follows that $\delta_1 \ge M/l$. Therefore, (1.4) holds for $j = 1$. Let us assume that (1.4) holds for some j, $1 \le j \le t-1$. Let us show that

$$M - (\delta_0 + \ldots + \delta_{j+1}) \le M \left(1 - \frac{1}{l}\right)^{j+1} . \quad (1.5)$$

Write $Q = M - (\delta_0 + \ldots + \delta_j)$. For $i = j$, from (1.3) it follows that $\delta_{j+1} \geq Q/l$. Using this inequality and (1.4) we obtain $M - (\delta_0 + \ldots + \delta_{j+1}) \leq Q - Q/l \leq Q\,(1 - 1/l) \leq M\,(1 - 1/l)^{j+1}$. Therefore, (1.5) holds. Thus, (1.4) holds.

Let $C_{\text{greedy}}(\alpha) = p$. It is clear that $C_{\text{greedy}}(\alpha) \leq C_{\text{greedy}}(\alpha - \beta) = t$. Therefore, $p \leq t$. It is clear that $\delta_1 + \ldots + \delta_{p-1} < |A|(1 - \alpha)$. Using (1.4) we obtain $M - M\,(1 - 1/l)^{p-1} \leq \delta_1 + \ldots + \delta_{p-1}$. Therefore, $|A|(1 - \alpha + \beta) - |A|(1 - \alpha + \beta)\,(1 - 1/l)^{p-1} < |A|(1 - \alpha)$. Hence, $|A|\beta < |A|(1 - \alpha + \beta)\,(1 - 1/l)^{p-1} = |A|(1 - \alpha + \beta)\,((l - 1)/l)^{p-1}$ and $(l/(l - 1))^{p-1} < (1 - \alpha + \beta)/\beta$. If we take the natural logarithm of both sides of this inequality, we obtain $(p - 1)\ln(1 + 1/(l - 1)) < \ln((1 - \alpha + \beta)/\beta)$. Taking into account that $l - 1$ is a natural number, and using the inequality $\ln(1 + 1/r) > 1/(r + 1)$, which holds for any natural r, we obtain $\ln(1 + 1/(l - 1)) > 1/l$. Therefore, $C_{\text{greedy}}(\alpha) = p < l\ln((1 - \alpha + \beta)/\beta) + 1 = l_{SC}(\alpha - \beta)\ln((1 - \alpha + \beta)/\beta) + 1$. □

Corollary 1.26. *Let $\alpha \in \mathbb{R}$, $0 < \alpha < 1$. Then $C_{\text{greedy}}(\alpha) < l_{SC}(0)\ln(1/\alpha) + 1$.*

If $l_{SC}(0)$ is a small number, then we have a good upper bound on $C_{\text{greedy}}(\alpha)$. If $l_{SC}(0)$ is a big number, then we have a big lower bound on $C_{\min}(0)$ and on $C_{\min}(\alpha)$ for some α.

1.1.5 Upper Bound on $C_{\text{greedy}}(\alpha)$

In this subsection, we obtain one more upper bound on $C_{\text{greedy}}(\alpha)$ which does not depend on $|A|$, and show that, in some sense, this bound is unimprovable.

Theorem 1.27. *Let α and β be real numbers such that $0 < \beta \leq \alpha < 1$. Then $C_{\text{greedy}}(\alpha) < C_{\min}(\alpha - \beta)\ln((1 - \alpha + \beta)/\beta) + 1$.*

Proof. By Theorem 1.25, $C_{\text{greedy}}(\alpha) < l_{SC}(\alpha - \beta)\ln((1 - \alpha + \beta)/\beta) + 1$, and by Theorem 1.22, $l_{SC}(\alpha - \beta) \leq C_{\min}(\alpha - \beta)$. □

Let us show that obtained bound is, in some sense, unimprovable.

Lemma 1.28. *Let α be a real number, $0 \leq \alpha < 1$, $j \in \{0, \ldots, |A| - 1\}$ and $j/|A| \leq \alpha < (j + 1)/|A|$. Then $C_{\min}(\alpha) = C_{\min}(j/|A|)$ and $C_{\text{greedy}}(\alpha) = C_{\text{greedy}}(j/|A|)$.*

Proof. Taking into account that $j/|A| \leq \alpha$ we conclude that $C_{\min}(\alpha) \leq C_{\min}(j/|A|)$ and $C_{\text{greedy}}(\alpha) \leq C_{\text{greedy}}(j/|A|)$.

Let $Q = \{B_{i_1}, \ldots, B_{i_t}\}$ be an arbitrary α-cover for (A, S). Let $M = |B_{i_1} \cup \ldots \cup B_{i_t}|$. It is clear that $M \geq |A|(1 - \alpha)$. Therefore, $1 - M/|A| \leq \alpha$. Taking into account that $\alpha < (j + 1)/|A|$ we obtain $|A| - M < j + 1$. Hence, $|A| - M \leq j$ and $|A| - j \leq M$. Therefore, $M \geq |A|(1 - j/|A|)$, and Q is also an $(j/|A|)$-cover. Thus, each α-cover is an $(j/|A|)$-cover. Using this fact it is not difficult to show that $C_{\min}(\alpha) \geq C_{\min}(j/|A|)$ and $C_{\text{greedy}}(\alpha) \geq C_{\text{greedy}}(j/|A|)$. □

Theorem 1.29. *There is no real $\delta < 1$ such that for any set cover problem (A, S) and for any real α and β, $0 < \beta \le \alpha < 1$, the following inequality holds:*

$$C_{\text{greedy}}(\alpha) \le \delta \left(C_{\min}(\alpha - \beta) \ln \left(\frac{1 - \alpha + \beta}{\beta} \right) + 1 \right) . \tag{1.6}$$

Proof. We assume the contrary: let such δ exist. We now consider an arbitrary α, $0 < \alpha < 1$, and an arbitrary set cover problem (A, S). Let $j \in \{0, \dots, |A| - 1\}$ and $j/|A| \le \alpha < (j+1)/|A|$. Using (1.6) we obtain

$$\begin{aligned} C_{\text{greedy}} \left(\frac{j}{|A|} + \frac{1}{2|A|} \right) &\le \delta \left(C_{\min} \left(\frac{j}{|A|} \right) \ln \left(\frac{1 - \frac{j}{|A|} - \frac{1}{2|A|} + \frac{1}{2|A|}}{\frac{1}{2|A|}} \right) + 1 \right) \\ &= \delta \left(C_{\min} \left(\frac{j}{|A|} \right) \ln \left(|A| - j \right) + C_{\min} \left(\frac{j}{|A|} \right) \ln 2 + 1 \right) . \end{aligned}$$

Lemma 1.28 now shows $C_{\text{greedy}}\left(j/|A| + j/(2|A|)\right) = C_{\text{greedy}}\left(j/|A|\right) = C_{\text{greedy}}\left(\alpha\right)$ and $C_{\min}(j/|A|) = C_{\min}(\alpha)$. Let us evaluate the number $|A| - j$. We have $j \le \alpha|A| < j + 1$. Therefore, $|A| - j - 1 < |A| - \alpha|A| \le |A| - j$ and $|A| - j = \lceil (1 - \alpha)|A| \rceil$. Finally, we have

$$C_{\text{greedy}}(\alpha) \le \delta \left(C_{\min}\left(\alpha\right) \ln \left(\lceil (1 - \alpha)|A| \rceil \right) + C_{\min}\left(\alpha\right) \ln 2 + 1 \right) . \tag{1.7}$$

Using Theorem 1.9 we conclude that for any natural $t \ge 2$ there exists a set cover problem (A_t, S_t) such that $\lceil (1 - \alpha)|A_t| \rceil = t$ and $C_{\text{greedy}}(\alpha, A_t, S_t) > C_{\min}(\alpha, A_t, S_t)(\ln t - \ln \ln t - 0.31)$. Let $C_t = C_{\min}(\alpha, A_t, S_t)$. Using (1.7) we obtain for any $t \ge 2$, $C_t(\ln t - \ln \ln t - 0.31) < \delta(C_t \ln t + C_t \ln 2 + 1)$. If we divide both sides of this inequality by $C_t \ln t$, we obtain

$$1 - \frac{\ln \ln t}{\ln t} - \frac{0.31}{\ln t} < \delta + \frac{\delta \ln 2}{\ln t} + \frac{\delta}{C_t \ln t} .$$

It is clear that $C_t \ge 1$. Therefore, with growth of t the left-hand side of this inequality tends to 1, and the right-hand side of this inequality tends to δ, which is impossible. □

1.1.6 Covers for the Most Part of Set Cover Problems

In this subsection, covers for the most part of set cover problem are discussed from theoretical and experimental points of view. In particular, we obtain some theoretical and experimental confirmations of the following informal 0.5-hypothesis for covers: for the most part of set cover problems, during each step the greedy algorithm chooses a subset which covers at least one-half of uncovered elements.

We assume that (A, S) is a set cover problem, the elements of A are enumerated by numbers $1, \ldots, n$, and sets from S are enumerated by numbers $1, \ldots, m$. It is possible that sets from S with different numbers are equal. There is a one-to-one correspondence between such set cover problems and tables with n rows and m columns filled by numbers from $\{0, 1\}$ and having no rows filled only by 0. Let $A = \{a_1, \ldots, a_n\}$ and $S = \{B_1, \ldots, B_m\}$. Then the problem (A, S) corresponds to the table which, for $i = 1, \ldots, n$ and $j = 1, \ldots, m$, has 1 at the intersection of i-th row and j-th column if and only if $a_i \in B_j$.

A table filled by numbers from $\{0, 1\}$ will be called *SC-table* if this table has no rows filled only by 0.

Lemma 1.30. *The number of SC-tables with n rows and m columns is at least $2^{mn} - 2^{mn-m+\log_2 n}$.*

Proof. Let $i \in \{1, \ldots, n\}$. The number of tables, in which the i-th row is filled by 0 only, is equal to 2^{mn-m}. Therefore, the number of tables, which are not SC-tables, is at most $n2^{mn-m} = 2^{mn-m+\log_2 n}$. Thus, the number of SC-tables is at least $2^{mn} - 2^{mn-m+\log_2 n}$. □

Exact Covers for the Most Part of Set Cover Problems

First, we study exact covers for the most part of set cover problems such that $m \geq \lceil \log_2 n \rceil + t$ and t is large enough.

Theorem 1.31. *Let us consider set cover problems (A, S) such that $A = \{a_1, \ldots, a_n\}$, $S = \{B_1, \ldots, B_m\}$ and $m \geq \lceil \log_2 n \rceil + t$, where t is a natural number. Let $i_1, \ldots, i_{\lceil \log_2 n \rceil + t}$ be pairwise different numbers from $\{1, \ldots, m\}$. Then the fraction of set cover problems (A, S), for which $\{B_{i_1}, \ldots, B_{i_{\lceil \log_2 n \rceil + t}}\}$ is an exact cover for (A, S), is at least $1 - 1/(2^t - 1)$.*

Proof. Let $k = \lceil \log_2 n \rceil + t$. The analyzed fraction is equal to the fraction of SC-tables with n rows and m columns which have no rows with only 0 at the intersection with columns $i_1, \ldots, i_k$. Such SC-tables will be called correct.

Let $j \in \{1, \ldots, t\}$. The number of tables with n rows and m columns filled by 0 and 1, in which the j-th row has only 0 at the intersection with columns $i_1, \ldots, i_k$, is equal to 2^{mn-k}. Therefore, the number of SC-tables, which are not correct, is at most $n2^{mn-k} = 2^{mn-k+\log_2 n}$. Using Lemma 1.30 we conclude that the fraction of correct SC-tables is at least

$$1 - \frac{2^{mn-k+\log_2 n}}{2^{mn} - 2^{mn-m+\log_2 n}} = 1 - \frac{1}{2^{k-\log_2 n} - 2^{k-m}} \geq 1 - \frac{1}{2^t - 1} \, .$$

□

For example, if $t = 7$, then for at least 99% of set cover problems (A, S) the subsets $B_{i_1}, \dots, B_{i_{\lceil \log_2 n \rceil + t}}$ form an exact cover for (A, S).

So if $m \geq \lceil \log_2 n \rceil + t$ and t is large enough, then for the most part of set cover problems there exist exact (and, consequently, partial) covers with small cardinality.

Partial Covers Constructed by Greedy Algorithm for the Most Part of Set Cover Problems

We now study the behavior of greedy algorithm for the most part of set cover problems such that $m \geq n + t$ and t is large enough.

Let us consider set cover problems (A, S) such that $A = \{a_1, \dots, a_n\}$ and $S = \{B_1, \dots, B_m\}$. A problem (A, S) will be called *saturated* if for any nonempty subset A' of A there exists a subset B_i from S which covers at least one-half of elements from A'. For a saturated set cover problem, the greedy algorithm at each step chooses a subset which covers at least one-half of uncovered elements. So for saturated set cover problems the 0.5-hypothesis is true.

Let us evaluate the number of saturated set cover problems. First, we prove an auxiliary statement.

Lemma 1.32. *Let k be a natural number and $\sigma \in \{0, 1\}$. Then the number of k-tuples from $\{0, 1\}^k$, in which the number of σ is less than $k/2$, is at most 2^{k-1}.*

Proof. Let k be even. Then the number of k-tuples from $\{0, 1\}^k$, in which the number of σ is less than $k/2$, is equal to $C_k^0 + \ldots + C_k^{k/2-1}$ that is less than 2^{k-1}. Let k be odd. Then the number of k-tuples from $\{0, 1\}^k$, in which the number of σ is less than $k/2$, is equal to $C_k^0 + \ldots + C_k^{\lfloor k/2 \rfloor}$ that is equal to 2^{k-1}. □

A table with n rows and m columns filled by numbers from $\{0, 1\}$ will be called *saturated* if for any $k \in \{1, \dots, n\}$, for any k rows there exists a column which has at least $k/2$ one's at the intersection with considered rows. Otherwise, the table will be called *unsaturated*.

Theorem 1.33. *Let us consider set cover problems (A, S) such that $A = \{a_1, \dots, a_n\}$, $S = \{B_1, \dots, B_m\}$ and $m > n$. Then the fraction of saturated set cover problems (A, S) is at least $1 - 1/(2^{m-n} - 1)$.*

Proof. It is clear that the analyzed fraction is equal to the fraction of saturated SC-tables.

Let us consider tables with n rows and m columns filled by numbers from $\{0, 1\}$. Let $k \in \{1, \dots, n\}$ and $i_1, \dots, i_k$ be pairwise different numbers from $\{1, \dots, n\}$. We now evaluate the number of tables in which at the intersection of each column with rows $i_1, \ldots, i_k$ the number of one's is less than $k/2$. Such tables will be called unsaturated in rows $i_1, \ldots, i_k$.

From Lemma 1.32 it follows that the number of k-tuples from $\{0,1\}^k$, in which the number of one's is less than $k/2$, is at most 2^{k-1}. Therefore, the number of tables, which are unsaturated in rows $i_1, \ldots, i_k$, is at most 2^{mn-m}.

There are 2^n different subsets of rows. Therefore, the number of unsaturated tables is at most 2^{mn+n-m}. Using Lemma 1.30 we conclude that the fraction of saturated SC-tables is at least

$$1 - \frac{2^{mn+n-m}}{2^{mn} - 2^{mn-m+\log_2 n}} = 1 - \frac{1}{2^{m-n} - 2^{\log_2 n - n}} \geq 1 - \frac{1}{2^{m-n} - 1} \; .$$

□

For example, if $m = n + 7$, then at least 99% of set cover problems are saturated. Let us analyze the work of greedy algorithm on an arbitrary saturated set cover problem (A, S). For $i = 1, 2, \ldots$, after the step number i at most $|A|/2^i$ elements from A are uncovered. We now evaluate the number $C_{\text{greedy}}(\alpha)$, where $0 < \alpha < 1$. It is clear that $C_{\text{greedy}}(\alpha) \leq i$, where i is a number such that $1/2^i \leq \alpha$. One can show that $1/2^{\lceil \log_2(1/\alpha) \rceil} \leq \alpha$. Therefore, $C_{\text{greedy}}(\alpha) \leq \lceil \log_2(1/\alpha) \rceil$. Some examples can be found in Table 1.1.

Table 1.1. Values of $\lceil \log_2(1/\alpha) \rceil$ for some α

α	0.5	0.3	0.1	0.01	0.001
Percentage of covered elements	50	70	90	99	99.9
$\lceil \log_2(1/\alpha) \rceil$	1	2	4	7	10

Let us evaluate the number $C_{\text{greedy}}(0)$. It is clear that all elements from A will be covered after a step number i if $|A|/2^i < 1$, i.e., if $i > \log_2 |A|$. If $\log_2 |A|$ is an integer, we can set $i = \log_2 |A| + 1$. Otherwise, we can set $i = \lceil \log_2 |A| \rceil$. Therefore, $C_{\text{greedy}}(0) \leq \log_2 |A| + 1$.

We now evaluate the number $l_{SC}(0)$. Let $\Delta(0, A, S) = (\delta_1, \ldots, \delta_m)$. Set $\delta_0 = 0$. Then $l_{SC}(0) = \max\{\lceil (|A| - (\delta_0 + \ldots + \delta_i))/\delta_{i+1} \rceil : i = 0, \ldots, m-1\}$. Since (A, S) is a saturated problem, we have $\delta_{i+1} \geq (|A| - (\delta_0 + \ldots + \delta_i))/2$ and $2 \geq (|A| - (\delta_0 + \ldots + \delta_i))/\delta_{i+1}$ for $i = 0, \ldots, m-1$. Therefore, $l_{SC}(0) \leq 2$. Using Corollary 1.24 we obtain $l_{SC}(\alpha) \leq 2$ for any α, $0 \leq \alpha < 1$.

Results of Experiments

We made some experiments with set cover problems (A, S) such that $|A| \in \{10, 50, 100, 1000, 3000, 5000\}$ and $|S| = 10$. For each value of $|A|$, we generated randomly 10 problems (A, S) such that for each element a_i from A and for each subset S_j from S the probability of a_i to be in S_j is equal to $1/2$. The results of experiments are represented in Tables 1.2 and 1.3.

In Table 1.2 the average percentage of elements covered at the i-th step of greedy algorithm is presented, $i = 1, \ldots, 10$. For example, 52.5 means that, on the average, 52.5% of elements remaining uncovered before i-th step are covered at i-th step.

Table 1.2. Average percentage of elements covered at i-th step of greedy algorithm

	Number of step i									
$\|A\|$	1	2	3	4	5	6	7	8	9	10
10	71.0	87.5	100.0							
50	62.4	67.5	80.1	100.0						
100	58.9	60.6	62.9	67.8	82.7	95.0	100.0			
1000	52.8	52.4	52.4	53.4	54.7	57.3	64.7	76.2	85.0	100.0
3000	51.2	51.5	52.5	52.6	53.6	54.2	56.9	61.2	72.3	100.0
5000	51.1	51.3	51.5	52.4	52.5	54.3	56.7	63.1	82.0	100.0

In Table 1.3 for each $\alpha \in \{0.1, 0.01, 0.001, 0.0\}$ the minimal, average and maximal cardinalities of α-covers constructed by the greedy algorithm are presented.

Table 1.3. Cardinalities of α-covers for set cover problems (A, S) with $|S| = 10$

	α											
$\|A\|$	0.1			0.01			0.001			0.0		
	min	avg	max	min	avg	max	min	avg	max	min	avg	max
10	2	2.0	2	2	2.4	3	2	2.4	3	2	2.4	3
50	2	2.6	3	4	4.0	4	4	4.0	4	4	4.0	4
100	3	3.0	3	5	5.5	7	5	5.5	7	5	5.5	7
1000	3	3.9	4	6	6.6	7	8	8.9	10	8	8.9	10
3000	4	4.0	4	6	6.9	7	8	9.0	10	9	9.9	10
5000	4	4.0	4	7	7.0	7	9	9.0	9	9	9.9	10

The obtained results show that for the most part of the considered set cover problems (not only for the case, where $|S| > |A|$) during each step the greedy algorithm chooses a subset which covers at least one-half of uncovered elements.

It must be also noted that with increase of step number the percentage of elements, covered at this step, grows for the most part of the considered set cover problems.

1.2 Partial Decision Rules

This section consists of seven subsections. In Sect. 1.2.1, main notions are described. In Sect. 1.2.2, relationships between partial covers and partial decision rules are considered. In Sect. 1.2.3, generalizations of Slavík's results to the case of partial decision rules are given. In Sect. 1.2.4, polynomial approximate algorithms for partial decision rule minimization (construction of partial decision rule with minimal length) are studied. In Sect. 1.2.5, upper and lower bounds on minimal length of partial decision rules based on an information about greedy algorithm work are investigated. In Sect. 1.2.6, an upper bound on the length of partial decision rule constructed by greedy algorithm is considered. In Sect. 1.2.7, decision rules for the most part of binary decision tables are discussed from theoretical and experimental points of view.

1.2.1 Main Notions

We assume that T is a decision table with n rows labeled with nonnegative integers (decisions) and m columns labeled with attributes (names of attributes) $f_1, \ldots, f_m$. This table is filled by nonnegative integers (values of attributes).

Let $r = (b_1, \ldots, b_m)$ be a row of T labeled with a decision d. By $U(T, r)$ we denote the set of rows from T which are different (in at least one column) from r and are labeled with decisions different from d. We will say that an attribute f_i *separates* a row $r' \in U(T, r)$ from the row r if the rows r and r' have different numbers at the intersection with column f_i. The pair (T, r) will be called a *decision rule problem.*

Let $0 \le \alpha < 1$. A decision rule

$$(f_{i_1} = b_{i_1}) \wedge \ldots \wedge (f_{i_t} = b_{i_t}) \to d \qquad (1.8)$$

is called an α-*decision rule* for (T, r) if attributes $f_{i_1}, \ldots, f_{i_t}$ separate from r at least $(1 - \alpha)|U(T, r)|$ rows from $U(T, r)$ (such rules are also called *partial* decision rules). The number t is called the *length* of the considered decision rule. If $U(T, r) = \emptyset$, then for any $f_{i_1}, \ldots, f_{i_t} \in \{f_1, \ldots, f_m\}$ the rule (1.8) is an α-decision rule for (T, r). The rule (1.8) with empty left-hand side (when $t = 0$) is also an α-decision rule for (T, r).

For example, 0.01-decision rule means that we should separate from r at least 99% of rows from $U(T, r)$. Note that a 0-decision rule is an exact decision rule. By $L_{\min}(\alpha) = L_{\min}(\alpha, T, r)$ we denote the minimal length of α-decision rule for (T, r).

We now describe a greedy algorithm with threshold α which constructs an α-decision rule for (T, r) (see Algorithm 2).

Let us denote by $L_{\text{greedy}}(\alpha) = L_{\text{greedy}}(\alpha, T, r)$ the length of constructed α-decision rule for (T, r).

Algorithm 2: Greedy algorithm for partial decision rule construction

```
Input  : Decision table $T$ with conditional attributes $f_1, \dots, f_m$, row $r = (b_1, \dots, b_m)$ of $T$ labeled
         with the decision $d$, and real number $\alpha$, $0 \le \alpha < 1$.
Output: $\alpha$-decision rule for $(T, r)$.
$Q \longleftarrow \emptyset$;
while attributes from $Q$ separate from $r$ less than $(1-\alpha)|U(T,r)|$ rows from $U(T,r)$ do
    select $f_i \in \{f_1, \dots, f_m\}$ with minimal index $i$ such that $f_i$ separates from $r$ the maximal number
    of rows from $U(T,r)$ unseparated by attributes from $Q$;
    $Q \longleftarrow Q \cup \{f_i\}$;
end
return $\bigwedge_{f_i \in Q}(f_i = b_i) \to d$;
```

1.2.2 Relationships Between Partial Covers and Partial Decision Rules

Let T be a decision table with m columns labeled with attributes $f_1, \dots, f_m$, r be a row from T, and $U(T, r)$ be a nonempty set.

We correspond a set cover problem $(A(T, r), S(T, r))$ to the considered decision rule problem (T, r) in the following way: $A(T, r) = U(T, r)$ and $S(T, r) = \{B_1, \dots, B_m\}$, where $B_1 = U(T, r, f_1), \dots, B_m = U(T, r, f_m)$ and for $i = 1, \dots, m$ the set $U(T, r, f_i)$ coincides with the set of rows from $U(T, r)$ separated by the attribute f_i from the row r.

Let during the construction of an α-decision rule for (T, r) the greedy algorithm choose consequently attributes $f_{j_1}, \dots, f_{j_t}$. Set $U(T, r, f_{j_0}) = \emptyset$ and for $i = 1, \dots, t$ set $\delta_i = |U(T, r, f_{j_i}) \setminus (U(T, r, f_{j_0}) \cup \ldots \cup U(T, r, f_{j_{i-1}}))|$. Let $\Delta(\alpha, T, r) = (\delta_1, \dots, \delta_t)$. It is not difficult to prove the following statement.

Proposition 1.34. *Let α be a real number such that $0 \le \alpha < 1$. Then $|U(T, r)| = |A(T, r)|$, $\Delta(\alpha, T, r) = \Delta(\alpha, A(T, r), S(T, r))$, $L_{\min}(\alpha, T, r) = C_{\min}(\alpha, A(T, r), S(T, r))$, and $L_{\text{greedy}}(\alpha, T, r) = C_{\text{greedy}}(\alpha, A(T, r), S(T, r))$.*

Let (A, S) be a set cover problem, $A = \{a_1, \dots, a_n\}$ and $S = \{B_1, \dots, B_m\}$. We correspond a decision rule problem $(T(A, S), r(A, S))$ to the set cover problem (A, S) in the following way. The table $T(A, S)$ contains m columns labeled with attributes $f_1, \dots, f_m$ and $n + 1$ rows filled by numbers from $\{0, 1\}$. For $i = 1, \dots, n$ and $j = 1, \dots, m$, the number 1 stays at the intersection of i-th row and j-th column if and only if $a_i \in B_j$. The $(n+1)$-th row is filled by 0. The first n rows are labeled with the decision 0. The last row is labeled with the decision 1. Let us denote by $r(A, S)$ the last row of table $T(A, S)$. For $i \in \{1, \dots, n+1\}$, we denote by r_i the i-th row. It is not difficult to see that $U(T(A, S), r(A, S)) = \{r_1, \dots, r_n\}$. Let $i \in \{1, \dots, n\}$ and $j \in \{1, \dots, m\}$. One can show that the attribute f_j separates the row $r_{n+1} = r(A, S)$ from the row r_i if and only if $a_i \in B_j$. It is not difficult to prove the following statements.

Proposition 1.35. *Let* $\alpha \in \mathrm{I\!R}$, $0 \leq \alpha < 1$, *and* $\{i_1, \ldots, i_t\} \subseteq \{1, \ldots, m\}$. *Then* $(f_{i_1} = 0) \wedge \ldots \wedge (f_{i_t} = 0) \rightarrow 1$ *is an* α*-decision rule for* $(T(A,S), r(A,S))$ *if and only if* $\{B_{i_1}, \ldots, B_{i_t}\}$ *is an* α*-cover for* (A,S).

Proposition 1.36. *Let* $\alpha \in \mathrm{I\!R}$ *and* $0 \leq \alpha < 1$. *Then* $|U(T(A,S),\ r(A,S))| = |A|$, $L_{\min}(\alpha, T(A,S), r(A,S)) = C_{\min}(\alpha, A, S)$, $L_{\text{greedy}}(\alpha, T(A,S), r(A,S)) = C_{\text{greedy}}(\alpha, A, S)$ *and* $\Delta(\alpha, T(A,S), r(A,S)) = \Delta(\alpha, A, S)$.

Proposition 1.37. *There exists a polynomial algorithm which for a given set cover problem* (A,S) *constructs the decision rule problem* $(T(A,S), r(A,S))$.

1.2.3 Precision of Greedy Algorithm

The following three statements are simple corollaries of results of Slavík (see Theorems 1.8–1.10). Let T be a decision table with m columns labeled with attributes $f_1, \ldots, f_m$, and r be a row of T.

Theorem 1.38. *Let* $0 \leq \alpha < 1$ *and* $\lceil (1-\alpha)|U(T,r)| \rceil \geq 2$. *Then* $L_{\text{greedy}}(\alpha) < L_{\min}(\alpha)(\ln \lceil (1-\alpha)|U(T,r)| \rceil - \ln\ln \lceil (1-\alpha)|U(T,r)| \rceil + 0.78)$.

Proof. Let us denote $(A,S) = (A(T,r), S(T,r))$. From Proposition 1.34 it follows that $|A| = |U(T,r)|$. Therefore, $\lceil (1-\alpha)|A| \rceil \geq 2$. Using Theorem 1.8 we obtain $C_{\text{greedy}}(\alpha, A, S) < C_{\min}(\alpha, A, S)(\ln \lceil (1-\alpha)|A| \rceil - \ln\ln \lceil (1-\alpha)|A| \rceil + 0.78)$.

Using Proposition 1.34 we conclude that $L_{\text{greedy}}(\alpha) = C_{\text{greedy}}(\alpha, A, S)$ and $L_{\min}(\alpha) = C_{\min}(\alpha, A, S)$. Taking into account that $|A| = |U(T,r)|$ we conclude that the statement of the theorem holds. □

Theorem 1.39. *Let* $0 \leq \alpha < 1$. *Then for any natural* $t \geq 2$ *there exists a decision rule problem* (T,r) *such that* $\lceil (1-\alpha)|U(T,r)| \rceil = t$ *and* $L_{\text{greedy}}(\alpha) > L_{\min}(\alpha)(\ln \lceil (1-\alpha)|U(T,r)| \rceil - \ln\ln \lceil (1-\alpha)|U(T,r)| \rceil - 0.31)$.

Proof. From Theorem 1.9 it follows that for any natural $t \geq 2$ there exists a set cover problem (A,S) such that $\lceil (1-\alpha)|A| \rceil = t$ and $C_{\text{greedy}}(\alpha,A,S) > C_{\min}(\alpha,A,S)(\ln \lceil (1-\alpha)|A| \rceil - \ln\ln \lceil (1-\alpha)|A| \rceil - 0.31)$.

Let us consider the decision rule problem $(T,r) = (T(A,S), r(A,S))$. From Proposition 1.36 it follows that $|U(T,r)| = |A|$, $C_{\text{greedy}}(\alpha, A, S) = L_{\text{greedy}}(\alpha, T, r)$ and $C_{\min}(\alpha, A, S) = L_{\min}(\alpha, T, r)$. Hence, the statement of the theorem holds. □

Theorem 1.40. *Let* $0 \leq \alpha < 1$ *and* $U(T,r) \neq \emptyset$. *Then* $L_{\text{greedy}}(\alpha) \leq L_{\min}(\alpha)(1 + \ln(\max_{j \in \{1,\ldots,m\}} |U(T,r,f_j)|))$.

Proof. Let us consider the set cover problem $(A, S) = (A(T, r), S(T, r))$. The inequality $C_{\text{greedy}}(\alpha, A, S) \leq C_{\min}(\alpha, A, S)(1 + \ln(\max_{j\in\{1,\ldots,m\}} |U(T, r, f_j)|))$ follows from Theorem 1.10.

Using Proposition 1.34 we conclude that $C_{\text{greedy}}(\alpha,A,S)= L_{\text{greedy}}(\alpha)$ and $C_{\min}(\alpha,A,S)= L_{\min}(\alpha)$. Therefore, the statement of the theorem holds. □

1.2.4 Polynomial Approximate Algorithms

Theorem 1.41. *Let $0 \leq \alpha < 1$. Then the problem of construction of α-decision rule with minimal length is NP-hard.*

Proof. From Theorem 1.13 it follows that the problem of construction of α-cover with minimal cardinality is NP-hard. Using Propositions 1.35 and 1.37 we conclude that there exists a polynomial-time reduction of the problem of construction of α-cover with minimal cardinality to the problem of construction of α-decision rule with minimal length. □

Let us generalize Theorem 1.18 to the case of partial decision rules.

Theorem 1.42. *Let $\alpha \in \mathbb{R}$ and $0 \leq \alpha < 1$. If $NP \not\subseteq DTIME(n^{O(\log\log n)})$, then for any ε, $0 < \varepsilon < 1$, there is no polynomial algorithm that for a given decision rule problem (T, r) with $U(T, r) \neq \emptyset$ constructs an α-decision rule for (T, r) which length is at most $(1 - \varepsilon)L_{\min}(\alpha, T, r) \ln |U(T, r)|$.*

Proof. We assume the contrary: let $NP \not\subseteq DTIME(n^{O(\log\log n)})$ and for some ε, $0 < \varepsilon < 1$, a polynomial algorithm $\mathcal{A}$ exist that for a given decision rule problem (T, r) with $U(T, r) \neq \emptyset$ constructs an α-decision rule for (T, r) which length is at most $(1 - \varepsilon)L_{\min}(\alpha, T, r) \ln |U(T, r)|$.

Let (A, S) be an arbitrary set cover problem, $A = \{a_1, \ldots, a_n\}$ and $S = \{B_1, \ldots, B_m\}$. From Proposition 1.37 it follows that there exists a polynomial algorithm which for a given set cover problem (A, S) constructs the decision rule problem $(T(A, S), r(A, S))$. Let us apply this algorithm and construct the decision rule problem $(T, r) = (T(A, S), r(A, S))$. Let us apply to the decision rule problem (T, r) the algorithm $\mathcal{A}$. As a result we obtain an α-decision rule

$$(f_{i_1} = 0) \wedge \ldots \wedge (f_{i_t} = 0) \to 1$$

for (T, r) such that $t \leq (1 - \varepsilon)L_{\min}(\alpha, T, r) \ln |U(T, r)|$. From Proposition 1.35 it follows that $\{B_{i_1}, \ldots, B_{i_t}\}$ is an α-cover for (A, S). Using Proposition 1.36 we obtain $|A| = |U(T, r)|$ and $L_{\min}(\alpha, T, r) = C_{\min}(\alpha, A, S)$. Therefore, $t \leq (1 - \varepsilon)C_{\min}(\alpha, A, S) \ln |A|$.

Thus, under the assumption $NP \nsubseteq DTIME(n^{O(\log\log n)})$, there exists a polynomial algorithm that for a given set cover problem (A, S) constructs an α-cover for (A, S) which cardinality is at most $(1-\varepsilon)C_{\min}(\alpha, A, S)\ln|A|$, but this fact contradicts Theorem 1.18. □

From Theorem 1.40 it follows that $L_{\text{greedy}}(\alpha) \leq L_{\min}(\alpha)(1 + \ln|U(T, r)|)$. From this inequality and from Theorem 1.42 it follows that, under the assumption $NP \nsubseteq DTIME(n^{O(\log\log n)})$, the greedy algorithm is close to the best polynomial approximate algorithms for partial decision rule minimization.

Let us generalize Theorem 1.19 to the case of partial decision rules.

Theorem 1.43. *Let α be a real number such that $0 \leq \alpha < 1$. If $P \neq NP$, then there exists $\varrho > 0$ such that there is no polynomial algorithm that for a given decision rule problem (T, r) with $U(T, r) \neq \emptyset$ constructs an α-decision rule for (T, r) which length is at most $\varrho L_{\min}(\alpha, T, r)\ln|U(T, r)|$.*

Proof. We now show that in the capacity of such ϱ we can choose ϱ from Theorem 1.19. Let us assume that the considered statement does not hold: let $P \neq NP$ and a polynomial algorithm $\mathcal{A}$ exist that for a given decision rule problem (T, r) with $U(T, r) \neq \emptyset$ constructs an α-decision rule for (T, r) which length is at most $\varrho L_{\min}(\alpha, T, r)\ln|U(T, r)|$.

Let (A, S) be an arbitrary set cover problem, $A = \{a_1, \ldots, a_n\}$ and $S = \{B_1, \ldots, B_m\}$. From Proposition 1.37 it follows that there exists a polynomial algorithm which for a given set cover problem (A, S) constructs the decision rule problem $(T(A, S), r(A, S))$. Let us apply this algorithm and construct the decision rule problem $(T, r) = (T(A, S), r(A, S))$. Let us apply to the problem (T, r) the algorithm $\mathcal{A}$. As a result we obtain an α-decision rule

$$(f_{i_1} = 0) \wedge \ldots \wedge (f_{i_t} = 0) \rightarrow 1$$

for (T, r) such that $t \leq \varrho L_{\min}(\alpha, T, r)\ln|U(T, r)|$. From Proposition 1.35 it follows that $\{B_{i_1}, \ldots, B_{i_t}\}$ is an α-cover for (A, S). Using Proposition 1.36 we obtain $|A| = |U(T, r)|$ and $L_{\min}(\alpha, T, r) = C_{\min}(\alpha, A, S)$. Therefore, $t \leq \varrho C_{\min}(\alpha, A, S)\ln|A|$.

Thus, under the assumption $P \neq NP$, there exists a polynomial algorithm that for a given set cover problem (A, S) constructs an α-cover for (A, S) which cardinality is at most $\varrho C_{\min}(\alpha, A, S)\ln|A|$, but this fact contradicts Theorem 1.19. □

1.2.5 Bounds on $L_{\min}(\alpha)$ Based on Information About Greedy Algorithm Work

In this subsection, we fix some information on the greedy algorithm work and find the best upper and lower bounds on $L_{\min}(\alpha)$ depending on this information.

Information on Greedy Algorithm Work

We assume that (T, r) is a decision rule problem, where T is a decision table with m columns labeled with attributes $f_1, \ldots, f_m$, $U(T, r) \neq \emptyset$, and α is a real number such that $0 \leq \alpha < 1$. Let us apply the greedy algorithm with threshold α to the problem (T, r). Let during the construction of α-decision rule the greedy algorithm choose consequently attributes $f_{j_1}, \ldots, f_{j_t}$. Set $U(T, r, f_{j_0}) = \emptyset$ and for $i = 1, \ldots, t$ set $\delta_i = |U(T, r, f_{j_i}) \setminus (U(T, r, f_{j_0}) \cup \ldots \cup P(U(T, r, f_{j_{i-1}}))|$. Let $\Delta(\alpha, T, r) = (\delta_1, \ldots, \delta_t)$. As information on the greedy algorithm work we will use the tuple $\Delta(\alpha, T, r)$, and numbers $|U(T, r)|$ and α. Note that $\delta_1 = \max\{|U(T, r, f_i)| : i = 1, \ldots, m\}$ and $t = L_{\text{greedy}}(\alpha, T, r)$.

Let us denote by P_{DR} the set of decision rule problems (T, r) with $U(T, r) \neq \emptyset$, and $D_{DR} = \{(\alpha, |U(T, r)|, \Delta(\alpha, T, r)) : \alpha \in \mathbb{R}, 0 \leq \alpha < 1, (T, r) \in P_{DR}\}$.

Lemma 1.44. $D_{DR} = D_{SC}$.

Proof. Let α be a real number, $0 \leq \alpha < 1$ and $(T, r) \in P_{DR}$. By Proposition 1.34, $(\alpha, |U(T, r)|, \Delta(\alpha, T, r)) = (\alpha, |A(T, r)|, \Delta(\alpha, A(T, r), S(T, r)))$. Therefore, $D_{DR} \subseteq D_{SC}$.

Let α be a real number, $0 \leq \alpha < 1$ and $(A, S) \in P_{SC}$. By Proposition 1.36, $(\alpha, |A|, \Delta(\alpha, A, S)) = (\alpha, |U(T(A, S), r(A, S))|, \Delta(\alpha, T(A, S), r(A, S)))$. Therefore, $D_{SC} \subseteq D_{DR}$. □

Note that the set D_{SC} was described in Lemma 1.20.

The Best Upper Bound on $L_{\min}(\alpha)$

We define a function $\mathcal{U}_{DR} : D_{DR} \to \mathbb{N}$. Let $(\alpha, n, (\delta_1, \ldots, \delta_t)) \in D_{DR}$. Then $\mathcal{U}_{DR}(\alpha, n, (\delta_1, \ldots, \delta_t)) = \max\{L_{\min}(\alpha, T, r) : (T, r) \in P_{DR}, |U(T, r)| = n, \Delta(\alpha, T, r) = (\delta_1, \ldots, \delta_t)\}$. It is clear that

$$L_{\min}(\alpha, T, r) \leq \mathcal{U}_{DR}(\alpha, |U(T, r)|, \Delta(\alpha, T, r))$$

is the best upper bound on $L_{\min}(\alpha)$ depending on α, $|U(T, r)|$ and $\Delta(\alpha, T, r)$.

Theorem 1.45. *Let* $(\alpha, n, (\delta_1, \ldots, \delta_t)) \in D_{DR}$. *Then*

$$\mathcal{U}_{DR}(\alpha, n, (\delta_1, \ldots, \delta_t)) = t\ .$$

Proof. Let (T, r) be an arbitrary decision rule problem such that $|U(T, r)| = n$ and $\Delta(\alpha, T, r) = (\delta_1, \ldots, \delta_t)$. It is cleat that $L_{\min}(\alpha, T, r) \leq L_{\text{greedy}}(\alpha, T, r) = t$. Therefore, $\mathcal{U}_{DR}(\alpha, n, (\delta_1, \ldots, \delta_t)) \leq t$.

Let us show that $\mathcal{U}_{DR}(\alpha, n, (\delta_1, \ldots, \delta_t)) \geq t$. Using Lemma 1.44 we obtain $(\alpha, n, (\delta_1, \ldots, \delta_t)) \in D_{SC}$. From here and from Theorem 1.21 it follows that there exists a set cover problem (A, S) such that $|A| = n$, $\Delta(\alpha, A, S) = (\delta_1, \ldots, \delta_t)$ and $C_{\min}(\alpha, A, S) = t$. Let us consider the decision rule problem $(T, r) = (T(A, S), r(A, S))$. From Proposition 1.36 it follows that $|U(T, r)| = n$, $\Delta(\alpha, T, r) = (\delta_1, \ldots, \delta_t)$ and $L_{\min}(\alpha, T, r) = t$. Therefore, $\mathcal{U}_{DR}(\alpha, n, (\delta_1, \ldots, \delta_t)) \geq t$. □

Thus, $L_{\min}(\alpha, T, r) \leq L_{\text{greedy}}(\alpha, T, r)$ is the best upper bound on $L_{\min}(\alpha)$ depending on α, $|U(T, r)|$ and $\Delta(\alpha, T, r)$.

The Best Lower Bound on $L_{\min}(\alpha)$

We define a function $\mathcal{L}_{DR} : D_{DR} \to \mathbb{N}$. Let $(\alpha, n, (\delta_1, \ldots, \delta_t)) \in D_{DR}$. Then $\mathcal{L}_{DR}(\alpha, n, (\delta_1, \ldots, \delta_t)) = \min\{L_{\min}(\alpha, T, r) : (T, r) \in P_{DR}, |U(T, r)| = n, \Delta(\alpha, T, r) = (\delta_1, \ldots, \delta_t)\}$. It is clear that

$$L_{\min}(\alpha, T, r) \geq \mathcal{L}_{DR}(\alpha, |U(T, r)|, \Delta(\alpha, T, r))$$

is the best lower bound on $L_{\min}(\alpha)$ depending on α, $|U(T, r)|$ and $\Delta(\alpha, T, r)$.

Let $(\alpha, n, (\delta_1, \ldots, \delta_t)) \in D_{DR}$. We now remind the definition of parameter $l(\alpha, n, (\delta_1, \ldots, \delta_t))$. Set $\delta_0 = 0$. Then

$$l(\alpha, n, (\delta_1, \ldots, \delta_t)) = \max\left\{\left\lceil \frac{\lceil (1-\alpha)n \rceil - (\delta_0 + \ldots + \delta_i)}{\delta_{i+1}} \right\rceil : i = 0, \ldots, t-1\right\}.$$

Theorem 1.46. *Let* $(\alpha, n, (\delta_1, \ldots, \delta_t)) \in D_{DR}$. *Then*

$$\mathcal{L}_{DR}(\alpha, n, (\delta_1, \ldots, \delta_t)) = l(\alpha, n, (\delta_1, \ldots, \delta_t)) .$$

Proof. Let (T, r) be an arbitrary decision rule problem such that $|U(T, r)| = n$ and $\Delta(\alpha, T, r) = (\delta_1, \ldots, \delta_t)$. We consider now the set cover problem $(A, S) = (A(T, r), S(T, r))$. From Proposition 1.34 it follows that $|A| = n$ and $\Delta(\alpha, A, S) = (\delta_1, \ldots, \delta_t)$. Using Theorem 1.22 we obtain $C_{\min}(\alpha, A, S) \geq l(\alpha, n, (\delta_1, \ldots, \delta_t))$. By Proposition 1.34, $C_{\min}(\alpha, A, S) = L_{\min}(\alpha, T, r)$. Therefore, we have $L_{\min}(\alpha, T, r) \geq l(\alpha, n, (\delta_1, \ldots, \delta_t))$ and $\mathcal{L}_{DR}(\alpha, n, (\delta_1, \ldots, \delta_t)) \geq l(\alpha, n, (\delta_1, \ldots, \delta_t))$.

Let us show that $\mathcal{L}_{DR}(\alpha, n, (\delta_1, \ldots, \delta_t)) \leq l(\alpha, n, (\delta_1, \ldots, \delta_t))$. By Lemma 1.44, $(\alpha, n, (\delta_1, \ldots, \delta_t)) \in D_{SC}$. From here and from Theorem 1.22 it follows that there exists a set cover problem (A, S) such that $|A| = n$, $\Delta(\alpha, A, S) = (\delta_1, \ldots, \delta_t)$ and $C_{\min}(\alpha, A, S) = l(\alpha, n, (\delta_1, \ldots, \delta_t))$. Let us consider the decision rule problem $(T, r) = (T(A, S), r(A, S))$. From Proposition 1.36 it follows that $|U(T, r)| = n$, $\Delta(\alpha, T, r) = (\delta_1, \ldots, \delta_t)$ and $L_{\min}(\alpha, T, r) = l(\alpha, n, (\delta_1, \ldots, \delta_t))$. Therefore, $\mathcal{L}_{DR}(\alpha, n, (\delta_1, \ldots, \delta_t)) \leq l(\alpha, n, (\delta_1, \ldots, \delta_t))$. □

Thus, $L_{\min}(\alpha, T, r) \geq l(\alpha, |U(T, r)|, \Delta(\alpha, T, r))$ is the best lower bound on $L_{\min}(\alpha)$ depending on α, $|U(T, r)|$ and $\Delta(\alpha, T, r)$.

Properties of the Best Lower Bound on $L_{\min}(\alpha)$

We assume that (T, r) is a decision rule problem from P_{DR}, and $\alpha \in \mathbb{R}$, $0 \leq \alpha < 1$. Let

$$l_{DR}(\alpha) = l_{DR}(\alpha, T, r) = l(\alpha, |U(T, r)|, \Delta(\alpha, T, r)).$$

Lemma 1.47. *Let* $\alpha_1, \alpha_2 \in \mathbb{R}$ *and* $0 \leq \alpha_1 < \alpha_2 < 1$*. Then* $l_{DR}(\alpha_1) \geq l_{DR}(\alpha_2)$*.*

Proof. Let $\Delta(\alpha_1, T, r) = (\delta_1, \ldots, \delta_{t_1})$ and $\Delta(\alpha_2, T, r) = (\delta_1, \ldots, \delta_{t_2})$. It is clear that $t_1 \geq t_2$. Set $\delta_0 = 0$. Let $j \in \{0, \ldots, t_2 - 1\}$ and

$$\left\lceil \frac{\lceil |U(T, r)|(1 - \alpha_2) \rceil - (\delta_0 + \ldots + \delta_j)}{\delta_{j+1}} \right\rceil = l_{DR}(\alpha_2) \; .$$

It is clear that $l_{DR}(\alpha_1) \geq \lceil (\lceil |U(T, r)|(1 - \alpha_1) \rceil - (\delta_0 + \ldots + \delta_j))/\delta_{j+1} \rceil \geq l_{DR}(\alpha_2)$. □

Corollary 1.48. $l_{DR}(0) = \max\{l_{DR}(\alpha) : 0 \leq \alpha < 1\}$.

The value $l_{DR}(\alpha)$ can be used for obtaining upper bounds on the length of partial decision rules constructed by the greedy algorithm.

Theorem 1.49. *Let* α *and* β *be real numbers such that* $0 < \beta \leq \alpha < 1$*. Then* $L_{\text{greedy}}(\alpha) < l_{DR}(\alpha - \beta) \ln((1 - \alpha + \beta)/\beta) + 1$*.*

Proof. Let us denote $(A, S) = (A(T, r), S(T, r))$. From Theorem 1.25 it follows that $C_{\text{greedy}}(\alpha, A, S) < l_{SC}(\alpha - \beta, A, S) \ln((1 - \alpha + \beta)/\beta) + 1$. Using Proposition 1.34 one can show that $l_{DR}(\alpha - \beta) = l_{DR}(\alpha - \beta, T, r) = l_{SC}(\alpha - \beta, A, S)$. From Proposition 1.34 it follows that $L_{\text{greedy}}(\alpha) = L_{\text{greedy}}(\alpha, T, r) = C_{\text{greedy}}(\alpha, A, S)$. Therefore, the statement of the theorem holds. □

Corollary 1.50. *Let* $\alpha \in \mathbb{R}$*,* $0 < \alpha < 1$*. Then* $L_{\text{greedy}}(\alpha) < l_{DR}(0) \ln(1/\alpha) + 1$*.*

If $l_{DR}(0)$ is a small number, then we have a good upper bound on $L_{\text{greedy}}(\alpha)$. If $l_{DR}(0)$ is a big number, then we have a big lower bound on $L_{\min}(0)$ and on $L_{\min}(\alpha)$ for some α.

1.2.6 Upper Bound on $L_{\text{greedy}}(\alpha)$

We assume that (T, r) is a decision rule problem from P_{DR}. In this subsection, we obtain an upper bound on $L_{\text{greedy}}(\alpha) = L_{\text{greedy}}(\alpha, T, r)$, which does not depend on $|U(T, r)|$, and show that, in some sense, this bound is unimprovable.

Theorem 1.51. *Let α and β be real numbers such that $0 < \beta \leq \alpha < 1$. Then $L_{\text{greedy}}(\alpha) < L_{\min}(\alpha - \beta)\ln((1-\alpha+\beta)/\beta) + 1$.*

Proof. By Theorem 1.49, $L_{\text{greedy}}(\alpha) < l_{DR}(\alpha - \beta)\ln((1-\alpha+\beta)/\beta) + 1$, and by Theorem 1.46, $l_{DR}(\alpha - \beta) \leq L_{\min}(\alpha - \beta)$. □

Let us show that obtained bound is, in some sense, unimprovable.

Theorem 1.52. *There is no real $\delta < 1$ such that for any decision rule problem $(T, r) \in P_{DR}$ and for any real α and β, $0 < \beta \leq \alpha < 1$, the following inequality holds: $L_{\text{greedy}}(\alpha) \leq \delta\,(L_{\min}(\alpha - \beta)\ln((1-\alpha+\beta)/\beta) + 1)$.*

Proof. We assume the contrary: let such δ exist. We now consider an arbitrary set cover problem (A, S) and arbitrary real α and β such that $0 < \beta \leq \alpha < 1$. Set $(T, r) = (T(A, S), r(A, S))$. Then

$$L_{\text{greedy}}(\alpha, T, r) \leq \delta\left(L_{\min}(\alpha - \beta, T, r)\ln\left(\frac{1-\alpha+\beta}{\beta}\right) + 1\right) .$$

By Proposition 1.36, $L_{\text{greedy}}(\alpha, T, r) = C_{\text{greedy}}(\alpha, A, S)$ and $L_{\min}(\alpha - \beta, T, r) = C_{\min}(\alpha - \beta, A, S)$. Therefore, there exists real $\delta < 1$ such that for any set cover problem (A, S) and for any real α and β, $0 < \beta \leq \alpha < 1$, the inequality $C_{\text{greedy}}(\alpha, A, S) \leq \delta\,(C_{\min}(\alpha - \beta, A, S)\ln((1-\alpha+\beta)/\beta) + 1)$ holds, which contradicts Theorem 1.29. □

1.2.7 Decision Rules for the Most Part of Binary Decision Tables

In this subsection, decision rules for the most part of binary decision tables are discussed from theoretical and experimental points of view. In particular, we obtain some theoretical and experimental confirmations of the following informal 0.5-hypothesis for decision rules: for the most part of decision tables for each row, under the construction of partial decision rule, during each step the greedy algorithm chooses an attribute which separates from the considered row r at least one-half of unseparated rows that are different from r and have other decisions.

Tests and Local Tests for the Most Part of Binary Information Systems

A *binary information system I* is a table with n rows (corresponding to objects) and m columns labeled with attributes $f_1, \ldots, f_m$. This table is filled by numbers from $\{0, 1\}$ (values of attributes). For $j = 1, \ldots, n$, we denote by r_j the j-th row of table I.

A subset $\{f_{i_1}, \ldots, f_{i_k}\}$ of attributes is a *test* for the information system I if these attributes separate any two rows r_j and r_l, where $j, l \in \{1, \ldots, n\}$ and $j \neq l$.

Adding an arbitrary decision attribute to the considered information system I we obtain a decision table T. For $j = 1, \ldots, n$, let $r_j = (b_1^j, \ldots, b_m^j)$ and d_j be the decision attached to r_j. If $\{f_{i_1}, \ldots, f_{i_k}\}$ is a test for the information system I, then for any $j \in \{1, \ldots, n\}$ the rule $(f_{i_1} = b_{i_1}^j) \wedge \ldots \wedge (f_{i_k} = b_{i_k}^j) \rightarrow d_j$ is a 0-decision rule for (T, r_j).

Let $m \geq \lceil 2\log_2 n \rceil + t$, where t is a natural number. Let $i_1, \ldots, i_{\lceil 2\log_2 n \rceil + t}$ be pairwise different numbers from $\{1, \ldots, m\}$. We now prove that the fraction of information systems, for which $\{f_{i_1}, \ldots, f_{i_{\lceil 2\log_2 n \rceil + t}}\}$ is a test, is at least $1 - 1/2^{t+1}$.

Theorem 1.53. *Let us consider binary information systems with n rows and m columns labeled with attributes $f_1, \ldots, f_m$. Let $m \geq \lceil 2\log_2 n \rceil + t$, where t is a natural number, and $i_1, \ldots, i_{\lceil 2\log_2 n \rceil + t}$ be different numbers from $\{1, \ldots, m\}$. Then the fraction of information systems, for which $\{f_{i_1}, \ldots, f_{i_{\lceil 2\log_2 n \rceil + t}}\}$ is a test, is at least $1 - 1/2^{t+1}$.*

Proof. Let $k = \lceil 2\log_2 n \rceil + t$, $j, l \in \{1, \ldots, n\}$ and $j \neq l$. The number of information systems, for which j-th and l-th rows are equal at the intersection with columns $f_{i_1}, \ldots, f_{i_k}$, is equal to 2^{mn-k}. The number of pairs $j, l \in \{1, \ldots, n\}$ such that $j \neq l$ is at most $n^2/2$. Therefore, the number of information systems, for which $\{f_{i_1}, \ldots, f_{i_k}\}$ is not a test, is at most $(n^2/2)2^{mn-k} = 2^{mn-k+2\log_2 n-1} \leq 2^{mn-t-1}$. Thus, the fraction of information systems, for which $\{f_{i_1}, \ldots, f_{i_k}\}$ is a test, is at least

$$\frac{2^{mn} - 2^{mn-t-1}}{2^{mn}} = 1 - \frac{1}{2^{t+1}} \; .$$

□

We now fix a set D of decision attributes. From the considered result it follows, for example, that for 99% of binary decision tables with n rows, $m \geq \lceil 2\log_2 n \rceil + 6$ conditional attributes and decision attribute from D for each row there exists an exact decision rule which length is equal to $\lceil 2\log_2 n \rceil + 6$.

It is possible to improve this bound if we consider decision rules not for all rows, but for one fixed row only.

Let $j \in \{1, \ldots, n\}$. A subset $\{f_{i_1}, \ldots, f_{i_k}\}$ of attributes will be called a *j-th local test* for the information system I if these attributes separate from the row r_j any row r_l, where $l \in \{1, \ldots, n\}$ and $l \neq j$.

Adding an arbitrary decision attribute to the considered information system I we obtain a decision table T. Let $r_j = (b_1, \ldots, b_m)$ and d be the decision attached to r_j. If $\{f_{i_1}, \dots, f_{i_k}\}$ is a j-th local test for the information system I, then $(f_{i_1} = b_{i_1}) \wedge \dots \wedge (f_{i_k} = b_{i_k}) \rightarrow d$ is a 0-decision rule for (T, r_j).

Let us fix a set D of decision attributes. If we prove the existence of good j-th local tests for the most part of binary information systems with n rows and m columns, then it means the existence of good decision rules for j-th row for the most part of binary decision tables with n rows, m conditional attributes and decision attributes from D.

Theorem 1.54. *Let us consider binary information systems with n rows and m columns labeled with attributes $f_1, \dots, f_m$. Let $m \geq \lceil \log_2 n \rceil + t$, where t is a natural number, $j \in \{1, \dots, n\}$ and $i_1, \dots, i_{\lceil \log_2 n \rceil + t}$ be pairwise different numbers from $\{1, \dots, m\}$. Then the fraction of information systems, for which $\{f_{i_1}, \dots, f_{i_{\lceil \log_2 n \rceil + t}}\}$ is a j-th local test, is at least $1 - 1/2^t$.*

Proof. Let $k = \lceil \log_2 n \rceil + t$, $l \in \{1, \dots, n\}$ and $l \neq j$. The number of information systems, for which j-th and l-th rows are equal at the intersection with columns $i_1, \dots, i_k$, is 2^{mn-k}. Therefore, the number of information systems, for which $\{f_{i_1}, \dots, f_{i_k}\}$ is not a j-th local test, is at most $n2^{mn-k} = 2^{mn-k+\log_2 n} \leq 2^{mn-t}$. Thus, the fraction of information systems, for which $\{f_{i_1}, \dots, f_{i_k}\}$ is a j-th local test, is at least $(2^{mn} - 2^{mn-t})/2^{mn} = 1 - 1/2^t$. □

Let us fix a set D of decision attributes and a number $j \in \{1, \dots, n\}$. From obtained result it follows that for 99% of binary decision tables with n rows, $m \geq \lceil \log_2 n \rceil + 7$ conditional attributes and the decision attribute from D for j-th row there exists an exact decision rule which length is equal to $\lceil \log_2 n \rceil + 7$.

Partial Decision Rules Constructed by Greedy Algorithm for the Most Part of Binary Decision Tables

Now we study the behavior of greedy algorithm for the most part of binary decision tables, under some assumption on relationships between the number of rows and the number of columns in tables.

Let I be a binary information system with n rows and m columns labeled with attributes $f_1, \dots, f_m$. For $j = 1, \dots, n$, we denote by r_j the j-th row of I. The information system I will be called *strongly saturated* if, for any row $r_j = (b_1, \dots, b_m)$ of I, for any $k \in \{1, \dots, n-1\}$ and for any k rows with numbers different from j, there exists a column f_i which has at least $k/2$ numbers $\neg b_i$ at the intersection with considered k rows.

First, we evaluate the number of strongly saturated binary information systems. After that, we study the work of greedy algorithm on a decision table obtained from a strongly saturated binary information system by adding a decision attribute. It is clear that the 0.5-hypothesis for decision rules holds for every such table.

Theorem 1.55. *Let us consider binary information systems with n rows and $m \geq n + \log_2 n$ columns labeled with attributes $f_1, \ldots, f_m$. Then the fraction of strongly saturated information systems is at least* $1 - 1/2^{m-n-\log_2 n+1}$.

Proof. Let us fix a number $j \in \{1, \ldots, n\}$, a tuple $\bar{b} = (b_1, \ldots, b_m) \in \{0, 1\}^m$, a number $k \in \{1, \ldots, n-1\}$ and k rows with numbers different from j. We now evaluate the number of information systems in which $r_j = \bar{b}$ and, for $i = 1, \ldots, m$, the column f_i has less than $k/2$ numbers $\neg b_i$ at the intersection with considered k rows. Such information systems will be called $(j, \bar{b})$-unsaturated in the considered k rows.

From Lemma 1.32 it follows that the number of tuples from $\{0, 1\}^k$, which have less than $k/2$ numbers $\neg b_i$, is at most 2^{k-1}. Therefore, the number of information systems, which are $(j, \bar{b})$-unsaturated in the considered k rows, is at most 2^{mn-2m}.

There are n variants for the choice of j, at most 2^{n-1} variants for the choice of $k \in \{1, \ldots, n-1\}$ and k rows with numbers different from j, and 2^m variants for the choice of tuple $\bar{b}$. Therefore, the number of information systems, which are not strongly saturated, is at most $n2^{n-1}2^m2^{mn-2m} = 2^{mn-2m+\log_2 n+n-1+m} = 2^{mn+\log_2 n+n-m-1}$, and the fraction of strongly saturated information systems is at least

$$\frac{2^{mn} - 2^{mn+\log_2 n+n-m-1}}{2^{mn}} = 1 - \frac{1}{2^{m-n-\log_2 n+1}} \, .$$

□

For example, if $m \geq n + \log_2 n + 6$, then at least 99% of binary information systems are strongly saturated.

Let us consider the work of greedy algorithm on an arbitrary decision table T obtained from a strongly saturated binary information system. Let r be an arbitrary row of table T. For $i = 1, 2, \ldots$, after the step number i at most $|U(T, r)|/2^i$ rows from $U(T, R)$ are unseparated from r. It is not difficult to show that $L_{\text{greedy}}(\alpha) \leq \lceil \log_2(1/\alpha) \rceil$ for any real α, $0 < \alpha < 1$. One can prove that $L_{\text{greedy}}(0) \leq \log_2 |U(T, r)| + 1$. It is easy to check that $l_{DR}(0) \leq 2$.

Results of Experiments

The first group of experiments is connected with the consideration of binary decision tables T containing $n \in \{10, 50, 100, 1000, 3000, 5000\}$ rows, $m \in \{10, 40, 100\}$ conditional attributes and one decision attribute with values from the set $\{1, \ldots, c\}$,

$c \in \{2, 10, 100\}$. For each triple of values (n, m, c), we generated randomly a decision table such that each element of this table is equal to b, $b \in \{0, 1\}$, with probability $1/2$, and each decision is equal to d, $d \in \{1, \ldots, c\}$, with probability $1/c$. For this table, we choose randomly 10 rows r. The results of experiments are represented in Tables 1.4–1.7.

In Table 1.4 the average percentage of rows from $U(T, r)$ separated from r at i-th step of greedy algorithm, $i = 1, \ldots, 10$, is presented for the case, where $m = 40$ and $c = 10$. For example, 53.10 means that, on the average, 53.10% of rows remaining unseparated before i-th step are separated at i-th step.

Table 1.4. Average percentage of rows separated at i-th step of greedy algorithm ($m = 40$ and $c = 10$)

Number	Number of step i									
of rows n	1	2	3	4	5	6	7	8	9	10
10	85.79	100.00								
50	65.99	74.71	94.67	100.00						
100	61.90	67.42	79.38	100.00						
1000	54.05	55.05	56.54	56.56	64.01	76.50	100.00			
3000	52.04	52.50	53.77	55.52	57.06	61.51	71.01	82.94	100.00	
5000	51.57	52.09	53.10	54.31	56.28	59.01	64.85	74.46	92.07	100.00

In Table 1.5 for each $\alpha \in \{0.1, 0.01, 0.001, 0.0\}$ the average length of α-decision rules constructed by the greedy algorithm is presented for decision tables with 10 conditional attributes.

Table 1.5. Average length of α-decision rules for decision tables with 10 conditional attributes

	α											
	0.1			0.01			0.001			0.0		
Number	Number of different decisions c											
of rows n	2	10	100	2	10	100	2	10	100	2	10	100
10	1.4	2.0	2.2	1.4	2.0	2.2	1.4	2.0	2.2	1.4	2.0	2.2
50	2.5	2.8	3.0	3.3	4.2	4.1	3.3	4.2	4.1	3.3	4.2	4.1
100	2.8	3.0	3.0	4.4	5.1	5.0	4.4	5.1	5.0	4.4	5.1	5.0
1000	3.2	3.5	3.9	5.8	6.1	6.2	7.8	8.4	8.7	7.8	8.4	8.7
3000	3.9	4.0	4.0	6.2	6.4	6.5	8.2	8.6	8.7	8.8	9.3	9.5
5000	4.0	4.0	4.0	6.4	6.8	6.8	8.6	8.9	9.1	9.0	9.9	9.9

In Table 1.6 for each $\alpha \in \{0.1, 0.01, 0.001, 0.0\}$ the average length of α-decision rules constructed by the greedy algorithm is presented for decision tables with 40 conditional attributes.

Table 1.6. Average length of α-decision rules for decision tables with 40 conditional attributes

	α											
	0.1			0.01			0.001			0.0		
Number	Number of different decisions c											
of rows n	2	10	100	2	10	100	2	10	100	2	10	100
10	1.3	2.0	2.0	1.3	2.0	2.0	1.3	2.0	2.0	1.3	2.0	2.0
50	2.0	2.1	2.5	2.6	3.0	3.3	2.6	3.0	3.3	2.6	3.0	3.3
100	2.1	2.9	2.9	3.3	4.2	4.0	3.3	4.2	4.0	3.3	4.2	4.0
1000	3.0	3.0	3.1	5.0	5.8	5.8	6.1	7.0	7.0	6.1	7.0	7.0
3000	3.1	4.0	3.9	6.0	6.0	6.0	7.4	8.0	7.9	7.7	8.5	8.7
5000	3.9	4.0	4.0	6.0	6.2	6.1	8.0	8.1	8.7	8.5	9.1	9.3

In Table 1.7 for each $\alpha \in \{0.1, 0.01, 0.001, 0.0\}$ the average length of α-decision rules constructed by the greedy algorithm is presented for decision tables with 100 conditional attributes.

Table 1.7. Average length of α-decision rules for decision tables with 100 conditional attributes

	α											
	0.1			0.01			0.001			0.0		
Number	Number of different decisions c											
of rows n	2	10	100	2	10	100	2	10	100	2	10	100
10	1.1	2.0	2.0	1.1	2.3	2.0	1.1	2.3	2.0	1.1	2.3	2.0
50	2.0	2.0	2.1	2.5	3.0	3.0	2.5	3.0	3.0	2.5	3.0	3.0
100	2.0	2.5	2.9	3.0	3.9	4.0	3.0	3.9	4.0	3.0	3.9	4.0
1000	3.0	3.0	3.0	5.0	5.1	5.3	6.0	6.4	6.8	6.0	6.4	6.8
3000	3.0	3.5	3.7	6.0	6.0	6.0	7.0	7.8	7.8	7.1	8.2	7.9
5000	3.4	4.0	4.0	6.0	6.0	6.0	7.6	8.0	8.0	8.0	8.9	8.7

The obtained results show that for the most part of the considered decision rule problems (not only for the case, where $m \geq n + \log_2 n$) during each step the greedy algorithm chooses an attribute which separates at least one-half of unseparated rows.

It must be also noted that with increase of step number the percentage of rows, separated at this step, grows for the most part of the considered decision rule problems.

The second group of experiments is connected with the comparison of quality of greedy algorithm (Algorithm 2) and the following its *modification*: for a given decision table T, row r of T, and real α, $0 \leq \alpha < 1$, we construct an α-decision rule for T and r using the greedy algorithm, and after that, by removing from this α-decision rule some conditions, we obtain an irreducible α-decision rule for T. *Irreducible* means that the considered rule is an α-decision rule for T and r, but if we remove from the left-hand side of this rule an arbitrary condition, then we obtain a rule which is not an α-decision rule for T and r.

We generate randomly 10000 binary decision tables with binary decision attributes containing 40 rows and 10 conditional attributes. For each $\alpha \in \{0.00, 0.02, 0.04, \ldots, 0.30\}$, we find the number of tables for which the greedy algorithm for the first row constructs an α-decision rule with minimal length. This number is contained in the column of Table 1.8 labeled with "Opt".

We find the number of tables T for which the modification of greedy algorithm constructs for the first row an irreducible α-decision rule which length is less than the length of α-decision rule constructed by the greedy algorithm. This number is contained in the column of Table 1.8 labeled with "Impr".

Also we find the number of tables T for which for the first row the modification of greedy algorithm constructs an irreducible α-decision rule with minimal length which is less than the length of α-decision rule constructed by the greedy algorithm. This number is contained in the column of Table 1.8 labeled with "Opt+".

Table 1.8. Comparison of the greedy algorithm and its modification

α	Opt	Impr	Opt+
0.00	8456	387	373
0.02	8456	387	373
0.04	8530	353	342
0.06	9017	201	200
0.08	9089	187	186
0.10	9164	181	181
0.12	9323	156	156
0.14	9500	111	111
0.16	9731	68	68
0.18	9849	45	45
0.20	9954	10	10
0.22	9973	5	5
0.24	9994	0	0
0.26	9998	0	0
0.28	9998	0	0
0.30	10000	0	0

For small values of α, the improvement connected with the use of the modification of greedy algorithm is noticeable. We use this modification in Chap. 4 under the construction of classifiers based on partial decision rules.

1.3 Conclusions

The chapter is devoted (mainly) to the theoretical and experimental analysis of greedy algorithms for construction of partial covers and decision rules.

The obtained results show that, under some natural assumptions on the class NP, these algorithms are close to the best polynomial approximate algorithms for the minimization of partial covers and rules. Based on an information received during greedy algorithm work it is possible to obtain lower and upper bounds on the minimal complexity of partial covers and rules. Experimental and some theoretical results show that, for the most part of randomly generated set cover problems and binary decision tables, greedy algorithms construct simple partial covers and rules with relatively high accuracy. In particular, these results confirm the 0.5-hypothesis for covers and decision rules.

2

Partial Covers and Decision Rules with Weights

In this chapter, we study the case, where each subset, used for covering, has its own weight, and we should minimize the total weight of subsets in partial cover. The same situation is with decision rules: each conditional attribute has its own weight, and we should minimize the total weight of attributes in decision rule. If weight of each attribute characterizes time complexity of attribute value computation, then we try to minimize total time complexity of computation of attributes from partial decision rule. If weight characterizes a risk of attribute value computation (as in medical or technical diagnosis), then we try to minimize total risk, etc.

In rough set theory various problems can be represented as set cover problems with weights:

- problem of construction of a reduct [55] or partial reduct with minimal total weight of attributes for an information system;
- problem of construction of a decision reduct [55] or partial decision reduct with minimal total weight of attributes for a decision table;
- problem of construction of a decision rule or partial decision rule with minimal total weight of attributes for a row of a decision table (note that this problem is closely connected with the problem of construction of a local reduct [55] or partial local reduct with minimal total weight of attributes);
- problem of construction of a subsystem of a given system of decision rules which “covers" the same set of rows and has minimal total weight of rules (in the capacity of a rule weight we can consider its length).

So the study of covers and partial covers with weights is of some interest for rough set theory and related theories such as test theory and LAD. In this chapter, we list some known results on the set cover problem with weight which can be useful in applications, and obtain certain new results.

From results obtained in [61, 63] it follows that the problem of construction of partial cover with minimal weight is NP-hard. Therefore, we should consider polynomial approximate algorithms for minimization of weight of partial covers.

In [58] a greedy algorithm with weights for partial cover construction was investigated. This algorithm is a generalization of well known greedy algorithm with weights for exact cover construction [8].

Using results from Chap. 1 (based on results from [11, 53] and technique created in [61, 63]) on precision of polynomial approximate algorithms for construction of partial cover with minimal cardinality and results from [58] on precision of greedy algorithm with weights we show that, under some natural assumptions on the class NP, the greedy algorithm with weights is close to the best polynomial approximate algorithms for construction of partial cover with minimal weight. However, we can try to improve results of the work of greedy algorithm with weights for some part of set cover problems with weight.

We generalize greedy algorithm with weights [58], and consider greedy algorithm with two thresholds. The first threshold gives the exactness of constructed partial cover, and the second one is an interior parameter of the considered algorithm. We prove that for the most part of set cover problems there exists a weight function and values of thresholds such that the weight of partial cover constructed by the greedy algorithm with two thresholds is less than the weight of partial cover constructed by usual greedy algorithm with weights.

We describe two polynomial algorithms which always construct partial covers that are not worse than the one constructed by usual greedy algorithm with weights, and for the most part of set cover problems there exists a weight function and a value of the first threshold such that the weight of partial covers constructed by the considered two algorithms is less than the weight of partial cover constructed by usual greedy algorithm with weights.

Information on greedy algorithm work can be used for obtaining lower bounds on minimal cardinality of partial covers (see Chap. 1). We fix some kind of information about greedy algorithm work and find unimprovable lower bound on minimal weight of partial cover depending on this information. Obtained results show that this bound is not trivial and can be useful for investigation of set cover problems.

There exist bounds on precision of greedy algorithm without weights for partial cover construction which do not depend on the cardinality of covered set [7, 26, 27, 31]. We obtain similar bound for the case of weight.

The most part of results obtained for partial covers is generalized to the case of partial decision rules for decision tables which, in general case, are inconsistent (a decision table is inconsistent if it has equal rows with different decisions). In particular, we show that:

- Under some natural assumptions on the class NP, greedy algorithms with weights are close to the best polynomial approximate algorithms for minimization of total weight of attributes in partial decision rules.
- Based on an information receiving during greedy algorithm work it is possible to obtain nontrivial lower bounds on minimal total weight of attributes in partial decision rules.
- There exist polynomial time modifications of greedy algorithms which for a part of decision tables give better results than usual greedy algorithms.

This chapter is, in some sense, an extension of Chap. 1 to the case of weights which are not equal to 1. However, problems considered in this chapter (and proofs of results) are more complicated than the ones considered in Chap. 1. Bounds obtained in this chapter are sometimes weaker than the corresponding bounds from Chap. 1. We should note also that even if all weights are equal to 1, then results of the work of greedy algorithms considered in this chapter can be different from the results of the work of greedy algorithms considered in Chap. 1. For example, for the case of decision rules the number of chosen attributes is the same, but the last attributes can differ.

This chapter is based on papers [32, 33, 34, 35].

The chapter consists of three sections. In Sect. 2.1, partial covers are studied. In Sect. 2.2, partial decision rules are considered. Section 2.3 contains short conclusions.

2.1 Partial Covers with Weights

This section consists of eight subsections. In Sect. 2.1.1, main notions are considered. In Sect. 2.1.2, some known results are listed. In Sect. 2.1.3, polynomial approximate algorithms for minimization of partial cover weight are studied. In Sect. 2.1.4, a comparison of usual greedy algorithm and greedy algorithm with two thresholds is given. Two modifications of greedy algorithm are considered in Sect. 2.1.5. Section 2.1.6 is devoted to the consideration of a lower bound on the minimal weight of partial cover depending on some information about the work of greedy algorithm with two thresholds. In Sect. 2.1.7, two bounds on precision of greedy algorithm with two thresholds are considered that do not depend on the cardinality of covered set. In Sect. 2.1.8, some experimental results are discussed.

2.1.1 Main Notions

We repeat here some definitions from Chap. 1 and consider generalizations of other definitions to the case of arbitrary natural weights.

Let $A = \{a_1, \ldots, a_n\}$ be a nonempty finite set. Elements of A are enumerated by numbers $1, \ldots, n$ (in fact, we fix a linear order on A). Let $S = \{B_i\}_{i \in \{1,\ldots,m\}} = \{B_1, \ldots, B_m\}$ be a family of subsets of A such that $B_1 \cup \ldots \cup B_m = A$. We will assume that S can contain equal subsets of A. The pair (A, S) will be called a *set cover problem*. Let w be a *weight function* which corresponds to each $B_i \in S$ a natural number $w(B_i)$. The triple (A, S, w) will be called a *set cover problem with weights*. Note that, in fact, the weight function w is given on the set of indexes $\{1, \ldots, m\}$. But, for simplicity, we are writing $w(B_i)$ instead of $w(i)$.

Let I be a subset of $\{1, \ldots, m\}$. The family $P = \{B_i\}_{i \in I}$ will be called a *subfamily* of S. The number $|P| = |I|$ will be called the *cardinality* of P. Let $P = \{B_i\}_{i \in I}$ and $Q = \{B_i\}_{i \in J}$ be subfamilies of S. The notation $P \subseteq Q$ will mean that $I \subseteq J$. Let us denote $P \cup Q = \{B_i\}_{i \in I \cup J}$, $P \cap Q = \{B_i\}_{i \in I \cap J}$, and $P \setminus Q = \{B_i\}_{i \in I \setminus J}$.

A subfamily $Q = \{B_{i_1}, \ldots, B_{i_t}\}$ of the family S will be called a *partial cover for* (A, S). Let α be a real number such that $0 \leq \alpha < 1$. The subfamily Q will be called an *α-cover for* (A, S) if $|B_{i_1} \cup \ldots \cup B_{i_t}| \geq (1 - \alpha)|A|$. For example, 0.01-cover means that we should cover at least 99% of elements from A. Note that a 0-cover is usual (exact) cover. The number $w(Q) = \sum_{j=1}^{t} w(B_{i_j})$ will be called the *weight* of the partial cover Q. Let us denote by $C_{\min}(\alpha) = C_{\min}(\alpha, A, S, w)$ the minimal weight of α-cover for (A, S).

Let α and γ be real numbers such that $0 \leq \gamma \leq \alpha < 1$. A *greedy algorithm with two thresholds α and γ* is presented on the next page (see Algorithm 3).

Let us denote by $C^{\gamma}_{\text{greedy}}(\alpha) = C^{\gamma}_{\text{greedy}}(\alpha, A, S, w)$ the weight of α-cover constructed by the considered algorithm for the set cover problem with weights (A, S, w).

Note that the greedy algorithm with two thresholds α and $\gamma = \alpha$ (greedy algorithm with equal thresholds) coincides with the greedy algorithm with weights considered in [58].

2.1.2 Some Known Results

In this subsection, we assume that the weight function has values from the set of positive real numbers. For natural m, we denote $H(m) = 1 + \ldots + 1/m$. It is known that

$$\ln m \leq H(m) \leq \ln m + 1 \ .$$

Let us consider some results for the case of exact covers, where $\alpha = 0$. In this case $\gamma = 0$. The first results belong to Chvátal.

Algorithm 3: Greedy algorithm with two thresholds α and γ for partial cover construction

```
Input  : Set cover problem with weights $(A, S, w)$, where $S = \{B_1, \dots, B_m\}$, and real numbers $\alpha$
         and $\gamma$ such that $0 \le \gamma \le \alpha < 1$.
Output: $\alpha$-cover for $(A, S)$.
$Q \longleftarrow \emptyset$;
$D \longleftarrow \emptyset$;
$M \longleftarrow \lceil |A|(1-\alpha) \rceil$;
$N \longleftarrow \lceil |A|(1-\gamma) \rceil$;
while $|D| < M$ do
    select $B_i \in S$ with minimal index $i$ such that $B_i \setminus D \neq \emptyset$ and the value
        $$\frac{w(B_i)}{\min\{|B_i \setminus D|, N - |D|\}}$$
    is minimal;
    $Q \longleftarrow Q \cup \{B_i\}$;
    $D \longleftarrow D \cup B_i$;
end
return $Q$;
```

Theorem 2.1. (Chvátal [8]) *For any set cover problem with weights (A, S, w), the inequality $C^0_{\text{greedy}}(0) \le C_{\min}(0) H(|A|)$ holds.*

Theorem 2.2. (Chvátal [8]) *For any set cover problem with weights (A, S, w), the inequality $C^0_{\text{greedy}}(0) \le C_{\min}(0) H\left(\max_{B_i \in S} |B_i|\right)$ holds.*

Chvátal proved in [8] that the bounds from Theorems 2.1 and 2.2 are almost unimprovable.

We now consider some results for the case, where $\alpha \ge 0$ and $\gamma = \alpha$. The first upper bound on $C^\alpha_{\text{greedy}}(\alpha)$ was obtained by Kearns.

Theorem 2.3. (Kearns [18]) *For any set cover problem with weights (A, S, w) and any α, $0 \le \alpha < 1$, the inequality $C^\alpha_{\text{greedy}}(\alpha) \le C_{\min}(\alpha)(2H(|A|) + 3)$ holds.*

This bound was improved by Slavík.

Theorem 2.4. (Slavík [58]) *For any set cover problem with weights (A, S, w) and any α, $0 \le \alpha < 1$, the inequality $C^\alpha_{\text{greedy}}(\alpha) \le C_{\min}(\alpha) H\left(\lceil (1-\alpha)|A| \rceil\right)$ holds.*

Theorem 2.5. (Slavík [58])) *For any set cover problem with weights (A, S, w) and any α, $0 \le \alpha < 1$, the inequality $C^\alpha_{\text{greedy}}(\alpha) \le C_{\min}(\alpha) H\left(\max_{B_i \in S} |B_i|\right)$ holds.*

Slavík proved in [58] that the bounds from Theorems 2.4 and 2.5 are unimprovable.

2.1.3 Polynomial Approximate Algorithms

In this subsection, we consider three theorems which follow immediately from Theorems 1.13, 1.18 and 1.19.

Let $0 \leq \alpha < 1$. We consider the following problem: for a given set cover problem with weights (A, S, w) it is required to find an α-cover for (A, S) with minimal weight.

Theorem 2.6. *Let $0 \leq \alpha < 1$. Then the problem of construction of α-cover with minimal weight is NP-hard.*

From this theorem it follows that we should consider polynomial approximate algorithms for minimization of α-cover weight.

Theorem 2.7. *Let $\alpha \in \mathbb{R}$ and $0 \leq \alpha < 1$. If $NP \not\subseteq DTIME(n^{O(\log \log n)})$, then for any ε, $0 < \varepsilon < 1$, there is no polynomial algorithm that for a given set cover problem with weights (A, S, w) constructs an α-cover for (A, S) which weight is at most $(1 - \varepsilon)C_{\min}(\alpha, A, S, w) \ln |A|$.*

Theorem 2.8. *Let α be a real number such that $0 \leq \alpha < 1$. If $P \neq NP$, then there exists $\delta > 0$ such that there is no polynomial algorithm that for a given set cover problem with weights (A, S, w) constructs an α-cover for (A, S) which weight is at most $\delta C_{\min}(\alpha, A, S, w) \ln |A|$.*

From Theorem 2.4 it follows that $C^{\alpha}_{\text{greedy}}(\alpha) \leq C_{\min}(\alpha)(1 + \ln |A|)$. From this inequality and from Theorem 2.7 it follows that, under the assumption $NP \not\subseteq DTIME(n^{O(\log \log n)})$, the greedy algorithm with two thresholds α and $\gamma = \alpha$ (in fact, the greedy algorithm with weights from [58]) is close to the best polynomial approximate algorithms for minimization of partial cover weight. From the considered inequality and from Theorem 2.8 it follows that, under the assumption $P \neq NP$, the greedy algorithm with two thresholds α and $\gamma = \alpha$ is not far from the best polynomial approximate algorithms for minimization of partial cover weight.

However, we can try to improve the results of the work of greedy algorithm with two thresholds α and $\gamma = \alpha$ for some part of set cover problems with weights.

2.1.4 Comparison of Usual Greedy Algorithm and Greedy Algorithm with Two Thresholds

The following example shows that if for greedy algorithm with two thresholds α and γ we will use γ such that $\gamma < \alpha$, we can obtain sometimes better results than in the case $\gamma = \alpha$.

Example 2.9. Let us consider a set cover problem (A, S, w) such that $A = \{1, 2, 3, 4, 5, 6\}$, $S = \{B_1, B_2\}$, $B_1 = \{1\}$, $B_2 = \{2, 3, 4, 5, 6\}$, $w(B_1) = 1$ and $w(B_2) = 4$. Set $\alpha = 0.5$. It means that we should cover at least $M = \lceil (1-\alpha)|A| \rceil = 3$ elements from A. If $\gamma = \alpha = 0.5$, then the result of the work of greedy algorithm with thresholds α and γ is the 0.5-cover $\{B_1, B_2\}$ which weight is equal to 5. If $\gamma = 0 < \alpha$, then the result of the work of greedy algorithm with thresholds α and γ is the 0.5-cover $\{B_2\}$ which weight is equal to 4.

In this subsection, we show that, under some assumptions on $|A|$ and $|S|$, for the most part of set cover problems (A, S) there exists a weight function w and real numbers α, γ such that $0 \le \gamma < \alpha < 1$ and $C^{\gamma}_{\text{greedy}}(\alpha, A, S, w) < C^{\alpha}_{\text{greedy}}(\alpha, A, S, w)$. First, we consider criterion of existence of such w, α and γ (see Theorem 2.10). The first part of the proof of this criterion is based on a construction similar to considered in Example 2.9.

Let A be a finite nonempty set and $S = \{B_1, \dots, B_m\}$ be a family of subsets of A. We will say that the family S is 1-*uniform* if there exists a natural number k such that $|B_i| = k$ or $|B_i| = k + 1$ for any nonempty subset B_i from S. We will say that S is *strongly* 1-*uniform* if S is 1-uniform and for any subsets $B_{l_1}, \dots, B_{l_t}$ from S the family $\{B_1 \setminus U, \dots, B_m \setminus U\}$ is 1-uniform, where $U = B_{l_1} \cup \dots \cup B_{l_t}$.

Theorem 2.10. *Let (A, S) be a set cover problem. Then the following two statements are equivalent:*

1. *The family S is not strongly* 1*-uniform.*
2. *There exists a weight function w and real numbers α and γ such that $0 \le \gamma < \alpha < 1$ and $C^{\gamma}_{\text{greedy}}(\alpha, A, S, w) < C^{\alpha}_{\text{greedy}}(\alpha, A, S, w)$.*

Proof. Let $S = \{B_1, \dots, B_m\}$, and the family S be not strongly 1-uniform. Let us choose minimal number of subsets $B_{l_1}, \dots, B_{l_t}$ from the family S (it is possible that $t = 0$) such that the family $\{B_1 \setminus U, \dots, B_m \setminus U\}$ is not 1-uniform, where $U = B_{l_1} \cup \dots \cup B_{l_t}$ (if $t = 0$, then $U = \emptyset$). Since $\{B_1 \setminus U, \dots, B_m \setminus U\}$ is not 1-uniform, there exist two subsets B_i and B_j from S such that $|B_i \setminus U| > 0$ and $|B_j \setminus U| \ge |B_i \setminus U| + 2$. Let us choose real α and γ such that $M = \lceil |A|(1-\alpha) \rceil = |U| + |B_i \setminus U| + 1$ and $N = \lceil |A|(1-\gamma) \rceil = |U| + |B_i \setminus U| + 2$. It is clear that $0 \le \gamma < \alpha < 1$. Let us define a weight function w as follows: $w(B_{l_1}) = \ldots = w(B_{l_t}) = 1$, $w(B_i) = |A| \times 2|B_i \setminus U|$, $w(B_j) = |A|(2|B_i \setminus U| + 3)$ and $w(B_r) = |A|(3|B_i \setminus U| + 6)$ for any B_r from S such that $r \notin \{i, j, l_1, \dots, l_t\}$.

We now consider the work of greedy algorithm with two thresholds α and $\gamma = \alpha$. One can show that during the first t steps the greedy algorithm will choose subsets $B_{l_1}, \dots, B_{l_t}$ (may be in an another order). It is clear that $|U| < M$. Therefore, the greedy algorithm should make the step number $t + 1$. During this step the greedy algorithm will choose a subset B_k from S with minimal number k for which $B_k \setminus U \neq \emptyset$ and the value

$$p(k) = \frac{w(B_k)}{\min\{|B_k \setminus U|, M - |U|\}} = \frac{w(B_k)}{\min\{|B_k \setminus U|, |B_i \setminus U| + 1\}}$$

is minimal.

It is clear that $p(i) = 2|A|$, $p(j) = (2 + 1/(|B_i \setminus U| + 1))|A|$ and $p(k) > 3|A|$ for any subset B_k from S such that $B_k \setminus U \neq \emptyset$ and $k \notin \{i, j, l_1, \ldots, l_t\}$. Therefore, during the step number $t + 1$ the greedy algorithm will choose the subset B_i. Since $|U| + |B_i \setminus U| = M - 1$, the greedy algorithm will make the step number $t+2$ and will choose a subset from S which is different from $B_{l_1}, \ldots, B_{l_t}, B_i$. As a result we obtain $C^{\alpha}_{\text{greedy}}(\alpha, A, S, w) \geq t + |A| \times 2|B_i \setminus U| + |A|(2|B_i \setminus U| + 3)$.

We now consider the work of greedy algorithm with two thresholds α and γ. One can show that during the first t steps the greedy algorithm will choose subsets $B_{l_1}, \ldots, B_{l_t}$ (may be in an another order). It is clear that $|U| < M$. Therefore, the greedy algorithm should make the step number $t + 1$. During this step the greedy algorithm will choose a subset B_k from S with minimal number k for which $B_k \setminus U \neq \emptyset$ and the value

$$q(k) = \frac{w(B_k)}{\min\{|B_k \setminus U|, N - |U|\}} = \frac{w(B_k)}{\min\{|B_k \setminus U|, |B_i \setminus U| + 2\}}$$

is minimal.

It is clear that $q(i) = 2|A|$, $q(j) = (2 - 1/(|B_i \setminus U| + 2))|A|$ and $q(k) \geq 3|A|$ for any subset B_k from S such that $B_k \setminus U \neq \emptyset$ and $k \notin \{i, j, l_1, \ldots, l_t\}$. Therefore, during the step number $t + 1$ the greedy algorithm will choose the subset B_j. Since $|U| + |B_j \setminus U| > M$, the α-cover constructed by greedy algorithm will be equal to $\{B_{l_1}, \ldots, B_{l_t}, B_j\}$. As a result we obtain $C^{\gamma}_{\text{greedy}}(\alpha, A, S, w) = t + |A|(2|B_i \setminus U| + 3)$. Since $C^{\alpha}_{\text{greedy}}(\alpha, A, S, w) \geq t + |A| \times 2|B_i \setminus U| + |A|(2|B_i \setminus U| + 3)$ and $|B_i \setminus U| > 0$, we conclude that $C^{\alpha}_{\text{greedy}}(\alpha, A, S, w) > C^{\gamma}_{\text{greedy}}(\alpha, A, S, w)$.

Let the family S be strongly 1-uniform. We consider arbitrary weight function w for S and real numbers α and γ such that $0 \leq \gamma < \alpha < 1$. Let us show that $C^{\gamma}_{\text{greedy}}(\alpha, A, S, w) \geq C^{\alpha}_{\text{greedy}}(\alpha, A, S, w)$. Let us denote $M = \lceil |A|(1 - \alpha) \rceil$ and $N = \lceil |A|(1 - \gamma) \rceil$. If $M = N$, then $C^{\gamma}_{\text{greedy}}(\alpha, A, S, w) = C^{\alpha}_{\text{greedy}}(\alpha, A, S, w)$. Let $N > M$.

We now apply the greedy algorithm with thresholds α and $\gamma = \alpha$ to the set cover problem with weights (A, S, w). Let during the construction of α-cover this algorithm choose sequentially subsets $B_{g_1}, \ldots, B_{g_t}$. Let us apply now the greedy algorithm with thresholds α and γ to the set cover problem with weights (A, S, w). If during the construction of α-cover this algorithm chooses sequentially subsets $B_{g_1}, \ldots, B_{g_t}$, then

$C^{\gamma}_{\rm greedy}(\alpha, A, S, w) = C^{\alpha}_{\rm greedy}(\alpha, A, S, w)$. Let there exist a nonnegative integer r, $0 \leq r \leq t-1$, such that during the first r steps the considered algorithm chooses subsets $B_{g_1}, \dots, B_{g_r}$, but at the step number $r+1$ the algorithm chooses a subset B_k such that $k \neq g_{r+1}$. Let us denote $B_{g_0} = \emptyset$, $D = B_{g_0} \cup \dots \cup B_{g_r}$ and $J = \{i : i \in \{1, \dots, m\}, B_i \setminus D \neq \emptyset\}$. It is clear that $g_{r+1}, k \in J$. For any $i \in J$, we denote

$$p(i) = \frac{w(B_i)}{\min\{|B_i \setminus D|, M - |D|\}} \ , \quad q(i) = \frac{w(B_i)}{\min\{|B_i \setminus D|, N - |D|\}} \ .$$

Since $k \neq g_{r+1}$, we conclude that there exists $i \in J$ such that $p(i) \neq q(i)$. Therefore, $|B_i \setminus D| > M - |D|$. Since S is strongly 1-uniform family, we have $|B_j \setminus D| \geq M - |D|$ for any $j \in J$. From here it follows, in particular, that $r+1 = t$, and $\{B_{g_1}, \dots, B_{g_{t-1}}, B_k\}$ is an α-cover for (A, S).

It is clear that $p(g_t) \leq p(k)$. Since $|B_k \setminus D| \geq M - |D|$ and $|B_{g_t} \setminus D| \geq M - |D|$, we have $p(k) = w(B_k)/(M - |D|)$, $p(g_t) = w(B_{g_t})/(M - |D|)$. Therefore, $w(B_{g_t}) \leq w(B_k)$.

Taking into account that $C^{\gamma}_{\rm greedy}(\alpha, A, S, w) = w(B_{g_1}) + \dots + w(B_{g_{t-1}}) + w(B_k)$ and $C^{\alpha}_{\rm greedy}(\alpha, A, S, w) = w(B_{g_1}) + \dots + w(B_{g_{t-1}}) + w(B_{g_t})$ we obtain $C^{\gamma}_{\rm greedy}(\alpha, A, S, w) \geq C^{\alpha}_{\rm greedy}(\alpha, A, S, w)$. □

Let us show that, under some assumptions on $|A|$ and $|S|$, the most part of set cover problems (A, S) is not 1-uniform and, therefore, is not strongly 1-uniform.

There is a one-to-one correspondence between set cover problems and tables filled by numbers from $\{0, 1\}$ and having no rows filled by 0 only. Let $A = \{a_1, \dots, a_n\}$ and $S = \{B_1, \dots, B_m\}$. Then the problem (A, S) corresponds to the table with n rows and m columns which for $i = 1, \dots, n$ and $j = 1, \dots, m$ has 1 at the intersection of i-th row and j-th column if and only if $a_i \in B_j$. Remind that a table filled by numbers from $\{0, 1\}$ is called *SC-table* if this table has no rows filled by 0 only.

Lemma 2.11. *Let $n \in \mathbb{N}$, $n \geq 4$ and $k \in \{0, \dots, n\}$. Then $C^k_n \leq C^{\lfloor n/2 \rfloor}_n < 2^n/\sqrt{n}$.*

Proof. It is well known (see, for example, [75], p. 178) that $C^k_n \leq C^{\lfloor n/2 \rfloor}_n$. Let n be even and $n \geq 4$. It is known (see [12], p. 278) that

$$C^{\lfloor n/2 \rfloor}_n \leq \frac{2^n}{\sqrt{\frac{3n}{2} + 1}} < \frac{2^n}{\sqrt{n}} \ .$$

Let n be odd and $n \geq 5$. Using well known equality $C^{\lfloor n/2 \rfloor}_n = C^{\lfloor n/2 \rfloor}_{n-1} + C^{\lfloor n/2 \rfloor - 1}_{n-1}$ and the fact, that $C^{\lfloor (n-1)/2 \rfloor}_{n-1} \geq C^k_{n-1}$ for any $k \in \{0, \dots, n-1\}$, we obtain $C^{\lfloor n/2 \rfloor}_n \leq 2C^{\lfloor (n-1)/2 \rfloor}_{n-1}$. Thus,

$$C^{\lfloor n/2 \rfloor}_n \leq \frac{2^n}{\sqrt{\frac{3(n-1)}{2} + 1}} < \frac{2^n}{\sqrt{\frac{3(n-1)}{3} + 1}} = \frac{2^n}{\sqrt{n}} \ .$$

Therefore, for any $n \geq 4$ the inequality $C^{\lfloor n/2 \rfloor}_n < 2^n/\sqrt{n}$ holds. □

Theorem 2.12. *Let us consider set cover problems (A, S) such that $A = \{a_1, \ldots, a_n\}$ and $S = \{B_1, \ldots, B_m\}$. Let $n \geq 4$ and $m \geq \log_2 n + 1$. Then the fraction of set cover problems which are not 1-uniform is at least*

$$1 - \frac{9^{m/2+1}}{n^{m/2-1}} \ .$$

Proof. The considered fraction is at least $(q - p)/q$, where q is the number of SC-tables with n rows and m columns, and p is the number of tables with n rows and m columns filled by 0 and 1 for each of which there exists $k \in \{1, \ldots, n-1\}$ such that the number of units in each column belongs to the set $\{0, k, k+1\}$.

From Lemma 1.30 it follows that $q \geq 2^{mn} - 2^{mn-m+\log_2 n}$. It is clear that $p \leq \sum_{k=1}^{n-1}(C_n^k + C_n^{k+1} + 1)^m$. From Lemma 2.11 it follows that $C_n^{\lfloor n/2 \rfloor} \geq C_n^k$ for any $k \in \{1, \ldots, n\}$. Therefore, $p \leq (n-1)\left(3C_n^{\lfloor n/2 \rfloor}\right)^m$. Using Lemma 2.11 we conclude that $3C_n^{\lfloor n/2 \rfloor} 2^n / \sqrt{n/9}$ for any $n \geq 4$. Therefore,

$$p < \frac{(n-1)2^{mn}}{\left(\frac{n}{9}\right)^{m/2}} \quad \text{and} \quad \frac{q-p}{q} = 1 - \frac{p}{q} > 1 - \frac{(n-1)2^{mn}}{\left(\frac{n}{9}\right)^{m/2}\left(2^{mn} - 2^{mn-m+\log_2 n}\right)} \ .$$

Taking into account that $m \geq \log_2 n + 1$ we obtain

$$\frac{q-p}{q} > 1 - \frac{2(n-1)}{\left(\frac{n}{9}\right)^{m/2}} > 1 - \frac{9^{m/2+1}}{n^{m/2-1}} \ .$$

□

So if n is large enough and $m \geq \log_2 n + 1$, then the most part of set cover problems (A, S) with $|A| = n$ and $|S| = m$ is not 1-uniform.

For example, the fraction of set cover problems (A, S) with $|A| = 81$ and $|S| = 20$, which are not 1-uniform, is at least $1 - 1/9^7 = 1 - 1/4782969$.

2.1.5 Two Modifications of Greedy Algorithm

Results obtained in the previous subsection show that the greedy algorithm with two thresholds is of some interest. In this subsection, we consider two polynomial modifications of greedy algorithm which allow us to use advantages of greedy algorithm with two thresholds.

Let (A, S, w) be a set cover problem with weights and α be a real number such that $0 \leq \alpha < 1$.

1. Of course, it is impossible to consider effectively all γ such that $0 \leq \gamma \leq \alpha$. Instead of this, we can consider all natural N such that $M \leq N \leq |A|$, where $M =$

$\lceil |A|(1-\alpha)\rceil$ (see Algorithm 3). For each $N \in \{M, \ldots, |A|\}$, we apply Algorithm 3 with parameters M and N to the set cover problem with weights (A, S, w) and after that choose an α-cover with minimal weight among constructed α-covers.

2. There exists also an another way to construct an α-cover which is not worse than the one obtained under consideration of all N such that $M \leq N \leq |A|$. Let us apply greedy algorithm with thresholds α and $\gamma = \alpha$ (see Algorithm 3) to the set cover problem with weights (A, S, w). Let the algorithm choose sequentially subsets $B_{g_1}, \ldots, B_{g_t}$. For each $i \in \{0, \ldots, t-1\}$, we find (if it is possible) a subset B_{l_i} from S with minimal weight $w(B_{l_i})$ such that $|B_{g_1} \cup \ldots \cup B_{g_i} \cup B_{l_i}| \geq M$, and form an α-cover $\{B_{g_1}, \ldots, B_{g_i}, B_{l_i}\}$ (if $i = 0$, then it will be the family $\{B_{l_0}\}$). After that, among constructed α-covers $\{B_{g_1}, \ldots, B_{g_t}\}$, ..., $\{B_{g_1}, \ldots, B_{g_i}, B_{l_i}\}$, ... we choose an α-cover with minimal weight. From Proposition 2.13 it follows that the constructed α-cover is not worse than the one constructed under consideration of all γ, $0 \leq \gamma \leq \alpha$, or (which is the same) all N, $M \leq N \leq |A|$.

Proposition 2.13. *Let (A, S, w) be a set cover problem with weights and α, γ be real numbers such that $0 \leq \gamma < \alpha < 1$. Let the greedy algorithm with two thresholds α and α, which is applied to (A, S, w), choose sequentially subsets $B_{g_1}, \ldots, B_{g_t}$. Let the greedy algorithm with two thresholds α and γ, which is applied to (A, S, w), choose sequentially subsets $B_{l_1}, \ldots, B_{l_k}$. Then either $k = t$ and $(l_1, \ldots, l_k) = (g_1, \ldots, g_t)$ or $k \leq t$, $(l_1, \ldots, l_{k-1}) = (g_1, \ldots, g_{k-1})$ and $l_k \neq g_k$.*

Proof. Let $S = \{B_1, \ldots, B_m\}$. Let us denote $M = \lceil |A|(1-\alpha)\rceil$ and $N = \lceil |A|(1-\gamma)\rceil$.

Let $(l_1, \ldots, l_k) \neq (g_1, \ldots, g_t)$. Since $\{B_{g_1}, \ldots, B_{g_{t-1}}\}$ is not an α-cover for (A, S), it is impossible that $k < t$ and $(l_1, \ldots, l_k) = (g_1, \ldots, g_k)$. Since $\{B_{g_1}, \ldots, B_{g_t}\}$ is an α-cover for (A, S), it is impossible that $k > t$ and $(l_1, \ldots, l_t) = (g_1, \ldots, g_t)$. Therefore, there exists $i \in \{0, \ldots, t-1\}$ such that during the first i steps algorithm with thresholds α and α and algorithm with thresholds α and γ choose the same subsets from S, but during the step number $i+1$ the algorithm with thresholds α and γ chooses a subset $B_{l_{i+1}}$ such that $l_{i+1} \neq g_{i+1}$.

Let us denote $B_{g_0} = \emptyset$, $D = B_{g_0} \cup \ldots \cup B_{g_i}$ and $J = \{j : j \in \{1, \ldots, m\}, B_j \backslash D \neq \emptyset\}$. It is clear that $g_{i+1}, l_{i+1} \in J$. For any $j \in J$, let

$$p(j) = \frac{w(B_j)}{\min\{|B_j \setminus D|, M - |D|\}} \quad \text{and} \quad q(j) = \frac{w(B_j)}{\min\{|B_j \setminus D|, N - |D|\}} \, .$$

Since $N \geq M$, we have $p(j) \geq q(j)$ for any $j \in J$. We now consider two cases.

Let $g_{i+1} < l_{i+1}$. In this case we have $p(g_{i+1}) \leq p(l_{i+1})$ and $q(g_{i+1}) > q(l_{i+1})$. Using inequality $p(g_{i+1}) \geq q(g_{i+1})$ we obtain $p(g_{i+1}) > q(l_{i+1})$ and $p(l_{i+1}) > q(l_{i+1})$. From the last inequality it follows that $|B_{l_{i+1}} \setminus D| > M - |D|$.

Let $g_{i+1} > l_{i+1}$. In this case we have $p(g_{i+1}) < p(l_{i+1})$ and $q(g_{i+1}) \geq q(l_{i+1})$. Using inequality $p(g_{i+1}) \geq q(g_{i+1})$ we obtain $p(g_{i+1}) \geq q(l_{i+1})$ and $p(l_{i+1}) > q(l_{i+1})$. From the last inequality it follows that $|B_{l_{i+1}} \setminus D| > M - |D|$.

So in any case we have $|B_{l_{i+1}} \setminus D| > M - |D|$. From this inequality it follows that after the step number $i+1$ the algorithm with thresholds α and γ should finish the work. Thus, $k = i+1$, $k \leq t$, $(l_1, \ldots, l_{k-1}) = (g_1, \ldots, g_{k-1})$ and $l_k \neq g_k$. □

2.1.6 Lower Bound on $C_{\min}(\alpha)$

In this subsection, we fix some information about the work of greedy algorithm with two thresholds and find the best lower bound on the value $C_{\min}(\alpha)$ depending on this information.

Let (A, S, w) be a set cover problem with weights and α, γ be real numbers such that $0 \leq \gamma \leq \alpha < 1$. Let us apply the greedy algorithm with thresholds α and γ to the set cover problem with weights (A, S, w). Let during the construction of α-cover the greedy algorithm choose sequentially subsets $B_{g_1}, \ldots, B_{g_t}$.

Let us denote $B_{g_0} = \emptyset$ and $\delta_0 = 0$. For $i = 1, \ldots, t$, we denote $\delta_i = |B_{g_i} \setminus (B_{g_0} \cup \ldots \cup B_{g_{i-1}})|$ and $w_i = w(B_{g_i})$.

As information on the greedy algorithm work we will use numbers $M_C = M_C(\alpha, \gamma, A, S, w) = \lceil |A|(1-\alpha) \rceil$ and $N_C = N_C(\alpha, \gamma, A, S, w) = \lceil |A|(1-\gamma) \rceil$, and tuples $\Delta_C = \Delta_C(\alpha, \gamma, A, S, w) = (\delta_1, \ldots, \delta_t)$, $W_C = W_C(\alpha, \gamma, A, S, w) = (w_1, \ldots, w_t)$.

For $i = 0, \ldots, t-1$, we denote

$$\varrho_i = \left\lceil \frac{w_{i+1}(M_C - (\delta_0 + \ldots + \delta_i))}{\min\{\delta_{i+1}, N_C - (\delta_0 + \ldots + \delta_i)\}} \right\rceil .$$

Let us define parameter $\varrho_C(\alpha, \gamma) = \varrho_C(\alpha, \gamma, A, S, w)$ as follows:

$$\varrho_C(\alpha, \gamma) = \max\{\varrho_i : i = 0, \ldots, t-1\} .$$

We will prove that $\varrho_C(\alpha, \gamma)$ is the best lower bound on $C_{\min}(\alpha)$ depending on M_C, N_C, Δ_C and W_C. This lower bound is based on a generalization of the following simple reasoning: if we should cover M elements, and the maximal cardinality of a subset from S is δ, then we should use at least $\lceil M/\delta \rceil$ subsets.

Theorem 2.14. *For any set cover problem with weights (A, S, w) and any real numbers α, γ, $0 \leq \gamma \leq \alpha < 1$, the inequality $C_{\min}(\alpha, A, S, w) \geq \varrho_C(\alpha, \gamma, A, S, w)$ holds, and there exists a set cover problem with weights (A', S', w') such that*

$$M_C(\alpha, \gamma, A', S', w') = M_C(\alpha, \gamma, A, S, w) ,$$
$$N_C(\alpha, \gamma, A', S', w') = N_C(\alpha, \gamma, A, S, w) ,$$

$$\begin{aligned} \Delta_C(\alpha,\gamma,A',S',w') &= \Delta_C(\alpha,\gamma,A,S,w)\ , \\ W_C(\alpha,\gamma,A',S',w') &= W_C(\alpha,\gamma,A,S,w)\ , \\ \varrho_C(\alpha,\gamma,A',S',w') &= \varrho_C(\alpha,\gamma,A,S,w)\ , \\ C_{\min}(\alpha,A',S',w') &= \varrho_C(\alpha,\gamma,A',S',w')\ . \end{aligned}$$

Proof. Let (A,S,w) be a set cover problem with weights, $S=\{B_1,\dots,B_m\}$, and α,γ be real numbers such that $0\le\gamma\le\alpha<1$. Let us denote $M=M_C(\alpha,\gamma,A,S,w)=\lceil|A|(1-\alpha)\rceil$ and $N=N_C(\alpha,\gamma,A,S,w)=\lceil|A|(1-\gamma)\rceil$. Let $\{B_{l_1},\dots,B_{l_k}\}$ be an optimal α-cover for (A,S,w), i.e., $w(B_{l_1})+\ldots+w(B_{l_k})=C_{\min}(\alpha,A,S,w)=C_{\min}(\alpha)$ and $|B_{l_1}\cup\ldots\cup B_{l_k}|\ge M$.

We now apply the greedy algorithm with thresholds α and γ to (A,S,w). Let during the construction of α-cover the greedy algorithm choose sequentially subsets $B_{g_1},\dots,B_{g_t}$. Set $B_{g_0}=\emptyset$.

Let $i\in\{0,\dots,t-1\}$. Let us denote $D=B_{g_0}\cup\ldots\cup B_{g_i}$. It is clear that after i steps of greedy algorithm work in the set $B_{l_1}\cup\ldots\cup B_{l_k}$ at least $|B_{l_1}\cup\ldots\cup B_{l_k}|-|B_{g_0}\cup\ldots\cup B_{g_i}|\ge M-|D|>0$ elements remained uncovered. After i-th step, $p_1=|B_{l_1}\setminus D|$ elements remained uncovered in the set B_{l_1}, ..., and $p_k=|B_{l_k}\setminus D|$ elements remained uncovered in the set B_{l_k}. We know that $p_1+\ldots+p_k\ge M-|D|>0$. Let, for simplicity, $p_1>0,\dots,p_r>0,p_{r+1}=\ldots=p_k=0$. For $j=1,\dots,r$, we denote $q_j=\min\{p_j,N-|D|\}$. It is clear that $N-|D|\ge M-|D|$. Therefore, $q_1+\ldots+q_r\ge M-|D|$. Let us consider numbers $w(B_{l_1})/q_1,\dots,w(B_{l_r})/q_r$. Let us show that at least one of these numbers is at most $\beta=(w(B_{l_1})+\ldots+w(B_{l_r}))/(q_1+\ldots+q_r)$. We assume the contrary. Then $w(B_{l_1})+\ldots+w(B_{l_r})=w(B_{l_1})q_1/q_1+\ldots+w(B_{l_r})q_r/q_r>(q_1+\ldots+q_r)\beta=w(B_{l_1})+\ldots+w(B_{l_r})$, which is impossible.

We know that $q_1+\ldots+q_r\ge M-|D|$ and $w(B_{l_1})+\ldots+w(B_{l_r})\le C_{\min}(\alpha)$. Therefore, $\beta\le C_{\min}(\alpha)/(M-|D|)$, and there exists $j\in\{1,\dots,k\}$ such that $B_{l_j}\setminus D\ne\emptyset$ and $w(B_{l_j})/\min\{|B_{l_j}\setminus D|,N-|D|\}\le\beta$. Hence,

$$\frac{w(B_{g_{i+1}})}{\min\{|B_{g_{i+1}}\setminus D|,N-|D|\}}\le\beta\le\frac{C_{\min}(\alpha)}{M-|D|}$$

and $C_{\min}(\alpha)\ge w(B_{g_{i+1}})(M-|D|)/\min\{|B_{g_{i+1}}\setminus D|,N-|D|\}$.

Taking into account that $C_{\min}(\alpha)$ is a natural number we obtain $C_{\min}(\alpha)\ge\left\lceil w(B_{g_{i+1}})(M-|D|)/\min\{|B_{g_{i+1}}\setminus D|,N-|D|\}\right\rceil=\varrho_i$. Since the last inequality holds for any $i\in\{0,\dots,t-1\}$ and $\varrho_C(\alpha,\gamma)=\varrho_C(\alpha,\gamma,A,S,w)=\max\{\varrho_i:i=0,\dots,t-1\}$, we conclude that $C_{\min}(\alpha)\ge\varrho_C(\alpha,\gamma)$.

Let us show that this bound is unimprovable depending on M_C, N_C, Δ_C and W_C. Let us consider a set cover problem with weights (A',S',w'), where $A'=A$, $S'=\{B_1,\dots,B_m,B_{m+1}\}$, $|B_{m+1}|=M$, $B_{g_1}\cup\ldots\cup B_{g_{t-1}}\subsetneq B_{m+1}\subseteq B_{g_1}\cup$

$\ldots \cup B_{g_t}$, $w'(B_1) = w(B_1), \ldots, w'(B_m) = w(B_m)$ and $w'(B_{m+1}) = \varrho_C(\alpha, \gamma)$. It is clear that $M_C(\alpha, \gamma, A', S', w') = M_C(\alpha, \gamma, A, S, w) = M$ and $N_C(\alpha, \gamma, A', S', w') = N_C(\alpha, \gamma, A, S, w) = N$. We show $\Delta_C(\alpha, \gamma, A', S', w') = \Delta_C(\alpha, \gamma, A, S, w)$ and $W_C(\alpha, \gamma, A', S', w') = W_C(\alpha, \gamma, A, S, w)$.

Let us show by induction on $i \in \{1, \ldots, t\}$ that for the set cover problem with weights (A', S', w') at the step number i the greedy algorithm with two thresholds α and γ will choose the subset B_{g_i}. Let us consider the first step. Set $D = \emptyset$. It is clear that $w'(B_{m+1})/\min\{|B_{m+1} \setminus D|, N - |D|\} = \varrho_C(\alpha, \gamma)/(M - |D|)$. From the definition of $\varrho_C(\alpha, \gamma)$ it follows that

$$\frac{w'(B_{g_1})}{\min\{|B_{g_1} \setminus D|, N - |D|\}} = \frac{w(B_{g_1})}{\min\{|B_{g_1} \setminus D|, N - |D|\}} \le \frac{\varrho_C(\alpha, \gamma)}{M - |D|} .$$

Using this fact and the inequality $g_1 < m + 1$ it is not difficult to prove that at the first step the greedy algorithm will choose the subset B_{g_1}.

Let $i \in \{1, \ldots, t-1\}$. Let us assume that the greedy algorithm made i steps for (A', S', w') and chose subsets $B_{g_1}, \ldots, B_{g_i}$. Let us show that at the step $i+1$ the subset $B_{g_{i+1}}$ will be chosen. Let us denote $D = B_{g_1} \cup \ldots \cup B_{g_i}$. Since $B_{g_1} \cup \ldots \cup B_{g_i} \subseteq B_{m+1}$ and $|B_{m+1}| = M$, we have $|B_{m+1} \setminus D| = M - |D|$. Therefore, $w'(B_{m+1})/\min\{|B_{m+1} \setminus D|, N - |D|\} = \varrho_C(\alpha, \gamma)/(M - |D|)$. From the definition of the parameter $\varrho_C(\alpha, \gamma)$ it follows that

$$\frac{w'(B_{g_{i+1}})}{\min\{|B_{g_{i+1}} \setminus D|, N - |D|\}} = \frac{w(B_{g_{i+1}})}{\min\{|B_{g_{i+1}} \setminus D|, N - |D|\}} \le \frac{\varrho_C(\alpha, \gamma)}{M - |D|} .$$

Using this fact and the inequality $g_{i+1} < m + 1$ it is not difficult to prove that at the step number $i + 1$ the greedy algorithm will choose the subset $B_{g_{i+1}}$.

Thus, $\Delta_C(\alpha, \gamma, A', S', w') = \Delta_C(\alpha, \gamma, A, S, w)$ and $W_C(\alpha, \gamma, A', S', w') = W_C(\alpha, \gamma, A, S, w)$. Hence, $\varrho_C(\alpha, \gamma, A', S', w') = \varrho_C(\alpha, \gamma, A, S, w) = \varrho_C(\alpha, \gamma)$. From been proven it follows that $C_{\min}(\alpha, A', S', w') \ge \varrho_C(\alpha, \gamma, A', S', w')$. It is clear that $\{B_{m+1}\}$ is an α-cover for (A', S') and the weight of $\{B_{m+1}\}$ is equal to $\varrho_C(\alpha, \gamma, A', S', w')$. Hence, $C_{\min}(\alpha, A', S', w') = \varrho_C(\alpha, \gamma, A', S', w')$. □

Let us consider a property of the parameter $\varrho_C(\alpha, \gamma)$ which is important for practical use of the bound from Theorem 2.14.

Proposition 2.15. *Let (A, S, w) be a set cover problem with weights and α, γ be real numbers such that $0 \le \gamma \le \alpha < 1$. Then $\varrho_C(\alpha, \alpha, A, S, w) \ge \varrho_C(\alpha, \gamma, A, S, w)$.*

Proof. Let $S = \{B_1, \ldots, B_m\}$, $M = \lceil |A|(1 - \alpha) \rceil$, $N = \lceil |A|(1 - \gamma) \rceil$, and $\varrho_C(\alpha, \alpha) = \varrho_C(\alpha, \alpha, A, S, w)$, $\varrho_C(\alpha, \gamma) = \varrho_C(\alpha, \gamma, A, S, w)$.

Let us apply the greedy algorithm with thresholds α and α to (A,S,w). Let during the construction of α-cover this algorithm choose sequentially subsets $B_{g_1},\ldots,\ B_{g_t}$. Let us denote $B_{g_0}=\emptyset$. For $j=0,\ldots,t-1$, we denote $D_j=B_{g_0}\cup\ldots\cup B_{g_j}$ and

$$\varrho_C(\alpha,\alpha,j)=\left\lceil\frac{w(B_{g_{j+1}})(M-|D_j|)}{\min\{|B_{g_{j+1}}\setminus D_j|,M-|D_j|\}}\right\rceil .$$

Then $\varrho_C(\alpha,\alpha)=\max\{\varrho_C(\alpha,\alpha,j):j=0,\ldots,t-1\}$.

We now apply the greedy algorithm with thresholds α and γ to (A,S,w). Let during the construction of α-cover this algorithm choose sequentially subsets $B_{l_1},\ldots,$ B_{l_k}. From Proposition 2.13 it follows that either $k=t$ and $(l_1,\ldots,l_k)=(g_1,\ldots,g_t)$ or $k\le t$, $(l_1,\ldots,l_{k-1})=(g_1,\ldots,g_{k-1})$ and $l_k\neq g_k$. Let us consider these two cases separately. Let $k=t$ and $(l_1,\ldots,l_k)=(g_1,\ldots,g_t)$. For $j=0,\ldots,t-1$, we denote

$$\varrho_C(\alpha,\gamma,j)=\left\lceil\frac{w(B_{g_{j+1}})(M-|D_j|)}{\min\{|B_{g_{j+1}}\setminus D_j|,N-|D_j|\}}\right\rceil .$$

Then $\varrho_C(\alpha,\gamma)=\max\{\varrho_C(\alpha,\gamma,j):j=0,\ldots,t-1\}$. Since $N\ge M$, we have $\varrho_C(\alpha,\gamma,j)\le\varrho_C(\alpha,\alpha,j)$ for $j=0,\ldots,t-1$. Hence, $\varrho_C(\alpha,\gamma)\le\varrho_C(\alpha,\alpha)$. Let $k\le t$, $(l_1,\ldots,l_{k-1})=(g_1,\ldots,g_{k-1})$ and $l_k\neq g_k$. Let us denote

$$\varrho_C(\alpha,\gamma,k-1)=\left\lceil\frac{w(B_{l_k})(M-|D_{k-1}|)}{\min\{|B_{l_k}\setminus D_{k-1}|,N-|D_{k-1}|\}}\right\rceil$$

and, for $j=0,\ldots,k-2$,

$$\varrho_C(\alpha,\gamma,j)=\left\lceil\frac{w(B_{g_{j+1}})(M-|D_j|)}{\min\{|B_{g_{j+1}}\setminus D_j|,N-|D_j|\}}\right\rceil .$$

Then $\varrho_C(\alpha,\gamma)=\max\{\varrho_C(\alpha,\gamma,j):j=0,\ldots,k-1\}$. Since $N\ge M$, we have $\varrho_C(\alpha,\gamma,j)\le\varrho_C(\alpha,\alpha,j)$ for $j=0,\ldots,k-2$. It is clear that

$$\begin{aligned}\frac{w(B_{l_k})}{\min\{|B_{l_k}\setminus D_{k-1}|,N-|D_{k-1}|\}}&\le\frac{w(B_{g_k})}{\min\{|B_{g_k}\setminus D_{k-1}|,N-|D_{k-1}|\}}\\&\le\frac{w(B_{g_k})}{\min\{|B_{g_k}\setminus D_{k-1}|,M-|D_{k-1}|\}} .\end{aligned}$$

Thus, $\varrho_C(\alpha,\gamma,k-1)\le\varrho_C(\alpha,\alpha,k-1)$, $\varrho_C(\alpha,\gamma)\le\varrho_C(\alpha,\alpha)$. □

2.1.7 Upper Bounds on $C^{\gamma}_{\text{greedy}}(\alpha)$

In this subsection, we study some properties of parameter $\varrho_C(\alpha,\gamma)$ and obtain two upper bounds on the value $C^{\gamma}_{\text{greedy}}(\alpha)$ which do not depend directly on cardinality of the set A and cardinalities of subsets B_i from S.

Theorem 2.16. *Let (A, S, w) be a set cover problem with weights and α, γ be real numbers such that $0 \leq \gamma < \alpha < 1$. Then*

$$C^{\gamma}_{\text{greedy}}(\alpha, A, S, w) < \varrho_C(\gamma, \gamma, A, S, w)\left(\ln\left(\frac{1-\gamma}{\alpha-\gamma}\right)+1\right) .$$

Proof. Let $S = \{B_1, \ldots, B_m\}$. Let us denote $M = \lceil |A|(1-\alpha)\rceil$ and $N = \lceil |A|(1-\gamma)\rceil$.

We now apply the greedy algorithm with thresholds γ and γ to (A, S, w). Let during the construction of γ-cover the greedy algorithm choose sequentially subsets $B_{g_1}, \ldots, B_{g_t}$. Let us denote $B_{g_0} = \emptyset$, for $i = 0, \ldots, t-1$ denote $D_i = B_{g_0} \cup \ldots \cup B_{g_i}$, and denote $\varrho = \varrho_C(\gamma, \gamma, A, S, w)$. Immediately from the definition of the parameter ϱ it follows that for $i = 0, \ldots, t-1$,

$$\frac{w(B_{g_{i+1}})}{\min\{|B_{g_{i+1}} \setminus D_i|, N - |D_i|\}} \leq \frac{\varrho}{N - |D_i|} . \tag{2.1}$$

Note that $\min\{|B_{g_{i+1}} \setminus D_i|, N - |D_i|\} = |B_{g_{i+1}} \setminus D_i|$ for $i = 0, \ldots, t-2$, since $\{B_{g_0}, \ldots, B_{g_{i+1}}\}$ is not a γ-cover for (A, S). Therefore, for $i = 0, \ldots, t-2$ we have $w(B_{g_{i+1}})/|B_{g_{i+1}} \setminus D_i| \leq \varrho/(N - |D_i|)$ and $(N - |D_i|)/\varrho \leq |B_{g_{i+1}} \setminus D_i|/w(B_{g_{i+1}})$. Thus, for $i = 1, \ldots, t-1$, during the step number i the greedy algorithm covers at least $(N - |D_{i-1}|)/\varrho$ elements on each unit of weight. From (2.1) it follows that for $i = 0, \ldots, t-1$,

$$w(B_{g_{i+1}}) \leq \frac{\varrho \min\{|B_{g_{i+1}} \setminus D_i|, N - |D_i|\}}{N - |D_i|} \leq \varrho . \tag{2.2}$$

Let us assume that $\varrho = 1$. Using (2.2) we obtain $w(B_{g_1}) = 1$. From this equality and (2.1) it follows that $|B_{g_1}| \geq N$. Therefore, $\{B_{g_1}\}$ is an α-cover for (A, S), and $C^{\gamma}_{\text{greedy}}(\alpha) = 1$. It is clear that $\ln((1-\gamma)/(\alpha-\gamma)) + 1 > 1$. Therefore, the statement of the theorem holds if $\varrho = 1$.

We assume now that $\varrho \geq 2$. Let $|B_{g_1}| \geq M$. Then $\{B_{g_1}\}$ is an α-cover for (A, S). Using (2.2) we obtain $C^{\gamma}_{\text{greedy}}(\alpha) \leq \varrho$. Since $\ln((1-\gamma)/(\alpha-\gamma)) + 1 > 1$, we conclude that the statement of the theorem holds if $|B_{g_1}| \geq M$. Let $|B_{g_1}| < M$. Then there exists $q \in \{1, \ldots, t-1\}$ such that $|B_{g_1} \cup \ldots \cup B_{g_q}| < M$ and $|B_{g_1} \cup \ldots \cup B_{g_{q+1}}| \geq M$.

Taking into account that for $i = 1, \ldots, q$ during the step number i the greedy algorithm covers at least $(N - |D_{i-1}|)/\varrho$ elements on each unit of weight we obtain $N - |B_{g_1} \cup \ldots \cup B_{g_q}| \leq N(1 - 1/\varrho)^{w(B_{g_1}) + \ldots + w(B_{g_q})}$. Let us denote $k = w(B_{g_1}) + \ldots + w(B_{g_q})$. Then $N - N(1-1/\varrho)^k \leq |B_{g_1} \cup \ldots \cup B_{g_q}| \leq M - 1$. Therefore, $|A|(1-\gamma) - |A|(1-\gamma)(1-1/\varrho)^k < |A|(1-\alpha)$, $1 - \gamma - 1 + \alpha < (1-\gamma)((\varrho-1)/\varrho)^k$, $(\varrho/(\varrho-1))^k < (1-\gamma)/(\alpha-\gamma)$, $(1 + 1/(\varrho-1))^k < (1-\gamma)/(\alpha-\gamma)$, and $k/\varrho < \ln((1-\gamma)/(\alpha-\gamma))$. To obtain the last inequality we use known inequality $\ln(1 + 1/r) > 1/(r+1)$ which holds for any natural r. It is clear that $C^{\gamma}_{\text{greedy}}(\alpha) = k + w(B_{q+1})$. Using (2.2) we conclude that $w(B_{q+1}) \leq \varrho$. Therefore, $C^{\gamma}_{\text{greedy}}(\alpha) < \varrho \ln((1-\gamma)/(\alpha-\gamma)) + \varrho$. □

Corollary 2.17. *Let ε be a real number, and $0 < \varepsilon < 1$. Then for any α such that $\varepsilon \leq \alpha < 1$ the following inequalities hold:*

$$\varrho_C(\alpha, \alpha) \leq C_{\min}(\alpha) \leq C_{\text{greedy}}^{\alpha-\varepsilon}(\alpha) < \varrho_C(\alpha-\varepsilon, \alpha-\varepsilon)\left(\ln \frac{1}{\varepsilon} + 1\right) .$$

For example, if $\varepsilon = 0.01$ and $0.01 \leq \alpha < 1$, then $\varrho_C(\alpha, \alpha) \leq C_{\min}(\alpha) \leq C_{\text{greedy}}^{\alpha-0.01}(\alpha) < 5.61\varrho_C(\alpha - 0.01, \alpha - 0.01)$, and if $\varepsilon = 0.1$ and $0.1 \leq \alpha < 1$, then $\varrho_C(\alpha, \alpha) \leq C_{\min}(\alpha) \leq C_{\text{greedy}}^{\alpha-0.1}(\alpha) < 3.31\varrho_C(\alpha - 0.1, \alpha - 0.1)$.

The obtained results show that the lower bound $C_{\min}(\alpha) \geq \varrho_C(\alpha, \alpha)$ is nontrivial.

Theorem 2.18. *Let (A, S, w) be a set cover problem with weights and α, γ be real numbers such that $0 \leq \gamma < \alpha < 1$. Then*

$$C_{\text{greedy}}^{\gamma}(\alpha, A, S, w) < C_{\min}(\gamma, A, S, w)\left(\ln\left(\frac{1-\gamma}{\alpha-\gamma}\right) + 1\right) .$$

Proof. Using Theorem 2.16 we obtain $C_{\text{greedy}}^{\gamma}(\alpha, A, S, w) < \varrho_C(\gamma, \gamma, A, S, w) \times (\ln((1-\gamma)/(\alpha-\gamma)) + 1)$. The inequality $\varrho_C(\gamma, \gamma, A, S, w) \leq C_{\min}(\gamma, A, S, w)$ follows from Theorem 2.14. □

Corollary 2.19. $C_{\text{greedy}}^{0.3}(0.5) < 2.26C_{\min}(0.3)$*,* $C_{\text{greedy}}^{0.1}(0.2) < 3.20C_{\min}(0.1)$*,* $C_{\text{greedy}}^{0.001}(0.01) < 5.71C_{\min}(0.001)$*,* $C_{\text{greedy}}^{0}(0.001) < 7.91C_{\min}(0)$*.*

Corollary 2.20. *Let $0 < \alpha < 1$. Then $C_{\text{greedy}}^{0}(\alpha) < C_{\min}(0)\left(\ln(1/\alpha) + 1\right)$.*

Corollary 2.21. *Let ε be a real number, and $0 < \varepsilon < 1$. Then for any α such that $\varepsilon \leq \alpha < 1$ the inequalities $C_{\min}(\alpha) \leq C_{\text{greedy}}^{\alpha-\varepsilon}(\alpha) < C_{\min}(\alpha - \varepsilon)\left(\ln(1/\varepsilon) + 1\right)$ hold.*

2.1.8 Results of Experiments for α-Covers

All experiments can be divided into three groups.

The First Group of Experiments

The first group of experiments is connected with study of quality of greedy algorithm with equal thresholds (where $\gamma = \alpha$ or, which is the same, $N = M$), and comparison of quality of greedy algorithm with equal thresholds and the first modification of greedy algorithm (where for each $N \in \{M, \ldots, |A|\}$ we apply greedy algorithm with parameters M and N to set cover problem with weights, and after that choose an α-cover with minimal weight among constructed α-covers).

We generate randomly 1000 set cover problems with weights (A, S, w) such that $|A| = 40$, $|S| = 10$ and $1 \leq w(B_i) \leq 1000$ for each $B_i \in S$.

For each $\alpha \in \{0.0, 0.1, \ldots, 0.9\}$, we find the number of problems (A, S, w) for which greedy algorithm with equal thresholds constructs an α-cover with minimal weight (optimal α-cover), i.e., $C^{\alpha}_{\mathrm{greedy}}(\alpha, A, S, w) = C_{\min}(\alpha, A, S, w)$. This number is contained in the row of Table 2.1 labeled with "Opt".

We find the number of problems (A, S, w) for which the first modification of greedy algorithm constructs an α-cover which weight is less than the weight of α-cover constructed by greedy algorithm with equal thresholds, i.e., there exists γ such that $0 \leq \gamma < \alpha$ and $C^{\gamma}_{\mathrm{greedy}}(\alpha, A, S, w) < C^{\alpha}_{\mathrm{greedy}}(\alpha, A, S, w)$. This number is contained in the row of Table 2.1 labeled with "Impr".

Also we find the number of problems (A, S, w) for which the first modification of greedy algorithm constructs an optimal α-cover which weight is less than the weight of α-cover constructed by greedy algorithm with equal thresholds, i.e., there exists γ such that $0 \leq \gamma < \alpha$ and $C^{\gamma}_{\mathrm{greedy}}(\alpha, A, S, w) = C_{\min}(\alpha, A, S, w) < C^{\alpha}_{\mathrm{greedy}}(\alpha, A, S, w)$. This number is contained in the row of Table 2.1 labeled with "Opt+".

Table 2.1. Results of the first group of experiments with α-covers

α	0.0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
Opt	330	623	674	858	814	711	939	995	1000	1000
Impr	0	53	42	37	13	29	13	2	0	0
Opt+	0	20	27	32	9	28	12	0	0	0

The obtained results show that the percentage of problems for which greedy algorithm with equal thresholds finds an optimal α-cover grows almost monotonically (with local minimum near to 0.4–0.5) from 33% up to 100%. The percentage of problems for which the first modification of greedy algorithm can improve the result of the work of greedy algorithm with equal thresholds is less than 6%. However, sometimes (for example, if $\alpha = 0.3$ or $\alpha = 0.6$) the considered improvement is noticeable.

The Second Group of Experiments

The second group of experiments is connected with comparison of quality of greedy algorithm with equal thresholds and the first modification of greedy algorithm.

We make 25 experiments (row "Nr" in Table 2.2 contains the number of experiment). Each experiment includes the work with three randomly generated families of set cover problems with weights (A, S, w) (1000 problems in each family) such that $|A| = n$, $|S| = m$ and w has values from the set $\{1, \ldots, v\}$.

If the column "n" contains one number, for example "40", it means that $|A| = 40$. If this row contains two numbers, for example "30–120", it means that for each of

1000 problems we choose the number n randomly from the set $\{30, \dots, 120\}$. The same situation is for the column "m".

If the column "α" contains one number, for example "0.1", it means that $\alpha = 0.1$. If this column contains two numbers, for example "0.2–0.4", it means that we choose randomly the value of α such that $0.2 \le \alpha \le 0.4$.

For each of the considered set cover problems with weights (A, S, w) and number α, we apply greedy algorithm with equal thresholds and the first modification of greedy algorithm. Column "$\#i$", $i = 1, 2, 3$, contains the number of problems (A, S, w) from the family number i for each of which the weight of α-cover, constructed by the first modification of greedy algorithm, is less than the weight of α-cover constructed by greedy algorithm with equal thresholds. In other words, in column "$\#i$" we have the number of problems (A, S, w) from the family number i such that there exists γ for

Table 2.2. Results of the second group of experiments with α-covers

Nr	n	m	v	α	#1	#2	#3	avg
1	1–100	1–100	1–10	0–1	1	1	4	2.0
2	1–100	1–100	1–100	0–1	10	13	14	12.33
3	1–100	1–100	1–1000	0–1	15	8	22	15.0
4	1–100	1–100	1–1000	0–0.2	27	23	39	29.66
5	1–100	1–100	1–1000	0.2–0.4	31	27	19	25.66
6	1–100	1–100	1–1000	0.4–0.6	16	14	22	17.33
7	1–100	1–100	1–1000	0.6–0.8	4	7	6	5.66
8	1–100	1–100	1–1000	0.8–1	0	1	0	0.33
9	100	1–30	1–1000	0–0.2	32	26	39	32.33
10	100	30–60	1–1000	0–0.2	40	36	33	36.33
11	100	60–90	1–1000	0–0.2	43	43	53	46.33
12	100	90–120	1–1000	0–0.2	43	45	33	40.33
13	1–30	30	1–1000	0–0.2	21	14	14	16.33
14	30–60	30	1–1000	0–0.2	47	43	40	43.33
15	60–90	30	1–1000	0–0.2	40	40	52	44.0
16	90–120	30	1–1000	0–0.2	32	47	33	37.33
17	40	10	1–1000	0.1	60	57	59	58.66
18	40	10	1–1000	0.2	43	38	37	39.33
19	40	10	1–1000	0.3	29	31	35	31.66
20	40	10	1–1000	0.4	4	13	13	10.0
21	40	10	1–1000	0.5	17	29	21	22.33
22	40	10	1–1000	0.6	10	15	13	12.66
23	40	10	1–1000	0.7	3	1	1	1.66
24	40	10	1–1000	0.8	0	0	0	0.0
25	40	10	1–1000	0.9	0	0	0	0.0

which $0 \leq \gamma < \alpha$ and $C^{\gamma}_{\text{greedy}}(\alpha, A, S, w) < C^{\alpha}_{\text{greedy}}(\alpha, A, S, w)$. The column "avg" contains the number $(\#1 + \#2 + \#3)/3$.

In experiments 1–3 we consider the case, where the parameter v increases. In experiments 4–8 the parameter α increases. In experiments 9–12 the parameter m increases. In experiments 13–16 the parameter n increases. In experiments 17–25 the parameter α increases. The results of experiments show that the value of $\#i$ can change from 0 to 60. It means that the percentage of problems, for which the first modification of greedy algorithm is better than the greedy algorithm with equal thresholds, can change from 0% to 6%.

The Third Group of Experiments

The third group of experiments is connected with investigation of quality of lower bound $C_{\min}(\alpha) \geq \varrho_C(\alpha, \alpha)$.

We choose natural n, m, v and real α, $0 \leq \alpha < 1$. For each chosen tuple (n, m, v, α), we generate randomly 30 set cover problems with weight (A, S, w) such that $|A| = n$, $|S| = m$ and w has values from the set $\{1, ..., v\}$. After that, we find values of $C^{\alpha}_{\text{greedy}}(\alpha, A, S, w)$ and $\varrho_C(\alpha, \alpha, A, S, w)$ for each of generated 30 problems. Note that

$$\varrho_C(\alpha, \alpha, A, S, w) \leq C_{\min}(\alpha, A, S, w) \leq C^{\alpha}_{\text{greedy}}(\alpha, A, S, w) \ .$$

Finally, we find mean values of $C^{\alpha}_{\text{greedy}}(\alpha, A, S, w)$ and $\varrho_C(\alpha, \alpha, A, S, w)$ for generated 30 problems.

Results of experiments can be found in Figs. 2.1 and 2.2. In these figures mean values of $\varrho_C(\alpha, \alpha, A, S, w)$ are called "average lower bound" and mean values of $C^{\alpha}_{\text{greedy}}(\alpha, A, S, w)$ are called "average upper bound".

In Fig. 2.1 (top) one can see the case, where $n \in \{1000, 2000, \ldots, 5000\}$, $m = 30$, $v = 1000$ and $\alpha = 0.01$.

In Fig. 2.1 (bottom) one can see the case, where $n = 1000$, $m \in \{10, 20, \ldots, 100\}$, $v = 1000$ and $\alpha = 0.01$.

In Fig. 2.2 (top) one can see the case, where $n = 1000$, $m = 30$, $v \in \{100, 200, \ldots, 1000\}$ and $\alpha = 0.01$.

In Fig. 2.2 (bottom) one can see the case, where $n = 1000$, $m = 30$, $v = 1000$ and $\alpha \in \{0.0, 0.1, \ldots, 0.9\}$.

Results of experiments show that the considered lower bound is nontrivial and can be useful in investigations.

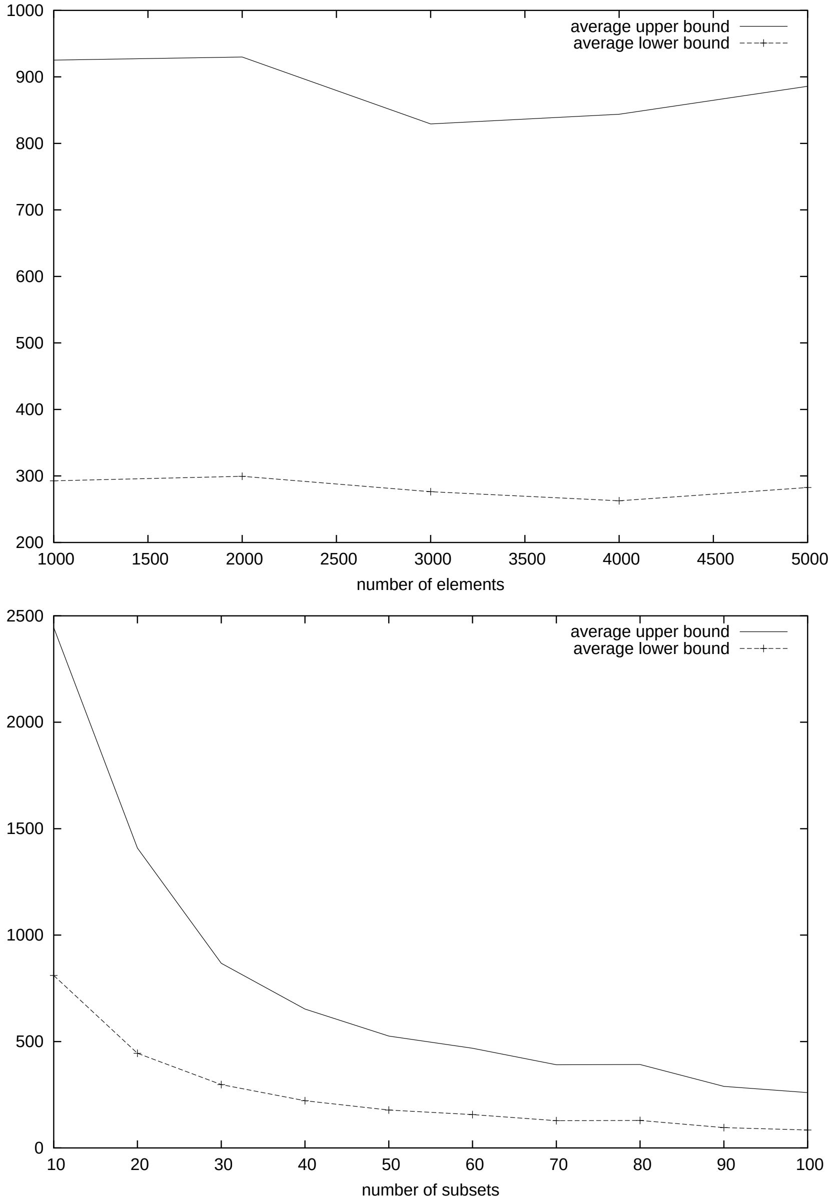

Fig. 2.1. Results of the third group of experiments with α-covers (n and m are changing)

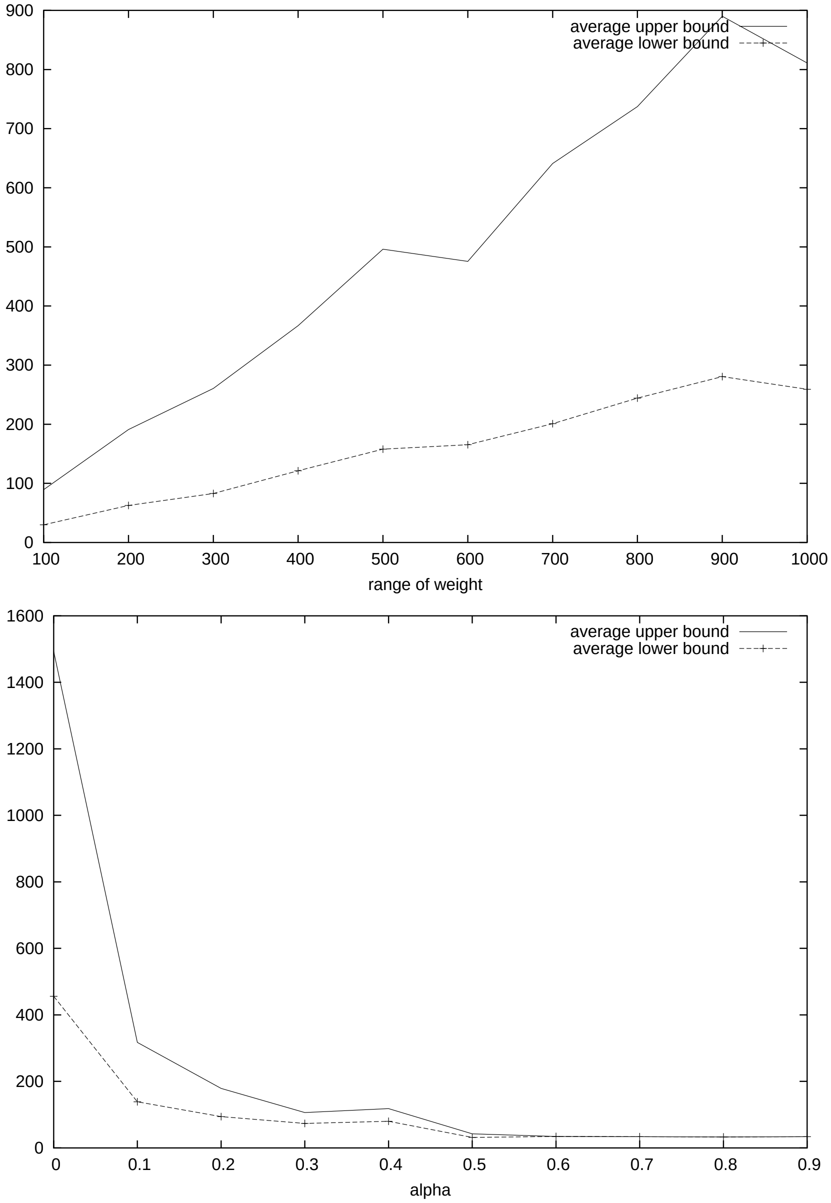

Fig. 2.2. Results of the third group of experiments with α-covers (v and α are changing)

2.2 Partial Decision Rules with Weights

This section consists of seven subsections. In Sect. 2.2.1, main notions are considered. In Sect. 2.2.2, some relationships between partial covers and partial decision rules are discussed. In Sect. 2.2.3, two bounds on precision of greedy algorithm with thresholds α and $\gamma = \alpha$ are considered. In Sect. 2.2.4, polynomial approximate algorithms for partial decision rule weight minimization are studied. Two modifications of greedy algorithm are considered in Sect. 2.2.5. Section 2.2.6 is devoted to consideration of some bounds on minimal weight of partial decision rules and weight of decision rules constructed by greedy algorithm with thresholds α and γ. In Sect. 2.2.7, some experimental results are discussed.

2.2.1 Main Notions

We repeat here some definitions from Chap. 1 and consider generalizations of other definitions to the case of arbitrary natural weights.

Let T be a table with n rows labeled with nonnegative integers (decisions) and m columns labeled with attributes (names of attributes) $f_1, \ldots, f_m$. This table is filled by nonnegative integers (values of attributes). The table T is called a *decision table*. Let w be a *weight function* for T which corresponds to each attribute f_i a natural number $w(f_i)$. Let $r = (b_1, \ldots, b_m)$ be a row of T labeled with a decision d.

Let us denote by $U(T, r)$ the set of rows from T which are different from r and are labeled with decisions different from d. We will say that an attribute f_i *separates* rows r and $r' \in U(T, r)$ if rows r and r' have different numbers at the intersection with the column f_i. For $i = 1, \ldots, m$, we denote by $U(T, r, f_i)$ the set of rows from $U(T, r)$ which attribute f_i separates from the row r.

Let α be a real number such that $0 \leq \alpha < 1$. A decision rule

$$(f_{i_1} = b_{i_1}) \wedge \ldots \wedge (f_{i_t} = b_{i_t}) \rightarrow d \tag{2.3}$$

is called an α-*decision rule* for T and r if attributes $f_{i_1}, \ldots, f_{i_t}$ separate from r at least $(1-\alpha)|U(T, r)|$ rows from $U(T, r)$. The number $\sum_{j=1}^{t} w(f_{i_j})$ is called the *weight* of the considered decision rule.

If $U(T, r) = \emptyset$, then for any $f_{i_1}, \ldots, f_{i_t} \in \{f_1, \ldots, f_m\}$ the rule (2.3) is an α-decision rule for T and r. Also, the rule (2.3) with empty left-hand side (where $t = 0$) is an α-decision rule for T and r. The weight of this rule is equal to 0.

For example, 0.01-decision rule means that we should separate from r at least 99% of rows from $U(T, r)$. Note that 0-rule is usual (exact) rule. Let us denote by $L_{\min}(\alpha) = L_{\min}(\alpha, T, r, w)$ the minimal weight of α-decision rule for T and r.

Let α, γ be real numbers such that $0 \leq \gamma \leq \alpha < 1$. We now describe a *greedy algorithm with thresholds* α *and* γ which constructs an α-decision rule for given T, r and weight function w (see Algorithm 4).

Algorithm 4: Greedy algorithm with two thresholds α and γ for partial decision rule construction

```
Input   : Decision table $T$ with conditional attributes $f_1, \dots, f_m$, row $r = (b_1, \dots, b_m)$ of $T$ labeled
          with the decision $d$, weight function $w : \{f_1, \dots, f_m\} \to \mathbb{N}$, and real numbers $\alpha$ and $\gamma$ such
          that $0 \leq \gamma \leq \alpha < 1$.
Output: $\alpha$-decision rule for $(T, r)$.
$Q \longleftarrow \emptyset$;
$D \longleftarrow \emptyset$;
$M \longleftarrow \lceil |U(T,r)|(1-\alpha) \rceil$;
$N \longleftarrow \lceil |U(T,r)|(1-\gamma) \rceil$;
while $|D| < M$ do
    select $f_i \in \{f_1, \dots, f_m\}$ with minimal index $i$ such that $U(T,r,f_i) \setminus D \neq \emptyset$ and the value
        $$\frac{w(f_i)}{\min\{|U(T,r,f_i) \setminus D|, N - |D|\}}$$
    is minimal;
    $Q \longleftarrow Q \cup \{f_i\}$;
    $D \longleftarrow D \cup U(T,r,f_i)$;
end
return $\bigwedge_{f_i \in Q} (f_i = b_i) \to d$;
```

Let us denote by $L^{\gamma}_{\text{greedy}}(\alpha) = L^{\gamma}_{\text{greedy}}(\alpha, T, r, w)$ the weight of α-decision rule constructed by the considered algorithm for given table T, row r and weight function w.

2.2.2 Relationships Between Partial Covers and Partial Decision Rules

Let (A, S, w) be a set cover problem with weights and α, γ be real numbers such that $0 \leq \gamma \leq \alpha < 1$. We now apply the greedy algorithm with thresholds α and γ to (A, S, w). Let during the construction of α-cover the greedy algorithm choose sequentially subsets $B_{j_1}, \dots, B_{j_t}$ from the family S. We denote $O_C(\alpha, \gamma, A, S, w) = (j_1, \dots, j_t)$.

Let T be a decision table with m columns labeled with attributes $f_1, \dots, f_m$, r be a row from T, and w be a weight function for T. Let $U(T, r)$ be a nonempty set.

We correspond a set cover problem with weights $(A(T,r), S(T,r), u_w)$ to the considered decision table T, row r and weight function w in the following way: $A(T,r) = U(T,r)$, $S(T,r) = \{B_1(T,r), \dots, B_m(T,r)\}$, where $B_1(T,r) = U(T,r,f_1), \dots, B_m(T,r) = U(T,r,f_m)$, and

$$u_w(B_1(T,r)) = w(f_1), \ldots, u_w(B_m(T,r)) = w(f_m) \; .$$

Let α, γ be real numbers such that $0 \le \gamma \le \alpha < 1$. We now apply the greedy algorithm with thresholds α and γ to decision table T, row r and weight function w. Let during the construction of α-decision rule the greedy algorithm choose sequentially attributes $f_{j_1}, \ldots, f_{j_t}$. We denote $O_L(\alpha, \gamma, T, r, w) = (j_1, \ldots, j_t)$.

Set $U(T, r, f_{j_0}) = \emptyset$. For $i = 1, \ldots, t$, we denote $w_i = w(f_{j_i})$ and

$$\delta_i = |U(T, r, f_{j_i}) \setminus (U(T, r, f_{j_0}) \cup \ldots \cup U(T, r, f_{j_{i-1}}))| \; .$$

Set $M_L(\alpha, \gamma, T, r, w) = \lceil |U(T,r)|(1-\alpha) \rceil$, $N_L(\alpha, \gamma, T, r, w) = \lceil |U(T,r)|(1-\gamma) \rceil$, $\mathit{\Delta}_L(\alpha, \gamma, T, r, w) = (\delta_1, \ldots, \delta_t)$ and $W_L(\alpha, \gamma, T, r, w) = (w_1, \ldots, w_t)$.

It is not difficult to prove the following statement.

Proposition 2.22. *Let T be a decision table with m columns labeled with attributes $f_1, \ldots, f_m$, r be a row of T, $U(T,r) \neq \emptyset$, w be a weight function for T, and α, γ be real numbers such that $0 \le \gamma \le \alpha < 1$. Then*

$$\begin{aligned}
|U(T,r)| &= |A(T,r)| \; , \\
|U(T,r,f_i)| &= |B_i(T,r)|, \;\; i = 1, \ldots, m \; , \\
O_L(\alpha, \gamma, T, r, w) &= O_C(\alpha, \gamma, A(T,r), S(T,r), u_w) \; , \\
M_L(\alpha, \gamma, T, r, w) &= M_C(\alpha, \gamma, A(T,r), S(T,r), u_w) \; , \\
N_L(\alpha, \gamma, T, r, w) &= N_C(\alpha, \gamma, A(T,r), S(T,r), u_w) \; , \\
\mathit{\Delta}_L(\alpha, \gamma, T, r, w) &= \mathit{\Delta}_C(\alpha, \gamma, A(T,r), S(T,r), u_w) \; , \\
W_L(\alpha, \gamma, T, r, w) &= W_C(\alpha, \gamma, A(T,r), S(T,r), u_w) \; , \\
L_{\min}(\alpha, T, r, w) &= C_{\min}(\alpha, A(T,r), S(T,r), u_w) \; , \\
L^{\gamma}_{\text{greedy}}(\alpha, T, r, w) &= C^{\gamma}_{\text{greedy}}(\alpha, A(T,r), S(T,r), u_w) \; .
\end{aligned}$$

Let (A, S, w) be a set cover problem with weights, where $A = \{a_1, \ldots, a_n\}$ and $S = \{B_1, \ldots, B_m\}$. We correspond a decision table $T(A,S)$, row $r(A,S)$ of $T(A,S)$ and a weight function v_w for $T(A,S)$ to the set cover problem with weights (A,S,w) in the following way. The table $T(A,S)$ contains m columns labeled with attributes $f_1, \ldots, f_m$ and $n+1$ rows filled by numbers from $\{0,1\}$. For $i = 1, \ldots, n$ and $j = 1, \ldots, m$, at the intersection of i-th row and j-th column the number 1 stays if and only if $a_i \in B_j$. The row number $n+1$ is filled by 0. The first n rows are labeled with the decision 0. The last row is labeled with the decision 1. We denote by $r(A,S)$ the last row of $T(A,S)$. Let $v_w(f_1) = w(B_1), \ldots, v_w(f_m) = w(B_m)$.

For $i = \{1, \ldots, n+1\}$, we denote by r_i the i-th row. It is not difficult to see that $U(T(A,S), r(A,S)) = \{r_1, \ldots, r_n\}$. Let $i \in \{1, \ldots, n\}$ and $j \in \{1, \ldots, m\}$. One can show that the attribute f_j separates the row $r_{n+1} = r(A, S)$ from the row r_i if and only if $a_i \in B_j$.

It is not difficult to prove the following statement.

Proposition 2.23. *Let (A, S, w) be a set cover problem with weights and α, γ be real numbers such that $0 \le \gamma \le \alpha < 1$. Then*

$$\begin{aligned}
|U(T(A,S), r(A,S))| &= |A| \ , \\
O_L(\alpha, \gamma, T(A,S), r(A,S), v_w) &= O_C(\alpha, \gamma, A, S, w) \ , \\
M_L(\alpha, \gamma, T(A,S), r(A,S), v_w) &= M_C(\alpha, \gamma, A, S, w) \ , \\
N_L(\alpha, \gamma, T(A,S), r(A,S), v_w) &= N_C(\alpha, \gamma, A, S, w) \ , \\
\Delta_L(\alpha, \gamma, T(A,S), r(A,S), v_w) &= \Delta_C(\alpha, \gamma, A, S, w) \ , \\
W_L(\alpha, \gamma, T(A,S), r(A,S), v_w) &= W_C(\alpha, \gamma, A, S, w) \ , \\
L_{\min}(\alpha, T(A,S), r(A,S), v_w) &= C_{\min}(\alpha, A, S, w) \ , \\
L^{\gamma}_{\text{greedy}}(\alpha, T(A,S), r(A,S), v_w) &= C^{\gamma}_{\text{greedy}}(\alpha, A, S, w) \ .
\end{aligned}$$

2.2.3 Precision of Greedy Algorithm with Equal Thresholds

The following two statements are simple corollaries of results of Slavík (see Theorems 2.4 and 2.5) and Proposition 2.22.

Theorem 2.24. *Let T be a decision table, r be a row of T, $U(T, r) \ne \emptyset$, w be a weight function for T, and α be a real number such that $0 \le \alpha < 1$. Then $L^{\alpha}_{\text{greedy}}(\alpha) \le L_{\min}(\alpha) H\left(\lceil (1-\alpha)|U(T, r)| \rceil\right)$.*

Theorem 2.25. *Let T be a decision table with m columns labeled with attributes $f_1, \ldots, f_m$, r be a row of T, $U(T, r) \ne \emptyset$, w be a weight function for T, $\alpha \in \mathbb{R}$, $0 \le \alpha < 1$. Then $L^{\alpha}_{\text{greedy}}(\alpha) \le L_{\min}(\alpha) H\left(\max_{i \in \{1,\ldots,m\}} |U(T, r, f_i)|\right)$.*

2.2.4 Polynomial Approximate Algorithms

In this subsection, we consider three theorems which follow immediately from Theorems 1.41–1.43.

Let $0 \le \alpha < 1$. We now consider the following problem: for a given decision table T, row r of T and weight function w for T it is required to find an α-decision rule for T and r with minimal weight.

Theorem 2.26. *Let $0 \le \alpha < 1$. Then the problem of construction of α-decision rule with minimal weight is NP-hard.*

So we should consider polynomial approximate algorithms for minimization of α-decision rule weight.

Theorem 2.27. *Let $\alpha \in \mathbb{R}$ and $0 \le \alpha < 1$. If $NP \not\subseteq DTIME(n^{O(\log \log n)})$, then for any ε, $0 < \varepsilon < 1$, there is no polynomial algorithm that for a given decision table T, row r of T with $U(T, r) \neq \emptyset$ and weight function w for T constructs an α-decision rule for T and r which weight is at most $(1 - \varepsilon)L_{\min}(\alpha, T, r, w) \ln |U(T, r)|$.*

Theorem 2.28. *Let α be a real number such that $0 \le \alpha < 1$. If $P \neq NP$, then there exists $\delta > 0$ such that there is no polynomial algorithm that for a given decision table T, row r of T with $U(T, r) \neq \emptyset$ and weight function w for T constructs an α-decision rule for T and r which weight is at most $\delta L_{\min}(\alpha, T, r, w) \ln |U(T, r)|$.*

From Theorem 2.24 it follows that $L^{\alpha}_{\text{greedy}}(\alpha) \le L_{\min}(\alpha)(1 + \ln |U(T, r)|)$. From this inequality and from Theorem 2.27 it follows that, under the assumption $NP \not\subseteq DTIME(n^{O(\log \log n)})$, the greedy algorithm with two thresholds α and $\gamma = \alpha$ is close to the best polynomial approximate algorithms for minimization of partial decision rule weight. From the considered inequality and from Theorem 2.28 it follows that, under the assumption $P \neq NP$, the greedy algorithm with two thresholds α and $\gamma = \alpha$ is not far from the best polynomial approximate algorithms for minimization of partial decision rule weight.

However, we can try to improve the results of the work of greedy algorithm with two thresholds α and $\gamma = \alpha$ for some part of decision tables.

2.2.5 Two Modifications of Greedy Algorithm

First, we consider binary diagnostic decision tables and prove that, under some assumptions on the number of attributes and rows, for the most part of tables for each row there exists a weight function w and numbers α, γ such that the weight of α-decision rule constructed by the greedy algorithm with thresholds α and γ is less than the weight of α-decision rule constructed by the greedy algorithm with thresholds α and α.

Binary means that the table is filled by numbers from the set $\{0, 1\}$ (all attributes have values from $\{0, 1\}$). *Diagnostic* means that rows of the table are labeled with pairwise different numbers (decisions). Let T be a binary diagnostic decision table with m columns labeled with attributes $f_1, \ldots, f_m$ and with n rows. We will assume that rows of T with numbers $1, \ldots, n$ are labeled with decisions $1, \ldots, n$ respectively. Therefore, the number of considered tables is equal to 2^{mn}. A decision table will be called *simple* if it has no equal rows.

Theorem 2.29. *Let us consider binary diagnostic decision tables with m columns labeled with attributes $f_1, \ldots, f_m$ and $n \geq 5$ rows labeled with decisions $1, \ldots, n$. The fraction of decision tables T, for each of which for each row r of T there exists a weight function w and numbers α, γ such that $0 \leq \gamma < \alpha < 1$ and $L^{\gamma}_{\text{greedy}}(\alpha, T, r, w) < L^{\alpha}_{\text{greedy}}(\alpha, T, r, w)$, is at least*

$$1 - \frac{n3^m}{(n-1)^{m/2-1}} - \frac{n^2}{2^m} \,.$$

Proof. Let T be a decision table and r be a row of T with number $s \in \{1, \ldots, n\}$.

We will say that a decision table T is 1-uniform relatively r if there exists natural p such that, for any attribute f_i of T, if $|U(T, r, f_i)| > 0$, then $|U(T, r, f_i)| \in \{p, p+1\}$. Using reasoning similar to the proof of Theorem 2.10 one can show that if T is not 1-uniform relatively r, then there exists a weight function w and numbers α, γ such that $0 \leq \gamma < \alpha < 1$ and $L^{\gamma}_{\text{greedy}}(\alpha, T, r, w) < L^{\alpha}_{\text{greedy}}(\alpha, T, r, w)$.

We evaluate the number of decision tables which are not 1-uniform relatively each row. Let $(\delta_1, \ldots, \delta_m) \in \{0, 1\}^m$. First, we evaluate the number of simple decision tables for which $r = (\delta_1, \ldots, \delta_m)$ and which are 1-uniform relatively r. Let us consider such a decision table T. It is clear that there exists $p \in \{1, \ldots, n-2\}$ such that for $i = 1, \ldots, m$ the column f_i contains exactly 0 or p, or $p+1$ numbers $\neg\delta_i$. Therefore, the number of considered decision tables is at most $\sum_{p=1}^{n-2} \left(C^p_{n-1} + C^{p+1}_{n-1} + 1\right)^m$. Using Lemma 2.11 we conclude that this number is at most

$$(n-2)\left(3C^{\lfloor (n-1)/2 \rfloor}_{n-1}\right)^m < (n-1)\left(\frac{3 \times 2^{n-1}}{\sqrt{n-1}}\right)^m = \frac{2^{mn-m}3^m}{(n-1)^{m/2-1}} \,.$$

There are 2^m variants for the choice of the tuple $(\delta_1, \ldots, \delta_m)$ and n variants for the choice of the number s of row r. Therefore, the number of simple decision tables, which are 1-uniform relatively at least one row, is at most

$$n2^m \frac{2^{mn-m}3^m}{(n-1)^{m/2-1}} = \frac{n2^{mn}3^m}{(n-1)^{m/2-1}} \,.$$

The number of tables, which are not simple, is at most $n^2 2^{mn-m}$. Hence, the number of tables, which are not 1-uniform for each row, is at least

$$2^{mn} - \frac{n2^{mn}3^m}{(n-1)^{m/2-1}} - n^2 2^{mn-m} \,.$$

Thus, the fraction, considered in the statement of the theorem, is at least

$$1 - \frac{n3^m}{(n-1)^{m/2-1}} - \frac{n^2}{2^m} \,.$$

□

So if $m \geq 6$ and n, $2^m/n^2$ are large enough, then for the most part of binary diagnostic decision tables for each row there exists a weight function w and numbers α, γ such that the weight of α-decision rule constructed by the greedy algorithm with thresholds α and γ is less than the weight of α-decision rule constructed by the greedy algorithm with thresholds α and α.

The obtained results show that the greedy algorithm with two thresholds α and γ is of some interest. Now we consider two polynomial modifications of greedy algorithm which allow us to use advantages of the greedy algorithm with two thresholds α and γ.

Let T be a decision table with m columns labeled with attributes $f_1, \dots, f_m$, $r = (b_1, \dots, b_m)$ be a row of T labeled with decision d, $U(T, r) \neq \emptyset$, w be a weight function for T and α be a real number such that $0 \leq \alpha < 1$.

1. It is impossible to consider effectively all γ such that $0 \leq \gamma \leq \alpha$. Instead of this, we can consider all natural N such that $M \leq N \leq |U(T,r)|$, where $M = \lceil |U(T,r)|(1-\alpha) \rceil$ (see Algorithm 4). For each $N \in \{M, \dots, |U(T,r)|\}$, we apply Algorithm 4 with parameters M and N to T, r and w, and after that choose an α-decision rule with minimal weight among constructed α-decision rules.
2. There exists also an another way to construct an α-decision rule which is not worse than the one obtained under consideration of all N such that $M \leq N \leq |U(T,r)|$. We now apply Algorithm 4 with thresholds α and $\gamma = \alpha$ to T, r and w. Let the algorithm choose sequentially attributes $f_{j_1}, \dots, f_{j_t}$. For each $i \in \{0, \dots, t-1\}$, we find (if it is possible) an attribute f_{l_i} of T with minimal weight $w(f_{l_i})$ such that the rule $(f_{j_1} = b_{j_1}) \wedge \dots \wedge (f_{j_i} = b_{j_i}) \wedge (f_{l_i} = b_{l_i}) \to d$ is an α-decision rule for T and r (if $i = 0$, then it will be the rule $(f_{l_0} = b_{l_0}) \to d$). After that, among constructed α-decision rules $(f_{j_1} = b_{j_1}) \wedge \dots \wedge (f_{j_t} = b_{j_t}) \to d$, ..., $(f_{j_1} = b_{j_1}) \wedge \dots \wedge (f_{j_i} = b_{j_i}) \wedge (f_{l_i} = b_{l_i}) \to d$, ... we choose an α-decision rule with minimal weight. From Proposition 2.30 it follows that the constructed α-decision rule is not worse than the one constructed under consideration of all γ, $0 \leq \gamma \leq \alpha$, or (which is the same) all N, $M \leq N \leq |U(T,r)|$.

Using Propositions 2.13 and 2.22 one can prove the following statement.

Proposition 2.30. *Let T be a decision table, r be a row of T, $U(T,r) \neq \emptyset$, w be a weight function for T and α, γ be real numbers such that $0 \leq \gamma < \alpha < 1$. Let the greedy algorithm with two thresholds α and α, which is applied to T, r and w, choose sequentially attributes $f_{g_1}, \dots, f_{g_t}$. Let the greedy algorithm with two thresholds α and γ, which is applied to T, r and w, choose sequentially attributes $f_{l_1}, \dots, f_{l_k}$. Then either $k = t$ and $(l_1, \dots, l_k) = (g_1, \dots, g_t)$ or $k \leq t$, $(l_1, \dots, l_{k-1}) = (g_1, \dots, g_{k-1})$ and $l_k \neq g_k$.*

2.2.6 Bounds on $L_{\min}(\alpha)$ and $L^{\gamma}_{\rm greedy}(\alpha)$

First, we fix some information about the work of greedy algorithm with two thresholds and find the best lower bound on the value $L_{\min}(\alpha)$ depending on this information.

Let T be a decision table, r be a row of T such that $U(T,r) \neq \emptyset$, w be a weight function for T, and α, γ be real numbers such that $0 \leq \gamma \leq \alpha < 1$. We now apply the greedy algorithm with thresholds α and γ to the decision table T, row r and the weight function w. Let during the construction of α-decision rule the greedy algorithm choose sequentially attributes $f_{g_1}, \ldots, f_{g_t}$.

Let us denote $U(T,r,f_{g_0}) = \emptyset$ and $\delta_0 = 0$. For $i = 1, \ldots, t$, we denote $\delta_i = |U(T,r,f_{g_i}) \setminus (U(T,r,f_{g_0}) \cup \ldots \cup U(T,r,f_{g_{i-1}}))|$ and $w_i = w(f_{g_i})$. As information on the greedy algorithm work we will use numbers

$$M_L = M_L(\alpha,\gamma,T,r,w) = \lceil |U(T,r)|(1-\alpha) \rceil \ ,$$
$$N_L = N_L(\alpha,\gamma,T,r,w) = \lceil |U(T,r)|(1-\gamma) \rceil$$

and tuples

$$\Delta_L = \Delta_L(\alpha,\gamma,T,r,w) = (\delta_1, \ldots, \delta_t) \ ,$$
$$W_L = W_L(\alpha,\gamma,T,r,w) = (w_1, \ldots, w_t) \ .$$

For $i = 0, \ldots, t-1$, we denote

$$\varrho_i = \left\lceil \frac{w_{i+1}(M_L - (\delta_0 + \ldots + \delta_i))}{\min\{\delta_{i+1}, N_L - (\delta_0 + \ldots + \delta_i)\}} \right\rceil \ .$$

Let us define parameter $\varrho_L(\alpha,\gamma) = \varrho_L(\alpha,\gamma,T,r,w)$ as follows:

$$\varrho_L(\alpha,\gamma) = \max\{\varrho_i : i = 0, \ldots, t-1\} \ .$$

We will show that $\varrho_L(\alpha,\gamma)$ is the best lower bound on $L_{\min}(\alpha)$ depending on M_L, N_L, Δ_L and W_L. Next statement follows from Theorem 2.14 and Propositions 2.22 and 2.23.

Theorem 2.31. *For any decision table T, any row r of T with $U(T,r) \neq \emptyset$, any weight function w for T, and any real numbers α, γ, $0 \leq \gamma \leq \alpha < 1$, the inequality $L_{\min}(\alpha,T,r,w) \geq \varrho_L(\alpha,\gamma,T,r,w)$ holds, and there exists a decision table T', a row r' of T' and a weight function w' for T' such that*

$$M_L(\alpha,\gamma,T',r',w') = M_L(\alpha,\gamma,T,r,w), N_L(\alpha,\gamma,T',r',w') = N_L(\alpha,\gamma,T,r,w),$$
$$\Delta_L(\alpha,\gamma,T',r',w') = \Delta_L(\alpha,\gamma,T,r,w), W_L(\alpha,\gamma,T',r',w') = W_L(\alpha,\gamma,T,r,w),$$
$$\varrho_L(\alpha,\gamma,T',r',w') = \varrho_L(\alpha,\gamma,T,r,w), L_{\min}(\alpha,T',r',w') = \varrho_L(\alpha,\gamma,T',r',w').$$

Let us consider a property of the parameter $\varrho_L(\alpha, \gamma)$ which is important for practical use of the bound from Theorem 2.31. Next statement follows from Propositions 2.15 and 2.22.

Proposition 2.32. *Let T be a decision table, r be a row of T with $U(T, r) \neq \emptyset$, w be a weight function for T, and α, γ be real numbers such that $0 \le \gamma \le \alpha < 1$. Then $\varrho_L(\alpha, \alpha, T, r, w) \geq \varrho_L(\alpha, \gamma, T, r, w)$.*

We now study some properties of parameter $\varrho_L(\alpha, \gamma)$ and obtain two upper bounds on the value $L^{\gamma}_{\text{greedy}}(\alpha)$ which do not depend directly on cardinality of the set $U(T, r)$ and cardinalities of subsets $U(T, r, f_i)$.

Next statement follows from Theorem 2.16 and Proposition 2.22.

Theorem 2.33. *Let T be a decision table, r be a row of T with $U(T, r) \neq \emptyset$, w be a weight function for T, $\alpha, \gamma \in \mathbb{R}$ and $0 \le \gamma < \alpha < 1$. Then $L^{\gamma}_{\text{greedy}}(\alpha, T, r, w) < \varrho_L(\gamma, \gamma, T, r, w) \left(\ln\left((1-\gamma)/(\alpha-\gamma)\right) + 1\right)$.*

Corollary 2.34. *Let $\varepsilon \in \mathbb{R}$ and $0 < \varepsilon < 1$. Then for any α, $\varepsilon \le \alpha < 1$, the inequalities $\varrho_L(\alpha, \alpha) \le L_{\min}(\alpha) \le L^{\alpha-\varepsilon}_{\text{greedy}}(\alpha) < \varrho_L(\alpha-\varepsilon, \alpha-\varepsilon)\left(\ln(1/\varepsilon) + 1\right)$ hold.*

For example, $\ln(1/0.01) + 1 < 5.61$ and $\ln(1/0.1) + 1 < 3.31$. The obtained results show that the lower bound $L_{\min}(\alpha) \geq \varrho_L(\alpha, \alpha)$ is nontrivial.

Next statement follows from Theorem 2.18 and Proposition 2.22.

Theorem 2.35. *Let T be a decision table, r be a row of T with $U(T, r) \neq \emptyset$, w be a weight function for T, $\alpha, \gamma \in \mathbb{R}$ and $0 \le \gamma < \alpha < 1$. Then $L^{\gamma}_{\text{greedy}}(\alpha, T, r, w) < L_{\min}(\gamma, T, r, w)\left(\ln\left((1-\gamma)/(\alpha-\gamma)\right) + 1\right)$.*

Corollary 2.36. *$L^{0.3}_{\text{greedy}}(0.5) < 2.26 L_{\min}(0.3)$, $L^{0.1}_{\text{greedy}}(0.2) < 3.20 L_{\min}(0.1)$, $L^{0.001}_{\text{greedy}}(0.01) < 5.71 L_{\min}(0.001)$, $L^{0}_{\text{greedy}}(0.001) < 7.91 L_{\min}(0)$.*

Corollary 2.37. *Let $0 < \alpha < 1$. Then $L^{0}_{\text{greedy}}(\alpha) < L_{\min}(0)\left(\ln(1/\alpha) + 1\right)$.*

Corollary 2.38. *Let ε be a real number, and $0 < \varepsilon < 1$. Then for any α such that $\varepsilon \le \alpha < 1$ the inequalities $L_{\min}(\alpha) \le L^{\alpha-\varepsilon}_{\text{greedy}}(\alpha) < L_{\min}(\alpha - \varepsilon)\left(\ln(1/\varepsilon) + 1\right)$ hold.*

2.2.7 Results of Experiments for α-Decision Rules

In this subsection, we will consider only binary decision tables T with binary decision attributes.

The First Group of Experiments

The first group of experiments is connected with study of quality of greedy algorithm with equal thresholds (where $\gamma = \alpha$ or, which is the same, $N = M$), and comparison of quality of greedy algorithm with equal thresholds and the first modification of greedy algorithm (where for each $N \in \{M, \ldots, |U(T, r)|\}$ we apply greedy algorithm with parameters M and N to decision table, row and weight function and after that choose an α-decision rule with minimal weight among constructed α-decision rules).

We generate randomly 1000 decision tables T, rows r and weight functions w such that T contains 40 rows and 10 conditional attributes $f_1, \ldots, f_{10}$, r is the first row of T, and $1 \leq w(f_i) \leq 1000$ for $i = 1, \ldots, 10$.

For each $\alpha \in \{0.1, \ldots, 0.9\}$, we find the number of triples (T, r, w) for which greedy algorithm with equal thresholds constructs an α-decision rule with minimal weight (an optimal α-decision rule), i.e., $L^{\alpha}_{\text{greedy}}(\alpha, T, r, w) = L_{\min}(\alpha, T, r, w)$. This number is contained in the row of Table 2.3 labeled with "Opt".

We find the number of triples (T, r, w) for which the first modification of greedy algorithm constructs an α-decision rule which weight is less than the weight of α-decision rule constructed by greedy algorithm with equal thresholds, i.e., there exists γ such that $0 \leq \gamma < \alpha$ and $L^{\gamma}_{\text{greedy}}(\alpha, T, r, w) < L^{\alpha}_{\text{greedy}}(\alpha, T, r, w)$. This number is contained in the row of Table 2.3 labeled with "Impr".

Also we find the number of triples (T, r, w) for which the first modification of greedy algorithm constructs an optimal α-decision rule which weight is less than the weight of α-decision rule constructed by greedy algorithm with equal thresholds, i.e., there exists γ such that $0 \leq \gamma < \alpha$ and $L^{\gamma}_{\text{greedy}}(\alpha, T, r, w) = L_{\min}(\alpha, T, r, w) < L^{\alpha}_{\text{greedy}}(\alpha, T, r, w)$. This number is contained in the row of Table 2.3 labeled with "Opt+".

Table 2.3. Results of the first group of experiments with α-decision rules

α	0.0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
Opt	434	559	672	800	751	733	866	966	998	1000
Impr	0	31	51	36	22	27	30	17	1	0
Opt+	0	16	35	28	17	26	25	13	1	0

The obtained results show that the percentage of triples (T, r, w), for which the greedy algorithm with equal thresholds finds an optimal α-decision rule, grows almost monotonically (with local minimum near to 0.4–0.5) from 43.4% up to 100%. The percentage of problems, for which the first modification of greedy algorithm can improve the result of the work of greedy algorithm with equal thresholds, is less than

6%. However, sometimes (for example, if $\alpha = 0.3$, $\alpha = 0.6$ or $\alpha = 0.7$) the considered improvement is noticeable.

The Second Group of Experiments

The second group of experiments is connected with comparison of quality of greedy algorithm with equal thresholds and the first modification of greedy algorithm.

We make 25 experiments (row "Nr" in Table 2.4 contains the number of experiment). Each experiment includes the work with three randomly generated families of triples (T, r, w) (1000 triples in each family) such that T contains n rows and m conditional attributes, r is the first row of T, and w has values from the set $\{1, \ldots, v\}$.

Table 2.4. Results of the second group of experiments with α-decision rules

Nr	n	m	v	α	#1	#2	#3	avg
1	1–100	1–100	1–10	0–1	4	2	4	3.33
2	1–100	1–100	1–100	0–1	7	14	13	11.33
3	1–100	1–100	1–1000	0–1	19	13	15	15.67
4	1–100	1–100	1–1000	0–0.2	20	39	22	27.00
5	1–100	1–100	1–1000	0.2–0.4	28	29	28	28.33
6	1–100	1–100	1–1000	0.4–0.6	22	23	34	26.33
7	1–100	1–100	1–1000	0.6–0.8	7	6	4	5.67
8	1–100	1–100	1–1000	0.8–1	0	1	0	0.33
9	100	1–30	1–1000	0–0.2	35	38	28	33.67
10	100	30–60	1–1000	0–0.2	47	43	31	40.33
11	100	60–90	1–1000	0–0.2	45	51	36	44.00
12	100	90–120	1–1000	0–0.2	37	40	55	44.00
13	1–30	30	1–1000	0–0.2	11	8	9	9.33
14	30–60	30	1–1000	0–0.2	20	22	35	25.67
15	60–90	30	1–1000	0–0.2	30	33	34	32.33
16	90–120	30	1–1000	0–0.2	40	48	38	42.00
17	40	10	1–1000	0.1	31	39	34	34.67
18	40	10	1–1000	0.2	37	39	47	41.00
19	40	10	1–1000	0.3	35	30	37	34.00
20	40	10	1–1000	0.4	27	20	27	24.67
21	40	10	1–1000	0.5	32	32	36	33.33
22	40	10	1–1000	0.6	28	26	24	26.00
23	40	10	1–1000	0.7	10	12	10	10.67
24	40	10	1–1000	0.8	0	2	0	0.67
25	40	10	1–1000	0.9	0	0	0	0.0

If the column "n" contains one number, for example "40", it means that $n = 40$. If this row contains two numbers, for example "30–120", it means that for each of

1000 triples we choose the number n randomly from the set $\{30, \ldots, 120\}$. The same situation is for the column "m".

If the column "α" contains one number, for example "0.1", it means that $\alpha = 0.1$. If this column contains two numbers, for example "0.2–0.4", it means that we choose randomly the value of α such that $0.2 \leq \alpha \leq 0.4$.

For each of the considered triples (T, r, w) and number α, we apply greedy algorithm with equal thresholds and the first modification of greedy algorithm. Column "#i", $i = 1, 2, 3$, contains the number of triples (T, r, w) from the family number i for each of which the weight of α-decision rule, constructed by the first modification of greedy algorithm, is less than the weight of α-decision rule constructed by the greedy algorithm with equal thresholds. In other words, in column "#i" we have the number of triples (T, r, w) from the family number i such that there exists γ for which $0 \leq \gamma < \alpha$ and $L^{\gamma}_{\text{greedy}}(\alpha, T, r, w) < L^{\alpha}_{\text{greedy}}(\alpha, T, r, w)$. The column "avg" contains the number $(\#1 + \#2 + \#3)/3$.

In experiments 1–3 we consider the case, where the parameter v increases. In experiments 4–8 the parameter α increases. In experiments 9–12 the parameter m increases. In experiments 13–16 the parameter n increases. In experiments 17–25 the parameter α increases. The results of experiments show that the value of $\#i$ can change from 0 to 55. It means that the percentage of triples, for which the first modification of greedy algorithm is better than the greedy algorithm with equal thresholds, can change from 0% to 5.5%.

The Third Group of Experiments

The third group of experiments is connected with investigation of quality of lower bound $L_{\min}(\alpha) \geq \varrho_L(\alpha, \alpha)$.

We choose natural n, m, v and real α, $0 \leq \alpha < 1$. For each chosen tuple (n, m, v, α), we generate randomly 30 triples (T, r, w) such that T contains n rows and m conditional attributes, r is the first row of T, and w has values from the set $\{1, ..., v\}$. After that, we find values of $L^{\alpha}_{\text{greedy}}(\alpha, T, r, w)$ and $\varrho_L(\alpha, \alpha, T, r, w)$ for each of generated 30 triples. Note that $\varrho_L(\alpha, \alpha, T, r, w) \leq L_{\min}(\alpha, T, r, w) \leq L^{\alpha}_{\text{greedy}}(\alpha, T, r, w)$. Finally, for generated 30 triples we find mean values of $L^{\alpha}_{\text{greedy}}(\alpha, T, r, w)$ and $\varrho_L(\alpha, \alpha, T, r, w)$.

Results of experiments can be found in Figs. 2.3 and 2.4. In these figures mean values of $\varrho_L(\alpha, \alpha, T, r, w)$ are called "average lower bound" and mean values of $L^{\alpha}_{\text{greedy}}(\alpha, T, r, w)$ are called "average upper bound".

In Fig. 2.3 (top) one can see the case, where $n \in \{1000, 2000, \ldots, 5000\}$, $m = 30$, $v = 1000$ and $\alpha = 0.01$.

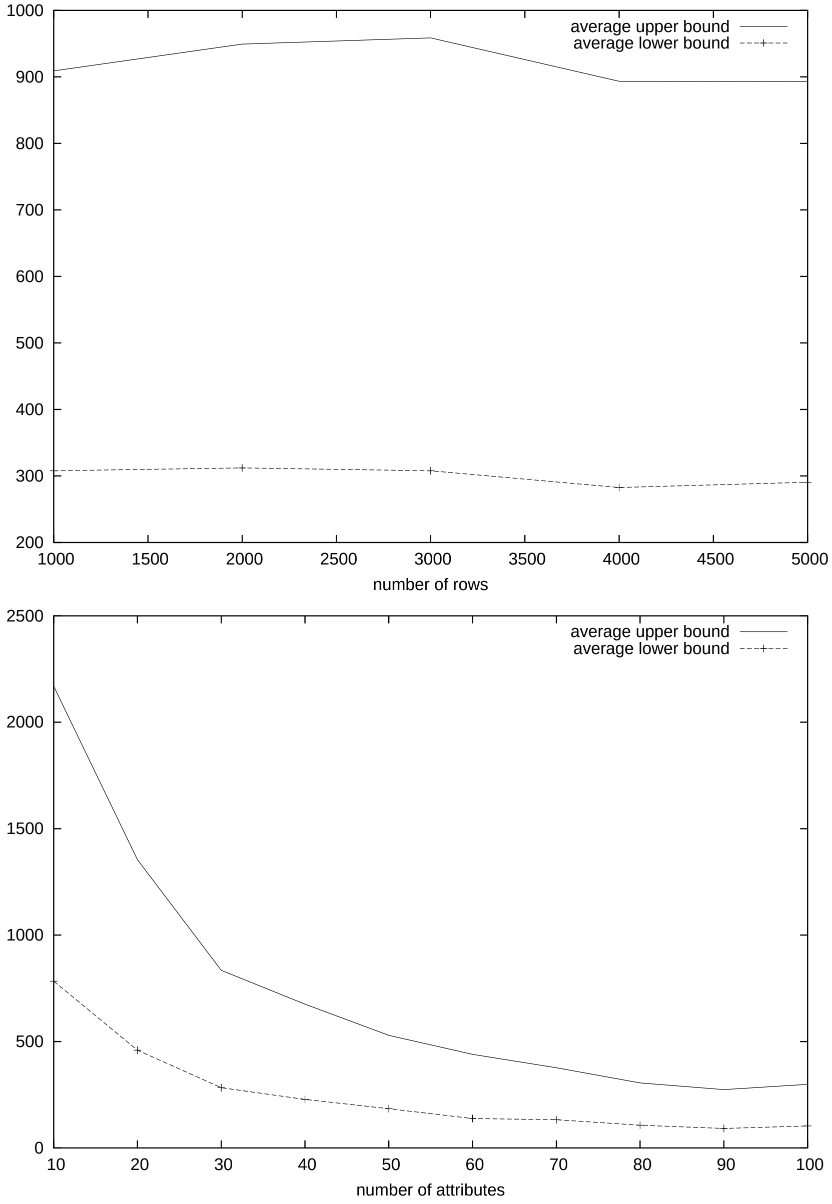

Fig. 2.3. Results of the third group of experiments with rules (n and m are changing)

In Fig. 2.3 (bottom) one can see the case, where $n = 1000$, $m \in \{10, 20, \ldots, 100\}$, $v = 1000$ and $\alpha = 0.01$.

In Fig. 2.4 (top) one can see the case, where $n = 1000$, $m = 30$, $v \in \{100, 200, \ldots, 1000\}$ and $\alpha = 0.01$.

In Fig. 2.4 (bottom) one can see the case, where $n = 1000$, $m = 30$, $v = 1000$ and $\alpha \in \{0.0, 0.1, \ldots, 0.9\}$.

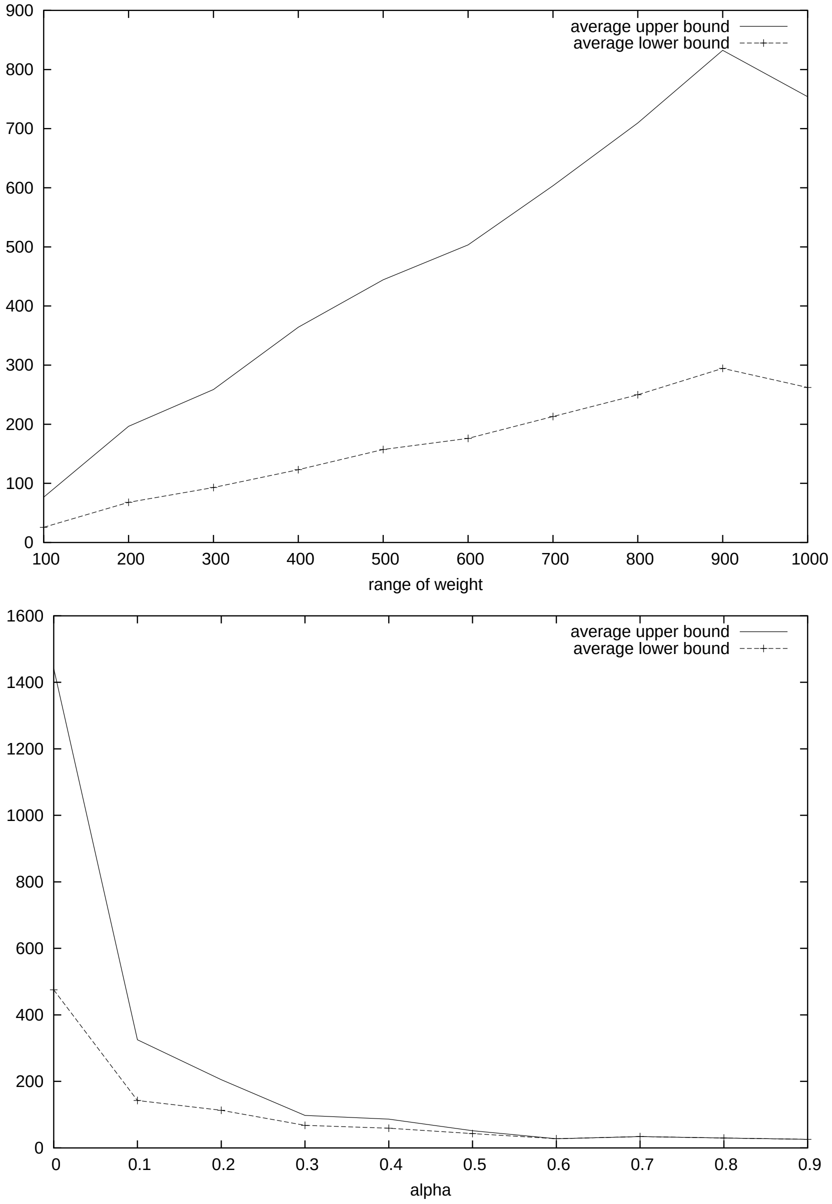

Fig. 2.4. Results of the third group of experiments with rules (v and α are changing)

Results of experiments show that the considered lower bound is nontrivial and can be useful in investigations.

2.3 Conclusions

The chapter is devoted (mainly) to theoretical and experimental analysis of greedy algorithms with weights for partial cover and decision rule construction.

Theoretical and experimental results show that the lower bounds on minimal weight of partial covers and decision rules, based on an information about greedy algorithm work, are nontrivial and can be used in practice.

Based on greedy algorithm with two thresholds we create new polynomial approximate algorithms for minimization of weights of partial covers and decision rules. Results of massive experiments with randomly generated set cover problems and binary decision tables show that these new algorithms can be useful in applications.

3

Construction of All Irreducible Partial Decision Rules

In this chapter, we study problem of construction of all irreducible partial decision rules. Efficient solution of this problem would allow (i) to find the best partial rules; (ii) to evaluate the importance of attributes; (iii) to create ensembles of classifiers; (iv) to evaluate changes after adding new objects into a decision table.

We consider binary decision tables with m conditional attributes, in which the number of rows is equal to $\lfloor m^{\alpha} \rfloor$, where α is a positive real number, and partial decision rules that can leave unseparated from a given row at most $5 \left\lceil (\log_2 m)^{\beta} \right\rceil$ different rows with different decisions, where β is a real number such that $\beta \geq 1$.

We show that for almost all such tables for any row with minor decision (minor decision is a decision which is attached to at most one-half of rows of decision table) the length of each irreducible partial decision rule is not far from $\alpha \log_2 m$ and the number of irreducible partial decision rules is not far from $m^{\alpha \log_2 m}$.

Based on these results, we prove that there is no algorithm which for almost all decision tables for each row with minor decision constructs the set of irreducible partial decision rules and has for these tables polynomial time complexity depending on the length of input. However, there exists an algorithm which for almost all decision tables for each row with minor decision constructs the set of irreducible partial decision rules and has for these tables polynomial time complexity depending on the length of input and the length of output.

This chapter is based on paper [37].

The chapter contains two sections. Section 3.1 contains description of the set $TAB_D(m, n)$ of decision tables which is used in Sect. 3.2. In Sect. 3.2, results for irreducible t-decision rules are discussed. Section 3.3 contains short conclusions.

3.1 Set $TAB_D(m, n)$ of Decision Tables

A *binary information system* is a table with n rows (objects) and m columns labeled with attributes (names of attributes) $f_1, \ldots, f_m$. This table is filled by numbers from the set $\{0, 1\}$ (values of attributes). The number of binary information systems with n rows and m columns is equal to 2^{mn}.

If for $i = 1, \ldots, n$ we attach to i-th row of a binary information system a natural number d_i (a decision), we obtain a *binary decision table*. In this decision table attributes $f_1, \ldots, f_m$ are called *conditional attributes*. The tuple $(d_1, \ldots, d_n)$ is called a *decision attribute*. A decision attribute $(d_1, \ldots, d_n)$ is called *degenerate* if $d_1 = \ldots = d_n$. Let D be a finite set of non-degenerate decision attributes. Then the cardinality of the set $TAB_D(m, n)$ of binary decision tables with n rows, m columns and decision attributes from D is equal to $|D|2^{mn}$.

Let us consider two examples of sets D of non-degenerate decision attributes: the set $\{1, 2\}^n \setminus \{(1, \ldots, 1), (2, \ldots, 2)\}$ of binary decision attributes, and the set of decision attributes $\{1, \ldots, n\}^n \setminus \{(1, \ldots, 1), \ldots, (n, \ldots, n)\}$ which allow us to simulate an arbitrary non-degenerate decision attribute for a decision table with n rows. Later we will assume that a finite set $D = D(n)$ of non-degenerate decision attributes is fixed for any n.

Let $\mathcal{P}$ be a property of decision tables and let $\mathcal{P}_D(m, n)$ be the number of decision tables from $TAB_D(m, n)$ for which $\mathcal{P}$ holds. The number $\mathcal{P}_D(m, n)/(|D|2^{mn})$ is called the *fraction* of decision tables from $TAB_D(m, n)$ for which the property $\mathcal{P}$ holds.

Let α be a positive real number. We consider also decision tables from the set $TAB_D(m, \lfloor m^\alpha \rfloor)$. We say that the property $\mathcal{P}$ holds *for almost all* decision tables from $TAB_D(m, \lfloor m^\alpha \rfloor)$ if the fraction $\mathcal{P}_D(m, \lfloor m^\alpha \rfloor)$ / $(|D|2^{m\lfloor m^\alpha \rfloor})$ of decision tables from $TAB_D(m, \lfloor m^\alpha \rfloor)$, for which the property $\mathcal{P}$ holds, tends to 1 as m tends to infinity.

3.2 Irreducible t-Decision Rules

This section consists of four subsections. In Sect. 3.2.1, bounds on the length of irreducible t-decision rules are obtained. In Sect. 3.2.2, bounds on the number of irreducible t-decision rules are considered. In Sect. 3.2.3, algorithms for construction of all irreducible t-decision rules are studied. In Sect. 3.2.4, results of some experiments with irreducible t-decision rules are discussed.

Let T be a decision table from $TAB_D(m, n)$ with n rows, m conditional attributes $f_1, \ldots, f_m$ and decision attribute $(d_1, \ldots, d_n)$. Let $r = (b_1, \ldots, b_m)$ be the row of T

with the number i. This row is labeled with the decision d_i. We will say that d_i is a *minor decision*, and r is a *row with minor decision* if

$$|\{j : j \in \{1, \ldots, n\}, d_j = d_i\}| \le \frac{n}{2} \,.$$

We denote by $U(T, r)$ the set of rows from T which are different from r and are labeled with decisions different from d_i. We will say that an attribute f_j *separates* rows r and $r' \in U(T, r)$ if rows r and r' have different numbers at the intersection with the column f_j.

Let t be a natural number. A decision rule

$$(f_{j_1} = b_{j_1}) \wedge \ldots \wedge (f_{j_p} = b_{j_p}) \Rightarrow d_i$$

is called a *t-decision rule* for T and r if attributes $f_{j_1}, \ldots, f_{j_p}$ separate from r at least $|U(T, r)| - t$ rows from the set $U(T, r)$. In this case we will say that attributes $f_{j_1}, \ldots, f_{j_p}$ *generate* a t-decision rule for T and r. Later we will consider only rules $(f_{j_1} = b_{j_1}) \wedge \ldots \wedge (f_{j_p} = b_{j_p}) \Rightarrow d_i$ for which $j_1 < \ldots < j_p$. The number p is called the *length* of the rule.

If we remove some conditions $f_{j_s} = b_{j_s}$, $s \in \{1, \ldots, p\}$, from the considered rule we obtain its *subrule*. A subrule of some rule is called *proper* if it is not equal to the initial rule. A t-decision rule for T and r is called *irreducible* if each proper subrule of this rule is not a t-decision rule for T and r.

3.2.1 Length of Irreducible t-Decision Rules

In this subsection, we consider lower and upper bounds on the length of irreducible t-decision rules for decision tables from $TAB_D(m, n)$ and rows with minor decisions, where $t = 5\left\lceil (\log_2 m)^\beta \right\rceil$ and β is a real number such that $\beta \ge 1$. Under some assumptions on m and n, we evaluate the fraction of decision tables for which the considered bounds hold for any irreducible t-decision rule for any row with minor decision.

Theorem 3.1. *Let m, n be natural numbers, $t = 5\left\lceil (\log_2 m)^\beta \right\rceil$, where β is a real number such that $\beta \ge 1$, and $\kappa = 2\lceil \log_2 n \rceil$. Then the fraction of decision tables from $TAB_D(m, n)$, for which for any row any κ attributes generate a t-decision rule, is at least $1 - 1/2^{\lceil \log_2 m \rceil \lceil \log_2 n \rceil}$.*

Proof. Let us consider a decision table T obtained from a binary information system by adding a decision attribute from D. Let $i_0 \in \{1, \ldots, n\}$, and let conditional attributes $f_{l_1}, \ldots, f_{l_\kappa}$ do not form a t-decision rule for T and the row with number i_0. Then there exist pairwise different numbers $j_1, \ldots, j_{t+1} \in \{1, \ldots, n\} \setminus \{i_0\}$ such that rows with numbers $j_1, \ldots, j_{t+1}$ coincide with the row with number i_0 at the intersection with columns $f_{l_1}, \ldots, f_{l_\kappa}$.

We now fix a number $i_0 \in \{1, \ldots, n\}$, $t+1$ pairwise different numbers $j_1, \ldots, j_{t+1} \in \{1, \ldots, n\} \setminus \{i_0\}$ and κ conditional attributes $f_{l_1}, \ldots, f_{l_\kappa}$. The number of binary information systems such that rows with numbers $j_1, \ldots, j_{t+1}$ coincide with the row with number i_0 at the intersection with columns $f_{l_1}, \ldots, f_{l_\kappa}$ is at most $2^{mn-\kappa(t+1)}$. There are at most m^κ variants for the choice of κ columns. There are at most n^{t+2} variants for the choice of numbers $i_0, j_1, \ldots, j_{t+1}$. Therefore, the number of decision tables T, in each of which there exists a row r and κ conditional attributes that do not generate a t-decision rule for T and r, is at most $|D|\, 2^{mn+\kappa \log_2 m+(t+2)\log_2 n-\kappa(t+1)} \leq |D|\, 2^{mn-\lceil \log_2 m \rceil \lceil \log_2 n \rceil}$. Then the fraction of decision tables, for which for any row any κ conditional attributes generate a t-decision rule, is at least

$$\frac{|D|\, 2^{mn} - |D|\, 2^{mn-\lceil \log_2 m \rceil \lceil \log_2 n \rceil}}{|D|\, 2^{mn}} = 1 - \frac{1}{2^{\lceil \log_2 m \rceil \lceil \log_2 n \rceil}} \; .$$

□

Theorem 3.2. *Let $m, n \in \mathbb{N}$, $m \geq 2\log_2 n + c$, where $c \in \mathbb{N}$, $c \geq 2$, $t = 5\left\lceil (\log_2 m)^\beta \right\rceil$, where $\beta \in \mathbb{R}$, $\beta \geq 1$, $\lceil n/2 \rceil > t$,*

$$\varrho = \left\lfloor \log_2 \left(\left\lceil \frac{n}{2} \right\rceil - t \right) - 3 - \log_2 \left\lceil (\log_2 m)^\beta \right\rceil - \log_2 \lceil \log_2 n \rceil \right\rfloor$$

and $\varrho > 0$. Then the fraction of decision tables from $TAB_D(m,n)$, for which for each row with minor decision any ϱ condition attributes do not generate a t-decision rule, is at least $1 - 1/2^{\min(c, \lceil \log_2 n \rceil \lceil \log_2 m \rceil)-1}$.

Proof. A binary information system will be called strongly separable if for any $i, j \in \{1, \ldots, n\}$ such that $i \neq j$ rows with numbers i and j are different. The number of binary information systems, for which rows with numbers i and j are equal, is equal to 2^{mn-m}. There are at most n^2 variants for the choice of i and j. Therefore, the number of binary information systems, which are not strongly separable, is at most $n^2 2^{mn-m} = 2^{mn+2\log_2 n-m}$. Thus, the number of strongly separable information systems is at least $2^{mn} - 2^{mn+2\log_2 n-m}$.

Let $\bar{d} = (d_1, \ldots, d_n) \in D$, T be a decision table obtained from a strongly separable information system I by adding the decision attribute $\bar{d}$, $i_0 \in \{1, \ldots, n\}$ and d_{i_0} be a minor decision. Then there are $p = \lceil n/2 \rceil$ pairwise different numbers $l_1, \ldots, l_p \in \{1, \ldots, n\}$ such that $d_{i_0} \neq d_{l_s}$ for $s = 1, \ldots, p$. Let $f_{i_1}, \ldots, f_{i_\varrho}$ generate a t-decision rule for T and row with the number i_0. Then among rows with numbers $l_1, \ldots, l_p$ at least $p - t$ rows are different from the row with number i_0 at the intersection with columns $f_{i_1}, \ldots, f_{i_\varrho}$.

We now fix t numbers $l_{j_1},\dots,l_{j_t}$ from $\{l_1,\dots,l_p\}$. Let us evaluate the number of information systems I such that for any $s \in \{l_1,\dots,l_p\} \setminus \{l_{j_1},\dots,l_{j_t}\}$ rows with numbers i_0 and l_s are different at the intersection with columns $f_{i_1},\dots,f_{i_\varrho}$. It is not difficult to see that the number of such information systems is equal to

$$2^{mn-\varrho(p-t)}(2^\varrho-1)^{p-t} = 2^{mn}\left(\frac{2^\varrho-1}{2^\varrho}\right)^{p-t} = 2^{mn}\left(\frac{2^\varrho-1}{2^\varrho}\right)^{2^\varrho(p-t)/2^\varrho} .$$

Using well known inequality $((u-1)/u)^u \le 1/e$, which holds for any natural u, we obtain

$$2^{mn}\left(\frac{2^\varrho-1}{2^\varrho}\right)^{2^\varrho(p-t)/2^\varrho} \le 2^{mn-(p-t)/2^\varrho} .$$

There are at most m^ϱ variants for the choice of ϱ attributes. There are at most n^t variants for the choice of t numbers $l_{j_1},\dots,l_{j_t}$. Therefore, the number of strongly separable information systems I, for which adding the decision attribute $\bar{d}$ can lead to obtaining a decision table that have a t-decision rule with ϱ attributes for the row with the number i_0, is at most $m^\varrho n^t 2^{mn-(p-t)/2^\varrho} = 2^{mn+\varrho\log_2 m+t\log_2 n-(p-t)/2^\varrho}$. There are at most n variants for the choice of the number i_0. Thus, the number of information systems I, for which adding the decision attribute $\bar{d}$ can lead to obtaining a decision table that have a t-decision rule with ϱ attributes for some row with minor decision, is at most $2^{mn+\varrho\log_2 m+(t+1)\log_2 n-(p-t)/2^\varrho}+2^{mn+2\log_2 n-m}$, where $2^{mn+2\log_2 n-m}$ is an upper bound on the number of information systems which are not strongly separable.

It is clear that and $\varrho\log_2 m+(t+1)\log_2 n \le 7\lceil\log_2 n\rceil\left\lceil(\log_2 m)^\beta\right\rceil$ and $(p-t)/2^\varrho \ge (p-t)8\lceil\log_2 n\rceil\left\lceil(\log_2 m)^\beta\right\rceil/(p-t) = 8\lceil\log_2 n\rceil\left\lceil(\log_2 m)^\beta\right\rceil$. We now obtain $2^{mn+\varrho\log_2 m+(t+1)\log_2 n-(p-t)/2^\varrho} \le 2^{mn-\lceil\log_2 n\rceil\left\lceil(\log_2 m)^\beta\right\rceil} \le 2^{mn-\lceil\log_2 n\rceil\lceil\log_2 m\rceil}$. Since $m \ge 2\log_2 n+c$, we have $2^{mn+2\log_2 n-m} \le 2^{mn-c}$. From here it follows that $2^{mn+\varrho\log_2 m+(t+1)\log_2 n-(p-t)/2^\varrho}+2^{mn+2\log_2 n-m} \le 2^{mn-\min(c,\lceil\log_2 n\rceil\lceil\log_2 m\rceil)+1}$.

Thus, the fraction of decision tables, for which any ϱ conditional attributes do not generate a t-decision rule for any row with minor decision, is at least

$$\frac{|D|2^{mn}-|D|2^{mn-\min(c,\lceil\log_2 n\rceil\lceil\log_2 m\rceil)+1}}{|D|2^{mn}} = 1-\frac{1}{2^{\min(c,\lceil\log_2 n\rceil\lceil\log_2 m\rceil)-1}} .$$

□

Corollary 3.3. *Let $m,n \in \mathbb{N}$, $m \ge 2\log_2 n+c$, where $c \in \mathbb{N}$, $c \ge 2$, $t = 5\left\lceil(\log_2 m)^\beta\right\rceil$, where $\beta \in \mathbb{R}$ and $\beta \ge 1$, $\lceil n/2\rceil > t$, $\kappa = 2\lceil\log_2 n\rceil$,*

$$\varrho = \left\lfloor \log_2\left(\left\lceil\frac{n}{2}\right\rceil - t\right) - 3 - \log_2\left\lceil(\log_2 m)^\beta\right\rceil - \log_2\lceil\log_2 n\rceil \right\rfloor$$

and $\varrho > 0$. Then the fraction of decision tables from $TAB_D(m,n)$, for which for each row with minor decision any κ conditional attributes generate a t-decision rule, and any ϱ condition attributes do not generate a t-decision rule, is at least $1 - 1/2^{\lceil \log_2 m \rceil \lceil \log_2 n \rceil} - 1/2^{\min(c, \lceil \log_2 n \rceil \lceil \log_2 m \rceil) - 1}$.

Corollary 3.4. *Let $m, n \in \mathbb{N}$, $m \geq 2 \log_2 n + c$, where $c \in \mathbb{N}$, $c \geq 2$, $t = 5 \left\lceil (\log_2 m)^\beta \right\rceil$, where $\beta \in \mathbb{R}$ and $\beta \geq 1$, $\lceil n/2 \rceil > t$, $\kappa = 2 \lceil \log_2 n \rceil$,*

$$\varrho = \left\lfloor \log_2 \left(\left\lceil \frac{n}{2} \right\rceil - t \right) - 3 - \log_2 \left\lceil (\log_2 m)^\beta \right\rceil - \log_2 \lceil \log_2 n \rceil \right\rfloor$$

and $\varrho > 0$. Then the fraction of decision tables from $TAB_D(m,n)$, for which for each row with minor decision the length of any irreducible t-decision rule is at most κ and at least $\varrho + 1$, is at least $1 - 1/2^{\lceil \log_2 m \rceil \lceil \log_2 n \rceil} - 1/2^{\min(c, \lceil \log_2 n \rceil \lceil \log_2 m \rceil) - 1}$.

3.2.2 Number of Irreducible t-Decision Rules

Let T be a decision table and r be a row of T with minor decision. We denote by $R(T, r, t)$ the number of irreducible t-decision rules for T and r. In this subsection, we consider decision tables from the set $TAB_D(m, \lfloor m^\alpha \rfloor)$, where $\alpha \in \mathbb{R}$ and $\alpha > 0$. We study irreducible t-decision rules, where $t = 5 \left\lceil (\log_2 m)^\beta \right\rceil$ and β is a real number such that $\beta \geq 1$. We present lower and upper bounds on the value $R(T, r, t)$ for almost all decision tables $T \in TAB_D(m, \lfloor m^\alpha \rfloor)$ and for each row r of T with minor decision.

Theorem 3.5. *Let $m \in \mathbb{N}$, $\alpha \in \mathbb{R}$, $\alpha > 0$, $t = 5 \left\lceil (\log_2 m)^\beta \right\rceil$, where β is a real number such that $\beta \geq 1$, and $\kappa = 2 \lceil \log_2 \lfloor m^\alpha \rfloor \rceil$. Then for almost all decision tables T from $TAB_D(m, \lfloor m^\alpha \rfloor)$ for any row r with minor decision any κ conditional attributes generate a t-decision rule, and $m^{(\alpha/4) \log_2 m} \leq R(T, r, t) \leq m^{3\alpha \log_2 m}$.*

Proof. Let $n = \lfloor m^\alpha \rfloor$. We now prove that for large enough m the fraction of decision tables T from $TAB_D(m, \lfloor m^\alpha \rfloor)$, for which for any row r with minor decision any κ conditional attributes generate a t-decision rule, and $m^{(\alpha/4) \log_2 m} \leq R(T, r, t) \leq m^{3\alpha \log_2 m}$, is at least $1 - 1/2^{\lceil \log_2 m \rceil \lceil \log_2 n \rceil - 2}$.

Let $\varrho = \left\lfloor \log_2 \left(\lceil n/2 \rceil - t \right) - 3 - \log_2 \left\lceil (\log_2 m)^\beta \right\rceil - \log_2 \lceil \log_2 n \rceil \right\rfloor$. From Corollary 3.3 it follows that for large enough m the fraction of decision tables, for which for any row with minor decision any κ conditional attributes generate a t-decision rule, and any ϱ conditional attributes do not generate a t-decision rule, is at least

$$1 - \frac{1}{2^{\lceil \log_2 m \rceil \lceil \log_2 n \rceil}} - \frac{1}{2^{\lceil \log_2 n \rceil \lceil \log_2 m \rceil - 1}} \geq 1 - \frac{1}{2^{\lceil \log_2 m \rceil \lceil \log_2 n \rceil - 2}} .$$

Let us consider an arbitrary decision table T and an arbitrary row r of T with minor decision for which any κ conditional attributes generate a t-decision rule, and any ϱ conditional attributes do not generate a t-decision rule. We now show that $m^{(\alpha/4)\log_2 m} \leq R(T,r,t) \leq m^{3\alpha \log_2 m}$ for large enough m.

It is clear that each t-decision rule has an irreducible t-decision rule as a subrule. Let Q be an irreducible t-decision rule. We now evaluate the number of t-decision rules of the length κ which have Q as a subrule. Let the length of Q is equal to p. One can show that $\varrho + 1 \leq p \leq \kappa$. There are $C_{m-p}^{\kappa-p}$ ways to obtain a t-decision rule of the length κ from Q by adding conditional attributes from $\{f_1, \ldots, f_m\}$. It it clear that $C_{m-p}^{\kappa-p} \leq C_m^{\kappa-p}$. If $\kappa < m/2$, then $C_m^{\kappa-p} \leq C_m^{\kappa-\varrho}$. Thus, for large enough m the number of t-decision rules of the length κ, which have Q as a subrule, is at most $C_m^{\kappa-\varrho}$.

The number of t-decision rules of the length κ is equal to C_m^{κ}. Hence,

$$R(T,r,t) \geq \frac{C_m^{\kappa}}{C_m^{\kappa-\varrho}} = \frac{(m-\kappa+1)\ldots(m-\kappa+\varrho)}{(\kappa-\varrho+1)\ldots\kappa} \geq \left(\frac{m-\kappa}{\kappa}\right)^{\varrho} .$$

For large enough m,

$$\frac{m-\kappa}{\kappa} = \frac{m - 2\lceil \log_2 \lfloor m^{\alpha} \rfloor \rceil}{2\lceil \log_2 \lfloor m^{\alpha} \rfloor \rceil} \geq m^{1/2} .$$

Therefore, $R(T,r,t) \geq m^{\varrho/2}$. It is clear that for large enough m the inequality $\varrho \geq (1/2)\alpha \log_2 m$ holds. Thus, for large enough m, $R(T,r,t) \geq m^{(\alpha/4)\log_2 m}$.

It is clear that the length of each irreducible t-decision rule is at most κ. Therefore, $R(T,r,t) \leq m^{\kappa}$. One can show that for large enough m the inequality $m^{\kappa} \leq m^{3\alpha \log_2 m}$ holds. Thus, $R(T,r,t) \leq m^{3\alpha\log_2 m}$ for large enough m. It is clear that $1 - 1/2^{\lceil \log_2 m \rceil \lceil \log_2 \lfloor m^{\alpha} \rfloor \rceil - 2}$ tends to 1 as m tends to infinity. Therefore, the statement of the theorem holds. □

3.2.3 Algorithms for Construction of All Irreducible t-Decision Rules

We study irreducible t-decision rules for decision tables from $TAB_D(m, \lfloor m^{\alpha} \rfloor)$, where α is a positive real number, $t = 5\left\lceil (\log_2 m)^{\beta} \right\rceil$, β is a real number, and $\beta \geq 1$. For a given decision table T and row r of T with minor decision, it is required to find all irreducible t-decision rules for T and r. For large enough m, the length of input for this problem is at least $m\lfloor m^{\alpha} \rfloor$ and at most $m\lfloor m^{\alpha} \rfloor + (\lfloor m^{\alpha} \rfloor + 1)\lceil \log_2 \lfloor m^{\alpha} \rfloor \rceil \leq m^{1+\alpha} + m^{2\alpha} \leq m^{2(1+\alpha)}$. The length of output for this problem is at least $R(T,r,t)$ and at most $mR(T,r,t)$.

Let $\kappa = 2\lceil \log_2 \lfloor m^{\alpha} \rfloor \rceil$. From Theorem 3.5 it follows that for almost all decision tables T from $TAB_D(m, \lfloor m^{\alpha} \rfloor)$ for any row r with minor decision any κ conditional attributes generate a t-decision rule and $m^{(\alpha/4)\log_2 m} \leq R(T,r,t) \leq m^{3\alpha \log_2 m}$.

Thus, there is no algorithm which for almost all decision tables from $TAB_D(m, \lfloor m^{\alpha} \rfloor)$ for each row with minor decision constructs the set of irreducible t-decision rules and has for these tables polynomial time complexity depending on the length of input.

Let us consider an algorithm which finds all nonempty subsets of the set $\{f_1, \ldots, f_m\}$ with at most κ attributes, and for each such subset recognizes if attributes from this subset generate an irreducible t-decision rule or not. It is clear that this recognition problem can be solved (for one subset) in polynomial time depending on the length of input.

From Theorem 3.5 it follows that for almost all decision tables from $TAB_D(m, \lfloor m^\alpha \rfloor)$ for any row with minor decision this algorithm finds all irreducible t-decision rules.

The considered algorithm works with at most m^κ subsets of $\{f_1, \ldots, f_m\}$. One can show that $m^\kappa \le m^{3\alpha \log_2 m}$ for large enough m. Using Theorem 3.5 we conclude that for almost all decision tables T from $TAB_D(m, \lfloor m^\alpha \rfloor)$ for any row r of T with minor decision $m^\kappa \le R(T, r, t)^{12}$.

Thus, there exists an algorithm which for almost all decision tables from $TAB_D(m, \lfloor m^\alpha \rfloor)$ constructs for any row with minor decision the set of irreducible t-decision rules and has for these tables polynomial time complexity depending on the length of input and the length of output.

3.2.4 Results of Experiments

We generate randomly 1000 binary decision tables T with 40 rows, 10 conditional attributes and binary decision attribute. As row r we choose the first row of T. For each table T, we find the minimal length of irreducible 5-decision rule for T and r, the maximal length of irreducible 5-decision rule for T and r and the number of irreducible 5-decision rules for T and r. Results of experiments are represented in Figs. 3.1–3.3.

These results illustrate the situation, where irreducible t-decision rules have relatively small length, and the number of irreducible t-decision rules is relatively small. The consideration of another values of t can lead to different results.

In Fig. 3.1 for each $i \in \{1, 2\}$ one can see the number of tables for which the minimal length of irreducible 5-decision rule is equal to i. For each $i \in \{0, 3, 4, \ldots, 10\}$, the considered number is equal to 0.

In Fig. 3.2 for each $i \in \{1, 2, 3, 4\}$ one can see the number of tables for which the maximal length of irreducible 5-decision rule is equal to i. For each $i \in \{0, 5, 6, \ldots, 10\}$, the considered number is equal to 0.

One can show that the number of irreducible 5-decision rules for the considered tables and rows is at most 252. In Fig. 3.3 for each $i \in \{0, 1, \ldots, 90\}$ one can see the number of tables for which the number of irreducible 5-decision rules is equal to i. For each $i \in \{91, 92, \ldots, 252\}$, the considered number is equal to 0.

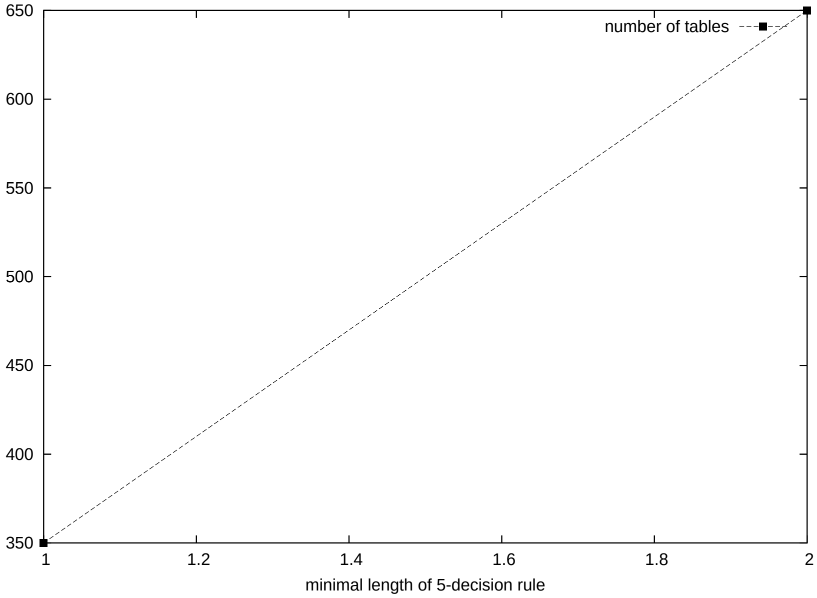

Fig. 3.1. Number of tables with given minimal length of irreducible 5-decision rule

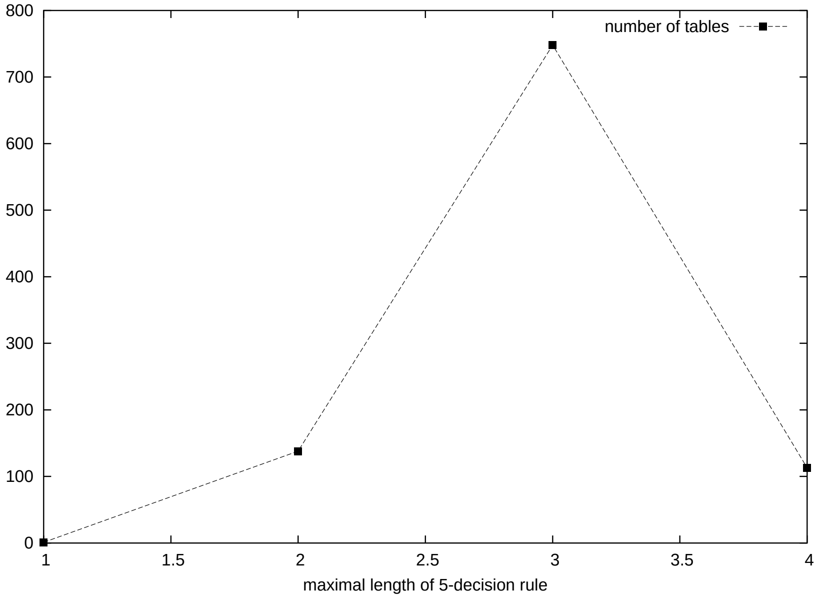

Fig. 3.2. Number of tables with given maximal length of irreducible 5-decision rule

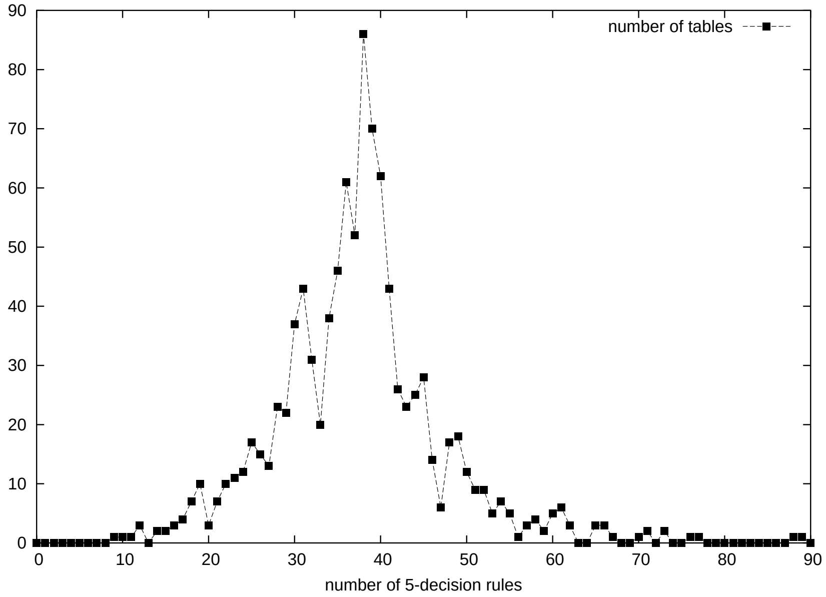

Fig. 3.3. Number of tables with given number of irreducible 5-decision rules

3.3 Conclusions

In the chapter, we show that, under some assumptions, there is no algorithm which for almost all decision tables for each row with minor decision constructs all irreducible t-decision rules and has for these tables polynomial time complexity depending on the length of input, but there exists an algorithm which for almost all decision tables for each row with minor decision constructs all irreducible t-decision rules and has for these tables polynomial time complexity depending on the total length of input and output.

The obtained results is a step towards the design of algorithms for construction of the set of all irreducible partial decision rules.

4

Experiments with Real-Life Decision Tables

This chapter is devoted to consideration of results of experiments with decision tables from UCI Repository of Machine Learning Databases [41]. The aim of the first group of experiments is to verify 0.5-hypothesis for real-life decision tables. We made experiments with 23 decision tables. Results of 20 experiments confirm 0.5-hypothesis for decision rules: under the construction of partial decision rule, during each step the greedy algorithm chooses an attribute which separates from r at least one-half of unseparated rows that are different from r and have other decisions.

The aim of the second group of experiments is the comparison of accuracy of classifiers based on exact and partial decision rules. The considered approach to construction of classifiers is the following: for a given decision table and each row we construct a (partial) decision rule using greedy algorithm. By removing some attributes from this (partial) decision rule we obtain an irreducible (partial) decision rule. The obtained system of rules jointly with simple procedure of voting can be considered as a classifier.

We made experiments with 21 decision tables using test-and-train method. In 11 cases, we found partial decision rules for which the accuracy of the constructed classifiers is better than the accuracy of classifiers based on exact decision rules. We made also experiments with 17 decision tables using cross-validation method. In 9 cases, we found partial decision rules for which the accuracy of the constructed classifiers is better than the accuracy of classifiers based on exact decision rules.

The results of experiments obtained for classifiers based on partial decision rules are comparable with the results of experiments for some classifiers from RSES [54].

This chapter is based on papers [81, 82].

The chapter consists of three sections. In Sect. 4.1, 0.5-hypothesis is considered for decision rules. In Sect. 4.2, classifiers are considered based on partial decision rules. Section 4.3 contains short conclusions.

4.1 0.5-Hypothesis for Decision Rules

Results of experiments with randomly generated decision tables and some theoretical results (see Chap. 1) confirm the following 0.5-hypothesis for decision rules: for the most part of decision tables for each row r, under the construction of partial decision rule, during each step the greedy algorithm chooses an attribute which separates from r at least one-half of unseparated rows that are different from r and have other decisions. It is not difficult to show that in such cases $L_{\text{greedy}}(\alpha) \leq \lceil \log_2(1/\alpha) \rceil$ for $\alpha > 0$, and $l_{DR}(\alpha) \leq 2$ for any α. In particular, $L_{\text{greedy}}(0.1) \leq 4$, $L_{\text{greedy}}(0.01) \leq 7$, and $L_{\text{greedy}}(0.001) \leq 10$. So using greedy algorithm it is possible to construct short partial decision rules with relatively high accuracy.

To verify this hypothesis for real-life decision tables we made additional experiments with the following 23 decision tables from [41]: “balance-scale", “balloons (adult+stretch)", “car", “flags", “hayes-roth.test", “krkopt", “kr-vs-kp", “monks-1.test", “monks-1.train", “monks-2.test", “monks-2.train", “monks-3.test", “monks-3.train", “lenses", “letter-recognition", “lymphography", “poker-hand-training.true", “nursery", “soybean-small", “spect_all", “shuttle-landing-control", “tic-tac-toe", and “zoo".

We apply to each of the considered tables and to each row of these tables the greedy algorithm with $\alpha = 0$. The main result of these experiments is the following: with the exception of the tables “kr-vs-kp", “spect_all" and “nursery" for each row r, under the construction of partial decision rule, during each step the greedy algorithm chooses an attribute which separates from r at least one-half of unseparated rows that are different from r and have other decisions. It means that not only for randomly generated, but also for real-life decision tables it is possible to construct short partial decision rules with relatively high accuracy using greedy algorithm.

Table 4.1 presents the average percentage of rows from $U(T, r)$, unseparated from the row r during the first $i - 1$ steps, which are separated from the row r at i-th step of the greedy algorithm, $i = 1, \ldots, 11$, under partial decision rule construction with parameter $\alpha = 0$. The column “Decision table” contains the name of decision table, the column “n” contains the number of rows in the table, and the column “m” contains the number of conditional attributes.

From the results presented in Table 4.1 it follows that the average percentage of rows separated at i-th step of greedy algorithm during partial decision rule construction is at least 50%. However, for “nursery", “spect_all" and “kr-vs-kp" we can find rows for which during some steps the greedy algorithm chooses attributes that separate less than 50% of unseparated rows.

Table 4.1. Average percentage of rows separated at i-th step of the greedy algorithm during partial decision rule construction

			Number of step i										
Decision table	n	m	1	2	3	4	5	6	7	8	9	10	11
soybean-small	47	35	100										
balloons	20	4	86.7	100									
monks-3.test	432	6	90.6	100									
shuttle-landing	15	6	99.5	100									
hayes-roth.test	28	4	88.2	97.5	100								
monks-1.test	432	6	83.3	77.8	100								
balance-scale	625	4	89.5	91.5	96.1	100							
flags	194	26	96.7	97.9	95.9	100							
lenses	24	4	84.5	61.5	91.7	100							
lymphography	148	18	91.7	95.5	98.4	100							
monks-1.train	124	6	84.6	81.6	94.9	100							
monks-3.train	122	6	89.0	95.6	93.7	100							
zoo	101	16	96.7	95.6	90.7	100							
poker-hand	25010	10	92.6	93.9	95.8	99.8	100						
tic-tac-toe	958	9	79.1	79.1	87.7	94.1	100						
car	1728	6	90.6	81.6	80.2	85.0	85.7	100					
krkopt	28056	6	89.8	88.7	88.6	89.7	92.2	100					
letter-recognition	20000	16	97.3	96.5	97.8	99.2	99.6	100					
monks-2.test	432	6	75.2	70.2	75.1	77.1	67.4	100					
monks-2.train	169	6	76.0	76.8	86.0	87.1	92.6	100					
nursery	12960	8	89.1	84.0	88.2	88.5	93.0	89.6	91.6	100			
spect_all	267	22	86.9	81.0	75.8	68.6	53.1	52.1	50.0	51.8	88.0	100	
kr-vs-kp	3196	36	91.0	86.0	89.1	91.8	87.0	85.7	86.7	83.7	76.7	75.0	100

Table 4.2 presents minimum, average and maximum length of α-decision rules constructed by the greedy algorithm for $\alpha \in \{0.0, 0.001, 0.01, 0.1\}$.

Table 4.2 gives us some information about maximum, minimum and average length of partial decision rules constructed by the greedy algorithm. For example, for the table “kr-vs-kp", which contains 36 conditional attributes, the maximum length of exact decision rule is equal to 11. Results presented in Table 4.2 show that the greedy algorithm constructs relatively short partial decision rules with relatively high accuracy.

Table 4.2. Minimum, average and maximum length of partial decision rules constructed by the greedy algorithm

Decision table	α											
	0.0			0.001			0.01			0.1		
	min	avg	max	min	avg	max	min	avg	max	min	avg	max
balance-scale	3.0	3.2	4.0	3.0	3.2	4.0	2.0	2.3	3.0	1.0	1.5	2.0
balloons(adult+stretch)	1.0	1.4	2.0	1.0	1.4	2.0	1.0	1.4	2.0	1.0	1.4	2.0
car	1.0	2.5	6.0	1.0	2.4	6.0	1.0	2.0	4.0	1.0	1.4	2.0
flags	1.0	2.0	4.0	1.0	2.0	4.0	1.0	1.7	3.0	1.0	1.1	2.0
hayes-roth.test	1.0	2.0	3.0	1.0	2.0	3.0	1.0	2.0	3.0	1.0	1.7	2.0
krkopt	3.0	5.2	6.0	2.0	3.8	4.0	2.0	2.7	3.0	1.0	1.6	2.0
kr-vs-kp	1.0	3.0	11.0	1.0	2.8	10.0	1.0	2.3	6.0	1.0	1.5	3.0
lenses	1.0	2.1	4.0	1.0	2.1	4.0	1.0	2.1	4.0	1.0	1.9	3.0
letter-recognition	1.0	3.0	6.0	1.0	2.3	4.0	1.0	1.7	3.0	1.0	1.0	2.0
lymphography	1.0	2.1	4.0	1.0	2.1	4.0	1.0	2.1	4.0	1.0	1.5	2.0
monks-1.test	1.0	2.3	3.0	1.0	2.3	3.0	1.0	2.3	3.0	1.0	1.8	2.0
monks-1.train	1.0	2.3	4.0	1.0	2.3	3.0	1.0	2.3	4.0	1.0	1.8	3.0
monks-2.test	3.0	4.9	6.0	3.0	4.9	6.0	3.0	4.1	5.0	2.0	2.0	2.0
monks-2.train	3.0	3.7	6.0	3.0	3.7	6.0	3.0	3.4	6.0	2.0	2.0	3.0
monks-3.test	1.0	1.8	2.0	1.0	1.8	2.0	1.0	1.8	2.0	1.0	1.5	2.0
monks-3.train	2.0	2.3	4.0	2.0	2.3	4.0	2.0	2.3	4.0	1.0	1.5	2.0
nursery	1.0	3.3	8.0	1.0	2.9	6.0	1.0	2.4	4.0	1.0	1.7	2.0
poker-hand-training-true	3.0	3.9	5.0	3.0	3.0	3.0	2.0	2.0	2.0	1.0	1.0	1.0
shuttle-landing-control	1.0	1.1	2.0	1.0	1.1	2.0	1.0	1.1	2.0	1.0	1.0	1.0
soybean-small	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
spect_all	1.0	3.2	10.0	1.0	3.2	10.0	1.0	2.9	10.0	1.0	1.6	7.0
tic-tac-toe	3.0	3.8	5.0	3.0	3.8	5.0	3.0	3.1	4.0	2.0	2.0	3.0
zoo	1.0	1.5	4.0	1.0	1.5	4.0	1.0	1.5	4.0	1.0	1.1	2.0

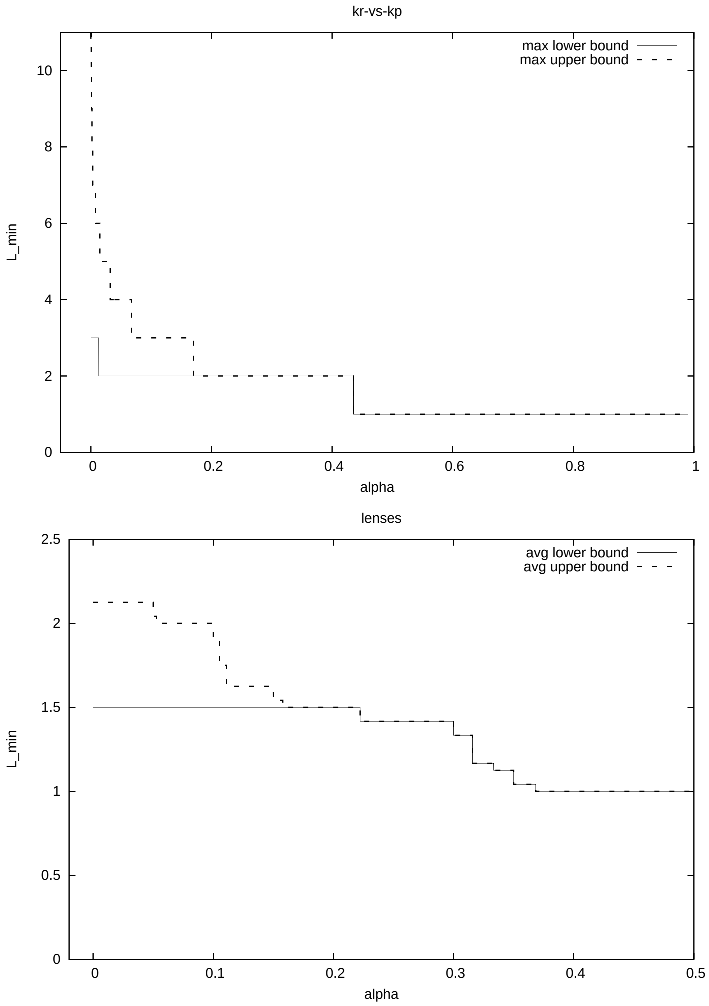

Fig. 4.1. Lower and upper bounds on minimal length of α-decision rules for “kr-vs-kp" and “lenses"

Figure 4.1 presents lower ($l_{DR}(\alpha)$, Theorem 1.46) and upper ($L_{\text{greedy}}(\alpha)$, Theorem 1.45) bounds on minimal length of α-decision rules for decision tables “kr-vs-kp" and “lenses". In the case of “kr-vs-kp", we consider maximum values of lower and upper bounds among all rows. In the case of “lenses", we consider average values of lower and upper bounds for all rows.

4.2 Classifiers Based on Partial Decision Rules

In this section, we compare accuracies of classifiers based on exact and partial decision rules, and some classifiers from RSES [54].

We consider the following problem of classification (prediction): for a given decision table T and a new object v given by values of conditional attributes from T for v it is required to generate a decision corresponding to v.

We now describe classifiers based on partial decision rules.

For every row r of the decision table T and given α, $0 \leq \alpha < 1$, we construct an α-decision rule for T and r by Algorithm 2. After that, by removing some conditions from this α-decision rule we obtain an irreducible α-decision rule for T and r. From the constructed set of irreducible α-decision rules we remove repeating rules. We denote the obtained set by $Rul(T, \alpha)$. For each rule from $Rul(T, \alpha)$, we compute the support of this rule which is the number of rows from T such that (i) the left-hand side of the rule is true for the considered row; (ii) the decision attached to the row is equal to the decision from the right-hand side of the rule.

The set $Rul(T, \alpha)$ can be considered as a classifier which for a given new object v creates a decision for this object using only values of conditional attributes for v. For each possible decision d, we compute the sum $M_d(v)$ of supports of rules from $Rul(T, \alpha)$ such that (i) the left-hand side of the considered rule is true for v, and (ii) the right-hand side of the rule is equal to d. If $M_d(v) > 0$ for at least one decision d, then we choose a decision d for which $M_d(v)$ has maximal value. Otherwise, we choose some fixed decision d_0.

To evaluate the accuracy of classifiers, we can use either train-and-test method or k-fold-cross-validation method. In the first case, we split the initial decision table into training and testing tables, construct a classifier using training table, and apply this classifier to rows from the testing table as to new objects. The accuracy of classification is the number of rows (objects) from the testing table, which are properly classified, divided by the number of rows in the testing table. In the second case, we split the initial decision table into k tables, and k times apply train-and-test method using each of k tables as the testing table. As a result, we obtain k accuracies of classification. The mean of these accuracies is considered as the “final” accuracy of classification.

We study decision tables from [41]. We remove from the table “flags" attributes “area", “population" and “name of the country", and consider “landmass" as the decision attribute. From the table “zoo" we remove the attribute “animal name".

We make experiments with 21 decision tables using train-and-test method. We randomly split decision tables in proportion 70% for training table and 30% for testing table. For “hayes-roth", “monks1", “monks2", “monks3" and “spect", we use existing training and testing tables.

For each table (with the exception of “kr-vs-kp" and “letter-recognition") we choose minimal $\alpha \in \{0.000, 0.001, 0.002, \ldots, 0.300\}$ for which the accuracy of constructed classifier is maximal. This value of α is denoted by α_{opt}. For “kr-vs-kp" and “letter-recognition", we choose minimal $\alpha \in \{0.00, 0.01,\ 0.02, \ldots, 0.50\}$ for which the accuracy of constructed classifier is maximal. This value of α is denoted by α_{opt}. The results of experiments can be found in Table 4.3 (“balloons (a+s)” means “balloons (adult+stretch)”, and “balloons (y-s+a-s)” means “balloons (yellow-small+adult-stretch)”). The use of partial decision rules (α-decision rules with $\alpha > 0$) leads to improvement of accuracy of classification for 11 decision tables.

Table 4.3. Accuracy of classifiers based on partial decision rules (train-and-test)

Decision table	Accuracy for $\alpha = 0$	Accuracy for $\alpha = \alpha_{\text{opt}}$	α_{opt}
balance	0.658	0.866	0.133
balloons (a+s)	1.000	1.000	0.000
balloons (y-s+a-s)	0.600	0.800	0.286
car	0.890	0.909	0.005
flags	0.627	0.678	0.019
hayes-roth	0.893	0.893	0.000
krkopt	0.386	0.433	0.001
kr-vs-kp	0.734	0.956	0.01
lenses	0.500	0.500	0.000
letter-recognition	0.221	0.221	0.00
lymphography	0.733	0.822	0.217
monks1	0.949	0.949	0.000
monks2	0.762	0.762	0.000
monks3	0.931	0.963	0.050
nursery	0.974	0.974	0.000
shuttle-landing	0.600	0.800	0.200
soybean-small	1.000	1.000	0.000
spect	0.818	0.840	0.075
spect_all	0.877	0.889	0.025
tic-tac-toe	0.931	0.931	0.000
zoo	0.968	0.968	0.000

We make also experiments with 17 decision tables using 10-fold-cross-validation method. For each table we choose minimal $\alpha \in \{0.000, 0.001, 0.002,\ \ldots, 0.300\}$ for which the accuracy of constructed classifier is maximal. This value of α is denoted by α_{opt}. Results of experiments can be found in Table 4.4. The use of partial decision rules (α-decision rules with $\alpha > 0$) leads to improvement of accuracy of classification for 9 decision tables.

Table 4.4. Accuracy of classifiers based on partial decision rules (cross-validation)

Decision table	Accuracy for $\alpha = 0$	Accuracy for $\alpha = \alpha_{\text{opt}}$	α_{opt}
balance	0.723	0.891	0.150
balloons (a+s)	1.000	1.000	0.000
balloons (y-s+a-s)	0.750	0.750	0.000
car	0.873	0.905	0.004
flags	0.608	0.613	0.007
hayes-roth	0.790	0.797	0.014
lenses	0.583	0.617	0.278
lymphography	0.778	0.805	0.040
monks1	1.000	1.000	0.000
monks2	0.565	0.671	0.290
monks3	1.000	1.000	0.000
shuttle-landing	0.450	0.450	0.000
soybean-small	0.980	0.980	0.000
spect	0.915	0.920	0.134
spect_all	0.851	0.862	0.029
tic-tac-toe	0.959	0.959	0.000
zoo	0.951	0.951	0.000

We compare accuracies of classifiers based on partial decision rules (really, a modification of these classifiers) and accuracies of some classifiers constructed by algorithms from RSES.

We make experiments with 21 tables from [41], presented in Table 4.3, using train-and-test method. Let T be one of these tables. As it was described earlier, we split this table into two subtables: training table T_{train} and testing table T_{test}. For the table T_{train}, we construct seven sets of decision rules: $Rul(0) = Rul(T_{\text{train}}, 0)$, $Rul(\alpha_{\text{opt}}) = Rul(T_{\text{train}}, \alpha_{\text{opt}})$, where α_{opt} is taken from Table 4.3, $Lem2(1)$, $Lem2(0.9)$, constructed by the "lem2 algorithm" from RSES for T_{train} with "cover parameter" equals to 1 and 0.9 respectively, $Cov(1)$, $Cov(0.9)$, constructed by the "covering algorithm" from RSES for T_{train} with "cover parameter" equals to 1 and 0.9 respectively, and Gen, constructed by the "genetic algorithm" from RSES for T_{train} with "number of reducts" equals to 10 and "normal speed".

The system RSES works with these sets of rules as with classifiers using "standard voting" which assigns to each rule the weight that is equal to the support of this rule: the number of rows from T_{train} such that (i) the left-hand side of the rule is true for the considered row; (ii) the decision attached to the row is equal to the decision from the right-hand side of the rule. RSES applies these classifiers to rows of the table T_{test} as to new objects.

Table 4.5. Comparison of accuracy of classifiers

Decision table	$\mathit{Rul}(0)$	$\mathit{Rul}(\alpha_{\text{opt}})$	$\mathit{Lem}2(1)$	$\mathit{Lem}2(0.9)$	$\mathit{Cov}(1)$	$\mathit{Cov}(0.9)$	Gen
balance	0.626	0.856	0.428	0.369	0.000	0.428	0.749
balloons (a+s)	1.000	1.000	0.833	0.833	0.500	0.500	1.000
balloons (y-s+a-s)	1.000	1.000	0.400	0.400	0.400	0.400	0.800
car	0.892	0.911	0.839	0.728	0.538	0.538	0.911
flags	0.475	0.526	0.305	0.305	0.102	0.102	0.627
hayes-roth	0.822	0.822	0.500	0.321	0.036	0.036	0.786
krkopt	0.379	0.404	0.128	0.121	0.000	0.314	0.444
kr-vs-kp	0.988	0.988	0.881	0.781	0.209	0.209	0.964
lenses	0.625	0.625	0.375	0.375	0.375	0.375	1.000
letter-recognition	0.702	0.719	0.589	0.571	0.035	0.317	
lymphography	0.800	0.889	0.512	0.489	0.111	0.111	0.867
monks1	0.949	0.949	0.743	0.632	0.250	0.250	0.866
monks2	0.715	0.715	0.620	0.563	0.000	0.291	0.736
monks3	0.921	0.954	0.694	0.660	0.000	0.768	0.944
nursery	0.961	0.961	0.908	0.836	0.344	0.344	
shuttle-landing	0.400	0.400	0.600	0.600	0.400	0.400	0.600
soybean-small	1.000	1.000	1.000	1.000	1.000	1.000	1.000
spect	0.743	0.781	0.252	0.235	0.246	0.246	0.759
spect_all	0.716	0.741	0.370	0.296	0.210	0.210	0.765
tic-tac-toe	0.885	0.885	0.917	0.806	0.000	0.264	0.962
zoo	0.968	0.968	0.935	0.935	0.613	0.613	1.000

Accuracies of the considered seven classifiers are represented in Table 4.5. The results obtained for classifiers based on partial rules (columns "$\mathit{Rul}(0)$" and "$\mathit{Rul}(\alpha_{\text{opt}})$") are comparable with the results obtained for classifiers from RSES (columns "$\mathit{Lem}2(1)$", "$\mathit{Lem}2(0.9)$", "$\mathit{Cov}(1)$", "$\mathit{Cov}(0.9)$", and "Gen").

4.3 Conclusions

In the chapter, the greedy algorithm for construction of partial decision rules is considered. Results of experiments show that for real-life decision tables the use of this algorithm allows us to obtain short partial decision rules with relatively high accuracy. These results confirm 0.5-hypothesis for decision rules.

Results of experiments with real-life decision tables show that classifiers based on partial decision rules are often better than the classifier based on exact decision rules.

5

Universal Attribute Reduction Problem

The attribute reduction problem (it is required to find a reduct with minimal or close to minimal cardinality) is one of the main problems of rough set theory [45, 47, 48, 49, 56, 70] and related theories such as test theory [6, 10, 65, 73, 76, 77] and LAD [1, 9]. There are different variants of the notion of reduct: reducts for information systems [45], usual decision and local reducts for decision tables [45, 55], decision and local reducts which are based on the generalized decision [55], etc. Interesting discussion of various kinds of reducts can be found in [47].

In this chapter, we consider an "universal" definition of reduct which covers at least part of possible variants. We use an approach considered in test theory [73]. Let T be a decision table and $\mathcal{P}$ be a subset of pairs of different (discernible) rows (objects) of T. Then a reduct for T relative to $\mathcal{P}$ is a minimal (relative to inclusion) subset of conditional attributes which separate all pairs from $\mathcal{P}$. All mentioned above kinds of reducts can be represented in such a form. We consider here not only exact, but also partial (approximate) reducts.

We begin our consideration from a data table which columns are labeled with discrete and continuous variables, and rows are tuples of values of variables on some objects. It is possible that this data table contains missing values [13, 21]. We consider the following classification problem: for a discrete variable we must find its value using values of all other variables. We do not use variables directly, but create some attributes with relatively small number of values based on the considered variables. As a result, we obtain a decision table with missing values in the general case. We define the universal attribute reduction problem for this table and consider a number of examples of known attribute reduction problems which can be represented as the universal one.

Based on results from Chap. 1, we obtain bounds on precision of greedy algorithm for partial test (super-reduct) construction. This algorithm is a simple generalization of greedy algorithm for set cover problem [16, 24, 44, 57, 58]. We prove that, under some natural assumptions on the class NP, the greedy algorithm is close to the best (from the point of view of precision) polynomial approximate algorithms for minimization of cardinality of partial tests. We show that based on an information received during greedy algorithm work it is possible to obtain a nontrivial lower bound on minimal cardinality of partial reduct. We obtain also a bound on precision of greedy algorithm which does not depend on the cardinality of the set $\mathcal{P}$.

This chapter is based on papers [36, 39].

The chapter consists of four sections. In Sect. 5.1, a transformation of a data table into a decision table is considered. In Sect. 5.2, the notion of the universal attribute reduction problem is discussed. In Sect. 5.3, greedy algorithm for construction of partial tests (partial super-reducts) is studied. Section 5.4 contains short conclusions.

5.1 From Data Table to Decision Table

A *data table* D is a table with k columns labeled with variables $x_1, \ldots, x_k$ and N rows which are interpreted as tuples of values of variables $x_1, \ldots, x_k$ on N objects $u_1, \ldots, u_N$. It is possible that D contains *missing values* which are denoted by " $-$ ".

As usual, we assume that each of variables x_i is either *discrete* (with values from some finite unordered set $V(x_i)$) or *continuous* (with values from a set $V(x_i) \subset \mathbb{R}$). We will assume that " $-$ " does not belong to $V(x_i)$.

Let us choose a variable $x_r \in \{x_1, \ldots, x_k\}$ and consider the problem of prediction of the value of x_r on a given object using only values of variables from the set $X = \{x_1, \ldots, x_k\} \setminus \{x_r\}$ on the considered object. If x_r is a discrete variable, then the problem of prediction is called the classification problem. If x_r is a continuous variable, then the considered problem is called the problem of regression. We consider only the classification problem. So x_r is a discrete variable.

We consider only two kinds of missing values: (i) missing value of x_i as an additional value of variable x_i which does not belong to $V(x_i)$, and (ii) missing value as an undefined value. In the last case, based on the value of x_i it is impossible to discern an object u_l from another object u_t if the value $x_i(u_l)$ is missing (undefined).

We now transform the data table D into a data table D^*. For each variable $x_i \in \{x_1, \ldots, x_k\}$, according to the nature of x_i we choose either the first or the second way for the work with missing values. In the first case, we add to $V(x_i)$ a new value which is not equal to " $-$ ", and write this new value instead of each missing value of x_i. In the second case, we leave all missing values of x_i untouched.

To solve the considered classification problem, we do not use variables from X directly. Instead of this, we use attributes constructed on the basis of these variables. Let us consider some examples.

Let $x_i \in X$ be a discrete variable. Let us divide the set $V(x_i)$ into relatively small number of nonempty disjoint subsets $V_1, \ldots, V_s$. Then the value of the considered attribute on an object u is equal to the value $j \in \{1, \ldots, s\}$ for which $x_i(u) \in V_j$. The value of this attribute on u is missing if and only if the value of x_i on u is missing.

Let $x_i \in X$ be a continuous variable and $c \in \mathbb{R}$. Then the value of the considered attribute on an object u is equal to 0 if $x_i(u) < c$, and is equal to 1 otherwise. The value of this attribute on u is missing if and only if the value of x_i on u is missing.

Let $x_{i_1}, \ldots, x_{i_t} \in X$ be continuous variables and f be a function from $\mathbb{R}^t$ to $\mathbb{R}$. Then the value of the considered attribute on an object u is equal to 0 if $f(x_{i_1}(u), \ldots, x_{i_t}(u)) < 0$, and is equal to 1 otherwise. The value of this attribute on u is missing if and only if the value of at least one variable from $\{x_{i_1}, \ldots, x_{i_t}\}$ on u is missing.

We now assume that the attributes $f_1, \ldots, f_m$ are chosen. Let, for simplicity, $u_1, \ldots, u_n$ be all objects from $\{u_1, \ldots, u_N\}$ such that the value of the variable x_r on the considered object is definite (is not missing).

We now describe a decision table T. This table contains m columns labeled with attributes $f_1, \ldots, f_m$, and n rows corresponding to objects $u_1, \ldots, u_n$ respectively. For $j = 1, \ldots, n$, the j-th row is labeled with the value $x_r(u_j)$ which will be considered later as the value of the decision attribute d. For any $i \in \{1, \ldots, m\}$ and $j \in \{1, \ldots, n\}$, the value $f_i(u_j)$ is at the intersection of the j-th row and the i-th column. If the value $f_i(u_j)$ is missing, then the symbol " $-$ " is at the intersection of the j-th row and the i-th column.

5.2 Problem of Attribute Reduction

In this section, we define the problem of attribute reduction, consider some examples and discuss the notions of reduct and decision rule studied in this thesis.

5.2.1 Definition of Problem

Let T be a decision table with m columns labeled with attributes $f_1, \ldots, f_m$ and n rows which are identified with objects $u_1, \ldots, u_n$. It is possible that T contains missing values denoted by " $-$ ". Each row is labeled with a decision which is interpreted as the value of the decision attribute d. Let $A = \{f_1, \ldots, f_m\}$ and $U = \{u_1, \ldots, u_n\}$.

We now define the *indiscernibility relation* $IND(T) \subseteq U \times U$. Let $u_l, u_t \in U$. Then $(u_l, u_t) \in IND(T)$ if and only if $f_i(u_l) = f_i(u_t)$ for any $f_i \in A$ such that the values $f_i(u_l)$ and $f_i(u_t)$ are definite (are not missing). Since T can contain missing values, the relation $IND(T)$ is not an equivalence relation in the general case, but it is a tolerance relation.

By $DIS(T)$ we denote the set of unordered pairs of objects u_l and u_t from U such that $(u_l, u_t) \notin IND(T)$. Let $(u_l, u_t) \in DIS(T)$ and $f_i \in A$. We will say that the attribute f_i *separates* the pair (u_l, u_t) if the values $f_i(u_l)$ and $f_i(u_t)$ are definite and $f_i(u_l) \neq f_i(u_t)$. For any $f_i \in A$, we denote by $DIS(T, f_i)$ the set of pairs from $DIS(T)$ which the attribute f_i separates.

Let $\mathcal{P}$ be a subset of $DIS(T)$. Let Q be a subset of A and α be a real number such that $0 \leq \alpha < 1$. We will say that Q is an *α-test for T relative to $\mathcal{P}$* (an *$(\alpha, \mathcal{P})$-test for T*) if attributes from Q separate at least $(1 - \alpha)|\mathcal{P}|$ pairs from $\mathcal{P}$. An $(\alpha, \mathcal{P})$-test for T is called an *α-reduct for T relative to $\mathcal{P}$* (an *$(\alpha, \mathcal{P})$-reduct for T*) if each proper subset of this $(\alpha, \mathcal{P})$-test is not an $(\alpha, \mathcal{P})$-test for T. If $\mathcal{P} = \emptyset$, then any subset Q of A is an $(\alpha, \mathcal{P})$-test for T, but only the empty set of attributes is an $(\alpha, \mathcal{P})$-reduct for T. Note that each $(\alpha, \mathcal{P})$-test contains an $(\alpha, \mathcal{P})$-reduct as a subset. The parameter α can be interpreted as an inaccuracy. If $\alpha = 0$, then we obtain the notion of exact test for T relative to $\mathcal{P}$ and the notion of exact reduct for T relative to $\mathcal{P}$.

The *problem of attribute reduction* is the following: for a given decision table T, subset $\mathcal{P}$ of the set $DIS(T)$ and real α, $0 \leq \alpha < 1$, it is required to find an $(\alpha, \mathcal{P})$-reduct for T (an $(\alpha, \mathcal{P})$-test for T) with minimal cardinality. Let us denote by $R_{\min}(\alpha) = R_{\min}(\alpha, \mathcal{P}, T)$ the minimal cardinality of an $(\alpha, \mathcal{P})$-reduct for T. Of course, it is possible to use another measures of reduct quality.

The considered problem can be easily reformulated as a set cover problem: we should cover the set $\mathcal{P}$ using minimal number of subsets from the family $\{\mathcal{P} \cap DIS(T, f_1), \ldots, \mathcal{P} \cap DIS(T, f_m)\}$. Therefore, we can use results, obtained for the set cover problem, for analysis of the attribute reduction problem.

5.2.2 Examples

We now consider examples of sets $\mathcal{P}$ corresponding to different kinds of reducts. It was impossible for us to find definitions of some kinds of reducts which are applicable to decision tables with missing values. In such cases we have extended existing definitions (if it was possible) trying to preserve their spirit.

For an arbitrary $u_l \in U$, let $[u_l]_T = \{u_t : u_t \in U, (u_l, u_t) \in IND(T)\}$ and $\partial_T(u_l) = \{d(u_t) : u_t \in [u_l]_T\}$. The set $\partial_T(u_l)$ is called the *generalized decision* for u_l. The *positive region* $POS(T)$ for T is the set of objects $u_l \in U$ such that $|\partial_T(u_l)| = 1$. The set $BN(T) = U \setminus POS(T)$ is called the *boundary region* for T.

1. *Reducts for the information system, obtained from T by removing the decision attribute d.* The set $\mathcal{P}$ is equal to $DIS(T)$ (we should preserve the indiscernibility relation).
2. *Usual decision reducts for T.* The set $\mathcal{P}$ is equal to the set of all pairs $(u_l, u_t) \in DIS(T)$ such that $d(u_l) \neq d(u_t)$ and at least one object from the pair belongs to $POS(T)$ (we should preserve the positive region).
3. *Decision reducts for T based on the generalized decision.* Let us assume T is without missing values. The set $\mathcal{P}$ is equal to the set of all pairs $(u_l, u_t) \in DIS(T)$ such that $\partial_T(u_l) \neq \partial_T(u_t)$.
4. *Maximally discerning decision reducts for T.* The set $\mathcal{P}$ is equal to the set of all pairs $(u_l, u_t) \in DIS(T)$ such that $d(u_l) \neq d(u_t)$.
5. *Usual local reducts for T and object $u_l \in POS(T)$.* The set $\mathcal{P}$ is equal to the set of all pairs $(u_l, u_t) \in DIS(T)$ such that $d(u_l) \neq d(u_t)$.
6. *Local reducts for T and object $u_l \in U$ based on the generalized decision.* Let us assume T is without missing values. The set $\mathcal{P}$ is equal to the set of all pairs $(u_l, u_t) \in DIS(T)$ such that $\partial_T(u_l) \neq \partial_T(u_t)$.
7. *Maximally discerning local reducts for T and object $u_l \in U$.* The set $\mathcal{P}$ is equal to the set of all pairs $(u_l, u_t) \in DIS(T)$ such that $d(u_l) \neq d(u_t)$.

5.2.3 Maximally Discerning Local Reducts

The notion of decision rule considered in this thesis is closest to the notion of maximally discerning local reduct. The consideration of maximally discerning local reducts for objects from the boundary region can lead to construction of a decision rule system which is applicable to wider class of new objects. We now consider an example.

T =

f_1	f_2	
0	0	1
0	0	2
0	1	2
1	0	1

S_1
$f_2 = 1 \rightarrow 2$
$f_1 = 1 \rightarrow 1$

S_2
$f_1 = 0 \wedge f_2 = 0 \rightarrow \{1, 2\}$
$f_2 = 1 \rightarrow \{2\}$
$f_1 = 1 \rightarrow \{1\}$

S_3
$f_2 = 0 \rightarrow 1$
$f_1 = 0 \rightarrow 2$
$f_2 = 1 \rightarrow 2$
$f_1 = 1 \rightarrow 1$

Fig. 5.1. Illustrations to Example 5.1

Example 5.1. Let us consider the decision table T and three systems of decision rules S_1, S_2 and S_3 obtained on the basis of usual local reducts, local reducts based on the generalized decision, and maximally discerning local reducts (see Fig. 5.1). Let us consider two new objects $(0, 2)$ and $(2, 0)$. Systems S_1 and S_2 have no rules which are realizable on the new objects. However, the system S_3 has rules which are realizable on these new objects and, moreover, correspond to these objects different decisions.

5.3 Greedy Algorithm

We now describe the greedy algorithm which for a given α, $0 \leq \alpha < 1$, decision table T and set of pairs $\mathcal{P} \subseteq DIS(T)$, $\mathcal{P} \neq \emptyset$, constructs an $(\alpha, \mathcal{P})$-test for T.

Algorithm 5: Greedy algorithm for partial test construction

```
Input  : Decision table $T$ with conditional attributes $f_1, \ldots, f_m$, set of pairs $\mathcal{P} \subseteq DIS(T)$, $\mathcal{P} \neq \emptyset$,
         and real number $\alpha$, $0 \leq \alpha < 1$.
Output: $(\alpha, \mathcal{P})$-test for $T$.
$Q \longleftarrow \emptyset$;
while $Q$ is not an $(\alpha, \mathcal{P})$-test for $T$ do
    select $f_i \in \{f_1, \ldots, f_m\}$ with minimal index $i$ such that $f_i$ separates the maximal number of
    pairs from $\mathcal{P}$ unseparated by attributes from $Q$;
    $Q \longleftarrow Q \cup \{f_i\}$;
end
return $Q$;
```

By $R_{\text{greedy}}(\alpha) = R_{\text{greedy}}(\alpha, \mathcal{P}, T)$ we denote the cardinality of the constructed $(\alpha, \mathcal{P})$-test for T.

5.3.1 Precision of Greedy Algorithm

Using Theorems 1.8–1.10 one can prove the following three theorems.

Theorem 5.2. *Let $0 \leq \alpha < 1$ and $\lceil (1-\alpha)|\mathcal{P}| \rceil \geq 2$. Then $R_{\text{greedy}}(\alpha) < R_{\min}(\alpha) \times (\ln \lceil (1-\alpha)|\mathcal{P}| \rceil - \ln \ln \lceil (1-\alpha)|\mathcal{P}| \rceil + 0.78)$.*

Theorem 5.3. *Let $0 \leq \alpha < 1$. Then for any natural $t \geq 2$ there exists a decision table T and a subset $\mathcal{P}$ of the set $DIS(T)$ such that $\lceil (1-\alpha)|\mathcal{P}| \rceil = t$ and $R_{\text{greedy}}(\alpha) > R_{\min}(\alpha)(\ln \lceil (1-\alpha)|\mathcal{P}| \rceil - \ln \ln \lceil (1-\alpha)|\mathcal{P}| \rceil - 0.31)$.*

Theorem 5.4. *Let $0 \leq \alpha < 1$. Then*

$$R_{\text{greedy}}(\alpha) \leq R_{\min}(\alpha) \left(1 + \ln \left(\max_{j \in \{1, \ldots, m\}} |\mathcal{P} \cap DIS(T, f_j)| \right) \right) .$$

5.3.2 Polynomial Approximate Algorithms

From results obtained in [43, 63] the next theorem follows.

Theorem 5.5. *Let* $0 \leq \alpha < 1$. *Then the problem of construction, for given* T *and* $\mathcal{P} \subseteq DIS(T)$, *an* $(\alpha, \mathcal{P})$*-reduct for* T *with minimal cardinality is* NP*-hard.*

From statements obtained in [38, 40] (based on results from [11, 53, 61, 63]) the next two theorems follow.

Theorem 5.6. *Let* $\alpha \in \mathbb{R}$ *and* $0 \leq \alpha < 1$. *If* $NP \not\subseteq DTIME(n^{O(\log \log n)})$, *then for any* ε, $0 < \varepsilon < 1$, *there is no polynomial algorithm that, for a given decision table* T *with* $DIS(T) \neq \emptyset$ *and nonempty subset* $\mathcal{P} \subseteq DIS(T)$, *constructs an* $(\alpha, \mathcal{P})$*-test for* T *which cardinality is at most* $(1 - \varepsilon) R_{\min}(\alpha, \mathcal{P}, T) \ln |\mathcal{P}|$.

From Theorem 5.4 it follows that $R_{\mathrm{greedy}}(\alpha) \leq R_{\min}(\alpha)(1 + \ln |\mathcal{P}|)$. From this inequality and from Theorem 5.6 it follows that, under the assumption $NP \not\subseteq DTIME(n^{O(\log \log n)})$, the greedy algorithm is close to the best polynomial approximate algorithms for partial test cardinality minimization.

Theorem 5.7. *Let* α *be a real number such that* $0 \leq \alpha < 1$. *If* $P \neq NP$, *then there exists* $\varrho > 0$ *such that there is no polynomial algorithm that, for a given decision table* T *with* $DIS(T) \neq \emptyset$ *and nonempty subset* $\mathcal{P} \subseteq DIS(T)$, *constructs an* $(\alpha, \mathcal{P})$*-test for* T *which cardinality is at most* $\varrho R_{\min}(\alpha, \mathcal{P}, T) \ln |\mathcal{P}|$.

From Theorems 5.4 and 5.7 it follows that, under the assumption $P \neq NP$, the greedy algorithm is not far from the best polynomial approximate algorithms for partial test cardinality minimization.

5.3.3 Lower Bound on $R_{\min}(\alpha)$

In this subsection, we fix some information about the greedy algorithm work and find a lower bound on $R_{\min}(\alpha)$ depending on this information.

Let us apply the greedy algorithm to α, T and $\mathcal{P}$. Let during the construction of $(\alpha, \mathcal{P})$-test for T the greedy algorithm choose consequently attributes $f_{j_1}, \ldots, f_{j_t}$. Let us denote by δ_1 the number of pairs from $\mathcal{P}$ separated by the attribute f_{j_1}. For $i = 2, \ldots, t$, we denote by δ_i the number of pairs from $\mathcal{P}$ which are not separated by attributes $f_{j_1}, \ldots, f_{j_{i-1}}$, but are separated by the attribute f_{j_i}. Let $\Delta(\alpha, \mathcal{P}, T) = (\delta_1, \ldots, \delta_t)$. As information on the greedy algorithm work we will use the tuple $\Delta(\alpha, \mathcal{P}, T)$ and numbers $|\mathcal{P}|$ and α.

We now define the parameter $l(\alpha) = l(\alpha, |\mathcal{P}|, \Delta(\alpha, \mathcal{P}, T))$. Let $\delta_0 = 0$. Then

$$l(\alpha) = \max\left\{\left\lceil \frac{\lceil (1-\alpha)|\mathcal{P}|\rceil - (\delta_0 + \ldots + \delta_i)}{\delta_{i+1}} \right\rceil : i = 0, \ldots, t-1\right\} .$$

Using Theorems 1.22 and 1.25 one can prove the following two theorems.

Theorem 5.8. *Let T be a decision table, $\mathcal{P} \subseteq DIS(T)$, $\mathcal{P} \neq \emptyset$, and α be a real number such that $0 \leq \alpha < 1$. Then $R_{\min}(\alpha, \mathcal{P}, T) \geq l(\alpha, |\mathcal{P}|, \Delta(\alpha, \mathcal{P}, T))$.*

The value $l(\alpha) = l(\alpha, |\mathcal{P}|, \Delta(\alpha, \mathcal{P}, T))$ can be used for the obtaining of upper bounds on cardinality of partial tests constructed by the greedy algorithm.

Theorem 5.9. *Let α and β be real numbers such that $0 < \beta \leq \alpha < 1$. Then*

$$R_{\mathrm{greedy}}(\alpha) < l(\alpha - \beta) \ln\left(\frac{1 - \alpha + \beta}{\beta}\right) + 1 .$$

From Theorem 5.9 it follows that the lower bound $R_{\min}(\alpha) \geq l(\alpha)$ is nontrivial. In Chap. 1, it is shown that for decision rules (maximally discerning local reducts) the bound $R_{\min}(\alpha) \geq l(\alpha)$ is the best lower bound on $R_{\min}(\alpha)$ depending on $\Delta(\alpha, \mathcal{P}, T)$, $|\mathcal{P}|$ and α (see Theorem 1.46).

5.3.4 Upper Bound on $R_{\mathrm{greedy}}(\alpha)$

In this subsection, we obtain an upper bound on $R_{\mathrm{greedy}}(\alpha) = R_{\mathrm{greedy}}(\alpha, \mathcal{P}, T)$ which does not depend on $|\mathcal{P}|$. The next statement follows immediately from Theorems 5.8 and 5.9.

Theorem 5.10. *Let α and β be real numbers such that $0 < \beta \leq \alpha < 1$. Then*

$$R_{\mathrm{greedy}}(\alpha) < R_{\min}(\alpha - \beta) \ln\left(\frac{1 - \alpha + \beta}{\beta}\right) + 1 .$$

In Chap. 1, it is shown that for decision rules (maximally discerning local reducts) this bound is, in some sense, unimprovable: it is impossible to multiply the right-hand side of the considered inequality by any real δ such that $\delta < 1$ (see Theorem 1.52).

5.4 Conclusions

The chapter is devoted to discussion of universal problem of attribute reduction and to analysis of greedy algorithm for this problem solving. The obtained results show that, under some natural assumptions on the class NP, greedy algorithm is close to the best polynomial approximate algorithms for the minimization of partial test cardinality. Based on an information received during greedy algorithm work it is possible to obtain nontrivial lower bound on the minimal cardinality of partial reducts.

A part of obtained results (Theorems 5.2, 5.4, 5.8, 5.9, and 5.10) is true for any special kind of reduct that can be represented as a $(0, \mathcal{P})$-reduct for appropriate $\mathcal{P}$, in particular, for usual decision and local reducts.

Another part of results (Theorems 5.3, 5.5, 5.6, and 5.7) is proved only for the whole universal attribute reduction problem and for maximally discerning decision and local reducts. To obtain, for an another special kind of reducts, results similar to Theorems 5.3, 5.5, 5.6, and 5.7 we should make additional investigations.

Final Remarks

In this thesis, we study partial decision rules for the case, where the weight of each conditional attribute of a decision table is equal to 1, and for the case, where conditional attributes can have arbitrary natural weights.

In both cases, under some natural assumptions on the class NP, greedy algorithms for partial decision rule construction are close to the best (from the point of view of accuracy) polynomial approximate algorithms for minimization of complexity of partial rules.

We consider the accuracy of algorithms in the worst case. It means that we can try to find algorithms which will work better than greedy algorithms for some part of problems. We make such attempts. The results of experiments with new polynomial approximate algorithms, which are modifications of greedy algorithms, seem to be promising.

We find new nontrivial lower bounds on the minimal complexity of partial decision rules based on an information obtained during the work of greedy algorithms. Experimental results show that these bounds can be used in practice.

One of the main aims of the thesis is to evaluate possibilities of the use of partial decision rules for the improvement of accuracy of classifiers, and for more compact representation of knowledge.

Results of experiments with decision tables from UCI Repository of Machine Learning Databases show that the accuracy of classifiers based on partial decision rules is often better than the accuracy of classifiers based on exact decision rules.

Experimental and some theoretical results confirm the following 0.5-hypothesis: in the most part of cases, greedy algorithm during each step chooses an attribute which separates at least one-half of unseparated rows that should be separated. It means that greedy algorithm constructs often short partial rules with relatively high accuracy.

We design also new algorithm for construction of the set of all irreducible partial decision rules for almost all decision tables of a special kind. The considered algorithm has too high time complexity to be used in practice. However, this algorithm has essentially lesser complexity than the brute-force algorithms.

The obtained results will further to wider use of partial decision rules in rough set theory and related theories such as test theory and logical analysis of data (LAD).

The most part of results of the thesis is based on the study of set cover problem. We formulate an “universal attribute reduction problem”, and show how the results obtained for the set cover problem can be used for the study of another kinds of reducts such as reducts for information systems, local and decision reducts based on the generalized decision, etc.

Bibliography

1. Boros E., Hammer P.L., Ibarki T., Kogan A., Mayoraz E., Muchnik I. (2000) An implementation of logical analysis of data. IEEE Transactions of Knowledge and Data Engineering 12, 292–306
2. Bazan G.J., Nguyen S.H., Nguyen T.T., Skowron A., Stepaniuk J. (1998) Decision rules synthesis for object classification. In: E. Orłowska (ed.), Incomplete information: rough set analysis, Physica-Verlag, Heidelberg, 23–57
3. Bronsztejn N.I., Siemiendiajew A.K. (1997) Mathematics–the encyclopaedic handbook. PWN, Warsaw (in Polish)
4. Cichosz P. (2000) Learning systems. WNT, Warsaw (in Polish)
5. Cormen T.H., Leiserson Ch.E , Rivest R.L., Stein C. (2005) Introduction to algorithms. WNT, Warsaw (in Polish)
6. Chegis I.A., Yablonskii S.V. (1958) Logical methods of electric circuit control. Trudy Matematicheskogo Instituta im. V.A. Steklova 51, 270–360 (in Russian)
7. Cheriyan J., Ravi R. (1998) Lecture notes on approximation algorithms for network problems. http://www.math.uwaterloo.ca/∼jcheriya/lecnotes.html
8. Chvátal V. (1979) A greedy heuristic for the set-covering problem. Mathematics of Operations Research 4, 233–235
9. Crama Y., Hammer P.L., Ibaraki T. (1988) Cause–effect relationships and partially defined Boolean functions. Ann. Oper. Res. 16, 299–326
10. Dmitriev A.N., Zhuravlev Yu.I., Krendelev F.P. (1966) On mathematical principles for classification of objects and phenomena. In: Zhuravlev Yu.I., Makarov S.V. (eds.) Diskretnyi Analiz 7. Akad. Nauk SSSR Sib. Otd. Inst. Mat., Novosibirsk, 3–15 (in Russian)
11. Feige U. (1996) A threshold of $\ln n$ for approximating set cover (preliminary version). In: Proc. 28th Annual ACM symposium on the theory of computing (Philadelphia, Pennsylvania, USA). ACM, 314–318
12. Gavrilov G.P., Sapozhenko A.A. (2004) Problems and exercises in discrete mathematics (third edition). Fizmatlit, Moscow (in Russian)

13. Grzymała-Busse J.W. (2004) Data with missing attribute values: generalization of indiscernibility relation and rule induction. In: Peters J.F., Skowron A., Grzymała-Busse J.W., Kostek B., Świniarski R.W., Szczuka M.S. (eds.) LNCS Transactions on Rough Sets I. LNCS 3100. Springer, Berlin Heidelberg New York, 78–95

14. Hand M., Mannila H., Smyth P. (2005) Principles of Data Mining. WNT, Warsaw (in Polish).

15. Jensen R., Shen Q. (2004) Semantics-preserving dimensionality reduction: rough and fuzzy-rough-based approaches. IEEE Transactions on Knowledge and Data Engineering 16, 1457–1471

16. Johnson D.S. (1974) Approximation algorithms for combinatorial problems. J. Comput. System Sci. 9, 256–278

17. Karp R.M. (1972) Reducibility among combinatorial problems. In: Miller R.E., Thatcher J.W. (eds.) Complexity of computer computations. Plenum Press, New York, 85–103

18. Kearns M.J. (1990) The computational complexity of machine learning. MIT Press, Cambridge Massachussetts

19. Komorowski J., Pawlak Z., Polkowski L., Skowron A. (1999), Rough sets: A tutorial. In S.K. Pal S.K. and Skowron A. (eds.) Rough-Fuzzy Hybridization: A New Trend in Decision-Making, Springer-Verlag, Singapor, 3–98.

20. Koronacki J., Ćwik J. (2005) Statistical learning systems. WNT, Warsaw (in Polish)

21. Kryszkiewicz M. (1999) Rules in incomplete information systems. Information Sciences 113, 271–292

22. Kuzjurin N.N. (1980) Asymptotical investigation of set covering problem. In: Yablonskii S.V. (ed.) Problemi Kibernetiki 37. Nauka Publishers, Moscow, 19–57 (in Russian)

23. Liu B., Abbass H.A., McKay B. (2003) Classification rule discovery with ant colony optimization. In: Proc. Intelligent agent technology. IEEE/WIC, 83–88

24. Lovász L. (1975) On the ratio of optimal integral and fractional covers. Discrete Math. 13, 383–390

25. Michalski S.R., Pietrzykowski J. (2007) iAQ: A program that discovers rules. In: Proc. 22nd Conference on artificial intelligence (Vancouver, Canada)

26. Moshkov M.Ju. (2003) Greedy algorithm for set cover in context of knowledge discovery problems. Electronic Notes in Theoretical Computer Science 82(4), 174–185

27. Moshkov M.Ju. (2003) On greedy algorithm for partial cover construction. In: Lupanov O.B. (ed.) Proc. Design and complexity of control systems (Nizhny Novgorod, Russia). Moscow University, Moscow, 57 (in Russian)

28. Moshkov M.Ju. (2005) On construction of the set of irreducible partial covers. In: Lupanov O.B., Kasim-Zade O.M., Chashkin A.V., Steinhöfel K. (eds.) Proc. Stochastic algorithms: foundations and applications. LNCS 3777. Springer, Berlin Heidelberg New York, 38–44
29. Moshkov M.Ju. (2005) On the set of partial reducts for the most part of binary decision tables. In: Czaja L. (ed.) Proc. Concurrency specification and programming 2 (Ruciane-Nida, Poland). Warsaw University, Warsaw, 333–341
30. Moshkov M.Ju. (2007) On algorithms for construction of all irreducible partial covers. Information Processing Letters 103, 66–70
31. Moshkov M.Ju., Piliszczuk M., Zielosko B. (2005) Greedy algorithm for construction of partial covers. In: Lupanov O.B. (ed.) Proc. Problems of theoretical cybernetics (Penza, Russia). Moscow University, Moscow, 103 (in Russian)
32. Moshkov M.Ju., Piliszczuk M., Zielosko B. (2006) Lower bounds on minimal weight of partial reducts and partial decision rules. In: Wang G., Peters J.F., Skowron A., Yao Y. (eds.) Proc. Rough sets and knowledge technology. LNAI 4062, Springer, Berlin Heidelberg New York, 290–296
33. Moshkov M.Ju., Piliszczuk M., Zielosko B. (2006) Lower bound on minimal weight of partial cover based on information about greedy algorithm work. In: Proc. Information processing and management of uncertainty in knowledge-based systems (Paris, France)
34. Moshkov M.Ju., Piliszczuk M., Zielosko B. (2006) On greedy algorithm with weights for construction of partial covers. In: Kłopotek M.A., Wierzchoń S.T., Trojanowski K. (eds.) Proc. Intelligent information processing and web mining. Advances in soft computing. Springer, Berlin Heidelberg New York, 391–395
35. Moshkov M.Ju., Piliszczuk M., Zielosko B. (2007) On partial covers, reducts and decision rules with weights. In: Peters J.F., Skowron A., Düntsch I., Grzymała-Busse J.W., Orłowska E,. Polkowski L. (eds.) LNCS Transactions on Rough Sets VI. LNCS 4374. Springer, Berlin Heidelberg New York, 211–246
36. Moshkov M.Ju., Piliszczuk M., Zielosko B. (2007) Universal attribute reduction problem. In: Kryszkiewicz M., Peters J.F., Rybinski H., Skowron A. (eds.) Proc. Rough sets and intelligent systems paradigms. LNCS 4585. Springer, Berlin Heidelberg New York, 417–426
37. Moshkov M.Ju., Piliszczuk M., Zielosko B. (2007) On construction of partial reducts and irreducible partial decision rules. Fundamenta Informaticae 75, 357–374
38. Moshkov M.Ju., Piliszczuk M., Zielosko B. (2008) On partial covers, reducts and decision rules. In: LNCS Transactions on Rough Sets VIII. LNCS 5084. Springer, Berlin Heidelberg New York, 258–296

39. Moshkov M.Ju., Piliszczuk M., Zielosko B. Universal problem of attribute reduction. In: LNCS Transactions on Rough Sets. Springer, Berlin Heidelberg New York (submitted)

40. Moshkov, M.J., Piliszczuk, M., Zielosko, B.: Partial covers, reducts and decision rules in rough sets: theory and applications. Studies in Computational Intelligence, vol. 145. Springer, Heidelberg (to appear, 2008)

41. Newman D.J., Hettich S., Blake C.L., Merz C.J. (1998) UCI repository of machine learning databases. University of California, Irvine, Department of information and computer sciences, http://www.ics.uci.edu/∼mlearn/MLRepository.html

42. Nguyen H.S. (2006) Approximate Boolean reasoning: foundations and applications in data mining. In: Peters J.F., Skowron A. (eds.) LNCS Transactions on Rough Sets V. LNCS 4100. Springer, Berlin Heidelberg New York, 344–523

43. Nguyen H.S., Ślęzak D. (1999) Approximate reducts and association rules – correspondence and complexity results. In: Zhong N., Skowron A., Ohsuga S. (eds.) Proc. Rough sets, fuzzy sets, data mining, and granular-soft computing. LNAI 1711. Springer, Berlin Heidelberg New York, 137–145

44. Nigmatullin R.G. (1969) The fastest descent method for covering problems. In: Proc. Questions of precision and efficiency of computer algorithms 5 (Kiev, USSR), 116–126 (in Russian)

45. Pawlak Z. (1991) Rough sets – theoretical aspects of reasoning about data. Kluwer Academic Publishers, Dordrecht Boston London

46. Pawlak Z. (1998) Rough set elements. In: Polkowski L., Skowron A. (eds.) Rough sets in knowledge discovery. Physica-Verlag, Heidelberg, 10–30

47. Pawlak Z., Skowron A. (2007) Rudiments of rough sets. Information Sciences 177, 3–27

48. Pawlak Z., Skowron A. (2007) Rough sets: some extensions. Information Sciences 177, 28–40

49. Pawlak Z., Skowron A. (2007) Rough sets and Boolean reasoning. Information Sciences 177, 41–73

50. Piliszczuk M., Zielosko B. (2007) Greedy algorithm for partial reducts and partial decision rules construction. In: Wakulicz-Deja A. (ed.) Proc. Decision support systems 2 (Zakopane, Poland). University of Silesia, Katowice, 130-137, (in Polish)

51. Quafafou M. (2000) α-RST: a generalization of rough set theory. Information Sciences 124, 301–316

52. Quinlan J.R. (1993) C4.5: Programs for machine learning. Morgan Kaufmann Publishers, San Mateo

53. Raz R., Safra S. (1997) A sub-constant error-probability low-degree test, and a sub-constant error-probability PCP characterization of NP. In: Proc. 29th Annual

ACM symposium on the theory of computing (El Paso, Texas, USA). ACM, 475–484

54. Rough Set Exploration System (RSES), http://logic.mimuw.edu.pl/∼rses
55. Skowron A. (2000) Rough sets in KDD. In: Shi Z., Faltings B., Musen M. (eds.) Proc. 16th IFIP World computer congress (Beijing, China). Publishing House of Electronic Industry, 1–14
56. Skowron A., Rauszer C. (1992) The discernibility matrices and functions in information systems. In: Slowinski R. (ed.) Intelligent decision support. Handbook of applications and advances of the rough set theory. Kluwer Academic Publishers, Dordrecht Boston London, 331–362
57. Slavík P. (1996) A tight analysis of the greedy algorithm for set cover (extended abstract). In: Proc. 28th Annual ACM symposium on the theory of computing (Philadelphia, Pennsylvania, USA). ACM, 435–441
58. Slavík P. (1998) Approximation algorithms for set cover and related problems. Ph.D. Thesis, University of New York at Buffalo
59. Stefanowski J., Krawiec K. (2004) Machine learning and neural networks. WPP, Poznań (in Polish)
60. Ślęzak D. (1996) Approximate reducts in decision tables. In: Proc. Information processing and management of uncertainty in knowledge-based systems 3 (Granada, Spain), 1159–1164
61. Ślęzak D. (2000) Normalized decision functions and measures for inconsistent decision tables analysis. Fundamenta Informaticae 44, 291–319
62. Ślęzak D. (2001) Approximate decision reducts. Ph.D. Thesis, Warsaw University, Warsaw (in Polish)
63. Ślęzak D. (2002) Approximate entropy reducts. Fundamenta Informaticae 53, 365–390
64. Ślęzak D., Wróblewski J. (2003) Order-based genetic algorithms for the search of approximate entropy reducts. In: Wang G., Liu Q., Yao Y., Skowron A. (eds.) Proc. Rough sets, fuzzy sets, data mining, and granular computing. LNAI 2639. Springer, Berlin Heidelberg New York, 308–311
65. Soloviev N.A. (1978) Tests (theory, construction, applications). Nauka Publishers, Novosibirsk (in Russian)
66. Vercellis C. (1984) A probabilistic analysis of the set covering problem. Ann. Oper. Res. 1, 255–271
67. Vazirani V.V. (2005) Approximation algorithms. WNT, Warsaw (in Polish)
68. Wakulicz-Deja A. (1990) Foundations of information retrieval systems. PWN, Warsaw-Łódź (in Polish)
69. Wakulicz-Deja A., Nowak A. (2007) From an information system to a decision support system. In: Kryszkiewicz M., Peters J.F., Rybinski H., Skowron A. (eds.)

Proc. Rough sets and intelligent systems paradigms. LNCS 4585. Springer, Berlin Heidelberg New York, 454–464.

70. Wakulicz-Deja A., Paszek P. (2003) Applying Rough Set Theory to Multi Stage Medical Diagnosing. Fundamenta Informaticae 54, 387-408
71. Wróblewski J. (2001) Ensembles of classifiers based on approximate reducts. Fundamenta Informaticae 47, 351–360
72. Wróblewski J. (1995) Finding minimal reducts using genetic algorithm. In: Proc. 2nd Annual Joint Conference on Information Sciences, (Warsaw, Poland). 186–189
73. Yablonskii S.V. (1975) Tests. In: Glushkov V.M. (ed.) Encyclopedia of cybernetics. Main Editorial Board of Ukrainian Soviet Encyclopedia, Kiev, 431–432 (in Russian)
74. Yablonskii S.V. (1988) Some problems of reliability and diagnosis in control systems. In: Yablonskii S.V. (ed.) Matematicheskie problemi kibernetiki 1. Nauka Publishers, Moscow, 5–25 (in Russian)
75. Yablonskii S.V. (2003) Introduction into discrete mathematics (fourth edition). Vishaya Shkola, Moscow (in Russian)
76. Zhuravlev Yu.I. (1958) The separability of subsets of vertices of the n-dimensional unit cube. Trudy Matematicheskogo Instituta im. V.A. Steklova 51:143–157 (in Russian)
77. Zhuravlev Yu.I. (1963) Inessential variables of partial Boolean functions. In: Vasil'ev Yu.L., Zhuravlev Yu.I., Korobkov V.K., Krichevskii R.E. (eds.) Diskretnyi Analiz 1. Akad. Nauk SSSR Sib. Otd. Inst. Mat., Novosibirsk, 28–31 (in Russian)
78. Ziarko W. (1993) Analysis of uncertain information in the framework of variable precision rough sets. Foundations of Computing and Decision Sciences 18, 381–396
79. Zielosko B. (2005) On partial decision rules. In: Czaja L. (ed.) Proc. Concurrency, specification and programming 2 (Ruciane-Nida, Poland). Warsaw University, Warsaw, 598–609
80. Zielosko B. (2006) Greedy algorithm for partial partial decision rules construction. In: Wakulicz-Deja A. (ed.) Proc. Decision support systems (Zakopane, Poland). University of Silesia, Katowice, 112-119, (in Polish)
81. Zielosko B., Piliszczuk M. (2008) Greedy algorithm for attribute reduction. Fundamenta Informaticae 85, 549–561
82. Zielosko B., Piliszczuk M., Kocjan A. (2008) Classifiers based on partial reducts and partial decision rules. In: Kłopotek M.A., Przepiórkowski A., Wierzchoń S.T. (eds.) Proc. Intelligent information systems, Zakopane, Poland, 431–438

List of Figures

List of Tables

List of Algorithms

Uniwersytet Śląski
Wydział Informatyki i Nauki o Materiałach
Instytut Informatyki

mgr Beata Zielosko

Streszczenie rozprawy doktorskiej

Generowanie i optymalizacja częściowych reguł decyzyjnych

Promotor:
prof. dr hab. Mikhail Ju. Moshkov

Sosnowiec, 2008

Uwagi

W niniejszym opracowaniu przedstawiono streszczenie pracy doktorskiej pt. „Construction and Optimization of Partial Decision Rules”, która została napisana w języku angielskim. W streszczeniu zostały pominięte twierdzenia wraz z dowodami. Przedstawiono jedynie najważniejsze rezultaty rozprawy oraz część bibliografii.

Streszczenie

Zagadnienia poruszane w rozprawie doktorskiej związane są z problemami występującymi w dziedzinie odkrywania i wydobywania wiedzy z danych (ang. data mining and knowledge discovery). Pierwszy problem, to reprezentacja wiedzy zawartej w tablicy decyzyjnej w formie dogodnej dla zrozumienia. W tej sytuacji długość opisu wiedzy odgrywa istotną rolę. Drugi problem, to przewidywanie wartości atrybutu decyzyjnego dla nowych obiektów. Wówczas dokładność przewidywania (klasyfikacji) ma duże znaczenie, np. w medycynie.

Te dwa cele (zwięzły opis i wysoka dokładność) wydają się być sprzeczne ze sobą, aczkolwiek wiadomo jest, że klasyfikatory ze zwięzłym opisem wiedzy są często bardziej dokładne. Przedstawiona rozprawa doktorska stanowi potwierdzenie tego faktu.

Praca poświęcona jest rozważaniom dotyczącym jednego z głównych pojęć teorii zbiorów przybliżonych: pojęciu reguły decyzyjnej (lokalnego reduktu) [16, 18, 20, 23, 24, 31, 32]. Definicja rozważanej w pracy częściowej reguły decyzyjnej została przedstawiona poniżej.

Niech T będzie tablicą z n wierszami oznaczonymi przez nieujemne liczby całkowite (decyzje) i m kolumnami oznaczonymi jako atrybuty warunkowe $f_1, \ldots, f_m$. Tablica ta wypełniona jest przez nieujemne liczby całkowite (wartości atrybutów). Tablica T jest nazywana tablicą decyzyjną. Powiemy, że atrybut f_i separuje (oddziela) wiersze r_1 i r_2 tablicy T, jeśli wiersze te posiadają różne wartości na przecieciu z kolumną f_i. Wiersze te nazywane są różnymi, jeśli przynjamniej jeden atrybut f_i je separuje.

Niech $r = (b_1, \ldots, b_m)$ będzie wierszem tablicy T oznaczonym przez decyzję d. Przez $U(T, r)$ oznaczamy zbiór wierszy z T, które są różne od r i są oznaczone przez decyzje inne niż d. Niech $\alpha \in \mathbb{R}$ i $0 \leq \alpha < 1$. Reguła decyzjna

$$(f_{i_1} = b_{i_1}) \wedge \ldots \wedge (f_{i_t} = b_{i_t}) \rightarrow d$$

jest nazywana *α-regułą decyzyjną* dla T i r jeśli atrybuty $f_{i_1}, \ldots, f_{i_t}$ separują od r przynjamniej $(1-\alpha)|U(T,r)|$ wierszy ze zbioru $U(T,r)$ (taka reguła nazywana jest także *częściową* regułą decyzyjną).

Dokładne reguły decyzyjne są szeroko stosowane w teorii zbiorów przybliżonych, zarówno dla konstruowania klasyfikatorów oraz jako sposób reprezentacji wiedzy [23]. Istnienie reguł decyzyjnych z małą liczbą atrybutów może ułatwić zrozumienie relacji pomiedzy atrybutami warunkowymi a decyzją. Należy także zauważyć, że pojecia podobne do pojęcia reguły decyzyjnej były badane w teorii testów (gdzie pojęcie testu kontrolnego nie jest dalekie od pojęcia reguły decyzyjnej) oraz w logicznej analizie danych LAD (ang. Logical Analysis of Data) (gdzie wzorzec jest analogią do reguły decyzyjnej).

Główna teza rozprawy doktorskiej jest następująca: zastosowanie algorytmów zachłannych do generowania reguł decyzyjnych pozwala uzyskać krótkie częściowe reguły decyzyjne o odpowiednio wysokiej jakości.

Dokładne reguły decyzyjne mogą być przeuczone tzn. zbyt mocno zależne od szumu lub (w przypadku klasyfikacji) zbyt mocno dopasowane do istniejących przypadków (obiektów). Jeśli reguły decyzyjne są traktowane jako sposób reprezentacji wiedzy wówczas, zamiast dokładnych reguł z wieloma atrybutami, nie gorsze wyniki można uzyskać stosując częściowe reguły decyzyjne z mniejszą liczbą atrybutów, które oddzielają od danego wiersza prawie wszystkie inne wiersze z inną decyzją.

Rozważana idea nie jest nowa. Od lat w teorii zbiorów przybliżonych częściowe redukty i częściowe reguły decyzyjne (częściowe redukty lokalne) są intensywnie badane przez H.S. Nguyena, A. Skowrona, D. Ślęzaka, Z. Pawlaka, J. Wróblewskiego i innych [1, 15, 17, 18, 33].

Istnieją różne podejścia do definiowania przybliżonych reduktów. W [15, 28] zostało udowodnione, że dla każdego z rozważanych podejść, problem minimalizacji częściowych reduktów (konstruowania częściowych reduktów o minimalnej liczności) jest NP-trudny. Podejście przedstawione w [15] jest podobne do podejścia rozważanego w tej rozprawie. Szczegółowa dyskusja dotycząca częściowych reguł decyzyjnych została zawarta w rozdziale 5 rozprawy. Przybliżone redukty były także badane przez W. Ziarko, M. Quafafou i innych, w rozszerzonym modelu zbiorów przybliżonych VPRS (ang. Variable Precision Rough Sets) i α-RST (ang. alpha Rough Set Theory).

Istnieją różne miary jakości reguł decyzyjnych: długość reguły, całkowita waga atrybutów zawartych w regule decyzyjnej, wsparcie reguły decyzyjnej i inne. W pracy koncentrujemy się na minimalizacji długości reguł (co pozwala konstruować klasyfikatory o większej dokładności lub uzyskać bardziej zwięzłą reprezentację wiedzy zawartej w tablicach decyzyjnych) oraz na minimalizacji całkowitej wagi reguł (co pozwala zmniejszyć złożoność czasową, koszt lub ryzyko pracy klasyfikatorów).

Istnieją różne podejścia do konstruowania reguł decyzyjnych: brute-force algorytm, który jest stosowany do tablic decyzyjnych o stosunkowo małej liczbie atrybutów, algorytmy genetyczne, symulowane wyżarzanie, wnioskowanie Boolowskie, algorytmy optymalizacji mrowiskowej, algorytmy oparte na tworzeniu drzew decyzyjnych, algorytmy zachłanne.

Każda z tych metod posiada liczne modyfikacje. Np. w przypadku drzew decyzyjnych można stosować algorytmy zachłanne oparte na entropii, indeksie Gini, dla konstruowania reguł decyzyjnych.

W przedstawianej rozprawie doktorskiej stosujemy algorytmy zachłanne do konstruowania reguł. Oczywiście, algorytmy te nie są nowe i były używane przez licznych autorów [3]. Nasz wybór związany jest z matematycznymi wynikami badań uzyskanymi dla algorytmów zachłannych. Zostało udowodnione, że biorąc pod uwagę pewne założenia dotyczące klasy NP, algorytmy zachłanne pozwalają uzyskać wyniki bliskie wynikom uzyskiwanym przez najlepsze przybliżone wielomianowe algorytmy dla optymalizacji reguł decyzyjnych.

Ważną częścią rozprawy doktorskiej jest matematyczna analiza problemu konstruowania częściowych reguł decyzyjnych, której wyniki są blisko związane z wynikami eksperyemntów. W wielu przypadkach, wyniki eksperymentów prowadziły do istotnych, nowych stwierdzeń a matematyczna analiza pozwoliła wybrać nowe kierunki badań.

Badania dotyczące częściowych reguł decyzyjnych opierają się na badaniach dotyczących częściowych pokryć. Niech $A = \{a_1, \ldots, a_n\}$ będzie niepustym, skończonym zbiorem i $S = \{B_1, \ldots, B_m\}$ będzie rodziną podzbiorów A taką, że $B_1 \cup \ldots \cup B_m = A$. Niech $\alpha \in \mathbb{R}$ i $0 \leq \alpha < 1$. Podrodzina $Q = \{B_{i_1}, \ldots, B_{i_t}\}$ rodziny S jest nazywana *α-pokryciem* dla pary (A, S) jeśli $|B_{i_1} \cup \ldots \cup B_{i_t}| \geq (1 - \alpha)|A|$.

Istnieje prosta redukcja problemu konstruowania 0-pokrycia o minimalnej liczności do problemu konstruowania 0-reguły decyzyjnej o minimalnej długości. Istnieje także odwrotna redukcja. Podobna sytuacja dotyczy częściowych pokryć i częściowych reguł (kiedy $\alpha > 0$). Fakt ten pozwolił wykorzystać różne matematyczne wyniki dotyczące problemu pokrycia zbioru uzyskane przez J. Cheriyana i R. Raviego, V. Chvá tala, U. Feigego [2], D.S. Johnsona [3], R.M. Karpa, M.J. Kearnsa, L. Lovásza, R.G. Nigmatullina, R. Raza i S. Safra, oraz P. Slavíka [25, 26], dla analizy częściowych reguł. Dodatkowo, korzystamy z techniki stworzonej przez D. Ślęzaka [28] dla dowodu NP-trudności optymalizacji częściowych reduktów.

Znane i uzyskane w tej pracy wyniki badań dla pokryć i częściowych pokryć mogą zostać wykorzystane w szerszym spektrum problemów rozważanych w teorii zbiorów przybliżonych, np. dla badania (i) reduktów i reguł systemów informacyjnych, (ii) reduktów i reguł dla tablic decyzyjnych z brakującymi wartościami, (iii) podsystemów danego systemu reguł, które „pokrywają” ten sam zbiór wierszy.

Rozprawa doktorska składa się z pięciu rozdziałów.

Rozdział 1 dotyczy częściowych pokryć i częściowych reguł decyzyjnych. Zawiera m.in podstawowe pojęcia, znane wyniki badań dla problemu pokrycia i częściowego pokrycia zbioru, relacje pomiędzy częściowymi pokryciami i częściowymi regułami decyzyjnymi, oszacowanie dokładności algorytmu zachłannego, oszacowanie górnych i dolnych granic minimalnej liczności częściowych pokryć oraz minimalnej długości częściowych reguł decyzyjnych, oszacowanie górnych granic liczności częściowych pokryć i odpowiednio długości częściowych reguł decyzyjnych, oraz wyniki badań dotyczące pokryć dla większej części problemów pokrycia zbioru i odpowiednio, wyniki badań dotyczące reguł decyzyjnych dla większej części binarnych tablic decyzyjnych.

W rozdziale 1 udowodniliśmy, że biorąc pod uwagę pewne założenia dotyczące klasy NP, algorytm zachłanny jest bliski (z punktu widzenia dokładności) najlepszym przybliżonym wielomianowym algorytmom dla optymalizacji częściowych pokryć.

Dane uzyskane podczas pracy algorytmu zachłannego mogą zostać wykorzystane do oszacowania dolnych i górnych granic minimalnej liczności częściowych pokryć. W ten sposób zostały znalezione najlepsze dolne i górne granice zależne od tych danych. Teoretyczne i eksperymentalne wyniki badań (rozdział 4) pokazują, że uzyskana dolna granica jest nietrywialna i może zostać użyta w praktycznych zastosowanich.

Dokonaliśmy także oszacowania granicy dokładności algorytmu zachłannego dla konstruowania częściowych pokryć, która nie zależy od liczności pokrywanego zbioru.

Udowodniliśmy, że dla większej części problemów pokrycia zbioru istnieją dokładne (i odpowiednio częściowe) pokrycia o małej liczności. Wyniki eksperymentów dla losowo generowanych problemów pokrycia zbioru pozwoliły sformuować nieformalną 0.5-hipotezę: dla większej części problemów pokrycia zbioru, algorytm zachłanny w każdej iteracji wybiera podzbiór, który pokrywa przynajmniej połowę niepokrytych dotychczas elementów.

Większa część wyników badań uzyskanych dla częściowych pokryć została uogólniona dla przypadku częściowych reguł decyzyjnych.

Pokazaliśmy, że przyjmując pewne założenia dotyczące klasy NP, algorytm zachłanny pozwala uzyskać wyniki bliskie wynikom uzyskiwanym przez najlepsze przybliżone wielomianowe algorytmy, dla minimalizacji długości częściowych reguł decyzyjnych.

Na podstawie danych uzyskanych podczas pracy algorytmu zachłannego zostały znalezione nietrywialne dolne i górne granice minimalnej długości częściowych reguł decyzyjnych.

Dla większej części losowo generowanych binarnych tablic decyzyjnych, algorytm zachłanny konstruuje proste częściowe reguły decyzyjne, o odpowiednio wysokiej jakości. Eksperymentalne i teoretyczne wyniki badań potwierdziły 0.5-hipotezę dla reguł decyzyjnych: dla większości tablic decyzyjnych, algorytm zachłanny podczas generowania reguły w każdej iteracji wybiera atrybut, który separuje przynajmniej połowę wierszy dotychczas nie oddzielonych.

Rozdział 2 dotyczy częściowych pokryć i częściowych reguł decyzyjnych z uwzględnieniem wag atrybutów. Zawiera m.in podstawowe pojęcia, znane wyniki badań dla problemu pokrycia i częściowego pokrycia zbioru z wagami, relacje pomiędzy częściowymi pokryciami i częściowymi regułami decyzyjnymi z uwzględnieniem wag, oszacowanie dokładności algorytmu zachłannego, porównanie zwykłego algorytmu zachłannego z wagami i algorytmu zachłannego z dwoma progami, oszacowanie dolnej granicy minimalnej wagi częściowego pokrycia oraz minimalnej całkowitej wagi atrybutów częściowej reguły decyzyjnej, oszacowanie górnej granicy wagi częściowego pokrycia i odpowiednio górnej granicy całkowitej wagi atrybutów tworzących częściową regułę decyzyjną, oraz wyniki eksperymentów dla częściowych pokryć z wagami i częściowych reguł decyzyjnych z wagami.

W rodziale 2 został zbadany przypadek, kiedy każdy podzbiór używany do pokrycia posiada własną wagę i należy zminimalizować całkowitą wagę podzbiorów tworzących częściowe pokrycie. Taka sama sytuacja dotyczy częściowych reguł decyzyjnych: każdy atrybut warunkowy posiada własną wagę i należy zminimalizować całkowitą wagę atrybutów tworzących częściową regułę decyzyjną. Waga atrybutu może chrakteryzować złożoność czasową, koszt lub ryzyko (w medycynie lub diagnostyce technicznej) obliczenia wartości atrybutu.

Większa cześć wyników badań przedstawionych w rozdziale 1 została uogólniona dla przypadku arbitralnych, naturalnych wag.

Udowodniliśmy, że biorąc pod uwagę pewne założenia dotyczące klasy NP, algorytm zachłanny z wagami jest bliski (z punktu widzenia dokładności) najlepszym przybliżonym wielomianowym algorytmom dla konstruowania częściowego pokrycia o minimalnej wadze i odpowiednio dla minimalizacji całkowitej wagi atrybutów tworzących częściową regułę decyzyjną.

Na podstawie danych uzyskanych podczas pracy algorytmu zachłannego oszacowaliśmy dolne granice minimalnej wagi częściowego pokrycia i odpowiednio minimalnej całkowitej wagi atrybutów tworzących częściową regułę decyzyjną.

Uogólniliśmy zwykły algorytm zachłanny z wagami i zbadalismy algorytm zachłanny z dwoma progami. Pierwszy próg określa dokładność konstruowanego pokrycia, drugi jest wewnętrznym parametrem rozważanego algorytmu. Udowodniliśmy, że dla większej części problemów pokrycia zbioru istnieje funkcja wagi i wartości progów takie, że waga częściowego pokrycia konstruowanego przez algorytm zachłanny z dwoma progami jest mniejsza, niż waga częściowego pokrycia konstruowanego przez zwykły algorytm zachłanny z wagami. Taka sama sytuacja dotyczy częściowych reguł decyzyjnych. Opierając się na algorytmie z dwoma progami stworzyliśmy nowe przybliżone wielomianowe algorytmy dla minimalizacji całkowitej wagi częściowych pokryć i częściowych reguł decyzyjnych. Wyniki dużej liczby eksperymentów dla losowo generowanych problemów pokrycia zbioru i binarnych tablic decyzyjnych pokazują, że algorytmy te mogą zostać wykorzystane w praktyce.

Rozdział 2 rozprawy doktorskiej, stanowi w pewnym sensie rozszerzenie rozdziału 1 dla przypadku, kiedy wagi nie są równe 1. Należy zauważyć, że nawet jeśli wagi są równe 1, to wyniki pracy algorytmów zachłannych rozważanych w tym rozdziale mogą się różnić od wyników przedstawionych w rozdziale 1. Np. dla reguł decyzyjnych liczba atrybutów wybieranych przez algorytm zachłanny jest taka sama, ale ostatnie atrybuty mogą się różnić.

Rozdział 3 dotyczy konstruowania wszystkich nieredukowalnych częściowych reguł decyzyjnych. Zawiera m.in podstawowe pojęcia dotyczące nieredukowalnych t-reguł decyzyjnych, oszacowanie długości i liczby tych reguł, algorytmy konstruowania wszystkich nierdukowalnych t-reguł decyzyjnych oraz wyniki eksperymentów.

Niech t będzie liczbą naturalną. Reguła decyzjna

$$(f_{j_1} = b_{j_1}) \wedge \ldots \wedge (f_{j_p} = b_{j_p}) \Rightarrow d_i$$

jest nazywana *t-regułą decyzyjną* dla tablicy decyzyjnej T i wiersza r, jeśli atrybuty $f_{j_1}, \ldots, f_{j_p}$ oddzielają od r przynajmniej $|U(T,r)| - t$ wierszy ze zbioru $U(T,r)$. W tej sytuacji powiemy, że atrybuty $f_{j_1}, \ldots, f_{j_p}$ *tworzą* t-regułę decyzyjną dla T i r. Jeśli usuniemy pewne warunki $f_{j_s} = b_{j_s}$, $s \in \{1, \ldots, p\}$, z rozważanej reguły, otrzymamy jej *podregułę*. Podreguła pewnej reguły jest nazywana *właściwą* jeśli nie jest równa początkowej regule. t-reguła decyzyjna dla T i r jest nazywana *nieredukowalną*, jeśli każda właściwa podreguła tej reguły nie jest t-regułą decyzyjną dla T i r.

W rozdziale 3 zostały zbadane binarne tablice decyzyjne z m atrybutami warunkowymi, w których liczba wierszy wynosi $\lfloor m^\alpha \rfloor$, gdzie α jest dodatnią liczbą rzeczywistą i częściowe reguły decyzyjne mogą pozostawić nie odseparowanych od danego wiersza najwyżej $5\left\lceil (\log_2 m)^\beta \right\rceil$ różnych wierszy z innymi decyzjami, gdzie β jest liczbą rzeczywistą taką, że $\beta \geq 1$.

Pokazaliśmy, że dla prawie wszystkich takich tablic, dla każdego wiersza z minor decyzją (minor decyzja to decyzja, która dotyczy najwyżej połowy wierszy z tablicy decyzyjnej), długość każdej nieredukowalnej częściowej reguły decyzyjnej nie jest większa od $\alpha \log_2 m$ i liczba nieredukowalnych częściowych reguł decyzyjnych nie jest daleka od $m^{\alpha \log_2 m}$.

Opierając się na tych wynikach udowodniliśmy, że nie istnieje algorytm, który dla prawie wszystkich tablic decyzyjnych, dla każdego wiersza z minor decyzją, konstruuje zbiór nieredukowalnych częściowych reguł decyzyjnych i posiada dla tych tablic wielomianową złożoność czasową zależną od długości danych wejściowych. Istnieje jednak algorytm, który dla prawie wszystkich tablic decyzyjnych, dla każdego wiersza z minor decyzją, konstruuje zbiór nieredukowalnych częściowych reguł decyzyjnych i posiada dla tych tablic wielomianową złożoność czasową zależną od długości danych wejściowych i długości danych wyjściowych.

Rozwiązanie problemu konstruowania wszystkich nieredukowalnych częściowych reguł decyzyjnych pozwoli np. (i) znaleźć najlepsze częściowe reguły, (ii) oszacować ważność atrybutów, (iii) tworzyć zespoły klasyfikatorów.

Rozdział 4 zawiera wyniki eksperymentów przeprowadzonych na na tablicach decyzyjnych znajdujących się w UCI Repository of Machine Learning Databases. Eksperymenty zostały podzielone na dwie grupy. Pierwsza dotyczy 0.5-hipotezy dla częściowych reguł decyzyjnych. Druga grupa eksperymentów dotyczy klasyfikacji z wykorzytsaniem częściowych reguł decyzyjnych.

Celem pierwszej grupy eksperymentów była weryfikacja 0.5-hipotezy dla danych rzeczywistych. Wykonalismy eksperymenty na 23 tablicach decyzyjnych. Wyniki 20 eksperymentów potwierdziły 0.5-hipotezę: podczas konstruowania częściowej reguły decyzyjnej, algorytm zachłanny w każdej iteracji wybiera atrybut, który oddziela od wiersza r przynajmniej połowę wierszy dotychczas nie oddzielonych, które są różne od r i posiadają inne decyzje. Oznacza to, że algorytm zachłanny często konstruuje krótkie częściowe reguły decyzyjne o odpowiednio wysokiej jakości. Szczególnie dla przypadku, kiedy 0.5-hipoteza jest prawdziwa, algorytm zachłanny konstruuje częściową regułę decyzyjną z siedmioma atrybutami, które separują od danego wiersza co najmmniej 99% różnych wierszy z innymi decyzjami. Takie krótkie częściowe reguły decyzyjne są dogodniejsze dla zrozumienia.

Celem drugiej grupy eksperymentów było porównanie dokładności klasyfikatorów opartych na dokładnych i częściowych regułach decyzyjnych. Rozważane podejście do konstruowania klasyfikatorów jest następujące: dla każdego wiersza danej tablicy decyzyjnej algorytm zachłanny konstruuje (częściową) regułę decyzyjną. Następnie przez usunięcie z takiej (częściowej) reguły decyzyjnej pewnych atrybutów otrzymujemy nieredukowalną (częściową) regułę decyzyjną. Uzyskany system reguł połączony z prostą procedurą głosowania stanowi klasyfikator. Metodą train-and-test wykonaliśmy eksperymenty dla 21 tablic decyzyjnych. W 11 przypadkach okazało się, że dokładność klasyfikatorów opartych na częściowych regułach decyzyjnych jest lepsza, niż dokładność klasyfikatorów opartych na dokładnych regułach decyzyjnych. Metodą cross-validation wykonaliśmy eksperymenty dla 17 tablic decyzyjnych. W 9 przypadkach okazało się, że dokładność klasyfikatorów opartych na częściowych regułach decyzyjnych jest lepsza, niż dokładność klasyfikatorów opartych na dokładnych regułach decyzyjnych.

Rozdział 5 dotyczy uniwersalnego problemu redukcji atrybutów. Zawiera m.in definicję problemu, warianty pojęcia redukt, oszacowanie dokładności algorytmu zachłannego dla konstruowania częściowego super-reduktu, oszacowanie dolnej granicy minimalnej liczności częściowego super-reduktu oraz oszacowanie górnej granicy liczności częściowego super-reduktu.

W rozdziale 5 został zbadany uniwerslany problem redukcji atrybutów. Niech T będzie tablicą decyzyjną i $\mathcal{P}$ będzie podzbiorem par różnych wierszy (obiektów) z T. Wówczas reduktem dla T względem $\mathcal{P}$ jest minimalny (w sensie zawierania) podzbiór atrybutów warunkowych, które oddzielają wszystkie pary wierszy od $\mathcal{P}$. Redukty dla systemów informacyjnych, redukty decyzyjne i lokalne redukty (reguły decyzyjne) dla tablic decyzyjnych, decyzyjne i loklane redukty oparte na uogólnionej decyzji, mogą być reprezentowane w takiej formie. W rozdziale zostały zbadane nie tylko dokładne ale także częściowe redukty. Został także przedstawiony scenariusz pracy z rzeczywistymi tablicami danych, które mogą zawierać zmienne ciągłe i dyskretne o dużej liczbie wartości, oraz zmienne z brakującymi wartościami.

Na podstawie wyników badań przedstawionych w rozdziale 1, dokonaliśmy oszacowania granic dokładności algorytmu zachłannego dla konstruowania super-reduktów. Udowodniliśmy, że poza kilkoma wyjątkami dotyczacymi klasy NP, algorytm zachłanny jest bliski (z punktu widzenia dokładności) przybliżonym wielomianowym algorytmom dla minimalizacji liczności częściowych super-reduktów. Na podstawie danych uzyskanych podczas pracy algorytmu zachłannego, uzyskaliśmy nietrywialną dolną granicę minimalnej liczności częściowego reduktu. Dokonaliśmy także oszacowania granicy dokładności algorytmu zachłannego, która nie zależy od liczności zbioru $\mathcal{P}$.

Eksperymentalne i teoretyczne wyniki badań przedstawione w pracy pokazują, że zastosowanie częściowych reguł decyzyjnych zamiast dokładanych reguł, pozwala uzyskać bardziej zwięzły opis wiedzy zawartej w tablicach decyzyjnych, oraz pozwala konstruować klasyfikatory o większej dokładności. Są to powody dla których należy zastosować częściowe reguły decyzyjne w dziedzinie odkrywania i wydobywania wiedzy z danych do reprezentacji wiedzy oraz do predykcji.

Wyniki badań uzyskane w tej pracy mogą być użyteczne dla naukowców z takich dziedzin jak np. uczenie maszynowe, odkrywanie i wydobywanie wiedzy z danych, zwłaszcza pracujących z teorią zbiorów przybliżonych.

Wyniki eksperymentów opisanych w rozdziałach 1 i 4 zostały przeprowadzone na oprogramowaniu, które wkrótce zostanie dołączone do systemu RSES – Rough Set Exploration System [22] (Instytut Matematyki, Uniwersytet Warszawski, kierownik projektu – prof. dr hab. Andrzej Skowron). Podstawowe funkcje biblioteki GRLib to: generowanie częściowych reguł decyzyjnych dla każdego wiersza lub wybranego wiersza z tablicy decyzyjnej, generowanie nieredukowalnych częściowych reguł decyzyjnych, wyznaczenie dolnej i górnej granicy minimalnej długości dla wybranej częściowej reguły decyzyjnej lub wyznaczenie minimalnej, średniej i maksymalnej wartości dolnej i górnej granicy minimalnej długości dla zbioru reguł.

Bibliografia

1. Bazan G.J., Nguyen S.H., Nguyen T.T., Skowron A., Stepaniuk J. (1998) Decision rules synthesis for object classification. In: E. Orłowska (ed.), Incomplete information: rough set analysis, Physica-Verlag, Heidelberg, 23–57
2. Feige U. (1996) A threshold of $\ln n$ for approximating set cover (preliminary version). In: Proc. 28th Annual ACM symposium on the theory of computing (Philadelphia, Pennsylvania, USA). ACM, 314–318
3. Johnson D.S. (1974) Approximation algorithms for combinatorial problems. J. Comput. System Sci. 9, 256–278
4. Komorowski J., Pawlak Z., Polkowski L., Skowron A. (1999), Rough sets: A tutorial. In S.K. Pal S.K. and Skowron A. (eds.) Rough-Fuzzy Hybridization: A New Trend in Decision-Making, Springer-Verlag, Singapor, 3–98.
5. Moshkov M.Ju. (2003) Greedy algorithm for set cover in context of knowledge discovery problems. Electronic Notes in Theoretical Computer Science 82(4), 174–185
6. Moshkov M.Ju. (2005) On construction of the set of irreducible partial covers. In: Lupanov O.B., Kasim-Zade O.M., Chashkin A.V., Steinhöfel K. (eds.) Proc. Stochastic algorithms: foundations and applications. LNCS 3777. Springer, Berlin Heidelberg New York, 38–44
7. Moshkov M.Ju., Piliszczuk M., Zielosko B. (2006) Lower bounds on minimal weight of partial reducts and partial decision rules. In: Wang G., Peters J.F., Skowron A., Yao Y. (eds.) Proc. Rough sets and knowledge technology. LNAI 4062, Springer, Berlin Heidelberg New York, 290–296
8. Moshkov M.Ju., Piliszczuk M., Zielosko B. (2006) Lower bound on minimal weight of partial cover based on information about greedy algorithm work. In: Proc. Information processing and management of uncertainty in knowledge-based systems (Paris, France)

9. Moshkov M.Ju., Piliszczuk M., Zielosko B. (2006) On greedy algorithm with weights for construction of partial covers. In: Kłopotek M.A., Wierzchoń S.T., Trojanowski K. (eds.) Proc. Intelligent information processing and web mining. Advances in soft computing. Springer, Berlin Heidelberg New York, 391–395
10. Moshkov M.Ju., Piliszczuk M., Zielosko B. (2007) On partial covers, reducts and decision rules with weights. In: Peters J.F., Skowron A., Düntsch I., Grzymała-Busse J.W., Orłowska E,. Polkowski L. (eds.) LNCS Transactions on Rough Sets VI. LNCS 4374. Springer, Berlin Heidelberg New York, 211–246
11. Moshkov M.Ju., Piliszczuk M., Zielosko B. (2007) Universal attribute reduction problem. In: Kryszkiewicz M., Peters J.F., Rybinski H., Skowron A. (eds.) Proc. Rough sets and intelligent systems paradigms. LNCS 4585. Springer, Berlin Heidelberg New York, 417–426
12. Moshkov M.Ju., Piliszczuk M., Zielosko B. (2007) On construction of partial reducts and irreducible partial decision rules. Fundamenta Informaticae 75, 357–374
13. Moshkov M.Ju., Piliszczuk M., Zielosko B. (2008) On partial covers, reducts and decision rules. In: LNCS Transactions on Rough Sets VIII. LNCS 5084. Springer, Berlin Heidelberg New York, 258–296
14. Nguyen H.S. (2006) Approximate Boolean reasoning: foundations and applications in data mining. In: Peters J.F., Skowron A. (eds.) LNCS Transactions on Rough Sets V. LNCS 4100. Springer, Berlin Heidelberg New York, 344–523
15. Nguyen H.S., Ślęzak D. (1999) Approximate reducts and association rules – correspondence and complexity results. In: Zhong N., Skowron A., Ohsuga S. (eds.) Proc. Rough sets, fuzzy sets, data mining, and granular-soft computing. LNAI 1711. Springer, Berlin Heidelberg New York, 137–145
16. Pawlak Z. (1991) Rough sets – theoretical aspects of reasoning about data. Kluwer Academic Publishers, Dordrecht Boston London
17. Pawlak Z. (1998) Rough set elements. In: Polkowski L., Skowron A. (eds.) Rough sets in knowledge discovery. Physica-Verlag, Heidelberg, 10–30
18. Pawlak Z., Skowron A. (2007) Rudiments of rough sets. Information Sciences 177, 3–27
19. Pawlak Z., Skowron A. (2007) Rough sets: some extensions. Information Sciences 177, 28–40
20. Pawlak Z., Skowron A. (2007) Rough sets and Boolean reasoning. Information Sciences 177, 41–73
21. Quafafou M. (2000) α-RST: a generalization of rough set theory. Information Sciences 124, 301–316
22. Rough Set Exploration System (RSES), http://logic.mimuw.edu.pl/∼rses

23. Skowron A. (2000) Rough sets in KDD. In: Shi Z., Faltings B., Musen M. (eds.) Proc. 16th IFIP World computer congress (Beijing, China). Publishing House of Electronic Industry, 1–14
24. Skowron A., Rauszer C. (1992) The discernibility matrices and functions in information systems. In: Slowinski R. (ed.) Intelligent decision support. Handbook of applications and advances of the rough set theory. Kluwer Academic Publishers, Dordrecht Boston London, 331–362
25. Slavík P. (1996) A tight analysis of the greedy algorithm for set cover (extended abstract). In: Proc. 28th Annual ACM symposium on the theory of computing (Philadelphia, Pennsylvania, USA). ACM, 435–441
26. Slavík P. (1998) Approximation algorithms for set cover and related problems. Ph.D. Thesis, University of New York at Buffalo
27. Ślęzak D. (1996) Approximate reducts in decision tables. In: Proc. Information processing and management of uncertainty in knowledge-based systems 3 (Granada, Spain), 1159–1164
28. Ślęzak D. (2000) Normalized decision functions and measures for inconsistent decision tables analysis. Fundamenta Informaticae 44, 291–319
29. Ślęzak D. (2001) Approximate decision reducts. Ph.D. Thesis, Warsaw University, Warsaw (in Polish)
30. Ślęzak D. (2002) Approximate entropy reducts. Fundamenta Informaticae 53, 365–390
31. Wakulicz-Deja A., Nowak A. (2007) From an information system to a decision support system. In: Kryszkiewicz M., Peters J.F., Rybinski H., Skowron A. (eds.) Proc. Rough sets and intelligent systems paradigms. LNCS 4585. Springer, Berlin Heidelberg New York, 454–464.
32. Wakulicz-Deja A., Paszek P. (2003) Applying Rough Set Theory to Multi Stage Medical Diagnosing. Fundamenta Informaticae 54, 387-408
33. Wróblewski J. (2001) Ensembles of classifiers based on approximate reducts. Fundamenta Informaticae 47, 351–360
34. Wróblewski J. (1995) Finding minimal reducts using genetic algorithm. In: Proc. 2nd Annual Joint Conference on Information Sciences, (Warsaw, Poland). 186–189
35. Zielosko B. (2005) On partial decision rules. In: Czaja L. (ed.) Proc. Concurrency, specification and programming 2 (Ruciane-Nida, Poland). Warsaw University, Warsaw, 598–609
36. Zielosko B., Piliszczuk M. (2008) Greedy algorithm for attribute reduction. Fundamenta Informaticae 85, 549–561