The Generalized Cordel Property in Combinatorial Optimization Problems

DISSERTATION

zur Erlangung des akademischen Grades
doctor rerum naturalium (Dr. rer. nat.)

vorgelegt dem Rat der Fakultät für Mathematik und Informatik
der Friedrich-Schiller-Universität Jena

von Diplom-Mathematikerin Lisa Schreiber
geboren am 30.07.1986 in Jena

1. Gutachter: ....

2. Gutachter: ....

3. Gutachter: ....

Tag der öffentlichen Verteidigung: ....

Contents

Deutsche Zusammenfassung

> „In einer beliebigen [Schach-]Position gibt es entweder nur einen, besten Zwangszug, oder aber drei etwa gleichwertige Züge.“ [Cor 1913]

Diese These bildet den Ausgangspunkt der vorliegenden Dissertation. Aufgestellt wurde sie von Oskar Cordel (1843-1913), einem Schachmeister, der viele bedeutende Beiträge zur Eröffnungstheorie geleistet hat. Leider starb Cordel, bevor er in einem Buch Erläuterungen zu seiner Behauptung niederschreiben und veröffentlichen konnte. Sein Verleger H. Ranneforth beendete Cordels letztes Buch „Theorie und Praxis des Schachspiels“ [Cor 1913] und formulierte im letzten Abschnitt das oben zitierte, sogenannte „Drei-Züge-Gesetz“. Wie Cordel es genau gemeint hat und ob er tatsächlich glaubte, damit eine allgemeingültige Regel gefunden zu haben, darüber können wir heute nur spekulieren.

Soweit wir wissen, gab es nach Cordels Tod keine weiteren Untersuchungen zu seiner Behauptung. Unser Ziel war es, in dieser Arbeit die mathematische Grundlage für eine Analyse der Gültigkeit des Gesetzes zu legen und einen ersten Eindruck zu gewinnen, ob es sich auch auf diskrete Optimierungsprobleme übertragen lässt.

Tatsächlich kann Cordels Überlegung in dieser Allgemeinheit nicht richtig sein. So kann man Schachstellungen finden, in denen es genau zwei Gewinnzüge gibt, die genauso schnell zum Ziel führen und somit gleich gut sind. Deshalb haben wir untersucht, wie häufig beziehungsweise unter welchen Bedingungen es gilt. Dafür wird zunächst in Kapitel 1 mit der **verallgemeinerten Cordel Eigenschaft (GeCoP)** eine mathematische Verallgemeinerung des Drei-Züge-Gesetzes eingeführt. Weiterhin gibt die **Cordel-Frequenz CF(k)** an, mit welcher Wahrscheinlichkeit eine zufällige Schachstellung mit k Figuren die verallgemeinerte Cordel Eigenschaft erfüllt. Während eine Cordel-Frequenz von 100% bedeutet, dass unsere Interpretation des Drei-Züge-Gesetzes (GeCoP) in jeder Schachstellung gilt, bedeutet eine Cordel-Frequenz von 0%, dass sie niemals gültig ist.

Eine Schwierigkeit bei der Überprüfung von (GeCoP) ist, dass man für eine gegebene Stellung die drei besten Züge ermitteln muss. Weiterhin sind für die exakte Berechnung der Cordel-Frequenz CF(k) **alle** legalen Schachstellungen mit k Figuren zu betrachten. Dies ist nur für Schachendspiele möglich, die komplett durchgerechnet werden können. Beispielsweise enthalten die Nalimov-Datenbanken komplette Analysen aller legalen Züge für Schachstellungen mit $3, 4$ oder 5 Figuren.
E. Bleicher [Ble 2005] hat uns umfassende Datensätze zur Verfügung gestellt, mit denen wir die Cordel-Frequenzen $\mathrm{CF}(3), \mathrm{CF}(4)$ und $\mathrm{CF}(5)$ exakt ermitteln konnten. Für Schachstellungen mit 6 Spielfiguren haben wir weiterhin mittels Monte-Carlo-Analyse

gute Schätzungen für die Cordel-Frequenzen bestimmt. Es zeigt sich, dass die Cordel-Frequenz für Schachendspiele zwischen 77% und 84% und damit deutlich über der 50%-Marke liegt. Oskar Cordels Drei-Züge-Gesetz ist damit eine gute Faustregel in Schachendspielen. Für Schach- und Zahleninteressierte bieten die ausführlichen Tabellen im Anhang (Appendix D, E und F) viel Spielraum für weiterführende Untersuchungen (nicht nur zum Drei-Züge-Gesetz).

Die Ergebnisse für Schach, die besagen, dass Cordels Drei-Züge-Gesetz zwar nicht in jeder Schachstellung, aber in circa 80% aller Endspiele gilt, haben uns darin bestärkt, die Definition der Cordel-Frequenz auch auf Optimierungsprobleme (vorrangig kombinatorische Optimierungsprobleme) zu übertragen. Dies bildet das Hauptaugenmerk dieser Dissertation.

In Analogie zu den drei besten Zügen im Schach kann man bei Optimierungsproblemen die drei besten Lösungen (sofern vorhanden) betrachten. In der Praxis ist aber häufig die zweitbeste Lösung eher uninteressant, da sie sich meist kaum von der optimalen Lösung unterscheidet. Als einfaches Beispiel lassen sich Routenplanungsprobleme anführen. Alle gängigen Routenplaner schlagen neben der optimalen Lösung mindestens eine gute Alternativroute vor, die bei weitem nicht die zweitschnellste Strecke ist. Diese würde sich nämlich vermutlich nur in einem kurzen Umweg über einen Parkplatz von der optimalen Lösung unterscheiden. Gute Alternativlösungen spielen in der Praxis daher eine sehr wichtige Rolle. Mit diesem Wissen haben wir uns dazu entschieden, die Cordel-Frequenz sehr allgemein für beliebige Auswahlregeln zu definieren. Untersucht haben wir:

- die naheliegende **Beste-Lösungen-Regel**, welche die drei besten zulässigen Lösungen auswählt
- und die **Penalty-Regel**, die eine optimale Lösung und zwei gute Alternativen, die sich hinreichend von der optimalen Lösung unterscheiden, auswählt.

Die Penalty-Methode, welche die Basis für die Auswahl von Lösungen mittels der Penalty-Regel bildet, wurde bereits in vielen anderen Arbeiten ausführlich untersucht [Ber 2000, ABS 2002, Sch 2003, Sam 2005, Dör 2009]. Bisher hat man sich dabei immer auf Optimierungsprobleme vom Summen-Typ beschränkt. In Kapitel 2 verallgemeinern wir die Penalty-Methode so, dass sie beispielsweise nun auch gut für das beschränkte und unbeschränkte Rucksackproblem sowie das Transportproblem mit nicht-ganzzahligen Kapazitäten anwendbar ist. Es wird gezeigt, dass trotz dieser Verallgemeinerung alle wichtigen Eigenschaften der Penalty-Methode (z.B. die Monotonie) erhalten bleiben.

In der ursprünglichen Variante der Penalty-Methode bestimmt man zunächst eine optimale Lösung $B^{(0)}$ des Minimierungsproblems. Anschließend werden alle Elemente (z.B. Straßenteile im Falle des Routenplanungsproblems), die von der optimalen Lösung verwendet werden, mit einem Faktor $(1 + \varepsilon)$ multipliziert und somit bestraft. Dabei ist $\varepsilon \geq 0$ der sogenannte Penalty-Parameter. Für dieses neue Optimierungsproblem mit bestraften Elementen wird nun wieder eine optimale Lösung bestimmt. Diese ist die ε-Penalty-Alternative $B^{(\varepsilon)}$.

Eine wichtige Eigenschaft der Penalty Methode, dass der Penalty-Parameter ε maßgeblich beeinflusst, wie die zugehörige Penalty-Alternative aussieht, kann sich in der Praxis als Problem erweisen. Wählt man ε zu klein, so erhält man statt einer alternativen Lösung wieder nur die bereits bekannte optimale Lösung $B^{(0)}$. Wählt man ε hingegen zu groß, so kann es passieren, dass die neu berechnete Alternative zu schlecht ist. Um also brauchbare Alternativen zu erhalten, muss man sehr vorsichtig bei der Wahl eines geeigneten ε sein.
Diese Arbeit schlägt ein neues Konzept vor, um dieses Problem zu umgehen. Ohne die konkrete Vorgabe eines Penalty-Parameters bestimmen wir die sogenannten k besten Penalty-Alternativen. Diese sind alle verschieden und zeichnen sich durch sehr gute Funktionswerte aus. Mit einer Modifikation des Algorithmus von Schwarz, die in Abschnitt 2.4 vorgestellt wird, lassen sich die k besten Penalty-Alternativen finden. Außerdem zeigen wir zahlreiche Möglichkeiten, die Berechnung zu beschleunigen, auf und diskutieren sie. Dieser Algorithmus ermöglicht uns auch die Ermittlung der drei Lösungen, die durch die Penalty-Regel für die Bestimmung der Cordel-Frequenz ausgewählt werden.

In den Kapiteln 3 und 4 werden experimentell ermittelte Cordel-Frequenzen für zahlreiche diskrete Optimierungsprobleme unter Verwendung der Penalty-Regel und der Beste-Lösungen-Regel präsentiert. Der Schwerpunkt liegt dabei deutlich auf der Untersuchung der Penalty-Regel. Hier wurden viele unterschiedliche Phänomene beobachtet, die in Abschnitt 3.7 zusammengefasst werden. Insgesamt hat sich gezeigt, dass typische Cordel-Frequenzen bei Verwendung der Penalty-Regel zwischen 15% und 30% liegen. Wenn deutlich andere Cordel-Frequenzen aufgetreten sind, ließ sich das immer mit einer speziellen Eigenschaft des Optimierungsproblems begründen. Im Gegensatz dazu scheinen typische Cordel-Frequenzen bei Verwendung der Beste-Lösungen-Regel bei circa 50% zu liegen. Diese Auswahlregeln bringen also erheblich unterschiedliche Cordel-Frequenzen mit sich.

Im theoretischen Kapitel 5 werden schließlich Erklärungsansätze für die experimentell ermittelten Cordel-Frequenzen aus den vorherigen Kapiteln geliefert. Unter einigen theoretischen Annahmen haben wir Cordel-Frequenzen unter Verwendung der Beste-Lösungen-Regel exakt berechnen können. Weiterhin untersuchten wir an einem theoretischen Modell, welche Frequenzen für die Penalty-Regel unter gewissen Annahmen zu erwarten sind. Alle theoretischen Untersuchungen bestätigten die experimentellen Ergebnisse und unsere in den vorigen Kapiteln aufgestellten Vermutungen.

Den Abschluss dieser Arbeit bildet eine umfangreiche Sammlung von offenen Fragen und Anregungen für weitere Untersuchungen zur verallgemeinerten Cordel Eigenschaft.

Acknowledgements

Writing this thesis has been one of the most significant academic challenges I have ever had to face. There have been many people, whose support, patience, and advice helped me finishing this work. It is to them that I owe my deepest gratitude for making this work enjoyable.

- Prof. Dr. Ingo Althöfer, who has been the best supervisor I could think of. His strong sense for exciting topics and questions motivated and advised me whenever I had to struggle during this thesis.
- Eiko Bleicher for providing those valuable comprehensive records for chess endgames and answering all of my questions regarding chess.
- My friends and colleagues Marlis Bärthel, Martin Dörnfelder, Jakob Erdmann, Nancy Kupfer, Alf Riedel, Christopher Schneider, Wesley Turner, and Rico Walter for their support, inspiration, and helpful discussions.
- Prof. Dr. Joachim Giesen and all members of his chair for Theoretical Computer Studies for many interesting, helpful, and funny lunch conversations.
- Being five months in Norway was a great experience for me and the lunch seminars of the algorithm group gave me insight into a similar but also different subject. Special thanks go to Fedor Fomin, Dag Haugland, Pinar Heggernes, Pim van 't Hof, and Ida Holen.
- Last but not least, I wish to thank everybody who proofread parts of this thesis for their invaluable feedback. These were Prof. Dr. Ingo Althöfer, Marlis Bärthel, Eiko Bleicher, Prof. Dr. Richard Lorenz, Alf Riedel, Christopher Schneider, Tim Lucke-genannt-Schönberg, my parents Kornelia and Peter Schreiber, and Wesley Turner. Especially, writing in English was sometimes a struggle for me, but these people helped me to master the challenge.

Chapter 1

The Cordel Property

1.1 Historical Background

Oskar Cordel (1843-1913) was a German chess master who wrote a couple of books on chess theory. His last book "Theorie und Praxis des Schachspiels" [Cor 1913] was published posthumously. Because Cordel died before having finished the work on this last book, his editor H. Ranneforth finalized it. In doing so he added a paragraph on what Cordel called the **"Three Moves Law"** though Ranneforth himself was not convinced of its validity. We give the following English translation of the original German citation, which can be found in Appendix A on page 129. The translation was done by I. Althöfer and L. Schreiber.

> "At this point it should be noted that Cordel established the so-called 'Three Moves Law'[1] based on long lasting analysis (several years). In an arbitrary chess position there is either only one best move (a 'forced move') or three moves of approximately equal strength. This law, so Cordel claimed, provides excellent assistance in testing the correctness of moves in chess. For him it established clarity, whenever chess books indicated that investigations were incomplete, such as for the Evans gambit, the Guioco Piano, the Ponziani Opening, etc. Where there were two apparently nearly equally good follow-ups, then his conviction was either one of them was wrong or a third one should exist. He stuck to his guns till he reached clarification. The preparatory work for a book on this 'Three Moves Law' was already far advanced. Here death upset his plans, too."

In no other part of discrete optimization a similarly old source on the distribution of extreme functional values of an optimization problem exists. Even more, there have been no investigations on the validity of this Three Moves Law set in motion until the end of 2008. But one has to keep in mind that the understanding of chess may have changed due to the upcoming of very strong chess computers since the 1980s. It may be that what Cordel called a law was a valid heuristic in former times but is no longer valid.

Because the paragraph written by Ranneforth is the only source on the Three Moves Law, we cannot be completely certain that Cordel meant the law as Ranneforth described is. But we know that Cordel worked on a book regarding the Three Moves Law which suggests that it was of high importance to him.

[1] Original German name: "Dreizügegesetz"

1.2 Mathematical Formulation of the Three Moves Law

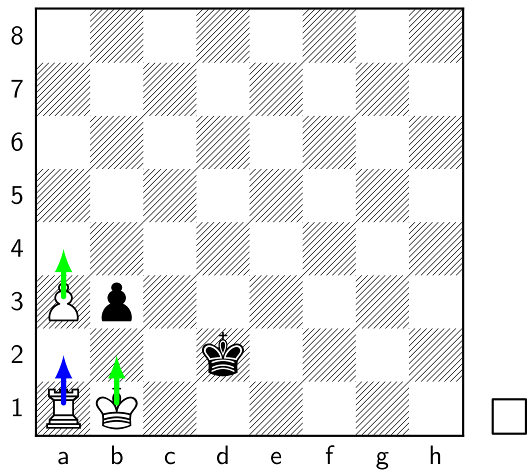
If we want to give a mathematical formulation of the Three Moves Law, it is essential to define what "equally good moves" are. Without loss of generality, from now on we assume that White is to move and introduce the following notation.

Definition 1.2.1 (Classification of Moves and Positions According to their Game Theoretical Values)
*We call a move a **W-move** (win) if it leads to a win for White assuming perfect play by both sides. Analogously we introduce the abbreviations **D-move** (draw) and **L-move** (loss). The outcome W, D, or L of the game, when move m is made, (again assuming perfect play by both sides) is called the **game theoretical value v(m) of m**.*

*Furthermore we call a chess position **W-position (or D- or L-position)** if White's best move is a W-move (or D- or L-move). Moreover, we introduce a more specific notation. Therefore let $m_1, \ldots, m_k$ be the k best moves in a given chess position ordered by their game theoretical values $s_1 = v(m_1), \ldots, s_k = v(m_k) \in \{W, D, L\}$. Then we call the position an S-position, where $S = (s_1, \ldots, s_k) \in \{W, D, L\}^k$ is a string containing the ordered game theoretical values of the k best moves.*

Note, that the term "move" includes only legal moves according to the chess rules.

Example 1.2.2
We consider the following chess position with White to move.



FEN: 8/8/8/8/8/Pp6/3k4/RK6 w - - 0 1

Obviously the white player has only three legal moves. Kb1-b2 and a3-a4 both lead to a win (they are W-moves) and Ra1-a2 leads to a draw (D-move). Hence, the chess position above is a WWD-position (with $k = 3$) or a WW-position if we consider only the two best moves. When regarding only the best move the position is just called a W-position.

We make the following observations.

1. W-moves are strictly better than D-moves, and D-moves are strictly better than L-moves.

2. A W-move is better than another W-move if it leads to a quicker checkmate (with fewer moves) assuming perfect play.

3. An L-move is better than another L-move if it leads to a terminal loss position more slowly assuming perfect play. This is because a longer game gives the opponent more opportunity to err, and the player to move a better chance of winning or at least reaching a draw.

4. D-moves are all equally good since there are no reasonable criteria for comparison. Prominent endgame tablebases for chess (for example the Nalimov tablebases) are distance to mate (DTM) tablebases which contain the number of moves to a final winning position assuming perfect play. As such the tablebases provide no values for D-moves.

This in mind, we get the following table for $k = 2$:

Move m_1	Move m_2	Comparison of m_1 and m_2
W-Move	W-Move	m_1 is better than m_2 if it leads to an earlier end
W-Move	D-Move	m_1 is always better than m_2
W-Move	L-Move	m_1 is always better than m_2
D-Move	D-Move	m_1 and m_2 are always equally good
D-Move	L-Move	m_1 is always better than m_2
L-Move	L-Move	m_1 is better than m_2 if it leads to a later end

Based on this classification we define the **value $\boldsymbol{f(m)}$ of a W-move (or L-move) $\boldsymbol{m}$** as the number of White's moves until the end of the game assuming perfect play on both sides. Remember that we assumed without loss of generality that White is to move. This allows us a more precise ranking of moves which does not only refer to the game theoretical value win, draw, or loss. Furthermore it enables us to compare two W-moves (or L-moves).

Definition 1.2.3 (Comparing two W-moves)
We say that a ***W-move $\boldsymbol{w_1}$ is better than another W-move $\boldsymbol{w_2}$*** *if*

$$f(w_1) < f(w_2)$$

holds. Furthermore we call two W-moves w_1 and w_2 ***equally good*** *if*

$$f(w_1) = f(w_2)$$

holds.

By analogy we say that an ***L-move l_1 is better than another L-move l_2*** *if*

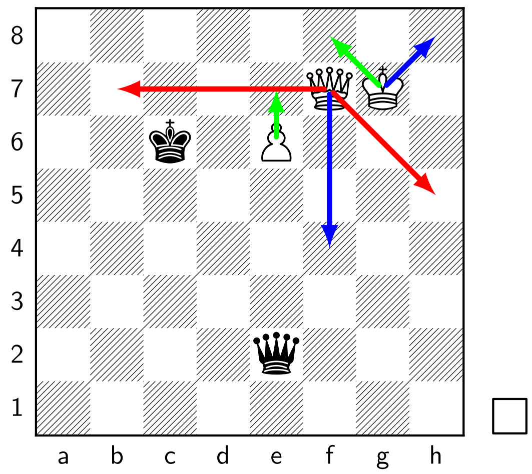
$$f(l_1) > f(l_2)$$

holds and we call two L-moves l_1 and l_2 ***equally good*** *if*

$$f(l_1) = f(l_2)$$

holds.

Example 1.2.4
We consider the following chess position and look at the marked moves. Here, moves marked in green are W-moves, moves in blue are D-moves and moves in red are L-moves. This color code is preserved in each of the following examples, too.



FEN: 8/5QK1/2k1P3/8/8/8/4q3/8 w - - 0 1

These six moves have the following values.

Move m	*Result Assuming Perfect Play*	*Type*	*Value f(m)*
e6-e7	*Win in 12 moves*	*W-move*	*12*
Kg7-f8	*Win in 61 moves*	*W-move*	*61*
Kg7-h8	*Draw*	*D-move*	–
Qf7-f4	*Draw*	*D-move*	–
Qf7-b7	*Loss in 9 moves*	*L-move*	*9*
Qf7-h5	*Loss in 8 moves*	*L-move*	*8*

Since the first W-move e6-e7 has a smaller value than the second W-move Kg7-f8

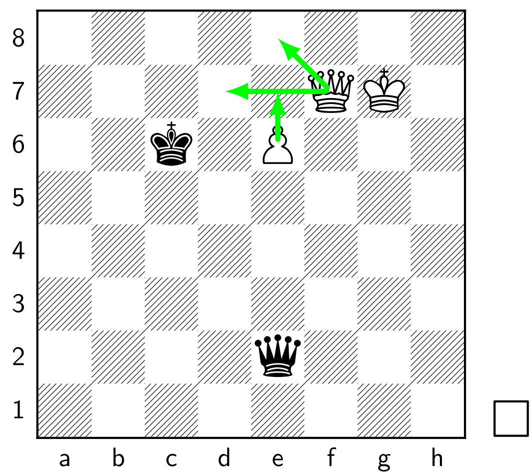
$$f(\textit{e6-e7}) = 12 < 61 = f(\textit{Kg7-f8})$$

it is better.
Likewise the first L-move Qf7-b7 is better than the second L-move Qf7-h5.

$$f(\textit{Qf7-b7}) = 9 > 8 = f(\textit{Qf7-h5})$$

For chess endgames with at most six pieces (incl. kings) there exist the Nalimov tablebases named in honour of their creator E. Nalimov. E. Bleicher provides an applet to query all of the currently published Nalimov tablebases on his website [Ble 2005]. There one can enter an arbitrary chess position with at most six pieces and the database tells immediately which of the legal moves lead to a victory, draw or loss and how many moves on each side have to be played in perfect play until the game ends. Thus the applet supplies the game-theoretical values (win, draw or loss) and the values of each move. In this way the values in the previous Example 1.2.4 and the following Example 1.2.5 were obtained.

Example 1.2.5
We consider the following chess position, which is the same as in Example 1.2.4.



FEN: 8/5QK1/2k1P3/8/8/8/4q3/8 w - - 0 1

By querying Bleichers applet [Ble 2005] we obtain the following tables which consist of all moves and their values.

W-Moves:

	Move	***Value***
$m_1 =$	*e6-e7*	*Win in 12*
$m_2 =$	*Qf7-d7*	*Win in 28*
$m_3 =$	*Qf7-e8*	*Win in 30*
	Kg7-g8	*Win in 31*
	Qf7-f5	*Win in 40*
	Qf7-f6	*Win in 59*
	Qf7-g6	*Win in 59*
	Kg7-f7	*Win in 61*
	Qf7-e7	*Win in 64*
	Kg7-h6	*Win in 64*
	Kg7-g6	*Win in 65*
	Kg7-f6	*Win in 66*

D-Moves:

Move	***Value***
Qf7-a7	*Draw*
Qf7-f4	*Draw*
Qf7-f8	*Draw*
Qf7-g8	*Draw*
Kg7-h7	*Draw*
Kg7-h8	*Draw*

L-Moves:

Move	***Value***
Qf7-b7	*Loss in 9*
Qf7-c7	*Loss in 8*
Qf7-f3	*Loss in 8*
Qf7-f2	*Loss in 8*
Qf7-f1	*Loss in 8*
Qf7-h5	*Loss in 8*

Thus we have e6-e7 as best move m_1, Qf7-d7 as second-best move m_2, and Qf7-e8 as third-best move m_3 with values

$$f(m_1) = 12 < f(m_2) = 28 < f(m_3) = 30 .$$

The best move m_1 in an arbitrary chess position is not always uniquely defined. But since we are only interested in the values of the best moves this ambiguity is not a problem. Thus, whenever we speak of the three best moves, we have in mind that they must not be uniquely determined.

Remark 1.2.6
The Nalimov endgame tablebases are distance to mate (DTM) tablebases which do not contain the ***fifty-move rule*** *of the World Chess Federation [FIDE]. According to this FIDE rule (for human play) a player can claim a draw if no pawn has been moved and no capture has been made in the last fifty moves. This rule prevents the game from continuing indefinitely without real progress.*
Since the tablebases do not take this rule into account, some W-moves with values greater than 50 *could lead to a draw in practice through an opponent's claim. Furthermore, there could exist a longer sequence of moves which leads to a final winning position without the possibility of a draw claim.*

Keeping our definition of equally good moves in mind, we want to decide whether Cordel's Three Moves Law holds in a given WWW- or LLL-position or not. Since Cordel never defined what he meant by "equally good best moves", this will be in fact only our interpretation of Cordel's law. But we still want to use his term Three Moves Law. We make the following observation regarding the connection between differences d_1, d_2 of the functional values and the Three Moves Law.

Lemma 1.2.7
Consider a WWW- or LLL-position where m_1, m_2, and m_3 are the three best moves with $f(m_1) \leq f(m_2) \leq f(m_3)$. Furthermore let

$$d_1 := |f(m_1) - f(m_2)| \quad \textit{and} \quad d_2 := |f(m_2) - f(m_3)|$$

be the differences of their functional values.

Then

$$d_1 \geq d_2 \tag{1.1}$$

implies that we have either exactly one best move m_1 or at least three equally good best moves m_1, m_2, m_3 with

$$f(m_1) = f(m_2) = f(m_3) .$$

Hence, the Three Moves Law holds.

Proof. Consider a WWW-position where the absolute differences of the values of the three best moves fulfill (1.1).
In case $d_1 > 0$ we have exactly one best move and the Three Moves Law holds. Thus, assume the best and second-best move to be equally good which implies $d_1 = 0$. Since d_1 and d_2 are defined as absolute values, it follows that $d_1, d_2 \geq 0$ holds. Hence,

$$0 = d_1 \overset{(1.1)}{\geq} d_2 \geq 0 \quad \Rightarrow \quad d_2 = 0$$

and the values of m_1, m_2, and m_3 are all equal. Consequently the three moves are equally good and the Three Moves Law holds in this case, too.

Analogously, (1.1) implies that the Three Moves Law holds in a given LLL-position.■

Based on this observation we introduce the **generalized Cordel property**, which is a property of WWW- and LLL-positions motivated by Cordel's Three Moves Law.

Definition 1.2.8 (Generalized Cordel Property for the Three Best Moves)
Let m_1, m_2, and m_3 be the three best moves in a given WWW- or LLL-position.

We say that m_1, m_2, and m_3 fulfill the ***generalized Cordel property (GeCoP)***[2] *if and only if*

$$d_1 := |f(m_1) - f(m_2)| \quad \geq \quad |f(m_2) - f(m_3)| =: d_2\,. \tag{GeCoP}$$

holds.

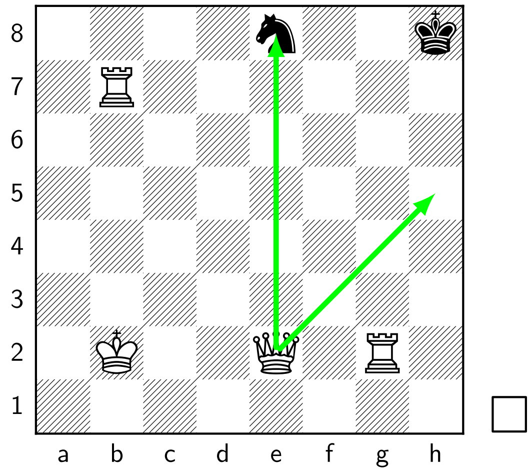
With the help of Definition 1.2.8 we want to examine how often Cordel's Three Moves Law is valid for chess. But obviously by our definition of equally good moves, the Three Moves Law cannot always be true. One can easily find chess positions with exactly two best moves, as in the following example.

Example 1.2.9
In the constructed chess position shown on the next page the white player has exactly two ways to checkmate immediately: $m_1 =$ Qe2×e8 and $m_2 =$ Qe2-h5. Thus $f(m_1) = f(m_2) = 1$ holds. Furthermore, the next best move m_3 leads to a win in two moves. Hence, $f(m_3) = 2$ and

$$d_1 = |f(m_1) - f(m_2)| = 0 \quad < \quad 1 = |f(m_2) - f(m_3)| = d_2$$

hold.

[2] We use the abbreviation (GeCoP) instead of the shorter form (GCP) in order to give an pronounceable name.



FEN: 4n2k/1R6/8/8/8/8/1K2Q1R1/8 w - - 0 1

Thus far, we have considered the generalized Cordel property only for the three best moves. Now, we want to go one step further and define the Cordel property for three arbitrarily chosen W-moves (or L-moves) m_1, m_2, m_3.

Definition 1.2.10 (Generalized Cordel Property for Chess)
Let m_1, m_2, and m_3 be three arbitrary W-moves or three L-moves in a given chess position. Furthermore let m_1 be the best and m_3 be the worst of the three moves. That means $f(m_1) \leq f(m_2) \leq f(m_3)$ should hold.

*We say that the ordered triple (m_1, m_2, m_3) fulfills the **generalized Cordel property (GeCoP)** if and only if*

$$d_1 := |f(m_1) - f(m_2)| \quad \geq \quad |f(m_2) - f(m_3)| =: d_2 \qquad \text{(GeCoP)}$$

holds.

Definition 1.2.10 is very general and also slightly unspecific, because it does not specify how to choose the three moves m_1, m_2, and m_3. As in the underlying Three Moves Law, m_1, m_2, and m_3 could be the three best moves. But one can imagine other selection rules also.

Definition 1.2.11 (Selection Rule)
*A rule S is called **selection rule** if it chooses three moves m_1, m_2, and m_3, where m_1 is the best and m_3 is the worst of the three moves.*

We suggest the following three selection rules for chess, where the first suggested rule is the previously considered best moves rule.

Definition 1.2.12 (Selection Rules for Chess)
For chess we define the following three selection rules.

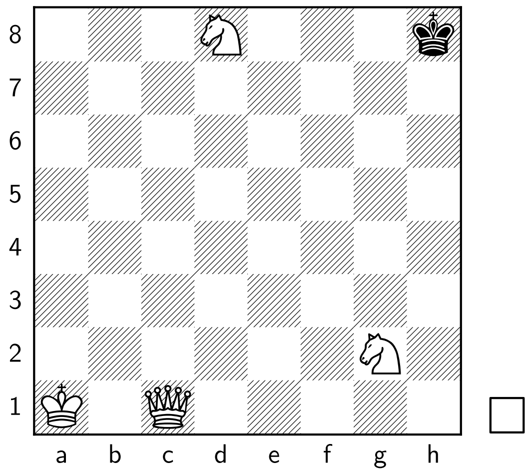
*(i) m_1, m_2, and m_3 are the three best moves in a given chess position. This is the **best moves rule**.*

(ii) *Let* $p_1, p_2, \ldots, p_k$ $(k \geq 3)$ *be the remaining pieces of the player who is in turn. Furthermore let* m_{p_i} *denote the best move he can make with piece* p_i*. The* ***best move per piece rule*** *selects the three best of these* k *candidates* $m_{p_1}, \ldots, m_{p_k}$*.*

(iii) *If the player to move has at least three different piece* ***types*** *left, then we can look for the best move for each* ***piece type*** *and pick* m_1, m_2, m_3 *as the three best of these moves. Since* m_1, m_2*, and* m_3 *are moves of different piece types by definition, we call this rule* ***best move per piece type rule****.*

As already mentioned before, m_1, m_2, m_3 may not be uniquely determined. This is not a problem, since we are only interested in their functional values. We illustrate the three suggested rules in the following example.

Example 1.2.13
Consider the following position.



Move	***Value***
$q_1 :=$ *Qc1-c7*	*Win in 3*
$q_2 :=$ *Qc1-g5*	*Win in 3*
$q_3 :=$ *Qc1-h6*	*Win in 3*
$n_{1,1} :=$ *Nd8-e6*	*Win in 3*
$q_4 :=$ *Qc1-c3*	*Win in 4*
$q_5 :=$ *Qc1-b2*	*Win in 4*
$n_{2,1} :=$ *Ng2-f4*	*Win in 4*
$n_{2,2} :=$ *Ng2-h4*	*Win in 4*
$k_1 :=$ *Ka1-b1*	*Win in 5*
$k_2 :=$ *Ka1-a2*	*Win in 5*
$k_3 :=$ *Ka1-b2*	*Win in 5*
⋮	⋮

FEN: 3N3k/8/8/8/8/8/6N1/K1Q5 w - - 0 1

Thus,

(i) *The best moves rule selects three of the four equally best moves* q_1, q_2, q_3 *and* $n_{1,1}$*.*

(ii) *The best move per piece rule selects*

- *one of the three equally best queen's move* q_1, q_2, q_3*,*
- *the best move* $n_{1,1}$ *of a knight (in this case the d8 knight), and*
- *one of the two equally best moves* $n_{2,1}, n_{2,2}$ *of the other knight (the g2 knight).*

(iii) *Finally the best move per piece type rule selects*

- *one of the three equally best queen's move* q_1, q_2, q_3*,*
- *the best knight move* $n_{1,1}$*, and*
- *one of the three equally best king's moves* k_1, k_2, k_3*.*

Though in this example all three selection rules provide different tuples of moves, there of course exist also chess positions where the rules select the same moves.

Note that not in every chess position the three suggested selection rules choose three W-moves or three L-moves as required in the definition of the generalized Cordel property. Thus, only if the selection rule chooses three W- or three L-moves, the validity of the generalized Cordel property can be checked.

Since the Three Moves Law does not hold in every chess position, as shown in Example 1.2.9, Althöfer, Bleicher, and Schreiber have tried to figure out, how often the Three Moves Law holds. Therefore we introduce the Cordel frequency to indicate how often the Three Moves Law, or to be more precise, the generalized Cordel property (GeCoP) holds. This and the results for chess are presented in the next Section 1.3. Afterwards, in Section 1.4 we generalize the concept and move on from chess to arbitrary optimization problems. This is what the remaining Chapters 2 - 5 deal with.

1.3 The Validity of the Three Moves Law in Chess Endgames

Althöfer, Bleicher, and Schreiber checked the Nalimov tablebases (cf. [Ble 2005]) for different chess endgames with 3 to 6 pieces. A report with the results of this huge stastical evaluation is in preparation [ABS 2012]. In [Alt 2009] Althöfer stated that for chess endgames the following two propositions hold:

1. It is **relatively rare** that a chess position has exactly two best moves.
2. The **average** distance between the best and second-best moves in a chess position is larger than the average distance between the second-best and third-best moves.

Based on these observations, Althöfer posed the question: "In which parts of a chess game (opening, middlegame or endgame) are the properties 1 and 2 most prominent?"

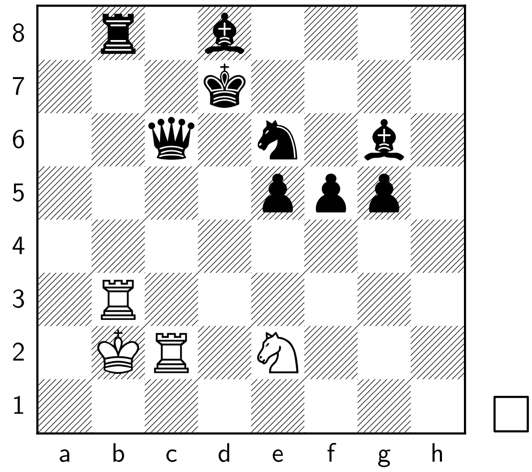
Motivated by this claim, which suggests that the Three Moves Law holds very often in chess endgames at least, we counted how often the Three Moves Law is valid in endgames with 3 to 6 pieces (including kings). To give more structure, we sorted the endgames by their piece distributions.

Definition 1.3.1 (Piece Distribution)
*Given a chess position S the **piece distribution** consists of the type of chess pieces of the player who is to move and the type of chess pieces of the opponent. In doing so the chess pieces of each player are sorted according to the order*

$$k \text{ (king)}, q \text{ (queen)}, r \text{ (rook)}, b \text{ (bishop)}, n \text{ (knight)}, p \text{ (pawn)}.$$

The position of the chess pieces on the board does not matter.

Example 1.3.2
We consider the following example where White is to move.



FEN: `1r1b4/3k4/2q1n1b1/4ppp1/8/1R6/1KR1N3/8 w - - 0 1`

White, on move, has the following remaining chess pieces sorted according to their values: 1 king, 2 rooks, 1 knight.
The black player who is not to move has the following remaining chess pieces sorted according to their values: 1 king, 1 queen, 1 rook, 2 bishops, 1 knight, 3 pawns.
*Thus, this is a **krrn** versus **kqrbbnppp** piece distribution. We often omit the "versus" and write **krrnkqrbbnppp**.*

For each of these piece distributions Bleicher provided detailed information by querying the Nalimov tablebases for all chess endgames with at most 6 pieces (including kings). He gave us a textfile for each piece distribution with the following information.

1. The number of WWW, WWD, WWL, WDD, WDL, WLL, DDD, DDL, DLL, LLL instances (for instances with at least three feasible moves).
2. The number of WW, WD, WL, DD, DL, and LL instances (for instances with exactly two legal moves).
3. The number of W, D, and L instances (for instances with exactly one legal move).
4. The number of checkmate instances (player who is to move has no feasible move and is checkmated) and stalemate instances (player who is to move has no feasible move but is not checkmated).
5. The number of WWW (or LLL) instances, which fulfill
   a) $d_1 > d_2$,
   b) $d_1 = d_2$, and
   c) $d_1 < d_2$.

Appendix B on page 131 shows the textfile for the piece distribution "king plus knight versus king plus pawn" as an illustration.

Given these data files we counted how often WWW- and LLL-positions with (GeCoP) occurred. The remaining chess positions were divided into Cordel and non-Cordel positions without regard to the concrete values (move numbers until the end of game). The following table shows the whole classification.

number of of feasible moves	Cordel positions	non-Cordel positions
1	W, D, L	-
2	WD, WL, DL	WW, DD, LL
at least 3	WWW with (GeCoP), WDD, WDL, WLL, DDD, DLL, LLL with (GeCoP)	WWW without (GeCoP), WWD, WWL, DDL, LLL without (GeCoP)

Appendix C starting on page 133 contains examples for each class of chess positions with at least three feasible moves.

Definition 1.3.3 (Cordel Frequency for Chess)
*The **Cordel frequency CF_d for a given piece distribution d** denotes the probability that a random nonterminal chess position (which is not already checkmate or stalemate) with piece distribution d belongs to one of the following categories:*

(i) WDD, WDL, WLL, DDD, DLL

(ii) WWW-position with $d_1 \geq d_2$ (GeCoP)

(iii) LLL-position with $d_1 \geq d_2$ (GeCoP)

(iv) WD, WL, DL (with exactly two feasible moves at all)

(v) W, D, L (with only one feasible move)

*Furthermore we define the **Cordel frequency $CF(k)$** as the mean of all Cordel frequencies CF_d for piece distributions d with exactly k chess pieces (including kings).*

Remark 1.3.4
By defining the Cordel frequency as the mean of the Cordel frequencies for all piece distributions, we assume each piece distribution to be equally probable. This is particularly meaningful since the data sets Bleicher provided, consist of different numbers of instances for each piece distribution. The reason for this is the different board symmetry and number of legal chess positions depending on the piece distribution.

For chess endgames with 3 chess pieces and the best moves rule the following relative frequencies occurred.

	WWW $d_1 \geq d_2$	WDD	WDL	WLL	DDD	DLL	LLL $d_1 \geq d_2$	WD	WL	DL	W	D	L	Sum
kkq	–	–	–	–	–	6%	51%	–	–	3%	–	1%	5%	66%
kkr	–	–	–	–	–	8%	54%	–	–	2%	–	0%	2%	66%
kpk	45%	8%	–	–	23%	–	–	0%	–	–	0%	0%	–	76%
kqk	78%	–	–	–	–	–	–	–	–	–	–	–	–	78%
krk	80%	–	–	–	–	–	–	–	–	–	–	–	–	80%
kkp	–	–	–	–	27%	6%	48%	–	–	0%	–	0%	0%	82%
kkb	–	–	–	–	95%	–	–	–	–	–	–	1%	–	96%
kkn	–	–	–	–	97%	–	–	–	–	–	–	1%	–	98%
kbk	–	–	–	–	100%	–	–	–	–	–	–	–	–	100%
knk	–	–	–	–	100%	–	–	–	–	–	–	–	–	100%
Mean	20%	1%	–	–	44%	2%	15%	0%	–	0%	0%	0%	1%	84%

Table 1.3.1: Frequencies of Cordel positions in chess endgames with 3 pieces (including kings). For each piece distribution the value in the right column is the corresponding Cordel frequency.

The "–" (for example for kkq, WWD) means that there exist no kkq endgame positions that are WWD. Otherwise, 0% means that there were instances which had this classification, but below 0.5%. Furthermore, note that all values are rounded up or down with the consequence that the sums in the right column are not always equivalent to the sums of the rounded values. By definition, this last column contains the Cordel frequencies of each piece distribution. The entry on the bottom right which is the mean of the Cordel frequencies for each piece distribution and equals the Cordel frequency for endgames with 3 pieces and the best moves rule.

For the sake of completeness, the following table shows all relative frequencies for non-Cordel instances (for endgames with 3 pieces and the best moves rule).

	WWW $d_1 < d_2$	WWD	WWL	DDL	LLL $d_1 < d_2$	WW	DD	LL	Sum
kkq	–	–	–	–	24%	–	–	10%	34%
kkr	–	–	–	–	29%	–	–	5%	34%
kpk	18%	5%	–	–	–	0%	0%	–	24%
kqk	22%	–	–	–	–	–	–	–	22%
krk	20%	–	–	–	–	–	–	–	20%
kkp	–	–	–	8%	9%	–	0%	1%	18%
kkb	–	–	–	–	–	–	4%	–	4%
kkn	–	–	–	–	–	–	2%	–	2%
kbk	–	–	–	–	–	–	–	–	–
knk	–	–	–	–	–	–	–	–	–
Mean	6%	1%	–	1%	6%	0%	1%	2%	16%

Table 1.3.2: Frequencies of **non-Cordel** positions in chess endgames with 3 pieces (including kings).

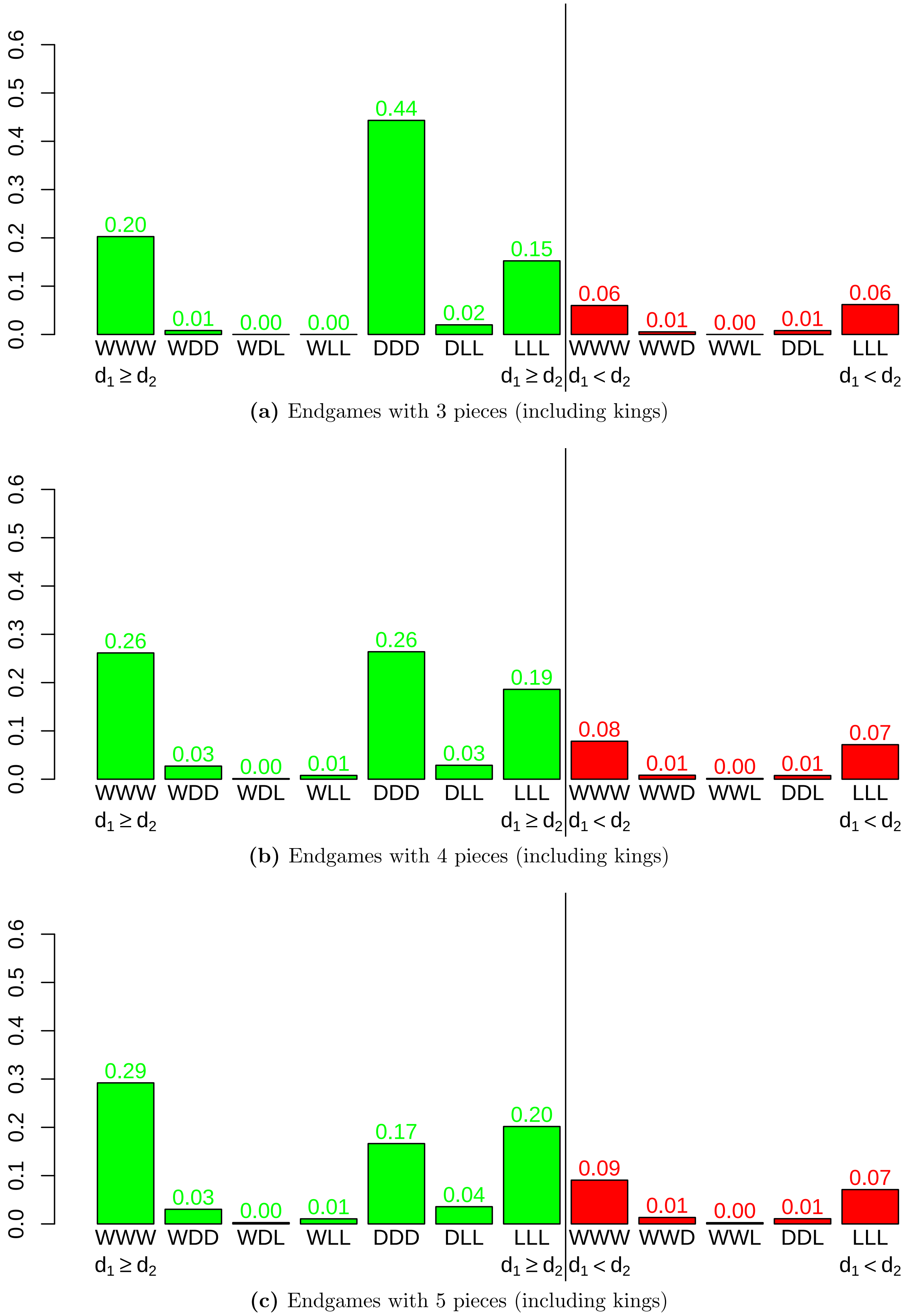
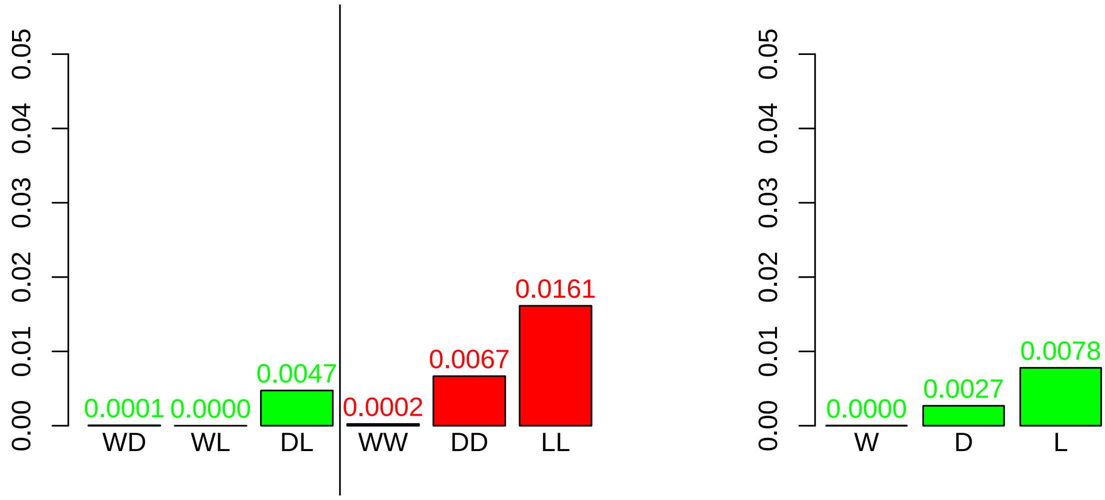
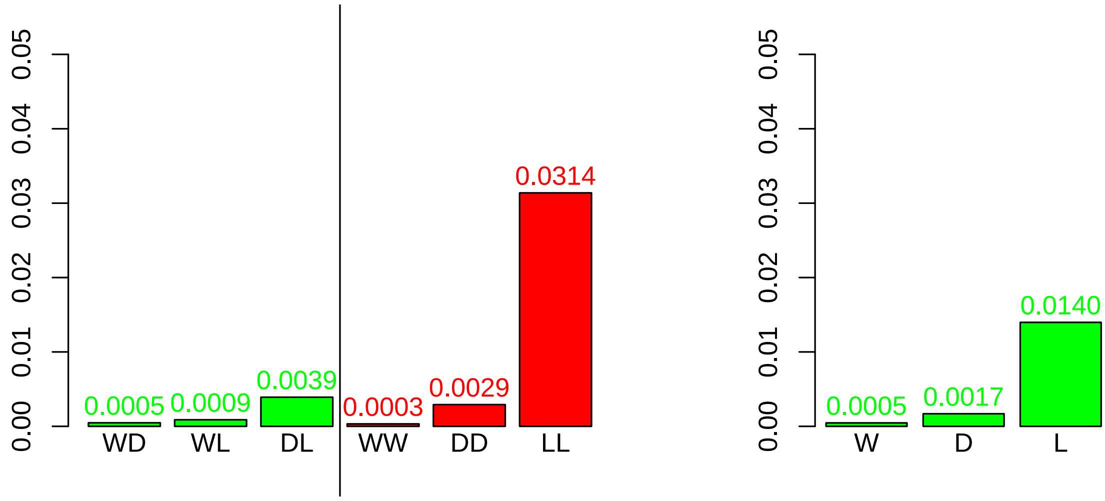


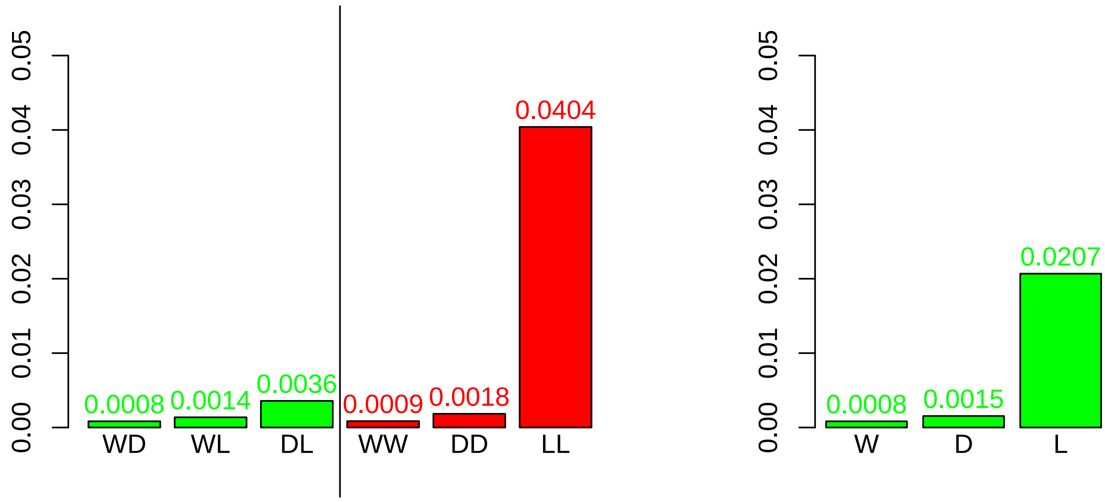
Figure 1.3.3: Best moves rule: Relative frequencies of different types of chess positions in endgames with 3, 4, or 5 pieces (including kings) with at least three feasible moves. We use green bars for Cordel and red bars for non-Cordel positions.



(a) Endgames with 3 pieces (including kings)



(b) Endgames with 4 pieces (including kings)



(c) Endgames with 5 pieces (including kings)

Figure 1.3.4: Best moves rule: Relative frequencies of different types of chess positions in endgames with 3, 4, or 5 pieces (including kings) with only two (left) or one (right) feasible moves. We use green bars for Cordel and red bars for non-Cordel positions.

Figure 1.3.3 (a)-(c) and Figure 1.3.4 (a)-(c) show the frequencies of each class of chess positions for endgames with three, four, or five pieces for the best moves rule. For $k = 3$ pieces these are the column means from the two tables above. The green marked bars indicate the instances where the Three Moves Law holds. Thus, adding up these relative frequencies provides the Cordel frequencies for the best moves rule.

The figures show that the results for endgames with 3, 4, or 5 pieces do not differ in principle. As the relative frequency of DDD instances decreases enormously, depending on the number of pieces, and the frequencies of DL and D instances decrease slightly, the frequencies of the remaining Cordel positions increase. On the other hand, all types of non-Cordel positons occur slightly more often in endgames with more pieces. Hence, the Cordel frequency slightly decreases (as the number of pieces on the board grows from 3 to 5). This is confirmed by the following table.

number of pieces	$k = 3$	$k = 4$	$k = 5$
CF(k)	$\approx 84\%$	$\approx 80\%$	$\approx 77\%$

Table 1.3.5: Cordel frequencies for chess endgames with the best moves rule.

Note, that the values are established by examining the whole databases and not through consideration of random chess positions.

In order to give an even better impression of the validity of the Three Moves Law, Bleicher also analyzed chess endgames with 6 pieces. Since there exist no tablebases for endgames where one player has only one piece (his king) left, only the 515 remaining 2 vs. 4, 3 vs. 3, and 4 vs. 2 piece distributions have been considered. But with our program's duration of about 120 hours per piece distribution[3], a complete analysis would take about 10 years. That is why we applied Monte Carlo sampling with an investigation of $1,000,000$ random chess position for each piece distribution.

In the same way the Cordel frequencies for all piece distributions with 3 to 6 pieces and the two remaining selection rules (the best move per piece and the best move per piece type rule) were established. The detailed results for each piece distribution and all three selection rules can be found in Appendix D starting on page 139 (best moves rule), Appendix E starting on page 159 (best move per piece rule) and Appendix F starting on page 179 (best move per piece type rule).

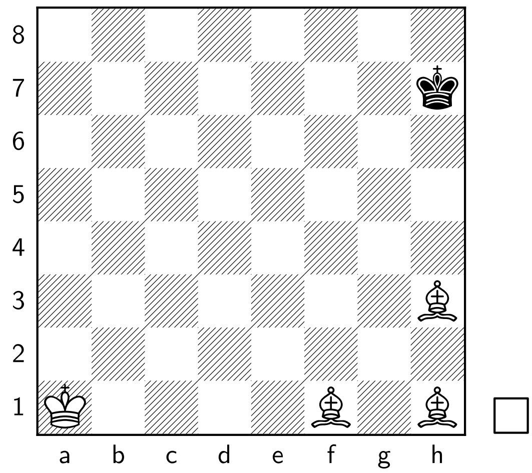
Altogether, the following Cordel frequencies were obtained. While the values for $k \in \{3, 4, 5\}$ were established by full tablebase analysis, the values for $k = 6$ are the results of Monte Carlo simulations.

number of pieces	$k = 3$	$k = 4$	$k = 5$	$k = 6$
CF(k) best moves rule	$\approx 84\%$	$\approx 80\%$	$\approx 77\%$	$\approx 75\%$
CF(k) best move per piece rule	$\approx 52\%$	$\approx 58\%$	$\approx 63\%$	$\approx 59\%$
CF(k) best move per piece type rule	$\approx 52\%$	$\approx 52\%$	$\approx 57\%$	$\approx 54\%$

[3] The durations may vary considerably depending on the piece distribution.

The results above show that the Cordel frequencies for the best moves rule are considerably greater than 50%. Thereby we have demonstrated that the Three Moves Law is a good rule of thumb, but of course it does not hold in every chess position. Nevertheless, our results cannot be carried over to reality without mentioning that we allowed any theoretically possible chess position regardless of whether the position is reasonable or not.

Example 1.3.5
This chess position is not very reasonable, since a good chess player would almost never promote two pawns to bishops of the same color!



FEN: 8/7k/8/8/8/7B/8/K4B1B w - - 0 1

Keeping this in mind, we want to make some educated guesses regarding the Cordel frequency CF(k) for the best moves rule and $k > 6$. An observation of the four established values for $k \in \{3, 4, 5, 6\}$ shows that the Cordel frequency decreases with increasing k, but more and more slightly. From this we conjecture that the Cordel frequency is monotonically decreasing in k for $k \leq 32$. We furthermore conjecture that the Cordel frequency CF(k), $k \leq 32$, for chess is always considerably greater than 50%.

The frequencies for the best move per piece and the best move per piece type rule ($k \leq 32$) are, as we presume, probably all between 50% and 60%. But without larger (endgame) tablebases these conjectures are difficult to prove.

We conclude by emphasizing that the Three Moves Law is a good rule of thumb at least in chess endgames. In particular we point out that the Cordel frequency for the best moves rule, which is probably the most important selection rule in practice, is considerably greater than 50%.

1.4 The Cordel Frequency for Arbitrary Optimization Problems

In Definition 1.2.10 we presented the generalized Cordel property as a property of moves in a given chess position. We generalize this concept by moving on from moves in chess to arbitrarily chosen feasible solutions x_1, x_2, and x_3 of a given optimization problem, where x_1 is the best and x_3 the worst of the three solutions.

Definition 1.4.1 (Generalized Cordel Property for Optimization Problems)
Let x_1, x_2, and x_3 be three arbitrary feasible solutions of a given optimization problem. Furthermore let x_1 be the best and x_3 be the worst of the three solutions. That means

(i) for minimization problems $f(x_1) \leq f(x_2) \leq f(x_3)$

(ii) and for maximization problems $f(x_1) \geq f(x_2) \geq f(x_3)$.

We say that the ordered triple (x_1, x_2, x_3) fulfills the ***generalized Cordel property (GeCoP)*** *if and only if*

$$d_1 := |f(x_1) - f(x_2)| \quad \geq \quad |f(x_2) - f(x_3)| =: d_2 \tag{GeCoP}$$

holds.

Note that this is in principle the same formula as for the definition of the generalized Cordel property for chess. The only difference is that we now consider three arbitrary feasible solutions x_1, x_2, x_3 instead of the three moves m_1, m_2, m_3. That is why we used the name (GeCoP) again.

Just as with chess, we define selection rules as follows.

Definition 1.4.2 (Selection Rule)
A rule S is called ***selection rule*** *if it chooses three feasible solutions x_1, x_2, and x_3, where x_1 is the best and x_3 is the worst of the three solutions.*

We propose the following two selection rules.

Definition 1.4.3 (Selection Rules for Optimization Problems)
Given an arbitrary optimization problem $\mathcal{P}$ we define the following selection rules.

(i) x_1, x_2, and x_3 are the three best solutions of $\mathcal{P}$. This rule is called the ***best solutions rule****.*

(ii) In Chapter 2 we introduce the penalty method which is a method to compute alternative solutions. The ***penalty selection rule*** *chooses an optimal solution x_1 and the first and second penalty alternatives as x_2 and x_3. This selection rule, introduced here very informally, is defined more precisely in the following Chapter 2.*

Note, that these two selection rules are not always applicable. In case of discrete optimization problems, the best solutions rule is always usable, but in case of continuous optimization problems there exists no second-best solution (see for example the continuous optimization problem from Example 2.3.9).

In practice, the penalty selection rule is of great interest because the three best solutions chosen by the best solutions rule are often too similar. In fact x_2 and x_3 are typically only micro mutations of x_1: For example when planning a route from Seattle to Chicago, x_2 might be exactly the same as x_1 except for a side trip through a parking lot in Minneapolis. This problem can be solved by using the penalty method which generates only true alternatives that have a good functional value but differ significantly from the optimal solution.

With these definitions of the general Cordel property and of selection rules for arbitrary optimization problems, we want to examine how often (GeCoP) holds for certain types of optimization problems. In addition to the selection rules, we also need a rule for how to generate random problem instances. Such generating rules cannot be generalized because they depend on the considered optimization problem. They will be defined later on, before we get to the experimental results.

Definition 1.4.4 (Cordel Frequency for Optimization Problems)
Consider a specific optimization problem T, a selection rule S, and a rule R for generating random instances of T.
*The **Cordel frequency CF** of (T, S, R) or shortly CF(T), when S and R are obvious, is the probability that* (GeCoP) *holds for a random instance, thus*

$$CF(T) := P(d_1 \geq d_2) = P(|f(x_1) - f(x_2)| \geq |f(x_2) - f(x_3)|) \; . \tag{1.2}$$

At first sight it may appear strange to weight $d_1 > d_2$ and $d_1 = d_2$ equally. One could think of $\overline{\text{CF}}(T) := P(d_1 > d_2) + \frac{1}{2}P(d_1 = d_2)$ as a better criterion, for example. But in fact the equality $d_1 = d_2$ is important for describing the Three Moves Law since in the case of three equally best solutions $f(m_1) = f(m_2) = f(m_3)$, and consequently $d_1 = d_2 = 0$, holds.

By defining d_1 as the absolute value of the difference $f(x_1) - f(x_2)$, the definition above can be applied to minimization as well as maximization problems. But the selection rule S must ensure that the functional values $f(x_1), f(x_2)$, and $f(x_3)$ are increasing (for minimization problems), or decreasing (for maximization problems), such that x_1 is always the best of the three solutions.

Remark 1.4.5
The Cordel frequency is defined as a probability. But for typical problems this probability cannot be computed, as far as we know, because of the computational difficult choosing candidates for a given selection rule. Thus the Cordel frequencies stated in this paper are in fact only relative frequencies and not the real probabilities. However, by the huge number of observed random instances the relative frequencies are good approximations for the real Cordel frequencies.

Chapter 2

The Penalty Method for General Sum-Type Problems

In his doctoral thesis [Sch 2003], S. Schwarz introduced the penalty method for specific discrete optimization problems called sum-type optimization problems. In this chapter we generalize his approach to more general optimization problems.

2.1 Motivation

If someone is planning a journey by car, he can use route planning software for computing the shortest route. But one must keep in mind that this software computes the best solution only based on average (expected) times or on pure lengths (cf. Subsection 3.3.1). The real travel times depend, for example, on the weather and traffic volume. Hence it could be helpful to have an alternative route at the back of one's mind which can be chosen if, for example, a traffic jam occurs on the originally planned route.

Interpreting this route planning problem as a shortest path problem in a weighted, directed graph, it is possible to compute the shortest and the second shortest route from the start to the finish. But the second shortest route has one main disadvantage. It is useless in practice since it is almost always just a *micro mutation* of the shortest route which hardly differs from the original optimal solution. For example the second best route might differ only in a side trip through a parking lot. Thus, this alternative solution will not help us avoiding a traffic jam on the main route.

We can formalize the example above slightly and get two requirements that a good alternative solution should fulfill.

1. An alternative solution should have a good functional value.
2. An alternative solution should differ considerably from the optimal solution because in case of a micro mutation the risk is high that a worsening of the original solution also implies a worsening of the similar alternative.

As shown in the routing example above, computing the second-best solution to a given optimization problem almost always provides micro mutations and thus no useful alternative solution.

The penalty method [Ber 2000, ABS 2002, Sch 2003, Sam 2005, Dör 2009] is an appropriate method to generate such alternative solutions.

2.2 The General Penalty Method

In his doctoral thesis [Sch 2003], Schwarz introduced the penalty method for sum-type optimization problems. Concretely he defined

Definition 2.2.1 (Σ-Type Optimization Problem, [Sch 2003, p. 2])
Let E be an arbitrary finite set and $S \subseteq \mathcal{P}(E)$ a subset of the power set of E. We call E ***the base set*** *and the elements of S* ***feasible subsets*** *of E. Let $w\colon E \to \mathbb{R}$ be a real-valued weight function on E. For every $B \in S$ we set $w(B) = \sum_{e\in B} w(e)$.*

We call the optimization problem $\min\limits_{B\in S} w(B)$ a ***sum-type optimization problem*** *and use the abbreviated name* ***Σ-type problem****.*

From now on the lifting of a function $w\colon E \to \mathbb{R}$ to subsets B of E as defined above will not be mentioned explicitly.

By substituting w with $-w$ every maximization problem can be formulated as a minimization problem, too. Thus the definition covers a large range of discrete and combinatorial optimization problems, such as

- the shortest path problem [AMO 1993, pp. 93-165]
- the minimum spanning tree problem [AMO 1993, pp. 510-542]
- the assignment problem [HK 2000, pp. 185-198]
- the traveling salesperson problem [Jun 2005, pp. 433-474]
- the binary knapsack problem [MT 1990, pp. 13-80]
- the sequence alignment problem [Les 2008, p. 243].

Let us have a look at the knapsack problem. Obviously the binary variant, which restricts the number of copies of each kind of item to one or zero, is a Σ-type problem. The bounded or unbounded knapsack problem (see problem definition in Appendix H on page 205 or [MT 1990, pp. 81-103]), where it is allowed to take more than one copy of each item, can be formulated as a Σ-type problem, too, by replicating items as often as they are allowed to be taken. We will illustrate this binary representation of the unbounded knapsack problem later in Example 2.2.9 on page 29.

In the same way the transportation problem with integer supplies s_i and demands d_i (see problem definition in Appendix H on page 206 or [BJS 1990, pp. 477-499]) can be formulated as Σ-type problem. But allowing the capacities to be real-valued transforms the former discrete optimization problem into a continuous problem which is not covered by Schwarz's definition of Σ-type problems. That is why we want to generalize the penalty method of Schwarz from Σ-type problems to a more general type of optimization problems. Actually, only minor effort is required since the definitions and theorems of Schwarz do not depend on his explicit definition of the Σ-type structure.

Definition 2.2.2 (General Penalty Method)
Assume a minimization problem of the form

$$\min_{B \in S} w'B$$

with a nonempty and bounded set of feasible solutions $S \subseteq \mathbb{R}^n$ and a real-valued weight-vector $w \in \mathbb{R}^n$. Further, let $0 \leq p \in \mathbb{R}^n$ denote a positive-real-valued penalty vector. For every $\varepsilon > 0$, let $B^{(\varepsilon)}$ be one of the optimal solutions of the problem

$$\min_{B \in S} w^{(\varepsilon)\prime} B \qquad \text{with} \qquad w^{(\varepsilon)} := w + \varepsilon \cdot p\,.$$

Additionally we define the solution $B^{(\infty)}$ as a solution with minimal value $p'B$, and among all such solutions with a minimal value $w'B$.

$$\begin{aligned} p'B^{(\infty)} &\leq p'B && \text{for all } B \in S \\ w'B^{(\infty)} &\leq w'B && \text{for all } B \in S \text{ with } p\,(B) = \min_{B \in S} p'B \end{aligned}$$

We write

$$B^{(\infty)} = \operatorname{lex}\min_{B \in S} \,(p'B, w'B)$$

where lex min *stands for lexicographical minimization as described above.*

*We call $B^{(\varepsilon)}$ an **ε-penalty alternative** or **ε-alternative** and $w^{(\varepsilon)}$ the **penalized weight**.*

We now introduce the following notation.

Definition 2.2.3 (Weight, Penalized Part, and Penalized Value)
*Let $B \in S$ be a feasible solution. Then we call $w(B) := w'B$ the **weight**, $p(B) := p'B$ the **penalized part**, and $f_\varepsilon(B) := w^{(\varepsilon)\prime} B$ the **penalized value or penalized weight** of B.*

We clarify the two definitions in several examples. We start with Example 2.2.5 where we compute ε-penalty alternatives for a shortest path problem. While shortest path problems are already covered by Schwarz's definition of the penalty method for Σ-type problems, the following Example 2.2.6 deals with the transportation problem with real-valued supplies and demands.

Remark 2.2.4

In principle it is not necessary to limit the definition to vectors w and p. The penalty method could be applied to arbitrary optimization problems where we are able to define a penalty function $p\colon S \to \mathbb{R}$. In that case the ε-penalty alternative would be an optimal solution of the optimization problem

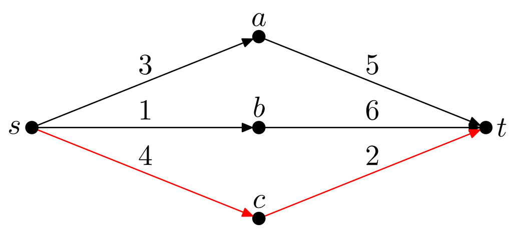
$$\min_{B \in S} w(B) + \varepsilon \cdot p(B)\,,$$

where $w\colon S \to \mathbb{R}$ and $p\colon S \to \mathbb{R}$ are arbitrary real-valued weight and penalty functions. Here, we chose to study only the case of linear functions, because in that case there exists a canonical choice of the penalty vector p, which we are going to present in Definition 2.2.7 on page 28.

Furthermore the assumption that $S \subseteq \mathbb{R}^n$ is bounded is not always essential. The advantage of this assumption is that it ensures the existence of an optimal solution for each considered optimization problem as long as S is nonempty.

Example 2.2.5

We consider the following simple shortest path problem where s is the start and t is the target.



The red path $s - b - t$ is the optimal solution with $w(s - b - t) = 6$.

To convert this shortest path problem into an optimization problem of the form $\min_{B \in S} w'B$ we introduce the weight vector[1]

$$w = [\underbrace{3}_{=w(s-a)}, \underbrace{5}_{=w(a-t)}, \underbrace{1}_{=w(s-b)}, \underbrace{6}_{=w(b-t)}, \underbrace{4}_{=w(s-c)}, \underbrace{2}_{=w(c-t)}]$$

representing the weights of the edges $[(s-a), (a-t), (s-b), (b-t), (s-c), (c-t)]$. Furthermore, we have to specify the set of feasible solutions $S \subseteq \{0,1\}^6$, which shall include the only three feasible paths from s to t namely $s-a-t$, $s-b-t$, and $s-c-t$. Thus we set

$$S = \{\underbrace{[1,1,0,0,0,0]}_{=s-a-t}, \underbrace{[0,0,1,1,0,0]}_{=s-b-t}, \underbrace{[0,0,0,0,1,1]}_{=s-c-t}\}\,,$$

where 1 means that the edge is used in the path and 0 means that the edge is not used.

[1] Note that we write vectors as row vectors with square brackets according to Matlab syntax [Matlab 2008].

Lastly, we need a penalty vector p*. For example* p *could penalize all edges in the optimal solution* $[0,0,0,0,1,1]$*, say* $p = [0,0,0,0,1.5,0.5]$*. This vector* $p \geq 0$ *can be chosen without any further restrictions. Our hidden agenda was to penalize all edges used in the optimal solution, but the concrete values* $p(s-c) = 1.5$ *and* $p(c-t) = 0.5$ *were chosen arbitrarily but positive.*

To compute an ε*-penalty alternative we consider the penalized weight vector* $w^{(\varepsilon)}$ *which is*

$$\begin{aligned} w^{(\varepsilon)} &= w + \varepsilon \cdot p = [3,\ 5,\ 1,\ 6,\ 4,\ 2] + \varepsilon \cdot [0,\ 0,\ 0,\ 0,\ 1.5,\ 0.5] \\ &= [3,\ 5,\ 1,\ 6,\ 4 + 1.5 \cdot \varepsilon,\ 2 + 0.5 \cdot \varepsilon]\,. \end{aligned}$$

Hence, p *penalizes the edges* $s-c$ *and* $c-t$ *by increasing their weights. Thereby the edge* $s-c$ *is penalized stronger than the edge* $c-t$ *because of*

$$p(s-c) = 1.5 > 0.5 = p(c-t)\,.$$

For $\boldsymbol{\varepsilon = 0.2}$ *we get*

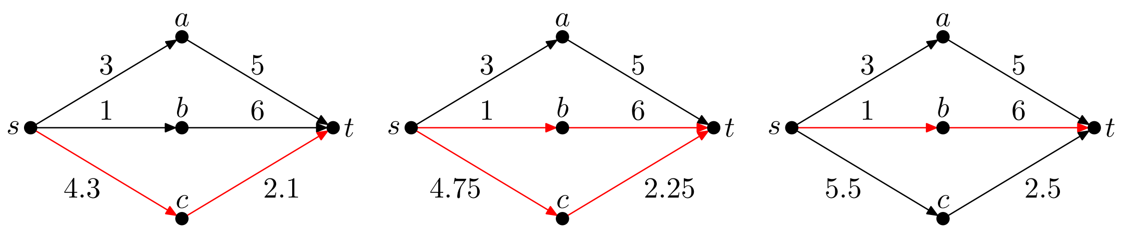
$$\begin{aligned} w^{(0.2)} &= w + 0.2 \cdot p = [3,\ 5,\ 1,\ 6,\ 4,\ 2] + 0.2 \cdot [0,\ 0,\ 0,\ 0,\ 1.5,\ 0.5] \\ &= [3,\ 5,\ 1,\ 6,\ 4.3,\ 2.1] \end{aligned}$$

as penalized weight vector and the penalized graph shown in Figure 2.2.1-(a). As we can see, $s-c-t$ *is still the shortest path from* s *to* t*. Thus,* $s-c-t$ *is not only the optimal solution but also the* 0.2*-penalty alternative.*

Now we increase ε *to* $\boldsymbol{\varepsilon = 0.5}$ *and get the following penalized weight vector and the graph shown in Figure 2.2.1-(b).*

$$\begin{aligned} w^{(0.5)} &= w + 1 \cdot p = [3,\ 5,\ 1,\ 6,\ 4,\ 2] + 0.5 \cdot [0,\ 0,\ 0,\ 0,\ 1.5,\ 0.5] \\ &= [3,\ 5,\ 1,\ 6,\ 4.75,\ 2.25] \end{aligned}$$

Therewith $s-b-t$ *and* $s-c-t$ *have the same weights now. Thus, both paths are* 0.5*-penalty alternatives.*



(a) $\varepsilon = 0.2$ **(b)** $\varepsilon = 0.5$ **(c)** $\varepsilon = 1$

Figure 2.2.1: Penalized graphs and penalty alternatives (red marked paths) for different penalty parameters ε.

Furthermore, we consider $\boldsymbol{\varepsilon = 1}$ *with*

$$\begin{aligned} w^{(1)} &= w + 1 \cdot p = [3,\ 5,\ 1,\ 6,\ 4,\ 2] + 1 \cdot [0,\ 0,\ 0,\ 0,\ 1.5,\ 0.5] \\ &= [3,\ 5,\ 1,\ 6,\ 5.5,\ 2.5] \end{aligned}$$

and the graph shown in Figure 2.2.1-(c). Here, path $s-c-t$ *has the penalized weight* $5.5+2.5 = 8$ *and path* $s-b-t$ *has the penalized weight* $1+6 = 7$. *Consequently* $s-b-t$ *is the* 1*-penalty alternative and* $s-c-t$ *is no penalty alternative any longer.*

Finally we consider $\varepsilon = \infty$. *The* ∞*-penalty alternative was defined as*

$$B^{(\infty)} = \operatorname*{lex\,min}_{B \in S} (p'B, w'B) = \operatorname*{lex\,min}_{B \in S} (p(B), w(B))$$

which means that $B^{(\infty)}$ *has a minimal value* $p(B) := p'B$. *We compute*

$$\begin{aligned} &\textit{for } s-a-t\textit{:} \quad p(s-a-t) = [0,\ 0,\ 0,\ 0,\ 1.5,\ 0.5]' \cdot [1,\ 1,\ 0,\ 0,\ 0,\ 0] = 0\,, \\ &\textit{for } s-b-t\textit{:} \quad p(s-b-t) = [0,\ 0,\ 0,\ 0,\ 1.5,\ 0.5]' \cdot [0,\ 0,\ 1,\ 1,\ 0,\ 0] = 0\,, \quad \textit{and} \\ &\textit{for } s-c-t\textit{:} \quad p(s-c-t) = [0,\ 0,\ 0,\ 0,\ 1.5,\ 0.5]' \cdot [0,\ 0,\ 0,\ 0,\ 1,\ 1] = 2\,. \end{aligned}$$

Hence, both $s-a-t$ *and* $s-c-t$ *have a minimal value* $p(B)$. *From these two paths we choose the solution with a minimal value* $w(B)$. *Because*

$$w\,(s-a-t) = 8 > 7 = w\,(s-b-t)$$

we get $B^{(\infty)} = s-b-t$. *This completes Example 2.2.5.*

After this illustrative example, the next example deals with the transportation problem with real valued supplies and demands. This real valued problem was not covered by Schwarz's penalty method.

Example 2.2.6
Consider two suppliers which supply $s_1 = 0.45$ *and* $s_2 = 0.55$ *units of a particular good. On the other hand side we have three recipients which require* $d_1 = 0.1$, $d_2 = 0.3$, *and* $d_3 = 0.6$ *units of this good. For the transportation of one unit from supplier* i *to recipient* j *transportation costs* c_{ij} *arise. Let*

$$C = \begin{bmatrix} 0.25 & 0.75 & 0.3 \\ 0.4 & 0.35 & 0.2 \end{bmatrix}$$

be the matrix of transportation costs.

A clear presentation is given by the following table.

	$d_1 = 0.1$	$d_2 = 0.3$	$d_3 = 0.6$
$s_1 = 0.45$	0.25	0.75	0.30
$s_2 = 0.55$	0.40	0.35	0.20

The transportation problem is now to satisfy the demands without exceeding the supplies and with minimal transportation costs. Hence, we get the following optimization problem.

$$\begin{aligned}
\min_{x\in R^{2\times 3}} \quad & 0.25x_{11} + 0.75x_{11} + 0.3x_{11} + 0.4x_{11} + 0.35x_{11} + 0.2x_{11} \\
s.t. \quad & x_{11} + x_{12} + x_{13} \leq s_1 = 0.45 \\
& x_{21} + x_{22} + x_{23} \leq s_2 = 0.55 \\
& x_{11} + x_{21} \geq d_1 = 0.1 \\
& x_{12} + x_{22} \geq d_2 = 0.3 \\
& x_{13} + x_{23} \geq d_3 = 0.6 \\
& x_{ij} \geq 0 \qquad \textit{for } i = 1,2 \textit{ and } j = 1,2,3
\end{aligned} \tag{2.1}$$

By writing the matrices as vectors we get the following representation.

$$\begin{aligned}
\min_{x\in \mathbb{R}^{6}} \quad & [0.25,\ 0.75,\ 0.3,\ 0.4,\ 0.35,\ 0.2]' \cdot x \\
s.t. \quad & x_1 + x_2 + x_3 \leq s_1 = 0.45 \\
& x_4 + x_5 + x_6 \leq s_2 = 0.55 \\
& x_1 + x_4 \geq d_1 = 0.1 \\
& x_2 + x_5 \geq d_2 = 0.3 \\
& x_3 + x_6 \geq d_3 = 0.6 \\
& x_i \geq 0 \qquad \textit{for } i = 1, \ldots, 6
\end{aligned} \tag{2.2}$$

For better readability we continue using the matrix representations of the optimization as in (2.1) *instead of the vector representation in* (2.2). *For this transportation problem*

$$B^{(0)} = \begin{bmatrix} 0.1 & 0 & 0.35 \\ 0 & 0.3 & 0.25 \end{bmatrix}$$

is the optimal solution. As in Example 2.2.5 before, we need to specify either a penalty vector or a penalty matrix. For penalty matrix

$$P = \begin{bmatrix} 0.025 & 0 & 0.105 \\ 0 & 0.105 & 0.05 \end{bmatrix}$$

we compute the penalty alternative for $\boldsymbol{\varepsilon = 2}$. *Therefore we consider the penalized transportation costs.*

$$\begin{aligned}
C^{(2)} := C + 2 \cdot P &= \begin{bmatrix} 0.25 & 0.75 & 0.3 \\ 0.4 & 0.35 & 0.2 \end{bmatrix} + 2 \cdot \begin{bmatrix} 0.025 & 0 & 0.105 \\ 0 & 0.105 & 0.05 \end{bmatrix} \\
&= \begin{bmatrix} 0.3 & 0.75 & 0.51 \\ 0.4 & 0.56 & 0.30 \end{bmatrix}
\end{aligned}$$

The penalized transportation problem provides the optimal solution

$$B^{(2)} = \begin{bmatrix} 0.1 & 0.3 & 0.05 \\ 0 & 0 & 0.55 \end{bmatrix} \neq \begin{bmatrix} 0.1 & 0 & 0.35 \\ 0 & 0.3 & 0.25 \end{bmatrix} = B^{(0)}$$

which is the 2*-penalty alternative. This completes Example 2.2.6.*

As we saw in the Examples 2.2.5 and 2.2.6, Definition 2.2.2 of the penalty method is rather general and needs to be specified for practical purposes because one needs to choose a practical penalty vector p. We recommend the following canonical penalty vector.

Definition 2.2.7 (Canonical Penalty Vector)
Assume $S \subseteq \mathbb{R}^n_+$ and let $B^{(0)}$ denote an optimal solution of the original optimization problem.

If all nonzero weights $w_i \neq 0, i \in \{1, \ldots, n\}$ have the same sign (all positive or all negative), we define the ***canonical penalty vector*** *p by*

$$\begin{aligned} p_i &= \begin{cases} |w_i|\, B_i^{(0)} & \textit{iff } B_i^{(0)} \neq 0, \textit{ i.e. element } i \textit{ is used in } B^{(0)} \\ 0 & \textit{iff } B_i^{(0)} = 0, \textit{ i.e. element } i \textit{ is not used in } B^{(0)} \end{cases} \\ &= |w_i| \cdot B_i^{(0)}. \end{aligned} \tag{2.3}$$

This means that when $w \geq 0$ each element i used in $B^{(0)}$ is punished by its weight w_i multiplied by the frequency or amount (for non-integer solutions) $B_i^{(0)}$. If there exist $i, j \in \{1, \ldots, n\}$ with $\operatorname{sgn}(w_i) = +1$ and $\operatorname{sgn}(w_j) = -1$, then we do not see a straight forward definition of a canonical penalty vector.

Example 2.2.8 (Continuation of Example 2.2.6)
In the transportation problem in Example 2.2.6 we had the following transportation costs C, optimal solution $B^{(0)}$ and penalty matrix P.

$$C = \begin{bmatrix} 0.25 & 0.75 & 0.3 \\ 0.4 & 0.35 & 0.2 \end{bmatrix}, \quad B^{(0)} = \begin{bmatrix} 0.1 & 0 & 0.35 \\ 0 & 0.3 & 0.25 \end{bmatrix}, \quad P = \begin{bmatrix} 0.025 & 0 & 0.105 \\ 0 & 0.105 & 0.05 \end{bmatrix}$$

In this example we already used the canonical penalty vector (or in this case penalty matrix) as the following computation shows.

$$\begin{bmatrix} 0.25 \cdot 0.1 & 0.75 \cdot 0 & 0.3 \cdot 0.35 \\ 0.4 \cdot 0 & 0.35 \cdot 0.3 & 0.2 \cdot 0.25 \end{bmatrix} = \begin{bmatrix} 0.025 & 0 & 0.105 \\ 0 & 0.105 & 0.05 \end{bmatrix} = P$$

In Appendix G , starting on page 199, one finds two longer Examples. In Example G.1 **all** penalty alternatives for a shortest path problem in an **undirected** graph are computed. Example G.2 deals with the same instance represented as a **directed** graph. We show that the canonical penalty vector provides different penalty alternatives for these two equivalent representations of the same graph. Hence, one must first think carefully about whether the canonical penalty vector is the right choice for concrete optimization problems.

The following three-page example deals with a similar problem which can occur in case of a knapsack problem.

Example 2.2.9

We consider the following unbounded knapsack problem (cf. Appendix H on page 205)

$$\max_{x\in\mathbb{N}^n} v'x \quad s.t. \quad w'x \le C \quad \Leftrightarrow \quad \min_{x\in\mathbb{N}^n} -v'x \quad s.t. \quad w'x \le C$$

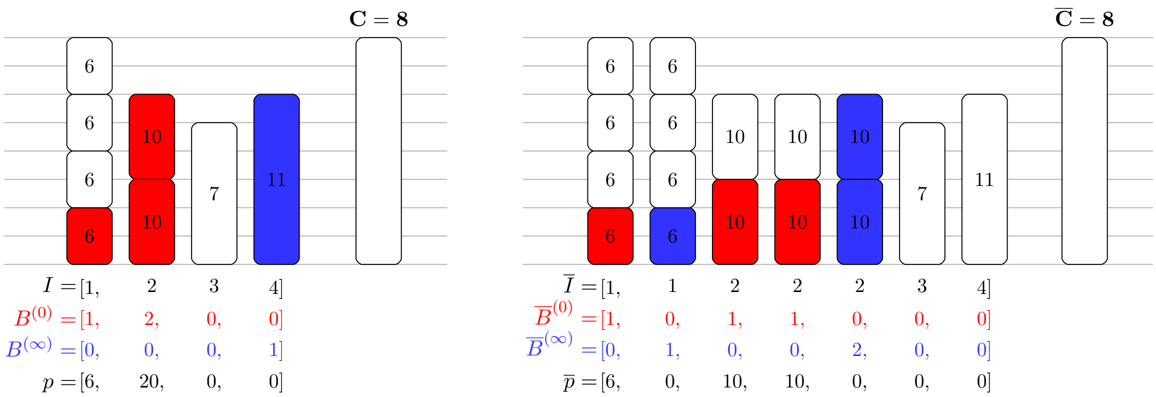
with

$$\begin{array}{llll} \textit{item values} & v=[6,10,7,11], & \textit{knapsack capacity} & C=8 \\ \textit{item weights} & w=[2,3,5,6], & \textit{number of items} & n=4. \end{array} \tag{UKP1}$$

In an unbounded knapsack problem each item can be packed into the knapsack as often as it fits into it. This implies that it is possible to copy items without changing the optimization problem. For example

$$\begin{array}{ll} \overline{v}=[6,6,\ 10,10,10,\ 7,\ 11], & \overline{C}=8 \\ \overline{w}=[2,2,\ 3,3,3,\ 5,\ 6], & \overline{n}=4. \end{array} \tag{UKP2}$$

represents the same optimization problem as (UKP1)*. These two representations of the same unbounded knapsack problem are illustrated in Figure 2.2.2 (a) and (b) below. The weights of the items and the knapsack capacity are represented by the heights of the boxes. The numbers in the boxes represent the item values.*



(a) Representation (UKP1) without multiple items.

(b) Representation (UKP2) with multiple items.

Figure 2.2.2: Two representations (UKP1) and (UKP2) of the same unbounded knapsack problem with item number vectors I and $\overline{I}$, optimal solutions $B^{(0)}$ and $\overline{B}^{(0)}$ marked in red, ∞-penalty alternatives $B^{(\infty)}$ and $\overline{B}^{(\infty)}$ marked in blue and penalty vectors p and $\overline{p}$.

Now we enumerate the items in (UKP1) *which shall be represented by the following item-number-vector* $I = [1,2,3,4]$*. We can do the same with problem* (UKP2) *which contains copies of some items. With the item number vector*

$$\overline{I} = [1,1,\ 2,2,2,\ 3,\ 4]$$

it is possible to figure out which of the original items is represented by $(\overline{v}_i, \overline{w}_i)$*. For example item* $j=3$ *in* (UKP2) *with value* $\overline{v}_j = 10$ *and weight* $\overline{w}_j = 3$ *is the same* *as item* $\overline{I}_j = 2$ *in* (UKP1). *Thus for* $I_i = \overline{I}_j$, *item* i *in problem* (UKP1) *and item* j *in problem* (UKP2) *are the same. The item index vector* $\overline{I}$ *also helps to transform feasible solutions of* (UKP2) *into feasible solutions of* (UKP1). *So, let* $\overline{B} \in \mathbb{N}^{\overline{n}}$ *be a feasible solution of* (UKP2). *Then the* i*-th element of the corresponding feasible solution* $B \in \mathbb{N}^n$ *is the sum of all components of* $\overline{B}$ *representing item* i

$$B_i = \sum_{j:\,\overline{I}_j = i} \overline{B}_j \,. \tag{2.4}$$

For example observe $\overline{B}^{(0)} = [1, 0,\ 1, 1, 0,\ 0,\ 0]$ *with functional value* $-\overline{v}\left(\overline{B}^{(0)}\right) = -26$. *This is an optimal solution to* (UKP2). *We compute*

$$\begin{aligned}
B_1^{(0)} &= \sum_{j:\,\overline{I}_j=1} \overline{B}_j^{(0)} = \sum_{j\in\{1,2\}} \overline{B}_j^{(0)} = 1\,,\\
B_2^{(0)} &= \sum_{j:\,\overline{I}_j=2} \overline{B}_j^{(0)} = \sum_{j\in\{3,4,5\}} \overline{B}_j^{(0)} = 2\,,\\
B_3^{(0)} &= \sum_{j:\,\overline{I}_j=3} \overline{B}_j^{(0)} = \overline{B}_6^{(0)} = 0\,,\\
B_4^{(0)} &= \sum_{j:\,\overline{I}_j=4} \overline{B}_j^{(0)} = \overline{B}_7^{(0)} = 0\,.
\end{aligned}$$

This means that the optimal solution $\overline{B}^{(0)} = [1, 0,\ 1, 1, 0,\ 0,\ 0]$ *to* (UKP2) *corresponds to the optimal solution* $B^{(0)} = [1, 2, 0, 0]$ *to* (UKP1) *with* $-v\left(B^{(0)}\right) = -26$. *Note that* $B^{(0)}$ *is the uniquely determined optimal solution to* (UKP1) *while the optimal solution to* (UKP2) *is not uniquely determined. In fact, each* $\overline{B} \in \mathbb{N}^{\overline{n}}$ *with*

$$\sum_{j:\,\overline{I}_j=1} \overline{B}_j = B_1^{(0)} = 1 \qquad \sum_{j:\,\overline{I}_j=2} \overline{B}_j = B_2^{(0)} = 2$$
$$\sum_{j:\,\overline{I}_j=3} \overline{B}_j = B_3^{(0)} = 0 \qquad \sum_{j:\,\overline{I}_j=4} \overline{B}_j = B_4^{(0)} = 0$$

is an optimal solution to (UKP2) *since* (UKP1) *and* (UKP2) *are representations of the same unbounded knapsack problem. The two chosen optimal solutions* $B^{(0)}$ *and* $\overline{B}^{(0)}$ *are shown in Figure 2.2.2 in red.*

These optimal solutions provide the canonical penalty vectors p *for* (UKP1) *and* $\overline{p}$ *for* (UKP2).

$$\begin{aligned}
p &= \left[v_1 \cdot B_1^{(0)},\ v_2 \cdot B_2^{(0)},\ v_3 \cdot B_3^{(0)},\ v_4 \cdot B_4^{(0)}\right]\\
&= [1 \cdot 6, 2 \cdot 10, 0, 0] = [6, 20, 0, 0]\\
\overline{p} &= \left[\overline{v}_1 \cdot \overline{B}_1^{(0)},\ \overline{v}_2 \cdot \overline{B}_2^{(0)},\ \overline{v}_3 \cdot \overline{B}_3^{(0)},\ \overline{v}_4 \cdot \overline{B}_4^{(0)},\ \overline{v}_5 \cdot \overline{B}_5^{(0)},\ \overline{v}_6 \cdot \overline{B}_6^{(0)},\ \overline{v}_7 \cdot \overline{B}_7^{(0)},\ \right]\\
&= [1 \cdot 6,\ 0,\ 1 \cdot 10,\ 1 \cdot 10,\ 0,\ 0,\ 0] = [6, 0,\ 10, 10, 0,\ 0,\ 0]
\end{aligned}$$

Note that in the case of a knapsack problem, the vector w *does no longer represents the objective function. Here, the vector* $-v$ *plays the role of* w. *That is why we inserted*

$-v$ into Formula (2.3) on page 28 which defines the canonical penalty vector.

Now we compute ∞-penalty alternatives which are solutions of the minimization problem

$$\operatorname*{lex\,min}_{B \in S} \left(p\left(B\right), -v\left(B\right)\right) \qquad \textit{for } (\text{UKP1})$$
$$\textit{or} \qquad \operatorname*{lex\,min}_{\overline{B} \in \overline{S}} \left(\overline{p}\left(\overline{B}\right), -\overline{v}\left(\overline{B}\right)\right) \qquad \textit{for } (\text{UKP2}).$$

In the compact representation (UKP1) with penalty vector p, the best solution to this problem ($B^{(\infty)}$) uses none of the items used in $B^{(0)}$. Thus we get $B^{(\infty)} = [0,0,0,1]$ with

$$-v\left(B^{(\infty)}\right) = -11 \qquad \textit{and} \qquad p\left(B^{(\infty)}\right) = 0.$$

In representation (UKP2) with penalty vector $\overline{p}$ there exists a second optimal solution with penalized part 0: $\overline{B}^{(\infty)} = [0,1,\ 0,0,2,\ 0,\ 0]$.

$$-\overline{v}\left(\overline{B}^{(\infty)}\right) = -26 \qquad \textit{and} \qquad \overline{p}\left(\overline{B}^{(\infty)}\right) = 0.$$

The two chosen ∞-alternatives are represented in blue in Figure 2.2.2.

Because

$$-\overline{v}\left(\overline{B}^{(\infty)}\right) = 26 = -\overline{v}\left(\overline{B}^{(0)}\right) = \min_{\overline{B} \in \overline{S}} -\overline{v}\left(\overline{B}\right) \tag{2.5}$$

we get

$$-\overline{v}\left(\cdot\overline{B}^{(\infty)}\right) + \varepsilon \cdot \underbrace{\overline{p}\left(\overline{B}^{(\infty)}\right)}_{=0} \overset{(2.5)}{\leq} -\overline{v}\left(\overline{B}\right) \leq -\overline{v}\left(\overline{B}\right) + \varepsilon \cdot \underbrace{\overline{p}\left(\overline{B}\right)}_{\geq 0}$$

for all $\overline{B} \in \overline{S}$ and all $\varepsilon \geq 0$. Hence, $\overline{B}^{(\infty)}$ is an ε-penalty alternative for all $\varepsilon \geq 0$. Actually $\overline{B}^{(\infty)}$ is uniquely determined ε-penalty alternative for each $\varepsilon > 0$.

If we take a closer look at $\overline{B}^{(\infty)}$, we see that it is only another representation of the optimal solution, because the corresponding 4-item solution is again $[1,2,0,0]$ which is not a ∞-penalty alternative for (UKP1). Thus the penalty method with the canonical penalty vector provides different alternatives for the two different representations of the same optimization problem.

One can fix this problem by penalizing every copy of the original items 1 and 2 in the second representation. The penalty vector

$$\overline{p}^{(2)} = [6,6,\ 20,20,20,\ 0,\ 0]$$

for (UKP1) provides the same penalty alternatives as p provides in (UKP1). This concludes Example 2.2.9.

In conclusion we give the following recommendation:
To use the canonical penalty vector one should think about a reasonable representation of the considered optimization problem. In case of the knapsack problem (cf. Example 2.2.9 starting on page 29) one should make sure that there are no multiple copies of the same item. This is also the reason why we do not represent an unbounded knapsack problem as a binary knapsack problem and apply Schwarz's penalty method to it, since this would lead to strange penalty alternatives. In case of the shortest path problem (cf. Examples G.1 and G.2 starting on page 199) one should think about whether a directed or an undirected graph is more reasonable.

Actually, in most cases the canonical penalty vector provides useful results, if one chooses a reasonable representation of the considered optimization problem. But sometimes one is interested in a more individual penalty function.

As mentioned before, the penalty method from Definition 2.2.2 is a generalization of the penalty method for Σ-type problems from Schwarz. The following lemma shows that this generalization still contains the original penalty method from Schwarz.

Lemma 2.2.10
Consider a Σ-type problem in the sense of Schwarz and the canonical penalty vector p like in (2.3).
Then the penalty method from Definition 2.2.2 is the same as the penalty method from Definition 2.2.1 in [Sch 2003, pp. 7-8]. That means both methods provide the same penalty alternatives.

Proof. Consider a Σ-type problem in the sense of Schwarz

$$\min_{B \in S} \sum_{e \in B} w(e)$$

with a finite base set E ($|E| = n$), a set of feasible subsets $S \subseteq \mathcal{P}(E)$, and a weight function $w\colon E \to \mathbb{R}$. Let the elements of E be sorted and called $e_1, \ldots, e_n$.

Like in the Examples 2.2.5, G.1, and G.2 on the pages 24, 199, and 201, respectively, we can represent each element $B \in S$ as a vector $\overline{B} \in \{0,1\}^n$ by

$$\overline{B}_i = \begin{cases} 1, & \text{iff } e_i \in B \\ 0, & \text{iff } e_i \notin B \end{cases} \quad \text{for } i = 1, \ldots, n\,.$$

Hence the set of feasible subsets $S \subseteq \mathcal{P}(E)$ can be transformed to the set of feasible solutions $\overline{S} \subseteq \mathbb{R}^n$ by collecting the vector representations $\overline{B}$ of all $B \in S$. In almost the same manner we can write the weight function w as a vector $\overline{w} \in \mathbb{R}^n$ by

$$\overline{w}_i = w\left(e_i\right) \quad \text{for } i = 1, \ldots, n\,. \tag{2.6}$$

This representation with $\overline{S}$ and $\overline{w}$ under (2.6) is bijective and thereby invertible.

Obviously this is only a change in representation. An optimal solution $\overline{B}^{(0)}$ to

$$\min_{\overline{B}\in\overline{S}} \overline{w}\left(\overline{B}\right)$$

is in fact the vector representation of a corresponding optimal solution $B^{(0)}$ to the original Σ-type problem

$$\min_{B\in S} \sum_{e\in B} w(e)\,.$$

Now we can apply the penalty method from Definition 2.2.2 to $\overline{w}$ and $\overline{S}$. With the canonical penalty vector p, solution $\overline{B}^{(\varepsilon)}$ arises as an optimal solution of $\min\limits_{\overline{B}\in\overline{S}} f_\varepsilon\left(\overline{B}\right)$ with

$$\begin{aligned} f_\varepsilon\left(\overline{B}\right) &= \overline{w}^{(\varepsilon)\prime}\overline{B} = (\overline{w}+\varepsilon\cdot p)'\overline{B} = \overline{w}'\overline{B} + \varepsilon p'\overline{B} \\ &= \overline{w}'\overline{B} + \varepsilon \sum_{i=1}^{n} \overline{w}_i \cdot \overline{B}_i^{(0)} \cdot \overline{B}_i = \sum_{i:\ B_i=1} \overline{w}_i + \varepsilon \sum_{i:\ B_i=B_i^{(0)}=1} \overline{w}_i \\ &= \sum_{e\in B} w\left(e\right) + \varepsilon \sum_{e\in B\cap B^{(0)}} w\left(e\right) = w(B) + \varepsilon\, w\left(B\cap B^{(0)}\right)\,. \end{aligned}$$

Thus $\overline{B}^{(\varepsilon)}$ is the vector-representation of

$$B^{(\varepsilon)} := \arg\min_{B\in S} \left[w(B) + \varepsilon\, w\left(B\cap B^{(0)}\right)\right],$$

which is the ε-penalty alternative according to Schwarz's penalty method. ■

Remark 2.2.11
In Lemma 2.2.10 we showed just that our penalty method from Definition 2.2.2 is the same as the penalty method from Definition 2.2.1 in [Sch 2003, pp. 7-8] if we use the canonical penalty vector.
But Schwarz stated also a second penalty method which allows the use of an arbitrary penalty vector p. In analogy to the previous Lemma 2.2.10 one can show that even Schwarz's more general penalty method from Definition 2.2.2 [Sch 2003, pp. 8-9] is covered by our definition.

Guided by Schwarz's notation, we introduce the following notation.

Definition 2.2.12 (General Σ-Type Problem)
A minimization problem of the form

$$\min_{B\in S} w'B$$

with the set of feasible solutions $S\subseteq\mathbb{R}^n$ *and a real-valued weight-vector* $w\in\mathbb{R}^n$ *is called **general sum-type problem** or **general Σ-type problem**.*

These are exactly the optimization problems for which we introduced the penalty method in Definition 2.2.2. Beside Σ-type problems, the bounded and unbounded knapsack problem, the real valued transportation problem and real valued network flow problems examples of general Σ-type problems. These three optimization problems are defined and explained in Appendix H starting on page 205.

2.3 Properties of the Penalty Method

We start our observations with a geometric interpretation of the penalty method. Therefore we consider the (w, p) diagram where every feasible solution $B \in S$ is represented by the ordered pair $(w(B), p(B))$. For the related bi-objective optimization problem

$$\min_{B \in S} (w(B), p(B))$$

we introduce the following notation.

Definition 2.3.1 (Efficient and Supported Points)
*A feasible solution $B \in S$ is **dominated** by another feasible solution $D \in S$, if*

$$w(D) \leq w(B) \quad \text{as well as} \quad p(D) \leq p(B)$$

hold and where at least one of the inequalities is fulfilled strictly.

*A feasible solution $B \in S$ is called **efficient** or **Pareto optimal** if no $D \in S \backslash \{B\}$ exists that dominates B.*

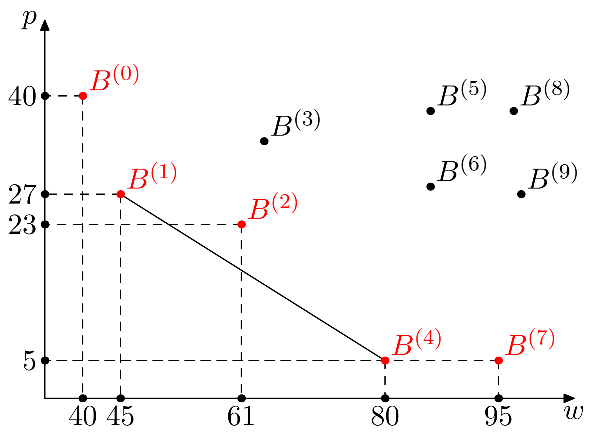
*Furthermore, we call $B \in S$ **supported** if there is some $\lambda \in \mathbb{R}^2_{\geq 0}$ such that B is an optimal solution of*

$$\min_{B \in S} \lambda_1 w(B) + \lambda_2 p(B) .$$

We illustrate the definitions above in the following Example 2.3.2.

Example 2.3.2
Consider an optimization problem with the following ten feasible solutions.

feasible solution B	$w(B)$	$p(B)$
$B^{(0)}$	40	40
$B^{(1)}$	45	27
$B^{(2)}$	61	23
$B^{(3)}$	64	34
$B^{(4)}$	80	5
$B^{(5)}$	86	38
$B^{(6)}$	86	28
$B^{(7)}$	95	5
$B^{(8)}$	97	38
$B^{(9)}$	98	27



The red-marked feasible solutions $B^{(0)}, B^{(1)}, B^{(2)}, B^{(4)}$ and $B^{(7)}$ are the efficient points. Except for $B^{(2)}$ these are also the supported points. But since $B^{(2)}$ is above the line segment from $B^{(1)}$ to $B^{(4)}$ there is no objective function $\lambda_1 w(B)+\lambda_2 p(B)$ with $\lambda_1, \lambda_2 \geq 0$ for which $B^{(2)}$ is an optimal solution. Thus, $B^{(2)}$ is not a supported point.

We make the following important observation which is essentially known from multi-objective optimization theory. But we have to be a little careful with the ∞-alternative $B^{(\infty)}$.

Lemma 2.3.3 (Penalty Alternatives and the Concept of Supported Points) *Given a general Σ-type problem, for every feasible solution $P \in S$ the following two statements are equivalent:*

(i) There exists an $\varepsilon \geq 0$ such that P is an ε-penalty alternative.

(ii) There exists a vector $\lambda \in \mathbb{R}^2$ with $\lambda_1 > 0$ and $\lambda_2 \geq 0$ such that P is an optimal solution of

$$\min_{B \in S} \lambda_1 w(B) + \lambda_2 p(B) .$$

Hence, in the (w, p) diagram each penalty alternative is a supported point.

Proof. The proof is complex for $\varepsilon = \infty$ but not difficult.

(ii) $\Rightarrow$ (i): Let $P \in S$ be an optimal solution to $\min_{B \in S} \lambda_1 w(B) + \lambda_2 p(B)$ with $\lambda_1 > 0$ and $\lambda_2 \geq 0$. Then P is an $\varepsilon = \frac{\lambda_2}{\lambda_1}$-penalty alternative, since

$$\begin{aligned} & \lambda_1 w(P) + \lambda_2 p(P) \leq \lambda_1 w(B) + \lambda_2 p(B) && \text{for all } B \in S \\ \Leftrightarrow \quad & w(P) + \frac{\lambda_2}{\lambda_1} p(P) \leq w(B) + \frac{\lambda_2}{\lambda_1} p(B) && \text{for all } B \in S \end{aligned}$$

holds.

(i) $\Rightarrow$ (ii): In the case $\mathbf{0 \leq \varepsilon < \infty}$ we have

$$w\left(B^{(\varepsilon)}\right) + \varepsilon p\left(B^{(\varepsilon)}\right) \leq w(B) + \varepsilon p(B)$$

for all $B \in S$. Thus $B^{(\varepsilon)}$ is a supported point because it minimizes the objective function $\lambda_1 w(B) + \lambda_2 p(B)$ with $\lambda_1 = 1$ and $\lambda_2 = \varepsilon$.

Now we consider the case $\boldsymbol{\varepsilon = \infty}$ which is a little bit more complex but still not difficult. Remember that $B^{(\infty)}$ was defined as

$$\begin{aligned} & B^{(\infty)} := \operatorname*{lex\,min}_{B \in S} \, (p(B), w(B)) \\ \Leftrightarrow \quad & p(B^{(\infty)}) \leq p(B) \quad \text{for each } B \in S && (2.7) \\ \text{and} \quad & w(B^{(\infty)}) \leq w(B) \quad \text{for each } B \in S \text{ with } p(B) = \min_{B' \in S} p(B') && (2.8) \end{aligned}$$

We set $\lambda_1 := 1$ and choose an arbitrary λ_2 with

$$\lambda_2 \geq 0 \quad \text{and} \quad \lambda_2 \geq \max_{\substack{B' \in S, \text{ with} \\ p(B') > p\left(B^{(\infty)}\right)}} \left\{ \frac{w\left(B^{(\infty)}\right) - w\left(B'\right)}{p\left(B'\right) - p\left(B^{(\infty)}\right)} \right\} \tag{2.9}$$

and claim that

$$\begin{aligned} & \lambda_1 w\left(B^{(\infty)}\right) + \lambda_2 p\left(B^{(\infty)}\right) \leq \lambda_1 w(B) + \lambda_2 p(B) \\ \Leftrightarrow \quad & w\left(B^{(\infty)}\right) + \lambda_2 p\left(B^{(\infty)}\right) \leq w(B) + \lambda_2 p(B) \end{aligned} \tag{2.10}$$

holds for all $B \in S$. To see this we consider two cases and show in the first case, that (2.10) holds for all $B \in S$ with $p(B) = p\left(B^{(\infty)}\right)$. The second case deals shows (2.10) for all $B \in S$ with $p(B) > p\left(B^{(\infty)}\right)$.

Case 1: $B \in S$ with $p(B) = p\left(B^{(\infty)}\right)$. We multiply inequality (2.7) by $\lambda_2 \geq 0$ and add (2.8), getting

$$w\left(B^{(\infty)}\right) + \lambda_2 p\left(B^{(\infty)}\right) \leq w(B) + \lambda_2 p(B)$$

for all $B \in S$ with $p(B) = p\left(B^{(\infty)}\right)$.

Case 2: $B \in S$ with $p(B) > p\left(B^{(\infty)}\right)$. The following inequalities are equivalent.

$$\begin{aligned} & w\left(B^{(\infty)}\right) + \lambda_2 p\left(B^{(\infty)}\right) \leq w(B) + \lambda_2 p(B) \\ \Leftrightarrow \quad & w\left(B^{(\infty)}\right) - w(B) \leq \lambda_2 \underbrace{\left(p(B) - p\left(B^{(\infty)}\right)\right)}_{>0} \\ \Leftrightarrow \quad & \frac{w\left(B^{(\infty)}\right) - w(B)}{p(B) - p\left(B^{(\infty)}\right)} \leq \lambda_2 \end{aligned} \tag{2.11}$$

From (2.9) it follows that (2.11) is true. ■

Example 2.3.4

In the previous Example 2.3.2 on page 34 we had four supported points, namely $B^{(0)}$, $B^{(1)}, B^{(4)}$, and $B^{(7)}$. But $B^{(7)}$ is only (equi-)optimal for $\lambda_1 = 0$. Hence, $B^{(7)}$ is a supported point, but not a penalty alternative. Conversely, the remaining three supported points $B^{(0)}, B^{(1)}$, and $B^{(4)}$ represent the penalty alternatives.

After this little geometric interpretation of penalty alternatives, which we will use in Section 5.3, we present the main results of Schwarz and some slight generalizations. It turns out that all the properties Schwarz proved for the penalty method for Σ-type problems hold for the general penalty method as well. This is because Schwarz did not use the Σ-type property in his proofs.

Lemma 2.3.5 (Properties of Penalty Alternatives, [Sch 2003, pp. 15-16])
The following two statements hold.

(i) Every penalty alternative P is optimal for all penalty parameters ε in a non-empty optimality interval $I_P = [\varepsilon_l, \varepsilon_r]$, $\varepsilon_l, \varepsilon_r \in \mathbb{R} \cup \{\infty\}$ and for no other parameters. The case $\varepsilon_l = \varepsilon_r$ is allowed. We call P an interval representative of I_P.

(ii) If P and P' are two penalty alternatives and I_P and $I_{P'}$ their optimality intervals, then only three cases are possible.

a) $I_P = I_{P'}$, iff $w(P) = w(P')$ and $p(P) = p(P')$.

b) $I_P \cap I_{P'} = \emptyset$.

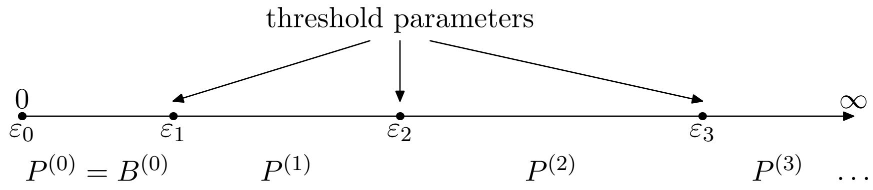
*c) $I_P \cap I_{P'} = \{\bar{\varepsilon}\}$. This means the intersection contains only a single epsilon. This happens if I_P and $I_{P'}$ are neighboring intervals. A penalty alternative whose optimality interval contains only a single $\varepsilon > 0$ is called **degenerate**. Thereby an optimal solution $B^{(0)}$ is not degenerate by definition.*

Proof. The proof can be found in [Sch 2003, pp. 16-17] and in Appendix I.1 starting on page 207. ■

Thus the interval $[0, \infty]$ can be decomposed into a set of intervals

$$[0, \varepsilon_1], [\varepsilon_1, \varepsilon_2], [\varepsilon_2, \varepsilon_3], \ldots \quad \text{with} \quad \varepsilon_1 < \varepsilon_2 < \ldots$$

and for each interval $I_i = [\varepsilon_i, \varepsilon_{i+1}]$ we have a representative solution $P^{(i)}$ with this optimality interval.



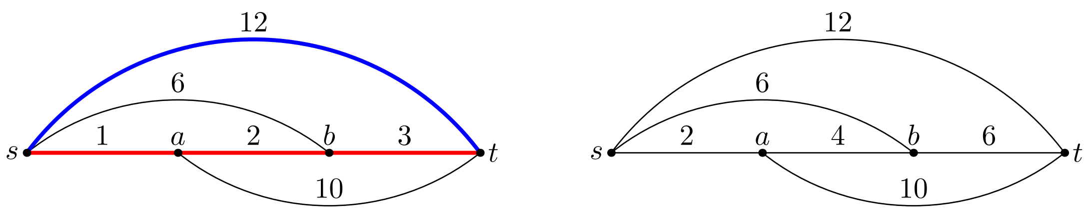
Definition 2.3.6 (i-th Penalty Alternative, k best Penalty Alternatives)
*Consider an interval decomposition with optimality intervals $[\varepsilon_0 := 0, \varepsilon_1], [\varepsilon_1, \varepsilon_2], [\varepsilon_2, \varepsilon_3], \ldots$ with $\varepsilon_1 < \varepsilon_2 < \ldots$ and different interval representatives $P^{(i)}$ for each interval. The border ε_i between the optimality intervals of $P^{(i-1)}$ and $P^{(i)}$ is called **threshold parameter between $P^{(i-1)}$ and $P^{(i)}$**.*

*The interval representative for $I_i = [\varepsilon_i, \varepsilon_{i+1}]$, which is denoted by $P^{(i)}$, is called the **i-th penalty alternative**. $P^{(i)}$ need not be uniquely determined. This is no problem since we are only interested in the functional values $w\left(P^{(i)}\right)$ and penalized parts $p\left(P^{(i)}\right)$. These values are equal for all interval representatives of the same interval (cf. Lemma 2.3.5).*
*Furthermore, the set $\left\{P^{(0)}, P^{(1)}, P^{(2)}, \ldots\right\}$ is called the set of **all penalty alternatives** and $P^{(0)}, P^{(1)}, \ldots, P^{(k-1)}$ are called the **k best penalty alternatives**.*

Note that condition $\varepsilon_1 < \varepsilon_2 < \ldots$ assures that the penalty alternatives $P^{(0)}, P^{(1)}, \ldots$ are not degenerate. This is especially important since it is possible to have several degenerate penalty alternatives which are optimal for the same penalty parameter ε but have all different weights.

For the computation of the Cordel frequency (cf. Definition 1.4.4 on page 19), only the weights $w\left(P^{(i)}\right)$ of the three best penalty alternatives $P^{(0)}, P^{(1)}, P^{(2)}$ are required. Thus, if we allow degenerate penalty alternatives, we would have to compute **all** degenerate alternatives in order to get uniquely determined weights of the three best alternatives. Otherwise, if we do not compute all degenerate penalty alternatives, the values of the three best penalty alternatives depend on which degenerate alternatives our algorithm computes. We come back to this problem in Remark 2.4.1 in Section 2.4 which deals with algorithmic issues.

Example 2.3.7
We consider the following shortest path problem.



(a) Original Graph ($\varepsilon = 0$). **(b)** Penalized Graph ($\varepsilon = 1$).

The red path $s-a-b-t$ is the shortest path from s to t and the blue marked path $s-b$ is the ∞-penalty alternative when penalizing with the canonical penalty vector.

One can easily check that $s-a-b-t$ has $[0,1]$ and that $s-b$ has $[1,\infty]$ as an optimality interval. But for $\varepsilon = 1$ (cf. Figure (b)) all paths from s to t (especially also $s-a-t$ and $s-b-t$) are penalty alternatives. Indeed, $s-a-t$ with weight 11 and $s-b-t$ with weight 9 are two degenerate penalty alternatives with different weights.

penalty alternative P	*weight* $w(P)$	*penalized part* $p(P)$	*optimality interval* I_P
$s-a-b-t$	6	6	$[0,1]$
$s-b-t$	9	3	$\{1\}$
$s-a-t$	11	1	$\{1\}$
$s-b$	12	0	$[1,\infty]$

Although $s-a-t$ and $s-b-t$ are penalty alternatives as well, the optimal solution $s-a-b-t$ is called the zeroth penalty alternative and $s-t$ is called the first (and not the third) penalty alternative.

The question is now whether this decomposition is always finite. We give the following theorem.

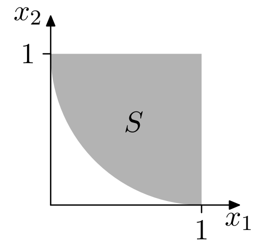
Theorem 2.3.8 (Finite Interval Decomposition, Based on [Sch 2003, p. 16]) *For Σ-type problems as well as linear optimization problems with a finite number of basic feasible solutions, the interval* $[0, \infty]$ *can be decomposed into a* ***finite*** *set of intervals* $[0, \varepsilon_1], [\varepsilon_1, \varepsilon_2], [\varepsilon_2, \varepsilon_3], \ldots, [\varepsilon_k, \infty]$ *such that for each interval* $I_i = [\varepsilon_i, \varepsilon_{i+1}]$ *we have a representative solution* $P^{(i)}$ *which is optimal for all* $\varepsilon \in I_i$*.*

Proof. For Σ-type problems the statement is obviously true because Σ-type problems have by definition only finitely many feasible solutions. With Lemma 2.3.5 it follows that $P^{(0)}, P^{(1)}, P^{(2)}, \ldots$ have to be all different and we can thus have only finitely many penalty alternatives and threshold parameters. This part has already been shown in [Sch 2003, p. 16].

For linear optimization problems with a finite number of basic feasible solutions the statement is also true because by increasing the penalty parameter ε we only change the slope of the objective function. Thus penalty alternatives have to be vertices of the polyhedron of feasible solutions. Since the vertices of the feasible polyhedron are the basic feasible solutions and since there are only finitely many basic feasible solutions by assumption, we can conclude that there are only finitely many threshold parameters again. ■

The following example shows that there exist general Σ-type problems with infinitely many threshold parameters.

Example 2.3.9 (Infinitely Many Threshold Parameters)
Consider the following minimization problem with a bounded feasible region S.



$$\min_{x \in \mathbb{R}^2} x_1 \qquad s.t. \quad x_2 \geq 1 - \sqrt{2x_1 - x_1^2}$$
$$0 \leq x_1, x_2 \leq 1$$

$x^{(0)} = [0, 1]$ *is the uniquely determined optimal solution. With the canonical penalty vector* $p = [0, 1]$*, an* ε*-penalty alternative is an optimal solution of the problem*

$$\min_{x \in \mathbb{R}^2} x_1 + \varepsilon x_2 \qquad s.t. \; x_2 \geq 1 - \sqrt{2x_1 - x_1^2}\,,$$
$$0 \leq x_1, x_2 \leq 1\,.$$

It follows, that

$$x^{(\varepsilon)} = \left[1 - \frac{1}{\sqrt{1+\varepsilon^2}}, 1 - \frac{\varepsilon}{\sqrt{1+\varepsilon^2}}\right] \in S \subseteq \mathbb{R}^2$$

is the uniquely determined ε*-penalty alternative. Thus, each* $\varepsilon \geq 0$ *provides another penalty alternative* $x^{(\varepsilon)}$*. Consequently, we have infinitely many threshold parameters even though the feasible region is bounded.*

From Althöfer, Berger and Schwarz we have the following theorem regarding the weights and penalized parts of penalty alternatives.

Theorem 2.3.10 (Althöfer, Berger, Schwarz [ABS 2002])
Let $w : E \to \mathbb{R}$ be a real-valued function and $p : E \to \mathbb{R}_+$ a positive real valued function on E. Let $B^{(\varepsilon)}$ be defined according to Definition 2.2.2 for $\varepsilon \in \mathbb{R}_+$. The following four statements hold:

(i) $p\left(B^{(\varepsilon)}\right)$ *is weakly monotonically decreasing in* ε.

(ii) $w\left(B^{(\varepsilon)}\right)$ *is weakly monotonically increasing in* ε.

(iii) $w\left(B^{(\varepsilon)}\right) - p\left(B^{(\varepsilon)}\right)$ *is weakly monotonically increasing in* ε.

(iv) $w\left(B^{(\varepsilon)}\right) + \varepsilon \cdot p\left(B^{(\varepsilon)}\right)$ *is weakly monotonically increasing in* ε.

Proof. The proof was given by [Sch 2003, p. 10]. In slightly different notation it can be found in Appendix I.2 starting on page 209. ■

Thus, the penalized part decreases and the weight of $B^{(\varepsilon)}$ increases as ε increases. The following theorem gives a more precise statement.

Theorem 2.3.10'
Given two arbitrary consecutive penalty alternatives $P^{(i)}$ and $P^{(i+1)}$ ($i \geq 0$), the following two statements hold.

(i) $p\left(P^{(i)}\right) > p\left(P^{(i+1)}\right)$.

(ii) For $i \geq 1$ it holds that $w\left(P^{(i)}\right) < w\left(P^{(i+1)}\right)$ and for $i = 0$ holds that $w\left(P^{(0)}\right) \leq w\left(P^{(1)}\right)$.

Statement (i) can be found in [Sch 2003, p. 17] in a slightly different context as an argument within a proof (proof of property (3)) rather than a theorem.

Proof. Let $I_i = [\varepsilon_i, \varepsilon_{i+1}]$ and $I_{i+1} = [\varepsilon_{i+1}, \varepsilon_{i+2}]$ denote the optimality intervals of $P^{(i)}$ and $P^{(i+1)}$. Then the following two statements hold.

1. $\varepsilon \leq \varepsilon_{i+1}$ for all ε for which $P^{(i)}$ is optimal.

2. $\varepsilon \geq \varepsilon_{i+1}$ for all ε for which $P^{(i+1)}$ is optimal.

Thus $p\left(P^{(i)}\right) \geq p\left(P^{(i+1)}\right)$ and $w\left(P^{(i)}\right) \leq w\left(P^{(i+1)}\right)$ hold according to Theorem 2.3.10.

Now we want to show that the strict inequalities

$$\begin{aligned} p\left(P^{(i)}\right) &> p\left(P^{(i+1)}\right) && \text{for } i \geq 0\,, \\ w\left(P^{(i)}\right) &< w\left(P^{(i+1)}\right) && \text{for } i \geq 1 \end{aligned}$$

hold.

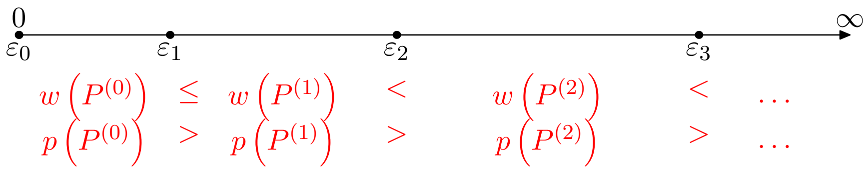
For the threshold parameter ε_k between $P^{(i)}$ and $P^{(i+1)}$ the following holds.

$$\begin{aligned} & f_\varepsilon\left(P^{(i)}\right) = f_\varepsilon\left(P^{(i+1)}\right) \\ \Leftrightarrow \quad & w\left(P^{(i)}\right) + \varepsilon p\left(P^{(i)}\right) = w\left(P^{(i+1)}\right) + \varepsilon p\left(P^{(i+1)}\right) \\ \Leftrightarrow \quad & w\left(P^{(i)}\right) - w\left(P^{(i+1)}\right) = \varepsilon\left(p\left(P^{(i+1)}\right) - p\left(P^{(i)}\right)\right) \end{aligned}$$

From this it follows that $p\left(P^{(i)}\right) = p\left(P^{(i+1)}\right)$ implies $w\left(P^{(i)}\right) = w\left(P^{(i+1)}\right)$, which with Lemma 2.3.5 implies that $P^{(i)}$ and $P^{(i+1)}$ have the same optimality intervals. This is a contradiction, because $P^{(i)}$ and $P^{(i+1)}$ are representatives for different optimality intervals.

In the case $i \geq 1$, $\varepsilon > 0$ holds. Thus we conclude analogously that $w\left(P^{(i)}\right) = w\left(P^{(i+1)}\right)$ implies $p\left(P^{(i)}\right) = p\left(P^{(i+1)}\right)$, which is again a contradiction. ■

So we get the following picture.



Lemma 2.3.11
The threshold parameter ε_i between the penalty alternatives $P^{(i-1)}$ and $P^{(i)}$ is

$$\varepsilon_i = \frac{w\left(P^{(i)}\right) - w\left(P^{(i-1)}\right)}{p\left(P^{(i-1)}\right) - p\left(P^{(i)}\right)}.$$

Proof. Since ε_i is the threshold parameter between $P^{(i-1)}$ and $P^{(i)}$ both penalty alternatives must have the same penalized value for $\varepsilon = \varepsilon_i$. With $p\left(P^{(i-1)}\right) \neq p\left(P^{(i)}\right)$, as guaranteed by Theorem 2.3.10', the following holds:

$$\begin{aligned} & f_{\varepsilon_i}\left(P^{(i-1)}\right) = f_{\varepsilon_i}\left(P^{(i)}\right) \\ \Leftrightarrow \quad & w\left(P^{(i-1)}\right) + \varepsilon_i \cdot p\left(P^{(i-1)}\right) = w\left(P^{(i)}\right) + \varepsilon_i \cdot p\left(P^{(i)}\right) \\ \Leftrightarrow \quad & \varepsilon_i = \frac{w\left(P^{(i)}\right) - w\left(P^{(i-1)}\right)}{p\left(P^{(i-1)}\right) - p\left(P^{(i)}\right)} \end{aligned}$$ ■

With this lemma it follows that for rational vectors $w, p \in \mathbb{Q}^n$ the threshold parameters $\varepsilon_1, \varepsilon_2, \ldots$ are also rational. Please note that the canonical penalty vector p from Definition 2.2.7 on page 28 is rational if the weight vector w is rational and if the used optimal solution $B^{(0)}$ is integer.

At the end of this section we formalize Definition 1.4.3 (ii) on page 18 of the penalty selection rule. This leads to the following definition of the Cordel Frequency.

Definition 2.3.12 (Cordel Frequency for the Penalty Selection Rule)
Consider an arbitrary general Σ-type problem. Then the ***penalty selection rule*** P *chooses the three best penalty alternatives* $x_1 = P^{(0)} = B^{(0)}, x_2 = P^{(1)}$, *and* $x_3 = P^{(2)}$ *with*

$$w(x_1) \leq w(x_2) < w(x_3).$$

Hence, the ***Cordel frequency for the penalty selection rule*** *is*

$$CF = P(d_1 \geq d_2) = P\left(w\left(P^{(1)}\right) - w\left(P^{(0)}\right) \geq w\left(P^{(2)}\right) - w\left(P^{(1)}\right)\right). \tag{2.12}$$

The next section deals with modifications of the algorithm of Schwarz in order to compute the k best penalty alternatives. Running time improvements and numerical stability are also covered there. Afterwards, in Chapter 3 starting on page 67, we present the Cordel frequencies for different examples of general Σ-type problems under the penalty selection rule.

2.4 Computation of the k Best Penalty Alternatives $P^{(0)}, P^{(1)}, \ldots, P^{(k-1)}$

2.4.1 The Algorithm of Schwarz for Computing all Penalty Alternatives

On page 15 of his doctoral thesis Schwarz introduced an algorithm for Σ-type problems which computes a sequence $\mathcal{P}$ of ε-optimal solutions covering all $\varepsilon \geq 0$. This is in fact the set of all penalty alternatives $\{P^{(0)}, P^{(1)}, \ldots\}$ as well as potentially some degenerate penalty alternatives. Since this algorithm also works for general Σ-type problems with finitely many penalty alternatives we present it here. Note that we made some slight changes especially in notations that do not affect the basic idea of the algorithm.

Algorithm 1 (Algorithm of Schwarz for Computing all Penalty Alternatives, [Sch 2003, p. 15])
The algorithm computes an ordered set of penalty alternatives $\mathcal{P}$ *and the corresponding ordered set of threshold parameters* $\mathcal{T}$.

Initialization: *Compute* $B^{(0)}$ *and* $B^{(\infty)}$. *As in Definition 2.2.2,* $B^{(0)}$ *is an optimal solution which minimizes* $w(B)$ *and* $B^{(\infty)}$ *is a solution which minimizes* $p(B)$ *and has a* $w(B)$*-value as small as possible.*

If $p\left(B^{(0)}\right) = p\left(B^{(\infty)}\right)$, *then the solution* $B^{(0)}$ *is optimal for all* $\varepsilon \geq 0$. *Hence, we set* $\mathcal{P} = \left[B^{(0)}\right]$ *and* $T = [\,]$ *stop.*

Otherwise set

number of computed penalty alternatives	$m := 2\,,$	
left penalty alternative and threshold parameter	$L := B^{(0)}\,,$	$\varepsilon_L := 0\,,$
right penalty alternative and threshold parameter	$R := B^{(\infty)}\,,$	$\varepsilon_R := \infty$

*and go to **Step 1**.*

Branching:

Step 1 *Compute the potential threshold parameter $\overline{\varepsilon}$ between L and R by*

$$\begin{aligned} w(L) + \overline{\varepsilon} \cdot p(L) &= w(R) + \overline{\varepsilon} \cdot p(R) \\ \Leftrightarrow \qquad \overline{\varepsilon} &:= \frac{w(R) - w(L)}{p(L) - p(R)}\,. \end{aligned}$$

Step 2 *Find an optimal solution $B^{(\overline{\varepsilon})}$ for the parameter $\overline{\varepsilon}$.*

Step 3 *If*

$$f_{\overline{\varepsilon}}\left(B^{(\overline{\varepsilon})}\right) = f_{\overline{\varepsilon}}(L) = f_{\overline{\varepsilon}}(R)\;, \tag{2.13}$$

set $\mathcal{T} := [\overline{\varepsilon}]$ and $\mathcal{P} = [\,]$ and branch no further.

Otherwise, $B^{(\overline{\varepsilon})}$ is a new penalty alternative which differs from both L and R. Thus we set $m := m + 1$ and branch:

*a) Compute all penalty alternatives $\mathcal{P}_L$ and all threshold parameters $\mathcal{T}_L$ in the left interval $(\varepsilon_L, \overline{\varepsilon})$. Hence, go to **Step 1** with*

$$L := L, \quad R := B^{(\overline{\varepsilon})} \quad \text{and} \quad \varepsilon_L := \varepsilon_L, \quad \varepsilon_R := \overline{\varepsilon}\,.$$

*b) Compute all penalty alternatives $\mathcal{P}_R$ and all threshold parameters $\mathcal{T}_R$ in the right interval $(\overline{\varepsilon}, \varepsilon_R)$. Hence, go to **Step 1** with*

$$L := B^{(\overline{\varepsilon})}, \quad R := R \quad \text{and} \quad \varepsilon_L := \overline{\varepsilon}, \quad \varepsilon_R := \varepsilon_R\,.$$

Afterwards set $\mathcal{P} := \left[\mathcal{P}_L, B^{(\overline{\varepsilon})}, \mathcal{P}_R\right]$ and $\mathcal{T} := [\mathcal{T}_L, \mathcal{T}_R]$.

After the branching we set $\mathcal{P} := \left[B^{(0)}, \mathcal{P}, B^{(\infty)}\right]$.

The slightly informal notation $\mathcal{P} := \left[\mathcal{P}_L, B^{(\overline{\varepsilon})}, \mathcal{P}_R\right]$ means, that we merge the three ordered sets of penalty alternatives $\mathcal{P}_L, B^{(\overline{\varepsilon})}$, and $\mathcal{P}_R$ successively. In doing so we simply leave out empty sets.

We call $\overline{\varepsilon}$ the potential threshold parameter between L and R because we do not know whether L and R are already neighboring alternatives or not. If condition (2.13) is fulfilled, then $\bar{\varepsilon}$ really is a threshold parameter. That is why we store $\bar{\varepsilon}$ in the set of threshold parameters T and stop further branching. Otherwise a new penalty alternative $B^{(\bar{\varepsilon})}$ was found and we have to continue branching. Thus in each iteration step either a new threshold parameter or a new penalty alternative is found.

Hence, it is sufficient to solve $2\#P-1$ problems of the type $\min_{B\in S} w(B)+\varepsilon p(B)$ if we have a finite number of penalty alternatives $\#P<\infty$. Otherwise if there are infinitely many penalty alternatives as in Example 2.3.9, the algorithm will never stop.

Remark 2.4.1
After completion of Algorithm 1 one has to do a post-processing step and exclude all penalty alternatives in $\mathcal{P}$ which are degenerate. There is no way to leave this step out as long as we cannot control which $\bar{\varepsilon}$-penalty alternatives our basic optimization algorithm computes.

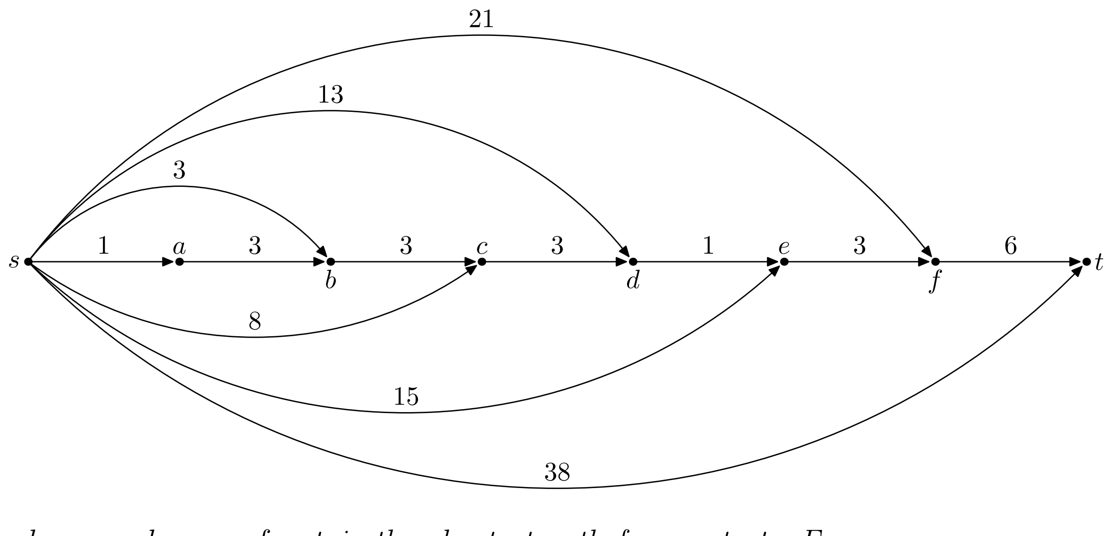
*For an optimization problem containing degenerate penalty alternatives, it is up to the algorithm that computes **one** optimal solution to the punished problem*

$$\min_{B\in S} w(B)+\varepsilon\cdot p(B)\,, \tag{2.14}$$

*whether and if so which degenerate penalty alternatives are computed. That is why we decided to leave out degenerate alternatives in Definition 2.3.6. For the computation of **all** degenerate and nondegenerate alternatives, an algorithm is required that computes **all** optimal solutions to the punished problem (2.14). In most cases this is very time-consuming and, thus, not practicable.*

We illustrate the algorithm in the following four page example.

Example 2.4.2
Consider the following shortest path problem.



$s-b-c-d-e-f-t$ is the shortest path from s to t. From now on we use the short name "$sbcdeft$" for this path. We get the following table containing the weights

and punished parts (for penalization with the canonical penalty vector). The columns on the right give the punished values for selected penalty parameters. These values are needed within the algorithm.

Path B	$w(B)$	$p(B)$	$f_{\frac{1}{3}}(B)$	$f_{\frac{1}{2}}(B)$	$f_{\frac{2}{3}}(B)$	$f_{\frac{3}{4}}(B)$	$f_1(B)$	$f_{\frac{14}{9}}(B)$	$f_{\frac{11}{6}}(B)$
"sbcdeft"	19	19	$= \mathbf{25\frac{1}{3}}$	$= 28\frac{1}{2}$	$= 31\frac{2}{3}$	$= 33\frac{1}{4}$	$= 38$	$= 48\frac{5}{9}$	$= 53\frac{5}{6}$
"sabcdeft"	20	16	$= \mathbf{25\frac{1}{3}}$	$= 28$	$= 30\frac{2}{3}$	$= 32$	$= 36$	$= 44\frac{8}{9}$	$= 49\frac{1}{3}$
"scdeft"	21	13	$= \mathbf{25\frac{1}{3}}$	$= \mathbf{27\frac{1}{2}}$	$= \mathbf{29\frac{2}{3}}$	$= 30\frac{3}{4}$	$= 34$	$= 41\frac{2}{9}$	$= 44\frac{5}{6}$
"sdeft"	23	10	$= 26\frac{1}{3}$	$= 28$	$= \mathbf{29\frac{2}{3}}$	$= \mathbf{30\frac{1}{2}}$	$= \mathbf{33}$	$= 38\frac{5}{9}$	$= 41\frac{1}{3}$
"seft"	24	9	$= 27$	$= 28\frac{1}{2}$	$= 30$	$= 30\frac{3}{4}$	$= \mathbf{33}$	$= 38$	$= 40\frac{1}{2}$
"sft"	27	6	$= 29$	$= 30$	$= 31$	$= 31\frac{1}{2}$	$= \mathbf{33}$	$= \mathbf{36\frac{1}{3}}$	$= \mathbf{38}$
"st"	38	0	$= 38$	$= 38$	$= 38$	$= 38$	$= 38$	$= 38$	$= \mathbf{38}$

In the following we present how Schwarz's algorithm works for this example. Figure 2.4.1 on page 47 shows the branching tree of the algorithm which gives a good overview on the interval division.

Initialization: *The optimal solution is $B^{(0)}$ = "sbcdeft" and the ∞-penalty alternative is $B^{(\infty)}$ = "st". Since $p\left(B^{(0)}\right) = 19 \neq 0 = p\left(B^{(\infty)}\right)$ holds, we can start the branching. Thus we set $m = 2$ and*

$$L := \text{"sbcdeft"}\,, \qquad \varepsilon_L := 0\,,$$
$$R := \text{"st"}\,, \qquad \varepsilon_R := \infty$$

*and go to **Step 1**.*

In the first iteration we compute

$$\overline{\varepsilon} = \frac{w(R) - w(L)}{p(L) - p(R)} = \frac{w(\text{"st"}) - w(\text{"sbcdeft"})}{p(\text{"sbcdeft"}) - p(\text{"st"})} = \frac{38 - 19}{19 - 0} = 1\,.$$

Now the algorithm computes a $\overline{\varepsilon}$-alternative. In fact, there are three equally good penalty alternatives, namely "sdeft", "seft" and "sft". Suppose the algorithm computes $B^{((1)}$ = "seft". Because of

$$f_{\overline{\varepsilon}}\left(B^{(\overline{\varepsilon})}\right) = f_{\overline{\varepsilon}}(\text{"seft"}) = 33 \neq 38 = f_{\overline{\varepsilon}}\left(B^{(0)}\right) = f_{\overline{\varepsilon}}\left(B^{(\infty)}\right)$$

we branch further and browse the intervals $(0, 1)$ and $(1, \infty)$ at the next level.

In the left interval $(0, 1)$ we compute the next potential threshold parameter by

$$\overline{\varepsilon} = \frac{w(R) - w(L)}{p(L) - p(R)} = \frac{w(\text{"seft"}) - w(\text{"sbcdeft"})}{p(\text{"sbcdeft"}) - p(\text{"seft"})} = \frac{24 - 19}{19 - 9} = \frac{1}{2}$$

and the corresponding penalty alternative $B^{\left(\frac{1}{2}\right)}$ = "scdeft". Again we continue branching since

$$f_{\overline{\varepsilon}}\left(B^{(\overline{\varepsilon})}\right) = f_{\overline{\varepsilon}}(\text{"scdeft"}) = 27.5 \neq 28.5 = f_{\overline{\varepsilon}}(\text{"sbcdeft"}) = f_{\overline{\varepsilon}}(\text{"seft"})$$

holds. This time we have to cut the interval $(0,1)$ *into the two halves* $\left(0,\frac{1}{2}\right)$ *and* $\left(\frac{1}{2},1\right)$.

The potential threshold parameter in the left interval $\left(0,\frac{1}{2}\right)$ *with* $L=$ *"sbcdeft" and* $R=$ *"scdeft" is*

$$\overline{\varepsilon}=\frac{w(R)-w(L)}{p(L)-p(R)}=\frac{w(\text{“}scdeft\text{”})-w(\text{“}sbcdeft\text{”})}{p(\text{“}sbcdeft\text{”})-p(\text{“}scdeft\text{”})}=\frac{21-19}{19-13}=\frac{1}{3}.$$

Again we have three equally best shortest pathes, namely "sbcdeft", "sabcdeft", and "scdeft". Suppose the algorithm computes $B^{\left(\frac{1}{3}\right)}=$ *"sabcdeft". Even though this is a penalty alternative which differs from both L and R we stop further branching since the stop criterion* (2.13)

$$f_{\overline{\varepsilon}}(\text{“}sabcdeft\text{”})=f_{\overline{\varepsilon}}(\text{“}sbcdeft\text{”})=f_{\overline{\varepsilon}}(\text{“}scdeft\text{”})\approx 25.3$$

is fulfilled. This means that $\overline{\varepsilon}$ *is a threshold parameter. Hence, we stop here and return* $\mathcal{T}:=[\overline{\varepsilon}]=\left[\frac{1}{3}\right]$ *and* $\mathcal{P}=[\,]$. *Note that since* $B^{\left(\frac{1}{3}\right)}=$ *"sabcdeft" differs from both L and R, it is a degenerate penalty alternative.*

One level higher we still have to search through the right interval $\left(\frac{1}{2},1\right)$. *This provides* $\mathcal{P}_R=[\text{“}sdeft\text{”}]$ *and* $\mathcal{T}_R=\left[\frac{2}{3},1\right]$ *(cf. Figure 2.4.1 on page 47). With* $\mathcal{P}_L=[\,]$ *and* $\mathcal{T}_L=\left[\frac{1}{3}\right]$ *we set*

$$\mathcal{P}:=\left[\mathcal{P}_L,B^{(\varepsilon)},\mathcal{P}_R\right]=[\underbrace{\text{“}scdeft\text{”}}_{B^{(\varepsilon)}},\underbrace{\text{“}sdeft\text{”}}_{\mathcal{P}_R}],$$
$$\mathcal{T}:=[\mathcal{T}_L,\mathcal{T}_R]=\Big[\underbrace{\frac{1}{3}}_{\mathcal{T}_L},\underbrace{\frac{2}{3},1}_{\mathcal{T}_R}\Big].$$

These two ordered sets are returned to the higher starting level. With $\mathcal{P}_R=[\text{“}sft\text{”}]$ *and* $\mathcal{T}_R=\left[1,\frac{11}{6}\right]$ *(again cf. Figure 2.4.1 on page 47) we set*

$$\mathcal{P}:=\left[\mathcal{P}_L,B^{(\varepsilon)},\mathcal{P}_R\right]=[\underbrace{\text{“}scdeft\text{”},\text{“}sdeft\text{”}}_{\mathcal{P}_L},\underbrace{\text{“}seft\text{”}}_{B^{(\varepsilon)}},\underbrace{\text{“}sft\text{”}}_{\mathcal{P}_R}],$$
$$\mathcal{T}:=[\mathcal{T}_L,\mathcal{T}_R]=\Big[\underbrace{\frac{1}{3},\frac{2}{3},1}_{\mathcal{T}_L},\underbrace{1,\frac{11}{6}}_{\mathcal{T}_R}\Big].$$

Therewith we finished the branching part of the algorithm.

While the set of threshold parameters is already complete, we still have to insert $B^{(0)}$ *and* $B^{(\infty)}$ *to the ordered set of penalty alternatives. Thus we set*

$$\mathcal{P}:=\left[B^{(0)},\mathcal{P},B^{(\infty)}\right]=[\underbrace{\text{“}sbcdeft\text{”}}_{B^{(0)}},\underbrace{\text{“}scdeft\text{”},\text{“}sdeft\text{”},\text{“}seft\text{”},\text{“}sft\text{”}}_{\mathcal{P}},\underbrace{\text{“}st\text{”}}_{B^{(\infty)}}],$$

and

$$\mathcal{T}=\left[\frac{1}{3},\frac{2}{3},1,1,\frac{11}{6}\right]$$

stays unchanged.

It is striking that $\mathcal{T}$ consists of two identical threshold parameters: the threshold parameter 1 is included twice. This means that the ordered set $\mathcal{P}$ contains a degenerate penalty alternative. Thus, we have to do a post-processing step as mentioned in Remark 2.4.1 in order to exclude the degenerate penalty alternatives. This step is quite easy. One only has to go through the threshold parameters and filter out the penalty alternative which is between the two identical threshold parameters.

- *"sbcdeft" is optimal in the interval* $\left[0, \frac{1}{3}\right]$.
- *"scdeft" is optimal in the interval* $\left[\frac{1}{3}, \frac{2}{3}\right]$.
- *"sdeft" is optimal in the interval* $\left[\frac{2}{3}, 1\right]$.
- ***"seft" is optimal in the interval*** $\left[\mathbf{1}, \mathbf{1}\right]$.
- *"sft" is optimal in the interval* $\left[1, \frac{11}{6}\right]$.
- *"st" is optimal in the interval* $\left[\frac{11}{6}, \infty\right]$.

Hence, we exclude the degenerate penalty alternative "seft". Therewith we get the final ordered set $\mathcal{P}$ containing all penalty alternatives and the ordered set $\mathcal{T}$ containing all

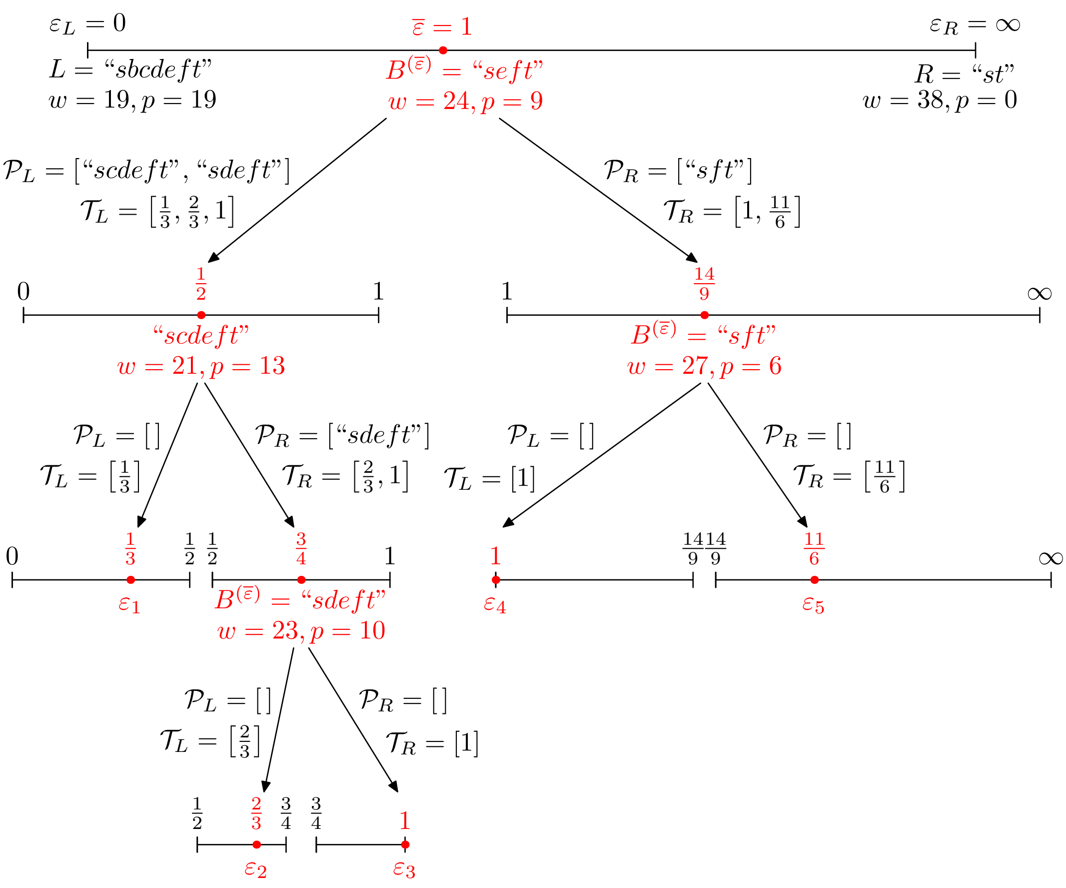


Figure 2.4.1: Branching tree of Schwarz's algorithm for Example 2.4.2.

threshold parameters.

$$\mathcal{P} = [\text{“}sbcdeft\text{”}, \text{“}scdeft\text{”}, \text{“}sdeft\text{”}, \text{“}sft\text{”}, \text{“}st\text{”}] \qquad \mathcal{T} = \left[\frac{1}{3}, \frac{2}{3}, 1, \frac{11}{6}\right]$$

This concludes this big introductory Example 2.4.2.

In the next sections we present some modifications of this basic algorithm in order to improve the running time and numerical stability for the computation of the m best penalty alternatives. Furthermore we present some experimental results on how much computation time one can save by applying the proposed modifications. The considered optimization problems and the rules for the generation of random instances are introduced in detail in the following Chapter 3 starting on page 67.

2.4.2 Left-First Traversal

Note that we need only the three best penalty alternatives $P^{(0)}, P^{(1)}$, and $P^{(2)}$ if we want to compute the Cordel frequency. Hence it is not necessary to compute **all** penalty alternatives, as done by the basic algorithm of Schwarz. That is why we suggest to shift the search more to the left side. This means that we first continue further branching in the left interval $(\varepsilon_L, \overline{\varepsilon})$ instead of the right interval $(\overline{\varepsilon}, \varepsilon_R)$.

Let k be the number of penalty alternatives that we want to compute. For example we set $k = 3$ if we are only interested in $P^{(0)}, P^{(1)}$, and $P^{(2)}$. If the left interval $(\varepsilon_L, \overline{\varepsilon})$ already provides enough penalty alternatives

$$|\mathcal{P}_L| \geq k - 2$$

we can stop the branching without examination of the right interval. Please keep in mind that $\mathcal{P}_L$ does not contain the optimal solution $P^{(0)}$ and the current solution $B^{(\overline{\varepsilon})}$. Thus $k - 2$ new alternative solutions are sufficient.

Example 2.4.3
Applying left-first traversal in Example 2.4.2 with $k = 3$*, the algorithm stops at the first level after the examination of the left interval* $(0, 1)$*. This interval provides*

$$\mathcal{P}_L = [\text{“}scdeft\text{”}, \text{“}sdeft\text{”}] \quad \text{with} \quad |\mathcal{P}_L| = 2 \geq k - 2\,.$$

Hence it is not necessary to investigate the right interval $(1, \infty)$*, too.*

In case $k = 4$ *the algorithm would also stop at the first level. But*

$$\left[B^{(0)}, \mathcal{P}_L, B^{(\overline{\varepsilon})}\right] = [\text{“}sbcdeft\text{”}, \text{“}scdeft\text{”}, \text{“}sdeft\text{”}, \text{“}seft\text{”}]$$

includes the degenerate penalty alternative “seft”. Thus, after excluding “seft” out, $\left[B^{(0)}, \mathcal{P}_L, B^{(\overline{\varepsilon})}\right]$ *does not contain the required* $k = 4$ *alternatives any longer.*

However, in this case the degenerate alternative "$seft$" can only be detected if also the neighboring penalty alternative "sft" is computed. Otherwise it is not possible to clearly recognize "$seft$" as degenerate penalty alternative.

We saw in the last example that problems with degenerate penalty alternatives can arise. Concretely, two kinds of problems may arise.

1. It is very rare that you have to compute a further penalty alternative, in order to decide, whether the last calculated alternative is degenerate or not (cf. Example 2.4.3 above for $k = 4$).
2. Sometimes, excluding degenerate alternatives reduces the number of computed penalty alternatives to such an extent that the remaining number of penalty is smaller than the predetermined number k.

In both cases one has to compute at least one more penalty alternative by investigation of the remaining right interval. In order to prevent such problems, we always try to compute one penalty alternative more than required. This gives us the following algorithm.

Algorithm 2 (Algorithm of Schwarz Combined with Left-First-Traversal for Computing the k Best Penalty Alternatives)

Initialization: *Compute $B^{(0)}$ and $B^{(\infty)}$.*
If $p\left(B^{(0)}\right) = p\left(B^{(\infty)}\right)$, then the solution $B^{(0)}$ is optimal for all $\varepsilon \geq 0$. Hence, we set $\mathcal{P} = \left\{B^{(0)}\right\}$ and $T = [\,]$ stop.
Otherwise set

$$m := 2\,, \qquad L := B^{(0)}\,, \qquad \varepsilon_L := 0\,, \qquad R := B^{(\infty)}\,, \qquad \varepsilon_R := \infty$$

and `canceled := false` *and go to **Step 1**.*

Branching:

Step 1 *Compute the potential threshold parameter*

$$\overline{\varepsilon} := \frac{w\,(R) - w\,(L)}{p\,(L) - p\,(R)}$$

between L and R.

Step 2 *Find an optimal solution $B^{(\overline{\varepsilon})}$ for the parameter $\overline{\varepsilon}$.*

Step 3 *If*

$$f_{\overline{\varepsilon}}\left(B^{(\overline{\varepsilon})}\right) = f_{\overline{\varepsilon}}\,(L) = f_{\overline{\varepsilon}}\,(R)\,, \tag{2.15}$$

set $\mathcal{T} := [\overline{\varepsilon}]$ and $\mathcal{P} = [\,]$ and branch no further.

Otherwise, $B^{(\bar{\varepsilon})}$ is a new penalty alternative which differs from both L and R. Thus we set $m := m+1$ and branch:

*a) Compute all penalty alternatives $\mathcal{P}_L$ and all threshold parameters $\mathcal{T}_L$ in the left interval $(\varepsilon_L, \bar{\varepsilon})$. Hence, go to **Step 1** with*

$$L := L, \quad R := B^{(\bar{\varepsilon})} \quad \text{and} \quad \varepsilon_L := \varepsilon_L, \quad \varepsilon_R := \bar{\varepsilon}\,.$$

If $|\mathcal{P}_l| \geq k-1$ holds, then we set

$$\mathcal{P}_R := [\,]\,, \qquad \mathcal{T}_R := [\,]\,, \qquad \texttt{canceled:= true}$$

*and stop further branching without investigation of the right interval. Otherwise, if $|\mathcal{P}_l| < k-1$ holds, then we go to **Step 3b**.*

*b) Compute all penalty alternatives $\mathcal{P}_R$ and all threshold parameters $\mathcal{T}_R$ in the right interval $(\bar{\varepsilon}, \varepsilon_R)$. Hence, go to **Step 1** with*

$$L := B^{(\bar{\varepsilon})}, \quad R := R \quad \text{and} \quad \varepsilon_L := \bar{\varepsilon}, \quad \varepsilon_R := \varepsilon_R\,.$$

Afterwards set $\mathcal{P} := \left[\mathcal{P}_L, B^{(\bar{\varepsilon})}, \mathcal{P}_R\right]$ and $\mathcal{T} := [\mathcal{T}_L, \mathcal{T}_R]$.

After the branching we set

$$\mathcal{P} := \begin{cases} \left[B^{(0)}, \mathcal{P}, B^{(\infty)}\right], & \text{if } \texttt{canceled = false}\,, \\ \left[B^{(0)}, \mathcal{P}\right], & \text{if } \texttt{canceled = true} \end{cases}$$

and exclude all penalty alternatives which are obviously degenerate. If afterwards $|\mathcal{P}| < k$ holds and if the branching was stopped (`canceled = true`), then the remaining right interval has to be investigated, too. Hence, for $\mathcal{P} = \left[B^{(0)}, P^{(1)}, \ldots, P^{(m)}\right]$ and $\mathcal{T} = [\varepsilon_1, \ldots, \varepsilon_m]$ with $m < k-1$ we set

$$k := k - m \quad \text{and} \quad \texttt{canceled:= false}$$

*and go to **Step 1** with*

$$L := P^{(m)}, \quad R := B^{(\infty)} \quad \text{and} \quad \varepsilon_L := \varepsilon_m, \quad \varepsilon_R := \infty\,.$$

Despite the problems with degenerate alternatives, application of left-first traversal can save a lot of computing time as the following experimental results show.

Experimental Results for the Shortest Path Problem in Grid Graphs

We analyzed the computation times for the computation of the penalty alternatives in shortest path problems in directed grid graphs (detailed information in Section 3.1 starting on page 67).

Figure 2.4.2 (a) shows the reduction of the number of computed penalty alternatives by applying left-first traversal for $k = 3$. For left-first traversal the average number converges to approximately 12 computed penalty alternatives. In contrast, the number of computed penalty alternatives for the computation of all penalty alternatives ($k = \infty$) does not seem to converge. This is because the larger the graphs are, the more penalty alternatives exist.

Hence, the reduction of running time increases for bigger instances. This can be seen in Figure 2.4.2 (b). The saving in runtime goes up to 72% for 500×500 grids and will be even larger for larger grids.

Experimental Results for Knapsack Problems

In Section 3.5 starting on page 82, b-bounded knapsack problems are introduced. For $b = 1$ the common binary knapsack problem and for $b = \infty$ the unbounded knapsack problem occurs.

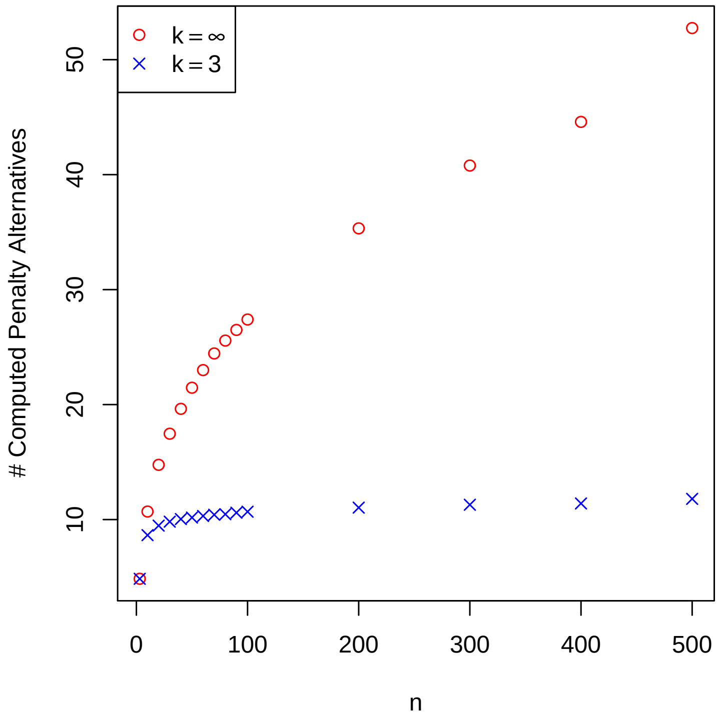
Again we investigated how much running time could be saved by left-first traversal. This is shown in Figure 2.4.3 (a). Except the case $b = \infty$, the improvement by applying left-first traversal is enormous and increasing for larger numbers of items.

In fact, the unbounded knapsack problem ($b = \infty$) is definitely an exception. For all other considered optimization problems, the number of penalty alternatives is increasing for bigger instances (cf. Figure 2.4.3 (b)). Only for the unbounded knapsack problem this was not the case. There, the number of penalty alternatives is first increasing until a maximal number of approximately 5.5 penalty alternatives is reached. This is followed by a decrease.

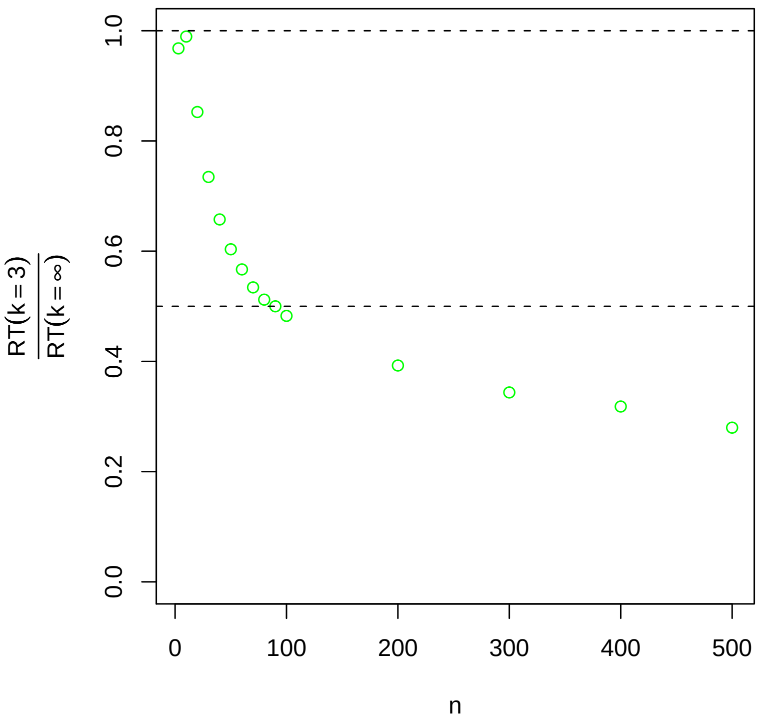
There are two reasons for this curve shape. The first reason is that unbounded knapsack problems with many items include many dominated items. Thus, an unbounded knapsack problem with $n = 100$ items might consist of only 5 non-dominated items. Another reason is, that unbounded knapsack problems with many items are very likely to contain one "super item" which has a very small weight and a very great usage (or profit). In this case, the optimal solution consists almost entirely of this super item. Consequently, in this case there are only few penalty alternatives.

The fact, that there are only few penalty alternatives at all in case of the unbounded knapsack problem, is also why left-first traversal cannot speed up the computation.

Overall, it has been shown, that left-first traversal is a very powerful tool for speeding up computation time. The time saving potential is enormous!

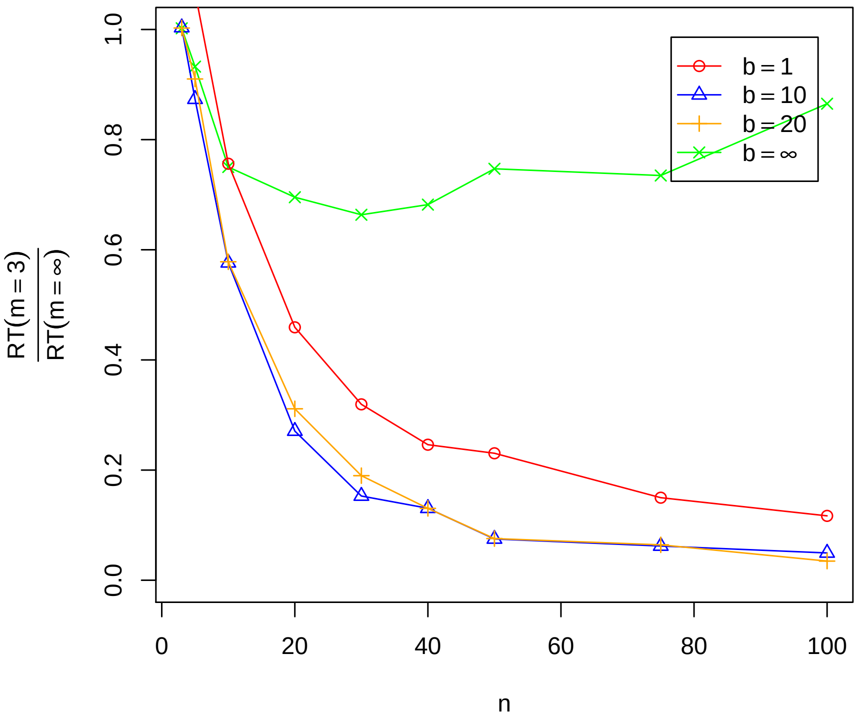


(a) Average number of computed penalty alternatives for the application of left-first traversal ($k = 3$) and for the computation of all penalty alternatives ($k = \infty$).

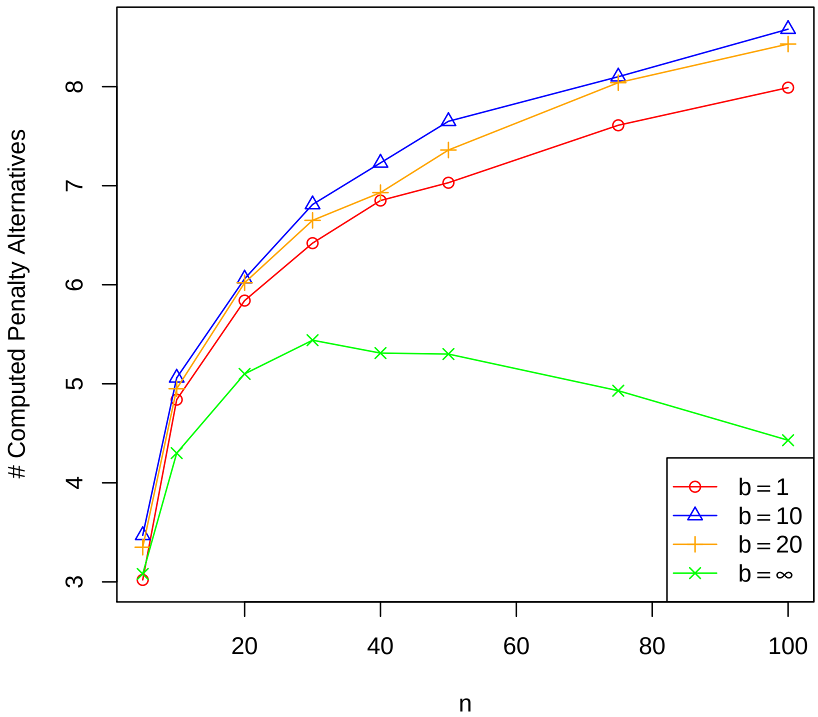


(b) Ratio $\frac{\mathrm{RT}(k=3)}{\mathrm{RT}(k=\infty)}$ of the running times $\mathrm{RT}(k = 3)$ for the computation of the $k = 3$ best penalty alternatives with left-first traversal and $\mathrm{RT}(k = \infty)$ for the computation of all penalty alternatives.

Figure 2.4.2: Shortest path problem in directed grid graphs (detailed information in Section 3.1 starting on page 67).



(a) Ratio $\frac{\mathrm{RT}(k=3)}{\mathrm{RT}(k=\infty)}$ of the running times $\mathrm{RT}(k = 3)$ for the computation of the $k = 3$ best penalty alternatives with left-first traversal and $\mathrm{RT}(k = \infty)$ for the computation of all penalty alternatives.



(b) Average number of computed penalty alternatives for left-first traversal with $k = 4$.

Figure 2.4.3: Binary ($b = 1$), 10-bounded, 20-bounded and unbounded ($b = \infty$) knapsack problem (detailed information in Section 3.5 starting on page 82).

2.4.3 Computation of $B^{(\infty)}$

In most cases it is really easy to compute $B^{(\infty)}$ but sometimes it can be difficult or at least time-consuming.

Computation of $B^{(\infty)}$ for Knapsack Problems

It is really easy to compute $B^{(\infty)}$ for the binary, bounded or unbounded knapsack problem, because for $B^{(0)} \neq 0$ only two cases can happen. If $B^{(0)}$ uses every item, then $B^{(\infty)} = 0$ is the ∞-alternative. Otherwise, if $B^{(0)}$ does not use every item, always a disjoint solution to $B^{(0)}$ exists. In this case we just need to exclude all items used in $B^{(0)}$ and solve the remaining knapsack problem.

Computation of $B^{(\infty)}$ for the Shortest Path Problem

The well-known algorithm of Dijkstra [Dij 1959] computes an optimal solution to the shortest path problem. This algorithm can be modified easily so that the ∞-alternative instead of the optimal solution is computed. Therefor the former Dijkstra algorithm, that only tries to minimizes **one** objective function, has to be changed to a **bi-objective** version which minimizes $\operatorname{lex}\min\limits_{B \in S} (p(B), w(B))$.

Computation of $B^{(\infty)}$ for the Minimum Spanning Tree Problem

In analogy to the bi-objective Dijkstra algorithm, that we introduced above for the shortest path problem, it is possible to implement a bi-objective Prim algorithm, for example.

General Computation of $B^{(\infty)}$ for Linear Optimization Problems

For linear optimization problems it is possible to compute $B^{(\infty)}$ with the help of the simplex algorithm. For a given linear optimization problem $\min\limits_{B \in S} w'B$, also

$$\min_{B \in S} p'B \tag{2.16}$$

is a linear optimization problem. Thus we can apply the simplex algorithm to (2.16). Now only two cases can happen. Either (2.16) has a uniquely determined solution which is then $B^{(\infty)}$, or (2.16) has infinitely many optimal solutions. In the second case the set of all optimal solutions is a face of the polyhedron of feasible solutions. Then $B^{(\infty)}$ is a solution on this face with minimal value $w(B)$. Hence, one only has to determine the weights of each edge of this face and choose the best one. This is then $B^{(\infty)}$. Unfortunately, this can be time-consuming.

2.4.4 Starting with a Given Upper Bound $\hat{\varepsilon}$

As we saw in the previous Section 2.4.3, the computation of $B^{(\infty)}$ is easy in most cases. But sometimes this task can be time-consuming although the computation of an ε-penalty alternative for each $0 \leq \varepsilon < \infty$ can be done fast.

Not for that reason alone, one can save computation time by starting with $R := B^{(\hat{\varepsilon})}$ for some given $\hat{\varepsilon}$. However, one must think of a good $\hat{\varepsilon}$ if one wants to take advantage of this approach. Furthermore one has to search through the interval $(\hat{\varepsilon}, \infty)$, too, if there are not enough alternatives in the left interval $(0, \hat{\varepsilon})$.

The following observation is easily checked.

Lemma 2.4.4
Initializing Algorithm 1 with $\varepsilon_R := \hat{\varepsilon}$ and $R := B^{(\hat{\varepsilon})}$ instead of $\varepsilon_R := \infty$ and $R := B^{(\infty)}$ provides all penalty alternatives $\mathcal{P}$ and all threshold parameters $\mathcal{T}$ in the interval $[0, \hat{\varepsilon}]$.

Hence, $\hat{\varepsilon} > \varepsilon_{k-1}$ ensures that the penalty alternatives $P^{(0)}, P^{(1)}, \ldots, P^{(k-1)}$ will be found. Consequently, the right interval $(\hat{\varepsilon}, \infty)$ has not to be searched for $\hat{\varepsilon} > \varepsilon_{k-1}$.

Example 2.4.5
Initializing the algorithm of Schwarz for Example 2.4.2 with $\hat{\varepsilon} = \frac{1}{2} < \frac{2}{3} = \varepsilon_2$ and $k = 3$ would provide only two of the penalty alternatives, namely the optimal solution "sbcdeft" and the first penalty alternative "scdeft". Hence, we have to examine the interval $(0.5, \infty)$, too.
Another possibility is to restart the procedure with a larger $\hat{\varepsilon}$. For example $\hat{\varepsilon} = 0.7 > \frac{2}{3} = \varepsilon_2$ would work here.

A Simple Learning Approach for the Computation of a Good $\hat{\varepsilon}$

Lemma 2.4.4 said that

$$\hat{\varepsilon} > \varepsilon_{k-1} \tag{2.17}$$

ensures that the k best penalty alternatives $P^{(0)}, P^{(1)}, \ldots, P^{(k-1)}$ are found. But still the question arises how to choose $\hat{\varepsilon}$ such that (2.17) is fulfilled, since we do not know ε_{k-1} without starting Schwarz's algorithm. Of course one could choose $\hat{\varepsilon}$ very large in order to assure (2.17), but if you want to improve the running time it would be good to choose $\hat{\varepsilon}$ as small as possible (but still big enough). That is why we present two simple learning approaches in order to find out what a good $\hat{\varepsilon}$ is.

Therefore we preset a probability $\alpha_1 \in [0, 1]$, for example $\alpha_1 = 0.95$ and try to determine $\hat{\varepsilon}$ such that

$$P(X < \hat{\varepsilon}) = \alpha_1 \tag{2.18}$$

holds, where the random variable X is the $(k-1)$th threshold parameter ε_{k-1} in a random instance with at least $k-1$ threshold parameters. We use the notation X instead of the longer notation X_{k-1} for better readability. By definition of $\hat{\varepsilon}$ through (2.18), $\hat{\varepsilon}$ is the α_1-quantile of X. Hence, starting with $\hat{\varepsilon}$ as upper bound will be successful with probability α_1.

Since we do not know the cumulative distribution function of the $(k-1)$th threshold parameter X, we cannot compute the α_1-quantile exactly. But what we can do is to estimate the α_1-quantile from instances, where we already computed the value of X. Therefor we propose a simple estimator.

Let $x_1, x_2, \ldots, x_m$ be the values of the $(k-1)$th threshold parameter in m random instances with at least $k-1$ threshold parameters. Thus $x_1, x_2, \ldots, x_m$ are the random variates of X in m independent random experiments. Then we define quantiles $q_{k-1}(\alpha)$ for $\alpha \in [0,1]$ as in Matlab's `quantile` function [Matlab 2008].

1. Let $x_{1:m} \le x_{2:m} \le \cdots \le x_{m:m}$ be the values $x_1, x_2, \ldots, x_m$ sorted increasingly. These values are taken as the $\frac{0.5}{m}, \frac{1.5}{m}, \ldots, \frac{m-0.5}{m}$ quantiles.
$$q_{k-1}\left(\frac{0.5}{m}\right) := x_{1:m}, \quad q_{k-1}\left(\frac{1.5}{m}\right) := x_{2:m}, \quad \ldots, \quad q_{k-1}\left(\frac{m-0.5}{m}\right) := x_{m:m}$$
2. For $\frac{i-0.5}{m} < \alpha < \frac{i+0.5}{m}$ $(1 \le i < m)$ the quantile is computed with linear interpolation. Hence, we define
$$\begin{aligned} q_{k-1}(\alpha) : &= x_{i:m} + \left(\alpha - \frac{i-0.5}{m}\right) \frac{x_{i+1:m} - x_{i:m}}{\frac{i+0.5}{m} - \frac{i-0.5}{m}} \\ &= x_{i:m} + \left(\alpha - \frac{i-0.5}{m}\right) (x_{i+1:m} - x_{i:m})\, m \end{aligned}$$
for $\frac{i-0.5}{m} < \alpha < \frac{i+0.5}{m}$ and $1 \le i < m$.
3. Furthermore, we set
$$\begin{aligned} q_{k-1}(\alpha) &:= x_{1:m} && \text{for } 0 \le \alpha \le \frac{0.5}{m}\,, \\ q_{k-1}(\alpha) &:= x_{m:m} && \text{for } \frac{m-0.5}{m} \le \alpha \le 1\,. \end{aligned}$$

Example 2.4.6
Let $k := 3$. Suppose that in ten random instances the following values for the second threshold parameter $\varepsilon_2 = \varepsilon_{k-1}$ occurred.

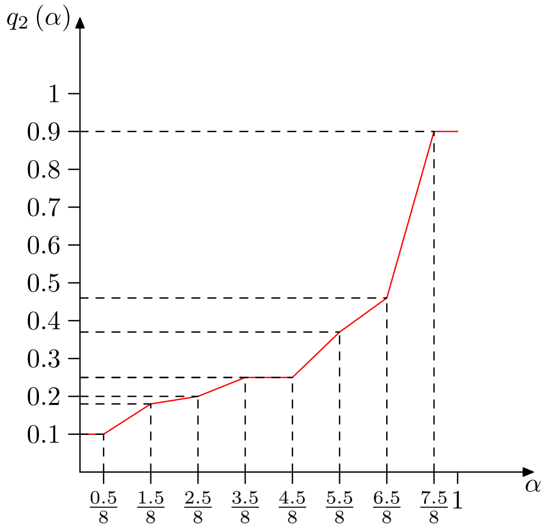
- *In two of the ten random instances, only two penalty alternatives (namely $B^{(0)}$ and $B^{(\infty)}$) existed.*
- *In the remaining eight random instances there were at least two threshold parameters. The following sorted random variates of X (values of the second threshold parameter) occurred.*

$$\begin{array}{llll} x_{1:8} = 0.10 & x_{2:8} = 0.18 & x_{3:8} = 0.20 & x_{4:8} = 0.25 \\ x_{5:8} = 0.25 & x_{6:8} = 0.37 & x_{7:8} = 0.45 & x_{8:8} = 0.90 \end{array}$$

This provides the following quantiles.

$$q_2\left(\frac{0.5}{8}\right) = q_2(0.0625) = 0.10 \qquad q_2\left(\frac{1.5}{8}\right) = q_2(0.1875) = 0.18$$
$$q_2\left(\frac{2.5}{8}\right) = q_2(0.3125) = 0.20 \qquad q_2\left(\frac{3.5}{8}\right) = q_2(0.4375) = 0.25$$
$$q_2\left(\frac{4.5}{8}\right) = q_2(0.5625) = 0.26 \qquad q_2\left(\frac{5.5}{8}\right) = q_2(0.6875) = 0.37$$
$$q_2\left(\frac{6.5}{8}\right) = q_2(0.8125) = 0.45 \qquad q_2\left(\frac{7.5}{8}\right) = q_2(0.9375) = 0.90$$

With linear interpolation and extra treatment of the margins the following quantile function $q_2(\alpha)$ for $0 \le \alpha \le 1$ arises.



Since $\frac{7.5}{8} < 0.95$ holds, the 0.95-quantile of X is $x_{m:m} = 0.9$ and because of $\frac{6.5}{8} < 0.9 < \frac{7.5}{8}$ the 0.9-quantile of X is computed by linear interpolation. It arises

$$\begin{aligned} q_2(0.9) &= x_{7:8} + \left(0.9 - \frac{6.5}{8}\right)(x_{8:8} - x_{7:8}) \cdot 8 \\ &= 0.45 + 0.0875 \cdot 0.45 \cdot 8 = 0.765 \end{aligned}$$

as 0.9-quantile of X.

Thereby we defined a simple estimator for α-quantiles of X ($\alpha \in [0,1]$). Of course the estimate is better for higher m. Therewith we can give our first simple learning algorithm.

Algorithm 3 (First Simple Learning Algorithm for the Computation of a Good $\hat{\varepsilon}$)

Initialization: *We start the algorithm with an arbitrary $\hat{\varepsilon}$ and a given probability $\alpha_1 \in [0,1]$. This could be for example $\hat{\varepsilon} := \infty$ or a smaller penalty parameter as well. Set $E_{k-1} := \emptyset$ and start the examination of random instances.*

Step 1 *Generate a random instance of the considered optimization problem type.*

Step 2 *Compute the k best penalty alternatives $P^{(0)}, P^{(1)}, \ldots, P^{(k-1)}$ and the corresponding $k-1$ smallest threshold parameters $\varepsilon_1 < \ldots < \varepsilon_{k-1}$ with Algorithm 1 and $\hat{\varepsilon}$ as upper bound. If necessary, the right interval $[\hat{\varepsilon}, \infty]$ must be searched through, too.*

Step 3 *If the instance has at least k penalty alternatives, then we insert the value of the $(k-1)$th threshold parameter into the set $E_{k-1} := E_{k-1} \cup \{\varepsilon_{k-1}\}$ and update $\hat{\varepsilon}$. Therefor $\hat{\varepsilon}$ is set to the α_1-quantile of the values in E_{k-1}.*

Otherwise, if the instance has less than k penalty alternatives, nothing has to be done in ***Step 3****.*

Repeat ***Step 1*** *-* ***Step 3*** *as many times as desired.*

In **Step 2** we have to search through the right interval $[\hat{\varepsilon}, \infty]$, too, if $\hat{\varepsilon}$ is to small. This can be very time-consuming. That is why we want to preset a second, larger bound $\hat{\varepsilon}_2$ which shall be used when the bound we tried first was to small. In this way we want to save computation time.

Hence, we preset **two** values $0 < \alpha_1 < \alpha_2 \leq 1$ and try to compute the α_1- and α_2-quantiles of the $(k-1)$th threshold parameter ε_{k-1}. Thus $\hat{\varepsilon}_1$ is an estimation of the α_1-quantile and $\hat{\varepsilon}_2$ is an estimation of the α_2-quantile of X.

Algorithm 4 (Second Simple Learning Algorithm for the Computation of a Good $\hat{\varepsilon}$)

Initialization: *We start the algorithm with* ***two*** *arbitrary bounds $\hat{\varepsilon}_1 \leq \hat{\varepsilon}_2$ and* ***two*** *given probabilities $0 < \alpha_1 < \alpha_2 \leq 1$. This could be for example $\hat{\varepsilon}_1 = \hat{\varepsilon}_2 := \infty$ or smaller penalty parameters as well. Set $E_{k-1} := \emptyset$ and start the examination of random instances.*

Step 1 *Generate a random instance of the considered optimization problem type.*

Step 2 *Try to compute the k best penalty alternatives $P^{(0)}, P^{(1)}, \ldots, P^{(k-1)}$ and the corresponding $k-1$ smallest threshold parameters $\varepsilon_1 < \ldots < \varepsilon_{k-1}$ with Algorithm 1 in the interval $[0, \hat{\varepsilon}_1]$. If this search does not provide the desired k penalty alternatives, then start the algorithm again in the interval $[\hat{\varepsilon}_1, \hat{\varepsilon}_2]$. If we still do not find enough penalty alternatives there, then the last interval $[\hat{\varepsilon}_2, \infty]$ has to be searched through, too.*

Step 3 *If the instance has at least k penalty alternatives, then we insert the value of the $(k-1)$th threshold parameter into the set $E_{k-1} := E_{k-1} \cup \{\varepsilon_{k-1}\}$*

and update $\hat{\varepsilon}_1$ and $\hat{\varepsilon}_2$. Therefore $\hat{\varepsilon}_1$ is set to the α_1-quantile and $\hat{\varepsilon}_2$ is set to the α_2-quantile of the values in E_{k-1}.

*Otherwise, if the instance has less than k penalty alternatives, nothing has to be done in **Step 3**.*

*Repeat **Step 1** - **Step 3** as many times as desired.*

The remaining question is, which quantiles α_1 and α_2 provide good computing times.

We spent some time analyzing the time saving for different quantiles (α_1, α_2) for the binary and bounded knapsack problem (with bound $b = 10$). Figures 2.4.4 (a) and (b) show the running time improvements for instances with 50 items. We tested α_1 and α_2 with an increment of 0.05. In both cases the quantiles $\alpha_1 = 0.75$ and $\alpha_2 = 0.95$ provide the largest running time improvements for the considered 1,000 random instances. It turns out, that the running time improvement increases as α_2 increases, as long as α_2 is not too big (in this case as long as $\alpha_2 < 1$). In the framed areas ($0.5 \leq \alpha_1 \leq 0.95$ and $0.55 \leq \alpha_2 \leq 1$) we also did a finer analysis with an increment of 0.01 for α_1 and α_2. These results are shown in Figures 2.4.5 (a) and (b).

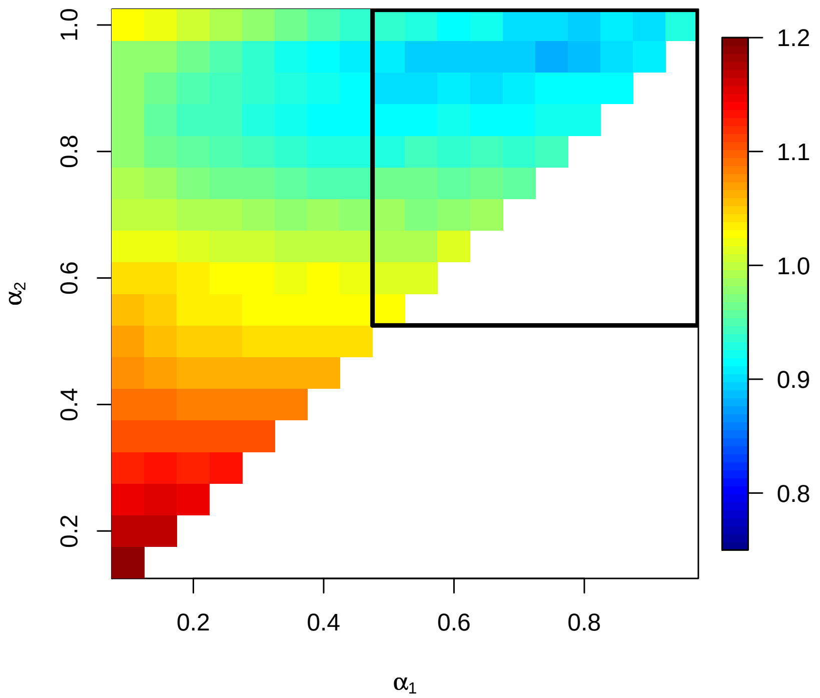
Besides the shown results for instances with 50 items and bounds $b \in \{1, 10\}$ we also tried different (α_1, α_2) for $n \in \{5, 10, 25, 50, 75, 100\}$ and $b \in \{1, 10, 20\}$. As a result we recommend the empirical values

$$\alpha_1 = 0.66 \quad \text{and} \quad \alpha_2 = 0.96\,.$$

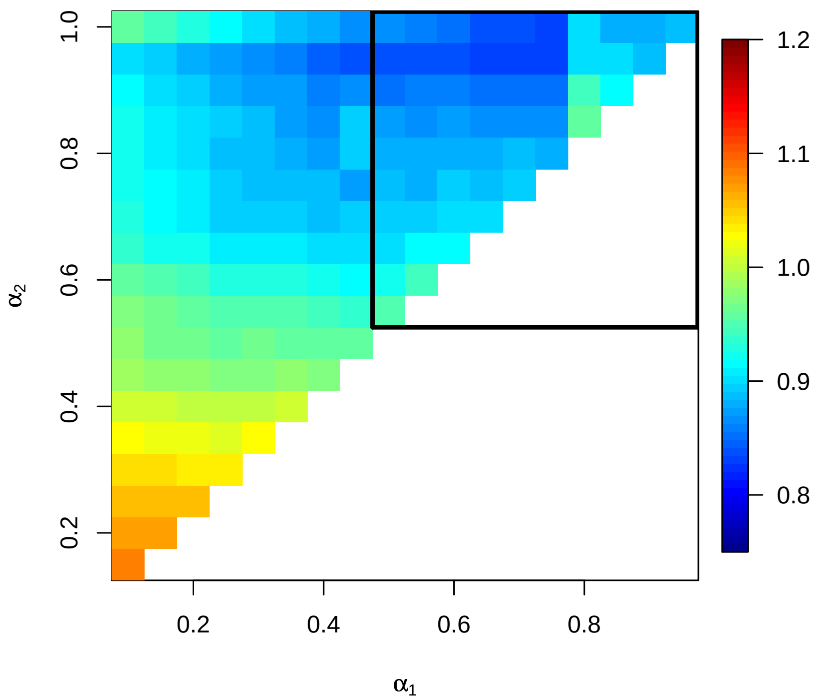
Experimental Results for Knapsack Problems

Figure 2.4.6 on page 62 shows the running time improvements achieved by the learning approach from Algorithm 4 on page 58 with the empirically determined quantiles $\alpha_1 = 0.66$ and $\alpha_2 = 0.96$. For the b-bounded knapsack problem with $b < \infty$ running time improvements of up to 23% were measured.

But for the unbounded knapsack problem ($b = \infty$) the learning approach can lead to higher computation times! This is due the fact, that left-first traversal for the unbounded knapsack problem does not leave much room for more running time improvements as the previous Figure 2.4.3 (a) on page 53 showed. For example, if left-first traversal computes 5 instead of the searched for 4 penalty alternatives, then an optimal starting interval ceiling $\hat{\varepsilon}$ can save at most further 20% of running time. Thus, in the case of the unbounded knapsack problem, learning of an optimal $\hat{\varepsilon}$ hardly speeds up the running time. In fact, this learning approach might even lead to bigger computations times.

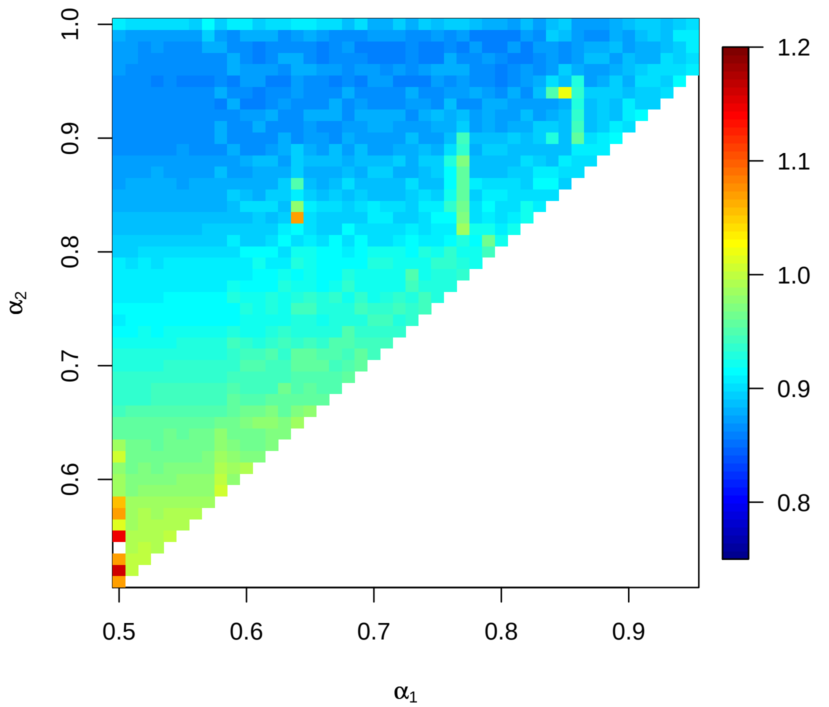


(a) Binary knapsack problem ($b = 1$) with $n = 50$ items.

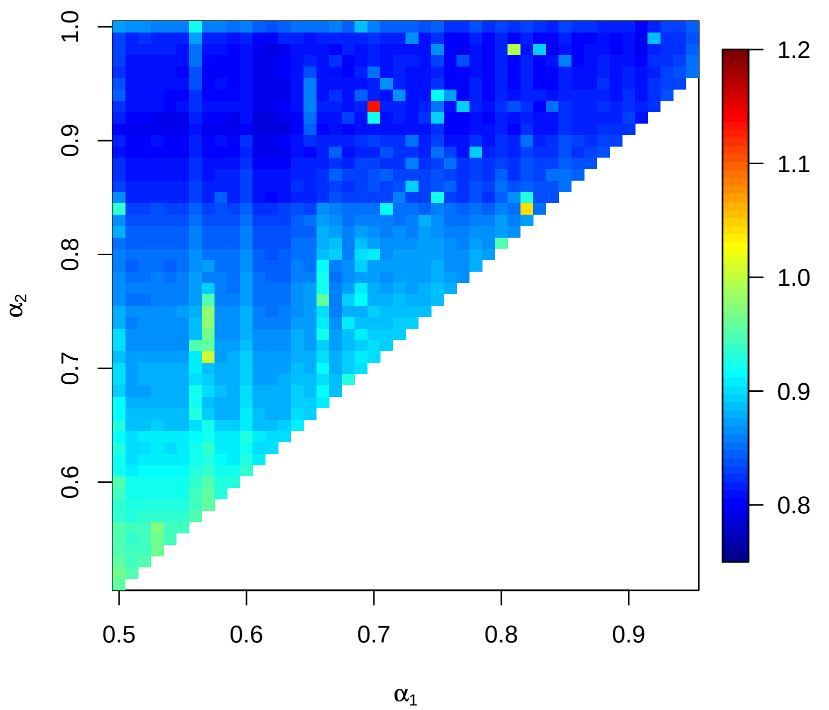


(b) Bounded knapsack problem with $n = 50$ items and bound $b = 10$.

Figure 2.4.4: Running time improvements with the simple learning approach from Algorithm 4 for different quantiles $0 < \alpha_1 < \alpha_2 < 1$ **with an increment of 0.05**.



(a) Binary knapsack problem ($b = 1$) with $n = 50$ items.



(b) Bounded knapsack problem with $n = 50$ items and bound $b = 10$.

Figure 2.4.5: Running time improvements with the simple learning approach from Algorithm 4 for different quantiles $0 < \alpha_1 < \alpha_2 < 1$ **with an increment of 0.01**.

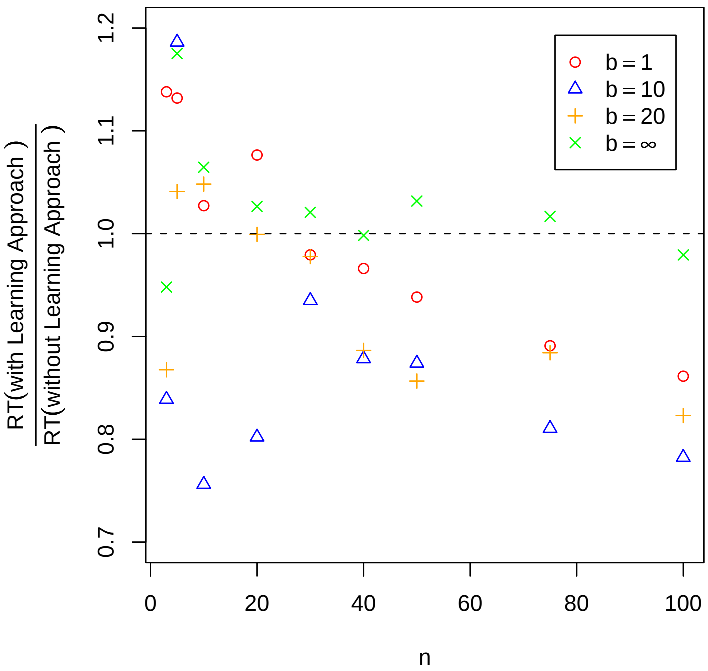


Figure 2.4.6: b-bounded knapsack problem: Saving achieved by the Learning Approach with $\alpha_1 = 0.44$ and $\alpha_2 = 0.96$.

Experimental Results for the Shortest Path Problem in Real Road Networks

In order to get an idea if the learning approach could speed up the computations we again considered the number of alternatives that left-first traversal computed. If this number is near the predetermined number k, then it is not necessary to compute a good $\hat{\varepsilon}$. In that case starting with an upper bound $\hat{\varepsilon}$ could even increase the running time as we saw in case of the unbounded knapsack problem.

Thus, we counted the average number of computed penalty alternatives for the shortest path problem in real road networks (cf. Section 3.3 starting on page 74). Figure 2.4.7 shows this number for each US-State. We see that there are two outstanding road maps – Alaska (AK) with averaged 4 and Hawaii (HI) with 2 computed alternatives. This is due to the fractured road networks that we find there. In Hawaii this is because of the islands and in Alaska we have many cities that are not served by road, sea or river but reachable by train. Thus planning a car route with penalty alternatives will never deliver many alternatives in such regions.

But, the number of computed penalty alternatives for the remaining US-States is also not very big. Hence, we could try to apply the learning approach here, but it will probably reduce the running time only a little.

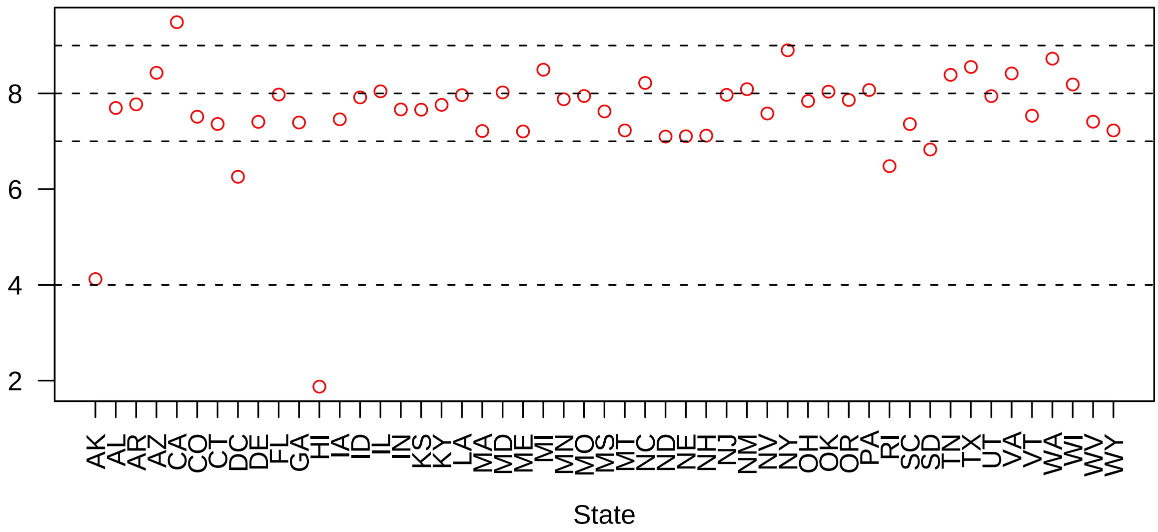


Figure 2.4.7: Average number of computed penalty alternatives with left-first traversal for $k = 3$ and each US-State.

2.4.5 Using the Solutions *L* or *R* as Initial Solution

A couple of optimization problems are solved with branch and bound techniques or other algorithms which use initial solutions. Some examples for problems solved by branch and bound algorithms are:

- Knapsack problem
- Integer and nonlinear programming
- Traveling salesman problem
- Cutting stock problem
- Quadratic assignment problem
- Maxiumum satisfiability problem.

Observe that it makes no differences whether L or R is used as initial solution, since both solutions have the same functional value in the optimization problem

$$\min_{B \in S} w(B) + \bar{\varepsilon} p(B) \,.$$

Experimental Results for Knapsack Problems

Since we used branch and bound techniques to solve our knapsack problems, it was possible to use L or R as initial solution. Figure 2.4.8 shows that an initial solution indeed can improve the running time, but not so heavy. The biggest improvement is obtained for $b = \infty$ where we can save up to 15% running time.

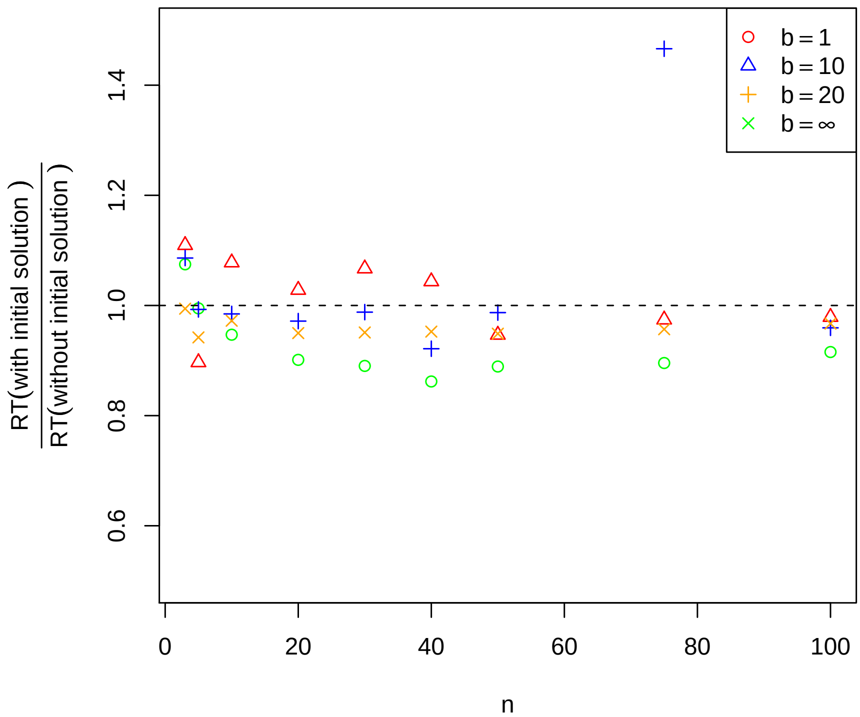


Figure 2.4.8: b-bounded knapsack problem: Saving achieved by using L (or R) as initial solution.

2.4.6 A More Stable Way to Evaluate (2.13)

An important part of Algorithm 1 on page 42 is the decision on whether $\bar{\varepsilon}$ is a new threshold parameter or not. In the algorithm the criterion for decision is whether equation (2.13)

$$f_{\bar{\varepsilon}}\left(B^{(\bar{\varepsilon})}\right) = f_{\bar{\varepsilon}}(L) = f_{\bar{\varepsilon}}(R) \tag{2.13}$$

holds or not. Unfortunately, this is not always easy to decide because of numerical instabilities. In fact, it is not advisable to rely on the numerical established values $f_{\bar{\varepsilon}}\left(B^{(\bar{\varepsilon})}\right)$, $f_{\bar{\varepsilon}}(L)$, and $f_{\bar{\varepsilon}}(R)$.

If $B^{(\bar{\varepsilon})} = L$ or $B^{(\bar{\varepsilon})} = R$ holds, which could be decided easily for optimization problems with integer solutions, then $\bar{\varepsilon}$ is a threshold parameter and (2.13) is fulfilled obviously. Otherwise, for $B^{(\bar{\varepsilon})} \neq L$ and $B^{(\bar{\varepsilon})} \neq R$, we still do not know whether $\bar{\varepsilon}$ is a threshold parameter or not, unfortunately. So we have to look for another criterion.

By the definition of $B^{(\bar{\varepsilon})}$ the inequality

$$f_{\bar{\varepsilon}}\left(B^{(\bar{\varepsilon})}\right) \leq f_{\bar{\varepsilon}}(B) \quad \text{holds for all } B \in S\,.$$

Thus negation of criterion (2.13) is equivalent to

$$f_{\bar{\varepsilon}}\left(B^{(\bar{\varepsilon})}\right) < f_{\bar{\varepsilon}}(L) = f_{\bar{\varepsilon}}(R)\,.$$

If one uses L or R as initial solution (as described in the previous Section 2.4.5), then the decision whether $B^{(\bar{\varepsilon})}$ is better than L and R can be left to the solution algorithm that computes the $\overline{\varepsilon}$-alternative. This algorithm should now return not only $B^{(\overline{\varepsilon})}$ but also a boolean `improved` that indicates whether a better solution than the initial solution (L or R) was found or not. Of course computational accuracy is still a matter within this solving procedure. Thus it is still recommended to check that $B^{(\overline{\varepsilon})}$ indeed differs from L and R for integer programs.

Furthermore, from Theorem 2.3.10' we know that for a new penalty alternative $B^{(\overline{\varepsilon})}$

$$p(L) < p\left(B^{(\overline{\varepsilon})}\right) < p(R)$$

and

$$w(L) \leq w\left(B^{(\overline{\varepsilon})}\right) < w(R)$$

hold. Thereby equality of $w(L)$ and $w\left(B^{(\overline{\varepsilon})}\right)$ is only possible for $L = B^{(0)}$.

Summarizing, we get the following lemma which shows three properties which can be easily verified. For programmers it is strongly recommended to check each property in order to make a reliable decision whether $B^{(\overline{\varepsilon})}$ is a new penalty alternative or not. This following lemma is basically just a compilation of earlier results. (i) is obviously clear and (ii) and (iii) are parts of Theorem 2.3.10'.

Lemma 2.4.7
Let $B^{(\overline{\varepsilon})}$ be a newly found penalty alternative that is to say

$$f_{\overline{\varepsilon}}\left(B^{(\overline{\varepsilon})}\right) < f_{\overline{\varepsilon}}(L) = f_{\overline{\varepsilon}}(R). \tag{2.19}$$

Then the following three statements hold.

(i) $B^{(\overline{\varepsilon})} \neq L$ *and* $B^{(\overline{\varepsilon})} \neq R$.

(ii) $p(L) < p\left(B^{(\overline{\varepsilon})}\right) < p(R)$.

(iii) $w(L) \leq w\left(B^{(\overline{\varepsilon})}\right) < w(R)$. *Equality of* $w(L)$ *and* $w\left(B^{(\overline{\varepsilon})}\right)$ *is only possible for* $L = B^{(0)}$.

Still, if (i), (ii), and (iii) are fulfilled one cannot be sure that (2.19) holds, without consideration of $f_{\overline{\varepsilon}}\left(B^{(\overline{\varepsilon})}\right), f_{\overline{\varepsilon}}(L)$, and $f_{\overline{\varepsilon}}(R)$. That is why we also make suggestions how to compute these three values. Since $\overline{\varepsilon}$ is internally saved as a rounded value, it is recommended to insert the exact representation of $\overline{\varepsilon}$

$$\overline{\varepsilon} = \frac{w(R) - w(L)}{p(L) - p(R)} \tag{2.20}$$

into (2.19). We transform

$$\begin{aligned}
& & f_{\overline{\varepsilon}}\left(B^{(\overline{\varepsilon})}\right) &= f_{\overline{\varepsilon}}(L) \\
&\Leftrightarrow & w\left(B^{(\overline{\varepsilon})}\right) + \overline{\varepsilon} \cdot p\left(B^{(\overline{\varepsilon})}\right) &= w(L) + \overline{\varepsilon} \cdot p(L) \\
&\overset{(2.20)}{\Leftrightarrow} & w\left(B^{(\overline{\varepsilon})}\right) + \frac{w(R) - w(L)}{p(L) - p(R)} \cdot p\left(B^{(\overline{\varepsilon})}\right) &= w(L) + \frac{w(R) - w(L)}{p(L) - p(R)} \cdot p(L) \\
&\Leftrightarrow & \left(w\left(B^{(\overline{\varepsilon})}\right) - w(L)\right) \cdot (p(L) - p(R)) &= (w(R) - w(L)) \cdot \left(p(L) - p\left(B^{(\overline{\varepsilon})}\right)\right).
\end{aligned} \tag{2.21}$$

For integer vectors $w, p \in \mathbb{Z}^n$ the final equation (2.21) can be verified numerically stable as long as the factors in (2.21) are not too large.

2.4.7 Summary of the Experimental Results

The previous experimental results on the running times showed that left-first traversal is the most powerful approach in order to reduce the computation time. In particular, left-first traversal will never increase the runtime. In case of the simple learning approach for learning appropriate initial bounds $\hat{\varepsilon}$, even enlargements of the running times were observed. Thus, it is recommended to think about whether this approach might be useful for the given optimization problem. Most often the learning approach really reduces the running time but the observed running time improvements were not as big as for left-first traversal. The same phenomenon occurred for the usage of L (or R) as initial solutions. Here, running time improvements occurred, but again only minor improvements arise.

Thus, we recommend to always implement left-first traversal. Furthermore, for optimization problems solved by branch and bound algorithms L or R should be used as initial solution. But we suggest to implement the learning approach only, if the distribution of the $(k-1)$th threshold parameter ε_{k-1} is of interest, too. Otherwise the effort is not worth it.

Chapter 3

Experimentally Observed Cordel Frequencies under the Penalty Selection Rule

In this chapter we present experimentally observed Cordel frequencies for shortest path problems (Sections 3.1, 3.2, and 3.3), the minimum spanning tree problem (Section 3.4) and for different knapsack problems (Section 3.5). Furthermore, we try to design instances where the penalty alternatives $P^{(0)}, P^{(1)}, \ldots, P^{(k)}$ for penalization with the canonical penalty vector fulfill

$$w\left(P^{(i)}\right) - w\left(P^{(i-1)}\right) = d_i \quad \text{for } i = 1, \ldots, k$$

with a given difference vector $d = (d_1, \ldots, d_k) \in \mathbb{R}^k_{>0}$. Obviously, the Cordel frequency cannot be 0% or 100%, if there exist instances for any difference vector $d \in \mathbb{R}^k_{>0}$. That is why we tried to give specifications for the construction of instances to a given difference vector d. Unfortunately, we were not able to do so for each of the considered problems but for some types of optimization problems our investigations were successful.

3.1 The Shortest Path Problem in Grid Graphs

We start with the shortest path problem in directed grid graphs.

Definition 3.1.1 (Directed Grid Graph)
A ***weighted directed m × n grid graph with height m and width n*** *is a graph $G = (V, E)$ with*

$$V = \{v_{i,j} : 1 \leq i \leq m, 1 \leq j \leq n\}$$

and

$$\begin{aligned} E = \; & \left\{(v_{i,j}, v_{i+1,j}) : 1 \leq i \leq m-1, 1 \leq j \leq n\right\} && \textit{(vertical edges)} \\ \cup \; & \left\{(v_{i,j}, v_{i,j+1}) : 1 \leq i \leq m, 1 \leq j \leq n-1\right\} && \textit{(horizontal edges)} \end{aligned}$$

and a weight function $w : E \to \mathbb{R}$. Furthermore we have two special vertices $s = v_{1,1}$ and $t = v_{m,n}$ which are the starting and the target node.

Thus the graph can be illustrated as a rectangular $m \times n$ grid, where each intersection point represents a vertex of G. Furthermore from each vertex there exist edges to its right and lower neighbors, if such neighbors exist. Figure 3.1.1 shows a 3×4 grid graph as an example.

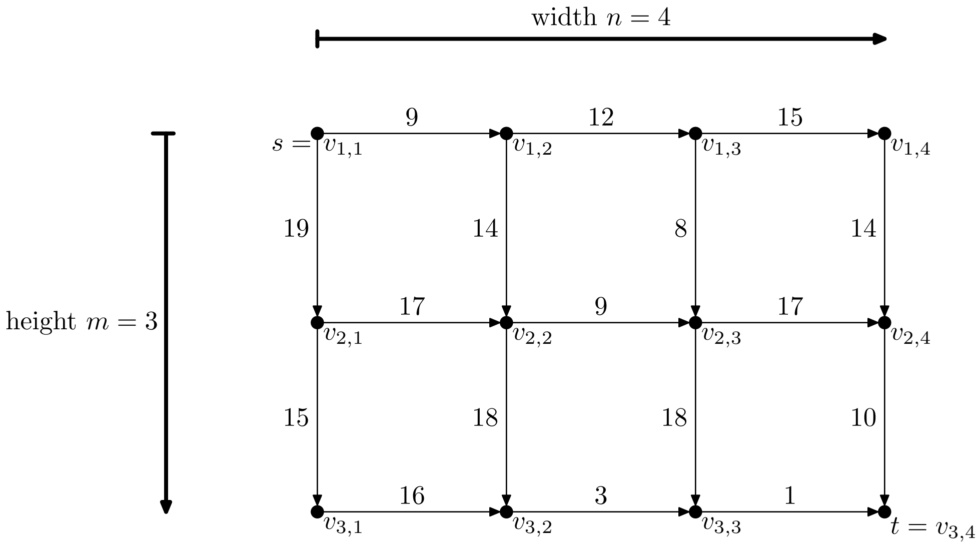


Figure 3.1.1: Example for a weighted directed 3×4 grid graph.

Grid graphs have one main advantage. By the use of dynamic programming it is possible to compute the shortest $s-t$ path in $\mathcal{O}(|V|)$. This procedure is substantially faster than Dijkstra's algorithm [Dij 1959]. Note that the running time of Dijkstra's algorithm with Fibonacci heaps is $\mathcal{O}(|E| + |V| \log |V|)$.

3.1.1 Construction of Grids for the Shortest Path Problem with Given Differences $\boldsymbol{d}$ and Threshold Parameters $\boldsymbol{\varepsilon}$

First, we show that for each difference vector $d = (d_1, \ldots, d_k) \in \mathbb{R}^k_{>0}$ and for each threshold parameter vector $\varepsilon = (\varepsilon_1, \ldots, \varepsilon_k)$ with $0 < \varepsilon_1 < \ldots < \varepsilon_k$ exists a directed $2 \times (k+1)$ grid graph, whose penalty-alternatives $P^{(0)}, P^{(1)}, \ldots, P^{(k)}$ fulfill

$$w\left(P^{(i)}\right) - w\left(P^{(i-1)}\right) = d_i \quad \text{for } i = 1, \ldots, k$$

and have the threshold parameters $\varepsilon_1, \ldots, \varepsilon_k$. It is therefore clear that the Cordel frequency cannot be 0 or 1.

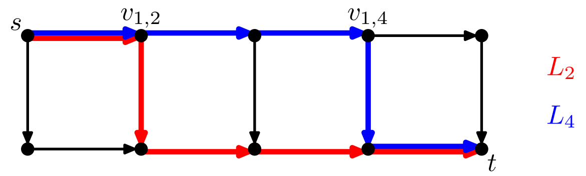
To prove this proposition we give an algorithm that constructs an appropriate $2 \times (k+1)$ grid graph. A $2 \times (k+1)$ grid has the structure of a horizontal ladder and each $s-t$ path can uniquely be described by its vertical edge.

Definition 3.1.2 (Ladder Graph, Ladder Path, Ladder Spoke)
Regarding its ladder shape a $2 \times n$ grid is called a ***ladder graph of width n****. For a ladder graph of witdh n we denote by L_i the $s-t$ path*

$$L_i := \{s = \underbrace{v_{1,1} - v_{1,2} - \ldots - v_{1,i}}_{\text{straightly to the right}} - \underbrace{v_{2,i} - v_{2,i+1} - \ldots - v_{2,n}}_{\text{straightly to the right}} = t\}$$

which turns downwards in vertex $v_{1,i}$. L_i is called the i-th ***ladder path****.*
Furthermore the i-th vertical edge $(v_{1,i}, v_{2,i})$ is called the i-th ***ladder spoke****.*

The following picture shows the two ladder paths L_2 (red) and L_4 (blue) in a 2×5 grid.



Our idea is to construct a $2 \times (k+1)$ grid, such that the penalty alternative $P^{(i)}$ is the ladder path L_{i+1}. Thus $P^{(0)}$ should be the leftmost ladder path and $P^{(k)}$ should be the rightmost ladder path.

Theorem 3.1.3 (Penalty Alternatives for Ladder Graphs)
Consider a given difference vector $d \in \mathbb{R}^k_{>0}$, a given threshold parameter vector $\varepsilon = (\varepsilon_1, \ldots, \varepsilon_k)$ with

$$0 < \varepsilon_1 < \ldots < \varepsilon_k$$

and a ladder path of width $k+1$ with the following weights as shown in Figure 3.1.2 on the next page.

1. *The first ladder spoke should have weight 1 and the i-th ladder spoke $(1 < i \leq k+1)$ should have weight $1 + d_1 + \cdots + d_{i-1}$.*
2. *The vertical edges $(v_{1,i}, v_{1,i+1})$ and $(v_{2,i}, v_{2,i+1})$ should have weight $\frac{d_1}{\varepsilon_1} - 1$ for $i = 1$ and weight $\frac{d_i}{\varepsilon_i}$ for $i = 2, \ldots, k$.*

Then the following two statements hold:

1. *The penalty alternatives $B^{(0)} = P^{(0)}, P^{(1)}, \ldots, P^{(k)} = B^{(\infty)}$ are the ladder paths $L_1, \ldots, L_{k+1}$ with*

$$P^{(i)} = L_{i+1} \quad \text{for } i = 0, \ldots, k\,.$$

2. *The penalty alternatives have the given weight differences $d = (d_1, \ldots, d_k)$ and threshold parameters $\varepsilon = (\varepsilon_1, \ldots, \varepsilon_k)$. This means that*

$$d_i = w\left(P^{(i)}\right) - w\left(P^{(i-1)}\right)$$
$$\text{and} \quad \varepsilon_i = \frac{w\left(P^{(i)}\right) - w\left(P^{(i-1)}\right)}{p\left(P^{(i-1)}\right) - p\left(P^{(i)}\right)}$$

hold for $i = 1, \ldots, k$.

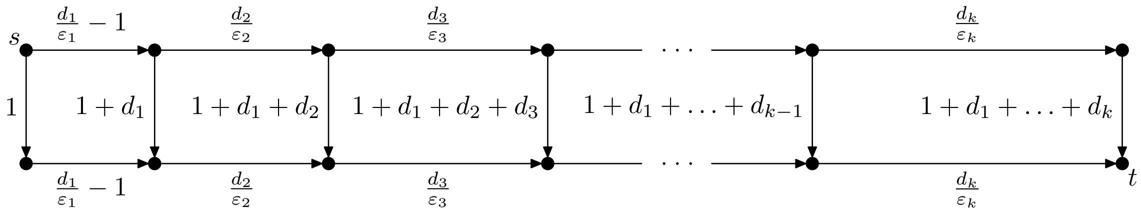


Figure 3.1.2: Construction scheme for a $2 \times (k+1)$ grid graph with penalty alternatives $P^{(0)}, \ldots, P^{(k)}$ with threshold parameters $\varepsilon_1, \ldots, \varepsilon_k$ and differences $d_i = w\left(P^{(i)}\right) - w\left(P^{(i-1)}\right)$ for $i = 1, \ldots, k$.

Proof. Since $d > 0$, the first ladder path L_1 with length $\frac{d_1}{\varepsilon_1} + \ldots + \frac{d_k}{\varepsilon_k}$ is a shortest $s-t$ path. Hence, for penalization with the canonical penalty vector and $B^{(0)} = L_1$, the i-th ladder path L_i $(1 < i \leq k+1)$ has the weight

$$w(L_i) = \sum_{j=1}^{k} \frac{d_j}{\varepsilon_j} + \sum_{j=1}^{i-1} d_j$$

and the penalized part

$$p(L_i) = \sum_{j=i}^{k} \frac{d_j}{\varepsilon_j}\,.$$

Thus, the ladder paths have the requested weight differences

$$w(L_{i+1}) - w(L_i) = d_i \quad \text{for } i = 1, \ldots, k\,.$$

With

$$p(L_i) - p(L_{i+1}) = \frac{d_i}{\varepsilon_i}$$

follows

$$\frac{w(L_{i+1}) - w(L_i)}{p(L_i) - p(L_{i+1})} = \frac{d_i}{\frac{d_i}{\varepsilon_i}} = \varepsilon_i\,.$$

Since all $s-t$ paths are ladder paths and the specified threshold parameters fulfill $0 \leq \varepsilon_1 < \ldots < \varepsilon_k$, the ladder paths $L_1, \ldots, L_k$ are the penalty alternatives with the requested weight differences d and threshold parameters ε. ■

Remark 3.1.4
If only a difference vector d is given, the vector of threshold parameters ε can be freely chosen with $0 < \varepsilon_1 < \ldots < \varepsilon_k$, for example $\varepsilon = (1, \ldots, k)$. Otherwise, if only a vector of threshold parameters is given, then $d > 0$ can be arbitrarily chosen, for example $d = (1, \ldots, 1)$.

Consequently, the Cordel frequency for a random $2 \times n$ grid $(n \geq 3)$ cannot be 0 or 1.

3.1.2 Experimental Results for the Shortest Path Problem in Grid Graphs

Before we present the experimentally observed Cordel frequencies, we explain how we generated random instances.

Definition 3.1.5 (Random Instances for the Shortest Path Problem in Directed Grid Graphs)
A ***random instance of height*** $m > 2$*,* ***width*** $n > 2$*,* ***and range*** $g \in \mathbb{N}$ *is a directed* $m \times n$ *grid with randomly uniformly distributed edge weights in the integer range* $[1, 10^g]$.

With the range parameter g we can control the range of edge weights. Huge values of g imply quasi-continuous instances, whereas small values imply discrete instances with many recurring edge weights.
The impact of the range parameter g on the Cordel frequencies was analyzed for every studied optimization problem. In fact, g has usually only a small impact on the Cordel frequency, as the following experimental results show. But some types of optimization problems occurred, where g heavily influences the Cordel frequency.

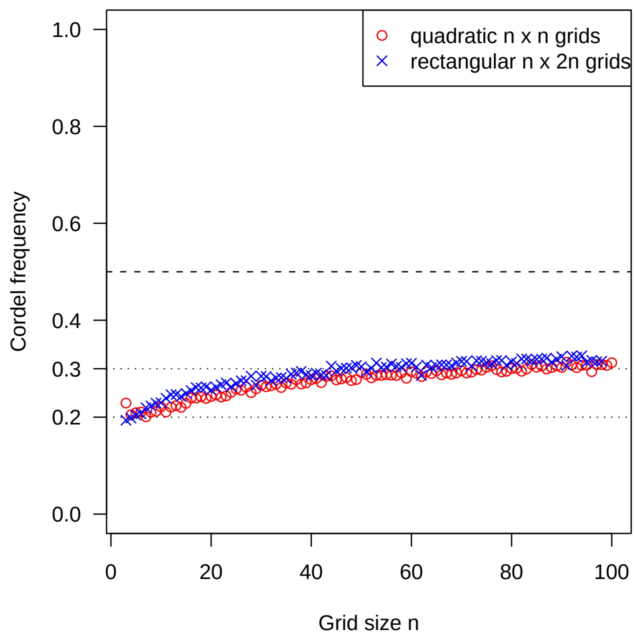


Figure 3.1.3: Experimentally observed Cordel frequencies $P(d_1 \geq d_2)$ for the shortest path problem in quadratic $n \times n$ and in rectangular $n \times 2n$ grid graphs with range parameter $g = 3$.

For the shortest path problem two types of grids were examined: quadratic $n \times n$ grids and rectangular $n \times 2n$ grids. For each triple (m, n, g) at least 10,000 random instances were generated and analyzed. As an example, here, we show the results for $g = 3$. The red circles in Figure 3.1.3 show the Cordel frequencies for the shortest path problem in **quadratic $n \times n$ grids**. The blue crosses show the Cordel frequencies for **rectangular $n \times 2n$ grids**. It turns out that both frequencies are nearly the same, which is not surprising. While the Cordel frequencies for $n \leq 100$ are approximately between 20% and 30%, they seem to converge to approximately $32\% - 35\%$ as $n \to \infty$.

For the sake of completeness, Figure J.1.1 (quadratic grids, appendix page 212) and Figure J.1.2 (rectangular grids, appendix page 213) show the experimentally observed Cordel frequencies for the shortest path problem for all considered range parameters $g \in \{2, \ldots, 7\}$. In addition, besides the Cordel frequency $P(d_1 \geq d_2)$ all graphs also contain the probability $P(d_2 \geq d_3)$. It turns out that the graphs for the six range parameters $g \in \{2, \ldots, 7\}$ do not differentiate greatly.

As a result, we state that the penalty alternatives for the shortest path problem in directed grid graphs fulfill the generalized Cordel property (GeCoP) roughly in every fourth or third random grid. Hence this is no prominent example for (GeCoP).

3.2 The Shortest Path Problem in Trellises

Besides the directed grid graphs a second type of graphs was studied.

Definition 3.2.1 (Trellis)
*An **(m, n) trellis with height m and width n** consists of a start vertex s, a target vertex t and $m \times n$ vertices in between, which are arranged in n columns and m rows. From each vertex, m directed edges lead to each vertex in the rightward column. Furthermore, m vertices lead from the start vertex s to each vertex in the first (leftmost) column and m vertices from each vertex in the last (rightmost) column lead to the target node t.*

*(n, n) trellises are called **quadratic** trellises.*

The following Figure 3.2.1 shows a $(3, 4)$ trellis without edge weights.

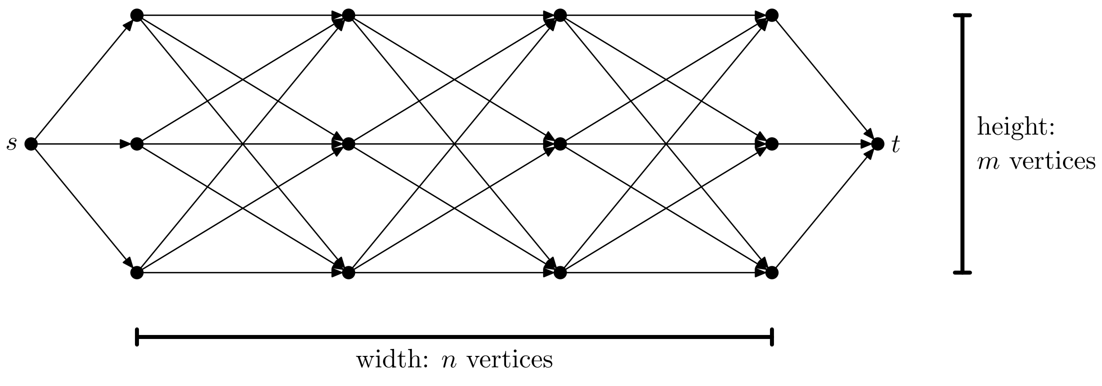


Figure 3.2.1: A $(3, 4)$ trellis (height 3, width 4).

Remark 3.2.2
Similar graphs occur in telecommunication theory, or more precisely in trellis modulation theory (cf. [Ung 1982] and [Lan 2004]).

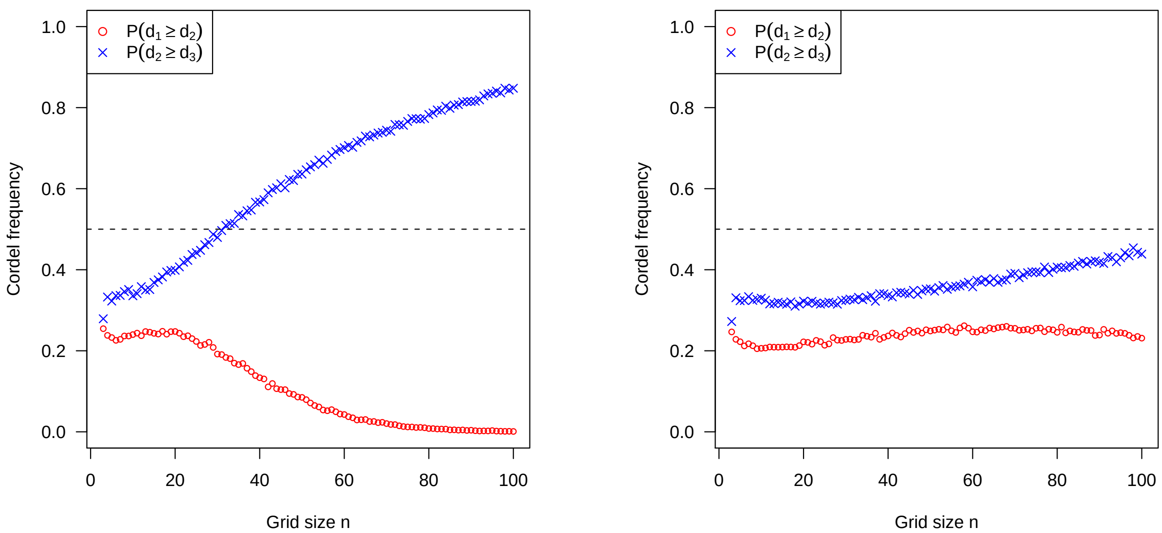
3.2.1 Experimental Results for the Shortest Path Problem in Trellises

Definition 3.2.3 (Random Instances for the Shortest Path Problem in Trellises)
A ***random instance of size*** (m, n) ***and range*** $g \in \mathbb{N}$ *is a* (m, n) *trellis with randomly uniformly distributed edge weights in the integer range* $[1, 10^g]$.

For edge weights between 1 and 100 ($g = 2$) it is highly probable that two equally best $s - t$ paths exist. In such cases

$$w\left(P^{(0)}\right) = w\left(P^{(1)}\right) = \min_{B \in S} w(B)$$

and consequently $d_1 = 0 \leq d_2$ hold. That is why the Cordel frequency is tending to 0% for $n \to \infty$. Along the same line of reasoning it follows that the Cordel frequency converges to 0% for each finite range parameter $g \in \mathbb{N}$. Figure 3.2.2 shows the Cordel frequencies for quadratic $n \times n$ trellises with range parameters $g = 2$ (left) and $g = 3$ (right). For $g = 2$ the limit 0% turns out very clearly and for $g = 3$ at least the frequencies are already slightly falling.



(a) $g = 2$ **(b)** $g = 3$

Figure 3.2.2: Cordel frequencies for **quadratic** $\boldsymbol{n \times n}$ trellis graphs.

Again, the complete results for all considered range parameters $g \in \{2, \ldots, 7\}$ can be found in the appendix. The following table shows where to find which graph.

	reference to the appendix
quadratic $n \times n$ trellises	Figure J.2.1, page 214
rectangular $m \times 2m$ trellises	Figure J.2.3, page 216
rectangular $2n \times n$ trellises	Figure J.2.5, page 218

As in the case of directed grid graphs, there was not much difference between the Cordel frequencies for quadratic and rectangular trellises. We summarize the results for quadratic $n \times n$ and for rectangular $m \times 2m$ and $2n \times n$ trellises:

1. The results for the three considered sizes (quadratic $n \times n$, rectangular $m \times 2m$, and rectangular $2n \times n$ trellises) are very similar.
2. The Cordel frequency for the shortest path problem under the penalty selection rule in trellises is around $20\% - 25\%$ if n is not too large.
3. For each $g \in \mathbb{N}$ the Cordel frequency has to converge to 0 for $n \to \infty$.

3.3 The Shortest Path Problem in Road Networks with Real Travel Times

3.3.1 Obtaining and Preparing Real Datasets

Even more interesting than examining the Cordel frequency in directed grid graphs or trellises, is the analysis of real road networks. For that purpose good datasets are needed.

On the homepage of the ninth DIMACS Implementation Challenge [DIMACS] one can download the road networks of the 50 US States and the District of Columbia as undirected graphs. For each edge the travel times, the spatial distance and road category is included. There, the travel time is the spatial distance divided by some average speed, which depends on the road category (table taken from [DIMACS]).

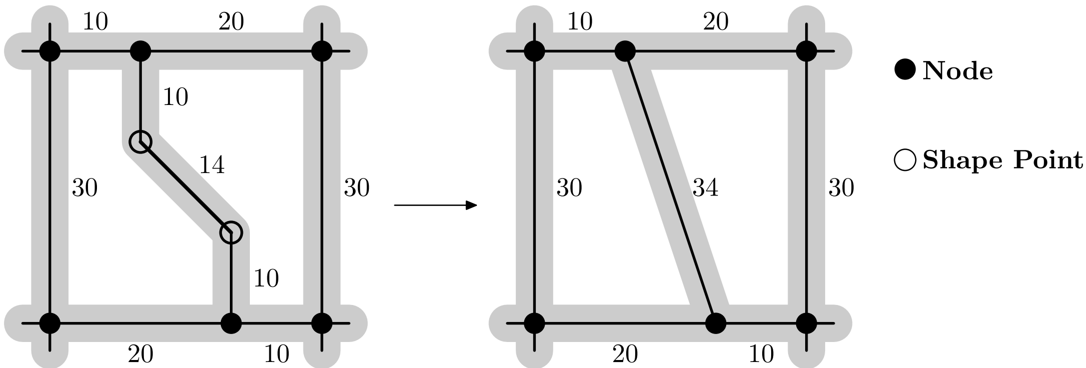
Category code	Category name	Average speed
11	primary highway with limited access (e.g. interstates)	1.0
21	primary road without limited access (e.g. US highways)	0.8
31	secondary and connecting road (e.g. state highways)	0.6
41	local, neighborhood, and rural road	0.4

With the likewise downloadable Perl script tiger2edimacs.pl the TIGER/Line files can be converted into the Standard Challenge 9 Format of the implementation challenge. By using the option `-T` the edge weights are set to the rounded off travel times.

Given such a real road network we want to determine the Cordel frequency for the penalty alternatives to this shortest path problem. Therefore we have to choose a random start and target vertex. But before we can choose random start and target nodes, we need to understand the representation of real road networks in the TIGER/Line files. Here, the following concept of nodes and shape points is essential.

Definition 3.3.1 (Shape Points and Nodes, [USCB, pp. 1-8])
Shape points *describe the position and curvature of a street but are not required to describe the topology of the road network. In contrast* ***nodes*** *are vertices that describe the topology of the graph.*

The following Figure 3.3.1 (a) illustrates the difference between nodes and shape points.



(a) Original graph representation.

(b) Graph representation without ambiguous shape points.

Figure 3.3.1: An example road network with it's two graph representations.

For the analysis of shortest path problems with randomly chosen start and target vertices s and t it is problematic to have shape points. This is since the number of shape points on a road segment affects the probability of this segment to be chosen as start or target segment. In order to reduce the effects of different representations we thinned out the graphs and cut out all shape points and redundant edges as shown in Figure 3.3.1 (b).

Table J.3.2 in the appendix on page 220 shows the size of the 51 road maps of the 50 US-States and the District of Columbia before and after the thinning out.

3.3.2 Experimental Results for the Shortest Path Problem in Road Networks with Real Travel Times

Definition 3.3.2 (Random Instances for the Shortest Path Problem in Real Road Networks)
A ***random instance of a road network*** *consists of the thinned out road network without shape points and two independent, uniform random vertices (starting node s and target node t with $s \neq t$) which are connected in the road network.*

For these thinned out road maps we determined the Cordel frequencies. Figure 3.3.2 shows the ten largest and lowest Cordel frequencies for the road maps of the 50 US-States and the District of Columbia. In addition, Figure J.3.1 in the appendix on page 220 shows all experimentally observed Cordel frequencies with confidence intervals.

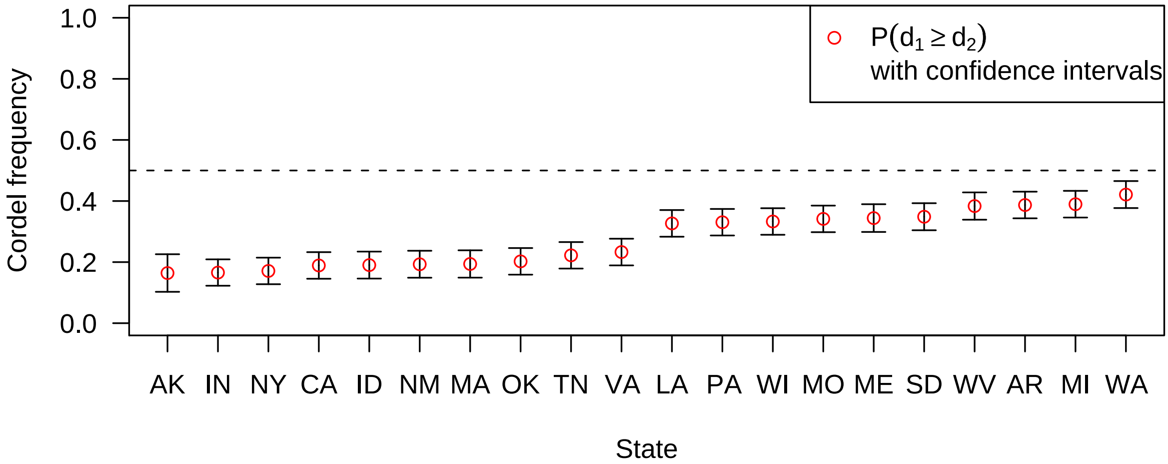


Figure 3.3.2: The ten largest and lowest Cordel frequencies $P(d_1 \geq d_2)$ under the penalty selection rule for the shortest path problem in TIGER/Line road maps of the US-States. See Table J.3.2 in the appendix on page 220 for a list of abbreviations.

As Figure 3.3.2 above and Figure J.3.1 in the appendix on page 220 show, the Cordel frequencies for the considered road networks vary between 15% and 42%. In some states (for example Alaska) the Cordel frequency is, with approximately 16%, exceptionally small and in other states (for example Arizona, Maine and Washington) the Cordel frequency is, with approximately 40%, much larger, but still considerably smaller than 50%.

An interesting field for future research could be to examine whether simple properties of the given graphs influence the Cordel frequency. Such properties could be for example the number of vertices and edges or the maximal or average node degree.

3.4 The Minimum Spanning Tree Problem

After these extensive investigations of the Cordel frequency for the shortest path problem on three different graph classes (grid graphs, trellises and road networks), we determine the Cordel frequency also for the minimum spanning tree problem. In contrast to the previous shortest path problem, now only one graph class is considered. Before we present the obtained results we define the optimization problem and construct instances for a given difference vector d.

Definition 3.4.1 (Minimum Spanning Tree)
Consider an undirected, connected Graph $G = (V, E)$ and a weight function $w : E \to \mathbb{R}$ on the edges of G.
Then a subgraph T of G without cycles, that connects all vertices V, is called a ***spanning tree****. We call T a* ***minimum spanning tree (MST)*** *if T is a spanning tree of G and if*

$$w(T) := \sum_{e \in T} w(e) \leq w(T') \quad \text{for all spanning trees } T' \text{ of } G$$

holds.

3.4.1 Construction of Grid Graphs for the MST Problem with Given Differences d

As for the shortest path problem, we try to construct instances of the minimum spanning tree problem, such that the related penalty alternatives $P^{(0)}, P^{(1)}, \ldots, P^{(k)}$ fulfill

$$w\left(P^{(i)}\right) - w\left(P^{(i-1)}\right) = d_i \quad \text{for } i = 1, \ldots, k$$

with a predefined difference vector $d = (d_1, \ldots, d_k) \in \mathbb{R}^k_{>0}$ and a predefined set of threshold parameters $0 < \varepsilon_1 < \cdots < \varepsilon_k$. Unfortunately, our construction works only for difference vectors d with $d_1 < d_2 < \ldots < d_k$ and without a predefined threshold parameter vector ε. However, we are convinced that there exist instances of the MST problem to each difference vector d.

Theorem 3.4.2 (Penalty Alternatives in Ladder Graphs)
Given a difference vector $d \in \mathbb{R}^k_{>0}$, with

$$0 < d_1 < d_2 < \cdots < d_k \,,$$

we consider an undirected ladder graph of width $k+1$ and the weights illustrated in Figure 3.4.1.

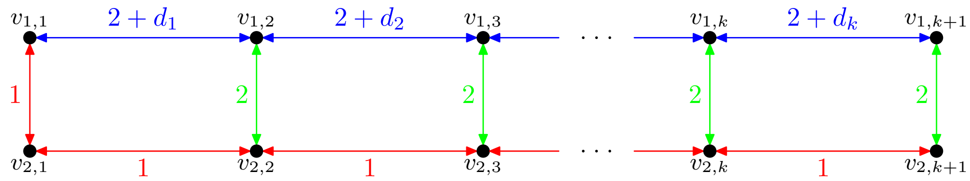


Figure 3.4.1: Construction scheme.

1. *The edges marked in red (the first ladder spoke $\{v_{1,1}, v_{2,1}\}$ and all edges in the lower row $\{v_{2,i}, v_{2,i+1}\}$ for $i = 1, \ldots, k$) have weight 1.*
2. *The remaining green ladder spokes $\{v_{1,i}, v_{2,i}\}$ for $i = 2, \ldots, k+1$ have weight 2.*
3. *The vertical blue edges in the upper row $\{v_{1,i}, v_{1,i+1}\}$ for $i = 1, \ldots k$ have the weights $2 + d_1, 2 + d_2, \ldots, 2 + d_k$ from left to right.*

Then the spanning tree $M^{(i)}$ $(i = 0, \ldots, k)$, which consists of

- *all red marked edges,*
- *the i left-most blue marked edges in the upper row $\{v_{1,1}, v_{1,2}\}, \{v_{1,2}, v_{1,3}\}, \ldots, \{v_{1,i}, v_{1,i+1}\}$,*
- *and the $k-i$ right-most green marked ladder spokes $\{v_{1,i+2}, v_{2,i+2}\}, \{v_{1,i+3}, v_{2,i+3}\}, \ldots, \{v_{1,k+1}, v_{2,k+1}\}$*

is the i-th penalty alternative $P^{(i)}$. Therewith the penalty alternatives have the given weight differences $d = (d_1, \ldots, d_k)$. See Figure 3.4.2 for an illustration of the penalty alternatives.

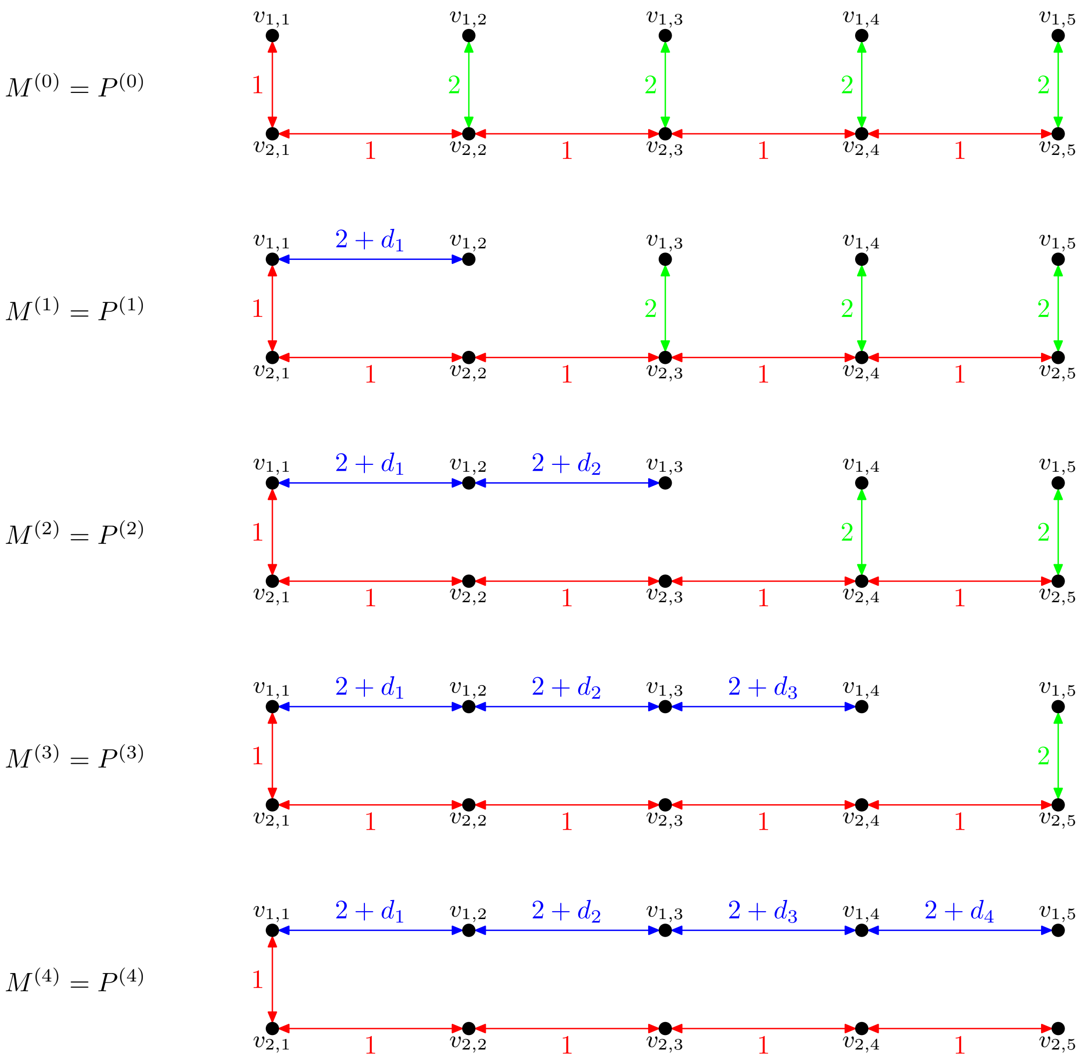


Figure 3.4.2: Penalty alternatives for the minimum spanning tree problem on the graph from Figure 3.4.1 with $k = 4$.

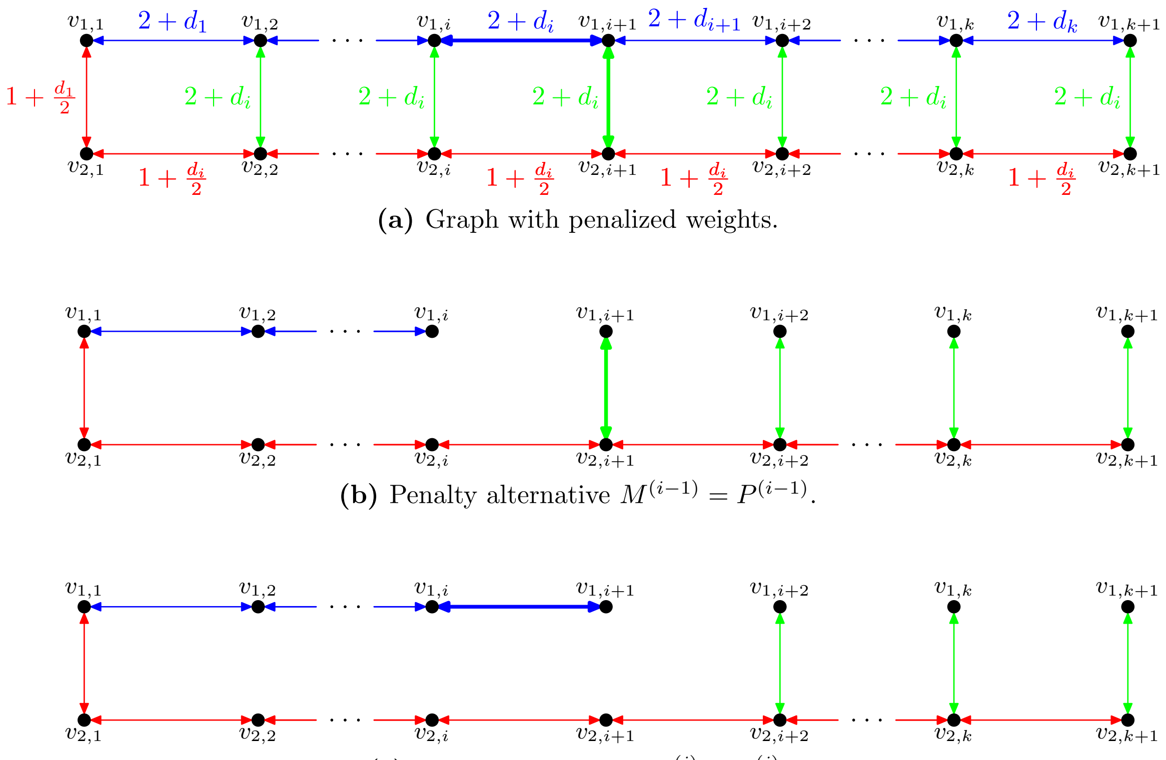
Proof. In a first step we show that the spanning trees $M^{(0)}, \dots, M^{(k)}$, stated in the theorem and illustrated in Figure 3.4.2, are indeed penalty alternatives. This is not difficult to prove but sometimes a little bit technical. As a matter of principle, each step of the proof is basically just an execution of Kruskal's algorithm.

We start with the optimal solution. Since $d_i > 0$ for $i = 1, \dots, k$ holds, the red marked edges with weight 1 and the green marked edges with weight 2 are the $2k+1$ shortest edges. Since these edges form a spanning tree, the spanning tree $M^{(0)}$ is the optimal spanning tree and the penalty alternative $P^{(0)}$.

Now we show that the spanning tree $M^{(i)}$ is a penalty alternative for each penalty parameter ε with $\varepsilon_{i-1} \leq \varepsilon \leq \varepsilon_i$ and

$$\varepsilon_0 := 0, \quad \varepsilon_1 := \frac{d_1}{2}, \quad \varepsilon_2 := \frac{d_2}{2}, \quad \dots \quad \varepsilon_k := \frac{d_k}{2}\,.$$

The weights of the edges in the graph punished with $\varepsilon_i := \frac{d_i}{2}$ are shown in Figure 3.4.3 (a).



(a) Graph with penalized weights.

(b) Penalty alternative $M^{(i-1)} = P^{(i-1)}$.

(c) Penalty alternative $M^{(i)} = P^{(i)}$.

Figure 3.4.3: Penalized graph and penalty alternatives for penalty parameter $\varepsilon_i := \frac{d_i}{2}$.

Obviously, for the penalized edge weights the following statements are true.

1. The penalized weight of the red edges is still smaller than the penalized weight of the green edges.

2. The first $i-1$ blue marked edges in the upper row $\{v_{1,1}, v_{1,2}\}, \{v_{1,2}, v_{1,3}\}, \dots, \{v_{1,i-1}, v_{1,i}\}$, with weights $2+d_1, 2+d_2, \dots 2+d_{i-1} \leq 2+d_i$ from left to right,

have also a smaller penalized weight than the green marked edges because of $d_1 < d_2 < \cdots < d_k$.

3. The i-th edge in the upper row $\{v_{1,i}, v_{1,i+1}\}$, indicated in bold, has the same weight as the green marked edges.

4. All the remaining edges in the upper column have a greater weight than the green marked edges.

Hence, the $k+i$ edges with the smallest penalized weights are

- all $k+1$ red marked edges
- and the first $i-1$ blue marked edges in the upper row $\{v_{1,1}, v_{1,2}\}, \{v_{1,2}, v_{1,3}\}, \ldots, \{v_{1,i-1}, v_{1,i}\}$.

These $k+i$ edges are chosen by Kruskal's algorithm, since they do not form a cycle. For the missing $k-i+1$ edges (a spanning tree consists of $2k-1$ edges in total), all green marked edges and the i-th edge in the upper row $\{v_{1,i}, v_{1,i+1}\}$ are worth considering, since they have the smallest penalized weights of all remaining edges. But the $i-1$ left-most green marked ladder spokes $\{v_{1,2}, v_{2,2}\}, \ldots, \{v_{1,i}, v_{2,i}\}$ cannot be chosen, since each of them would provoke at least one cycle. Hence, Kruskal's algorithm chooses the $k-i-1$ right-most green marked ladder spokes $\{v_{1,i+2}, v_{2,i+2}\}, \ldots, \{v_{1,k+1}, v_{2,k+1}\}$ and one of the two edges indicated in bold – either the $(i+1)$-th ladder spoke $\{v_{1,i+1}, v_{2,i+1}\}$ or the i-th edge in the upper row $\{v_{1,i}, v_{1,i+1}\}$. These two spanning trees are in fact the trees $M^{(i-1)}$ and $M^{(i)}$ described in the theorem.

Therewith we showed, that $\varepsilon_i := \frac{d_i}{2}$ is the threshold parameter between $M^{(i)}$ and $M^{(i+1)}$. Furthermore, the k-th spanning tree $M^{(k)}$ is obviously the ∞-penalty alternative. Together with the optimal spanning tree for the unpunished problem, we get the following table, containing all penalty alternatives with their weight and penalized part.

i	$w(P^i)$	$p(P^i)$
0	$3k+1$	$3k+1$
1	$3k+1+d_1$	$3k+1-2$
2	$3k+1+d_1+d_2$	$3k+1-4$
$\vdots$	$\vdots$	$\vdots$
k	$3k+1+d_1+d_2+\cdots+d_k$	$3k+1-2k$

In general

$$w\left(P^{(i)}\right) = 3k+1+d_1+\cdots+d_i$$
$$p\left(P^{(i)}\right) = 3k+1-2i$$

hold and consequently

$$\varepsilon_i = \frac{w\left(P^{(i)}\right) - w\left(P^{(i-1)}\right)}{p\left(P^{(i-1)}\right) - w\left(P^{(i)}\right)} = \frac{d_i}{2}$$

is the i-th threshold parameter, as claimed.

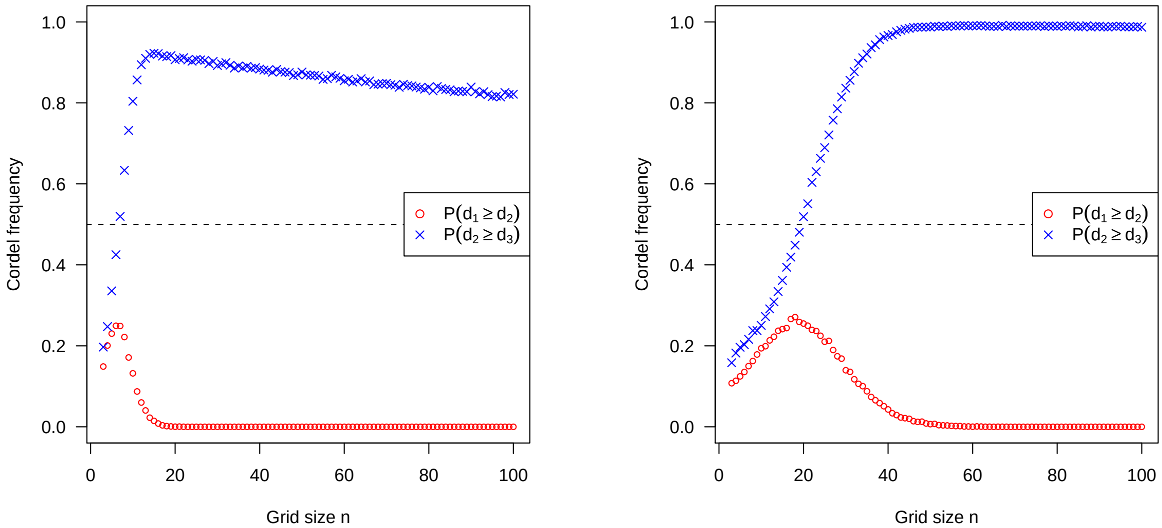
So we proved Theorem 3.4.2. ■

3.4.2 Experimental Results for the MST Problem

As for the shortest path problem, we determined the Cordel frequency for random grid graphs. But in contrast to the shortest path problem, we used undirected instead of directed graphs for the minimum spanning tree problem.

Definition 3.4.3 (Random Instances for the Minimum Spanning Tree Problem in Undirected Grid Graphs)
A ***random instance of size*** (m, n) ***and range*** $g \in \mathbb{N}$ *is an undirected* $m \times n$ *grid with randomly uniformly distributed edge weights in the integer range* $[1, 10^g]$.

As in case of the shortest path problem in trellises, it is highly probable that there exist two equally best minimal spanning trees. Thus, the Cordel frequency is again converging to 0% for $n \to \infty$. In contrast to the shortest path problem, this limit can be observed very clearly even for $g = 3$ and small n, as the following Figure 3.4.4 shows.



(a) $g = 2$ **(b)** $g = 3$

Figure 3.4.4: Cordel frequencies for undirected **quadratic** $\boldsymbol{n \times n}$ grid graphs.

We summarize the results for the minimum spanning tree problem in undirected grid graphs:

1. Again, the results for quadratic and rectangular grids are very similar.
2. The graph of the Cordel frequency, as a function of n, starts with approximately 10% and is then increasing until the maximal Cordel frequency of $\approx 25\%$ is reached. Then the Cordel frequency is decreasing and has to converge to 0% for $n \to \infty$.

3.5 Knapsack Problems

Finally, we determined the Cordel frequency for knapsack problems. We define three different types of knapsack problems.

Definition 3.5.1 (Knapsack Problems)
Consider a weight vector $w \in \mathbb{R}^n_{\geq 0}$ and a vector of values $v \in \mathbb{R}^n_{\geq 0}$. Thus we have n items $(w_1, v_1), \ldots, (w_n, v_n)$ with weights w_i and values v_i. Furthermore let $C \geq 0$ be a given knapsack capacity.
Then a knapsack problem is a problem of the following type.

$$\begin{aligned} \max \sum_{i=1}^{n} v_i x_i & \\ \textit{subject to } \sum_{i=1}^{n} w_i x_i \leq C & \\ x_i \in \{0, 1, \ldots, b\} & \qquad \textit{for } i = 1, \ldots, n \end{aligned}$$

We consider the following three different types of knapsack problems:

*(i) In case of $b = 1$, we call the problem above **0-1-knapsack problem or binary knapsack problem [KP]**.*

*(ii) In case of $b = \infty$, where we have no upper bound for x_i, we call the problem **unbounded knapsack problem [UKP]**.*

*(iii) In case of $1 \leq b < \infty$, we call the problem **b-bounded knapsack problem [BKP(b)]**.*

Obviously, the following remarks hold.

Remark 3.5.2
(i) KP and BKP(1) are equivalent.

(ii) Let $P^{(0)}, \ldots, P^{(k-1)}$ be the k best penalty alternatives for a given unbounded knapsack problem UKP and let

$$b_{UKP} := \max_{\substack{0 \leq i \leq k-1, \\ 1 \leq j \leq n}} P_j^{(i)}$$

be the maximal frequency of used items in these penalty alternatives. Then $P^{(0)}, \ldots, P^{(k-1)}$ are the k best penalty alternatives for BKP(b_{UKP}), too.

Hence it is reasonable to consider the penalty alternatives for the following sequence of problems

$$\text{KP} - \text{BKP}(2) - \text{BKP}(3) - \ldots - \text{BKP}(b_{\text{UKP}} - 1) - \text{UKP}$$

with weights w and values v. In this way we could analyze a transition from the binary knapsack problem to the unbounded knapsack problem. Unfortunately, solving BKP(b) instances is very time-consuming even for small b (for example $b = 10$). Thus, in fact, we only established the Cordel frequencies for some selected bounds b.

Construction of Instances for a Given Difference Vector *d*

As in the previous sections, we tried to construct knapsack instances to a given difference vector $d \in \mathbb{R}^k_{>0}$. Unfortunately, our efforts were not successful. Even for difference vectors d with certain properties (for example $d_1 < d_2 < \ldots < d_k$) we did not find a construction scheme. This leads us to the supposition that their might be difference vectors $d \in \mathbb{R}^k_{>0}$ for which no instance of the knapsack problem exists. But to decide whether this supposition is true or not, further investigations are necessary.

3.5.1 Experimental Results for the Knapsack Problem

Definition 3.5.3 (Random Instances for the Knapsack Problem)
According to [MT 1990, p. 67], we considered three types of random instances. In each of the three cases w_i was chosen uniformly randomly in $[1, 10^g]$ with a range parameter $g \in \mathbb{N}$. We differentiated:

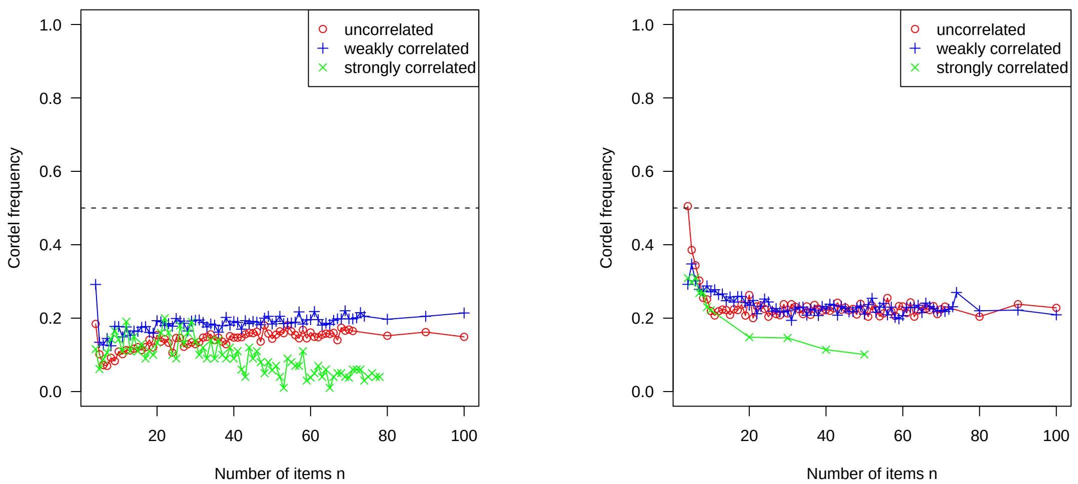
uncorrelated instances: *v_i was chosen uniformly randomly in $[1, 10^g]$, too.*

weakly correlated instances: *v_i was chosen uniformly randomly in $[w_i - 10^{g-1}, w_i + 10^g]$ with respect to $p_i \geq 1$.*

strongly correlated instances: $v_i = w_j + 10^{g-1}$

In each case the knapsack capacity was set to $C = \frac{1}{2} \sum_{i=1}^{n} w_i$.

Again use a range parameter g, whereupon a huge g implies quasi-continuous instances. Hence, we analyzed two transitions: The transition from discrete (small range parameter g) to quasi-continuous instances (huge range parameter g) and the transition from the binary ($b = 1$) to the unbounded ($b = \infty$) knapsack problem.

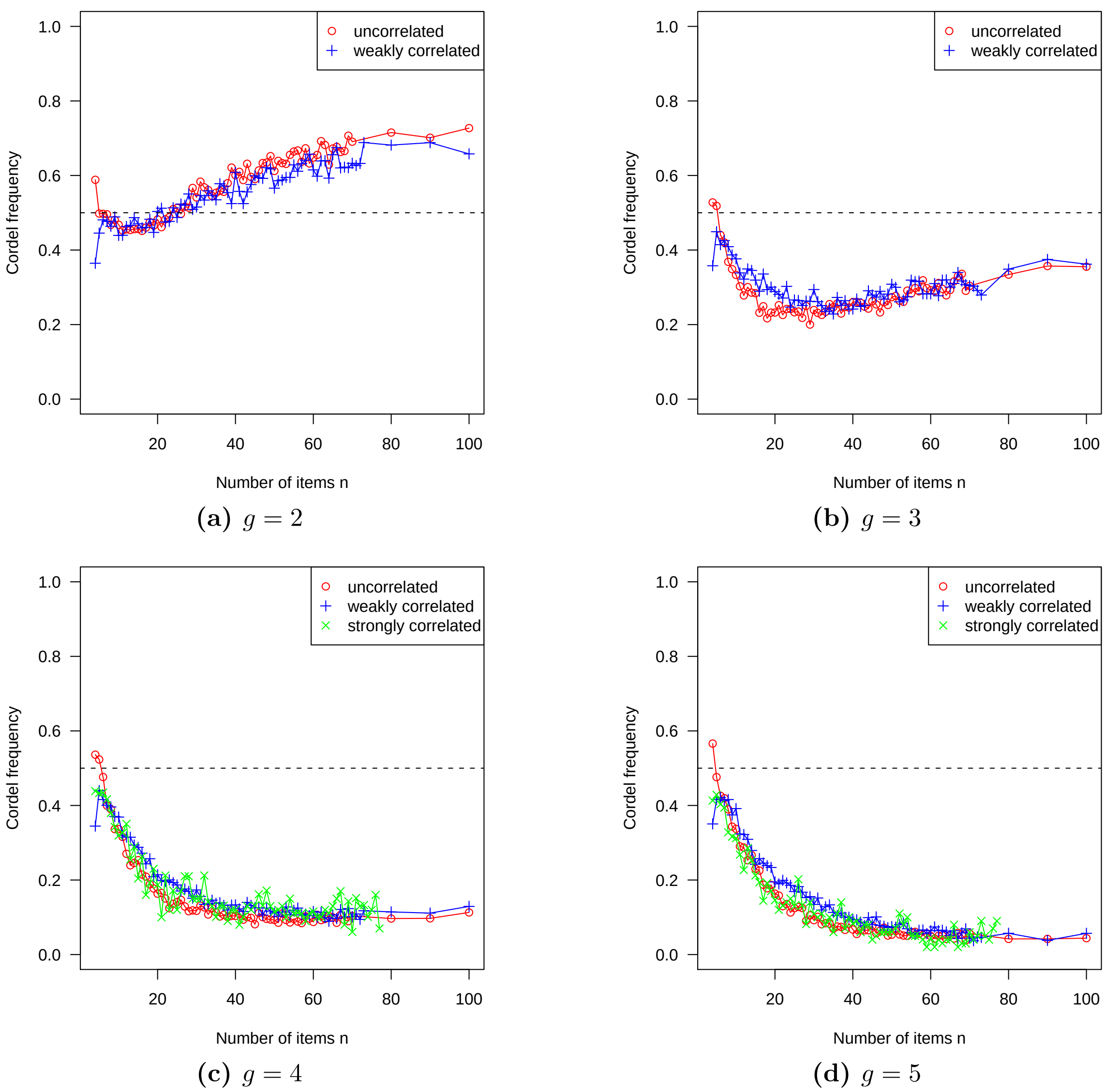


(a) binary knapsack problem ($b = 1$) **(b)** bounded knapsack problem ($b = 10$)

Figure 3.5.1: Cordel frequencies for b-bounded knapsack problems with $g = 4$ and $b < \infty$.

We start with the analysis of the b-bounded knapsack problem with $\boldsymbol{b < \infty}$. Figure 3.5.1 shows the experimentally observed Cordel frequencies for range parameter $g = 4$ and bounds $b \in \{1, 10\}$. It turns out that the Cordel frequency for strong correlated instances is usually a little bit smaller than the Cordel frequency for uncorrelated and weakly correlated instances, especially for greater n. While the Cordel frequency for very small item numbers ($n \leq 5$) can be even about 50%, the frequencies are quickly stabilizing at a low level of $15\% - 25\%$. The two pictures suggest a slight upward trend for greater bounds b which cannot be substantiated. The curve shape, presented here for $g = 4$, is found again for the other range size parameters $g \in \{2, \ldots, 7\}$ as the figures in Appendix J.5 (pp. 228-230) show.

In case of the **unbounded knapsack problem** ($\boldsymbol{b = \infty}$), the Cordel frequencies for all three distributions seem to be the same, as Figure 3.5.2 shows.



(a) $g = 2$ (b) $g = 3$ (c) $g = 4$ (d) $g = 5$

Figure 3.5.2: Cordel frequencies for the **unbounded knapsack problem** ($\boldsymbol{b = \infty}$).

Here, a very surprising phenomenon occurs: The Cordel frequency **heavily** depends on the range parameter g! While the graphs for $g \geq 4$ and $3 \leq n \leq 100$ all look the same, the graphs for $g = 2$ and $g = 3$ show a complete different curve shape.

- For $\boldsymbol{g = 2}$ the Cordel frequency for small n ($n \leq 30$) is approximately 50%. For increasing n the Cordel frequency is increasing and seems to converge to $\approx 70\%$.
- For $\boldsymbol{g = 3}$, however, the graph first decreases from $\approx 50\%$ to $\approx 25\%$, before it slightly increases. This time, the Cordel frequency seems to converge to $\approx 35\%$ which is considerably smaller than the suspected limit of $\approx 70\%$ for $g = 2$.
- As already mentioned, for $g \in \{4, \ldots, 7\}$ the graphs look nearly all the same. The Cordel frequency is decreasing from $\approx 50\%$ to a suspected limit $0 - 10\%$. This decrease becomes weaker and weaker for increasing n. The only difference the four pictures for $g \in \{4, \ldots, 7\}$ (cf. Figure J.5.4 (c)-(d) on page 231) show, is that the limit which seems to become smaller for greater range parameters g.
- Note that the suggested limits are not confirmed by investigation of instances with $n \gg 100$. Hence, these limits should be treated with caution.

The complete results can be found in the appendix. Note that the running time for the computation of the three best penalty alternatives for strongly correlated instances is very long. That is why we considered only small n and only few random instances there.

	reference to the appendix
$b = 1$	Figure J.5.1, page 228
$b = 10$	Figure J.5.2, page 229
$b = 20$	Figure J.5.3, page 230
$b = \infty$	Figure J.5.4, page 231

3.6 Adjusted Frequencies without Consideration of Multiple (Equally) Best Solutions

For some optimization problems it is very probable to have at least two optimal solutions for great n. Two or more equally best optimal solutions imply

$$w\left(P^{(0)}\right) = w\left(P^{(1)}\right) = \min_{B \in S} w(B)$$

and consequently $d_1 = 0$. Hence, (GeCoP) cannot be fulfilled because of $d_2 \geq 0 = d_1$.

This phenomenon was observed for the shortest path problem in trellises (cf. Subsection 3.2.1, starting on page 73) and for the minimum spanning tree problem in undirected grid graphs (cf. Subsection 3.4.2, starting on page 81). In both cases, the probability of having at least two (equally best) optimal solutions is converging to 100% for increasing trellises or grid graphs respectively ($n \to \infty$). Consequently, the

Cordel frequency is tending to 0% for $n \to \infty$ as Figures 3.2.2 (p. 73) and 3.4.4 (p. 81) for the shortest path problem and the MST problem show, respectively.

One can eliminate the influence of two or more equally best optimal solutions, by ignoring the second equi-optimal solution. This can be done by considering the so-called **adjusted Cordel frequency** $\boldsymbol{P\left(\overline{d}_1 \geq \overline{d}_2\right)}$ with

$$\overline{d}_1 := \begin{cases} w\left(P^{(1)}\right) - w\left(P^{(0)}\right), & \text{for } w\left(P^{(0)}\right) \neq w\left(P^{(1)}\right), \\ w\left(P^{(2)}\right) - w\left(P^{(0)}\right), & \text{for } w\left(P^{(0)}\right) = w\left(P^{(1)}\right), \end{cases} \tag{3.1}$$

and

$$\overline{d}_2 := \begin{cases} w\left(P^{(2)}\right) - w\left(P^{(1)}\right), & \text{for } w\left(P^{(0)}\right) \neq w\left(P^{(1)}\right), \\ w\left(P^{(3)}\right) - w\left(P^{(2)}\right), & \text{for } w\left(P^{(0)}\right) = w\left(P^{(1)}\right). \end{cases} \tag{3.2}$$

In fact, this can be understood as a redefinition of the penalty selection rule.

In the next two subsections we present the experimentally observed **adjusted** frequencies for the shortest path problem in trellises and for the MST problem in undirected grid graphs. This consideration of the adjusted Cordel frequency is complementary in order to attain a comprehensive picture.

3.6.1 Experimental Results for the Shortest Path Problem in Trellises

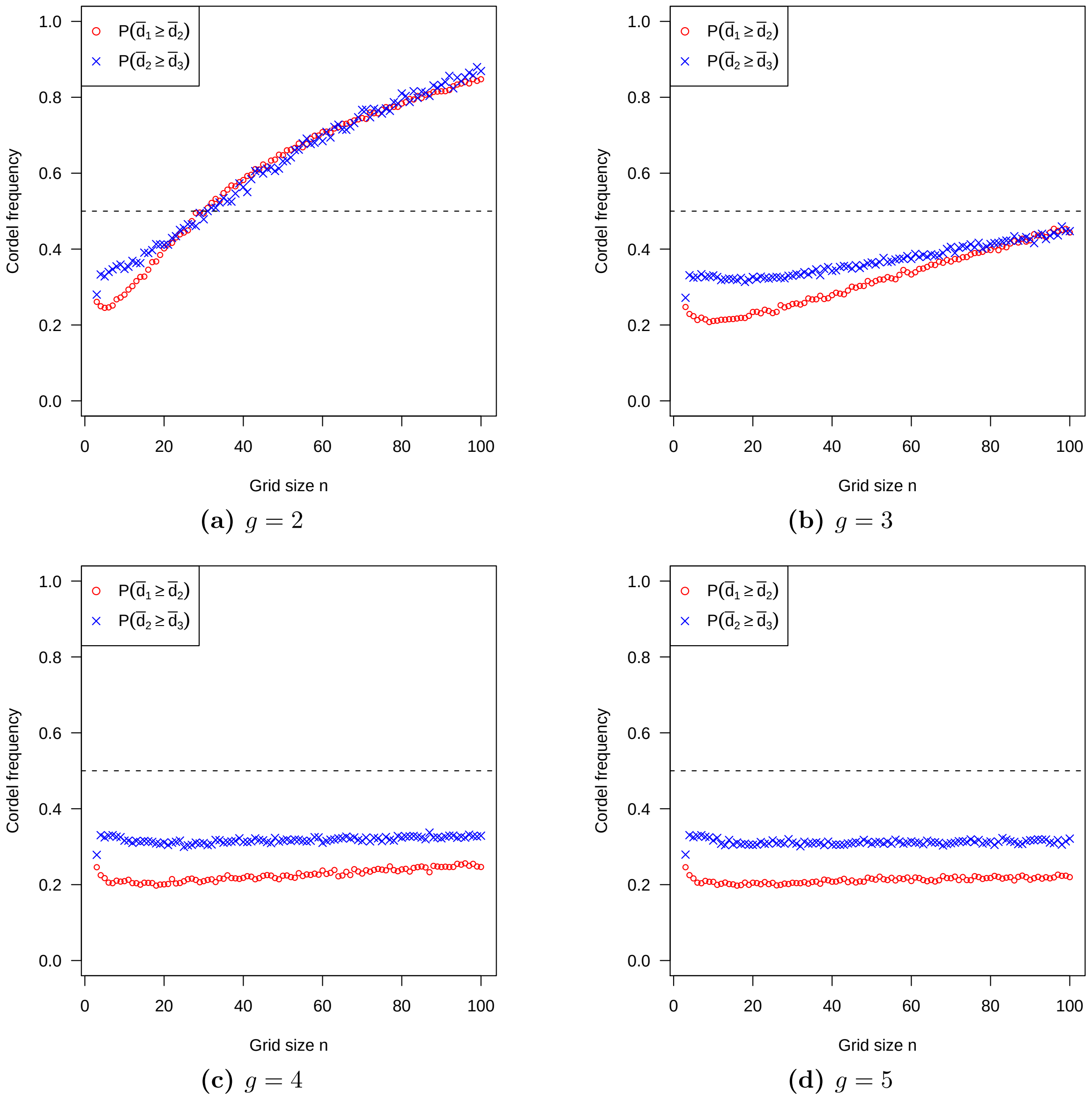
Figure 3.6.1 shows the adjusted Cordel frequencies for the shortest path problem in trellises. Like the original Cordel frequency, the adjusted Cordel frequency starts with 20% − 25% and increases subsequently. For $g = 2$ it seems that the adjusted Cordel frequency $P\left(\overline{d}_1 \geq \overline{d}_2\right)$ is tending to 100% (or at least to a limit greater than 85%) which might be quite surprising. For $g = 3$ and $n = 100$ the adjusted Cordel frequencies are also already above 50%. But for $g > 3$ the adjusted frequencies for $3 \leq n \leq 100$ are roughly stable around 20 − 30%. Here, no trend can be detected without consideration of larger grid sizes n.

In fact, the courses of these new graphs are not very surprising if you look at $P(d_2 \geq d_3)$ represented by the blue crosses in Figure 3.2.2 on page 73. In fact, for each instance with at least two optimal solutions ($w\left(P^{(0)}\right) = w\left(P^{(1)}\right)$),

$$\begin{aligned} \overline{d}_1 &= w\left(P^{(2)}\right) - w\left(P^{(0)}\right) = w\left(P^{(2)}\right) - w\left(P^{(1)}\right) = d_2 \\ \text{and} \quad \overline{d}_2 &= w\left(P^{(3)}\right) - w\left(P^{(2)}\right) = d_3\,, \end{aligned}$$

and consequently $P\left(\overline{d}_1 \geq \overline{d}_2\right) = P(d_2 \geq d_3)$ hold. Hence, $P\left(\overline{d}_1 \geq \overline{d}_2\right)$ and $P(d_2 \geq d_3)$ are nearly the same if it is very probable that there are at least two optimal solutions.

The complete results for all considered range parameters $g \in \{2, \ldots, 7\}$ and grid sizes can be found in the appendix. The following table shows where to find which graph.



(a) $g = 2$ **(b)** $g = 3$ **(c)** $g = 4$ **(d)** $g = 5$

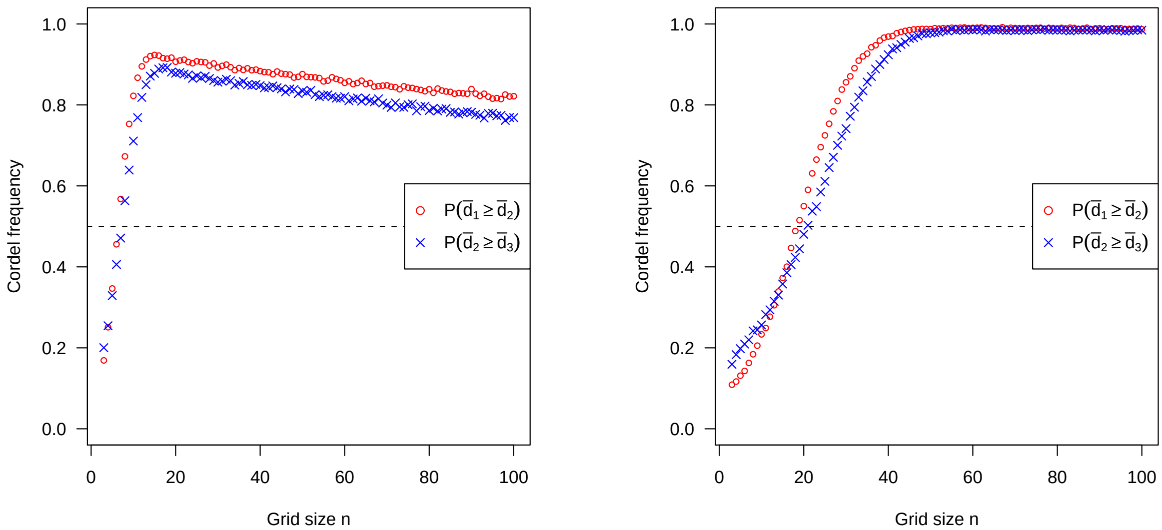
Figure 3.6.1: Adjusted Cordel frequencies $P\left(\bar{d}_1 \geq \bar{d}_2\right)$ for **quadratic** $\boldsymbol{n \times n}$ trellis graphs.

	Cordel frequencies $P\left(d_1 \geq d_2\right)$	adjusted Cordel frequencies $P\left(\bar{d}_1 \geq \bar{d}_2\right)$
quadratic $n \times n$ trellises	Figure J.2.1, page 214	Figure J.2.2, page 215
rectangular $m \times 2m$ trellises	Figure J.2.3, page 216	Figure J.2.4, page 217
rectangular $2n \times n$ trellises	Figure J.2.5, page 218	Figure J.2.6, page 219

The results for rectangular $m \times 2m$ trellises are very similar to the results for quadratic $n \times n$ trellises. But for rectangular $2n \times n$ trellises and $g = 2$ the picture is significantly different. Here, the adjusted frequencies do not seem to converge to a limit greater than 85%. Instead, for $n \geq 50$ the adjusted frequencies seem to be constant around $\approx 75\%$. For $g > 2$ the pictures look as usual. Hence, there is a difference between $m \times 2m$ trellises which are twice as wide as they are high and $2n \times n$ trellises which are twice as high as they are wide.

3.6.2 Experimental Results for the MST Problem

Eliminating the influence of more than one optimal solutions by considering the adjusted Cordel frequency $P\left(\overline{d}_1 \geq \overline{d}_2\right)$, leads to the frequencies shown in Figure 3.6.2. The adjusted frequency starts again with $\approx 10\%$ and increases subsequently. For $g = 3$ it seems that the adjusted Cordel frequency $P\left(\overline{d}_1 \geq \overline{d}_2\right)$ tends to 100%. But for $g = 2$, the function is decreasing for $n > 20$ with an unknown limit, which may be between 50% and 70%. But this suggestion really needs to be verified through consideration of larger grids.



(a) $g = 2$ **(b)** $g = 3$

Figure 3.6.2: Adjusted Cordel frequencies $P\left(\overline{d}_1 \geq \overline{d}_2\right)$ for undirected **quadratic** $\boldsymbol{n \times n}$ grid graphs **without consideration of the second optimum**.

In the appendix the graphs for all considered range parameters g can be found. Note, that for $g \in \{6, 7\}$ only small grids were considered.

	Cordel frequencies $P\left(d_1 \geq d_2\right)$	adjusted Cordel frequencies $P\left(\overline{d}_1 \geq \overline{d}_2\right)$
quadratic $n \times n$ grids	Figure J.4.1, page 224	Figure J.4.2, page 225
rectangular $n \times 2n$ grids	Figure J.4.3, page 226	Figure J.4.4, page 227

3.7 Summary of the Key Results

We conclude this chapter with a summary of the key results, obtained in all three problem classes (shortest path problem, minimum spanning tree problem and knapsack problem). The table below gives an overview of where which phenomenon was observed, including the associated page references.

Observation 1 For some optimization problems the probability of having at least two optimal solutions is converging to 1 for $n \to \infty$. In this case the Cordel frequency has to converge to 0, since two or more optimal solutions imply

$$w\left(P^{(0)}\right) = w\left(P^{(1)}\right) = \min_{B \in S} w(B)$$

and consequently $d_1 = 0$. Here, (GeCoP) cannot be fulfilled, since $d_2 \geq 0 = d_1$ holds.

Observation 2 Considering the **adjusted** Cordel frequencies $P\left(\overline{d}_1 \geq \overline{d}_2\right)$ with

$$\overline{d}_1 := \begin{cases} w\left(P^{(1)}\right) - w\left(P^{(0)}\right), & \text{for } w\left(P^{(0)}\right) \neq w\left(P^{(1)}\right) \\ w\left(P^{(2)}\right) - w\left(P^{(0)}\right), & \text{for } w\left(P^{(0)}\right) = w\left(P^{(1)}\right) \end{cases}$$

and

$$\overline{d}_2 := \begin{cases} w\left(P^{(2)}\right) - w\left(P^{(1)}\right), & \text{for } w\left(P^{(0)}\right) \neq w\left(P^{(1)}\right) \\ w\left(P^{(3)}\right) - w\left(P^{(2)}\right), & \text{for } w\left(P^{(0)}\right) = w\left(P^{(1)}\right) \end{cases}$$

we are able to eliminate the influence of a second (equi-)optimal solution (cf. **Observation 1**). If it is very probable to have at least two optimal solutions, then $P(d_2 \geq d_3) \approx P\left(\overline{d}_1 \geq \overline{d}_2\right)$ holds. Eliminating the influence of a second optimal solutions leads to adjusted Cordel frequencies, that very often seem to converge to a limit between 75% and 100%

Observation 3 For the unbounded knapsack problem the limit of the Cordel frequency seems to depend heavily on the range parameter g. The experimental results suggest limits between $\approx 75\%$ ($g = 2$, cf. Figure 3.5.2 (a) on page 84) and $\approx 0\%$ ($g = 7$, cf. Figure J.5.4 (f) on page 231 in the appendix).

Observation 4 If none of the three unexpected observations above occurred, limits between 15% and 25% were observed. In this case, all observed graphs were very smooth. For $n \geq 30$ the determined Cordel frequencies were nearly constant. For us, this is the most important observation. We even go so far to say that **typical** Cordel frequencies for the penalty selection rule are between 15% and 30%.

Observation 5 For real road networks, Cordel frequencies between 16% and 42% occurred.

References for the observed results:

	Reference
Observation 1	shortest path problem in trellises (pp. 73 - 74) minimal spanning tree problem in grid graphs (pp. 81 - 81)
Observation 2	shortest path problem in trellises (pp. 86 - 87) minimal spanning tree problem in grid graphs (pp. 88 - 88)
Observation 3	unbounded knapsack problem (pp. 83)
Observation 4	shortest path problem in grid graphs (pp. 71 - 72) bounded knapsack problem with $b < \infty$ (pp. 83 - 85)
Observation 5	shortest path problem in road networks with real travel times (pp. 75 - 76)

Chapter 4

The Cordel Frequency under the Best Solutions Rule

4.1 The Algorithm of E. Lawler for Computing the k Best Solutions

In 1972 E. Lawler [Law 1972] presented an algorithm for computing the k best solutions to discrete optimization problems. This algorithm was a specialization of Murty's method [Mur 1968] for the assignment problem. Since Lawler's algorithm is very general and not only applicable for the assignment problem, we want to present it here. To be more precise, Lawler's method is applicable to all binary optimization problems with an efficient procedure to determine the optimal solution when some variables have fixed values. This is, for example, the case for ranking the k shortest simple paths, the k least costly spanning trees and the k best solutions to the binary knapsack problem. And, as Lawlers method is an advancement of Murty's algorithm, it is applicable to the assignment problem as well.

Algorithm 5 (Algorithm for Computing the k Best Solutions to Binary Optimization Problems, [Law 1972])

Initialization: *Compute an optimal solution to the original problem and put this solution together with the variable $s = 0$ in the list L. Set $m = 1$.*

Step 1 *Remove the best solution and the corresponding variable s form L. This solution is the m-th best solution $x^{(m)}$.*

Step 2 *If $m = k$ then stop. Otherwise continue with **Step 3**.*

Step 3 *$x^{(m)}$ was obtained by fixing the values of $x_1, x_2, \dots, x_s$. With these s values been kept, we consider the following $n-s$ subproblems by stepwise fixing the remaining variables as follows:*

$$
\begin{array}{llll}
(1) & x_{s+1} = 1 - x_{s+1}^{(m)} & & \\
(2) & x_{s+1} = x_{s+1}^{(m)}, & x_{s+2} = 1 - x_{s+2}^{(m)} & \\
\vdots & & & \\
(n-s) & x_{s+1} = x_{s+1}^{(m)}, & x_{s+2} = x_{s+2}^{(m)}, & \dots \quad x_n = x_n^{(m)}
\end{array}
$$

Compute an optimal solution to each of these $n - s$ problems and store all of them together with the index of the last fixed variable in the list L. Set $m = m + 1$ and go to ***Step 1****.*

The basic idea of the algorithm is to partition the set of feasible solutions. More precisely, if S is the set of feasible solutions for the subproblem which provided $x^{(m)}$ as optimal solution, then the set $S \setminus \left\{x^{(m)}\right\}$ is partitioned into $n - s$ sets $S^{(1)}, \ldots, S^{(n-s)}$. Here, $S^{(i)}$ denotes the set of feasible solutions for subproblem (i) defined in **Step 3** of the algorithm. Hence, the branching excludes exactly $x^{(m)}$ from further considerations. View [Law 1972] for additional hints on general storage reduction and efficiency improvement.

4.2 Experimentally Obtained Cordel Frequencies for the Best Solutions Rule

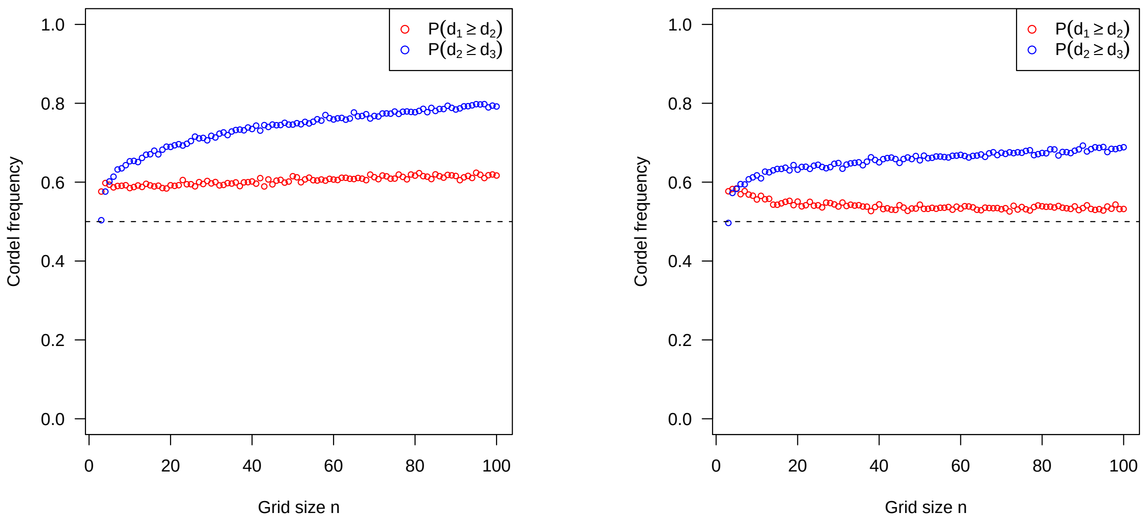
The algorithm of Lawler may sound elegant and versatile, but unfortunately it takes a long time to compute the three best solutions to a given optimization problem, when the number of variables becomes too great. That is why we did not analyze as many optimization problems as we did in case of the penalty selection rule. In the following we present the experimentally observed Cordel frequencies for the shortest path problem (Subsection 4.2.1) and the minimum spanning tree problem (Subsection 4.2.2) in grid graphs.

4.2.1 Experimental Results for the Shortest Path Problem in Grid Graphs

For the best solutions rule the same kind of random instances was considered as for the penalty selection rule.

Definition 4.2.1 (Random Instances for the Shortest Path Problem in Directed Grid Graphs (cf. Definition 3.1.5 on page 71))
A ***random instance of height*** $m > 2$*,* ***width*** $n > 2$ ***and range*** $g \in \mathbb{N}$ *is a directed* $m \times n$ *grid with randomly uniformly distributed edge weights in the integer range* $[1, 10^g]$.

For the shortest path problem two types of pictures emerged. As an example, Figure 4.2.1 shows the experimentally determined Cordel frequencies for **quadratic $n \times n$ grid graphs** and $g \in \{2, 3\}$. For $g = 2$ the red dots, which represent the Cordel frequency, seem to be constant $\approx 60\%$. But for $g \in \{3, 4, 5, 6, 7\}$, where all graphs look similar, the frequency seems to tend to 50%.



(a) $g = 2$ **(b)** $g = 3$

Figure 4.2.1: Cordel frequencies for **quadratic** $\boldsymbol{n \times n}$ grid graphs.

For **rectangular** $\boldsymbol{n \times 2n}$ **grids** the pictures are almost identical. To see all results, please observe figures K.1.1 (quadratic grids) and K.1.2 (rectangular grids) in the Appendix on pages 234 and 235.

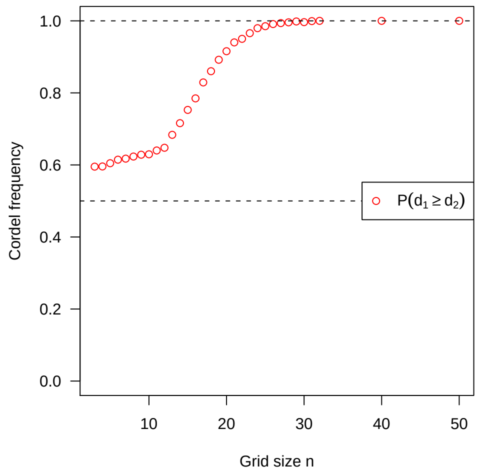
4.2.2 Experimental Results for the Minimum Spanning Tree Problem in Grid Graphs

The same kind of random instances was considered as in case of the penalty selection rule.

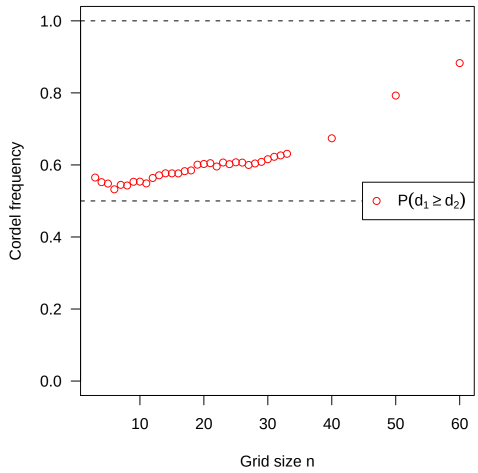
Definition 4.2.2 (Random Instances for the Minimum Spanning Tree Problem in Directed Grid Graphs (cf. Definition 3.4.3 on page 81))
A ***random instance of size*** (m, n) ***and range*** $g \in \mathbb{N}$ *is an undirected* $m \times n$ *grid with randomly uniformly distributed edge weights in the integer range* $[1, 10^g]$.

Due to the long running times, not each grid size parameter $n \in \{3, \ldots, 100\}$ was considered and for larger n only very few random instances (partially less than 100) were analyzed.

As already discussed in Section 3.6 (starting on page 85) the probability of having multiple (equally) best optimal solutions is converging to 1 for $n \to \infty$. In analogy, for n large enough it is very likely to have at least three equally best optimal solutions implying $d_1 = d_2 = 0$. Consequently, the Cordel frequency is tending to 100% for $n \to \infty$. This trend can be seen very clearly for $g = 2$ and $g = 3$ (cf. Figure 4.2.2). But for $g > 3$ and $n \times n$ grids with $n \leq 100$, the Cordel frequencies seem to be constantly 50%. However, as mentioned earlier, even for $g > 3$ the Cordel frequency has to tend to 100% for $n \to \infty$.



(a) $g = 2$



(b) $g = 3$

Figure 4.2.2: Cordel frequencies for **quadratic** $\boldsymbol{n \times n}$ grids. Note that for $n \geq 30$ only few instances were considered.

Again, the results for quadratic $n \times n$ grids and rectangular $n \times 2n$ grids are almost identical. Figures K.2.1 (quadratic grids) and K.2.2 (rectangular grids) in the Appendix on pages 236 and 237 show all experimentally observed Cordel frequencies for the considered range parameters $g = 2, \ldots, 7$.

4.2.3 Comparison with the Results for the Penalty Selection Rule

Although only two optimization problems have been studied for the best solutions rule, it can be seen that both selection rules provide different results. We suspect that typical Cordel frequencies under the best solutions rule are between 50% or 60%. But to prove or disprove this conjecture more extensive investigations are necessary.

The next chapter provides some more results for the best solutions rule. There, we exactly compute Cordel frequencies under the best solutions rule for optimization problems where the probability density function of the functional values is given.

Chapter 5

Theoretical Models

5.1 Best Solutions Rule for Optimization Problems with Density f

In this section we want to compute the Cordel frequency for optimization problems under the assumption that we know the distribution of the functional values of all feasible solutions. These results can be compared with the experimentally determined Cordel frequencies presented in the previous Chapter 4 (starting on page 91). Furthermore these theoretical considerations may help in order to decide what are typical Cordel frequencies for the best solutions rule. For this purpose we consider the following model.

Definition 5.1.1 (Optimization Problem with Density f)
Let $f : \mathbb{R} \to \mathbb{R}_{\geq 0}$ be an arbitrary probability density function. We say that a ***random optimization problem has density f****, if the functional values of the feasible solutions are independent and identically distributed with density f. Additionally, F denotes the cumulative distribution function of the functional values.*

Therewith we can define the Cordel frequency for minimization and maximization problems with density f.

Definition 5.1.2 (Cordel Frequency of an Optimization Problem with Density f)
Suppose that $(V_1, \ldots, V_n)$ are n independent random variables with density f representing the functional values of an optimization problem with exactly n feasible solutions and density f. By

$$V_{1:n} \leq V_{2:n} \leq \ldots \leq V_{n:n}$$

we denote the corresponding order statistics (cf. [ABN 1992]) which consists of the values $V_1, \ldots, V_n$ sorted in ascending order.
Then $CF_{f,\min}(n)$ denotes the ***Cordel frequency of a minimization problem with density f****, which is the probability that the three best solutions of a minimization problem with density f and n feasible solutions fulfill the generalized Cordel property. Hence,*

$$CF_{f,\min}(n) := P\left(d_1 \geq d_2\right) = P\left(V_{2:n} - V_{1:n} \geq V_{3:n} - V_{2:n}\right) . \tag{5.1}$$

In analogy $CF_{f,\max}(n)$ denotes the ***Cordel frequency of a maximization problem with density f*** *which is*

$$CF_{f,\max}(n) := P\left(d_{n-1} \geq d_{n-2}\right) = P\left(V_{n:n} - V_{n-1:n} \geq V_{n-1:n} - V_{n-2:n}\right) . \tag{5.2}$$

Remark 5.1.3
The Cordel frequency of an optimization problem with density f is translation invariant and scale invariant for positive factors regarding the density function f:

$$\begin{aligned} CF_{f+t}(n) &= CF_f(n) \quad \textit{for all } t \in R \\ \textit{and} \quad CF_{sf}(n) &= CF_f(n) \quad \textit{for all } s > 0 \end{aligned}$$

This is due to the fact that the equivalences

$$v_2 - v_1 \geq v_3 - v_2 \quad \Leftrightarrow \quad (v_2 + t) - (v_1 + t) \geq (v_3 + t) - (v_2 + t)$$

and

$$v_2 - v_1 \geq v_3 - v_2 \quad \Leftrightarrow \quad sv_2 - sv_1 \geq sv_3 - sv_2$$

hold for all $t \in \mathbb{R}$ and $s > 0$.

Before computing the Cordel frequencies for concrete density functions f we first discuss which results are to be expected.

Conjecture 5.1.4
Consider a density function $f : [0, 1] \to [0, 1]$.

If $\lim_{x\to 1} f(x) > 0$ holds and if f is continuous, then the Cordel frequency for the maximization problem converges to $\frac{1}{2}$. Hence,

$$\lim_{n\to\infty} CF_{f,\max}(n) = \frac{1}{2}$$

holds.

This conjecture can easily been transferred to density functions $f : [a, b] \to [0, 1]$ with $a, b \in \mathbb{R}$ and to minimization instead of maximization problems. Indeed, the observations presented in the following subsections 5.1.1, 5.1.2, and 5.1.3 match this conjecture.

For density functions defined over the whole range $\mathbb{R}$, obviously $\lim_{x\to\infty} f(x) = 0$ holds. Here, we cannot apply the above conjecture. The question is, whether in this case every value $v \in [0, 1]$ is a possible limit for the Cordel frequency. That is to say: Does there exist a density function $f_v : \mathbb{R} \to [0, 1]$ with

$$\lim_{n\to\infty} \mathrm{CF}_{f,\max}(n) = v$$

for each $v \in [0, 1]$? In section 5.1.5 (starting on page 107) we come back to this question and give our conjecture.

After this preliminary considerations we start computing the Cordel frequency. Given the joint density function $f_{1,2,3:n}(v_1, v_2, v_3)$ of the three smallest functional values we can compute the Cordel frequency by

$$\begin{aligned} \mathrm{CF}_{f,\min}(n) &= P\left(V_{2:n} - V_{1:n} \geq V_{3:n} - V_{2:n}\right) \\ &= \int_{-\infty}^{\infty} \int_{v_1}^{\infty} \int_{v_2}^{\infty} \mathbf{1}_{v_2 - v_1 \geq v_3 - v_2} \cdot f_{1,2,3:n}(v_1, v_2, v_3)\, \mathrm{d}v_3\, \mathrm{d}v_2\, \mathrm{d}v_1 \\ &= \int_{-\infty}^{\infty} \int_{v_1}^{\infty} \int_{v_2}^{2v_2 - v_1} f_{1,2,3:n}(v_1, v_2, v_3)\, \mathrm{d}v_3\, \mathrm{d}v_2\, \mathrm{d}v_1\,. \end{aligned} \tag{5.3}$$

In the last step we used

$$v_2 - v_1 \geq v_3 - v_2 \quad \Leftrightarrow \quad v_3 \leq 2v_2 - v_1$$

and since $2v_2 - v_1 \geq v_2 + (v_2 - v_1) \geq v_2$ we can use $2v_2 - v_1$ as upper bound for v_3 without violating the condition $v_3 \geq v_2$.

From order statistics (cf. [ABN 1992, p. 24]) we know that the joint density function of the three smallest values $V_{1:n}, V_{2:n}, V_{3:n}$ equals

$$f_{1,2,3:n}(v_1, v_2, v_3) = \frac{n!}{(n-3)!} \left(1 - F(v_3)\right)^{n-3} f(v_1)\, f(v_2)\, f(v_3) \tag{5.4}$$

for $-\infty < v_1 \leq v_2 \leq v_3 < \infty$ where f denotes the underlying density function and F denotes the cumulative distribution function of the non-ordered values. Insertion of equation (5.4) into equation (5.3) yields

$$\begin{aligned} \mathrm{CF}_{f,\min}(n) &= \int_{-\infty}^{\infty} \int_{v_1}^{\infty} \int_{v_2}^{2v_2 - v_1} f_{1,2,3:n}(v_1, v_2, v_3)\, \mathrm{d}v_3\, \mathrm{d}v_2\, \mathrm{d}v_1 \\ &= \frac{n!}{(n-3)!} \int_{-\infty}^{\infty} f(v_1) \int_{v_1}^{\infty} f(v_2) \int_{v_2}^{2v_2 - v_1} f(v_3) \left(1 - F(v_3)\right)^{n-3} \mathrm{d}v_3\, \mathrm{d}v_2\, \mathrm{d}v_1. \end{aligned}$$

This multiple integral is often very difficult to compute, since the inner integral

$$\int_{v_2}^{2v_2 - v_1} f(v_3) \left(1 - F(v_3)\right)^{n-3} \mathrm{d}v_3$$

is the most complex one. Using Fubini's theorem for positive functions, we can simplify the equation by changing the order of integration such that the most complex integral comes in the outer shell.

$$
\begin{aligned}
\mathrm{CF}_{f,\min}(n) &= \int_{-\infty}^{\infty}\int_{-\infty}^{v_3}\int_{-\infty}^{2v_2-v_3} f_{1,2,3:n}\left(v_1,v_2,v_3\right)\,\mathrm{d}v_1\,\mathrm{d}v_2\,\mathrm{d}v_3 \\
&= \frac{n!}{(n-3)!}\int_{-\infty}^{\infty} f\left(v_3\right)\left(1-F\left(v_3\right)\right)^{n-3}\int_{-\infty}^{v_3} f\left(v_2\right)\int_{-\infty}^{2v_2-v_3} f\left(v_1\right)\mathrm{d}v_1\,\mathrm{d}v_2\,\mathrm{d}v_3 \\
&= \frac{n!}{(n-3)!}\int_{-\infty}^{\infty} f\left(v_3\right)\left(1-F\left(v_3\right)\right)^{n-3}\int_{-\infty}^{v_3} f\left(v_2\right)F\left(2v_2-v_3\right)\,\mathrm{d}v_2\,\mathrm{d}v_3 \quad (5.5)
\end{aligned}
$$

Note that $v_1 \leq 2v_2 - v_3$ ensures $v_1 \leq v_2$ as well, since $2v_1 - v_3 \leq v_2$ for $v_2 \leq v_3$ holds.

In analogy we compute the Cordel frequency for maximization problems with density f. With

$$
f_{n-2,n-1,n:n}\left(v_{n-2},v_{n-1},v_n\right) = \frac{n!}{(n-3)!}F\left(v_{n-2}\right)^{n-3} f\left(v_{n-2}\right) f\left(v_{n-1}\right) f\left(v_n\right)
$$

for $-\infty < v_{n-2} \leq v_{n-1} \leq v_n < \infty$ (cf. [ABN 1992, p. 26]) and

$$
d_{n-1} \geq d_{n-2} \quad\Leftrightarrow\quad v_n - v_{n-1} \geq v_{n-1} - v_{n-2} \quad\Leftrightarrow\quad v_n \geq 2v_{n-1} - v_{n-2} \geq v_{n-1}
$$

for $v_{n-1} \geq v_{n-2}$ follows

$$
\begin{aligned}
&\mathrm{CF}_{f,\max}(n) = P\left(V_{n:n} - V_{n-1:n} \geq V_{n-1:n} - V_{n-2:n}\right) \\
&= \int_{-\infty}^{\infty}\int_{v_{n-2}}^{\infty}\int_{2v_{n-1}-v_{n-2}}^{\infty} f_{n-2,n-1,n:n}\left(v_{n-2},v_{n-1},v_n\right)\,\mathrm{d}v_n\,\mathrm{d}v_{n-1}\,\mathrm{d}v_{n-2} \\
&= \frac{n!}{(n-3)!}\int_{-\infty}^{\infty} F\left(v_{n-2}\right)^{n-3} f\left(v_{n-2}\right)\int_{v_{n-2}}^{\infty} f\left(v_{n-1}\right)\int_{2v_{n-1}-v_{n-2}}^{\infty} f\left(v_n\right)\,\mathrm{d}v_n\,\mathrm{d}v_{n-1}\,\mathrm{d}v_{n-2} \\
&= \frac{n!}{(n-3)!}\int_{-\infty}^{\infty} F\left(v_{n-2}\right)^{n-3} f\left(v_{n-2}\right)\int_{v_{n-2}}^{\infty} f\left(v_{n-1}\right)\left(1-F\left(2v_{n-1}-v_{n-2}\right)\right)\,\mathrm{d}v_{n-1}\,\mathrm{d}v_{n-2}\,.
\end{aligned}
$$

The final equation (5.5) is used to compute the Cordel frequency for different distributions f. The results are presented in the next subsections with Section 5.1.1 dealing with distributions on the interval $[0,1]$, Section 5.1.2 considering distributions on $[0,\infty)$, and Section 5.1.3 covering distributions on the whole range $\mathbb{R}$. Appendix L starting on page 239 gives a summary of the considered distributions containing the density and cumulative density functions as well as the graphs of the density functions. Furthermore, Appendix M.1 starting on page 243 includes the Maple worksheets [Maple 2008] containing all computations and results.

5.1.1 Cordel Frequencies for Distributions on $[0, 1]$

The most prominent probability distribution on $[0,1]$ is the continuous uniform distribution with density function

$$
u(x) := \begin{cases} 1 & \text{for } 0 \leq x \leq 1\,, \\ 0 & \text{for } x < 0 \text{ or } x > 1\,. \end{cases}
$$

For comparison we also examined two probability distributions with linearly increasing (i for increasing) and decreasing (d for down) density function, namely

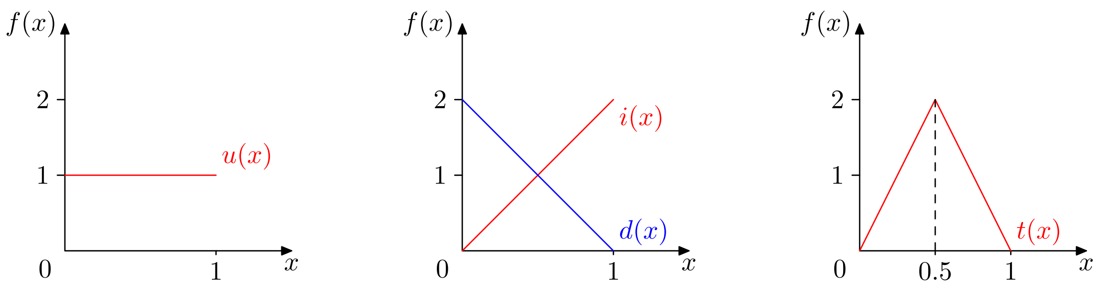
$$i(x) := \begin{cases} 2x & \text{for } 0 \leq x \leq 1\,, \\ 0 & \text{for } x < 0 \text{ or } x > 1 \end{cases} \qquad \text{and} \qquad d(x) := \begin{cases} 2-2x & \text{for } 0 \leq x \leq 1\,, \\ 0 & \text{for } x < 0 \text{ or } x > 1\,. \end{cases}$$

In fact, the decreasing distribution is not discussed explicitly since maximizing the increasing distribution provides the same Cordel frequency as minimizing the decreasing distribution.

Finally, the centered triangular distribution, which is a combination of the increasing and decreasing distribution, was considered.

$$t(x) := \begin{cases} 4x & \text{for } 0 \leq x \leq \frac{1}{2} \\ 4-4x & \text{for } \frac{1}{2} \leq x \leq 1 \\ 0 & \text{for } x < 0 \text{ and } x > 1 \end{cases}$$

The corresponding cumulative distribution functions are to be found in Appendix L on page 239. The figures below show the graphs of the density functions.



(a) uniform distribution **(b)** increasing and decreasing distribution **(c)** triangular distribution

Before we compute the Cordel frequencies for these three distributions we give thought to the integration intervals. The equation

$$\begin{aligned} \mathrm{CF}_{f,\min}(n) &= \frac{n!}{(n-3)!} \int_{-\infty}^{\infty} f(v_3)\,(1-F(v_3))^{n-3} \int_{-\infty}^{v_3} f(v_2)\,F(2v_2-v_3)\,\mathrm{d}v_2\,\mathrm{d}v_3 \\ &= \frac{n!}{(n-3)!} \int_{0}^{1} f(v_3)\,(1-F(v_3))^{n-3} \int_{\frac{v_3}{2}}^{v_3} f(v_2)\,F(2v_2-v_3)\,\mathrm{d}v_2\,\mathrm{d}v_3\,. \end{aligned}$$

holds since $f(x) = 0$ for $x \notin [0,1]$ and $F(2v_2 - v_3) = 0$ for $2v_2 - v_3 < 0$ which is equivalent to $v_2 < \frac{v_3}{2}$. Therewith we can compute the Cordel frequencies for the three distributions.

Uniform Distribution

For minimizing the uniform distribution the Cordel frequency $\mathrm{CF}_{u,\min}(n) = \frac{1}{2}$ arises as the following computation shows. We used a compact representation where complex calculation steps may be verified with Maple [Maple 2008], for example.

$$\begin{aligned}
\mathrm{CF}_{u,\min}(n) &= \frac{n!}{(n-3)!}\int_0^1 u\left(v_3\right)\left(1-U\left(v_3\right)\right)^{n-3}\int_{\frac{v_3}{2}}^{v_3} u\left(v_2\right)U(2v_2-v_3)\,\mathrm{d}v_2\,\mathrm{d}v_3\\
&= \frac{n!}{(n-3)!}\int_0^1 \left(1-v_3\right)^{n-3}\int_{\frac{v_3}{2}}^{v_3} 2v_2-v_3\,\mathrm{d}v_2\,\mathrm{d}v_3\\
&= \frac{n!}{(n-3)!}\int_0^1 \left(1-v_3\right)^{n-3}\left[v_2^2-v_3v_2\right]_{v_2=\frac{v_3}{2}}^{v_3}\,\mathrm{d}v_3\\
&= \frac{n!}{(n-3)!}\int_0^1 \left(1-v_3\right)^{n-3}\left(v_3^2-v_3^2-\frac{1}{4}v_3^2+\frac{1}{2}v_3^2\right)\,\mathrm{d}v_3\\
&= \frac{n!}{(n-3)!}\cdot\frac{1}{4}\int_0^1 \left(1-v_3\right)^{n-3}v_3^2\,\mathrm{d}v_3\\
&= \frac{1}{4}\left[-(1-v_3)^{n-2}\left(v_3^2n^2+v_3n-3v_3^2n-4v_3+2v_3^2+2\right)\right]_{v_3=0}^{1}\\
&= \frac{1}{4}\cdot 2=\frac{1}{2}
\end{aligned}$$

Obviously, the Cordel frequency for maximization and minimization problem has to be the same in case of the uniform distribution. Thus,

$$\mathrm{CF}_{u,\min}(n)=\mathrm{CF}_{u,\max}(n)=\frac{1}{2}\,.$$

Increasing Distribution

$$\begin{aligned}
\mathrm{CF}_{i,\min}(n) &= \frac{n!}{(n-3)!}\int_0^1 i\left(v_3\right)\left(1-I\left(v_3\right)\right)^{n-3}\int_{\frac{v_3}{2}}^{v_3} i\left(v_2\right)I(2v_2-v_3)\,\mathrm{d}v_2\,\mathrm{d}v_3\\
&= \frac{n!}{(n-3)!}\int_0^1 2v_3\cdot\left(1-v_3^2\right)^{n-3}\int_{\frac{v_3}{2}}^{v_3} 2v_2\cdot(2v_2-v_3)^2\,\mathrm{d}v_2\,\mathrm{d}v_3\\
&= \frac{4\cdot n!}{(n-3)!}\int_0^1 v_3\left(1-v_3^2\right)^{n-3}\int_{\frac{v_3}{2}}^{v_3} 4v_2^3-4v_3v_2^2+v_3^2v_2\,\mathrm{d}v_2\,\mathrm{d}v_3\\
&= \frac{4\cdot n!}{(n-3)!}\int_0^1 \left(1-v_3^2\right)^{n-3}\left[v_2^4-\frac{4}{3}v_3v_2^3+\frac{1}{2}v_3^2v_2^2\right]_{v_2=\frac{v_3}{2}}^{v_3}\,\mathrm{d}v_3\\
&= \frac{7}{12}\frac{n!}{(n-3)!}\int_0^1 \left(1-v_3^2\right)^{n-3}v_3^5\,\mathrm{d}v_3\\
&= \frac{7}{12}\left[-\frac{1}{2}(1-v_3^2)^{n-2}\left(v_3^4n^2+2v_3^2n-3v_3^4n-4v_3^2+2v_3^2+2\right)\right]_{v_3=0}^{1}\\
&= \frac{7}{12}
\end{aligned}$$

In analogy the Cordel frequency for maximization problems with density i was computed.

$$
\begin{aligned}
&\mathrm{CF}_{i,\max}(n)\\
&= \frac{n!}{(n-3)!}\int_0^1 I\left(v_{n-2}\right)^{n-3} i\left(v_{n-2}\right)\int_{v_{n-2}}^1 i\left(v_{n-1}\right)\left(1-I\left(2v_{n-1}-v_{n-2}\right)\right)\,\mathrm{d}v_{n-1}\,\mathrm{d}v_{n-2}\\
&= \frac{n!}{(n-3)!}\int_0^1 2v_{n-2}^{2n-5}\int_{v_{n-2}}^1 2v_{n-1}\underbrace{\left(1-I\left(2v_{n-1}-v_{n-2}\right)\right)}_{=0\text{ for }2v_{n-1}-v_{n-2}\geq 1}\,\mathrm{d}v_{n-1}\,\mathrm{d}v_{n-2}\\
&= \frac{n!}{(n-3)!}\int_0^1 2v_{n-2}^{2n-5}\int_{v_{n-2}}^{\frac{1}{2}+\frac{1}{2}v_{n-2}} 2v_{n-1}\left(1-\left(2v_{n-1}-v_{n-2}\right)^2\right)\,\mathrm{d}v_{n-1}\,\mathrm{d}v_{n-2}\\
&= \frac{n!}{(n-3)!}\int_0^1 2v_{n-2}^{2n-5}\int_{v_{n-2}}^{\frac{1}{2}+\frac{1}{2}v_{n-2}} -8v_{n-1}^3+8v_{n-1}^2v_{n-2}+2v_{n-1}\left(1-v_{n-2}^2\right)\,\mathrm{d}v_{n-1}\,\mathrm{d}v_{n-2}\\
&= \frac{n!}{(n-3)!}\int_0^1 2v_{n-2}^{2n-5}\left[-2v_{n-1}^4+\frac{8}{3}v_{n-1}^3v_{n-2}+v_{n-1}^2\left(1-v_{n-2}^2\right)\right]_{v_{n-1}=v_{n-2}}^{\frac{1}{2}+\frac{1}{2}v_{n-2}}\,\mathrm{d}v_{n-2}\\
&= \frac{n!}{(n-3)!}\int_0^1 2v_{n-2}^{2n-5}\left(\frac{7}{24}v_{n-2}^4-\frac{3}{4}v_{n-2}^2+\frac{1}{3}v_{n-2}+\frac{1}{8}\right)\,\mathrm{d}v_{n-2}\\
&= \frac{n!}{(n-3)!}\left[\frac{7}{24n}v_{n-2}^{2n}-\frac{3}{4(n-1)}v_{n-2}^{2n-2}+\frac{2}{3(2n-3)}v_{n-2}^{2n-3}+\frac{1}{8(n-2)}v_{n-2}^{2n-4}\right]_{v_{n-2}=0}^1\\
&= \frac{n!}{(n-3)!}\left(\frac{7}{24n}-\frac{3}{4(n-1)}+\frac{2}{3(2n-3)}+\frac{1}{8(n-2)}\right)\\
&= \frac{n!}{(n-3)!}\cdot\frac{4n-7}{4n(n-1)(n-2)(2n-3)}\\
&= \frac{4n-7}{4(2n-3)}\xrightarrow{n\to\infty}\frac{1}{2}
\end{aligned}
$$

Thus, $\mathrm{CF}_{i,\max}(n)$ is strictly monotonic increasing and converges to $\frac{1}{2}$ for $n \to \infty$.

There is a very considerable difference between this and the previous results. While the Cordel frequencies for minimization problems with uniform and increasing density function as well as the Cordel frequency for maximization problems with uniform density function are both independent of n, the Cordel frequency for maximization problems with density i depends on n. But we will see that the Cordel frequency for the centered triangular distributions also depends on n.

Centered Triangular Distribution

For the centered triangular distribution the computation of the Cordel frequency is a little bit more difficult. Here we have to split the multiple integral into four parts and add up the values of the four integrals. This interval decomposition is caused by the density function t of the triangular distribution which is a composite function.

Thus we get

$$\begin{aligned}\mathrm{CF}_{t,\min}(n) &= \frac{n!}{(n-3)!}\int_0^1 t\,(v_3)\,(1-T\,(v_3))^{n-3}\int_{\frac{v_3}{2}}^{v_3} t\,(v_2)\,T(2v_2-v_3)\,\mathrm{d}v_2\,\mathrm{d}v_3\\ &= \frac{n!}{(n-3)!}\,(\mathrm{Int}_1+\mathrm{Int}_2+\mathrm{Int}_3+\mathrm{Int}_4)\end{aligned}$$

with

$$\begin{aligned}\mathrm{Int}_1 &:= \int_0^{\frac{1}{2}} 4v_3\cdot\left(1-2v_3^2\right)^{n-3}\int_{\frac{v_3}{2}}^{v_3} 4v_2\cdot 2(2v_2-v_3)^2\,\mathrm{d}v_2\,\mathrm{d}v_3\,,\\ \mathrm{Int}_2 &:= \int_{\frac{1}{2}}^1 (4-4v_3)\cdot\left(2v_3^2-4v_3+2\right)^{n-3}\int_{\frac{v_3}{2}}^{\frac{1}{2}} 4v_2\cdot 2(2v_2-v_3)^2\,\mathrm{d}v_2\,\mathrm{d}v_3\,,\\ \mathrm{Int}_3 &:= \int_{\frac{1}{2}}^1 (4-4v_3)\cdot\left(2v_3^2-4v_3+2\right)^{n-3}\int_{\frac{1}{2}}^{\frac{1}{4}+\frac{v_3}{2}} (4-4v_2)\cdot 2\,(2v_2-v_3)^2\;\mathrm{d}v_2\,\mathrm{d}v_3\,,\\ \mathrm{Int}_4 &:= \int_{\frac{1}{2}}^1 (4-4v_3)\cdot\left(2v_3^2-4v_3+2\right)^{n-3}\\ &\qquad\qquad \int_{\frac{1}{4}+\frac{v_3}{2}}^{v_3} (4-4v_2)\cdot\left(-2\,(2v_2-v_3)^2+4\,(2v_2-v_3)-1\right)\;\mathrm{d}v_2\,\mathrm{d}v_3\,.\end{aligned}$$

These four integrals were not calculated by hand. Adding up the values obtained with the help of Maple [Maple 2008] (cf. Maple worksheet in Appendix M.1 starting on page 243) the Cordel frequency

$$\mathrm{CF}_{t,\min}(n) = \frac{7}{12} - \frac{5n-3}{6\cdot 2^n(2n-3)}$$

arises. As in the case of the uniform distribution, again the Cordel frequency is the same for minimization and maximization problems (cf. Appendix M.1 starting on page 243). Thus,

$$\mathrm{CF}_{t,\max}(n) = \mathrm{CF}_{t,\min}(n) = \frac{7}{12} - \frac{5n-3}{6\cdot 2^n(2n-3)} \xrightarrow{n\to\infty} \frac{7}{12}\,.$$

$\mathrm{CF}_{t,\min}(n)$ and $\mathrm{CF}_{t,\max}(n)$ are strictly monotonic increasing in n. But while the Cordel frequency for maximizing optimization problems with the increasing density i was always smaller than $\frac{1}{2}$, here, $\frac{1}{2} \leq \mathrm{CF}_{t,\min}(n) = \mathrm{CF}_{t,\max}(n)$ for $n \geq 3$ holds. The limit of the sequence is $\frac{7}{12}$ which is the Cordel frequency of the increasing distribution. This result was actually to be expected since the left side of the triangular distribution (where the three best functional values are usually to be found) is an increasing distribution.

Figure 5.1.1 on the next page shows the six computed Cordel frequencies for distributions on $[0,1]$ for comparison.

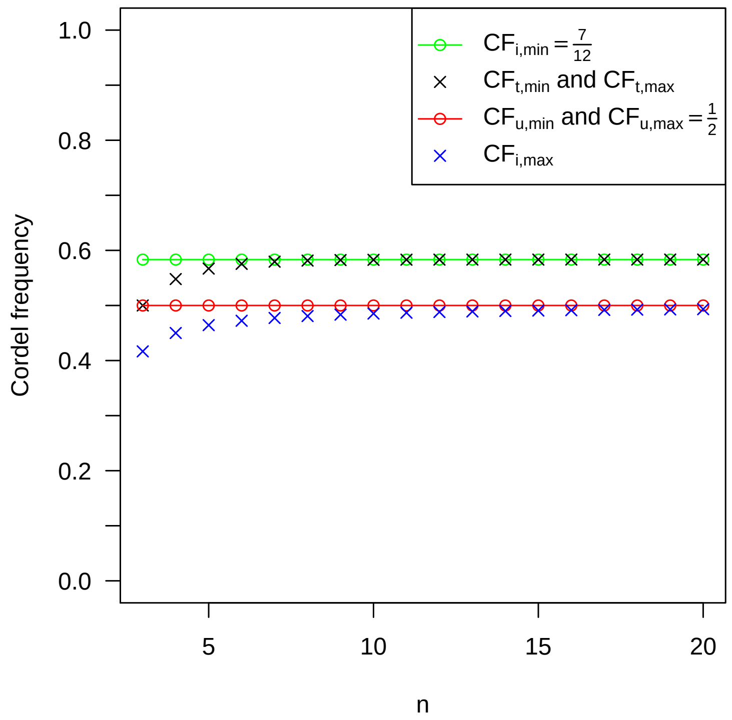


Figure 5.1.1: Cordel frequencies sorted from high to low: minimize i (green), minimize or maximize t (black), minimize or maximize u (red), maximize i (blue).

5.1.2 Cordel Frequencies for Distributions on $[0, \infty)$

We considered three distributions on the non-negative real numbers.

- The exponential distribution with parameter $\lambda > 0$:

$$e_\lambda(x) := \begin{cases} \lambda \mathrm{e}^{-\lambda x} & \text{for } x \leq 0, \\ 0 & \text{for } x < 0. \end{cases}$$

- The (re-)normalized positive part of the standard normal distribution:

$$\overline{\varphi}(x) := \begin{cases} 2 \cdot \varphi(x) = \frac{2}{\sqrt{2\pi}} \mathrm{e}^{-\frac{x^2}{2}} & \text{for } 0 \leq x, \\ 0 & \text{for } x < 0. \end{cases}$$

  This is the same as the density function of the absolute value of a standard normal random variable.

- The (re-)normalized positive part of the logistic distribution with mean value $\mu = 0$ and scale parameter $s = 1$:

$$\bar{l}_s(x) := \begin{cases} 2 \cdot l(x) = \frac{2\mathrm{e}^{-x/s}}{s\left(1+\mathrm{e}^{-x/s}\right)^2} & \text{for } 0 \leq x, \\ 0 & \text{for } x < 0. \end{cases}$$

  This is the same as the density function of the absolute value of a logistic random variable.

See Appendix L starting on page 239 for the cumulated distribution functions and graphs of the density functions.

Exponential Distribution
We compute the Cordel frequency for minimization problems with exponential density as follows.

$$\begin{aligned}
\mathrm{CF}_{e_\lambda,\min}(n) &= \frac{n!}{(n-3)!}\int_0^\infty e_\lambda\left(v_3\right)\left(1-E_\lambda\left(v_3\right)\right)^{n-3}\int_{\frac{v_3}{2}}^{v_3} e_\lambda\left(v_2\right)E_\lambda(2v_2-v_3)\,\mathrm{d}v_2\,\mathrm{d}v_3\\
&= \frac{n!}{(n-3)!}\int_0^\infty \lambda\mathrm{e}^{-\lambda v_3}\cdot\mathrm{e}^{-\lambda v_3(n-3)}\int_{\frac{v_3}{2}}^{v_3}\lambda\mathrm{e}^{-\lambda v_2}\cdot\left(1-\mathrm{e}^{-\lambda(2v_2-v_3)}\right)\,\mathrm{d}v_2\,\mathrm{d}v_3\\
&= \frac{n!}{(n-3)!}\int_0^\infty \lambda\mathrm{e}^{-\lambda v_3(n-2)}\int_{\frac{v_3}{2}}^{v_3}\lambda\mathrm{e}^{-\lambda v_2}-\lambda\mathrm{e}^{-\lambda(3v_2-v_3)}\,\mathrm{d}v_2\,\mathrm{d}v_3\\
&= \frac{n!}{(n-3)!}\int_0^\infty \lambda\mathrm{e}^{-\lambda v_3(n-2)}\left[-\mathrm{e}^{-\lambda v_2}+\frac{1}{3}\mathrm{e}^{-\lambda(3v_2-v_3)}\right]_{v_2=\frac{v_3}{2}}^{v_3}\,\mathrm{d}v_3\\
&= \frac{n!}{(n-3)!}\int_0^\infty \lambda\mathrm{e}^{-\lambda v_3(n-2)}\cdot\left(-\mathrm{e}^{-\lambda v_3}+\frac{1}{3}\mathrm{e}^{-2\lambda v_3}+\frac{2}{3}\mathrm{e}^{-\frac{1}{2}\lambda v_3}\right)\,\mathrm{d}v_3\\
&= \frac{n!}{(n-3)!}\int_0^\infty -\lambda\mathrm{e}^{-\lambda v_3(n-1)}+\frac{\lambda}{3}\mathrm{e}^{-\lambda v_3 n}+\frac{2\lambda}{3}\mathrm{e}^{-\lambda v_3\left(n-\frac{3}{2}\right)}\,\mathrm{d}v_3\\
&= \frac{n!}{(n-3)!}\left[\frac{1}{n-1}\mathrm{e}^{-\lambda v_3(n-1)}-\frac{1}{3n}\mathrm{e}^{-\lambda v_3 n}-\frac{4}{6n-9}\mathrm{e}^{-\lambda v_3\left(n-\frac{3}{2}\right)}\right]_{v_3=0}^{\infty}\\
&= \frac{n!}{(n-3)!}\left(-\frac{1}{(n-1)}+\frac{1}{3n}+\frac{4}{6n-9}\right)\\
&= \frac{n-2}{2n-3}\xrightarrow{n\to\infty}\frac{1}{2}
\end{aligned}$$

We see that the Cordel frequency does not depend on the parameter λ of the exponential distribution. Independently of λ, the Cordel frequency is strictly monotonic increasing in n and converges to $\frac{1}{2}$.

The following Cordel frequency occurs for maximization problems with exponential density.

$$\begin{aligned}
\mathrm{CF}_{e_\lambda,\max}(n) &= \frac{n!}{(n-3)!}\int_0^\infty e_\lambda\left(v_{n-2}\right)E_\lambda\left(v_{n-2}\right)^{n-3}\\
&\qquad\int_{v_{n-2}}^\infty e_\lambda\left(v_{n-1}\right)\left(1-E_\lambda(2v_{n-1}-v_{n-2})\right)\,\mathrm{d}v_{n-1}\,\mathrm{d}v_{n-2}\\
&= \frac{n!}{(n-3)!}\int_0^\infty \lambda\mathrm{e}^{-\lambda v_{n-2}}\left(1-\mathrm{e}^{-\lambda v_{n-2}}\right)^{n-3}\int_{v_{n-2}}^\infty \lambda\mathrm{e}^{-\lambda(3v_{n-1}-v_{n-2})}\,\mathrm{d}v_{n-1}\,\mathrm{d}v_{n-2}\\
&= \frac{n!}{(n-3)!}\int_0^\infty \lambda\mathrm{e}^{-\lambda v_{n-2}}\left(1-\mathrm{e}^{-\lambda v_{n-2}}\right)^{n-3}\left[-\frac{1}{3}\mathrm{e}^{-\lambda(3v_{n-1}-v_{n-2})}\right]_{v_{n-1}=v_{n-2}}^{\infty}\,\mathrm{d}v_{n-2}\\
&= \frac{n!}{(n-3)!}\int_0^\infty \frac{\lambda}{3}\mathrm{e}^{-3\lambda v_{n-2}}\left(1-\mathrm{e}^{-\lambda v_{n-2}}\right)^{n-3}\,\mathrm{d}v_{n-2}
\end{aligned}$$

$$
\begin{aligned}
&= \frac{n!}{(n-3)!}\int_0^1 \frac{1}{3}(1-t)^2\, t^{n-3}\,\mathrm{d}t \qquad\qquad (\text{substitution } t := 1-\mathrm{e}^{-\lambda v_{n-2}})\\
&= \frac{n!}{3(n-3)!}\left[\frac{1}{n}t^n - \frac{2}{n-1}t^{n-1} + \frac{1}{n-2}t^{n-2}\right]_{t=0}^{1}\\
&= \frac{n!}{3(n-3)!}\left(\frac{1}{n} - \frac{2}{n-1} + \frac{1}{n-2}\right)\\
&= \frac{2}{3}
\end{aligned}
$$

Again, the Cordel frequency is independent of λ. But in contrast to the minimization problem now the Cordel frequency is also independent of n.

Normalized Positive Part of the Standard Normal Distribution
Since the cumulative distribution function of the standard normal distribution can only be evaluated numerically, which holds for the normalized positive part too, the Cordel frequency cannot be computed exactly. But with the help of Maple [Maple 2008] we calculated the numerical values for several n. The results are shown in Figure 5.1.2.

Normalized Positive Part of the Logistic Distribution
Again, we were not able to compute the Cordel frequency exactly even though the cumulative distribution function can be computed. Instead we used Maple's [Maple 2008] numerical integration for the computation of approximated values.

Figure 5.1.2 shows the precisely calculated Cordel frequencies for optimization problems with exponential density and the numerical results for the normalized positive part of the standard normal and logistic distribution.

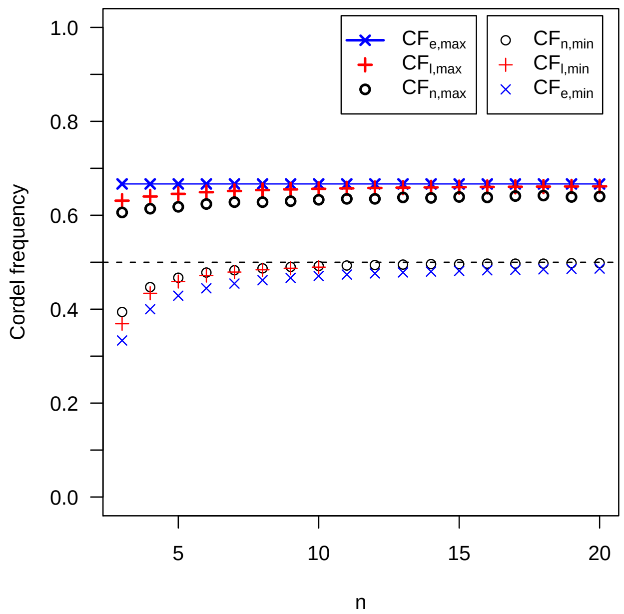


Figure 5.1.2: Cordel frequencies sorted from high to low: exponential distribution e_λ (blue), positive part of the logistic distribution $l = \bar{l}_s$ with scale parameter $s = 1$ (red), positive part of the standard normal distribution $n = \overline{\varphi}$ (black).

5.1.3 Cordel Frequencies for Distributions on $(-\infty, \infty)$

In addition, not only the normalized positive part of the standard normal distribution and logistic distribution were considered, but also the underlying complete distributions on $(-\infty, \infty)$. But as in the case of the triangular distribution the results will converge to the Cordel frequencies for maximizing the positive part. That is why we do not present further results here.

5.1.4 Summary of all Results

Altogether, the following Cordel frequencies sorted from high to low were determined.

$$\begin{aligned}
\mathrm{CF}_{e_\lambda,\max}(n) &= \frac{2}{3} \\
\mathrm{CF}_{i,\min}(n) &= \frac{7}{12} \\
\mathrm{CF}_{t,\min}(n) &= \mathrm{CF}_{t,\max}(n) = \frac{7}{12} - \frac{5n-3}{6 \cdot 2^n(2n-3)} \\
\mathrm{CF}_{u,\min}(n) &= \mathrm{CF}_{u,\max}(n) = \frac{1}{2} \\
\mathrm{CF}_{i,\max}(n) &= \frac{4n-7}{8n-12} \\
\mathrm{CF}_{e_\lambda,\min}(n) &= \frac{n-2}{2n-3}
\end{aligned}$$

Additionally, Monte-Carlo Integration for $n = 10,000$ lead us to the following conjectures.

(i) $\mathrm{CF}_{\varphi,\max}(n), \mathrm{CF}_{\overline{\varphi},\max}(n), \mathrm{CF}_{l_s,\max}(n)$, and $\mathrm{CF}_{\bar{l}_s,\max}(n)$ seem to converge strictly monotonic increasing to $\frac{2}{3}$.

(ii) $\mathrm{CF}_{\varphi,\min}(n), \mathrm{CF}_{\overline{\varphi},\min}(n), \mathrm{CF}_{l_s,\min}(n)$, and $\mathrm{CF}_{\bar{l}_s,\min}(n)$ seem to converge strictly monotonic increasing to $\frac{1}{2}$.

It turned out that the Cordel frequency often differs for minimization and maximization problems of the same distribution which is not surprising. Furthermore, unfortunately no clear trend (Cordel frequency almost always above ore below 50%) was observable. But in fact, the proved or suggested limits of the Cordel frequency are all greater than or equal to 50%. Therewith we come back to the question following Conjecture 5.1.4 (page 96). There we asked, whether each value $v \in [0, 1]$ is a possible limit to $\mathrm{CF}_{f,\max}(n)$ as $n \to \infty$. The following Section 5.1.5 deals with this question a little bit more detailed. Afterwards in Section 5.1.6 (starting on page 108) we look briefly at the expected values of the differences d_1 and d_2.

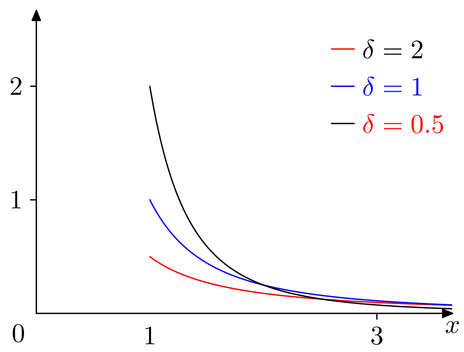
5.1.5 Experimental Results for the δ-Distribution

In this section we consider a special type of distributions, called δ-distributions.

Definition 5.1.5 **(δ-Distribution)**
A ***δ-distributed*** *optimization problem ($\delta > 0$) is an maximization problem with density function*

$$f_\delta(x) = \begin{cases} 0, & \text{for } x \le 1 \\ \frac{\delta}{x^{1+\delta}} & \text{for } x \ge 1 \end{cases}$$

as shown in the figure on the right.



By Monte Carlo integration we tried to determine the limit $\lim_{n\to\infty} \mathrm{CF}_{f_\delta,\max}(n)$ for various $\delta > 0$. The following Figure 5.1.3 shows the experimentally observed Cordel frequencies $\mathrm{CF}_{f_\delta,\max}(100,000)$ which are a very good approximation for the real limits. These experimental considerations show, that the δ-distributed maximization problems provide limits between $\frac{2}{3}$ and 1. It seems, that each value v in the open interval $\left(\frac{2}{3}, 1\right)$ occurs as the limit of $\mathrm{CF}_{f_\delta,\max}(n)$ for a $\delta > 0$. It remains an open problem to compute the real limits and therewith to prove this observation.

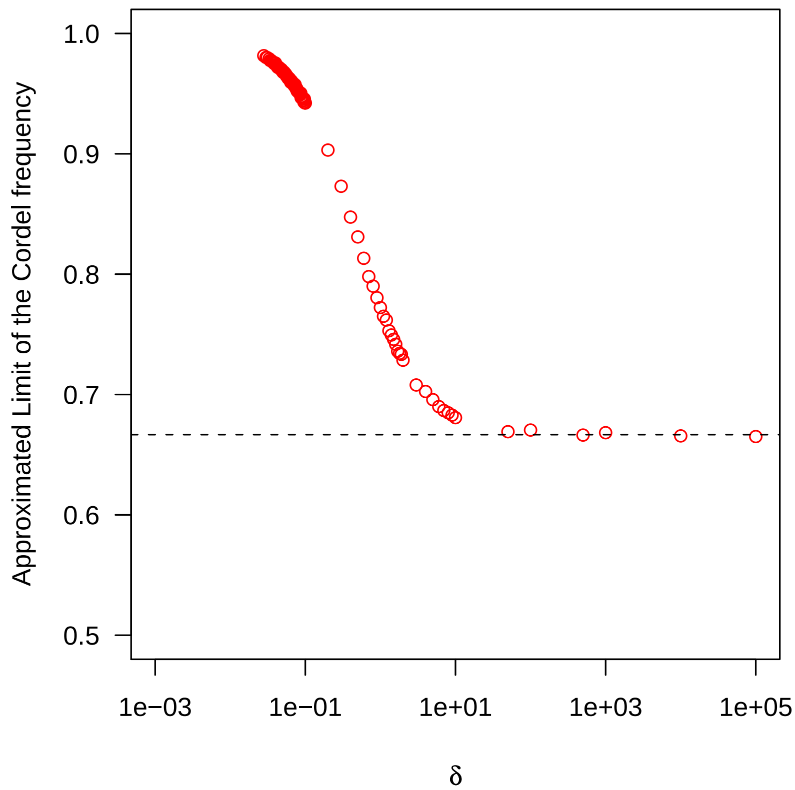


Figure 5.1.3: Experimentally established limits of the Cordel frequency for δ-distributed maximization problems.

But still the question is whether the remaining interval $\left[0, \frac{2}{3}\right]$ also provides possible limits for the Cordel frequency. As we saw in the summary in Subsection 5.1.4 on page 106, all proven or suggested limits of the Cordel frequencies were greater than or equal to 0.5. That makes us conjecture

$$\lim_{n\to\infty} \mathrm{CF}_{f,\min} \geq \frac{1}{2} \quad \text{and} \quad \lim_{n\to\infty} \mathrm{CF}_{f,\max} \geq \frac{1}{2}\,.$$

5.1.6 Expected values $\mathbb{E}\left(d_1\right)$ and $\mathbb{E}\left(d_2\right)$

With the joint density function of the two smallest values

$$f_{1,2:n}\left(v_1, v_2\right) = \frac{n!}{(n-2)!}\left(1 - F\left(v_2\right)\right)^{n-2} f\left(v_1\right) f\left(v_2\right)$$

one can compute the expected value of the difference d_1. Thereby the calculation step marked with PI is the result of integration by parts.

$$\begin{aligned}
\mathbb{E}_{f,\min}\left(d_1\right) &= n(n-1)\int_{-\infty}^{\infty} f\left(v_2\right)\left(1-F\left(v_2\right)\right)^{n-2}\int_{-\infty}^{v_2}\left(v_2-v_1\right) f\left(v_1\right)\mathrm{d}v_1\mathrm{d}v_2\\
&= n(n-1)\int_{-\infty}^{\infty} f\left(v_2\right)\left(1-F\left(v_2\right)\right)^{n-2}\\
&\qquad\qquad \left[v_2\int_{-\infty}^{v_2} f\left(v_1\right)\mathrm{d}v_1 - \int_{-\infty}^{v_2} v_1 f\left(v_1\right)\mathrm{d}v_1\right]\mathrm{d}v_2\\
&\overset{\mathrm{PI}}{=} n(n-1)\int_{-\infty}^{\infty} f\left(v_2\right)\left(1-F\left(v_2\right)\right)^{n-2}\\
&\qquad\qquad \left(v_2 F\left(v_2\right) - \left[v_1 F\left(v_1\right)\right]_{v_1=-\infty}^{v_2} + \int_{-\infty}^{v_2} F\left(v_1\right)\mathrm{d}v_1\right)\mathrm{d}v_2\\
&= n(n-1)\int_{-\infty}^{\infty} f\left(v_2\right)\left(1-F\left(v_2\right)\right)^{n-2}\int_{-\infty}^{v_2} F\left(v_1\right)\mathrm{d}v_1\mathrm{d}v_2
\end{aligned}$$

In analogy, with the help of density function

$$f_{2,3:n}\left(v_2, v_3\right) = \frac{n!}{(n-3)!} F\left(v_2\right)\left(1 - F\left(v_3\right)\right)^{n-3} f\left(v_2\right) f\left(v_3\right)$$

one computes

$$\mathbb{E}_{f,\min}\left(d_2\right) = \frac{n!}{(n-3)!}\int_{-\infty}^{\infty} f\left(v_3\right)\left(1-F\left(v_3\right)\right)^{n-3}\int_{-\infty}^{v_3}\left(v_3-v_2\right) F(v_2) f\left(v_2\right)\mathrm{d}v_2\mathrm{d}v_3\,.$$

Furthermore we compute

$$\mathbb{E}_{f,\max}\left(d_1\right) := \mathbb{E}\left(V_{n:n} - V_{n-1:n}\right) \quad \text{and} \quad \mathbb{E}_{f,\max}\left(d_2\right) := \mathbb{E}\left(V_{n-1:n} - V_{n-2:n}\right)$$

which represent the differences of the three best functional values for maximization problems.

The following expected values for minimization problems are found by inserting the different density functions.

$$\mathbb{E}_{u,\min}(d_1) = \frac{1}{n+1} \qquad \mathbb{E}_{u,\min}(d_2) = \frac{1}{n+1}$$

$$\mathbb{E}_{i,\min}(d_1) = \frac{\sqrt{\pi}}{4}\frac{n!}{\Gamma\left(n+\frac{3}{2}\right)} \qquad \mathbb{E}_{i,\min}(d_2) = \frac{3\sqrt{\pi}}{16}\frac{n!}{\Gamma\left(n+\frac{3}{2}\right)}$$

$$\mathbb{E}_{e_\lambda,\min}(d_1) = \frac{1}{\lambda(n-1)} \qquad \mathbb{E}_{e_\lambda,\min}(d_2) = \frac{1}{\lambda(n-2)}$$

For maximization problems the following expected values arise:

$$\mathbb{E}_{u,\max}(d_1) = \frac{1}{n+1} \qquad \mathbb{E}_{u,\max}(d_2) = \frac{1}{n+1}$$

$$\mathbb{E}_{i,\max}(d_1) = \frac{2n}{4n^2-1} \qquad \mathbb{E}_{i,\max}(d_2) = \frac{4n(n-1)}{8n^3-12n^2-2n+3}$$

$$\mathbb{E}_{e_\lambda,\max}(d_1) = \frac{1}{\lambda} \qquad \mathbb{E}_{e_\lambda,\max}(d_2) = \frac{1}{2\lambda}\,.$$

For the remaining distributions no simple closed formulas have been discovered.

Finally we compare these expected values by considering the quotients

$$Q_{f,\min} := \frac{\mathbb{E}_{f,\min}(d_1)}{\mathbb{E}_{f,\min}(d_1)+\mathbb{E}_{f,\min}(d_2)} \quad \text{and} \quad Q_{f,\max} := \frac{\mathbb{E}_{f,\max}(d_1)}{\mathbb{E}_{f,\max}(d_1)+\mathbb{E}_{f,,\max}(d_2)}\,.$$

Computation leads to the following values.

$$Q_{u,\min} = \frac{1}{2} \qquad Q_{u,\max} = \frac{1}{2}$$

$$Q_{i,\min} = \frac{4}{7} \qquad Q_{i,\max} = \frac{2n-3}{4n-5}$$

$$Q_{e_\lambda,\min} = \frac{n-2}{2n-3} \qquad Q_{e_\lambda,\max} = \frac{2}{3}$$

Remark 5.1.6
In case of the uniform and in case of the exponential distribution the ratio of the expected values is exactly the Cordel frequency. It holds

$$Q_{u,\min} = CF_{u,\min}, \quad Q_{u,\max} = CF_{u,\max},$$
$$\textit{and} \quad Q_{e_\lambda,\min} = CF_{e,\min}, \quad Q_{e_\lambda,\max} = CF_{e,\max}.$$

But for the increasing distribution $Q_{e_\lambda,\min} \neq CF_{e_\lambda,\min}$ *and* $Q_{e_\lambda,\max} \neq CF_{e_\lambda,\max}$ *hold.*

With this interesting observation we finish this section on the Cordel frequency for optimization problems with density f. The next section deals with the Cordel frequency for random Σ-type problems.

5.2 Best Solutions Rule for Random General Σ-Type Problems

Since we considered many general Σ-type problems (see examples in Appendix H starting on page 205) we want to present exactly computed Cordel frequencies for "random general Σ-type problems ", as well. These problems are defined as follows.

Definition 5.2.1 (Random General Σ-Type Problem)
Given n random independent coefficients $c_1, \ldots, c_n > 0$ for some $n \geq 1$ we call the discrete optimization problem

$$\min_{x_1,\ldots,x_n \in \mathbb{N}_0} \sum_{i=1}^{n} c_i x_i \tag{5.6}$$

a ***random general Σ-type problem****.*

From now on we make the assumption: $n \geq 3$.

Since all coefficients c_i are positive, the optimal solution to (5.6) is $x^{(0)} = (0, \ldots, 0)$ with $f_0 := f\left(x^{(0)}\right) = 0$. The second-best solution $x^{(1)}$ is then given by

$$x_i^{(1)} = \begin{cases} 1, & \text{for } i = i^* \\ 0, & \text{for } i \neq i^* \end{cases}$$

where i^* is an index with $c_{i^*} = \min\{c_1, \ldots, c_n\}$. Again we use the notation from order statistics (cf. [ABN 1992]). Thus, with $0 < c_{1:n} \leq c_{2:n} \leq \ldots \leq c_{n:n}$ the second-best solution of the random Σ-type problem (5.6) is given by $x^{(1)} = (1, 0, \ldots, 0)$ with $f_1 := f\left(x^{(1)}\right) = c_{1:n}$. But for the third-best solution $x^{(2)}$ there are two possibilities: either $x^{(2)} = (2, 0, \ldots, 0)$ with functional value $2c_{1:n}$ or $x^{(2)} = (0, 1, 0, \ldots, 0)$ with functional value $c_{2:n}$. Hence, $f_2 := f\left(x^{(2)}\right) = \min\{2c_{1:n}, c_{2:n}\}$ holds.

In summary,

$$f_0 = 0\,, \qquad f_1 = c_{1:n}\,, \qquad \text{and} \qquad f_2 = \min\{2c_{1:n}, c_{2:n}\} \tag{5.7}$$

hold.

5.2.1 Cordel Frequencies for Different Distributions

With (5.7) one simply proves the following corollary.

Corollary 5.2.2
Given a random Σ-type problem with the three best functional values $f_0 \leq f_1 \leq f_2$,

$$d_1 := |f_0 - f_1| \geq |f_1 - f_2| =: d_2$$

is always fulfilled.

Thus, the Cordel Frequency for random Σ-type problems with n randomly generated independent coefficients $c_1, \ldots, c_n$ and the best solutions rule is

$$CF(n) := P(d_1 \geq d_2) = 1$$

independently from n and the distribution of $C_1, \ldots, C_n$.

Proof. After the preliminary considerations we easily check:

$$\begin{aligned}\mathrm{CF}(n) :&= P(|f_0 - f_1| \geq |f_1 - f_2|) = P(f_1 - f_0 \geq f_2 - f_1) \\ &= P\Big(C_{1:n} - 0 \geq \min\{2C_{1:n}, C_{2:n}\} - C_{1:n}\Big) \\ &= P\Big(2C_{1:n} \geq \min\{2C_{1:n}, C_{2:n}\}\Big) = 1\,. \end{aligned}$$ ■

Now we go a step further and investigate how much influence the equality $d_1 = d_2$ has on the Cordel Frequency. Or to be more precise, we want to compute the probability that $d_1 > d_2$ holds. Obviously, the following equation is true.

$$P(d_1 > d_2) = P\Big(2C_{1:n} > \min\{2C_{1:n}, C_{2:n}\}\Big) = P(2C_{1:n} > C_{2:n})$$

Thus, we can compute the probability by evaluating the following multiple integral.

$$P(d_1 > d_2) = P(2C_{1:n} > C_{2:n}) = \int_0^\infty \int_{c_1}^{2c_1} f_{1,2:n}(c_1, c_2)\,\mathrm{d}c_2\,\mathrm{d}c_1$$

with

$$f_{1,2:n}(c_1, c_2) = n(n-1)(1 - F(c_2))^{n-2} f(c_1) f(c_2)\ .$$

As in the previous section, choosing the appropriate order of integration simplifies the multiple integral.

$$\begin{aligned}P(d_1 > d_2) &= n(n-1)\int_0^\infty (1 - F(c_2))^{n-2} f(c_2) \int_{\frac{c_2}{2}}^{c_2} f(c_1)\,\mathrm{d}c_1\,\mathrm{d}c_2 \\ &= n(n-1)\int_0^\infty (1 - F(c_2))^{n-2} f(c_2)\left(F(c_2) - F\left(\frac{c_2}{2}\right)\right)\mathrm{d}c_2\end{aligned}$$

On the following pages we present the results for some distributions. In Appendix M.2 starting on page 250 the corresponding Maple worksheets [Maple 2008] with all results can be found.

Uniform Distribution on $[0, 1]$

$$\begin{aligned}P_u(d_1 > d_2) &= n(n-1)\int_0^1 (1 - U(c_2))^{n-2} u(c_2)\left(U(c_2) - U\left(\frac{c_2}{2}\right)\right)\mathrm{d}c_2 \\ &= n(n-1)\int_0^1 (1 - c_2)^{n-2}\left(c_2 - \frac{c_2}{2}\right)\mathrm{d}c_2\end{aligned}$$

$$= \frac{n(n-1)}{2} \int_0^1 c_2 \left(1-c_2\right)^{n-2} \, \mathrm{d}c_2$$

$$= \frac{n(n-1)}{2} \left[\frac{-\left(c_2(n-1)+1\right)\left(1-c_2\right)^{n-1}}{n(n-1)} \right]_{c_2=0}^{1} = \frac{1}{2}$$

Increasing Distribution on $[0, 1]$

$$P_i\left(d_1 > d_2\right) = n(n-1) \int_0^\infty \left(1 - I\left(c_2\right)\right)^{n-2} i\left(c_2\right) \left(I\left(c_2\right) - I\left(\frac{c_2}{2}\right)\right) \mathrm{d}c_2$$

$$= \frac{3n(n-1)}{2} \int_0^1 c_2^3 \left(1-c_2^2\right)^{n-2} \, \mathrm{d}c_2$$

$$= \frac{3n(n-1)}{2} \left[\frac{-\left(1+c_2^2 n - c_2^2\right)\left(1-c_2^2\right)^{n-1}}{2n(n-1)} \right]_{c_2=0}^{1} = \frac{3}{4}$$

Decreasing Distribution on $[0, 1]$

$$P_d\left(d_1 > d_2\right) = n(n-1) \int_0^1 \left(1 - D\left(c_2\right)\right)^{n-2} d\left(c_2\right) \left(D\left(c_2\right) - D\left(\frac{c_2}{2}\right)\right) \mathrm{d}c_2$$

$$= n(n-1) \int_0^1 \left(c_2^2 - 2c_2 + 1\right)^{n-2} \left(2-2c_2\right) \left(2c_2 - c_2^2 - c_2 + \frac{c_2^2}{4}\right) \mathrm{d}c_2$$

$$= 2n(n-1) \int_0^1 \left(1-c_2\right)^{2n-3} \left(c_2 - \frac{3}{4}c_2^2\right) \mathrm{d}c_2$$

$$= 2n(n-1) \left(\int_0^1 c_2 \left(1-c_2\right)^{2n-3} \, \mathrm{d}c_2 - \frac{3}{4} \int_0^1 c_2^2 \left(1-c_2\right)^{2n-3} \, \mathrm{d}c_2 \right)$$

$$\stackrel{\text{PI}}{=} n \left(\int_0^1 \left(1-c_2\right)^{2n-2} \, \mathrm{d}c_2 - \frac{3}{2} \int_0^1 c_2 \left(1-c_2\right)^{2n-2} \, \mathrm{d}c_2 \right)$$

$$\stackrel{\text{PI}}{=} n \left(\left[\frac{-1}{2n-1} \left(1-c_2\right)^{2n-1} \right]_{c_2=0}^{1} - \frac{3}{2\left(2n-1\right)} \int_0^1 \left(1-c_2\right)^{2n-1} \, \mathrm{d}c_2 \right)$$

$$= \frac{n}{2n-1} + \frac{3n}{2\left(2n-1\right)} \left[\frac{1}{2n} \left(1-c_2\right)^{2n} \right]_{c_2=0}^{1}$$

$$= \frac{n}{2n-1} - \frac{3}{4\left(2n-1\right)} = \frac{4n-3}{8n-4} \xrightarrow{n\to\infty} \frac{1}{2}$$

The calculation steps marked with PI are the results of integration by parts. The probability converges strictly monotonic increasing in n to $\frac{1}{2}$.

Centered Triangular Distribution on $[0,1]$
Splitting up the multi integral leads to

$$\begin{aligned} P_t\left(d_1 > d_2\right) &= n(n-1)\int_0^1 \left(1-T\left(c_2\right)\right)^{n-2} t\left(c_2\right)\left(T\left(c_2\right)-T\left(\frac{c_2}{2}\right)\right)\,\mathrm{d}c_2 \\ &= n(n-1)\left(\mathrm{Int}_1+\mathrm{Int}_2\right) \end{aligned}$$

with

$$\begin{aligned} \mathrm{Int}_1 :&= \int_0^{\frac{1}{2}} \left(1-2c_2^2\right)^{n-2}\cdot 4c_2\cdot\left(2c_2^2-\frac{c_2^2}{2}\right)\mathrm{d}c_2 \\ &= \frac{3}{4n(n-1)} - \frac{3(n+1)}{2^{n+2}\cdot n(n-1)} \\ \text{and} \qquad \mathrm{Int}_2 :&= \int_{\frac{1}{2}}^1 \left(2-4c_2+2c_2^2\right)^{n-2}\cdot(4-4c_2)\cdot\left(-2c_2^2+4c_2-1-\frac{c_2^2}{2}\right)\mathrm{d}c_2 \\ &= \frac{6n^2+3n-5}{2^{n+2}\cdot n(n-1)(2n-1)}\,. \end{aligned}$$

Thus,

$$P_t\left(d_1>d_2\right) = n(n-1)\left(\mathrm{Int}_1+\mathrm{Int}_2\right) = \frac{3}{4} - \frac{1}{2^{n+1}\left(2n-1\right)} \xrightarrow{n\to\infty} \frac{3}{4}$$

holds, which converges strictly monotonic increasing to $\frac{3}{4}$.

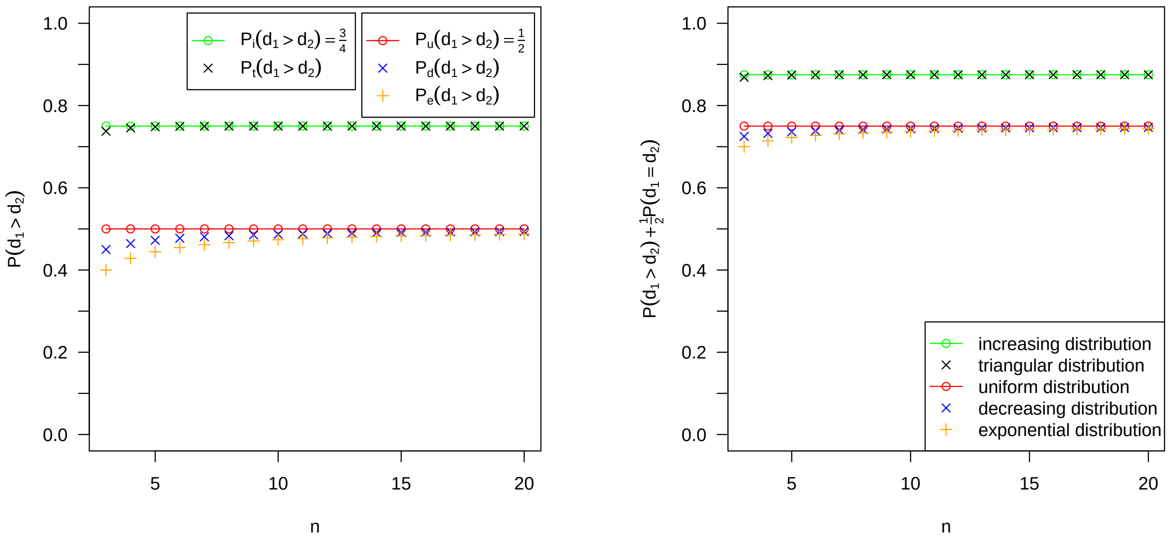
Exponential Distribution with Parameter $\boldsymbol{\lambda} > 0$

$$\begin{aligned} P_{e_\lambda}\left(d_1>d_2\right) &= n(n-1)\int_0^\infty \left(1-E_\lambda\left(c_2\right)\right)^{n-2} e_\lambda\left(c_2\right)\left(E_\lambda\left(c_2\right)-E_\lambda\left(\frac{c_2}{2}\right)\right)\mathrm{d}c_2 \\ &= n(n-1)\int_0^\infty \mathrm{e}^{-\lambda c_2(n-2)}\cdot\lambda\mathrm{e}^{-\lambda c_2}\left(-\mathrm{e}^{-\lambda c_2}+\mathrm{e}^{-\frac{1}{2}\lambda c_2}\right)\mathrm{d}c_2 \\ &= n(n-1)\left(\int_0^\infty \lambda\mathrm{e}^{-\lambda c_2\left(n-\frac{1}{2}\right)}\,\mathrm{d}c_2 - \int_0^\infty \lambda\mathrm{e}^{-\lambda c_2 n}\,\mathrm{d}c_2\right) \\ &= n(n-1)\left(\left[\frac{-1}{n-\frac{1}{2}}\mathrm{e}^{-\lambda c_2\left(n-\frac{1}{2}\right)}\right]_{c_2=0}^\infty - \left[\frac{-1}{n}\mathrm{e}^{-\lambda c_2 n}\right]_{c_2=0}^\infty\right) \\ &= n(n-1)\left(\frac{1}{n-\frac{1}{2}}-\frac{1}{n}\right) = \frac{n-1}{2n-1} \xrightarrow{n\to\infty} \frac{1}{2} \end{aligned}$$

Again, the probability converges strictly monotonic in n to $\frac{1}{2}$.

Summary of all Results

The following Figure 5.2.1 shows all established probabilities.



(a) $P(d_1 > d_2)$ **(b)** $P(d_1 > d_2) + \frac{1}{2} P(d_1 = d_2)$

Figure 5.2.1: $P(d_1 > d_2)$ sorted form high to low: increasing i (green), triangular t (black), uniform u (red), decreasing d (blue) and exponential e_λ (orange) distribution.

5.2.2 Expected Values $\mathbb{E}(d_1)$ and $\mathbb{E}(d_2)$

With

$$d_1 = c_{1:n} \quad \text{and} \quad d_2 = \min\{2c_{1:n}, c_{2:n}\} = \begin{cases} c_{1:n}, & \text{for } 2c_{1:n} \le c_{2:n} \\ c_{2:n} - c_{1:n}, & \text{for } 2c_{1:n} \ge c_{2:n} \end{cases}$$

follows

$$\mathbb{E}(d_1) = \int_0^\infty c_1 f_{1:n}(c_1)\, \mathrm{d}c_1 = n \int_0^\infty c_1 (1 - F(c_1))^{n-1} f(c_1)\, \mathrm{d}c_1 \tag{5.8}$$

and

$$\begin{aligned} \mathbb{E}(d_2) &= \int_0^\infty \left(\int_0^{\frac{c_2}{2}} c_1 f_{1,2:n}(c_1, c_2)\, \mathrm{d}c_1 + \int_{\frac{c_2}{2}}^\infty (c_2 - c_1) f_{1,2:n}(c_1, c_2)\, \mathrm{d}c_1 \right) \mathrm{d}c_2 \\ &= n(n-1) \int_0^\infty (1 - F(c_2))^{n-2} f(c_2) \left(\int_0^{\frac{c_2}{2}} c_1 f(c_1)\, \mathrm{d}c_1 \right. \\ &\qquad \left. + \int_{\frac{c_2}{2}}^\infty (c_2 - c_1) f(c_1)\, \mathrm{d}c_1 \right) \mathrm{d}c_2 \end{aligned} \tag{5.9}$$

Maple [Maple 2008] computed the following expected values (cf. Appendix M.2 starting on page 250).

$$\mathbb{E}_u(d_1) = \frac{1}{n+1} \qquad \mathbb{E}_u(d_2) = \frac{1}{4(n+1)}$$

$$\mathbb{E}_i(d_1) = \frac{\sqrt{\pi}}{2}\frac{n!}{\Gamma\left(n+\frac{3}{2}\right)} \qquad \mathbb{E}_i(d_2) = \frac{\sqrt{\pi}}{16}\frac{n!}{\Gamma\left(n+\frac{3}{2}\right)}$$

$$\mathbb{E}_d(d_1) = \frac{1}{2n+1} \qquad \mathbb{E}_d(d_2) = \frac{n}{8n^2-2}$$

$$\mathbb{E}_{e_\lambda}(d_1) = \frac{1}{n\lambda} \qquad \mathbb{E}_{e_\lambda}(d_2) = \frac{n}{(4n^2-4n+1)\,\lambda}$$

Therewith the following quotients $Q_f := \frac{\mathbb{E}_f(d_1)}{\mathbb{E}_f(d_1)+\mathbb{E}_f(d_2)}$ occurred.

$$Q_u = \frac{4}{5}$$

$$Q_i = \frac{8}{9}$$

$$Q_d = \frac{4n-2}{5n-2} \xrightarrow{n\to\infty} \frac{4}{5}$$

$$Q_{e_\lambda} = \frac{4n^2-4n+1}{5n^2-4n+1} \xrightarrow{n\to\infty} \frac{4}{5}$$

Observe, that $Q_f > \frac{1}{2}$ holds for every $n \geq 3$ and for every considered distribution. The following Figure 5.2.2 shows all quotients for comparison.

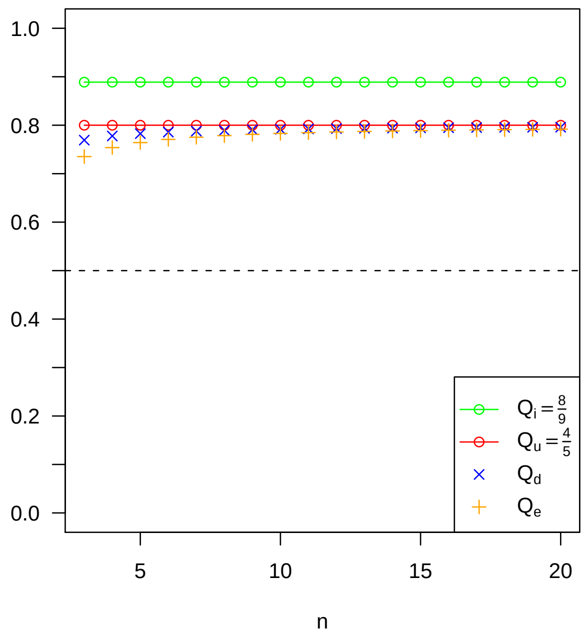


Figure 5.2.2: Quotients $Q_f := \frac{\mathbb{E}_f(d_1)}{\mathbb{E}_f(d_1)+\mathbb{E}_f(d_2)}$ sorted form high to low: increasing i (green), uniform u (red), decreasing d (blue) and exponential e_λ (orange) distribution.

5.3 Penalty Selection Rule

5.3.1 Uniform Distribution Estimate

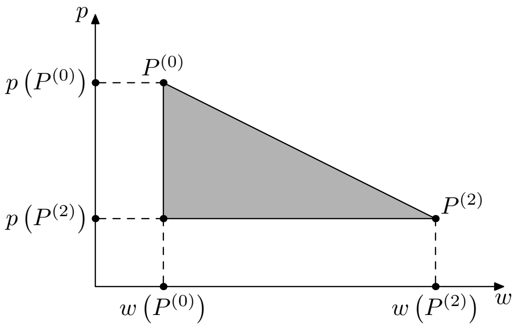
In this section we present an estimate by Althöfer, which gives a good hint in which magnitude Cordel frequencies for the penalty selection rule are to be expected.

In Lemma 2.3.3 on page 35 we determined that penalty alternatives are supported points in the (w, p) diagram. Thus $\left(w\left(P^{(1)}\right), p\left(P^{(1)}\right)\right)$ lies inside of the triangle with vertices

$$\left(w\left(P^{(0)}\right), p\left(P^{(0)}\right)\right), \quad \left(w\left(P^{(2)}\right), p\left(P^{(2)}\right)\right), \quad \text{and} \quad \left(w\left(P^{(0)}\right), p\left(P^{(2)}\right)\right).$$

We call this triangle, which is shown in the figure on the right, the **triangle of feasible points for $\boldsymbol{P^{(1)}}$**. To simplify notation, we use $P^{(1)}$ as name for the pair $\left(w\left(P^{(1)}\right), p\left(P^{(1)}\right)\right)$, too, and say that $P^{(1)}$ lies inside of the triangle

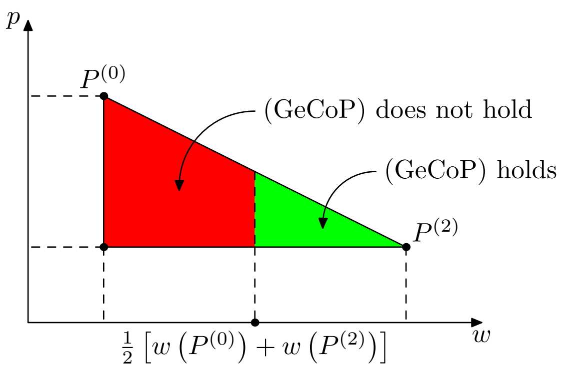
$$\triangle P^{(0)}, P^{(2)}, \left(w\left(P^{(0)}\right), p\left(P^{(2)}\right)\right).$$



This triangle spanned by $P^{(0)}$, $P^{(2)}$, and $\left(w\left(P^{(0)}\right), p\left(P^{(2)}\right)\right)$ can be dissected into two parts: one part where (GeCoP) holds and one part where (GeCoP) does not hold. Both parts are easy to identify, since (GeCoP) holds, whenever

$$\begin{aligned} & d_1 := w\left(P^{(1)}\right) - w\left(P^{(0)}\right) \geq w\left(P^{(2)}\right) - w\left(P^{(1)}\right) =: d_2 \\ \Leftrightarrow \quad & w\left(P^{(1)}\right) \geq \frac{1}{2}\left[w\left(P^{(0)}\right) + w\left(P^{(2)}\right)\right] \end{aligned}$$

is fulfilled.



Hence, if $P^{(1)}$ is uniformly distributed in this triangle, then the Cordel frequency is

$$P\left(d_1 \geq d_2\right) = \frac{1}{4}.$$

Likewise, $P(d_i \geq d_{i+1}) = \frac{1}{4}$ holds for every i, for uniformly distributed penalty alternatives $P^{(i)}$ within the triangle of feasible points spanned by $P^{(i-1)}, P^{(i+1)}$, and $\left(w\left(P^{(i-1)}\right), p\left(P^{(i+1)}\right)\right)$. These 25% are indeed the magnitude of our experimentally observed Cordel frequencies for penalty alternatives.

But in fact it cannot be expected that $P^{(1)}$ is equally distributed in the feasible triangle. In practice one probably has more than one feasible solution within the triangle, but only one supported point namely $P^{(1)}$. Hence, $P^{(1)}$ will not be equally distributed. In the next subsection we try to estimate the real distribution of $P^{(1)}$ inside of the feasible triangle.

5.3.2 Analysis of the Actual Distribution of $P^{(1)}$ Inside of the Feasible Triangle

Apart from concrete optimization problems we want to determine the distribution of $P^{(1)}$ inside of the feasible triangle. Therefore we observe a random instance represented by n random points in $\mathbb{R}^2$. Thus, each random point has a random weight and a random penalized part which are independently of another.

Since the penalty alternatives are more or less equivalent to the supported points (cf. Lemma 2.3.3 on page 35), we studied the distribution of the second-best support point (when sorting the supported points by weight). As for the penalty method we call the best supported point (best regarding the weight) $P^{(0)}$, the second-best $P^{(1)}$ and the third-best supported point $P^{(2)}$. In order to compare the results for different random instances, the weights and penalized parts were scaled and shifted such that

$$w\left(P^{(0)}\right) = 0,\ w\left(P^{(2)}\right) = 1, \quad \text{and} \quad p\left(P^{(0)}\right) = 1,\ p\left(P^{(2)}\right) = 0$$

hold. This is done by the following transformations

$$\overline{w}(B) := \frac{w(B) - w\left(P^{(0)}\right)}{w\left(P^{(2)}\right) - w\left(P^{(0)}\right)} \geq 0 \tag{5.10}$$

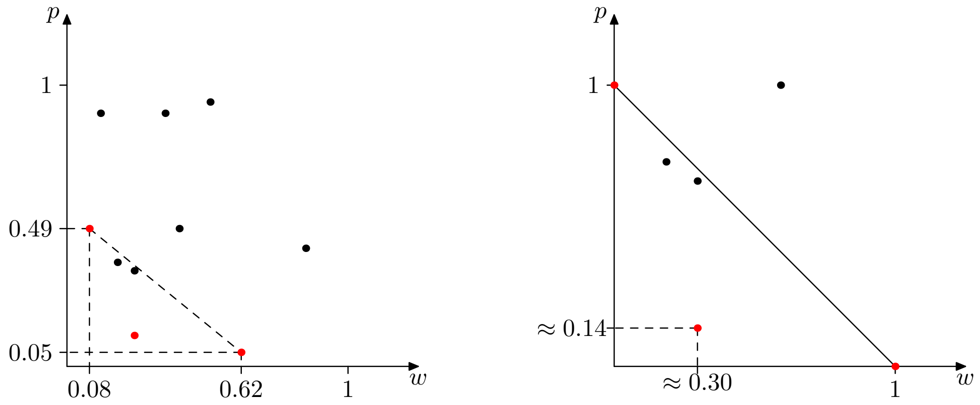
and

$$\overline{p}(B) := \frac{p(B) - p\left(P^{(2)}\right)}{p\left(P^{(0)}\right) - p\left(P^{(2)}\right)} \tag{5.11}$$

for all feasible solution $B \in S$.

Example 5.3.1
Consider the following instance with 10 *random points in the unit square shown in Figure 5.3.1 (a). In this example the supported points, which represent the penalty alternatives, have the following weights and penalized parts:*

$$\begin{array}{ll} w\left(P^{(0)}\right) = 0.08 & p\left(P^{(0)}\right) = 0.49 \\ w\left(P^{(1)}\right) = 0.24 & p\left(P^{(1)}\right) = 0.11 \\ w\left(P^{(2)}\right) = 0.62 & p\left(P^{(2)}\right) = 0.05 \end{array}$$



(a) Original instance with three supported points marked in red.

(b) Instance after the scaling and shifting.

Figure 5.3.1: Illustration of the scaling and shifting done by equations (5.10) and (5.11).

Applying transformations (5.10) *and* (5.11) *it follows:*

$$\overline{w}\left(P^{(0)}\right) = \frac{0.08 - 0.08}{0.62 - 0.08} = 0 \qquad \overline{p}\left(P^{(0)}\right) = \frac{0.49 - 0.05}{0.49 - 0.05} = 1$$

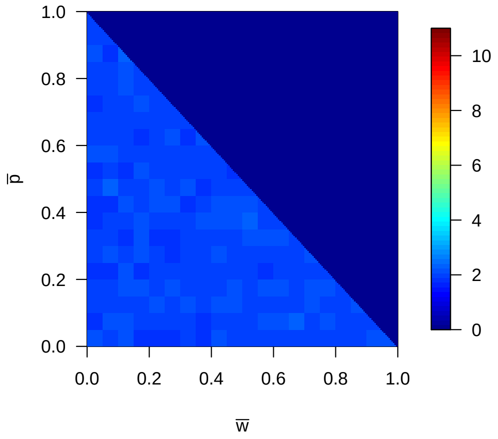
$$\overline{w}\left(P^{(1)}\right) = \frac{0.24 - 0.08}{0.62 - 0.08} = \frac{8}{27} \approx 0.30 \qquad \overline{p}\left(P^{(1)}\right) = \frac{0.11 - 0.05}{0.49 - 0.05} = \frac{3}{22} \approx 0.14$$

$$\overline{w}\left(P^{(2)}\right) = \frac{0.62 - 0.08}{0.62 - 0.08} = 1 \qquad \overline{p}\left(P^{(2)}\right) = \frac{0.05 - 0.05}{0.49 - 0.05} = 0$$

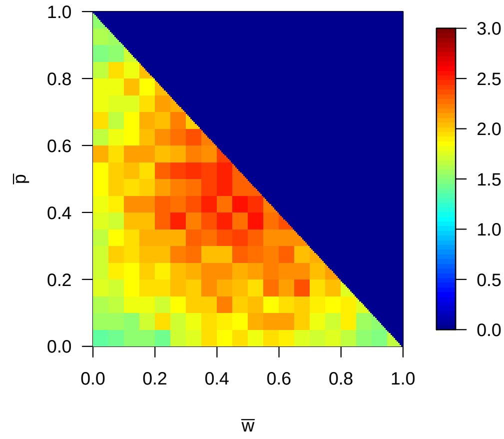
These transformed coordinates are shown in Figure 5.3.1 (b). Observe that only the lower left corner is shown. The transformed coordinates of the remaining four points are too big to appear in this section.

In this way we can analyze the distribution of $\overline{w}\left(P^{(1)}\right)$ and $\overline{p}\left(P^{(1)}\right)$ for random instances with n points in $\mathbb{R}^2$ and at least three supported points. These n random points are independently distributed with density $f : \mathbb{R}^2 \to \mathbb{R}_{\geq 0}$, which has to be predefined. We considered two distributions f. The uniform distribution in the unit square $[0,1]^2$ and the standard normal distribution in $\mathbb{R}^2$.
Figure 5.3.2 on the previous page shows the experimentally observed histograms for the uniform distribution (figures on the left) and for the normal distribution (figures on the right). For each $n \in \{3, 10, 50\}$ we considered $1,000,000$ random instances.

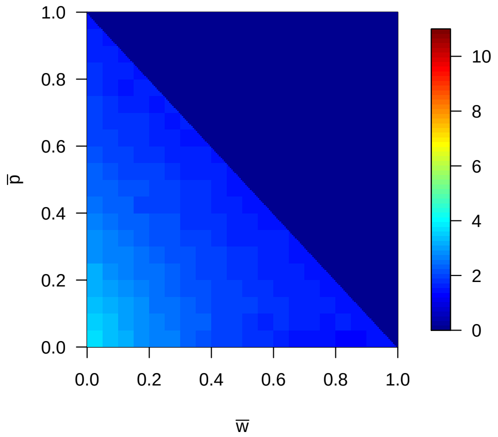
For random instances with $n = 3$ uniformly distributed points in the unit square and at least three supported points (cf. Figure 5.3.2 (a)), $P^{(1)}$ has to be uniformly distributed within the feasible triangle. But for instances with more than three random points ($n > 3$), $P^{(1)}$ is no longer uniformly distributed. Figures 5.3.2 (c) and (e) show, that $P^{(1)}$ is moving further and further away from the diagonal hypotenuse of the feasible triangle. It seems that $\mathbb{E}\left(\overline{w}\left(P^{(1)}\right), \overline{p}\left(P^{(1)}\right)\right)$ converges to $(0,0)$ or at least to a point near $(0,0)$ for $n \to \infty$. Note that the transformations (5.10) and (5.11) were applied in order to get these scaled histograms.



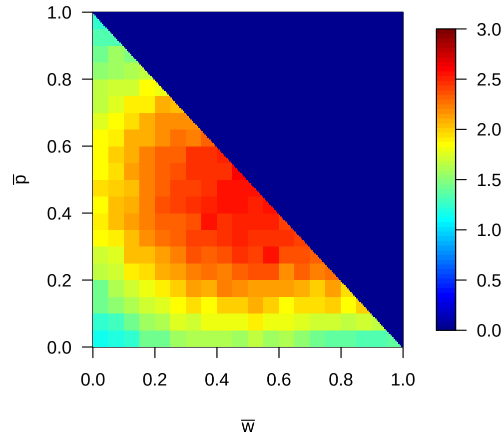
(a) $n = 3$, uniform distribution



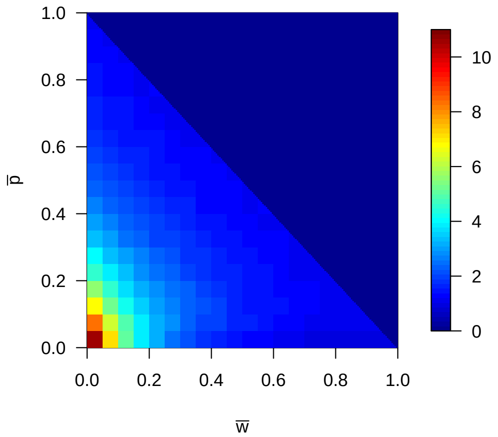
(b) $n = 3$, normal distribution



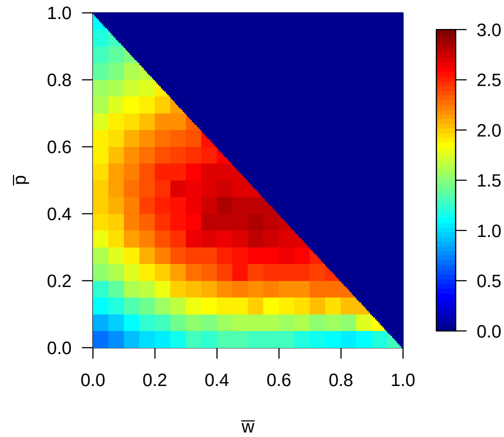
(c) $n = 10$, uniform distribution



(d) $n = 10$, normal distribution



(e) $n = 50$, uniform distribution



(f) $n = 50$, normal distribution

Figure 5.3.2: Experimentally established histogram of the joint distribution of $\overline{w}\left(P^{(1)}\right)$ and $\overline{p}\left(P^{(1)}\right)$ for n **uniformly** distributed points in the unit square (left) and for n **normally** distributed points in $\mathbb{R}^2$ (right).

For normally distributed points a different pictures emerges as shown by Figures 5.3.2 (b), (d), and (f). Here the distribution spreads more widely. Furthermore, $P^{(1)}$ is tending more to the middle of the hypotenuse and not to the edge $(0, 0)$.

These different distributions of $P^{(1)}$ inside of the feasible triangle for uniformly and normally distributed points, implicate different Cordel frequencies for both distributions. Figure 5.3.3 shows the experimentally observed frequencies.

While the normal distributions provides frequencies which are very close to 25%, the Cordel frequency for uniformly distributed points seems to converge to approximately 14% for $n \to \infty$. These experimentally observed frequencies are presented in the following Figure 5.3.3.

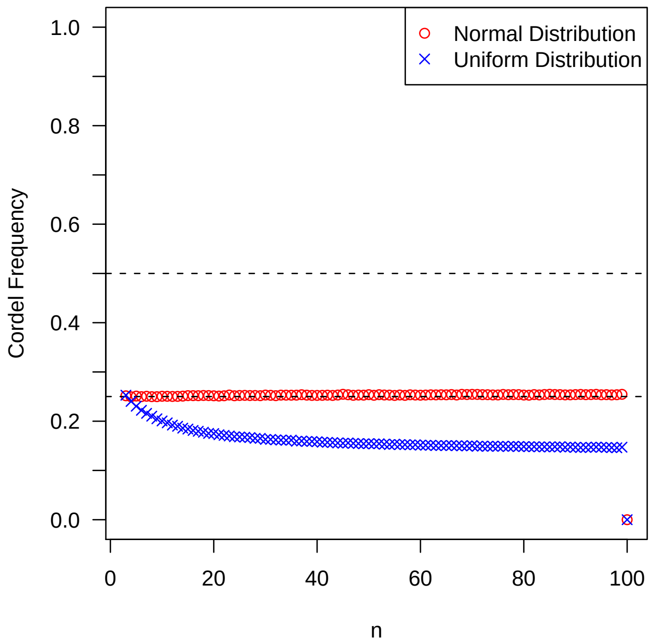


Figure 5.3.3: Experimentally established Cordel frequencies for n normally distributed points in $\mathbb{R}^2$ (red) or n uniformly distributed points in the unit square (blue).

These Cordel frequencies have the same order of magnitude as the experimentally observed Cordel frequencies for random optimization problems presented in Chapter 3 (starting on page 67) and Appendix J (starting on page 211). Hence, the results from Chapter 3 may be explained with this model and are therefore not surprising.

Chapter 6

Open Problems and Conclusions

6.1 Open Problems and Future Work

Chapter 1: The Cordel Property

1. We first introduced the Cordel frequency for chess. In this context we suggested three selection rules – the best moves rule (probably the most evident selection rule) and the best move per piece as well as the best move per piece type rule (cf. Definition 1.2.12 on page 8). It could be interesting to think of further selection rules and to compare the Cordel frequencies they yield with the Cordel frequencies presented in this doctoral dissertation.

2. In Definition 1.3.3 on page 12 we gave our definition for the Cordel frequency for chess. While we considered for WWW- and LLL-instances the generalized Cordel property

$$d_1 := |f(m_1) - f(m_2)| \quad \geq \quad |f(m_2) - f(m_3)| =: d_2\,, \qquad \text{(GeCoP)}$$

   which takes the values $f(m_1), f(m_2)$, and $f(m_3)$ into account, the remaining chess positions were divided into Cordel and non-Cordel positions without regards to concrete values (cf. table on page 12).

   In contrast to this one could also consider the frequencies for the following approaches:

   a) One could think of a definition of the Cordel frequency where concrete values of moves (distance to mate) are never considered. In this case only the game theoretical values (W, D, or L) are regarded. This leads to the following division.

Cordel positions	non-Cordel positions
W, D, L	-
WD, WL, DL	WW, DD, LL
WWW, WDD, WDL, WLL, DDD, DLL, LLL	WWD, WWL, DDL

   This approach increases the Cordel frequencies under the best moves rule to somewhere between a stunning 94% and 97% (for $k \in \{3, 4, 5\}$ pieces).

b) Remember that we call two W-moves (or L-moves) m_1 and m_2 equally good, iff $f(m_1) = f(m_2)$ holds. This definition suggests the consideration of the following Cordel- and non-Cordel division where m_1, m_2, and m_3 denote the three moves chosen by the selection rule.

Cordel positions	non-Cordel positions
W, D, L	-
WD, WL, DL	DD
WW and LL with $f(m_1) \neq f(m_2)$	WW and LL with $f(m_1) = f(m_2)$
WWW and LLL with $f(m_1) \neq f(m_2)$ or $f(m_1) = f(m_2) = f(m_3)$	WWW and LLL with $f(m_1) = f(m_2) \neq f(m_3)$
WWD and WWL with $f(m_1) \neq f(m_2)$	WWD and WWL with $f(m_1) = f(m_2)$
WDD, WDL, WLL, DDD, DLL,	DDL

A more detailed analysis of Cordel frequencies for these two approaches might be interesting. One could find more interpretations of Cordel's Three Moves Law.

3. The extensive tables in Appendices D (pp. 139-154), E (pp. 159-174), and F (pp. 179-194) provide a very detailed overview of the Cordel frequencies for each piece distribution with $k \in \{3, 4, 5\}$ pieces. An interesting question is whether chess masters can draw further conclusions from these datasets.

4. On page 17 at the end of Subsection 1.3 we speculate about the Cordel frequencies for chess positions with more then six pieces, $k > 6$. It remains an open question whether all Cordel frequencies $\mathrm{CF}(k), k \leq 32$, are greater than 50% under the best moves rule actually. The same question arises for the best move per piece and for the best move per piece type rule.

Chapter 2: The Penalty Method for General Sum-Type Problems

5. With Definition 2.2.2 on page 23 we gave a generalized definition of Schwarz's penalty method [Sch 2003, pp. 7-8] which works for general Σ-type problems (instead of just Σ-type problems which Schwarz considered). Besides, Schwarz [Sch 2003, p. 15] also introduced the "Linear Programming Penalty Method" which later on became called the "Mutual Penalty Method". It remains an open question what a general mutual penalty method for general Σ-type problems would look like.

6. In Definition 2.2.7 on page 28 we presented a canonical penalty vector for general Σ-type problems where all non-zero weights $w_i \neq 0, i \in \{1, \ldots, n\}$ have the same sign (all positive or all negative). Though this restriction was satisfied in all the optimization problems we considered, it might be good to have a definition of a canonical penalty vector in mind without sign restrictions on the weights.

7. Definition 2.3.6 on page 37 gave a definition of the k best penalty alternatives which explicitly excluded degenerate penalty alternatives (alternatives which are optimal only for a single $\varepsilon > 0$). As we noted in Remark 2.4.1 this exclusion

of degenerate penalty alternatives is especially important when computing the k best penalty alternatives, since one needs an algorithm that computes **all** optimal solutions to a punished problem

$$\min_{B \in S} w(B) + \varepsilon \cdot p(B)$$

in order to compute all the degenerate penalty alternatives, too. But since most common optimization algorithms compute only **one** solution to a given optimization problem this would cause problems. It is an open question whether this exclusion of degenerate penalty alternatives affects the Cordel frequency or not.

Chapter 3: Experimentally Observed Cordel Frequencies under the Penalty Selection Rule

8. Which further optimization problems are interesting to study and which Cordel frequencies do they yield?

9. In Subsection 3.3.2 (pp. 75-76) the Cordel frequencies for the shortest path problem in road networks with real travel times in all US-States were presented. Since these frequencies varied between 15% and 42% we asked at the end of the subsection if there are simple properties of the given graphs influencing the Cordel frequencies. Such properties could be, for example, the number of vertices and edges of the graph or the maximal or average node degree.

10. While we were able to give an construction scheme to every difference vector $d = (d_1, \ldots, d_k) \in \mathbb{R}^k_{>0}$ and threshold parameter vector $\varepsilon = (\varepsilon_1, \ldots, \varepsilon_k)$ with $0 < \varepsilon_1 < \ldots < \varepsilon_k$ for the shortest path problem in grid graphs, our investigations for the minimum spanning tree problem were only partially successful. For the knapsack problem, where we did not find any construction scheme, we even suggest that there might be difference vectors d for which no knapsack instance exists. Here, further examinations are necessary in order to decide, whether this last conjecture is really true or not. If one finds a construction scheme for each difference vector $d \in \mathbb{R}^k_{>0}$, this conjecture would be disproved. But initially it might be good to have a construction scheme at least for difference vectors d with certain properties (for example $d_1 < d_2 < \cdots < d_k$).

11. On page 85 we presented some suggested limits for the Cordel frequency under the penalty selection rule for the unbounded knapsack problem. Here, further investigations for supporting or refuting these suggestions are necessary.

Chapter 4: The Cordel Frequency under the Best Solutions Rule

12. Since we only considered two types of optimization problems for the best solutions rule, it would be good to analyze more problems. This would help in order to decide if the results for the shortest path problem and for the minimum spanning tree problem are really characteristic, as we believe them to be.

Chapter 5: Theoretical Models

13. In Subsection 5.1.5 (pp. 107-108) we presented the δ-distribution and experimentally observed limits for the Cordel frequency for δ-distributed optimization

problems. We suggested that every value in the open interval $\left(\frac{2}{3}, 1\right)$ occurs as the limit of $\mathrm{CF}_{f_\delta,\max}(n)$ for at least one $\delta > 0$. It remains an open problem to prove this by analytic investigations.

Furthermore we conjectured that

$$\lim_{n\to\infty} \mathrm{CF}_{f,\min} \geq \frac{1}{2} \quad \text{and} \quad \lim_{n\to\infty} \mathrm{CF}_{f,\max} \geq \frac{1}{2}$$

hold. To provide a distribution function f_l for each limit $l \in \left[\frac{1}{2}, 1\right]$ would be sufficient to prove this conjecture.

14. In Section 5.3 (page 116-120) we presented an estimate for the Cordel frequency under the penalty selection rule considering the so-called feasible triangle where $P^{(1)}$ (used as an abbreviation for the pair $\left(w\left(P^{(1)}\right), p\left(P^{(1)}\right)\right)$, too) is to be found. Consideration of different distributions of $P^{(1)}$ inside of this triangle led to estimates between 14% and 25% for the Cordel frequency. It remains an open question what the actual distribution of $P^{(1)}$ inside of this feasible triangle looks like for concrete types of optimization problems.

6.2 Conclusion

Oskar Cordel's Three Moves Law for chess from 1913 provided the basis for this doctoral dissertation. Up to now, there have been no investigations on this law as far as we know. Because of that our aim was to lay the foundation for further investigations on this law and to gain a first impression whether a similar law can be applied for discrete optimization problems, too.

We started our investigations with the introduction of the **generalized Cordel property** (GeCoP) for chess (cf. Definition 1.2.8, p. 7), which is a mathematical generalization of Cordel's Three Moves Law. In order to investigate how many chess positions fulfill the generalized Cordel property, the **Cordel frequency** was introduced. A Cordel frequency of 100% implies that our interpretation of the Three Moves Law is valid in every chess position and a Cordel frequency of 0% means that it is never valid. With the help of Bleicher [Ble 2005] we were able to examine **all** chess endgames with at most 5 pieces (including kings). It has been shown that the Cordel frequency for chess endgames is between 75% and 85% and therefore considerably greater than 50%. As a first very important result we conclude that Oskar Cordel's Three Moves Law is a good rule of thumb, at least in chess endgames even though the magnificent 100% is not reached at all.
But we did not only determine how often (GeCoP) holds for the three best moves. By the introduction of selection rules we gave a very general definition of the Cordel frequency which allowed us, for example, to even compute the Cordel frequency for the three best moves of distinct chess pieces, for example. This doctoral dissertation provides very detailed results for every piece distribution (for example two kings and one rook) which might give chess masters the possibility of drawing further conclusions

for endgame theory.

The results for chess, which provided a Cordel frequency considerably greater than 50%, confirmed us that there is really something like a Three Moves Law in chess. This has led us to go a step further and to introduce the Cordel frequency for optimization problems. In analogy to chess, where not only the three best moves but also generalizations have been considered, we proposed two selection rules for optimization problems:

- the obvious best solutions rule which actually chooses the three best feasible solutions
- and the penalty selection rule which chooses the optimal solution and two good alternative solutions which should sufficiently differ from the optimal solution.

The penalty selection rule was well-studied already for a specific type of optimization problems. However, we were able to present a generalized penalty method which is applicable to even more optimization problems! It has been shown that still all important properties of the penalty method (as for example the monotonicity) hold. Furthermore, we introduced the new concept of ranking and numbering penalty alternatives which allows us to refer to the k best penalty alternatives.

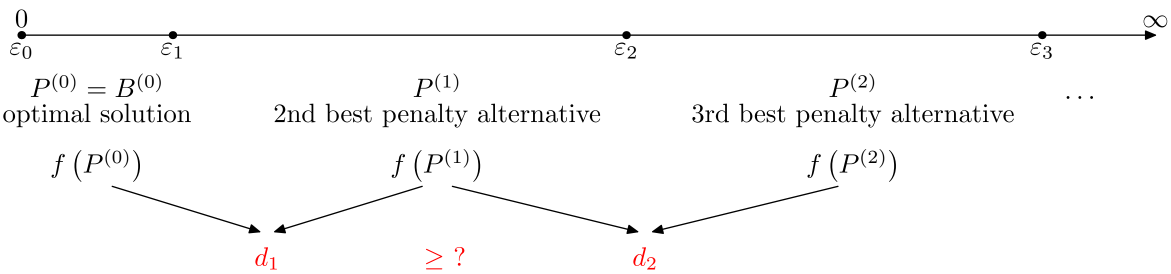


Figure 6.2.1: Schematic representation of the numbering of penalty alternatives. The red criterion shows the generalized Cordel property (GeCoP).

This approach has one decisive advantage: The former concept of the penalty method intended to compute the penalty alternative to a predefined penalty parameter $\varepsilon > 0$. This procedure might lead to penalty alternatives which are

- either equal to the optimal solution (if ε was too small)
- or bad with respect to the objective function (if ε was too large)

and, thus, in both cases useless. By considering the k best penalty alternatives we prevent these troubles caused by this difficult choice of an appropriate penalty parameter ε. The second best penalty alternative will never be equal to the optimal solution and the second best penalty alternative will always have the best functional value of all non degenerate penalty alternatives differing from the optimal solution.

With an adaption of Schwarz's algorithm we were able to give a method for computing these k best penalty alternatives. Additionally, we proposed several approaches for reducing the run time of this basic algorithm. It was found that left-first-traversal is the most powerful tool which can reduce the run time by up to 80% (five times faster) and maybe even more. This allows us to compute the three best penalty alternatives, which are needed for the experimental determination of the Cordel frequency, really fast.

Comprehensive experimental investigations showed, that typical Cordel frequencies under the penalty selection rule are between 15% and 30%. This means that discrete optimization problems under the penalty selection rule are in some measure non-Cordel.

Of course, a very important question is whether it is useful to check the validity of a chess rule of thumb for optimization problems. Anyway, the Cordel frequency provides information on the usefulness of penalty alternatives. As the Cordel frequency under the penalty selection rule was defined as

$$\mathrm{CF} := \Pr\left(d_1 \geq d_2\right),$$

a Cordel frequency which is much smaller than 50% means that mostly $d_1 < d_2$ holds. Consequently, the third best penalty alternative is significantly worse than the second best penalty alternative. Applied to the former penalty method this could mean that one should assure that the penalty parameter ε is not chosen too large, since that entails penalty alternatives whose functional value is usually significantly worse than the functional values of penalty alternatives to smaller penalty parameters ε.

After this extensive analysis of the Cordel frequency under the penalty selection rule, the best solutions rule was not examined as detailed. The problem is that only for some type of optimization problems specific, fast algorithms exist which compute the three best solutions. Instead, for most optimization problems current algorithms allow only the investigation of very few, small instances. Our results suggest that typical Cordel frequencies are about 50%. Hence, again no Cordel property can be shown.

Contrary to what we suggested up to now, indeed the Cordel property occurred for some optimization problems. But we believe that what we call typical optimization problems usually do not show a Cordel behavior. Whenever Cordel frequencies considerably greater that 50% have been observed, a special property of the optimization problem was the reason for that. It might be interesting to study more optimization problems in order to support or refute our beliefs.

We also tried to substantiate our claims by the computation of the Cordel frequency under some theoretical assumptions. This approach was successful in the case of the penalty selection rule. Indeed, the estimate yielded frequencies between 14% and 25%. This is exactly the range of the frequencies that we called typical Cordel frequencies under the penalty selection rule before!

Altogether, it turned out that the investigation of the Cordel frequency is a really interesting field of research where many questions are still open. Especially, there is a lot of scope for interpretations and applications of the results. Here, more work is needed to really find out what the results presented here mean in practice. We are convinced that future research will fill this gap.

6.3 Oskar Cordel's Three Moves Law Might be All Around

We conclude this final chapter with the provocative thesis

"Oskar Cordel's Three Moves Law might be all around – one only has to search for it!"

and some suggestions where else Oskar Cordel's Three Moves Law might be found.

1. Since the Three Moves Law was originally meant for chess, it seems obvious to examine other games.

2. We also analyzed the distance between the results of the three best teams in the german soccer national league "Fußball Bundesliga" in the years 1963-2010. Here the Cordel frequency was 52%. This is in fact a very poor result telling us that the Three Moves Law cannot be transferred to soccer. But what about other games and leagues? Having for example a non-Cordel behavior for sport leagues would mean that there are usually exactly two best teams fighting for the title. Of course it might be exciting to explore this.

3. Althöfer had the funny idea of questioning search engines like Google and Bing for the German phrases "einen Wunsch frei", "zwei Wünsche frei", "drei Wünsche frei". We also searched for their English translations "grant a wish", "grant two wishes", "grant three wishes". The following number of results occured:

	Google	Bing
"einen Wunsch frei"	≈ 3,300,000 results	≈ 36,200 results
"zwei Wünsche frei"	≈ 251,000 results	≈ 3,980 results
"drei Wünsche frei"	≈ 689,000 results	≈ 38,300 results
"grant a wish"	≈ 1,850,000 results	≈ 55,300 results
"grant two wishes"	≈ 25,500 results	≈ 4,980 results
"grant three wishes"	≈ 837,000 results	≈ 13,700 results

   Both search engines provided significantly fewer results for the request with "two" ("zwei") than for one and three. Of course one should think about if this result is really due to the Cordel property.

4. Where else could you imagine to find Cordel's Three Moves Law? Go out and search for it!

Appendix A

Original German Text by H. Ranneforth

Refers to page 1.

Quotation from [Cor 1913]: “An dieser Stelle sei folgendes bemerkt: Auf Grund jahrelanger stellte Cordel folgendes, wie er es nannte: „Dreizügegesetz“, auf. In einer beliebigen Position gibt es entweder nur einen, besten Zwangszug, oder aber drei etwa gleichwertige Züge. Dieses Gesetz, so lehrte er, übe eine vorzügliche Kontrolle aus über die Richtigkeit irgend einer Wendung im Schachspiel. Es habe für ihn namentlich da Klarheit geschaffen, wo in den Lehrbüchern steht, daß die Untersuchungen noch nicht abgeschlossen seien, wie im Evansgambit, Giuoco piano, Englischen Springerspiel u. dgl. mehr. Lagen zwei anscheinend etwa gleichwertige Fortsetzungen vor, so war er davon überzeugt, daß entweder eine falsch, oder noch eine dritte vorhanden sein müsse, und er ließ dann nicht locker, bis Klarheit geschaffen war. Die Vorarbeiten für ein Buch über dies „Dreizügegesetz“ waren bereits weit vorgeschritten. Auch hier hat ihm der Tod die Feder aus der Hand genommen.”

Appendix B

An Example Textfile: King Plus Knight Versus King Plus Pawn

Refers to page 12.

```
Cordel_knkp_wtm.txt

Cordel Codes
WWW: 23 0.000488918 %
WWD: 38 0.000807777 %
WWL: 0 0 %
WDD: 147 0.00312482 %
WDL: 0 0 %
WLL: 0 0 %
DDD: 3598295 76.49 %
DDL: 209730 4.45829 %
DLL: 286701 6.09449 %
LLL: 608288 12.9306 %
WW: 0 0 %
WD: 0 0 %
WL: 0 0 %
DD: 415 0.00882178 %
DL: 361 0.00767389 %
LL: 56 0.00119041 %
W: 0 0 %
D: 110 0.0023383 %
L: 100 0.00212573 %
CM: 2 4.25146e-05 %
SM: 0 0 %
Number of moves
0 moves: 2 4.25146e-05 %
1 moves: 210 0.00446403 %
2 moves: 832 0.0176861 %
3 moves: 7344 0.156114 %
4 moves: 10954 0.232852 %
5 moves: 84969 1.80621 %
6 moves: 114502 2.434 %
7 moves: 308395 6.55565 %
8 moves: 416548 8.85469 %
```

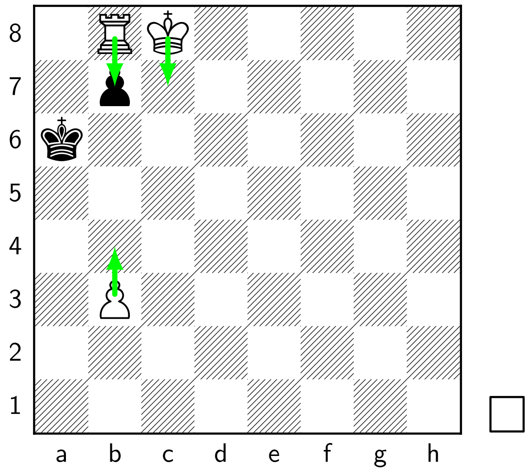
```
9 moves: 637431 13.5501 %
10 moves: 423773 9.00827 %
11 moves: 813642 17.2958 %
12 moves: 569569 12.1075 %
13 moves: 527467 11.2125 %
14 moves: 365460 7.76869 %
15 moves: 203374 4.32318 %
16 moves: 219794 4.67223 %
Average number of moves: 10.8351
Number of positions: 4704266
WWW delta_1 > delta_2: 6
WWW delta_1 = delta_2: 8
WWW delta_1 < delta_2: 9
LLL delta_1 > delta_2: 128418
LLL delta_1 = delta_2: 278929
LLL delta_1 < delta_2: 200941
WWW Sum delta_1: 14
WWW Sum delta_2: 20
LLL Sum delta_1: 367639
LLL Sum delta_2: 419738
```

Appendix C

Some Chess Examples

Refers to page 12.

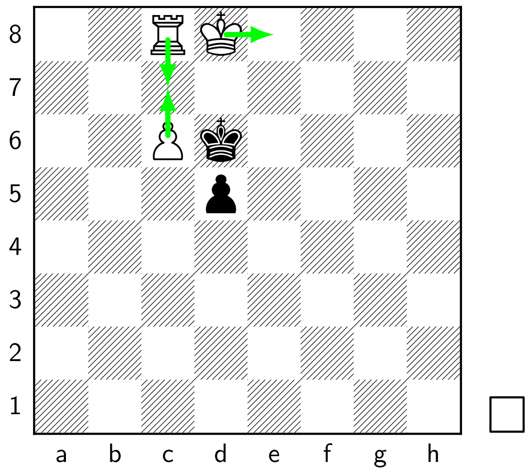
WWW-Example with $d_1 > d_2$



Move	Value
Rb8xb7	Win in 5
Kc8-c7	Win in 12
b3-b4	Win in 13
Kc8-d8	Win in 13
Kc8-d7	Win in 13
Rb8-a8	Win in 14

FEN: `1RK5/1p6/k7/8/8/1P6/8/8 w - - 0 1`

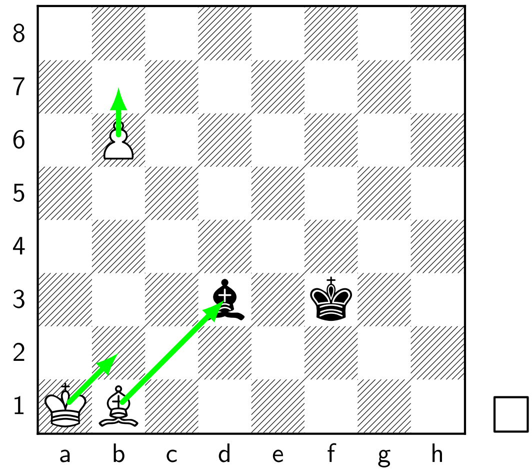
WWW-Example with $d_1 = d_2$



Move	Value
c6-c7	Win in 8
Rc8-c7	Win in 12
Kd8-e8	Win in 16
Rc8-b8	Win in 18
Rc8-a8	Win in 18

FEN: `2RK4/8/2Pk4/3p4/8/8/8/8 w - - 0 1`

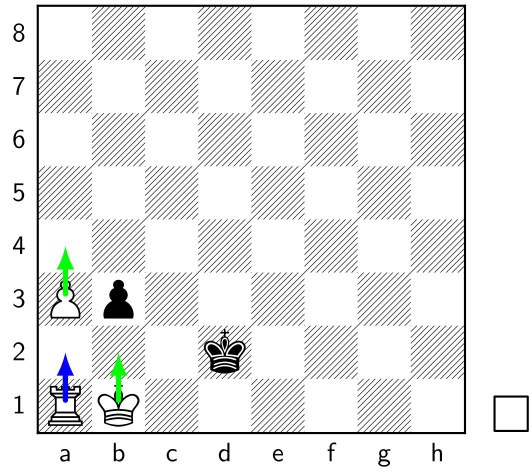
WWW-Example with $d_1 < d_2$



Move	Value
Bb1xd3	Win in 11
b6-b7	Win in 12
Ka1-b2	Win in 23
Ka1-a2	Win in 25
Bb1-a2	Draw
Bb1-c2	Draw

FEN: 8/8/1P6/8/8/3b1k2/8/KB6 w - - 0 1

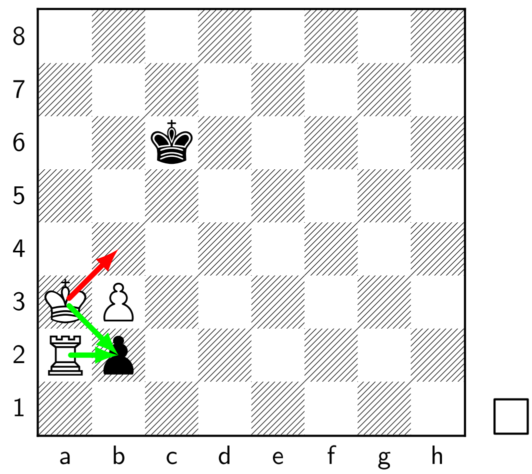
WWD-Example



Move	Value
Kb1-b2	Win in 11
a3-a4	Win in 11
Ra1-a2	Draw

FEN: 8/8/8/8/8/Pp6/3k4/RK6 w - - 0 1

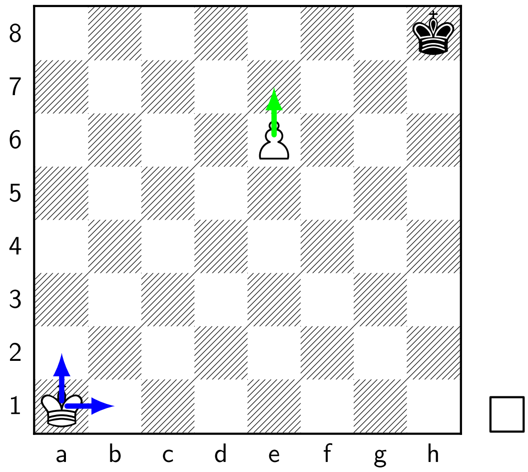
WWL-Example



Move	Value
Ra2xb2	Win in 11
Ka3xb2	Win in 12
Ka3-b4	Loss in 37
Ka3-a4	Loss in 34
b3-b4	Loss in 26
Ra2-a1	Loss in 6

FEN: 8/8/2k5/8/8/KP6/Rp6/8 w - - 0 1

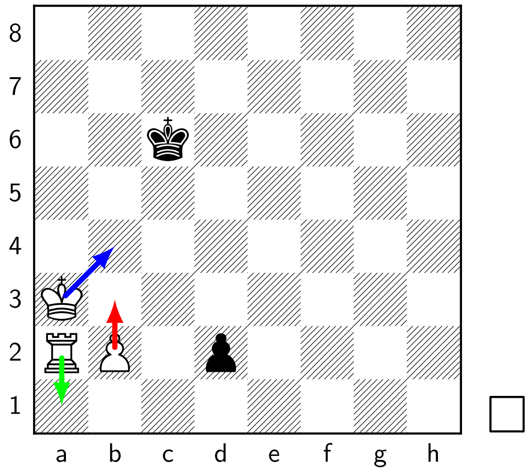
WDD-Example



Move	Value
e6-e7	Win in 10
Ka1-b1	Draw
Ka1-a2	Draw
Ka1-b2	Draw

FEN: `7k/8/4P3/8/8/8/8/K7 w - - 0 1`

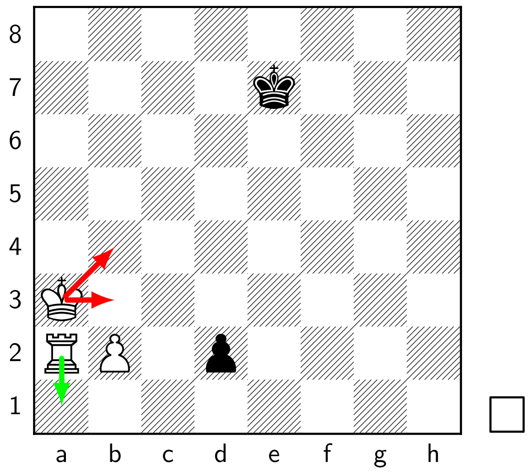
WDL-Example



Move	Value
Ra2-a1	Win in 15
Ka3-b4	Draw
b2-b3	Loss in 88
b2-b4	Loss in 29
Ka3-b3	Loss in 28
Ka3-a4	Loss in 28

FEN: `8/8/2k5/8/8/K7/RP1p4/8 w - - 0 1`

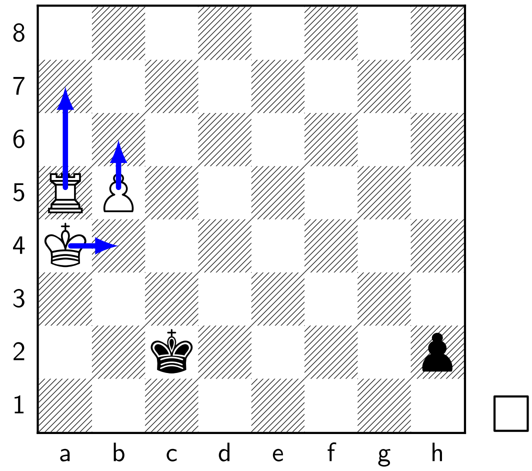
WLL-Example



Move	Value
Ra2-a1	Win in 14
Ka3-b3	Loss in 85
Ka3-b4	Loss in 85
b2-b3	Loss in 51
Ka3-a4	Loss in 38
b2-b4	Loss in 33

FEN: `8/4k3/8/8/8/K7/RP1p4/8 w - - 0 1`

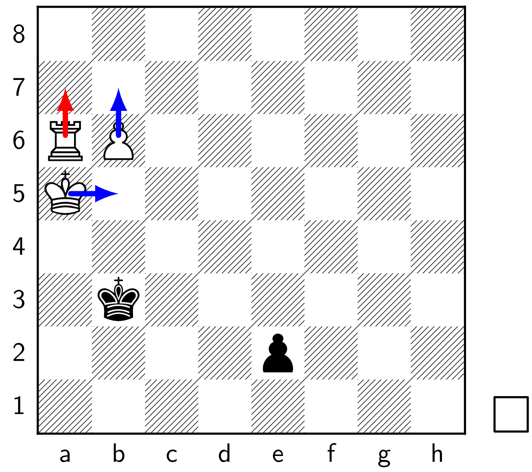
DDD-Example



Move	Value
Ka4-b4	Draw
Ra5-a7	Draw
b5-b6	Draw
Ra5-a6	Loss in 35
Ra5-a8	Loss in 35
Ka4-a3	Loss in 22

`FEN: 8/8/8/RP6/K7/8/2k4p/8 w - - 0 1`

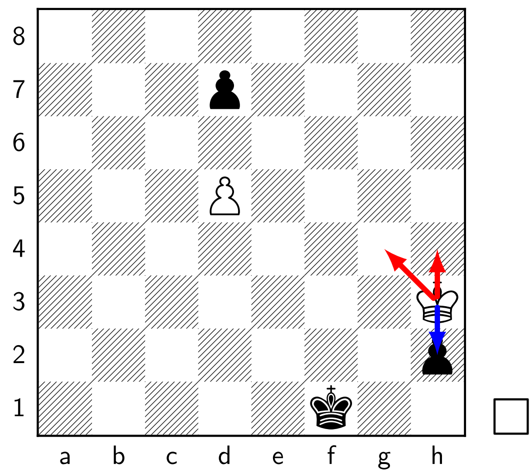
DDL-Example



Move	Value
Ka5-b5	Draw
b6-b7	Draw
Ra6-a7	Loss in 30
Ra6-a8	Loss in 26

`FEN: 8/8/RP6/K7/8/1k6/4p3/8 w - - 0 1`

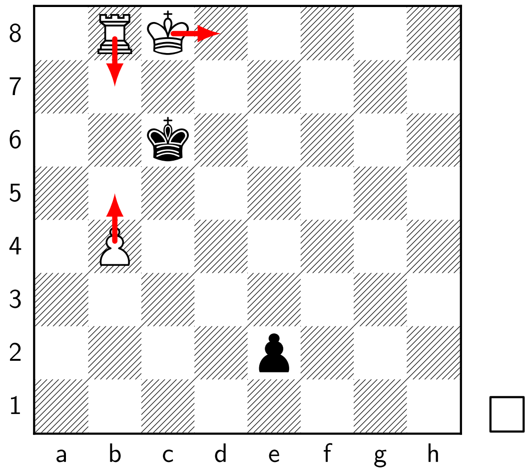
DLL-Example



Move	Value
Kh3xh2	Draw
Kh3-h4	Loss in 9
Kh3-g4	Loss in 9
Kh3-g3	Loss in 7
d5-d6	Loss in 6

`FEN: 8/3p4/8/3P4/8/7K/7p/5k2 w - - 0 1`

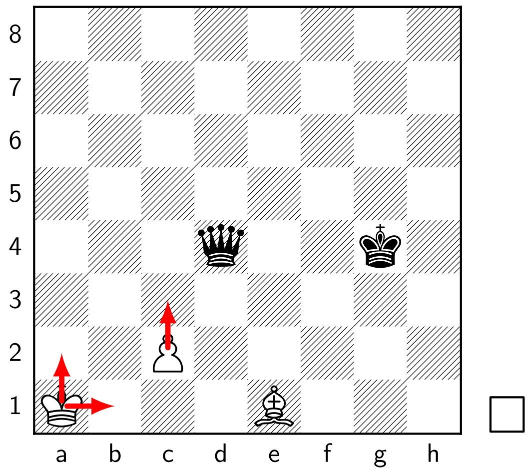
LLL-Example with $d_1 > d_2$



Move	Value
Rb8-b7	Loss in 37
b4-b5	Loss in 35
Kc8-d8	Loss in 34
Rb8-a8	Loss in 25
Rb8-b6	Loss in 8
Rb8-b5	Loss in 8

FEN: `1RK5/8/2k5/8/1P6/8/4p3/8 w - - 0 1`

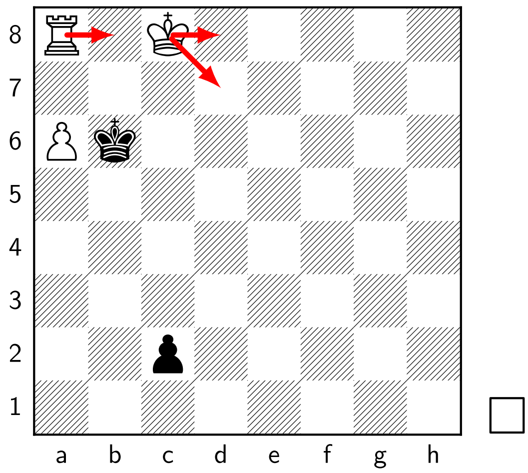
LLL-Example with $d_1 = d_2$



Move	Value
Ka1-a2	Loss in 11
Ka1-b1	Loss in 10
c2-c3	Loss in 9
Be1-c3	Loss in 7

FEN: `8/8/8/8/3q2k1/8/2P5/K3B3 w - - 0 1`

LLL-Example with $d_1 < d_2$



Move	Value
Ra8-b8	Loss in 39
Kc8-d7	Loss in 31
Kc8-d8	Loss in 23
a6-a7	Loss in 13
Ra8-a7	Loss in 10
Kc8-b8	Loss in 7

FEN: `R1K5/8/Pk6/8/8/8/2p5/8 w - - 0 1`

Appendix D

Detailed Results for Chess – Best Moves Rule

Refers to page 16.

Overview of the results presented in this chapter:

D.1 Endgames with 3 Pieces

Cordel Instances

	WWW	WDD	WDL	WLL	DDD	DLL	LLL	WD	WL	DL	W	D	L	Sum
	$d_1 \geq d_2$						$d_1 \geq d_2$							
kkq	–	–	–	–	–	6%	51%	–	–	3%	–	1%	5%	66%
kkr	–	–	–	–	–	8%	54%	–	–	2%	–	0%	2%	66%
kpk	45%	8%	–	–	23%	–	–	0%	–	–	0%	0%	–	76%
kqk	78%	–	–	–	–	–	–	–	–	–	–	–	–	78%
krk	80%	–	–	–	–	–	–	–	–	–	–	–	–	80%
kkp	–	–	–	–	27%	6%	48%	–	–	0%	–	0%	0%	82%
kkb	–	–	–	–	95%	–	–	–	–	–	–	1%	–	96%
kkn	–	–	–	–	97%	–	–	–	–	–	–	1%	–	98%
kbk	–	–	–	–	100%	–	–	–	–	–	–	–	–	100%
knk	–	–	–	–	100%	–	–	–	–	–	–	–	–	100%
Mean	20%	1%	–	–	44%	2%	15%	0%	–	0%	0%	0%	1%	84%

Non-Cordel Instances

	WWW	WWD	WWL	DDL	LLL	WW	DD	LL	Sum
	$d_1 < d_2$				$d_1 < d_2$				
kkq	–	–	–	–	24%	–	–	10%	34%
kkr	–	–	–	–	29%	–	–	5%	34%
kpk	18%	5%	–	–	–	0%	0%	–	24%
kqk	22%	–	–	–	–	–	–	–	22%
krk	20%	–	–	–	–	–	–	–	20%
kkp	–	–	–	8%	9%	–	0%	1%	18%
kkb	–	–	–	–	–	–	4%	–	4%
kkn	–	–	–	–	–	–	2%	–	2%
kbk	–	–	–	–	–	–	–	–	–
knk	–	–	–	–	–	–	–	–	–
Mean	6%	1%	–	1%	6%	0%	1%	2%	16%

D.2 Endgames with 4 Pieces

Cordel Instances

	WWW	WDD	WDL	WLL	DDD	DLL	LLL	WD	WL	DL	W	D	L	Sum
	$d_1 \geq d_2$						$d_1 \geq d_2$							
kkqr	–	–	–	–	–	–	45%	–	–	–	–	–	13%	58%
kkqq	–	–	–	–	–	–	39%	–	–	–	–	–	20%	59%
kkrr	–	–	–	–	–	–	53%	–	–	–	–	–	8%	60%
kkqb	–	–	–	–	–	4%	45%	–	–	3%	–	2%	7%	61%
kkqn	–	–	–	–	–	4%	47%	–	–	3%	–	1%	6%	62%
kkqp	–	–	–	–	–	1%	58%	–	–	1%	–	0%	6%	66%
kppk	63%	2%	–	–	2%	–	–	0%	–	–	0%	0%	–	67%
kkrb	–	–	–	–	–	6%	55%	–	–	2%	–	0%	4%	67%
kkrn	–	–	–	–	–	7%	56%	–	–	2%	–	0%	3%	68%
kqkr	66%	1%	1%	2%	0%	0%	0%	0%	0%	0%	0%	0%	0%	70%
kkbn	–	–	–	–	0%	16%	52%	–	–	1%	–	0%	1%	70%
kkrp	–	–	–	–	–	2%	65%	–	–	0%	–	0%	3%	70%
kpkq	0%	0%	0%	5%	0%	9%	53%	0%	2%	1%	1%	1%	1%	72%
knkq	–	–	–	–	–	15%	54%	–	–	3%	–	1%	1%	73%
kpkr	0%	8%	1%	5%	6%	8%	43%	0%	0%	0%	0%	0%	0%	73%
kbnk	73%	0%	–	–	0%	–	–	–	–	–	–	–	–	74%
krrk	74%	–	–	–	–	–	–	–	–	–	–	–	–	74%
krkq	0%	0%	1%	23%	0%	5%	41%	0%	2%	0%	1%	0%	1%	74%
kkbp	–	–	–	–	4%	9%	62%	–	–	0%	–	0%	1%	76%
krkp	64%	3%	1%	1%	8%	0%	0%	0%	0%	0%	0%	0%	0%	76%
knpk	69%	3%	–	–	4%	–	–	–	–	–	–	–	–	76%
kknp	–	–	–	–	4%	10%	62%	–	–	0%	–	0%	1%	76%
kqkb	75%	1%	–	–	0%	–	–	0%	–	–	0%	0%	–	77%
kqkn	76%	0%	–	–	1%	–	–	0%	–	–	0%	0%	–	77%
kbkq	–	–	–	–	0%	18%	55%	–	–	3%	–	1%	1%	77%
kpkp	18%	8%	1%	2%	25%	6%	19%	0%	0%	0%	0%	0%	0%	77%
kqrk	78%	–	–	–	–	–	–	–	–	–	–	–	–	78%
kbpk	71%	3%	–	–	4%	–	–	0%	–	–	0%	–	–	78%
kqnk	78%	–	–	–	–	–	–	–	–	–	–	–	–	78%
krpk	79%	0%	–	–	–	–	–	–	–	–	0%	–	–	79%
kqkp	78%	0%	0%	–	1%	–	–	0%	0%	–	0%	0%	0%	79%
kqpk	79%	–	–	–	–	–	–	–	–	–	–	–	–	79%
kqbk	80%	–	–	–	–	–	–	–	–	–	–	–	–	80%
kkpp	–	–	–	–	2%	4%	74%	–	–	0%	–	0%	1%	81%
kkbb	–	–	–	–	45%	8%	25%	–	–	1%	–	1%	1%	81%
krnk	81%	0%	–	–	–	–	–	–	–	–	–	–	–	81%
krbk	82%	0%	–	–	–	–	–	–	–	–	–	–	–	82%

continued on next page

	WWW $d_1 \geq d_2$	WDD	WDL	WLL	DDD	DLL	LLL $d_1 \geq d_2$	WD	WL	DL	W	D	L	Sum
krkn	13%	19%	–	–	52%	–	–	0%	–	–	0%	0%	–	83%
kqqk	84%	–	–	–	–	–	–	–	–	–	–	–	–	84%
knkr	–	0%	–	–	70%	9%	7%	–	–	0%	–	0%	0%	86%
krkb	2%	23%	–	–	65%	–	–	0%	–	–	0%	0%	–	89%
kbbk	39%	0%	–	–	51%	–	–	–	–	–	–	–	–	90%
kpkn	10%	14%	0%	–	67%	0%	–	0%	0%	0%	0%	0%	0%	91%
knkp	0%	0%	–	–	76%	6%	9%	–	–	0%	–	0%	0%	91%
kqkq	0%	27%	2%	5%	54%	1%	0%	2%	0%	0%	1%	0%	0%	93%
kpkb	4%	15%	–	–	75%	–	–	0%	–	0%	0%	0%	–	94%
krkr	1%	22%	1%	2%	68%	0%	0%	0%	0%	0%	0%	0%	0%	95%
kbkp	–	0%	–	–	85%	6%	4%	–	–	0%	–	0%	0%	95%
kbkr	–	–	–	–	91%	2%	2%	–	–	0%	–	0%	0%	95%
kknn	–	–	–	–	95%	–	–	–	–	0%	–	1%	–	96%
knkb	–	0%	–	–	99%	–	–	–	–	–	–	0%	–	99%
kbkb	–	0%	–	–	99%	–	–	–	–	–	–	0%	–	99%
knkn	–	0%	–	–	100%	–	–	–	–	–	–	0%	–	100%
kbkn	–	0%	–	–	100%	–	–	–	–	–	–	0%	–	100%
knnk	–	0%	–	–	100%	–	–	–	–	–	–	–	–	100%
Mean	26%	3%	0%	1%	26%	3%	19%	0%	0%	0%	0%	0%	1%	80%

Non-Cordel Instances

	WWW $d_1 < d_2$	WWD	WWL	DDL	LLL $d_1 < d_2$	WW	DD	LL	Sum
kkqr	–	–	–	–	19%	–	–	23%	42%
kkqq	–	–	–	–	11%	–	–	30%	41%
kkrr	–	–	–	–	24%	–	–	16%	40%
kkqb	–	–	–	–	23%	–	–	16%	39%
kkqn	–	–	–	–	25%	–	–	13%	38%
kkqp	–	–	–	–	20%	–	–	13%	34%
kppk	32%	1%	–	–	–	0%	0%	–	33%
kkrb	–	–	–	–	23%	–	–	9%	33%
kkrn	–	–	–	–	25%	–	–	8%	32%
kqkr	27%	1%	1%	0%	0%	1%	0%	0%	30%
kkbn	–	–	–	1%	24%	–	0%	5%	30%
kkrp	–	–	–	–	23%	–	–	7%	30%
kpkq	0%	0%	0%	0%	20%	0%	0%	7%	28%
knkq	–	–	–	1%	23%	–	0%	3%	27%
kpkr	0%	1%	0%	3%	19%	0%	0%	3%	27%
kbnk	25%	1%	–	–	–	–	–	–	26%
krrk	26%	0%	–	–	–	–	–	–	26%

continued on next page

	WWW $d_1 < d_2$	WWD	WWL	DDL	LLL $d_1 < d_2$	WW	DD	LL	Sum
krkq	–	0%	2%	0%	21%	0%	0%	2%	26%
kkbp	–	–	–	3%	17%	–	0%	4%	24%
krkp	20%	2%	2%	1%	0%	0%	0%	–	24%
knpk	22%	2%	–	–	–	0%	0%	–	24%
kknp	–	–	–	5%	17%	–	0%	2%	24%
kqkb	22%	1%	–	–	–	0%	0%	–	23%
kqkn	23%	0%	–	–	–	0%	0%	–	23%
kbkq	–	–	–	1%	19%	–	0%	3%	23%
kpkp	9%	5%	1%	3%	5%	0%	0%	0%	23%
kqrk	22%	–	–	–	–	–	–	–	22%
kbpk	20%	2%	–	–	–	0%	–	–	22%
kqnk	22%	0%	–	–	–	–	–	–	22%
krpk	21%	0%	–	–	–	0%	–	–	21%
kqkp	21%	0%	0%	0%	–	0%	–	–	21%
kqpk	21%	0%	–	–	–	–	–	–	21%
kqbk	20%	–	–	–	–	–	–	–	20%
kkpp	–	–	–	2%	16%	–	0%	2%	19%
kkbb	–	–	–	1%	11%	–	4%	4%	19%
krnk	19%	0%	–	–	–	–	–	–	19%
krbk	18%	0%	–	–	–	–	–	–	18%
krkn	7%	9%	–	–	–	0%	0%	–	17%
kqqk	16%	–	–	–	–	–	–	–	16%
knkr	–	–	–	9%	3%	–	1%	1%	14%
krkb	1%	9%	–	–	–	0%	0%	–	11%
kbbk	10%	0%	–	–	–	–	–	–	10%
kpkn	5%	3%	–	0%	–	0%	1%	0%	9%
knkp	0%	0%	–	4%	4%	–	0%	0%	9%
kqkq	0%	2%	2%	1%	0%	0%	1%	0%	7%
kpkb	2%	2%	–	0%	–	0%	1%	–	6%
krkr	0%	3%	1%	1%	0%	0%	1%	0%	5%
kbkp	–	–	–	4%	1%	–	0%	0%	5%
kbkr	–	–	–	2%	1%	–	1%	0%	5%
kknn	–	–	–	0%	–	–	4%	–	4%
knkb	–	–	–	–	–	–	1%	–	1%
kbkb	–	–	–	–	–	–	1%	–	1%
knkn	–	–	–	–	–	–	0%	–	0%
kbkn	–	–	–	–	–	–	0%	–	0%
knnk	–	0%	–	–	–	–	–	–	0%
Mean	8%	1%	0%	1%	7%	0%	0%	3%	20%

D.3 Endgames with 5 Pieces

Cordel Instances

	WWW	WDD	WDL	WLL	DDD	DLL	LLL	WD	WL	DL	W	D	L	Sum
	$d_1 \geq d_2$						$d_1 \geq d_2$							
kkrrb	–	–	–	–	–	–	47%	–	–	–	–	–	10%	58%
kkqbn	–	–	–	–	–	0%	47%	–	–	0%	–	0%	11%	58%
kkqrn	–	–	–	–	–	–	43%	–	–	–	–	–	15%	58%
kkrrn	–	–	–	–	–	–	49%	–	–	–	–	–	9%	58%
kkqrb	–	–	–	–	–	–	41%	–	–	–	–	–	17%	58%
kkqqb	–	–	–	–	–	–	33%	–	–	–	–	–	25%	58%
kkqqn	–	–	–	–	–	–	36%	–	–	–	–	–	22%	59%
kkqbb	–	–	–	–	–	1%	43%	–	–	2%	–	1%	12%	59%
kkqrp	–	–	–	–	–	–	46%	–	–	0%	–	–	13%	59%
kkqqp	–	–	–	–	–	–	39%	–	–	–	–	–	20%	59%
kkqbp	–	–	–	–	–	0%	50%	–	–	0%	–	0%	9%	59%
kkrrr	–	–	–	–	–	–	45%	–	–	–	–	–	15%	60%
kkrrp	–	–	–	–	–	–	53%	–	–	0%	–	–	7%	60%
kkqqr	–	–	–	–	–	–	30%	–	–	–	–	–	31%	60%
kkqrr	–	–	–	–	–	–	38%	–	–	–	–	–	23%	61%
kkqqq	–	–	–	–	–	–	22%	–	–	–	–	–	39%	61%
kkqnp	–	–	–	–	–	0%	53%	–	–	0%	–	0%	8%	61%
kkqnn	–	–	–	–	–	3%	46%	–	–	3%	–	1%	8%	62%
kkrbn	–	–	–	–	–	0%	57%	–	–	0%	–	0%	5%	63%
kpppk	63%	0%	–	–	0%	–	–	0%	–	–	0%	0%	–	63%
kpkqq	–	0%	0%	0%	0%	0%	49%	–	0%	0%	0%	0%	13%	63%
kkrbb	–	–	–	–	–	3%	53%	–	–	1%	–	0%	6%	63%
kpkqr	–	0%	0%	0%	0%	1%	55%	0%	0%	0%	0%	0%	7%	63%
krrkr	60%	3%	0%	0%	1%	0%	–	0%	0%	0%	0%	0%	0%	64%
kqrkq	54%	3%	1%	3%	2%	0%	0%	1%	0%	0%	1%	0%	0%	65%
kkrnn	–	–	–	–	–	6%	54%	–	–	2%	–	0%	4%	66%
kppkp	39%	5%	1%	2%	6%	3%	10%	0%	0%	0%	0%	0%	0%	66%
kpkrr	0%	0%	0%	0%	0%	5%	58%	0%	0%	0%	0%	0%	2%	67%
kkqpp	–	–	–	–	–	0%	60%	–	–	0%	–	0%	7%	67%
kkrbp	–	–	–	–	–	0%	63%	–	–	0%	–	0%	4%	67%
kpkqb	0%	0%	0%	0%	0%	9%	50%	0%	0%	3%	0%	1%	3%	68%
kqqkq	62%	4%	0%	0%	1%	0%	0%	0%	0%	0%	1%	0%	0%	68%
kkrnp	–	–	–	–	–	0%	65%	–	–	0%	–	0%	3%	68%
knkqq	–	–	–	–	–	0%	58%	–	–	0%	–	0%	10%	68%
kbkqq	–	–	–	–	–	0%	59%	–	–	0%	–	0%	10%	69%
kqkqq	0%	0%	0%	0%	1%	20%	35%	0%	0%	4%	0%	2%	6%	69%
krrkq	9%	13%	1%	6%	32%	2%	2%	1%	1%	0%	1%	0%	0%	70%

continued on next page

	WWW	WDD	WDL	WLL	DDD	DLL	LLL	WD	WL	DL	W	D	L	Sum
	$d_1 \geq d_2$						$d_1 \geq d_2$							
kpkqn	0%	0%	0%	1%	0%	10%	51%	0%	0%	3%	0%	1%	2%	70%
kpkrb	0%	1%	1%	0%	1%	10%	54%	0%	0%	1%	0%	0%	1%	70%
kqkrp	55%	4%	0%	2%	6%	1%	1%	0%	0%	0%	0%	0%	0%	70%
kpkrn	0%	1%	2%	1%	2%	10%	53%	0%	0%	1%	0%	0%	1%	70%
kkrpp	–	–	–	–	–	0%	67%	–	–	0%	–	0%	3%	70%
krkrr	–	0%	0%	0%	1%	26%	40%	0%	0%	1%	0%	1%	1%	70%
krkqq	–	–	0%	0%	–	0%	61%	–	0%	0%	0%	0%	9%	70%
kqkrb	23%	18%	1%	2%	21%	1%	2%	0%	0%	0%	0%	0%	0%	71%
knkrn	–	0%	0%	0%	1%	21%	48%	–	0%	1%	–	0%	0%	71%
kqkrn	34%	15%	1%	2%	17%	1%	1%	0%	0%	0%	0%	0%	0%	71%
krpkb	64%	3%	0%	–	4%	–	–	0%	0%	0%	0%	0%	0%	71%
kqkbn	68%	1%	0%	1%	1%	1%	0%	0%	0%	0%	0%	0%	0%	71%
kkbbn	–	–	–	–	0%	4%	65%	–	–	0%	–	0%	3%	71%
knkrb	–	0%	0%	0%	1%	20%	49%	–	–	1%	–	0%	1%	71%
knkqr	–	–	0%	0%	–	5%	59%	–	–	2%	–	1%	4%	71%
knkrr	–	–	0%	0%	0%	12%	56%	–	–	1%	–	0%	1%	71%
kbnkp	60%	3%	1%	2%	3%	1%	2%	0%	0%	0%	0%	0%	0%	72%
knpkp	46%	5%	1%	2%	10%	2%	5%	0%	0%	0%	0%	0%	0%	72%
kqknn	62%	4%	0%	0%	6%	–	–	0%	0%	0%	0%	0%	–	72%
kqkbb	66%	2%	0%	1%	2%	0%	0%	0%	0%	0%	0%	0%	0%	72%
krpkn	67%	2%	0%	0%	2%	0%	–	0%	0%	0%	0%	0%	0%	72%
knnkq	–	0%	0%	0%	9%	22%	37%	0%	–	3%	–	1%	0%	72%
kppkr	6%	10%	2%	7%	12%	5%	30%	0%	0%	0%	0%	0%	0%	72%
kqkqr	0%	2%	4%	13%	3%	13%	29%	0%	2%	1%	1%	0%	3%	72%
kbkrr	–	–	–	–	0%	18%	52%	–	–	1%	–	0%	1%	73%
krpkq	1%	5%	2%	20%	5%	4%	31%	0%	3%	0%	1%	0%	1%	73%
kqrkr	71%	1%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	73%
krnkb	69%	2%	0%	–	2%	–	–	0%	0%	0%	0%	0%	–	73%
kpkbn	2%	7%	1%	2%	3%	16%	41%	0%	0%	0%	0%	0%	0%	73%
kqpkr	71%	1%	0%	1%	0%	0%	0%	0%	0%	0%	0%	0%	0%	73%
knkqb	–	–	0%	0%	0%	11%	55%	0%	–	3%	–	1%	2%	73%
krnkn	72%	1%	–	–	1%	–	–	0%	–	–	0%	0%	–	73%
krbkb	70%	2%	–	–	2%	–	–	0%	0%	0%	0%	0%	0%	73%
knpkq	0%	1%	1%	12%	2%	7%	47%	0%	2%	1%	1%	0%	1%	73%
kbnkq	0%	1%	2%	16%	0%	5%	45%	0%	2%	0%	1%	0%	0%	73%
krpkp	72%	1%	0%	1%	0%	0%	0%	0%	0%	0%	0%	0%	0%	73%
knppk	73%	0%	–	–	0%	–	–	0%	–	–	–	–	–	74%
krrkp	74%	0%	0%	0%	0%	0%	–	0%	0%	0%	0%	0%	0%	74%
kknnn	–	–	–	–	1%	20%	50%	–	–	1%	–	0%	2%	74%
kbpkp	53%	5%	1%	3%	7%	1%	3%	0%	0%	0%	0%	0%	0%	74%
knkqn	–	–	0%	0%	–	13%	55%	0%	–	3%	–	1%	1%	74%
kkbnn	–	–	–	–	0%	8%	63%	–	–	1%	–	0%	2%	74%

continued on next page

	WWW	WDD	WDL	WLL	DDD	DLL	LLL	WD	WL	DL	W	D	L	Sum
	$d_1 \geq d_2$						$d_1 \geq d_2$							
knkrp	–	0%	0%	0%	4%	17%	53%	–	0%	0%	–	0%	0%	74%
krrkn	73%	0%	–	–	0%	–	–	0%	–	–	0%	0%	–	74%
kbkrn	–	0%	0%	0%	2%	24%	46%	–	–	1%	–	0%	0%	74%
kpkqp	0%	0%	0%	2%	0%	6%	62%	0%	1%	1%	0%	0%	2%	74%
kbkqr	–	–	–	–	0%	8%	59%	–	–	2%	–	1%	4%	74%
krbkn	72%	0%	–	–	1%	–	–	0%	–	–	0%	0%	–	74%
krkpp	46%	5%	1%	2%	13%	3%	5%	0%	0%	0%	0%	0%	0%	74%
kqkrr	10%	18%	1%	5%	29%	2%	5%	1%	0%	0%	0%	0%	1%	74%
kqqkr	74%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	74%
kqnkr	71%	2%	0%	0%	1%	0%	0%	0%	0%	0%	0%	0%	0%	74%
kbbkq	0%	0%	1%	9%	2%	13%	45%	0%	1%	1%	0%	1%	0%	74%
kbkrb	–	0%	0%	0%	1%	24%	47%	–	0%	1%	–	0%	1%	74%
kkbbp	–	–	–	–	1%	4%	67%	–	–	0%	–	0%	2%	75%
krrkb	73%	1%	–	–	1%	–	–	0%	–	–	0%	0%	–	75%
kqbkr	72%	2%	0%	0%	1%	0%	0%	0%	0%	0%	0%	0%	0%	75%
kqknp	71%	1%	0%	0%	2%	0%	0%	0%	0%	0%	0%	0%	0%	75%
kkbbb	–	–	–	–	21%	7%	43%	–	–	1%	–	1%	3%	75%
krkqr	–	0%	0%	0%	0%	13%	54%	0%	0%	2%	0%	1%	4%	75%
kppkn	28%	12%	0%	0%	35%	0%	0%	0%	0%	0%	0%	0%	0%	75%
krkqb	–	0%	0%	1%	0%	21%	47%	0%	0%	3%	0%	1%	2%	75%
krrrk	75%	0%	–	–	–	–	–	–	–	–	–	–	–	75%
kqkbp	70%	1%	0%	1%	2%	0%	0%	0%	0%	0%	0%	0%	0%	75%
krrpk	75%	0%	–	–	0%	–	–	–	–	–	–	–	–	75%
kkbnp	–	–	–	–	0%	1%	73%	–	–	0%	–	0%	1%	76%
knkqp	–	0%	0%	0%	–	9%	63%	0%	0%	2%	0%	1%	1%	76%
kppkq	0%	1%	1%	9%	1%	10%	49%	0%	3%	0%	1%	0%	1%	76%
kbkrp	–	0%	0%	0%	5%	21%	49%	–	0%	0%	–	0%	0%	76%
kpkrp	0%	1%	3%	5%	0%	6%	60%	0%	0%	0%	0%	0%	0%	76%
kpkpp	9%	5%	1%	3%	7%	8%	42%	0%	0%	0%	0%	0%	0%	76%
kbkqb	–	–	0%	0%	0%	14%	55%	–	0%	3%	–	1%	2%	76%
kbpkq	0%	1%	1%	14%	3%	6%	47%	0%	2%	0%	1%	0%	1%	76%
krrnk	76%	0%	–	–	–	–	–	–	–	–	–	–	–	76%
kpknp	6%	6%	1%	2%	12%	11%	38%	0%	0%	0%	0%	0%	0%	76%
kbppk	76%	0%	–	–	0%	–	–	0%	–	–	0%	–	–	76%
krkqp	0%	0%	0%	16%	0%	5%	51%	0%	2%	0%	1%	0%	1%	76%
kbkqn	–	–	0%	0%	0%	15%	55%	–	–	3%	–	1%	1%	76%
kknnp	–	–	–	–	1%	8%	66%	–	–	0%	–	0%	1%	77%
krnkq	1%	8%	4%	12%	24%	10%	14%	0%	2%	0%	1%	0%	0%	77%
kkbpp	–	–	–	–	0%	1%	75%	–	–	0%	–	0%	1%	77%
krkqn	–	0%	0%	2%	0%	21%	47%	0%	0%	3%	0%	1%	1%	77%
krbkq	2%	10%	4%	9%	32%	8%	7%	1%	2%	0%	1%	0%	0%	77%
kqpkb	76%	1%	–	0%	0%	–	0%	0%	0%	0%	0%	0%	0%	77%

continued on next page

	WWW $d_1 \geq d_2$	WDD	WDL	WLL	DDD	DLL	LLL $d_1 \geq d_2$	WD	WL	DL	W	D	L	Sum
kqpkn	77%	0%	–	–	0%	–	–	0%	–	–	0%	0%	0%	77%
kqkpp	75%	1%	0%	0%	1%	0%	0%	0%	0%	0%	0%	0%	0%	77%
krnkp	74%	1%	0%	0%	1%	0%	0%	0%	0%	0%	0%	0%	0%	77%
krrbk	77%	0%	–	–	0%	–	–	–	–	–	–	–	–	77%
knnnk	75%	1%	–	–	1%	–	–	–	–	–	–	–	–	77%
kqrkb	77%	0%	–	–	0%	–	–	0%	–	–	0%	0%	–	77%
kbkqp	–	–	0%	0%	0%	13%	61%	–	0%	2%	–	1%	1%	78%
kqrkn	78%	0%	–	–	0%	–	–	0%	–	–	0%	0%	–	78%
kqpkp	78%	0%	0%	0%	0%	–	–	0%	0%	0%	0%	0%	0%	78%
kpkbp	3%	6%	2%	3%	8%	11%	45%	0%	0%	0%	0%	0%	0%	78%
kqbkb	77%	1%	–	0%	0%	–	0%	0%	0%	0%	0%	0%	0%	78%
kqnkp	78%	0%	0%	0%	0%	–	–	0%	0%	0%	0%	0%	0%	78%
kqnkb	77%	1%	0%	–	0%	–	–	0%	0%	–	0%	0%	–	78%
knnpk	75%	2%	–	–	2%	–	–	–	–	–	–	–	–	78%
krbkp	76%	1%	0%	0%	1%	0%	0%	0%	0%	0%	0%	0%	0%	78%
kqpkq	20%	20%	1%	4%	28%	0%	0%	2%	0%	0%	1%	0%	0%	78%
kknpp	–	–	–	–	0%	1%	76%	–	–	0%	–	0%	1%	78%
kqnkn	77%	0%	–	–	1%	–	–	0%	–	–	0%	0%	–	78%
krpkr	28%	16%	1%	1%	32%	0%	0%	0%	0%	0%	0%	0%	0%	78%
kbnpk	78%	0%	–	–	0%	–	–	–	–	–	–	–	–	78%
kqrkp	78%	0%	0%	–	0%	–	–	0%	0%	0%	0%	0%	0%	78%
krppk	78%	0%	–	–	0%	–	–	0%	–	–	0%	–	–	78%
kqqkb	79%	0%	–	–	–	–	–	0%	–	–	0%	–	–	79%
kqrpk	79%	0%	–	–	0%	–	–	–	–	–	–	–	–	79%
kqbkn	78%	0%	–	–	1%	–	–	0%	–	–	0%	0%	–	79%
kqbkp	79%	0%	0%	0%	0%	–	–	0%	0%	0%	0%	0%	0%	79%
kqppk	79%	0%	–	–	0%	–	–	–	–	–	0%	–	–	79%
kbnnk	79%	0%	–	–	0%	–	–	–	–	–	–	–	–	79%
kqrrk	80%	0%	–	–	–	–	–	–	–	–	–	–	–	80%
kqnpk	80%	0%	–	–	0%	–	–	–	–	–	–	–	–	80%
kbbnk	80%	0%	–	–	0%	–	–	–	–	–	–	–	–	80%
kqqkn	80%	0%	–	–	–	–	–	0%	–	–	0%	0%	–	80%
krnpk	80%	0%	–	–	0%	–	–	–	–	–	–	–	–	80%
kqrnk	80%	0%	–	–	–	–	–	–	–	–	–	–	–	80%
kppkb	22%	14%	0%	0%	45%	0%	0%	0%	0%	0%	0%	0%	0%	80%
kqbpk	80%	0%	–	–	–	–	–	–	–	–	–	–	–	80%
kkppp	–	–	–	–	0%	0%	79%	–	–	0%	–	0%	1%	81%
kqrbk	81%	0%	–	–	–	–	–	–	–	–	–	–	–	81%
kbbpk	78%	1%	–	–	2%	–	–	–	–	–	–	–	–	81%
kqnnk	81%	0%	–	–	–	–	–	–	–	–	–	–	–	81%
krbpk	81%	0%	–	–	0%	–	–	–	–	–	0%	–	–	81%
krnnk	81%	0%	–	–	0%	–	–	–	–	–	–	–	–	81%

continued on next page

	WWW $d_1 \geq d_2$	WDD	WDL	WLL	DDD	DLL	LLL $d_1 \geq d_2$	WD	WL	DL	W	D	L	Sum
kqbnk	82%	0%	–	–	0%	–	–	–	–	–	–	–	–	82%
kqbkq	9%	23%	1%	3%	41%	0%	0%	2%	0%	0%	1%	0%	0%	82%
krbnk	82%	0%	–	–	0%	–	–	–	–	–	–	–	–	82%
kbpkn	25%	14%	0%	0%	43%	0%	0%	0%	0%	0%	0%	0%	0%	82%
krbbk	82%	0%	–	–	0%	–	–	–	–	–	–	–	–	82%
kqqkp	82%	0%	0%	–	0%	–	–	0%	0%	0%	0%	0%	0%	83%
knkbb	–	0%	0%	0%	50%	13%	20%	–	–	0%	–	0%	0%	83%
kqbbk	83%	0%	–	–	–	–	–	–	–	–	–	–	–	83%
kqqpk	83%	–	–	–	–	–	–	–	–	–	–	–	–	83%
kqqnk	84%	–	–	–	–	–	–	–	–	–	–	–	–	84%
kbbkn	32%	0%	–	–	52%	–	–	0%	–	–	0%	0%	–	84%
krkrp	0%	10%	2%	5%	39%	10%	17%	0%	0%	0%	0%	0%	0%	84%
kqqbk	84%	–	–	–	–	–	–	–	–	–	–	–	–	84%
knkpp	0%	0%	0%	0%	47%	10%	28%	–	–	0%	–	0%	0%	84%
kpkbb	1%	8%	0%	0%	42%	9%	23%	0%	0%	0%	0%	0%	0%	84%
knpkn	21%	13%	0%	0%	51%	0%	0%	0%	0%	0%	0%	0%	0%	85%
kqkqp	0%	17%	3%	9%	38%	6%	7%	1%	1%	1%	1%	0%	0%	85%
kbbkp	34%	1%	0%	1%	47%	1%	1%	0%	0%	0%	0%	0%	0%	86%
kqnkq	5%	25%	2%	4%	47%	0%	0%	2%	0%	0%	1%	0%	0%	86%
krbkr	5%	23%	0%	0%	58%	0%	0%	0%	0%	0%	0%	0%	0%	86%
kqqrk	86%	–	–	–	–	–	–	–	–	–	–	–	–	86%
kpknn	2%	8%	0%	0%	59%	9%	9%	0%	0%	0%	0%	0%	0%	86%
knnkp	13%	7%	0%	0%	63%	2%	1%	0%	–	0%	0%	0%	0%	86%
knkbp	0%	0%	0%	0%	57%	10%	20%	–	0%	0%	–	0%	0%	86%
kbbbk	61%	0%	–	–	26%	–	–	–	–	–	–	–	–	87%
kbpkb	13%	15%	0%	0%	58%	0%	0%	0%	0%	0%	0%	0%	0%	87%
krknp	4%	15%	0%	0%	63%	2%	1%	0%	0%	0%	0%	0%	0%	87%
knpkr	2%	17%	1%	1%	61%	3%	2%	0%	0%	0%	0%	0%	0%	88%
knpkb	12%	16%	0%	0%	61%	0%	0%	0%	0%	0%	0%	0%	0%	88%
knknp	0%	0%	0%	0%	64%	8%	17%	–	–	0%	–	0%	0%	88%
kqqqk	89%	–	–	–	–	–	–	–	–	–	–	–	–	89%
krnkr	3%	23%	0%	0%	62%	0%	0%	0%	0%	0%	0%	0%	0%	89%
kqkqb	0%	18%	2%	4%	50%	5%	3%	1%	1%	1%	1%	1%	1%	89%
kbkpp	0%	0%	0%	0%	59%	11%	19%	–	–	0%	–	0%	0%	89%
kbpkr	3%	19%	0%	0%	64%	1%	1%	0%	0%	0%	0%	0%	0%	89%
krkbp	1%	16%	0%	0%	67%	3%	2%	0%	0%	0%	0%	0%	0%	90%
kbnkn	4%	18%	–	–	68%	–	–	0%	–	–	0%	0%	–	90%
kqkqn	0%	21%	2%	4%	53%	3%	2%	2%	1%	1%	1%	1%	0%	91%
kbkbp	0%	0%	0%	0%	72%	9%	10%	–	–	0%	–	0%	0%	92%
knkbn	–	0%	0%	–	85%	5%	3%	–	–	0%	–	0%	0%	92%
krkrb	0%	1%	0%	0%	84%	4%	3%	0%	0%	1%	0%	0%	0%	93%
kbknp	0%	0%	0%	0%	77%	7%	9%	–	0%	0%	–	0%	0%	93%

continued on next page

	WWW	WDD	WDL	WLL	DDD	DLL	LLL	WD	WL	DL	W	D	L	Sum
	$d_1 \geq d_2$						$d_1 \geq d_2$							
kbnkr	1%	20%	0%	0%	72%	0%	0%	0%	0%	0%	0%	0%	0%	94%
krkrn	0%	3%	0%	0%	85%	3%	2%	0%	0%	0%	0%	0%	0%	94%
kbbkr	1%	11%	0%	0%	82%	0%	0%	0%	0%	0%	0%	0%	0%	95%
kbnkb	1%	20%	0%	–	74%	0%	–	0%	0%	0%	0%	0%	0%	95%
kbbkb	1%	10%	–	–	84%	0%	–	0%	–	–	0%	0%	0%	95%
kbkbb	–	0%	0%	0%	93%	2%	1%	0%	–	0%	–	0%	0%	96%
kbkbn	–	0%	–	0%	93%	2%	1%	–	–	0%	–	0%	0%	96%
krkbn	0%	3%	0%	0%	92%	1%	1%	0%	0%	0%	0%	0%	0%	97%
krkbb	0%	1%	0%	0%	94%	1%	1%	0%	0%	0%	0%	0%	0%	97%
knnkr	–	0%	–	0%	97%	0%	0%	–	–	0%	–	0%	0%	98%
krknn	0%	5%	0%	0%	93%	0%	0%	0%	–	0%	0%	0%	0%	98%
knknn	–	0%	–	–	100%	0%	0%	–	–	0%	–	0%	0%	100%
knnkb	0%	0%	–	–	100%	–	–	0%	–	0%	–	0%	–	100%
kbknn	–	0%	–	–	100%	0%	0%	–	–	0%	–	0%	0%	100%
knnkn	0%	0%	–	–	100%	–	–	0%	–	–	–	0%	–	100%
Mean	29%	3%	0%	1%	17%	4%	20%	0%	0%	0%	0%	0%	2%	77%

Non-Cordel Instances

	WWW	WWD	WWL	DDL	LLL	WW	DD	LL	Sum
	$d_1 < d_2$				$d_1 < d_2$				
kkrrb	–	–	–	–	22%	–	–	20%	42%
kkqbn	–	–	–	–	20%	–	–	22%	42%
kkqrn	–	–	–	–	17%	–	–	25%	42%
kkrrn	–	–	–	–	23%	–	–	19%	42%
kkqrb	–	–	–	–	15%	–	–	27%	42%
kkqqb	–	–	–	–	10%	–	–	32%	42%
kkqqn	–	–	–	–	11%	–	–	31%	41%
kkqbb	–	–	–	–	18%	–	–	23%	41%
kkqrp	–	–	–	–	18%	–	–	23%	41%
kkqqp	–	–	–	–	12%	–	–	29%	41%
kkqbp	–	–	–	–	21%	–	–	20%	41%
kkrrr	–	–	–	–	15%	–	–	25%	40%
kkrrp	–	–	–	–	23%	–	–	17%	40%
kkqqr	–	–	–	–	7%	–	–	32%	40%
kkqrr	–	–	–	–	10%	–	–	29%	39%
kkqqq	–	–	–	–	6%	–	–	33%	39%
kkqnp	–	–	–	–	21%	–	–	17%	39%
kkqnn	–	–	–	–	22%	–	–	17%	38%
kkrbn	–	–	–	–	23%	–	–	14%	37%
kpppk	37%	0%	–	–	–	0%	0%	–	37%

continued on next page

	WWW $d_1 < d_2$	WWD	WWL	DDL	LLL $d_1 < d_2$	WW	DD	LL	Sum
kpkqq	–	0%	0%	0%	13%	–	0%	24%	37%
kkrbb	–	–	–	–	21%	–	0%	16%	37%
kpkqr	–	0%	0%	0%	18%	–	0%	19%	37%
krrkr	33%	2%	0%	0%	–	1%	0%	0%	36%
kqrkq	26%	3%	3%	0%	0%	2%	0%	0%	35%
kkrnn	–	–	–	–	23%	–	0%	11%	34%
kppkp	24%	5%	1%	1%	3%	0%	0%	0%	34%
kpkrr	0%	0%	0%	0%	22%	0%	0%	12%	33%
kkqpp	–	–	–	0%	19%	–	–	15%	33%
kkrbp	–	–	–	–	21%	–	0%	12%	33%
kpkqb	0%	0%	0%	0%	21%	0%	0%	10%	32%
kqqkq	24%	5%	0%	0%	0%	2%	0%	0%	32%
kkrnp	–	–	–	–	22%	–	–	10%	32%
knkqq	–	–	–	–	13%	–	–	18%	32%
kbkqq	–	–	–	0%	14%	–	0%	17%	31%
kqkqq	0%	0%	0%	5%	15%	0%	1%	9%	31%
krrkq	9%	13%	2%	2%	2%	0%	1%	1%	30%
kpkqn	0%	0%	0%	0%	22%	0%	0%	8%	30%
kpkrb	0%	0%	0%	5%	20%	0%	0%	6%	30%
kqkrp	24%	3%	1%	1%	1%	1%	0%	0%	30%
kpkrn	0%	0%	0%	5%	21%	0%	0%	4%	30%
kkrpp	–	–	–	0%	21%	–	–	9%	30%
krkrr	–	0%	0%	5%	20%	–	0%	4%	30%
krkqq	–	–	–	0%	13%	–	0%	16%	30%
kqkrb	12%	13%	1%	2%	1%	0%	1%	0%	29%
knkrn	–	–	0%	3%	25%	–	0%	2%	29%
kqkrn	16%	10%	1%	1%	1%	0%	0%	0%	29%
krpkb	26%	3%	0%	0%	–	0%	0%	0%	29%
kqkbn	27%	1%	0%	1%	0%	0%	0%	0%	29%
kkbbn	–	–	–	0%	18%	–	0%	10%	29%
knkrb	–	–	0%	2%	23%	–	0%	3%	29%
knkqr	–	–	–	0%	17%	–	0%	11%	29%
knkrr	–	–	–	1%	21%	–	0%	6%	29%
kbnkp	22%	3%	1%	1%	1%	0%	0%	0%	28%
knpkp	18%	5%	1%	1%	3%	0%	0%	0%	28%
kqknn	25%	3%	0%	0%	–	0%	0%	–	28%
kqkbb	25%	1%	1%	0%	0%	1%	0%	0%	28%
krpkn	26%	2%	0%	–	–	0%	0%	–	28%
knnkq	–	0%	0%	8%	17%	–	0%	2%	28%
kppkr	4%	5%	1%	3%	12%	0%	0%	1%	28%
kqkqr	0%	0%	1%	4%	16%	0%	0%	6%	28%
kbkrr	–	–	–	3%	19%	–	0%	5%	27%

continued on next page

	WWW $d_1 < d_2$	WWD	WWL	DDL	LLL $d_1 < d_2$	WW	DD	LL	Sum
krpkq	1%	2%	3%	2%	16%	0%	0%	2%	27%
kqrkr	25%	1%	0%	0%	0%	1%	0%	0%	27%
krnkb	26%	1%	–	0%	–	0%	0%	–	27%
kpkbn	1%	1%	0%	4%	19%	0%	0%	2%	27%
kqpkr	24%	1%	1%	0%	0%	1%	0%	0%	27%
knkqb	–	–	–	1%	20%	–	0%	6%	27%
krnkn	26%	1%	–	–	–	0%	0%	–	27%
krbkb	25%	1%	–	–	–	0%	0%	0%	27%
knpkq	0%	0%	1%	2%	20%	0%	0%	3%	27%
kbnkq	0%	1%	3%	1%	20%	0%	0%	2%	27%
krpkp	24%	1%	2%	0%	0%	0%	0%	0%	27%
knppk	26%	0%	–	–	–	0%	0%	–	26%
krrkp	26%	0%	0%	0%	–	0%	0%	–	26%
kknnn	–	–	–	3%	18%	–	0%	6%	26%
kbpkp	19%	4%	1%	1%	1%	0%	0%	0%	26%
knkqn	–	–	–	1%	21%	–	0%	5%	26%
kkbnn	–	–	–	0%	19%	–	0%	8%	26%
knkrp	–	0%	0%	4%	21%	–	0%	1%	26%
krrkn	26%	0%	–	–	–	0%	0%	–	26%
kbkrn	–	–	–	5%	19%	–	0%	2%	26%
kpkqp	0%	0%	0%	0%	17%	0%	0%	8%	26%
kbkqr	–	–	–	1%	16%	–	0%	9%	26%
krbkn	26%	1%	–	–	–	0%	0%	–	26%
krkpp	16%	3%	2%	2%	2%	0%	0%	0%	26%
kqkrr	4%	11%	1%	4%	3%	0%	1%	1%	26%
kqqkr	25%	0%	0%	0%	0%	0%	0%	0%	26%
kqnkr	23%	2%	0%	0%	0%	1%	0%	0%	26%
kbbkq	0%	1%	2%	4%	17%	0%	0%	2%	26%
kbkrb	–	–	–	4%	19%	–	0%	3%	26%
kkbbp	–	–	–	1%	16%	–	0%	8%	25%
krrkb	25%	1%	–	–	–	0%	0%	–	25%
kqbkr	23%	2%	0%	0%	0%	1%	0%	0%	25%
kqknp	24%	1%	0%	0%	0%	0%	0%	0%	25%
kkbbb	–	–	–	0%	12%	–	2%	10%	25%
krkqr	–	–	0%	1%	15%	–	0%	9%	25%
kppkn	17%	8%	0%	0%	0%	0%	0%	0%	25%
krkqb	–	0%	0%	2%	18%	0%	0%	5%	25%
krrrk	25%	0%	–	–	–	–	–	–	25%
kqkbp	23%	1%	0%	0%	0%	0%	0%	0%	25%
krrpk	24%	0%	–	–	–	–	–	–	25%
kkbnp	–	–	–	0%	18%	–	0%	6%	24%
knkqp	–	–	0%	0%	19%	0%	0%	5%	24%

continued on next page

	WWW $d_1 < d_2$	WWD	WWL	DDL	LLL $d_1 < d_2$	WW	DD	LL	Sum
kppkq	1%	0%	1%	1%	17%	0%	0%	4%	24%
kbkrp	–	–	–	6%	17%	–	0%	1%	24%
kpkrp	0%	1%	0%	1%	19%	0%	0%	4%	24%
kpkpp	5%	2%	2%	3%	11%	0%	0%	0%	24%
kbkqb	–	–	–	1%	18%	–	0%	5%	24%
kbpkq	0%	0%	2%	2%	17%	0%	0%	3%	24%
krrnk	24%	0%	–	–	–	–	–	–	24%
kpknp	3%	2%	0%	5%	12%	0%	0%	1%	24%
kbppk	24%	0%	–	–	–	0%	0%	–	24%
krkqp	0%	0%	1%	0%	19%	0%	0%	3%	24%
kbkqn	–	–	–	1%	18%	–	0%	4%	24%
kknnp	–	–	–	2%	17%	–	0%	4%	23%
krnkq	1%	3%	3%	6%	7%	0%	0%	2%	23%
kkbpp	–	–	–	0%	18%	–	0%	5%	23%
krkqn	–	0%	0%	2%	18%	0%	0%	4%	23%
krbkq	1%	6%	3%	7%	4%	0%	0%	1%	23%
kqpkb	22%	1%	–	–	–	0%	0%	0%	23%
kqpkn	23%	0%	0%	–	–	0%	0%	–	23%
kqkpp	22%	1%	0%	0%	0%	0%	0%	0%	23%
krnkp	21%	2%	0%	0%	0%	0%	0%	0%	23%
krrbk	23%	0%	–	–	–	–	–	–	23%
knnnk	19%	3%	–	–	–	–	–	–	23%
kqrkb	22%	0%	–	–	–	0%	0%	–	23%
kbkqp	–	–	–	1%	18%	–	0%	4%	22%
kqrkn	22%	0%	–	–	–	0%	0%	–	22%
kqpkp	22%	0%	0%	0%	–	0%	0%	–	22%
kpkbp	2%	1%	0%	4%	13%	0%	0%	1%	22%
kqbkb	21%	1%	–	–	–	0%	0%	0%	22%
kqnkp	22%	0%	0%	0%	–	0%	0%	–	22%
kqnkb	21%	1%	0%	–	–	0%	0%	–	22%
knnpk	20%	2%	–	–	–	–	–	–	22%
krbkp	20%	2%	0%	0%	0%	0%	0%	0%	22%
kqpkq	9%	7%	2%	1%	0%	1%	1%	0%	22%
kknpp	–	–	–	0%	18%	–	0%	3%	22%
kqnkn	22%	0%	–	–	–	0%	0%	–	22%
krpkr	13%	7%	1%	0%	0%	0%	1%	0%	22%
kbnpk	22%	0%	–	–	–	0%	–	–	22%
kqrkp	22%	0%	0%	0%	–	0%	0%	–	22%
krppk	22%	0%	–	–	–	0%	–	–	22%
kqqkb	21%	0%	–	–	–	0%	–	–	21%
kqrpk	21%	0%	–	–	–	–	–	–	21%
kqbkn	21%	0%	–	–	–	0%	0%	–	21%

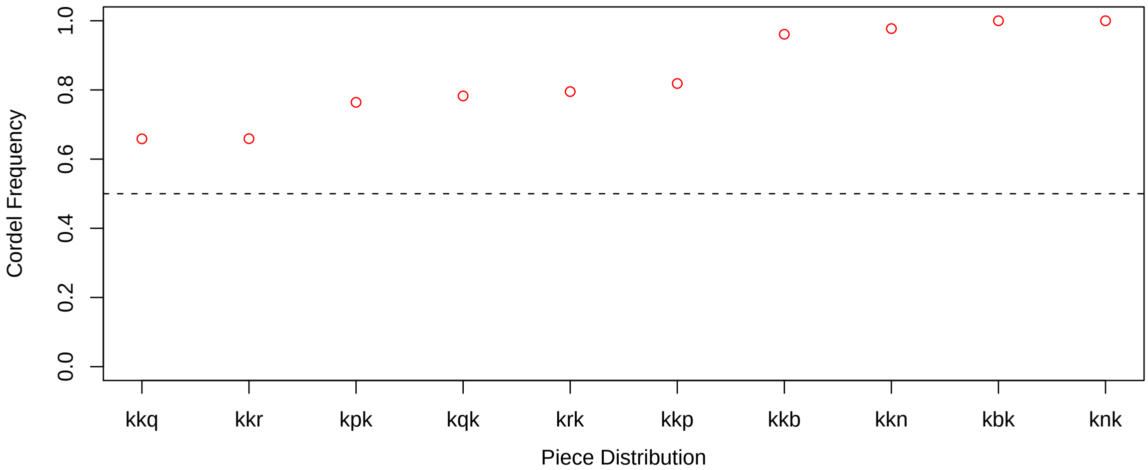
continued on next page

	WWW $d_1 < d_2$	WWD	WWL	DDL	LLL $d_1 < d_2$	WW	DD	LL	Sum
kqbkp	21%	0%	0%	0%	–	0%	0%	–	21%
kqppk	21%	0%	–	–	–	0%	–	–	21%
kbnnk	21%	0%	–	–	–	–	–	–	21%
kqrrk	20%	0%	–	–	–	–	–	–	20%
kqnpk	20%	0%	–	–	–	–	–	–	20%
kbbnk	20%	0%	–	–	–	–	–	–	20%
kqqkn	20%	0%	–	–	–	0%	0%	–	20%
krnpk	20%	0%	–	–	–	0%	–	–	20%
kqrnk	20%	0%	–	–	–	–	–	–	20%
kppkb	11%	8%	0%	0%	0%	0%	1%	0%	20%
kqbpk	20%	0%	–	–	–	–	–	–	20%
kkppp	–	–	–	0%	17%	–	0%	2%	19%
kqrbk	19%	0%	–	–	–	–	–	–	19%
kbbpk	19%	1%	–	–	–	–	–	–	19%
kqnnk	19%	0%	–	–	–	–	–	–	19%
krbpk	19%	0%	–	–	–	0%	–	–	19%
krnnk	19%	0%	–	–	–	–	–	–	19%
kqbnk	18%	0%	–	–	–	–	–	–	18%
kqbkq	4%	10%	2%	1%	0%	0%	1%	0%	18%
krbnk	18%	0%	–	–	–	–	–	–	18%
kbpkn	12%	6%	0%	0%	0%	0%	0%	–	18%
krbbk	18%	0%	–	–	–	–	–	–	18%
kqqkp	17%	0%	0%	0%	–	0%	0%	–	17%
knkbb	–	–	–	3%	13%	–	1%	1%	17%
kqbbk	17%	0%	–	–	–	–	–	–	17%
kqqpk	17%	0%	–	–	–	–	–	–	17%
kqqnk	16%	0%	–	–	–	–	–	–	16%
kbbkn	16%	0%	–	–	–	0%	0%	–	16%
krkrp	0%	1%	1%	5%	8%	0%	0%	1%	16%
kqqbk	16%	–	–	–	–	–	–	–	16%
knkpp	0%	0%	0%	6%	10%	–	0%	0%	16%
kpkbb	1%	1%	0%	1%	9%	0%	1%	2%	16%
knpkn	9%	6%	0%	0%	0%	0%	0%	0%	15%
kqkqp	0%	1%	2%	5%	5%	0%	1%	1%	15%
kbbkp	10%	1%	1%	2%	0%	0%	0%	0%	14%
kqnkq	2%	7%	2%	1%	0%	0%	1%	0%	14%
krbkr	3%	11%	0%	0%	0%	0%	1%	0%	14%
kqqrk	14%	–	–	–	–	–	–	–	14%
kpknn	1%	2%	0%	5%	5%	0%	1%	0%	14%
knnkp	7%	4%	0%	2%	1%	0%	0%	0%	14%
knkbp	0%	0%	0%	7%	7%	–	0%	0%	14%
kbbbk	13%	0%	–	–	–	–	–	–	13%

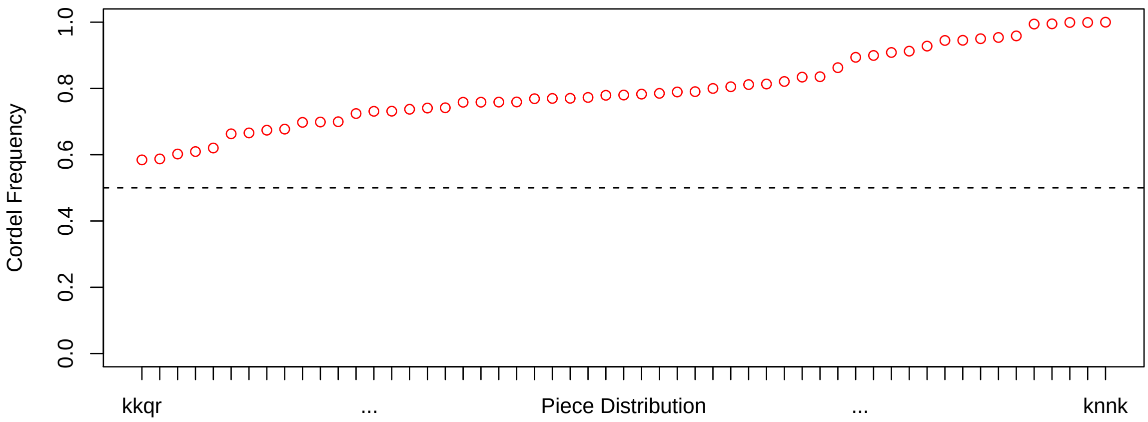
continued on next page

	WWW $d_1 < d_2$	WWD	WWL	DDL	LLL $d_1 < d_2$	WW	DD	LL	Sum
kbpkb	6%	6%	0%	0%	0%	0%	0%	–	13%
krknp	3%	7%	0%	3%	1%	0%	0%	0%	13%
knpkr	2%	4%	0%	4%	1%	0%	1%	0%	12%
knpkb	6%	6%	0%	0%	0%	0%	0%	0%	12%
knknp	0%	0%	0%	5%	6%	–	0%	0%	12%
kqqqk	11%	–	–	–	–	–	–	–	11%
krnkr	2%	8%	0%	0%	0%	0%	1%	0%	11%
kqkqb	0%	1%	1%	4%	2%	0%	1%	1%	11%
kbkpp	0%	0%	0%	5%	5%	–	0%	0%	11%
kbpkr	2%	5%	0%	1%	1%	0%	1%	0%	11%
krkbp	0%	5%	0%	4%	1%	0%	0%	0%	10%
kbnkn	3%	7%	–	0%	–	0%	0%	–	10%
kqkqn	0%	1%	1%	3%	1%	0%	1%	1%	9%
kbkbp	–	0%	0%	5%	3%	–	0%	0%	8%
knkbn	–	–	–	6%	1%	–	1%	0%	8%
krkrb	0%	0%	0%	4%	1%	0%	2%	1%	7%
kbknp	0%	0%	0%	4%	3%	–	0%	0%	7%
kbnkr	0%	5%	0%	0%	0%	0%	1%	0%	6%
krkrn	0%	0%	0%	3%	1%	0%	2%	0%	6%
kbbkr	0%	4%	0%	0%	0%	0%	1%	0%	5%
kbnkb	0%	5%	–	0%	–	0%	0%	0%	5%
kbbkb	0%	4%	–	0%	–	0%	0%	–	5%
kbkbb	–	0%	–	2%	0%	–	1%	0%	4%
kbkbn	–	–	–	3%	0%	–	1%	0%	4%
krkbn	0%	0%	0%	1%	0%	0%	0%	0%	3%
krkbb	0%	0%	0%	1%	0%	0%	1%	0%	3%
knnkr	–	0%	–	1%	0%	–	1%	0%	2%
krknn	0%	1%	0%	0%	–	0%	0%	0%	2%
knknn	–	–	–	0%	0%	–	0%	0%	0%
knnkb	0%	0%	–	0%	–	–	0%	–	0%
kbknn	–	–	–	0%	0%	–	0%	0%	0%
knnkn	0%	0%	–	0%	–	–	0%	–	0%
Mean	9%	1%	0%	1%	7%	0%	0%	4%	23%

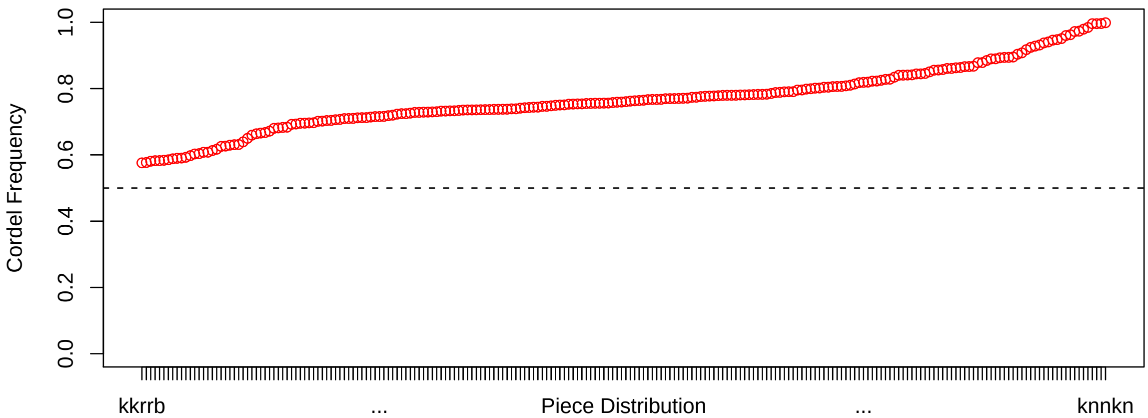
D.4 Figures



(a) Endgames with 3 pieces (including kings)

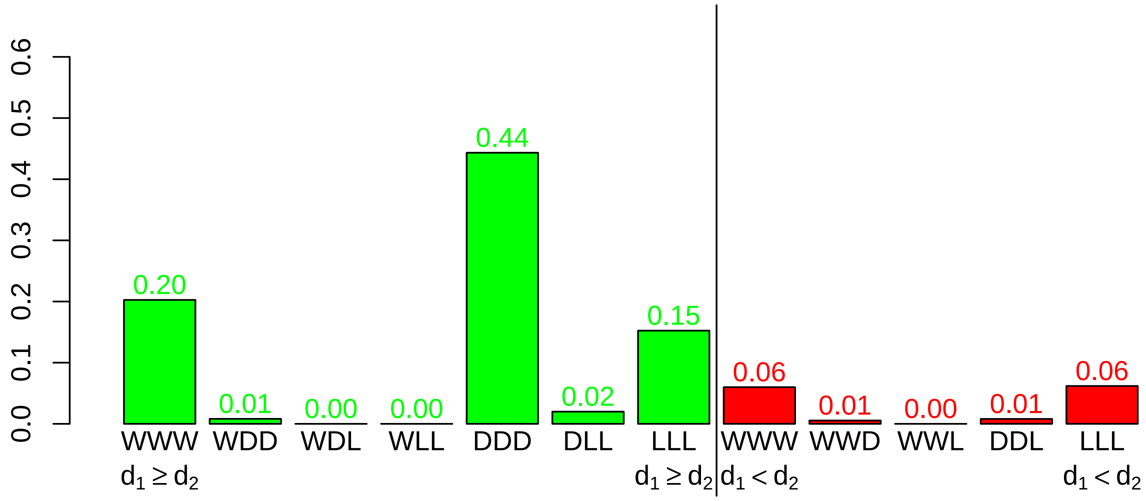


(b) Endgames with 4 pieces (including kings)

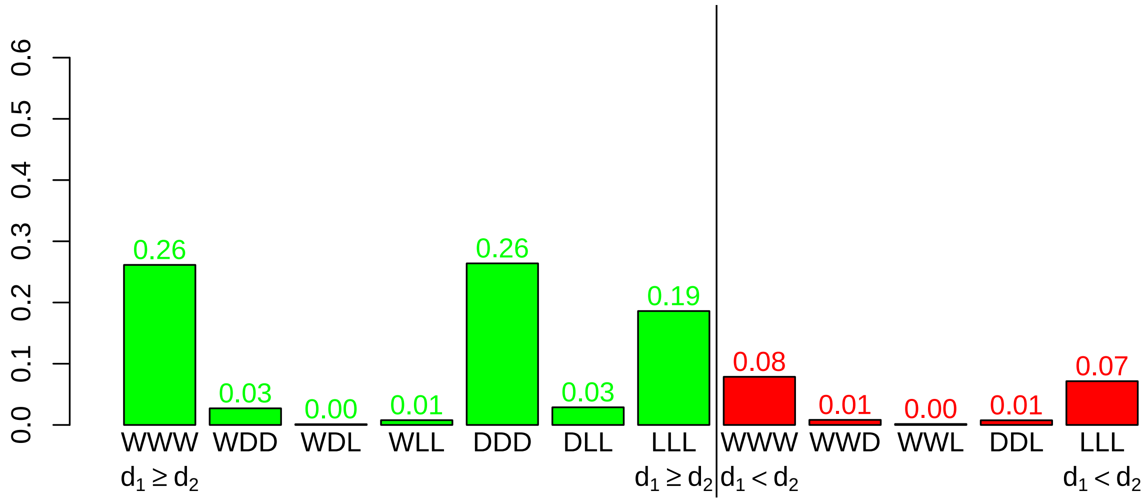


(c) Endgames with 5 pieces (including kings)

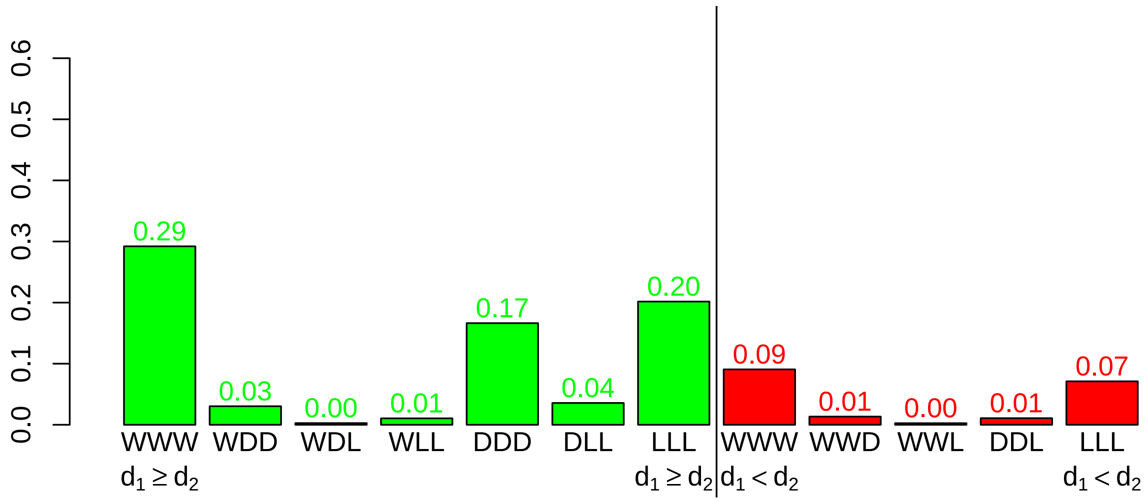
Figure D.4.1: Cordel Frequencies for different piece configurations



(a) Endgames with 3 pieces (including kings)

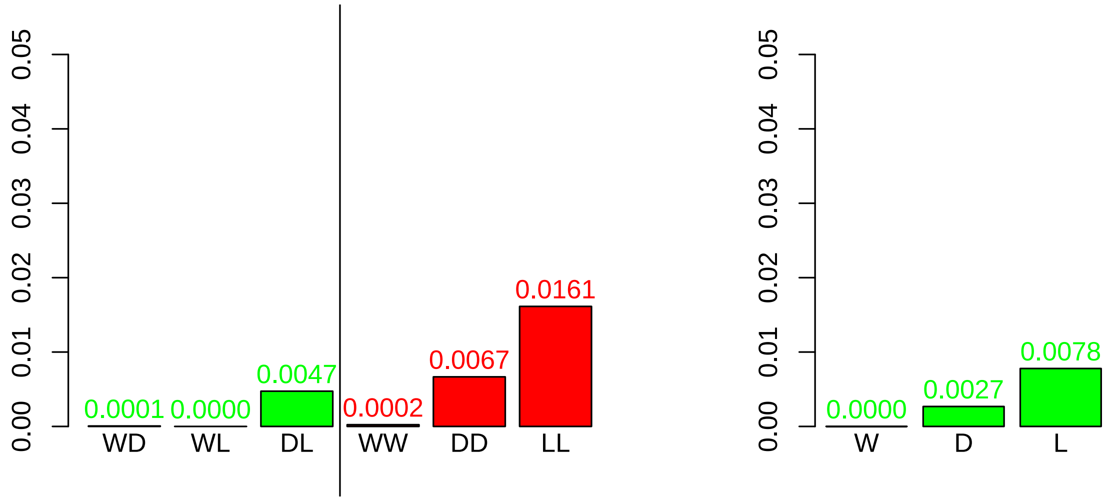


(b) Endgames with 4 pieces (including kings)

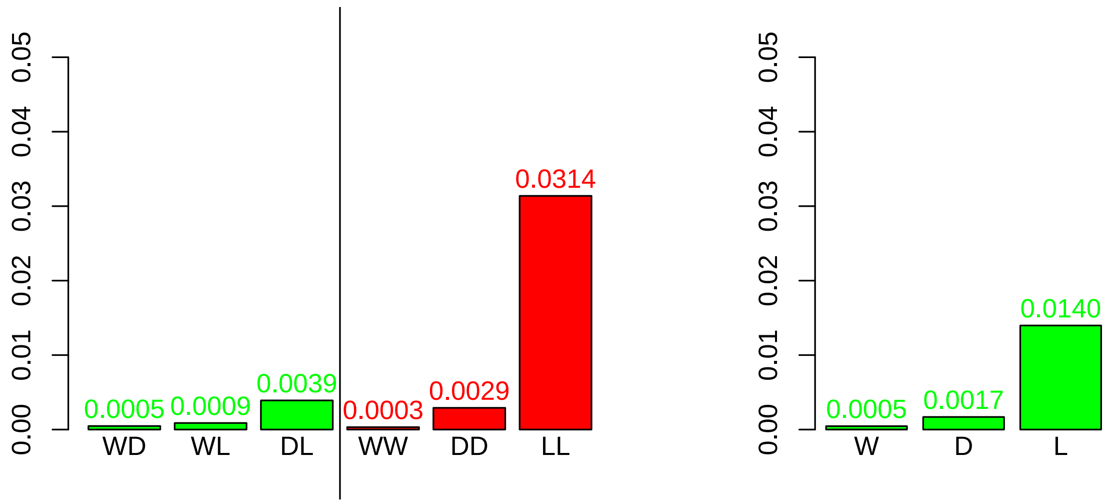


(c) Endgames with 5 pieces (including kings)

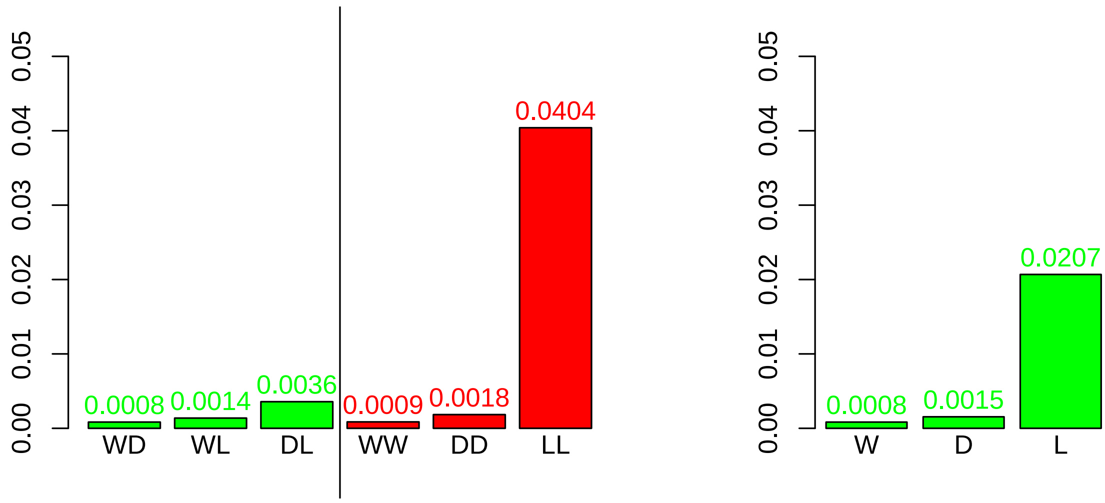
Figure D.4.2: Relative frequencies of different types of chess positions in endgames with 3, 4, or 5 pieces (including kings) with at least three feasible moves.



(a) Endgames with 3 pieces (including kings)



(b) Endgames with 4 pieces (including kings)



(c) Endgames with 5 pieces (including kings)

Figure D.4.3: Relative frequencies of different types of chess positions in endgames with 3, 4, or 5 pieces (including kings) with only two (left) or one (right) feasible moves.

D.5 Monte Carlo Samples for Endgames with 6 Pieces

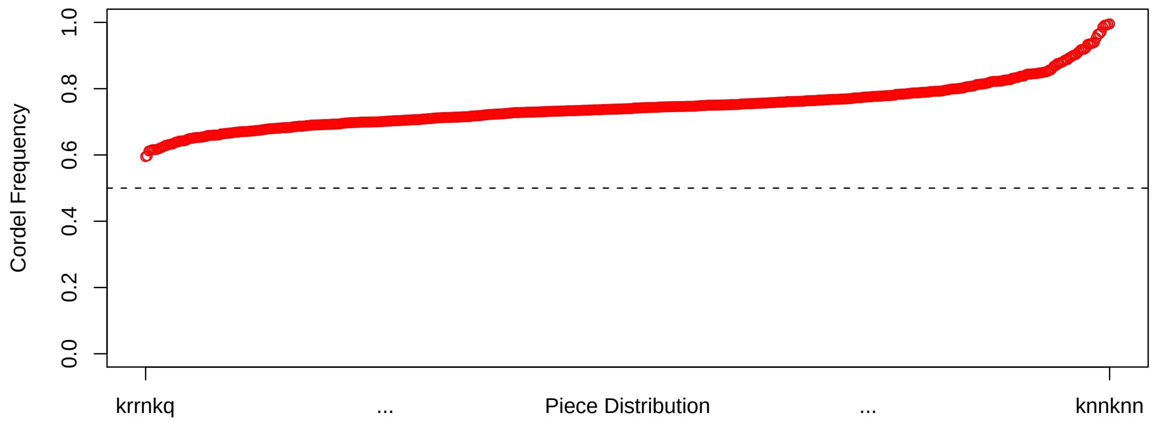
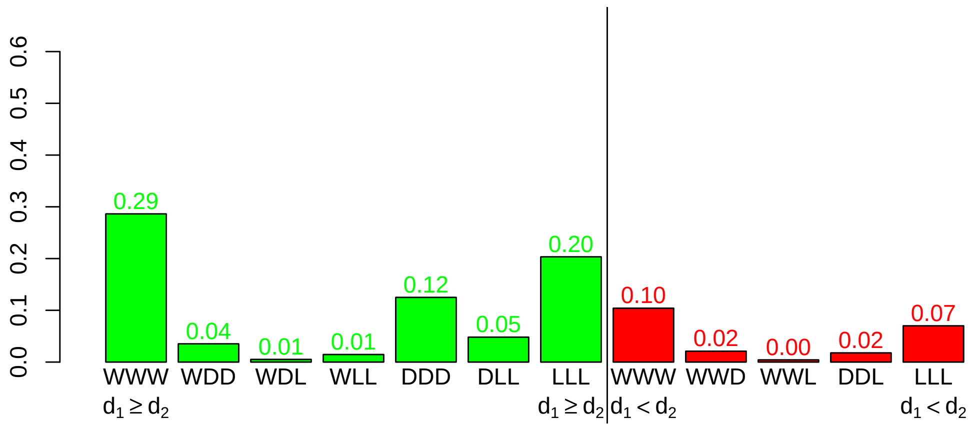
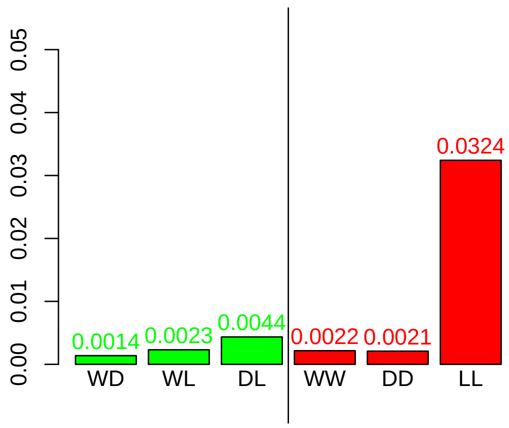


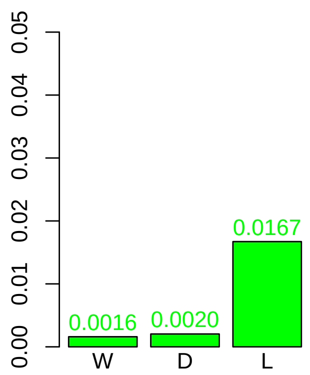
Figure D.5.1: Cordel Frequencies for the 515 different piece configurations.



(a) Endgames with at least 3 feasible moves.



(b) Endgames with exactly 2 feasible moves.



(c) Endgames with exactly 1 feasible move.

Figure D.5.2: Relative frequencies of different types of chess positions in endgames with 6 pieces (including kings) with only two (left) or one (right) feasible moves.

Appendix E

Detailed Results for Chess – Best Move Per Piece Rule

Refers to page 16.

Overview of the results presented in this chapter:

E.1 Endgames with 3 Pieces

Cordel Instances

	WWW $d_1 \geq d_2$	WDD	WDL	WLL	DDD	DLL	LLL $d_1 \geq d_2$	WD	WL	DL	W	D	L	Sum
knk	–	–	–	–	–	–	–	–	–	–	–	0%	–	0%
kbk	–	–	–	–	–	–	–	–	–	–	–	0%	–	0%
kqk	–	–	–	–	–	–	–	0%	–	–	0%	–	–	0%
krk	–	–	–	–	–	–	–	5%	–	–	0%	–	–	5%
kpk	–	–	–	–	–	–	–	15%	–	–	2%	2%	–	18%
kkb	–	–	–	–	–	–	–	–	–	–	–	100%	–	100%
kkn	–	–	–	–	–	–	–	–	–	–	–	100%	–	100%
kkp	–	–	–	–	–	–	–	–	–	–	–	42%	58%	100%
kkq	–	–	–	–	–	–	–	–	–	–	–	10%	90%	100%
kkr	–	–	–	–	–	–	–	–	–	–	–	10%	90%	100%
Mean	–	–	–	–	–	–	–	2%	–	–	0%	26%	24%	52%

Non-Cordel Instances

	WWW $d_1 < d_2$	WWD	WWL	DDL	LLL $d_1 < d_2$	WW	DD	LL	Sum
knk	–	–	–	–	–	–	100%	–	100%
kbk	–	–	–	–	–	–	100%	–	100%
kqk	–	–	–	–	–	100%	–	–	100%
krk	–	–	–	–	–	95%	–	–	95%
kpk	–	–	–	–	–	60%	22%	–	82%
kkb	–	–	–	–	–	–	–	–	–
kkn	–	–	–	–	–	–	–	–	–
kkp	–	–	–	–	–	–	–	–	–
kkq	–	–	–	–	–	–	–	–	–
kkr	–	–	–	–	–	–	–	–	–
Mean	–	–	–	–	–	25%	22%	–	48%

E.2 Endgames with 4 Pieces

Cordel Instances

	WWW	WDD	WDL	WLL	DDD	DLL	LLL	WD	WL	DL	W	D	L	Sum
	$d_1 \geq d_2$						$d_1 \geq d_2$							
kbkn	–	–	–	–	–	–	–	0%	–	–	–	7%	–	7%
knkn	–	–	–	–	–	–	–	0%	–	–	–	8%	–	8%
kbkb	–	–	–	–	–	–	–	0%	–	–	–	11%	–	11%
knkb	–	–	–	–	–	–	–	0%	–	–	–	11%	–	11%
kqkn	–	–	–	–	–	–	–	7%	–	–	4%	1%	–	12%
kqkp	–	–	–	–	–	–	–	9%	1%	0%	2%	0%	0%	12%
kbkp	–	–	–	–	–	–	–	0%	–	13%	–	3%	0%	16%
knkp	–	–	–	–	–	–	–	0%	0%	15%	0%	3%	0%	17%
kqkb	–	–	–	–	–	–	–	16%	–	–	6%	0%	–	22%
krkp	–	–	–	–	–	–	–	8%	9%	2%	2%	0%	0%	22%
kpkn	–	–	–	–	–	–	–	14%	0%	0%	2%	10%	0%	27%
kpkb	–	–	–	–	–	–	–	13%	–	0%	4%	13%	–	30%
kpkp	–	–	–	–	–	–	–	11%	3%	9%	3%	3%	2%	32%
kbkr	–	–	–	–	–	–	–	–	–	19%	–	11%	2%	33%
krkn	–	–	–	–	–	–	–	26%	–	–	2%	5%	–	33%
krkb	–	–	–	–	–	–	–	26%	–	–	3%	4%	–	33%
kqkr	–	–	–	–	–	–	–	9%	19%	0%	7%	0%	0%	36%
kbkq	–	–	–	–	–	–	–	–	–	13%	–	9%	15%	37%
knkq	–	–	–	–	–	–	–	–	–	10%	–	9%	19%	37%
knkr	–	–	–	–	–	–	–	0%	0%	22%	–	13%	4%	39%
krkr	–	–	–	–	–	–	–	7%	14%	7%	4%	8%	0%	41%
kpkr	–	–	–	–	–	–	–	4%	8%	7%	3%	5%	16%	44%
krkq	–	–	–	–	–	–	–	0%	19%	5%	7%	1%	13%	45%
kpkq	–	–	–	–	–	–	–	0%	2%	7%	6%	4%	27%	46%
kppk	42%	2%	–	–	1%	–	–	1%	–	–	0%	0%	–	46%
kqqk	47%	–	–	–	–	–	–	–	–	–	–	–	–	47%
kqkq	–	–	–	–	–	–	–	9%	23%	8%	7%	7%	0%	54%
krrk	55%	0%	–	–	–	–	–	–	–	–	–	–	–	55%
kqrk	57%	0%	–	–	–	–	–	–	–	–	–	–	–	57%
kqpk	58%	0%	–	–	–	–	–	0%	–	–	0%	–	–	59%
knpk	49%	7%	–	–	3%	–	–	1%	–	–	0%	0%	–	59%
kbpk	52%	5%	–	–	3%	–	–	1%	–	–	0%	0%	–	60%
kbnk	52%	9%	–	–	0%	–	–	0%	–	–	–	–	–	62%
krpk	61%	0%	–	–	–	–	–	0%	–	–	0%	–	–	62%
krnk	63%	3%	–	–	–	–	–	–	–	–	–	–	–	66%
krbk	64%	3%	–	–	–	–	–	0%	–	–	–	–	–	67%
kqnk	72%	0%	–	–	–	–	–	–	–	–	–	–	–	73%

continued on next page

	WWW $d_1 \geq d_2$	WDD	WDL	WLL	DDD	DLL	LLL $d_1 \geq d_2$	WD	WL	DL	W	D	L	Sum
kqbk	74%	0%	–	–	–	–	–	0%	–	–	–	–	–	75%
kbbk	29%	5%	–	–	50%	–	–	0%	–	–	–	–	–	84%
knnk	–	0%	–	–	100%	–	–	–	–	–	–	–	–	100%
kkbb	–	–	–	–	–	–	–	–	–	–	–	59%	41%	100%
kkbn	–	–	–	–	–	–	–	–	–	–	–	18%	82%	100%
kkbp	–	–	–	–	–	–	–	–	–	–	–	17%	83%	100%
kknn	–	–	–	–	–	–	–	–	–	–	–	100%	–	100%
kknp	–	–	–	–	–	–	–	–	–	–	–	18%	82%	100%
kkpp	–	–	–	–	–	–	–	–	–	–	–	8%	92%	100%
kkqb	–	–	–	–	–	–	–	–	–	–	–	9%	91%	100%
kkqn	–	–	–	–	–	–	–	–	–	–	–	9%	91%	100%
kkqp	–	–	–	–	–	–	–	–	–	–	–	2%	98%	100%
kkqq	–	–	–	–	–	–	–	–	–	–	–	–	100%	100%
kkqr	–	–	–	–	–	–	–	–	–	–	–	–	100%	100%
kkrb	–	–	–	–	–	–	–	–	–	–	–	9%	91%	100%
kkrn	–	–	–	–	–	–	–	–	–	–	–	9%	91%	100%
kkrp	–	–	–	–	–	–	–	–	–	–	–	2%	98%	100%
kkrr	–	–	–	–	–	–	–	–	–	–	–	–	100%	100%
Mean	14%	1%	–	–	3%	–	–	3%	2%	3%	1%	7%	24%	58%

Non-Cordel Instances

	WWW $d_1 < d_2$	WWD	WWL	DDL	LLL $d_1 < d_2$	WW	DD	LL	Sum
kbkn	–	–	–	–	–	–	93%	–	93%
knkn	–	–	–	–	–	–	92%	–	92%
kbkb	–	–	–	–	–	–	89%	–	89%
knkb	–	–	–	–	–	–	89%	–	89%
kqkn	–	–	–	–	–	88%	0%	–	88%
kqkp	–	–	–	–	–	87%	1%	–	88%
kbkp	–	–	–	–	–	–	79%	5%	84%
knkp	–	–	–	–	–	0%	70%	13%	83%
kqkb	–	–	–	–	–	78%	0%	–	78%
krkp	–	–	–	–	–	72%	6%	0%	78%
kpkn	–	–	–	–	–	16%	57%	0%	73%
kpkb	–	–	–	–	–	7%	63%	–	70%
kpkp	–	–	–	–	–	26%	21%	22%	68%
kbkr	–	–	–	–	–	–	66%	1%	67%
krkn	–	–	–	–	–	20%	47%	–	67%
krkb	–	–	–	–	–	6%	61%	–	67%
kqkr	–	–	–	–	–	64%	0%	–	64%

continued on next page

	WWW $d_1 < d_2$	WWD	WWL	DDL	LLL $d_1 < d_2$	WW	DD	LL	Sum
kbkq	–	–	–	–	–	–	1%	61%	63%
knkq	–	–	–	–	–	–	1%	62%	63%
knkr	–	–	–	–	–	–	54%	7%	61%
krkr	–	–	–	–	–	4%	55%	0%	59%
kpkr	–	–	–	–	–	1%	5%	50%	56%
krkq	–	–	–	–	–	2%	0%	53%	55%
kpkq	–	–	–	–	–	0%	1%	53%	54%
kppk	41%	4%	–	–	–	8%	1%	–	54%
kqqk	50%	4%	–	–	–	0%	–	–	53%
kqkq	–	–	–	–	–	3%	43%	0%	46%
krrk	45%	0%	–	–	–	0%	–	–	45%
kqrk	42%	1%	–	–	–	0%	–	–	43%
kqpk	36%	0%	–	–	–	5%	–	–	41%
knpk	28%	10%	–	–	–	3%	1%	–	41%
kbpk	28%	8%	–	–	–	4%	1%	–	40%
kbnk	32%	6%	–	–	–	0%	0%	–	38%
krpk	32%	1%	–	–	–	5%	–	–	38%
krnk	31%	2%	–	–	–	0%	–	–	34%
krbk	30%	2%	–	–	–	0%	–	–	33%
kqnk	27%	1%	–	–	–	0%	–	–	27%
kqbk	24%	1%	–	–	–	0%	–	–	25%
kbbk	14%	1%	–	–	–	0%	0%	–	16%
knnk	–	–	–	–	–	–	0%	–	0%
kkbb	–	–	–	–	–	–	–	–	–
kkbn	–	–	–	–	–	–	–	–	–
kkbp	–	–	–	–	–	–	–	–	–
kknn	–	–	–	–	–	–	–	–	–
kknp	–	–	–	–	–	–	–	–	–
kkpp	–	–	–	–	–	–	–	–	–
kkqb	–	–	–	–	–	–	–	–	–
kkqn	–	–	–	–	–	–	–	–	–
kkqp	–	–	–	–	–	–	–	–	–
kkqq	–	–	–	–	–	–	–	–	–
kkqr	–	–	–	–	–	–	–	–	–
kkrb	–	–	–	–	–	–	–	–	–
kkrn	–	–	–	–	–	–	–	–	–
kkrp	–	–	–	–	–	–	–	–	–
kkrr	–	–	–	–	–	–	–	–	–
Mean	8%	1%	–	–	–	9%	18%	6%	42%

E.3 Endgames with 5 Pieces

Cordel Instances

	WWW $d_1 \geq d_2$	WDD	WDL	WLL	DDD	DLL	LLL $d_1 \geq d_2$	WD	WL	DL	W	D	L	Sum
kbknn	–	–	–	–	–	–	–	0%	–	0%	–	14%	0%	14%
knknn	–	–	–	–	–	–	–	0%	–	0%	–	15%	0%	15%
krknn	–	–	–	–	–	–	–	5%	0%	0%	0%	12%	0%	18%
kqqkq	8%	3%	0%	0%	0%	0%	0%	4%	0%	0%	7%	0%	0%	22%
kqkpp	–	–	–	–	–	–	–	15%	5%	0%	3%	0%	0%	23%
kqknp	–	–	–	–	–	–	–	10%	6%	0%	6%	1%	0%	23%
knkpp	–	–	–	–	–	–	–	0%	0%	19%	0%	4%	1%	24%
kbkpp	–	–	–	–	–	–	–	0%	0%	20%	0%	4%	1%	25%
krkbb	–	–	–	–	–	–	–	1%	0%	9%	0%	14%	1%	25%
kbknp	–	–	–	–	–	–	–	0%	0%	17%	0%	7%	2%	26%
knknp	–	–	–	–	–	–	–	0%	0%	17%	0%	7%	3%	27%
kqrkq	10%	1%	1%	1%	0%	0%	0%	2%	4%	0%	8%	2%	0%	29%
knkbb	–	–	–	–	–	–	–	0%	0%	9%	–	14%	6%	30%
kbkbb	–	–	–	–	–	–	–	0%	0%	9%	0%	19%	1%	30%
krkbn	–	–	–	–	–	–	–	3%	0%	13%	0%	12%	1%	30%
kqqkr	26%	0%	0%	–	0%	–	–	0%	0%	0%	4%	0%	0%	30%
kqknn	–	–	–	–	–	–	–	21%	–	0%	8%	2%	–	31%
kbkbn	–	–	–	–	–	–	–	0%	0%	14%	0%	16%	1%	31%
kqkbp	–	–	–	–	–	–	–	12%	11%	1%	7%	1%	0%	32%
kqkbn	–	–	–	–	–	–	–	9%	11%	0%	9%	1%	1%	32%
knkbp	–	–	–	–	–	–	–	0%	0%	19%	0%	9%	4%	32%
kbkbp	–	–	–	–	–	–	–	0%	0%	19%	0%	10%	2%	32%
krkpp	–	–	–	–	–	–	–	7%	14%	8%	3%	0%	1%	32%
kpkpp	–	–	–	–	–	–	–	6%	5%	10%	2%	3%	7%	33%
knkrp	–	–	–	–	–	–	–	0%	0%	15%	0%	5%	13%	33%
knkbn	–	–	–	–	–	–	–	0%	0%	17%	–	15%	2%	35%
knkqp	–	–	–	–	–	–	–	0%	0%	6%	0%	6%	23%	35%
kbkqp	–	–	–	–	–	–	–	0%	0%	9%	0%	7%	19%	35%
kbkrp	–	–	–	–	–	–	–	0%	0%	20%	0%	6%	10%	35%
kpknn	–	–	–	–	–	–	–	8%	0%	9%	1%	15%	4%	37%
kpknp	–	–	–	–	–	–	–	6%	5%	12%	2%	6%	9%	38%
kpkrp	–	–	–	–	–	–	–	1%	7%	4%	1%	2%	23%	39%
knkrn	–	–	–	–	–	–	–	0%	0%	16%	0%	5%	17%	39%
kqkbb	–	–	–	–	–	–	–	19%	8%	0%	10%	2%	0%	39%
knkrr	–	–	–	–	–	–	–	–	0%	9%	0%	5%	26%	40%
kbkrr	–	–	–	–	–	–	–	–	–	15%	–	6%	20%	40%
krkqq	–	–	–	–	–	–	–	–	0%	0%	0%	0%	40%	40%

continued on next page

	WWW $d_1 \geq d_2$	WDD	WDL	WLL	DDD	DLL	LLL $d_1 \geq d_2$	WD	WL	DL	W	D	L	Sum
knkrb	–	–	–	–	–	–	–	0%	0%	15%	0%	7%	19%	40%
kpkbp	–	–	–	–	–	–	–	4%	6%	10%	2%	6%	12%	40%
krkqp	–	–	–	–	–	–	–	0%	13%	5%	5%	1%	16%	41%
kqrkr	32%	0%	0%	0%	0%	–	–	2%	0%	0%	6%	0%	0%	41%
kbkqn	–	–	–	–	–	–	–	0%	0%	12%	0%	8%	21%	41%
kbkrn	–	–	–	–	–	–	–	0%	0%	22%	–	5%	14%	41%
kbkqr	–	–	–	–	–	–	–	–	–	6%	–	5%	30%	41%
knkqn	–	–	–	–	–	–	–	0%	0%	9%	–	8%	24%	42%
kpkbb	–	–	–	–	–	–	–	6%	3%	5%	1%	18%	9%	42%
krknp	–	–	–	–	–	–	–	18%	2%	14%	1%	6%	1%	42%
kbkrb	–	–	–	–	–	–	–	0%	0%	20%	0%	7%	16%	43%
knkqr	–	–	–	–	–	–	–	–	0%	3%	0%	4%	36%	44%
kpkqp	–	–	–	–	–	–	–	0%	1%	4%	2%	3%	33%	44%
krkrn	–	–	–	–	–	–	–	1%	2%	22%	0%	16%	3%	44%
kbkqb	–	–	–	–	–	–	–	0%	0%	11%	0%	8%	25%	44%
knkqb	–	–	–	–	–	–	–	0%	0%	8%	–	8%	28%	44%
kqqkb	41%	0%	–	–	–	–	–	0%	–	–	3%	–	–	44%
kbkqq	–	–	–	–	–	–	–	–	–	0%	–	0%	44%	44%
kpkrn	–	–	–	–	–	–	–	1%	3%	10%	1%	6%	25%	44%
krrkr	29%	3%	0%	0%	0%	0%	–	4%	0%	0%	8%	0%	0%	44%
krkqr	–	–	–	–	–	–	–	0%	0%	12%	0%	5%	28%	45%
krkbp	–	–	–	–	–	–	–	16%	2%	18%	3%	6%	2%	45%
kpkbn	–	–	–	–	–	–	–	4%	5%	12%	1%	7%	16%	46%
kpkrb	–	–	–	–	–	–	–	0%	2%	10%	0%	6%	28%	46%
kpkrr	–	–	–	–	–	–	–	0%	1%	5%	0%	1%	40%	46%
krkrp	–	–	–	–	–	–	–	3%	12%	17%	3%	7%	5%	47%
kqqkp	46%	0%	–	–	0%	–	–	0%	0%	–	1%	0%	0%	47%
krkqn	–	–	–	–	–	–	–	0%	2%	19%	1%	7%	18%	47%
krkqb	–	–	–	–	–	–	–	0%	1%	19%	0%	7%	20%	47%
kqqkn	44%	0%	–	–	–	–	–	0%	–	–	3%	0%	–	47%
kqkrp	–	–	–	–	–	–	–	10%	25%	3%	8%	1%	1%	47%
krkrb	–	–	–	–	–	–	–	0%	0%	27%	0%	16%	4%	48%
knkqq	–	–	–	–	–	–	–	–	–	0%	–	0%	49%	49%
krkrr	–	–	–	–	–	–	–	0%	0%	24%	0%	6%	19%	49%
kpkqn	–	–	–	–	–	–	–	0%	1%	7%	1%	8%	33%	50%
krrkb	43%	1%	–	–	0%	–	–	1%	–	–	4%	0%	–	50%
krrkn	45%	0%	–	–	0%	–	–	0%	–	–	4%	0%	–	51%
kqrkb	46%	0%	–	–	0%	–	–	1%	–	–	4%	0%	–	51%
krrkq	0%	6%	5%	2%	16%	1%	0%	1%	3%	3%	6%	4%	3%	51%
kpkqr	–	–	–	–	–	–	–	0%	0%	1%	0%	1%	50%	51%
krrkp	49%	0%	0%	0%	0%	–	–	0%	0%	0%	2%	0%	0%	51%
kqqpk	52%	0%	–	–	–	–	–	0%	–	–	0%	–	–	52%

continued on next page

	WWW	WDD	WDL	WLL	DDD	DLL	LLL	WD	WL	DL	W	D	L	Sum
	$d_1 \geq d_2$						$d_1 \geq d_2$							
kpkqb	–	–	–	–	–	–	–	0%	0%	6%	0%	8%	37%	52%
kppkp	23%	6%	2%	3%	3%	3%	6%	2%	0%	1%	3%	0%	0%	53%
krpkb	32%	9%	0%	–	1%	–	–	3%	0%	0%	6%	1%	0%	53%
kqrkn	50%	0%	–	–	–	–	–	0%	–	–	4%	0%	–	53%
krpkn	39%	6%	0%	0%	1%	0%	–	1%	0%	0%	5%	1%	0%	54%
kqqnk	54%	0%	–	–	–	–	–	–	–	–	–	–	–	54%
kqrkp	53%	0%	0%	0%	0%	–	–	0%	0%	–	1%	0%	0%	54%
kpppk	54%	0%	–	–	0%	–	–	0%	–	–	0%	0%	–	55%
kqpkp	50%	2%	0%	1%	0%	–	–	0%	0%	–	2%	0%	0%	55%
kqbkr	38%	7%	0%	0%	0%	0%	–	4%	0%	0%	5%	0%	0%	55%
kqpkn	47%	3%	0%	0%	–	–	–	0%	0%	–	5%	0%	0%	55%
kqkqq	–	–	–	–	–	–	–	0%	1%	23%	0%	8%	23%	55%
kqqbk	57%	0%	–	–	–	–	–	–	–	–	–	–	–	57%
kqpkb	44%	5%	0%	–	0%	–	–	3%	–	–	6%	0%	0%	57%
krpkp	47%	2%	1%	5%	0%	0%	0%	0%	0%	0%	2%	0%	0%	59%
krbkn	46%	7%	–	–	0%	–	–	0%	–	–	5%	1%	–	59%
kpkqq	–	–	–	–	–	–	–	0%	0%	0%	0%	0%	59%	59%
krnkb	40%	11%	–	–	0%	–	–	2%	0%	0%	5%	1%	–	60%
kqkqn	–	–	–	–	–	–	–	6%	19%	14%	6%	10%	3%	60%
knpkp	28%	9%	2%	4%	7%	3%	3%	1%	0%	0%	2%	0%	0%	60%
krbkb	41%	10%	–	–	0%	–	–	2%	–	–	5%	1%	0%	60%
kqpkr	36%	2%	1%	6%	0%	0%	0%	2%	4%	0%	8%	0%	0%	60%
krnkn	46%	8%	–	–	0%	–	–	0%	0%	0%	5%	1%	–	60%
kqnkr	38%	10%	0%	0%	0%	0%	–	5%	1%	0%	6%	0%	0%	60%
kqkqr	–	–	–	–	–	–	–	1%	18%	17%	5%	3%	17%	61%
kbpkp	34%	8%	2%	5%	5%	2%	2%	1%	0%	0%	2%	0%	0%	61%
krbkp	52%	6%	0%	0%	1%	0%	0%	0%	0%	0%	2%	0%	0%	61%
krrnk	62%	0%	–	–	–	–	–	–	–	–	–	–	–	62%
krrpk	62%	0%	–	–	0%	–	–	0%	–	–	0%	–	–	62%
kqrpk	62%	0%	–	–	0%	–	–	0%	–	–	0%	–	–	62%
kbnkp	38%	11%	2%	4%	2%	2%	1%	0%	0%	0%	2%	0%	0%	63%
krnkp	51%	7%	1%	1%	1%	0%	0%	0%	0%	0%	2%	0%	0%	63%
kqbkb	50%	5%	–	–	0%	–	–	3%	–	–	5%	0%	0%	63%
kqkrn	–	–	–	–	–	–	–	28%	18%	4%	7%	5%	2%	63%
kqppk	63%	0%	–	–	0%	–	–	0%	–	–	0%	–	–	64%
krrrk	64%	0%	–	–	–	–	–	–	–	–	–	–	–	64%
kqkqb	–	–	–	–	–	–	–	5%	18%	19%	5%	11%	5%	64%
krrbk	64%	0%	–	–	0%	–	–	–	–	–	–	–	–	64%
kqnkn	55%	4%	–	–	–	–	–	0%	–	–	4%	1%	–	64%
kqkqp	–	–	–	–	–	–	–	3%	24%	21%	6%	6%	4%	64%
kqrnk	64%	0%	–	–	–	–	–	–	–	–	–	–	–	64%
kqrrk	64%	0%	–	–	–	–	–	–	–	–	–	–	–	64%

continued on next page

	WWW $d_1 \geq d_2$	WDD	WDL	WLL	DDD	DLL	LLL $d_1 \geq d_2$	WD	WL	DL	W	D	L	Sum
knppk	65%	0%	–	–	0%	–	–	0%	–	–	0%	0%	–	65%
kqbkn	57%	3%	–	–	0%	–	–	0%	–	–	4%	1%	–	65%
kqrbk	65%	0%	–	–	–	–	–	–	–	–	–	–	–	65%
kqnkb	51%	7%	–	–	0%	–	–	3%	0%	–	5%	0%	–	66%
kppkn	17%	12%	0%	0%	26%	0%	0%	2%	0%	0%	4%	4%	0%	66%
kqnkp	60%	4%	0%	0%	0%	0%	–	0%	0%	0%	2%	0%	0%	66%
krbkq	0%	3%	6%	6%	13%	13%	3%	1%	3%	3%	7%	3%	5%	67%
kqqrk	67%	0%	–	–	–	–	–	–	–	–	–	–	–	67%
kqbkp	63%	2%	0%	0%	0%	–	–	0%	0%	0%	2%	0%	0%	67%
kbppk	67%	0%	–	–	0%	–	–	0%	–	–	0%	0%	–	67%
krpkr	10%	8%	2%	6%	21%	1%	0%	2%	2%	2%	8%	5%	0%	68%
krpkq	0%	2%	3%	13%	1%	4%	18%	1%	5%	1%	8%	1%	10%	68%
kbbkq	–	0%	2%	6%	0%	9%	31%	0%	2%	2%	3%	4%	9%	68%
kbnnk	66%	3%	–	–	0%	–	–	–	–	–	–	–	–	68%
knnnk	56%	11%	–	–	1%	–	–	–	–	–	–	–	–	69%
krnkq	0%	2%	5%	9%	11%	12%	6%	1%	3%	3%	7%	2%	7%	69%
krppk	69%	0%	–	–	0%	–	–	0%	–	–	0%	–	–	69%
knnkq	–	0%	0%	0%	3%	17%	25%	0%	0%	4%	–	9%	12%	69%
kppkr	2%	5%	2%	8%	6%	4%	19%	1%	1%	1%	6%	4%	9%	69%
kbnpk	69%	0%	–	–	0%	–	–	0%	–	–	0%	–	–	69%
kbpkq	0%	0%	1%	7%	1%	5%	28%	0%	3%	1%	7%	2%	13%	69%
knnpk	65%	3%	–	–	2%	–	–	0%	–	–	–	–	–	70%
kbbnk	67%	2%	–	–	0%	–	–	–	–	–	–	–	–	70%
kbnkq	0%	0%	2%	10%	0%	4%	33%	0%	3%	1%	7%	1%	11%	70%
krnpk	71%	0%	–	–	0%	–	–	0%	–	–	0%	–	–	71%
knpkq	0%	0%	1%	6%	1%	6%	30%	0%	2%	1%	7%	2%	16%	71%
kqnpk	71%	0%	–	–	0%	–	–	0%	–	–	0%	–	–	71%
kqkrb	–	–	–	–	–	–	–	30%	19%	7%	7%	6%	2%	71%
krbpk	72%	0%	–	–	0%	–	–	0%	–	–	0%	–	–	72%
kqbkq	0%	13%	7%	4%	29%	1%	0%	2%	3%	2%	6%	5%	0%	72%
krbbk	70%	2%	–	–	0%	–	–	0%	–	–	–	–	–	72%
kbbpk	69%	2%	–	–	2%	–	–	0%	–	–	0%	–	–	72%
krnnk	70%	2%	–	–	0%	–	–	–	–	–	–	–	–	72%
kppkb	12%	14%	0%	–	33%	–	–	2%	0%	0%	7%	5%	0%	73%
krbnk	74%	0%	–	–	0%	–	–	–	–	–	–	–	–	74%
kqbpk	74%	0%	–	–	–	–	–	0%	–	–	0%	–	–	74%
kqpkq	2%	13%	5%	11%	15%	1%	0%	4%	7%	3%	7%	6%	0%	74%
kbpkn	15%	15%	0%	0%	36%	0%	–	2%	0%	0%	3%	4%	0%	74%
kqqqk	74%	–	–	–	–	–	–	–	–	–	–	–	–	74%
kppkq	0%	0%	1%	4%	0%	9%	28%	0%	2%	1%	9%	1%	21%	75%
kqbnk	75%	0%	–	–	0%	–	–	–	–	–	–	–	–	75%
kqnnk	75%	0%	–	–	–	–	–	–	–	–	–	–	–	75%

continued on next page

	WWW $d_1 \geq d_2$	WDD	WDL	WLL	DDD	DLL	LLL $d_1 \geq d_2$	WD	WL	DL	W	D	L	Sum
kqbbk	75%	0%	–	–	–	–	–	0%	–	–	–	–	–	75%
kqkrr	–	–	–	–	–	–	–	17%	25%	17%	5%	6%	6%	76%
knpkn	12%	14%	0%	0%	42%	0%	–	1%	0%	0%	2%	5%	0%	77%
kbbbk	49%	3%	–	–	26%	–	–	–	–	–	–	–	–	78%
kqnkq	0%	10%	7%	6%	32%	1%	0%	2%	4%	3%	6%	5%	0%	78%
kbbkp	23%	5%	1%	1%	42%	2%	1%	0%	0%	0%	1%	1%	0%	78%
kbpkr	1%	13%	1%	0%	38%	7%	0%	3%	0%	3%	4%	7%	1%	79%
krbkr	1%	18%	0%	0%	49%	0%	0%	2%	0%	0%	4%	5%	0%	79%
kbbkn	20%	7%	–	–	47%	–	–	0%	–	0%	2%	4%	–	79%
knpkr	1%	12%	1%	1%	37%	8%	1%	2%	0%	2%	4%	9%	2%	80%
kbpkb	6%	15%	0%	–	47%	0%	–	2%	0%	0%	4%	6%	0%	80%
knpkb	6%	14%	0%	–	49%	–	–	2%	0%	0%	4%	6%	0%	81%
knnkp	7%	12%	0%	0%	57%	3%	1%	0%	–	0%	1%	2%	0%	82%
krnkr	1%	17%	0%	0%	51%	0%	0%	2%	0%	1%	4%	6%	0%	83%
kbbkr	0%	10%	0%	0%	68%	0%	0%	1%	0%	0%	2%	7%	0%	88%
kbnkr	0%	15%	0%	0%	60%	0%	0%	2%	0%	1%	4%	6%	0%	89%
kbnkn	2%	19%	0%	–	62%	–	–	1%	0%	0%	1%	6%	–	90%
knnkr	–	0%	0%	0%	77%	0%	0%	0%	–	1%	–	12%	0%	91%
kbbkb	0%	8%	0%	–	74%	–	–	1%	–	0%	2%	7%	0%	92%
kbnkb	0%	16%	0%	–	66%	–	–	1%	–	0%	3%	5%	0%	92%
knnkb	–	0%	–	–	87%	–	–	0%	–	0%	0%	8%	–	95%
knnkn	0%	0%	–	–	92%	–	–	0%	–	0%	0%	7%	–	99%
kkbbb	–	–	–	–	–	–	–	–	–	–	–	31%	69%	100%
kkbbn	–	–	–	–	–	–	–	–	–	–	–	4%	96%	100%
kkbbp	–	–	–	–	–	–	–	–	–	–	–	7%	93%	100%
kkbnn	–	–	–	–	–	–	–	–	–	–	–	8%	92%	100%
kkbnp	–	–	–	–	–	–	–	–	–	–	–	1%	99%	100%
kkbpp	–	–	–	–	–	–	–	–	–	–	–	1%	99%	100%
kknnn	–	–	–	–	–	–	–	–	–	–	–	25%	75%	100%
kknnp	–	–	–	–	–	–	–	–	–	–	–	12%	88%	100%
kknpp	–	–	–	–	–	–	–	–	–	–	–	1%	99%	100%
kkppp	–	–	–	–	–	–	–	–	–	–	–	1%	99%	100%
kkqbb	–	–	–	–	–	–	–	–	–	–	–	4%	96%	100%
kkqbn	–	–	–	–	–	–	–	–	–	–	–	0%	100%	100%
kkqbp	–	–	–	–	–	–	–	–	–	–	–	0%	100%	100%
kkqnn	–	–	–	–	–	–	–	–	–	–	–	8%	92%	100%
kkqnp	–	–	–	–	–	–	–	–	–	–	–	0%	100%	100%
kkqpp	–	–	–	–	–	–	–	–	–	–	–	0%	100%	100%
kkqqb	–	–	–	–	–	–	–	–	–	–	–	–	100%	100%
kkqqn	–	–	–	–	–	–	–	–	–	–	–	–	100%	100%
kkqqp	–	–	–	–	–	–	–	–	–	–	–	–	100%	100%
kkqqq	–	–	–	–	–	–	–	–	–	–	–	–	100%	100%

continued on next page

	WWW $d_1 \geq d_2$	WDD	WDL	WLL	DDD	DLL	LLL $d_1 \geq d_2$	WD	WL	DL	W	D	L	Sum
kkqqr	–	–	–	–	–	–	–	–	–	–	–	–	100%	100%
kkqrb	–	–	–	–	–	–	–	–	–	–	–	–	100%	100%
kkqrn	–	–	–	–	–	–	–	–	–	–	–	–	100%	100%
kkqrp	–	–	–	–	–	–	–	–	–	–	–	0%	100%	100%
kkqrr	–	–	–	–	–	–	–	–	–	–	–	–	100%	100%
kkrbb	–	–	–	–	–	–	–	–	–	–	–	4%	96%	100%
kkrbn	–	–	–	–	–	–	–	–	–	–	–	0%	100%	100%
kkrbp	–	–	–	–	–	–	–	–	–	–	–	0%	100%	100%
kkrnn	–	–	–	–	–	–	–	–	–	–	–	8%	92%	100%
kkrnp	–	–	–	–	–	–	–	–	–	–	–	0%	100%	100%
kkrpp	–	–	–	–	–	–	–	–	–	–	–	0%	100%	100%
kkrrb	–	–	–	–	–	–	–	–	–	–	–	–	100%	100%
kkrrn	–	–	–	–	–	–	–	–	–	–	–	–	100%	100%
kkrrp	–	–	–	–	–	–	–	–	–	–	–	0%	100%	100%
kkrrr	–	–	–	–	–	–	–	–	–	–	–	–	100%	100%
Mean	18%	2%	0%	1%	6%	1%	1%	2%	2%	4%	2%	4%	21%	63%

Non-Cordel Instances

	WWW $d_1 < d_2$	WWD	WWL	DDL	LLL $d_1 < d_2$	WW	DD	LL	Sum
kbknn	–	–	–	–	–	–	86%	–	86%
knknn	–	–	–	–	–	–	85%	0%	85%
krknn	–	–	–	–	–	1%	81%	–	82%
kqqkq	41%	26%	0%	0%	0%	10%	0%	0%	78%
kqkpp	–	–	–	–	–	76%	1%	0%	77%
kqknp	–	–	–	–	–	75%	1%	0%	77%
knkpp	–	–	–	–	–	0%	40%	36%	76%
kbkpp	–	–	–	–	–	0%	51%	24%	75%
krkbb	–	–	–	–	–	0%	74%	0%	75%
kbknp	–	–	–	–	–	0%	64%	10%	74%
knknp	–	–	–	–	–	0%	53%	20%	73%
kqrkq	29%	16%	16%	0%	0%	9%	1%	0%	71%
knkbb	–	–	–	–	–	–	43%	27%	70%
kbkbb	–	–	–	–	–	–	70%	0%	70%
krkbn	–	–	–	–	–	0%	69%	0%	70%
kqqkr	62%	0%	0%	–	–	8%	0%	0%	70%
kqknn	–	–	–	–	–	66%	4%	–	69%
kbkbn	–	–	–	–	–	–	69%	0%	69%
kqkbp	–	–	–	–	–	67%	1%	0%	68%
kqkbn	–	–	–	–	–	68%	0%	0%	68%

continued on next page

	WWW $d_1 < d_2$	WWD	WWL	DDL	LLL $d_1 < d_2$	WW	DD	LL	Sum
knkbp	–	–	–	–	–	0%	46%	23%	68%
kbkbp	–	–	–	–	–	0%	57%	11%	68%
krkpp	–	–	–	–	–	51%	10%	6%	68%
kpkpp	–	–	–	–	–	14%	6%	47%	67%
knkrp	–	–	–	–	–	0%	5%	62%	67%
knkbn	–	–	–	–	–	–	63%	2%	65%
knkqp	–	–	–	–	–	0%	1%	64%	65%
kbkqp	–	–	–	–	–	–	1%	64%	65%
kbkrp	–	–	–	–	–	–	7%	58%	65%
kpknn	–	–	–	–	–	4%	49%	9%	63%
kpknp	–	–	–	–	–	10%	10%	42%	62%
kpkrp	–	–	–	–	–	1%	0%	60%	61%
knkrn	–	–	–	–	–	–	3%	58%	61%
kqkbb	–	–	–	–	–	59%	1%	0%	61%
knkrr	–	–	–	–	–	–	1%	59%	60%
kbkrr	–	–	–	–	–	–	2%	58%	60%
krkqq	–	–	–	–	–	–	0%	60%	60%
knkrb	–	–	–	–	–	–	3%	57%	60%
kpkbp	–	–	–	–	–	5%	7%	48%	60%
krkqp	–	–	–	–	–	1%	0%	58%	59%
kqrkr	40%	12%	0%	0%	–	7%	0%	0%	59%
kbkqn	–	–	–	–	–	–	1%	58%	59%
kbkrn	–	–	–	–	–	–	5%	54%	59%
kbkqr	–	–	–	–	–	–	1%	58%	59%
knkqn	–	–	–	–	–	–	1%	58%	58%
kpkbb	–	–	–	–	–	2%	31%	25%	58%
krknp	–	–	–	–	–	9%	48%	1%	58%
kbkrb	–	–	–	–	–	–	4%	53%	57%
knkqr	–	–	–	–	–	–	0%	56%	56%
kpkqp	–	–	–	–	–	0%	0%	56%	56%
krkrn	–	–	–	–	–	0%	56%	1%	56%
kbkqb	–	–	–	–	–	–	1%	55%	56%
knkqb	–	–	–	–	–	–	1%	55%	56%
kqqkb	50%	0%	–	–	–	6%	–	–	56%
kbkqq	–	–	–	–	–	–	0%	56%	56%
kpkrn	–	–	–	–	–	0%	2%	54%	56%
krrkr	28%	23%	0%	0%	–	4%	0%	0%	56%
krkqr	–	–	–	–	–	–	1%	54%	55%
krkbp	–	–	–	–	–	3%	49%	2%	55%
kpkbn	–	–	–	–	–	4%	4%	47%	54%
kpkrb	–	–	–	–	–	0%	1%	52%	54%
kpkrr	–	–	–	–	–	0%	0%	53%	54%

continued on next page

	WWW $d_1 < d_2$	WWD	WWL	DDL	LLL $d_1 < d_2$	WW	DD	LL	Sum
krkrp	–	–	–	–	–	2%	30%	21%	53%
kqqkp	51%	1%	0%	–	–	1%	0%	–	53%
krkqn	–	–	–	–	–	0%	2%	51%	53%
krkqb	–	–	–	–	–	0%	2%	51%	53%
kqqkn	49%	0%	–	–	–	4%	0%	–	53%
kqkrp	–	–	–	–	–	49%	3%	1%	53%
krkrb	–	–	–	–	–	0%	51%	1%	52%
knkqq	–	–	–	–	–	–	–	51%	51%
krkrr	–	–	–	–	–	–	4%	47%	51%
kpkqn	–	–	–	–	–	0%	0%	50%	50%
krrkb	31%	14%	–	–	–	5%	0%	–	50%
krrkn	39%	8%	–	–	–	3%	0%	–	49%
kqrkb	37%	7%	–	–	–	6%	0%	–	49%
krrkq	0%	11%	18%	11%	0%	4%	2%	2%	49%
kpkqr	–	–	–	–	–	0%	0%	49%	49%
krrkp	42%	5%	1%	0%	–	1%	0%	–	49%
kqqpk	45%	3%	–	–	–	0%	–	–	48%
kpkqb	–	–	–	–	–	0%	0%	48%	48%
kppkp	23%	7%	1%	2%	5%	7%	1%	2%	47%
krpkb	21%	17%	0%	0%	–	8%	1%	–	47%
kqrkn	40%	2%	–	–	–	4%	0%	–	47%
krpkn	27%	12%	0%	0%	–	6%	1%	–	46%
kqqnk	44%	2%	–	–	–	0%	–	–	46%
kqrkp	44%	0%	0%	–	–	1%	0%	–	46%
kpppk	44%	0%	–	–	–	1%	0%	–	45%
kqpkp	35%	2%	0%	0%	–	7%	0%	–	45%
kqbkr	23%	16%	0%	0%	–	5%	0%	0%	45%
kqpkn	33%	3%	0%	–	–	9%	0%	–	45%
kqkqq	–	–	–	–	–	0%	2%	42%	45%
kqqbk	42%	2%	–	–	–	0%	–	–	43%
kqpkb	29%	3%	0%	–	–	10%	0%	–	43%
krpkp	30%	3%	2%	0%	0%	6%	0%	0%	41%
krbkn	29%	9%	–	–	–	2%	0%	–	41%
kpkqq	–	–	–	–	–	–	0%	41%	41%
krnkb	23%	13%	–	–	–	3%	1%	–	40%
kqkqn	–	–	–	–	–	2%	37%	1%	40%
knpkp	17%	11%	1%	2%	4%	4%	1%	0%	40%
krbkb	23%	13%	–	–	–	3%	1%	–	40%
kqpkr	23%	3%	3%	0%	0%	11%	0%	0%	40%
krnkn	28%	10%	–	–	–	2%	0%	–	40%
kqnkr	22%	12%	1%	0%	–	5%	0%	0%	40%
kqkqr	–	–	–	–	–	1%	2%	37%	39%

continued on next page

	WWW $d_1 < d_2$	WWD	WWL	DDL	LLL $d_1 < d_2$	WW	DD	LL	Sum
kbpkp	19%	10%	1%	1%	2%	4%	1%	0%	39%
krbkp	28%	9%	1%	0%	0%	1%	0%	0%	39%
krrnk	38%	0%	–	–	–	0%	–	–	38%
krrpk	38%	0%	–	–	–	0%	–	–	38%
kqrpk	37%	1%	–	–	–	0%	–	–	38%
kbnkp	24%	7%	3%	1%	2%	1%	0%	0%	37%
krnkp	28%	8%	1%	0%	0%	1%	0%	0%	37%
kqbkb	25%	8%	–	–	–	4%	0%	–	37%
kqkrn	–	–	–	–	–	26%	11%	0%	37%
kqppk	36%	0%	–	–	–	0%	–	–	36%
krrrk	36%	0%	–	–	–	0%	–	–	36%
kqkqb	–	–	–	–	–	2%	32%	2%	36%
krrbk	36%	0%	–	–	–	0%	–	–	36%
kqnkn	28%	4%	–	–	–	3%	0%	–	36%
kqkqp	–	–	–	–	–	2%	24%	9%	36%
kqrnk	35%	1%	–	–	–	0%	–	–	36%
kqrrk	35%	0%	–	–	–	0%	–	–	36%
knppk	34%	0%	–	–	–	0%	0%	–	35%
kqbkn	27%	4%	–	–	–	3%	0%	–	35%
kqrbk	34%	1%	–	–	–	0%	–	–	35%
kqnkb	24%	7%	–	–	–	4%	0%	–	34%
kppkn	16%	9%	0%	0%	0%	4%	5%	0%	34%
kqnkp	27%	5%	0%	0%	–	1%	0%	–	34%
krbkq	0%	2%	7%	14%	2%	3%	2%	3%	33%
kqqrk	33%	0%	–	–	–	0%	–	–	33%
kqbkp	26%	5%	0%	–	–	1%	0%	–	33%
kbppk	32%	0%	–	–	–	0%	0%	–	33%
krpkr	10%	12%	2%	1%	0%	4%	3%	0%	32%
krpkq	0%	1%	3%	3%	13%	2%	1%	9%	32%
kbbkq	–	0%	2%	4%	16%	1%	1%	8%	32%
kbnnk	30%	2%	–	–	–	0%	–	–	32%
knnnk	21%	11%	–	–	–	–	–	–	31%
krnkq	0%	1%	5%	12%	6%	2%	1%	4%	31%
krppk	30%	0%	–	–	–	0%	0%	–	31%
knnkq	–	–	–	8%	14%	–	2%	7%	31%
kppkr	3%	3%	2%	3%	11%	1%	1%	6%	31%
kbnpk	31%	0%	–	–	–	0%	0%	–	31%
kbpkq	0%	0%	1%	2%	15%	1%	0%	11%	31%
knnpk	26%	4%	–	–	–	0%	0%	–	30%
kbbnk	30%	1%	–	–	–	0%	–	–	30%
kbnkq	0%	0%	2%	1%	17%	2%	0%	8%	30%
krnpk	29%	0%	–	–	–	0%	0%	–	29%

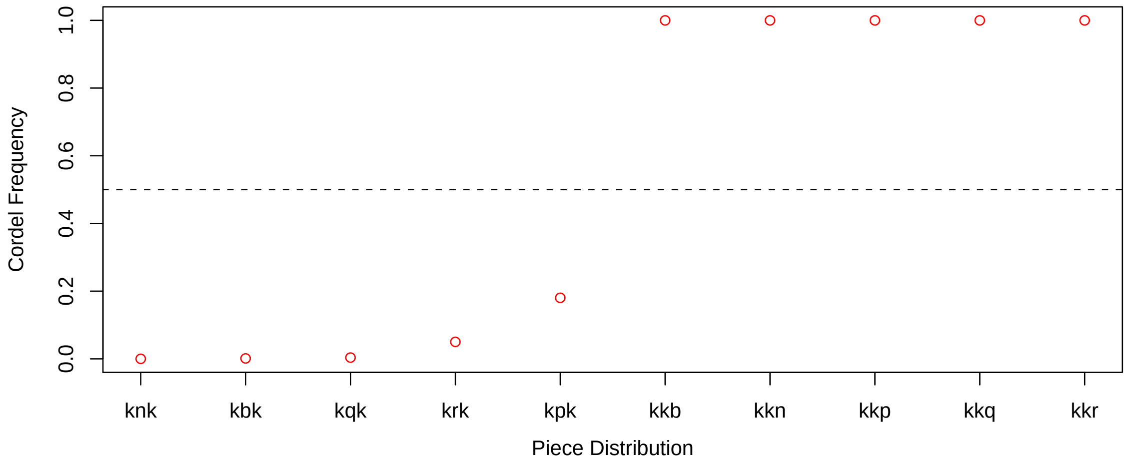
continued on next page

	WWW $d_1 < d_2$	WWD	WWL	DDL	LLL $d_1 < d_2$	WW	DD	LL	Sum
knpkq	0%	0%	1%	2%	15%	1%	0%	10%	29%
kqnpk	28%	1%	–	–	–	0%	–	–	29%
kqkrb	–	–	–	–	–	15%	13%	1%	29%
krbpk	28%	0%	–	–	–	0%	0%	–	28%
kqbkq	0%	9%	7%	3%	0%	4%	5%	0%	28%
krbbk	27%	0%	–	–	–	0%	0%	–	28%
kbbpk	26%	1%	–	–	–	0%	0%	–	28%
krnnk	26%	2%	–	–	–	0%	–	–	28%
kppkb	8%	9%	0%	0%	–	3%	7%	–	27%
krbnk	26%	0%	–	–	–	0%	–	–	26%
kqbpk	25%	1%	–	–	–	0%	–	–	26%
kqpkq	3%	9%	2%	1%	0%	5%	5%	0%	26%
kbpkn	10%	10%	0%	0%	–	2%	4%	–	26%
kqqqk	25%	0%	–	–	–	0%	–	–	26%
kppkq	0%	0%	0%	1%	14%	1%	0%	9%	25%
kqbnk	24%	1%	–	–	–	0%	–	–	25%
kqnnk	24%	1%	–	–	–	0%	–	–	25%
kqbbk	24%	1%	–	–	–	0%	–	–	25%
kqkrr	–	–	–	–	–	5%	14%	5%	24%
knpkn	9%	8%	0%	0%	–	1%	4%	–	23%
kbbbk	22%	0%	–	–	–	0%	0%	–	22%
kqnkq	0%	6%	5%	2%	0%	4%	5%	0%	22%
kbbkp	12%	2%	2%	4%	1%	0%	1%	0%	22%
kbpkr	1%	6%	0%	6%	0%	1%	7%	1%	21%
krbkr	1%	14%	0%	0%	0%	1%	5%	0%	21%
kbbkn	14%	4%	–	0%	–	1%	1%	–	21%
knpkr	1%	5%	0%	7%	0%	1%	6%	1%	20%
kbpkb	5%	8%	0%	0%	–	1%	6%	–	20%
knpkb	5%	7%	0%	0%	–	1%	6%	–	19%
knnkp	6%	6%	0%	5%	1%	0%	0%	0%	18%
krnkr	1%	10%	0%	0%	0%	1%	5%	0%	17%
kbbkr	0%	3%	0%	0%	0%	1%	8%	0%	12%
kbnkr	0%	4%	0%	1%	0%	1%	6%	0%	11%
kbnkn	1%	8%	–	–	–	0%	1%	–	10%
knnkr	–	0%	–	2%	0%	–	7%	0%	9%
kbbkb	0%	4%	–	0%	–	0%	3%	–	8%
kbnkb	0%	4%	–	0%	–	1%	3%	–	8%
knnkb	0%	0%	–	–	–	–	5%	–	5%
knnkn	0%	0%	–	–	–	–	1%	–	1%
kkbbb	–	–	–	–	–	–	–	–	–
kkbbn	–	–	–	–	–	–	–	–	–
kkbbp	–	–	–	–	–	–	–	–	–

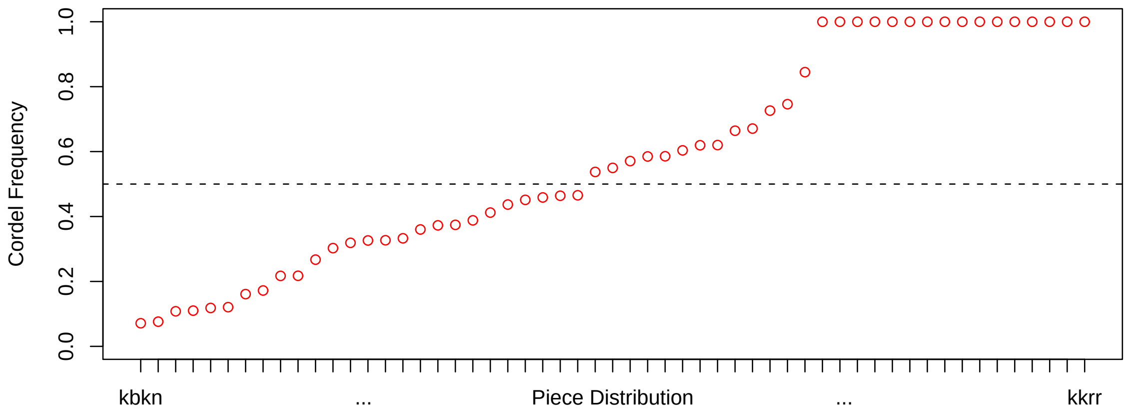
continued on next page

	WWW $d_1 < d_2$	WWD	WWL	DDL	LLL $d_1 < d_2$	WW	DD	LL	Sum
kkbnn	–	–	–	–	–	–	–	–	–
kkbnp	–	–	–	–	–	–	–	–	–
kkbpp	–	–	–	–	–	–	–	–	–
kknnn	–	–	–	–	–	–	–	–	–
kknnp	–	–	–	–	–	–	–	–	–
kknpp	–	–	–	–	–	–	–	–	–
kkppp	–	–	–	–	–	–	–	–	–
kkqbb	–	–	–	–	–	–	–	–	–
kkqbn	–	–	–	–	–	–	–	–	–
kkqbp	–	–	–	–	–	–	–	–	–
kkqnn	–	–	–	–	–	–	–	–	–
kkqnp	–	–	–	–	–	–	–	–	–
kkqpp	–	–	–	–	–	–	–	–	–
kkqqb	–	–	–	–	–	–	–	–	–
kkqqn	–	–	–	–	–	–	–	–	–
kkqqp	–	–	–	–	–	–	–	–	–
kkqqq	–	–	–	–	–	–	–	–	–
kkqqr	–	–	–	–	–	–	–	–	–
kkqrb	–	–	–	–	–	–	–	–	–
kkqrn	–	–	–	–	–	–	–	–	–
kkqrp	–	–	–	–	–	–	–	–	–
kkqrr	–	–	–	–	–	–	–	–	–
kkrbb	–	–	–	–	–	–	–	–	–
kkrbn	–	–	–	–	–	–	–	–	–
kkrbp	–	–	–	–	–	–	–	–	–
kkrnn	–	–	–	–	–	–	–	–	–
kkrnp	–	–	–	–	–	–	–	–	–
kkrpp	–	–	–	–	–	–	–	–	–
kkrrb	–	–	–	–	–	–	–	–	–
kkrrn	–	–	–	–	–	–	–	–	–
kkrrp	–	–	–	–	–	–	–	–	–
kkrrr	–	–	–	–	–	–	–	–	–
Mean	11%	2%	0%	0%	1%	4%	7%	11%	37%

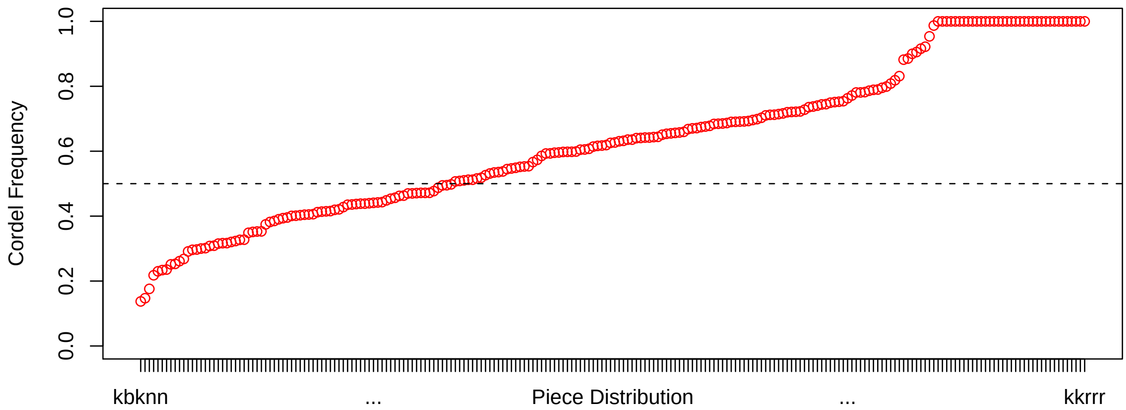
E.4 Figures



(a) Endgames with 3 pieces (including kings)

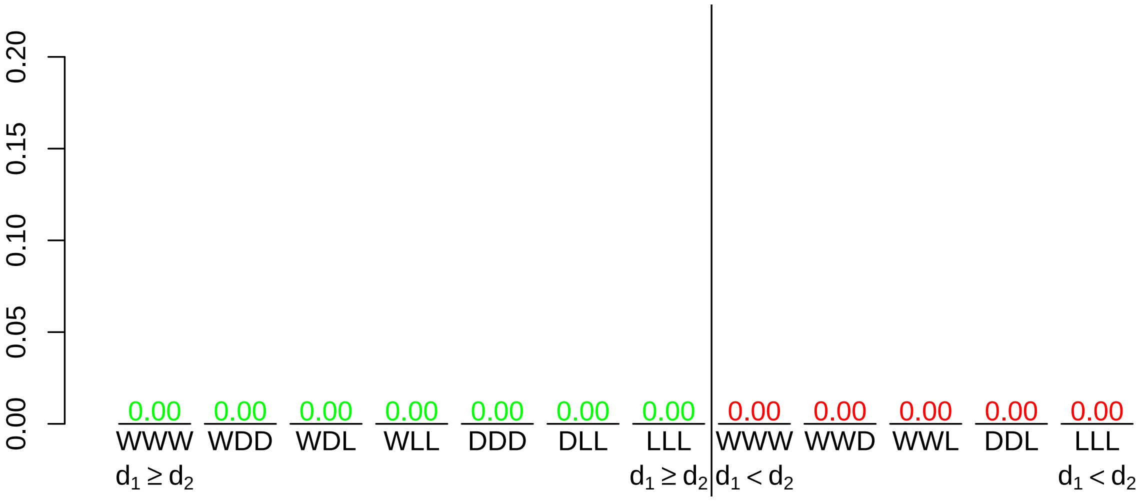


(b) Endgames with 4 pieces (including kings)

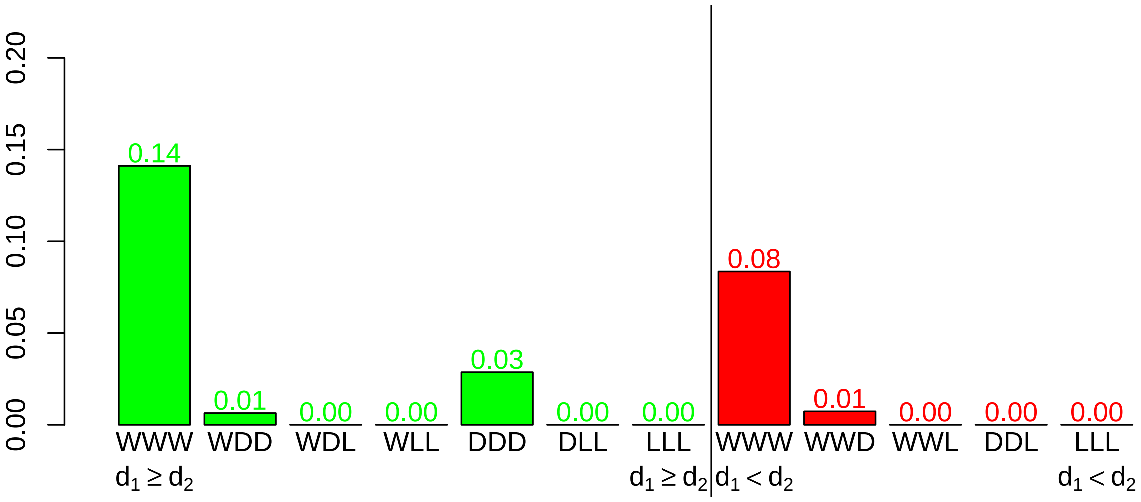


(c) Endgames with 5 pieces (including kings)

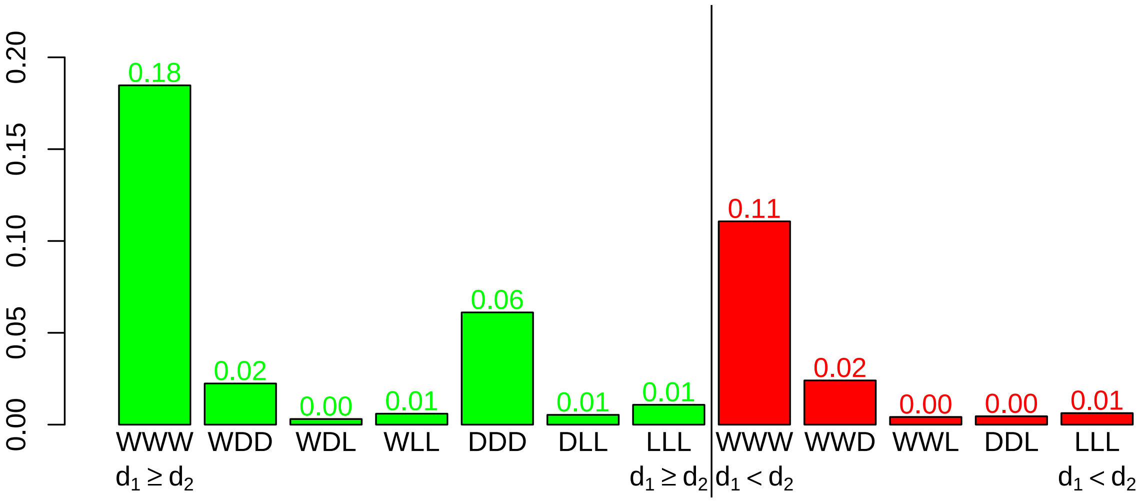
Figure E.4.1: Cordel Frequencies for different piece configurations



(a) Endgames with 3 pieces (including kings)

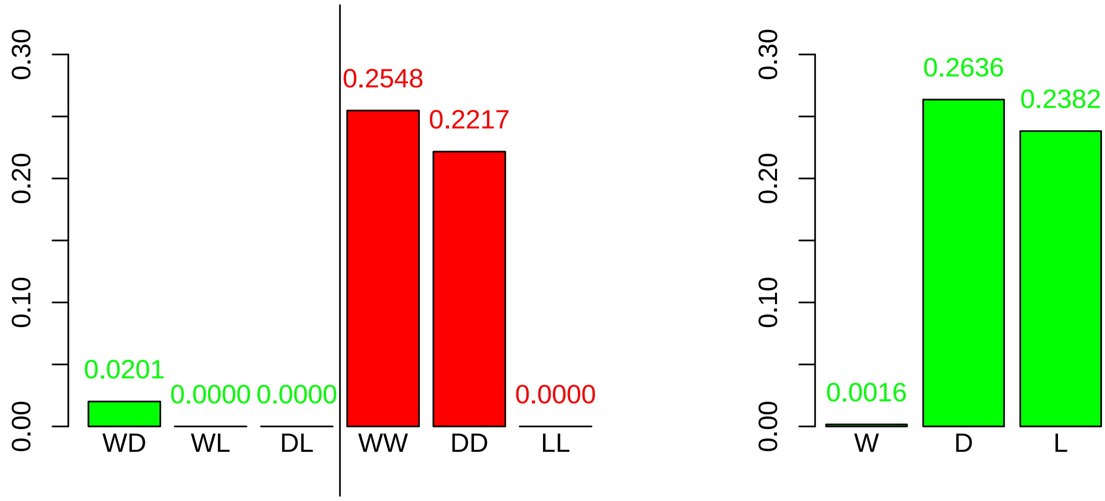


(b) Endgames with 4 pieces (including kings)

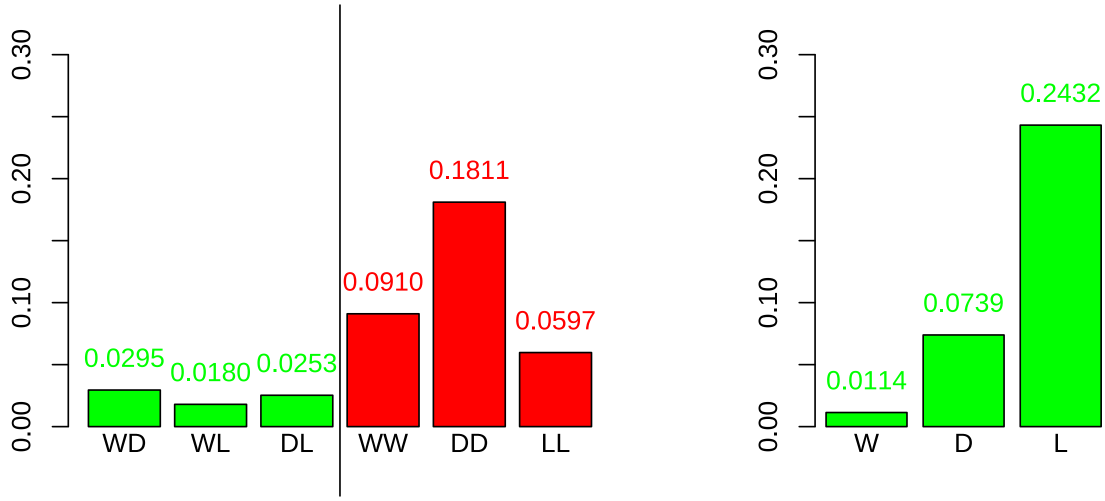


(c) Endgames with 5 pieces (including kings)

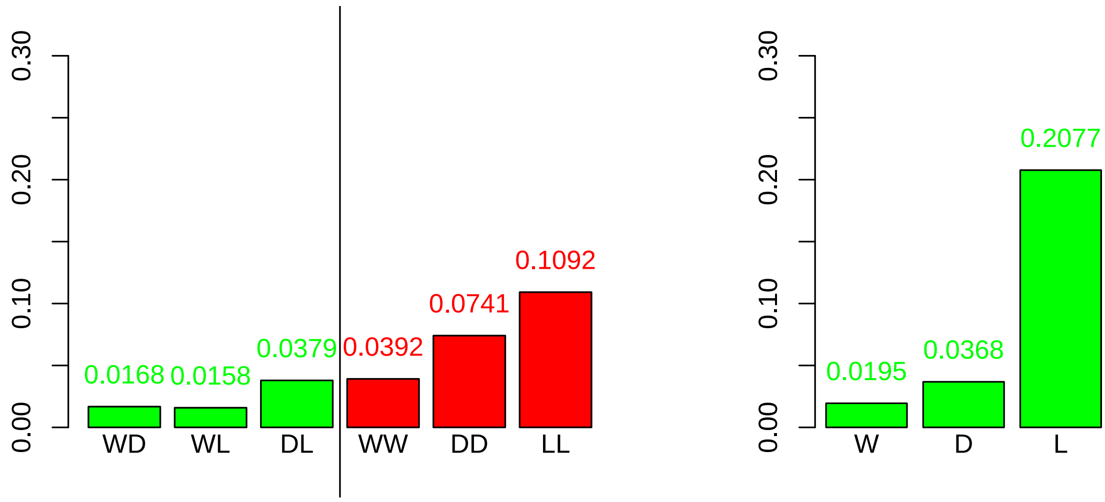
Figure E.4.2: Relative frequencies of different types of chess positions in endgames with 3, 4, or 5 pieces (including kings) with at least three feasible moves.



(a) Endgames with 3 pieces (including kings)



(b) Endgames with 4 pieces (including kings)



(c) Endgames with 5 pieces (including kings)

Figure E.4.3: Relative frequencies of different types of chess positions in endgames with 3, 4, or 5 pieces (including kings) with only two (left) or one (right) feasible moves.

E.5 Monte Carlo Samples for Endgames with 6 Pieces

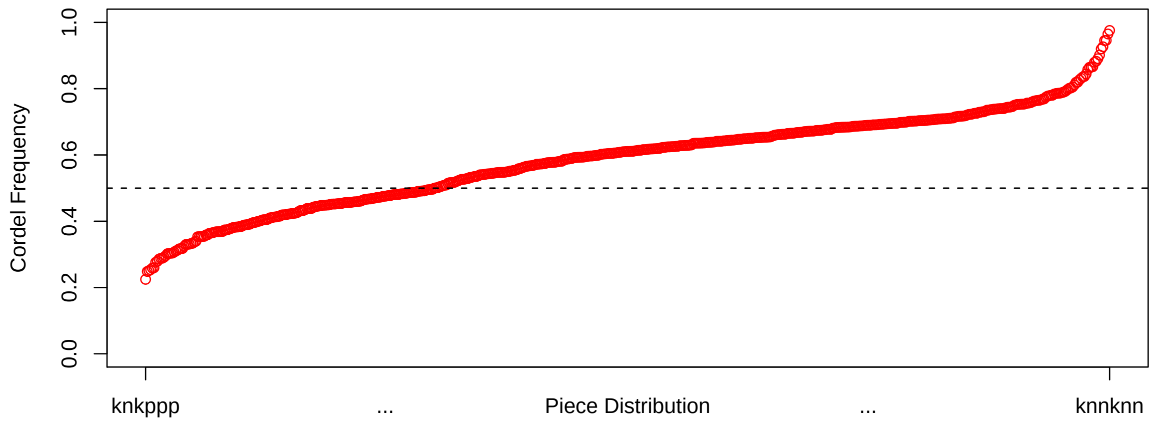
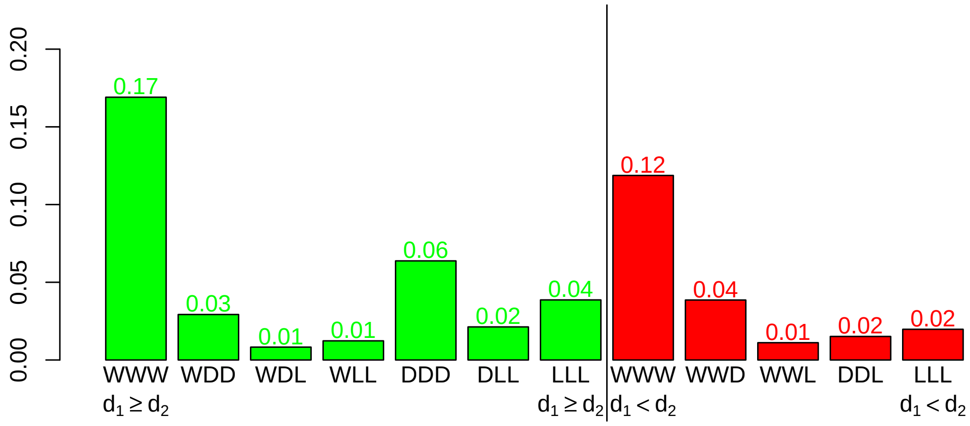
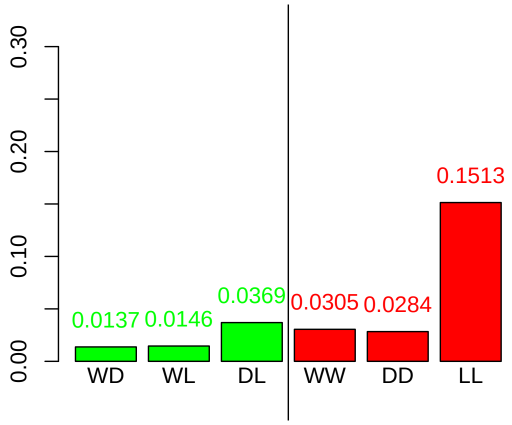


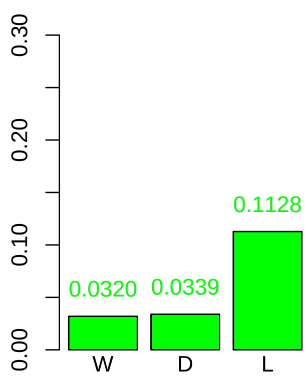
Figure E.5.1: Cordel Frequencies for the 515 different piece configurations.



(a) Endgames with at least 3 feasible moves.



(b) Endgames with exactly 2 feasible moves.



(c) Endgames with exactly 1 feasible move.

Figure E.5.2: Relative frequencies of different types of chess positions in endgames with 6 pieces (including kings) with only two (left) or one (right) feasible moves.

Appendix F

Detailed Results for Chess – Best Move Per Piece Type Rule

Refers to page 16.

Overview of the results presented in this chapter:

F.1 Endgames with 3 Pieces

Cordel Instances

	WWW $d_1 \geq d_2$	WDD	WDL	WLL	DDD	DLL	LLL $d_1 \geq d_2$	WD	WL	DL	W	D	L	Sum
knk	–	–	–	–	–	–	–	–	–	–	–	0%	–	0%
kbk	–	–	–	–	–	–	–	–	–	–	–	0%	–	0%
kqk	–	–	–	–	–	–	–	0%	–	–	0%	–	–	0%
krk	–	–	–	–	–	–	–	5%	–	–	0%	–	–	5%
kpk	–	–	–	–	–	–	–	15%	–	–	2%	2%	–	18%
kkb	–	–	–	–	–	–	–	–	–	–	–	100%	–	100%
kkn	–	–	–	–	–	–	–	–	–	–	–	100%	–	100%
kkp	–	–	–	–	–	–	–	–	–	–	–	42%	58%	100%
kkq	–	–	–	–	–	–	–	–	–	–	–	10%	90%	100%
kkr	–	–	–	–	–	–	–	–	–	–	–	10%	90%	100%
Mean	–	–	–	–	–	–	–	2%	–	–	0%	26%	24%	52%

Non-Cordel Instances

	WWW $d_1 < d_2$	WWD	WWL	DDL	LLL $d_1 < d_2$	WW	DD	LL	Sum
knk	–	–	–	–	–	–	100%	–	100%
kbk	–	–	–	–	–	–	100%	–	100%
kqk	–	–	–	–	–	100%	–	–	100%
krk	–	–	–	–	–	95%	–	–	95%
kpk	–	–	–	–	–	60%	22%	–	82%
kkb	–	–	–	–	–	–	–	–	–
kkn	–	–	–	–	–	–	–	–	–
kkp	–	–	–	–	–	–	–	–	–
kkq	–	–	–	–	–	–	–	–	–
kkr	–	–	–	–	–	–	–	–	–
Mean	–	–	–	–	–	25%	22%	–	48%

F.2 Endgames with 4 Pieces

Cordel Instances

	WWW	WDD	WDL	WLL	DDD	DLL	LLL	WD	WL	DL	W	D	L	Sum
	$d_1 \geq d_2$						$d_1 \geq d_2$							
knnk	–	–	–	–	–	–	–	0%	–	–	–	0%	–	0%
krrk	–	–	–	–	–	–	–	0%	–	–	0%	–	–	0%
kqqk	–	–	–	–	–	–	–	4%	–	–	0%	–	–	4%
kppk	–	–	–	–	–	–	–	4%	–	–	0%	0%	–	4%
kbbk	–	–	–	–	–	–	–	5%	–	–	0%	0%	–	5%
kbkn	–	–	–	–	–	–	–	0%	–	–	–	7%	–	7%
knkn	–	–	–	–	–	–	–	0%	–	–	–	8%	–	8%
kbkb	–	–	–	–	–	–	–	0%	–	–	–	11%	–	11%
knkb	–	–	–	–	–	–	–	0%	–	–	–	11%	–	11%
kqkn	–	–	–	–	–	–	–	7%	–	–	4%	1%	–	12%
kqkp	–	–	–	–	–	–	–	9%	1%	0%	2%	0%	0%	12%
kbkp	–	–	–	–	–	–	–	0%	–	13%	–	3%	0%	16%
knkp	–	–	–	–	–	–	–	0%	0%	15%	0%	3%	0%	17%
kqkb	–	–	–	–	–	–	–	16%	–	–	6%	0%	–	22%
krkp	–	–	–	–	–	–	–	8%	9%	2%	2%	0%	0%	22%
kpkn	–	–	–	–	–	–	–	14%	0%	0%	2%	10%	0%	27%
kpkb	–	–	–	–	–	–	–	13%	–	0%	4%	13%	–	30%
kpkp	–	–	–	–	–	–	–	11%	3%	9%	3%	3%	2%	32%
kbkr	–	–	–	–	–	–	–	–	–	19%	–	11%	2%	33%
krkn	–	–	–	–	–	–	–	26%	–	–	2%	5%	–	33%
krkb	–	–	–	–	–	–	–	26%	–	–	3%	4%	–	33%
kqkr	–	–	–	–	–	–	–	9%	19%	0%	7%	0%	0%	36%
kbkq	–	–	–	–	–	–	–	–	–	13%	–	9%	15%	37%
knkq	–	–	–	–	–	–	–	–	–	10%	–	9%	19%	37%
knkr	–	–	–	–	–	–	–	0%	0%	22%	–	13%	4%	39%
krkr	–	–	–	–	–	–	–	7%	14%	7%	4%	8%	0%	41%
kpkr	–	–	–	–	–	–	–	4%	8%	7%	3%	5%	16%	44%
krkq	–	–	–	–	–	–	–	0%	19%	5%	7%	1%	13%	45%
kpkq	–	–	–	–	–	–	–	0%	2%	7%	6%	4%	27%	46%
kqkq	–	–	–	–	–	–	–	9%	23%	8%	7%	7%	0%	54%
kqrk	57%	0%	–	–	–	–	–	–	–	–	–	–	–	57%
kqpk	58%	0%	–	–	–	–	–	0%	–	–	0%	–	–	59%
knpk	49%	7%	–	–	3%	–	–	1%	–	–	0%	0%	–	59%
kbpk	52%	5%	–	–	3%	–	–	1%	–	–	0%	0%	–	60%
kbnk	52%	9%	–	–	0%	–	–	0%	–	–	–	–	–	62%
krpk	61%	0%	–	–	–	–	–	0%	–	–	0%	–	–	62%
krnk	63%	3%	–	–	–	–	–	–	–	–	–	–	–	66%

continued on next page

	WWW $d_1 \geq d_2$	WDD	WDL	WLL	DDD	DLL	LLL $d_1 \geq d_2$	WD	WL	DL	W	D	L	Sum
krbk	64%	3%	–	–	–	–	–	0%	–	–	–	–	–	67%
kqnk	72%	0%	–	–	–	–	–	–	–	–	–	–	–	73%
kqbk	74%	0%	–	–	–	–	–	0%	–	–	–	–	–	75%
kkbb	–	–	–	–	–	–	–	–	–	–	–	59%	41%	100%
kkbn	–	–	–	–	–	–	–	–	–	–	–	18%	82%	100%
kkbp	–	–	–	–	–	–	–	–	–	–	–	17%	83%	100%
kknn	–	–	–	–	–	–	–	–	–	–	–	100%	–	100%
kknp	–	–	–	–	–	–	–	–	–	–	–	18%	82%	100%
kkpp	–	–	–	–	–	–	–	–	–	–	–	8%	92%	100%
kkqb	–	–	–	–	–	–	–	–	–	–	–	9%	91%	100%
kkqn	–	–	–	–	–	–	–	–	–	–	–	9%	91%	100%
kkqp	–	–	–	–	–	–	–	–	–	–	–	2%	98%	100%
kkqq	–	–	–	–	–	–	–	–	–	–	–	–	100%	100%
kkqr	–	–	–	–	–	–	–	–	–	–	–	–	100%	100%
kkrb	–	–	–	–	–	–	–	–	–	–	–	9%	91%	100%
kkrn	–	–	–	–	–	–	–	–	–	–	–	9%	91%	100%
kkrp	–	–	–	–	–	–	–	–	–	–	–	2%	98%	100%
kkrr	–	–	–	–	–	–	–	–	–	–	–	–	100%	100%
Mean	11%	1%	–	–	0%	–	–	3%	2%	3%	1%	7%	24%	52%

Non-Cordel Instances

	WWW $d_1 < d_2$	WWD	WWL	DDL	LLL $d_1 < d_2$	WW	DD	LL	Sum
knnk	–	–	–	–	–	–	100%	–	100%
krrk	–	–	–	–	–	100%	–	–	100%
kqqk	–	–	–	–	–	96%	–	–	96%
kppk	–	–	–	–	–	95%	2%	–	96%
kbbk	–	–	–	–	–	45%	51%	–	95%
kbkn	–	–	–	–	–	–	93%	–	93%
knkn	–	–	–	–	–	–	92%	–	92%
kbkb	–	–	–	–	–	–	89%	–	89%
knkb	–	–	–	–	–	–	89%	–	89%
kqkn	–	–	–	–	–	88%	0%	–	88%
kqkp	–	–	–	–	–	87%	1%	–	88%
kbkp	–	–	–	–	–	–	79%	5%	84%
knkp	–	–	–	–	–	0%	70%	13%	83%
kqkb	–	–	–	–	–	78%	0%	–	78%
krkp	–	–	–	–	–	72%	6%	0%	78%
kpkn	–	–	–	–	–	16%	57%	0%	73%
kpkb	–	–	–	–	–	7%	63%	–	70%

continued on next page

	WWW $d_1 < d_2$	WWD	WWL	DDL	LLL $d_1 < d_2$	WW	DD	LL	Sum
kpkp	–	–	–	–	–	26%	21%	22%	68%
kbkr	–	–	–	–	–	–	66%	1%	67%
krkn	–	–	–	–	–	20%	47%	–	67%
krkb	–	–	–	–	–	6%	61%	–	67%
kqkr	–	–	–	–	–	64%	0%	–	64%
kbkq	–	–	–	–	–	–	1%	61%	63%
knkq	–	–	–	–	–	–	1%	62%	63%
knkr	–	–	–	–	–	–	54%	7%	61%
krkr	–	–	–	–	–	4%	55%	0%	59%
kpkr	–	–	–	–	–	1%	5%	50%	56%
krkq	–	–	–	–	–	2%	0%	53%	55%
kpkq	–	–	–	–	–	0%	1%	53%	54%
kqkq	–	–	–	–	–	3%	43%	0%	46%
kqrk	42%	1%	–	–	–	0%	–	–	43%
kqpk	36%	0%	–	–	–	5%	–	–	41%
knpk	28%	10%	–	–	–	3%	1%	–	41%
kbpk	28%	8%	–	–	–	4%	1%	–	40%
kbnk	32%	6%	–	–	–	0%	0%	–	38%
krpk	32%	1%	–	–	–	5%	–	–	38%
krnk	31%	2%	–	–	–	0%	–	–	34%
krbk	30%	2%	–	–	–	0%	–	–	33%
kqnk	27%	1%	–	–	–	0%	–	–	27%
kqbk	24%	1%	–	–	–	0%	–	–	25%
kkbb	–	–	–	–	–	–	–	–	–
kkbn	–	–	–	–	–	–	–	–	–
kkbp	–	–	–	–	–	–	–	–	–
kknn	–	–	–	–	–	–	–	–	–
kknp	–	–	–	–	–	–	–	–	–
kkpp	–	–	–	–	–	–	–	–	–
kkqb	–	–	–	–	–	–	–	–	–
kkqn	–	–	–	–	–	–	–	–	–
kkqp	–	–	–	–	–	–	–	–	–
kkqq	–	–	–	–	–	–	–	–	–
kkqr	–	–	–	–	–	–	–	–	–
kkrb	–	–	–	–	–	–	–	–	–
kkrn	–	–	–	–	–	–	–	–	–
kkrp	–	–	–	–	–	–	–	–	–
kkrr	–	–	–	–	–	–	–	–	–
Mean	6%	1%	–	–	–	15%	21%	6%	48%

F.3 Endgames with 5 Pieces

Cordel Instances

	WWW $d_1 \geq d_2$	WDD	WDL	WLL	DDD	DLL	LLL $d_1 \geq d_2$	WD	WL	DL	W	D	L	Sum
kpppk	–	–	–	–	–	–	–	0%	–	–	0%	0%	–	0%
krrrk	–	–	–	–	–	–	–	1%	–	–	0%	–	–	1%
kqqkp	–	–	–	–	–	–	–	1%	0%	–	1%	0%	0%	2%
kbbbk	–	–	–	–	–	–	–	3%	–	–	0%	0%	–	3%
kqqkn	–	–	–	–	–	–	–	0%	–	–	3%	0%	–	3%
kqqkb	–	–	–	–	–	–	–	0%	–	–	3%	–	–	3%
kqqkr	–	–	–	–	–	–	–	0%	0%	–	4%	0%	0%	5%
knnkn	–	–	–	–	–	–	–	0%	–	–	0%	7%	–	7%
krrkp	–	–	–	–	–	–	–	5%	1%	0%	2%	0%	0%	8%
knnkb	–	–	–	–	–	–	–	0%	–	–	0%	9%	–	9%
kqqqk	–	–	–	–	–	–	–	11%	–	–	0%	–	–	11%
krrkn	–	–	–	–	–	–	–	8%	–	–	5%	0%	–	13%
kbknn	–	–	–	–	–	–	–	0%	–	0%	–	14%	0%	14%
knnkr	–	–	–	–	–	–	–	0%	0%	2%	0%	12%	0%	15%
knknn	–	–	–	–	–	–	–	0%	–	0%	–	15%	0%	15%
kbbkn	–	–	–	–	–	–	–	10%	–	–	2%	4%	–	16%
krknn	–	–	–	–	–	–	–	5%	0%	0%	0%	12%	0%	18%
kbbkp	–	–	–	–	–	–	–	7%	4%	6%	1%	1%	0%	19%
krrkb	–	–	–	–	–	–	–	15%	–	–	4%	0%	–	19%
kbbkb	–	–	–	–	–	–	–	12%	0%	0%	2%	7%	0%	20%
knnnk	–	–	–	–	–	–	–	22%	–	–	0%	0%	–	22%
kbbkr	–	–	–	–	–	–	–	13%	0%	1%	2%	7%	0%	22%
kppkp	–	–	–	–	–	–	–	11%	5%	4%	3%	0%	0%	23%
kqkpp	–	–	–	–	–	–	–	15%	5%	0%	3%	0%	0%	23%
kqknp	–	–	–	–	–	–	–	10%	6%	0%	6%	1%	0%	23%
knkpp	–	–	–	–	–	–	–	0%	0%	19%	0%	4%	1%	24%
knnkp	–	–	–	–	–	–	–	15%	0%	6%	1%	2%	0%	24%
kbkpp	–	–	–	–	–	–	–	0%	0%	20%	0%	4%	1%	25%
krkbb	–	–	–	–	–	–	–	1%	0%	9%	0%	14%	1%	25%
kppkn	–	–	–	–	–	–	–	17%	0%	0%	4%	4%	0%	25%
kbknp	–	–	–	–	–	–	–	0%	0%	17%	0%	7%	2%	26%
knknp	–	–	–	–	–	–	–	0%	0%	17%	0%	7%	3%	27%
kqrkq	10%	1%	1%	1%	0%	0%	0%	2%	4%	0%	8%	2%	0%	29%
knkbb	–	–	–	–	–	–	–	0%	0%	9%	–	14%	6%	30%
kbkbb	–	–	–	–	–	–	–	0%	0%	9%	0%	19%	1%	30%
krkbn	–	–	–	–	–	–	–	3%	0%	13%	0%	12%	1%	30%
kqknn	–	–	–	–	–	–	–	21%	–	0%	8%	2%	–	31%

continued on next page

	WWW	WDD	WDL	WLL	DDD	DLL	LLL	WD	WL	DL	W	D	L	Sum
	$d_1 \geq d_2$						$d_1 \geq d_2$							
kbkbn	–	–	–	–	–	–	–	0%	0%	14%	0%	16%	1%	31%
kppkb	–	–	–	–	–	–	–	19%	0%	0%	7%	5%	0%	31%
kqkbp	–	–	–	–	–	–	–	12%	11%	1%	7%	1%	0%	32%
kqkbn	–	–	–	–	–	–	–	9%	11%	0%	9%	1%	1%	32%
knkbp	–	–	–	–	–	–	–	0%	0%	19%	0%	9%	4%	32%
kbkbp	–	–	–	–	–	–	–	0%	0%	19%	0%	10%	2%	32%
krkpp	–	–	–	–	–	–	–	7%	14%	8%	3%	0%	1%	32%
kpkpp	–	–	–	–	–	–	–	6%	5%	10%	2%	3%	7%	33%
knkrp	–	–	–	–	–	–	–	0%	0%	15%	0%	5%	13%	33%
knkbn	–	–	–	–	–	–	–	0%	0%	17%	–	15%	2%	35%
knkqp	–	–	–	–	–	–	–	0%	0%	6%	0%	6%	23%	35%
kbkqp	–	–	–	–	–	–	–	0%	0%	9%	0%	7%	19%	35%
kbkrp	–	–	–	–	–	–	–	0%	0%	20%	0%	6%	10%	35%
krrkr	–	–	–	–	–	–	–	28%	0%	0%	8%	0%	0%	37%
kpknn	–	–	–	–	–	–	–	8%	0%	9%	1%	15%	4%	37%
kpknp	–	–	–	–	–	–	–	6%	5%	12%	2%	6%	9%	38%
kqqkq	–	–	–	–	–	–	–	30%	0%	0%	7%	0%	0%	38%
kpkrp	–	–	–	–	–	–	–	1%	7%	4%	1%	2%	23%	39%
knkrn	–	–	–	–	–	–	–	0%	0%	16%	0%	5%	17%	39%
kqkbb	–	–	–	–	–	–	–	19%	8%	0%	10%	2%	0%	39%
knkrr	–	–	–	–	–	–	–	–	0%	9%	0%	5%	26%	40%
kbkrr	–	–	–	–	–	–	–	–	–	15%	–	6%	20%	40%
krkqq	–	–	–	–	–	–	–	–	0%	0%	0%	0%	40%	40%
knkrb	–	–	–	–	–	–	–	0%	0%	15%	0%	7%	19%	40%
kpkbp	–	–	–	–	–	–	–	4%	6%	10%	2%	6%	12%	40%
krkqp	–	–	–	–	–	–	–	0%	13%	5%	5%	1%	16%	41%
kqrkr	32%	0%	0%	0%	0%	–	–	2%	0%	0%	6%	0%	0%	41%
kbkqn	–	–	–	–	–	–	–	0%	0%	12%	0%	8%	21%	41%
kbkrn	–	–	–	–	–	–	–	0%	0%	22%	–	5%	14%	41%
kbkqr	–	–	–	–	–	–	–	–	–	6%	–	5%	30%	41%
knkqn	–	–	–	–	–	–	–	0%	0%	9%	–	8%	24%	42%
kpkbb	–	–	–	–	–	–	–	6%	3%	5%	1%	18%	9%	42%
krknp	–	–	–	–	–	–	–	18%	2%	14%	1%	6%	1%	42%
kbbkq	–	–	–	–	–	–	–	0%	11%	14%	3%	4%	10%	42%
kbkrb	–	–	–	–	–	–	–	0%	0%	20%	0%	7%	16%	43%
knkqr	–	–	–	–	–	–	–	–	0%	3%	0%	4%	36%	44%
kpkqp	–	–	–	–	–	–	–	0%	1%	4%	2%	3%	33%	44%
krkrn	–	–	–	–	–	–	–	1%	2%	22%	0%	16%	3%	44%
kbkqb	–	–	–	–	–	–	–	0%	0%	11%	0%	8%	25%	44%
knkqb	–	–	–	–	–	–	–	0%	0%	8%	–	8%	28%	44%
kbkqq	–	–	–	–	–	–	–	–	–	0%	–	0%	44%	44%
kpkrn	–	–	–	–	–	–	–	1%	3%	10%	1%	6%	25%	44%

continued on next page

	WWW $d_1 \geq d_2$	WDD	WDL	WLL	DDD	DLL	LLL $d_1 \geq d_2$	WD	WL	DL	W	D	L	Sum
krkqr	–	–	–	–	–	–	–	0%	0%	12%	0%	5%	28%	45%
kqrrk	45%	0%	–	–	–	–	–	0%	–	–	0%	–	–	45%
kppkr	–	–	–	–	–	–	–	9%	12%	6%	6%	4%	9%	45%
krkbp	–	–	–	–	–	–	–	16%	2%	18%	3%	6%	2%	45%
kpkbn	–	–	–	–	–	–	–	4%	5%	12%	1%	7%	16%	46%
kpkrb	–	–	–	–	–	–	–	0%	2%	10%	0%	6%	28%	46%
kqqrk	46%	0%	–	–	–	–	–	0%	–	–	0%	–	–	46%
kpkrr	–	–	–	–	–	–	–	0%	1%	5%	0%	1%	40%	46%
krkrp	–	–	–	–	–	–	–	3%	12%	17%	3%	7%	5%	47%
krkqn	–	–	–	–	–	–	–	0%	2%	19%	1%	7%	18%	47%
krkqb	–	–	–	–	–	–	–	0%	1%	19%	0%	7%	20%	47%
kqkrp	–	–	–	–	–	–	–	10%	25%	3%	8%	1%	1%	47%
knnkq	–	–	–	–	–	–	–	0%	0%	26%	0%	9%	12%	47%
krkrb	–	–	–	–	–	–	–	0%	0%	27%	0%	16%	4%	48%
kppkq	–	–	–	–	–	–	–	0%	6%	11%	9%	1%	21%	48%
knkqq	–	–	–	–	–	–	–	–	–	0%	–	0%	49%	49%
krkrr	–	–	–	–	–	–	–	0%	0%	24%	0%	6%	19%	49%
kpkqn	–	–	–	–	–	–	–	0%	1%	7%	1%	8%	33%	50%
kqrkb	46%	0%	–	–	0%	–	–	1%	–	–	4%	0%	–	51%
kpkqr	–	–	–	–	–	–	–	0%	0%	1%	0%	1%	50%	51%
kpkqb	–	–	–	–	–	–	–	0%	0%	6%	0%	8%	37%	52%
krpkb	32%	9%	0%	–	1%	–	–	3%	0%	0%	6%	1%	0%	53%
kqrkn	50%	0%	–	–	–	–	–	0%	–	–	4%	0%	–	53%
krpkn	39%	6%	0%	0%	1%	0%	–	1%	0%	0%	5%	1%	0%	54%
kqrkp	53%	0%	0%	0%	0%	–	–	0%	0%	–	1%	0%	0%	54%
kqpkp	50%	2%	0%	1%	0%	–	–	0%	0%	–	2%	0%	0%	55%
kqbkr	38%	7%	0%	0%	0%	0%	–	4%	0%	0%	5%	0%	0%	55%
kqpkn	47%	3%	0%	0%	–	–	–	0%	0%	–	5%	0%	0%	55%
kqkqq	–	–	–	–	–	–	–	0%	1%	23%	0%	8%	23%	55%
knnpk	52%	3%	–	–	1%	–	–	1%	–	–	0%	0%	–	57%
kqpkb	44%	5%	0%	–	0%	–	–	3%	–	–	6%	0%	0%	57%
krpkp	47%	2%	1%	5%	0%	0%	0%	0%	0%	0%	2%	0%	0%	59%
kbbpk	55%	2%	–	–	1%	–	–	0%	–	–	0%	0%	–	59%
krbkn	46%	7%	–	–	0%	–	–	0%	–	–	5%	1%	–	59%
kpkqq	–	–	–	–	–	–	–	0%	0%	0%	0%	0%	59%	59%
krnkb	40%	11%	–	–	0%	–	–	2%	0%	0%	5%	1%	–	60%
kqkqn	–	–	–	–	–	–	–	6%	19%	14%	6%	10%	3%	60%
knpkp	28%	9%	2%	4%	7%	3%	3%	1%	0%	0%	2%	0%	0%	60%
krbkb	41%	10%	–	–	0%	–	–	2%	–	–	5%	1%	0%	60%
kqpkr	36%	2%	1%	6%	0%	0%	0%	2%	4%	0%	8%	0%	0%	60%
krnkn	46%	8%	–	–	0%	–	–	0%	0%	0%	5%	1%	–	60%
kqppk	60%	0%	–	–	0%	–	–	0%	–	–	0%	–	–	60%

continued on next page

	WWW	WDD	WDL	WLL	DDD	DLL	LLL	WD	WL	DL	W	D	L	Sum
	$d_1 \geq d_2$						$d_1 \geq d_2$							
kqnkr	38%	10%	0%	0%	0%	0%	–	5%	1%	0%	6%	0%	0%	60%
kqkqr	–	–	–	–	–	–	–	1%	18%	17%	5%	3%	17%	61%
kbpkp	34%	8%	2%	5%	5%	2%	2%	1%	0%	0%	2%	0%	0%	61%
krbkp	52%	6%	0%	0%	1%	0%	0%	0%	0%	0%	2%	0%	0%	61%
kqrpk	62%	0%	–	–	0%	–	–	0%	–	–	0%	–	–	62%
kbnnk	60%	3%	–	–	0%	–	–	0%	–	–	0%	–	–	62%
kbnkp	38%	11%	2%	4%	2%	2%	1%	0%	0%	0%	2%	0%	0%	63%
krnkp	51%	7%	1%	1%	1%	0%	0%	0%	0%	0%	2%	0%	0%	63%
kqbkb	50%	5%	–	–	0%	–	–	3%	–	–	5%	0%	0%	63%
kqkrn	–	–	–	–	–	–	–	28%	18%	4%	7%	5%	2%	63%
kqkqb	–	–	–	–	–	–	–	5%	18%	19%	5%	11%	5%	64%
kqnkn	55%	4%	–	–	–	–	–	0%	–	–	4%	1%	–	64%
kqkqp	–	–	–	–	–	–	–	3%	24%	21%	6%	6%	4%	64%
kqrnk	64%	0%	–	–	–	–	–	–	–	–	–	–	–	64%
kbbnk	62%	2%	–	–	0%	–	–	0%	–	–	–	–	–	64%
kqbkn	57%	3%	–	–	0%	–	–	0%	–	–	4%	1%	–	65%
kqrbk	65%	0%	–	–	–	–	–	–	–	–	–	–	–	65%
kqnkb	51%	7%	–	–	0%	–	–	3%	0%	–	5%	0%	–	66%
kqnkp	60%	4%	0%	0%	0%	0%	–	0%	0%	0%	2%	0%	0%	66%
krppk	66%	0%	–	–	0%	–	–	0%	–	–	0%	–	–	66%
krnnk	64%	2%	–	–	0%	–	–	0%	–	–	0%	–	–	66%
krbkq	0%	3%	6%	6%	13%	13%	3%	1%	3%	3%	7%	3%	5%	67%
kqbbk	67%	0%	–	–	–	–	–	0%	–	–	0%	–	–	67%
krbbk	65%	2%	–	–	0%	–	–	0%	–	–	0%	–	–	67%
kqbkp	63%	2%	0%	0%	0%	–	–	0%	0%	0%	2%	0%	0%	67%
knppk	67%	0%	–	–	0%	–	–	0%	–	–	0%	0%	–	67%
krpkr	10%	8%	2%	6%	21%	1%	0%	2%	2%	2%	8%	5%	0%	68%
kbppk	67%	0%	–	–	0%	–	–	0%	–	–	0%	0%	–	68%
krpkq	0%	2%	3%	13%	1%	4%	18%	1%	5%	1%	8%	1%	10%	68%
kqnnk	68%	0%	–	–	–	–	–	0%	–	–	0%	–	–	68%
krnkq	0%	2%	5%	9%	11%	12%	6%	1%	3%	3%	7%	2%	7%	69%
krrpk	69%	0%	–	–	–	–	–	0%	–	–	0%	–	–	69%
kbnpk	69%	0%	–	–	0%	–	–	0%	–	–	0%	–	–	69%
kbpkq	0%	0%	1%	7%	1%	5%	28%	0%	3%	1%	7%	2%	13%	69%
kbnkq	0%	0%	2%	10%	0%	4%	33%	0%	3%	1%	7%	1%	11%	70%
krnpk	71%	0%	–	–	0%	–	–	0%	–	–	0%	–	–	71%
knpkq	0%	0%	1%	6%	1%	6%	30%	0%	2%	1%	7%	2%	16%	71%
kqnpk	71%	0%	–	–	0%	–	–	0%	–	–	0%	–	–	71%
kqkrb	–	–	–	–	–	–	–	30%	19%	7%	7%	6%	2%	71%
krbpk	72%	0%	–	–	0%	–	–	0%	–	–	0%	–	–	72%
kqbkq	0%	13%	7%	4%	29%	1%	0%	2%	3%	2%	6%	5%	0%	72%
krbnk	74%	0%	–	–	0%	–	–	–	–	–	–	–	–	74%

continued on next page

	WWW $d_1 \geq d_2$	WDD	WDL	WLL	DDD	DLL	LLL $d_1 \geq d_2$	WD	WL	DL	W	D	L	Sum
kqbpk	74%	0%	–	–	–	–	–	0%	–	–	0%	–	–	74%
kqpkq	2%	13%	5%	11%	15%	1%	0%	4%	7%	3%	7%	6%	0%	74%
kbpkn	15%	15%	0%	0%	36%	0%	–	2%	0%	0%	3%	4%	0%	74%
krrkq	–	–	–	–	–	–	–	18%	28%	14%	7%	4%	3%	75%
kqbnk	75%	0%	–	–	0%	–	–	–	–	–	–	–	–	75%
krrbk	76%	0%	–	–	0%	–	–	0%	–	–	0%	–	–	76%
kqkrr	–	–	–	–	–	–	–	17%	25%	17%	5%	6%	6%	76%
krrnk	76%	0%	–	–	–	–	–	0%	–	–	–	–	–	77%
kqqbk	75%	2%	–	–	–	–	–	0%	–	–	0%	–	–	77%
knpkn	12%	14%	0%	0%	42%	0%	–	1%	0%	0%	2%	5%	0%	77%
kqqpk	75%	3%	–	–	–	–	–	0%	–	–	0%	–	–	78%
kqnkq	0%	10%	7%	6%	32%	1%	0%	2%	4%	3%	6%	5%	0%	78%
kbpkr	1%	13%	1%	0%	38%	7%	0%	3%	0%	3%	4%	7%	1%	79%
krbkr	1%	18%	0%	0%	49%	0%	0%	2%	0%	0%	4%	5%	0%	79%
knpkr	1%	12%	1%	1%	37%	8%	1%	2%	0%	2%	4%	9%	2%	80%
kbpkb	6%	15%	0%	–	47%	0%	–	2%	0%	0%	4%	6%	0%	80%
knpkb	6%	14%	0%	–	49%	–	–	2%	0%	0%	4%	6%	0%	81%
kqqnk	79%	2%	–	–	–	–	–	0%	–	–	–	–	–	81%
krnkr	1%	17%	0%	0%	51%	0%	0%	2%	0%	1%	4%	6%	0%	83%
kbnkr	0%	15%	0%	0%	60%	0%	0%	2%	0%	1%	4%	6%	0%	89%
kbnkn	2%	19%	0%	–	62%	–	–	1%	0%	0%	1%	6%	–	90%
kbnkb	0%	16%	0%	–	66%	–	–	1%	–	0%	3%	5%	0%	92%
kkbbb	–	–	–	–	–	–	–	–	–	–	–	31%	69%	100%
kkbbn	–	–	–	–	–	–	–	–	–	–	–	4%	96%	100%
kkbbp	–	–	–	–	–	–	–	–	–	–	–	7%	93%	100%
kkbnn	–	–	–	–	–	–	–	–	–	–	–	8%	92%	100%
kkbnp	–	–	–	–	–	–	–	–	–	–	–	1%	99%	100%
kkbpp	–	–	–	–	–	–	–	–	–	–	–	1%	99%	100%
kknnn	–	–	–	–	–	–	–	–	–	–	–	25%	75%	100%
kknnp	–	–	–	–	–	–	–	–	–	–	–	12%	88%	100%
kknpp	–	–	–	–	–	–	–	–	–	–	–	1%	99%	100%
kkppp	–	–	–	–	–	–	–	–	–	–	–	1%	99%	100%
kkqbb	–	–	–	–	–	–	–	–	–	–	–	4%	96%	100%
kkqbn	–	–	–	–	–	–	–	–	–	–	–	0%	100%	100%
kkqbp	–	–	–	–	–	–	–	–	–	–	–	0%	100%	100%
kkqnn	–	–	–	–	–	–	–	–	–	–	–	8%	92%	100%
kkqnp	–	–	–	–	–	–	–	–	–	–	–	0%	100%	100%
kkqpp	–	–	–	–	–	–	–	–	–	–	–	0%	100%	100%
kkqqb	–	–	–	–	–	–	–	–	–	–	–	–	100%	100%
kkqqn	–	–	–	–	–	–	–	–	–	–	–	–	100%	100%
kkqqp	–	–	–	–	–	–	–	–	–	–	–	–	100%	100%
kkqqq	–	–	–	–	–	–	–	–	–	–	–	–	100%	100%

continued on next page

	WWW $d_1 \geq d_2$	WDD	WDL	WLL	DDD	DLL	LLL $d_1 \geq d_2$	WD	WL	DL	W	D	L	Sum
kkqqr	–	–	–	–	–	–	–	–	–	–	–	–	100%	100%
kkqrb	–	–	–	–	–	–	–	–	–	–	–	–	100%	100%
kkqrn	–	–	–	–	–	–	–	–	–	–	–	–	100%	100%
kkqrp	–	–	–	–	–	–	–	–	–	–	–	0%	100%	100%
kkqrr	–	–	–	–	–	–	–	–	–	–	–	–	100%	100%
kkrbb	–	–	–	–	–	–	–	–	–	–	–	4%	96%	100%
kkrbn	–	–	–	–	–	–	–	–	–	–	–	0%	100%	100%
kkrbp	–	–	–	–	–	–	–	–	–	–	–	0%	100%	100%
kkrnn	–	–	–	–	–	–	–	–	–	–	–	8%	92%	100%
kkrnp	–	–	–	–	–	–	–	–	–	–	–	0%	100%	100%
kkrpp	–	–	–	–	–	–	–	–	–	–	–	0%	100%	100%
kkrrb	–	–	–	–	–	–	–	–	–	–	–	–	100%	100%
kkrrn	–	–	–	–	–	–	–	–	–	–	–	–	100%	100%
kkrrp	–	–	–	–	–	–	–	–	–	–	–	0%	100%	100%
kkrrr	–	–	–	–	–	–	–	–	–	–	–	–	100%	100%
Mean	15%	2%	0%	0%	3%	0%	1%	3%	2%	4%	2%	4%	21%	57%

Non-Cordel Instances

	WWW $d_1 < d_2$	WWD	WWL	DDL	LLL $d_1 < d_2$	WW	DD	LL	Sum
kpppk	–	–	–	–	–	100%	0%	–	100%
krrrk	–	–	–	–	–	99%	–	–	99%
kqqkp	–	–	–	–	–	98%	0%	–	98%
kbbbk	–	–	–	–	–	71%	26%	–	97%
kqqkn	–	–	–	–	–	97%	–	–	97%
kqqkb	–	–	–	–	–	97%	–	–	97%
kqqkr	–	–	–	–	–	95%	0%	–	95%
knnkn	–	–	–	–	–	0%	93%	–	93%
krrkp	–	–	–	–	–	92%	0%	–	92%
knnkb	–	–	–	–	–	0%	91%	–	91%
kqqqk	–	–	–	–	–	89%	–	–	89%
krrkn	–	–	–	–	–	87%	0%	–	87%
kbknn	–	–	–	–	–	–	86%	–	86%
knnkr	–	–	–	–	–	0%	85%	0%	85%
knknn	–	–	–	–	–	–	85%	0%	85%
kbbkn	–	–	–	–	–	36%	48%	–	84%
krknn	–	–	–	–	–	1%	81%	–	82%
kbbkp	–	–	–	–	–	37%	43%	2%	81%
krrkb	–	–	–	–	–	80%	0%	–	81%
kbbkb	–	–	–	–	–	2%	77%	–	80%

continued on next page

	WWW $d_1 < d_2$	WWD	WWL	DDL	LLL $d_1 < d_2$	WW	DD	LL	Sum
knnnk	–	–	–	–	–	77%	1%	–	78%
kbbkr	–	–	–	–	–	2%	76%	0%	78%
kppkp	–	–	–	–	–	58%	6%	13%	77%
kqkpp	–	–	–	–	–	76%	1%	0%	77%
kqknp	–	–	–	–	–	75%	1%	0%	77%
knkpp	–	–	–	–	–	0%	40%	36%	76%
knnkp	–	–	–	–	–	15%	58%	2%	76%
kbkpp	–	–	–	–	–	0%	51%	24%	75%
krkbb	–	–	–	–	–	0%	74%	0%	75%
kppkn	–	–	–	–	–	43%	31%	0%	75%
kbknp	–	–	–	–	–	0%	64%	10%	74%
knknp	–	–	–	–	–	0%	53%	20%	73%
kqrkq	29%	16%	16%	0%	0%	9%	1%	0%	71%
knkbb	–	–	–	–	–	–	43%	27%	70%
kbkbb	–	–	–	–	–	–	70%	0%	70%
krkbn	–	–	–	–	–	0%	69%	0%	70%
kqknn	–	–	–	–	–	66%	4%	–	69%
kbkbn	–	–	–	–	–	–	69%	0%	69%
kppkb	–	–	–	–	–	28%	40%	–	69%
kqkbp	–	–	–	–	–	67%	1%	0%	68%
kqkbn	–	–	–	–	–	68%	0%	0%	68%
knkbp	–	–	–	–	–	0%	46%	23%	68%
kbkbp	–	–	–	–	–	0%	57%	11%	68%
krkpp	–	–	–	–	–	51%	10%	6%	68%
kpkpp	–	–	–	–	–	14%	6%	47%	67%
knkrp	–	–	–	–	–	0%	5%	62%	67%
knkbn	–	–	–	–	–	–	63%	2%	65%
knkqp	–	–	–	–	–	0%	1%	64%	65%
kbkqp	–	–	–	–	–	–	1%	64%	65%
kbkrp	–	–	–	–	–	–	7%	58%	65%
krrkr	–	–	–	–	–	62%	0%	–	63%
kpknn	–	–	–	–	–	4%	49%	9%	63%
kpknp	–	–	–	–	–	10%	10%	42%	62%
kqqkq	–	–	–	–	–	61%	0%	0%	62%
kpkrp	–	–	–	–	–	1%	0%	60%	61%
knkrn	–	–	–	–	–	–	3%	58%	61%
kqkbb	–	–	–	–	–	59%	1%	0%	61%
knkrr	–	–	–	–	–	–	1%	59%	60%
kbkrr	–	–	–	–	–	–	2%	58%	60%
krkqq	–	–	–	–	–	–	0%	60%	60%
knkrb	–	–	–	–	–	–	3%	57%	60%
kpkbp	–	–	–	–	–	5%	7%	48%	60%

continued on next page

	WWW $d_1 < d_2$	WWD	WWL	DDL	LLL $d_1 < d_2$	WW	DD	LL	Sum
krkqp	–	–	–	–	–	1%	0%	58%	59%
kqrkr	40%	12%	0%	0%	–	7%	0%	0%	59%
kbkqn	–	–	–	–	–	–	1%	58%	59%
kbkrn	–	–	–	–	–	–	5%	54%	59%
kbkqr	–	–	–	–	–	–	1%	58%	59%
knkqn	–	–	–	–	–	–	1%	58%	58%
kpkbb	–	–	–	–	–	2%	31%	25%	58%
krknp	–	–	–	–	–	9%	48%	1%	58%
kbbkq	–	–	–	–	–	1%	1%	55%	58%
kbkrb	–	–	–	–	–	–	4%	53%	57%
knkqr	–	–	–	–	–	–	0%	56%	56%
kpkqp	–	–	–	–	–	0%	0%	56%	56%
krkrn	–	–	–	–	–	0%	56%	1%	56%
kbkqb	–	–	–	–	–	–	1%	55%	56%
knkqb	–	–	–	–	–	–	1%	55%	56%
kbkqq	–	–	–	–	–	–	0%	56%	56%
kpkrn	–	–	–	–	–	0%	2%	54%	56%
krkqr	–	–	–	–	–	–	1%	54%	55%
kqrrk	51%	3%	–	–	–	0%	–	–	55%
kppkr	–	–	–	–	–	9%	10%	36%	55%
krkbp	–	–	–	–	–	3%	49%	2%	55%
kpkbn	–	–	–	–	–	4%	4%	47%	54%
kpkrb	–	–	–	–	–	0%	1%	52%	54%
kqqrk	47%	7%	–	–	–	0%	–	–	54%
kpkrr	–	–	–	–	–	0%	0%	53%	54%
krkrp	–	–	–	–	–	2%	30%	21%	53%
krkqn	–	–	–	–	–	0%	2%	51%	53%
krkqb	–	–	–	–	–	0%	2%	51%	53%
kqkrp	–	–	–	–	–	49%	3%	1%	53%
knnkq	–	–	–	–	–	–	8%	45%	53%
krkrb	–	–	–	–	–	0%	51%	1%	52%
kppkq	–	–	–	–	–	1%	1%	50%	52%
knkqq	–	–	–	–	–	–	–	51%	51%
krkrr	–	–	–	–	–	–	4%	47%	51%
kpkqn	–	–	–	–	–	0%	0%	50%	50%
kqrkb	37%	7%	–	–	–	6%	0%	–	49%
kpkqr	–	–	–	–	–	0%	0%	49%	49%
kpkqb	–	–	–	–	–	0%	0%	48%	48%
krpkb	21%	17%	0%	0%	–	8%	1%	–	47%
kqrkn	40%	2%	–	–	–	4%	0%	–	47%
krpkn	27%	12%	0%	0%	–	6%	1%	–	46%
kqrkp	44%	0%	0%	–	–	1%	0%	–	46%

continued on next page

	WWW $d_1 < d_2$	WWD	WWL	DDL	LLL $d_1 < d_2$	WW	DD	LL	Sum
kqpkp	35%	2%	0%	0%	–	7%	0%	–	45%
kqbkr	23%	16%	0%	0%	–	5%	0%	0%	45%
kqpkn	33%	3%	0%	–	–	9%	0%	–	45%
kqkqq	–	–	–	–	–	0%	2%	42%	45%
knnpk	30%	8%	–	–	–	5%	1%	–	43%
kqpkb	29%	3%	0%	–	–	10%	0%	–	43%
krpkp	30%	3%	2%	0%	0%	6%	0%	0%	41%
kbbpk	32%	3%	–	–	–	6%	0%	–	41%
krbkn	29%	9%	–	–	–	2%	0%	–	41%
kpkqq	–	–	–	–	–	–	0%	41%	41%
krnkb	23%	13%	–	–	–	3%	1%	–	40%
kqkqn	–	–	–	–	–	2%	37%	1%	40%
knpkp	17%	11%	1%	2%	4%	4%	1%	0%	40%
krbkb	23%	13%	–	–	–	3%	1%	–	40%
kqpkr	23%	3%	3%	0%	0%	11%	0%	0%	40%
krnkn	28%	10%	–	–	–	2%	0%	–	40%
kqppk	39%	0%	–	–	–	0%	–	–	40%
kqnkr	22%	12%	1%	0%	–	5%	0%	0%	40%
kqkqr	–	–	–	–	–	1%	2%	37%	39%
kbpkp	19%	10%	1%	1%	2%	4%	1%	0%	39%
krbkp	28%	9%	1%	0%	0%	1%	0%	0%	39%
kqrpk	37%	1%	–	–	–	0%	–	–	38%
kbnnk	35%	2%	–	–	–	0%	0%	–	38%
kbnkp	24%	7%	3%	1%	2%	1%	0%	0%	37%
krnkp	28%	8%	1%	0%	0%	1%	0%	0%	37%
kqbkb	25%	8%	–	–	–	4%	0%	–	37%
kqkrn	–	–	–	–	–	26%	11%	0%	37%
kqkqb	–	–	–	–	–	2%	32%	2%	36%
kqnkn	28%	4%	–	–	–	3%	0%	–	36%
kqkqp	–	–	–	–	–	2%	24%	9%	36%
kqrnk	35%	1%	–	–	–	0%	–	–	36%
kbbnk	34%	2%	–	–	–	0%	0%	–	36%
kqbkn	27%	4%	–	–	–	3%	0%	–	35%
kqrbk	34%	1%	–	–	–	0%	–	–	35%
kqnkb	24%	7%	–	–	–	4%	0%	–	34%
kqnkp	27%	5%	0%	0%	–	1%	0%	–	34%
krppk	33%	0%	–	–	–	0%	0%	–	34%
krnnk	31%	3%	–	–	–	0%	0%	–	34%
krbkq	0%	2%	7%	14%	2%	3%	2%	3%	33%
kqbbk	32%	2%	–	–	–	0%	–	–	33%
krbbk	32%	1%	–	–	–	0%	0%	–	33%
kqbkp	26%	5%	0%	–	–	1%	0%	–	33%

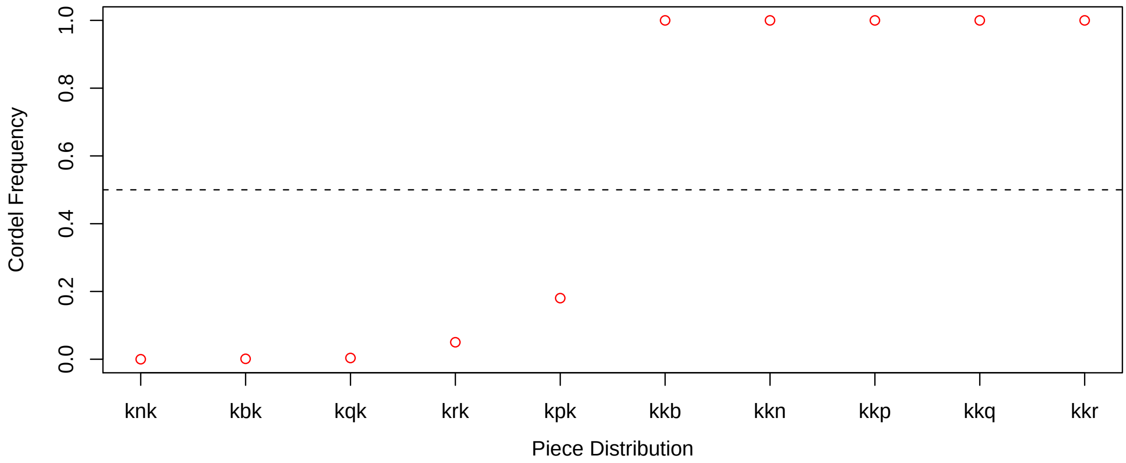
continued on next page

	WWW $d_1 < d_2$	WWD	WWL	DDL	LLL $d_1 < d_2$	WW	DD	LL	Sum
knppk	32%	1%	–	–	–	0%	0%	–	33%
krpkr	10%	12%	2%	1%	0%	4%	3%	0%	32%
kbppk	31%	0%	–	–	–	1%	0%	–	32%
krpkq	0%	1%	3%	3%	13%	2%	1%	9%	32%
kqnnk	31%	1%	–	–	–	0%	–	–	32%
krnkq	0%	1%	5%	12%	6%	2%	1%	4%	31%
krrpk	24%	0%	–	–	–	7%	0%	–	31%
kbnpk	31%	0%	–	–	–	0%	0%	–	31%
kbpkq	0%	0%	1%	2%	15%	1%	0%	11%	31%
kbnkq	0%	0%	2%	1%	17%	2%	0%	8%	30%
krnpk	29%	0%	–	–	–	0%	0%	–	29%
knpkq	0%	0%	1%	2%	15%	1%	0%	10%	29%
kqnpk	28%	1%	–	–	–	0%	–	–	29%
kqkrb	–	–	–	–	–	15%	13%	1%	29%
krbpk	28%	0%	–	–	–	0%	0%	–	28%
kqbkq	0%	9%	7%	3%	0%	4%	5%	0%	28%
krbnk	26%	0%	–	–	–	0%	–	–	26%
kqbpk	25%	1%	–	–	–	0%	–	–	26%
kqpkq	3%	9%	2%	1%	0%	5%	5%	0%	26%
kbpkn	10%	10%	0%	0%	–	2%	4%	–	26%
krrkq	–	–	–	–	–	5%	19%	2%	25%
kqbnk	24%	1%	–	–	–	0%	–	–	25%
krrbk	23%	1%	–	–	–	0%	–	–	24%
kqkrr	–	–	–	–	–	5%	14%	5%	24%
krrnk	23%	1%	–	–	–	0%	–	–	23%
kqqbk	18%	4%	–	–	–	0%	–	–	23%
knpkn	9%	8%	0%	0%	–	1%	4%	–	23%
kqqpk	14%	2%	–	–	–	7%	–	–	22%
kqnkq	0%	6%	5%	2%	0%	4%	5%	0%	22%
kbpkr	1%	6%	0%	6%	0%	1%	7%	1%	21%
krbkr	1%	14%	0%	0%	0%	1%	5%	0%	21%
knpkr	1%	5%	0%	7%	0%	1%	6%	1%	20%
kbpkb	5%	8%	0%	0%	–	1%	6%	–	20%
knpkb	5%	7%	0%	0%	–	1%	6%	–	19%
kqqnk	15%	3%	–	–	–	0%	–	–	19%
krnkr	1%	10%	0%	0%	0%	1%	5%	0%	17%
kbnkr	0%	4%	0%	1%	0%	1%	6%	0%	11%
kbnkn	1%	8%	–	–	–	0%	1%	–	10%
kbnkb	0%	4%	–	0%	–	1%	3%	–	8%
kkbbb	–	–	–	–	–	–	–	–	–
kkbbn	–	–	–	–	–	–	–	–	–
kkbbp	–	–	–	–	–	–	–	–	–

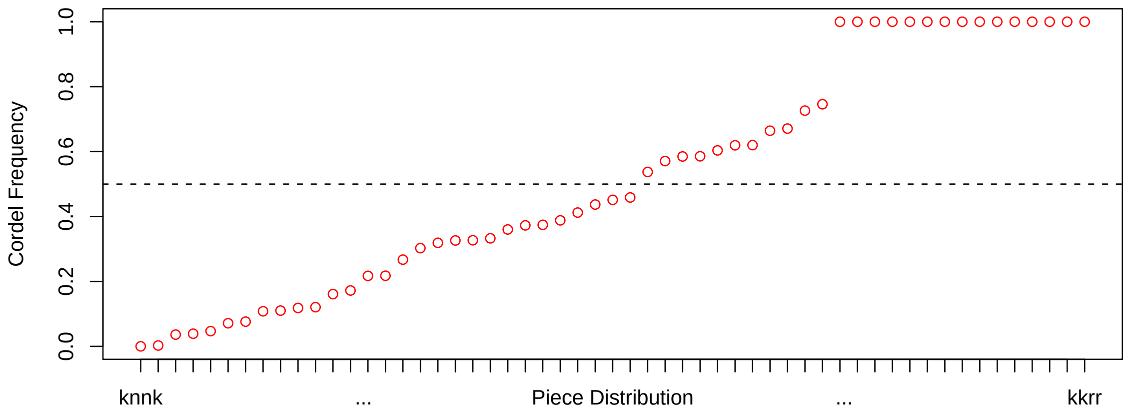
continued on next page

	WWW $d_1 < d_2$	WWD	WWL	DDL	LLL $d_1 < d_2$	WW	DD	LL	Sum
kkbnn	–	–	–	–	–	–	–	–	–
kkbnp	–	–	–	–	–	–	–	–	–
kkbpp	–	–	–	–	–	–	–	–	–
kknnn	–	–	–	–	–	–	–	–	–
kknnp	–	–	–	–	–	–	–	–	–
kknpp	–	–	–	–	–	–	–	–	–
kkppp	–	–	–	–	–	–	–	–	–
kkqbb	–	–	–	–	–	–	–	–	–
kkqbn	–	–	–	–	–	–	–	–	–
kkqbp	–	–	–	–	–	–	–	–	–
kkqnn	–	–	–	–	–	–	–	–	–
kkqnp	–	–	–	–	–	–	–	–	–
kkqpp	–	–	–	–	–	–	–	–	–
kkqqb	–	–	–	–	–	–	–	–	–
kkqqn	–	–	–	–	–	–	–	–	–
kkqqp	–	–	–	–	–	–	–	–	–
kkqqq	–	–	–	–	–	–	–	–	–
kkqqr	–	–	–	–	–	–	–	–	–
kkqrb	–	–	–	–	–	–	–	–	–
kkqrn	–	–	–	–	–	–	–	–	–
kkqrp	–	–	–	–	–	–	–	–	–
kkqrr	–	–	–	–	–	–	–	–	–
kkrbb	–	–	–	–	–	–	–	–	–
kkrbn	–	–	–	–	–	–	–	–	–
kkrbp	–	–	–	–	–	–	–	–	–
kkrnn	–	–	–	–	–	–	–	–	–
kkrnp	–	–	–	–	–	–	–	–	–
kkrpp	–	–	–	–	–	–	–	–	–
kkrrb	–	–	–	–	–	–	–	–	–
kkrrn	–	–	–	–	–	–	–	–	–
kkrrp	–	–	–	–	–	–	–	–	–
kkrrr	–	–	–	–	–	–	–	–	–
Mean	8%	2%	0%	0%	0%	10%	10%	12%	43%

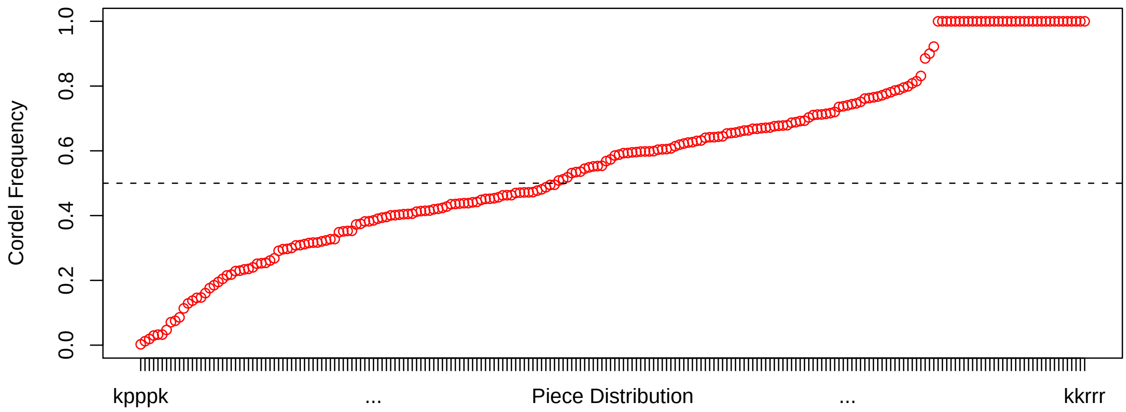
F.4 Figures



(a) Endgames with 4 pieces (including kings)

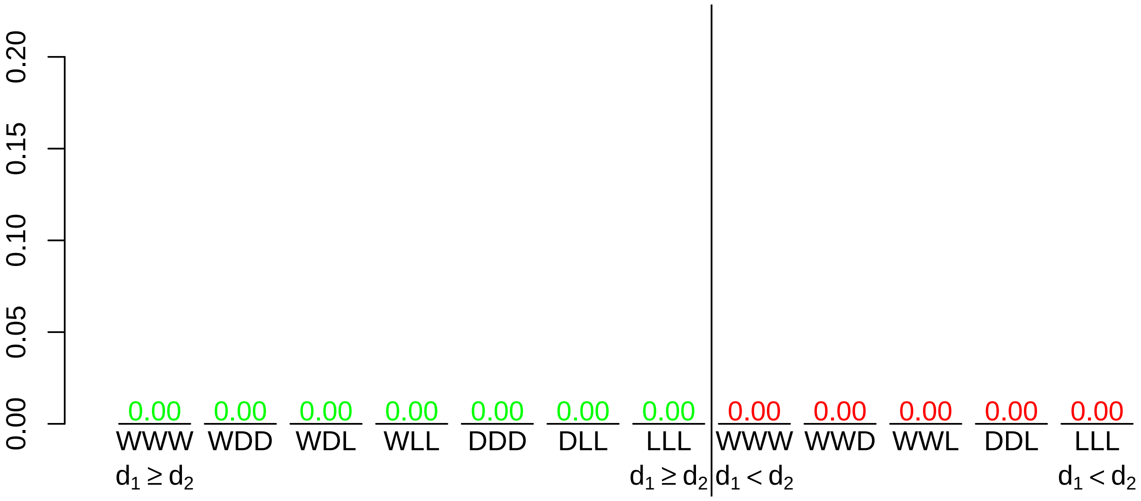


(b) Endgames with 4 pieces (including kings)

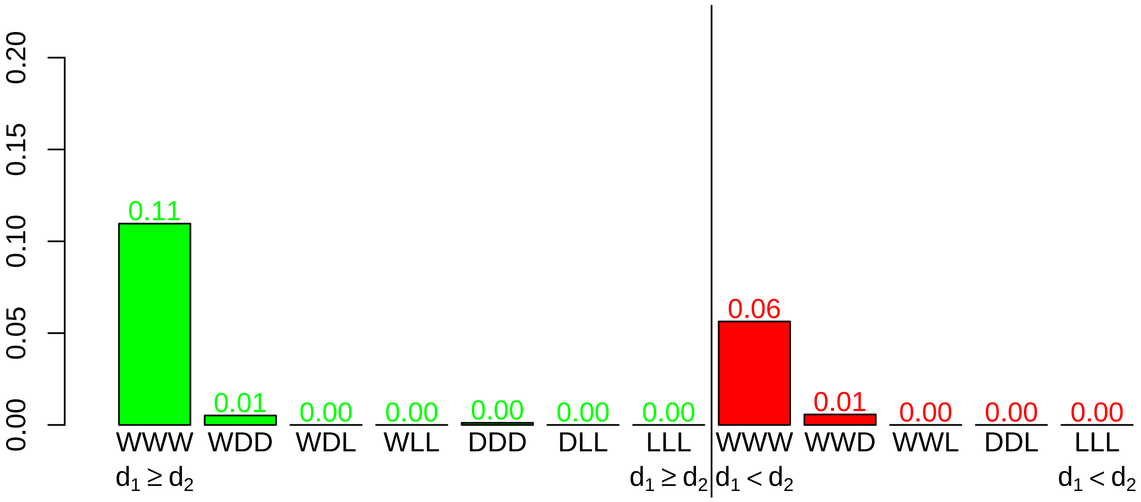


(c) Endgames with 5 pieces (including kings)

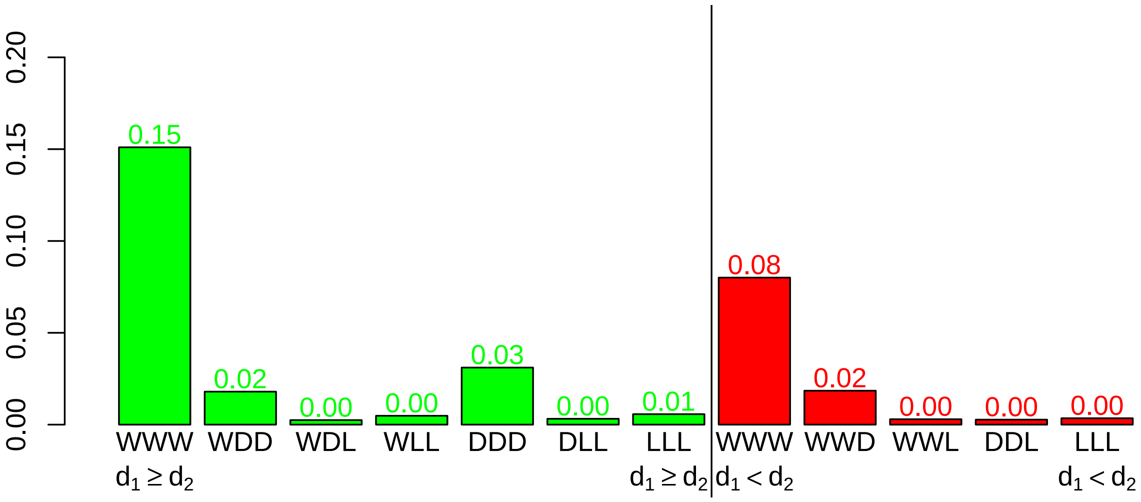
Figure F.4.1: Cordel Frequencies for different piece configurations



(a) Endgames with 3 pieces (including kings)

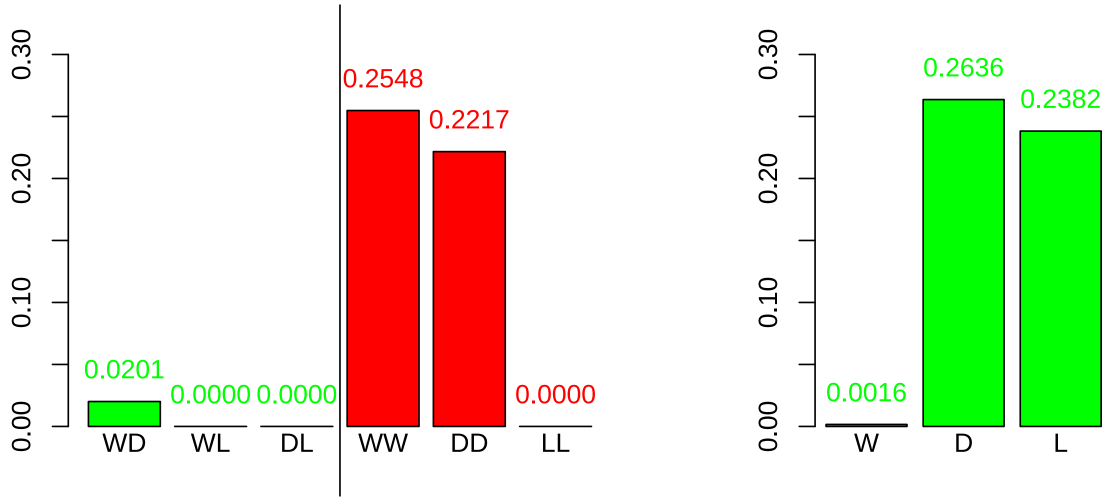


(b) Endgames with 4 pieces (including kings)

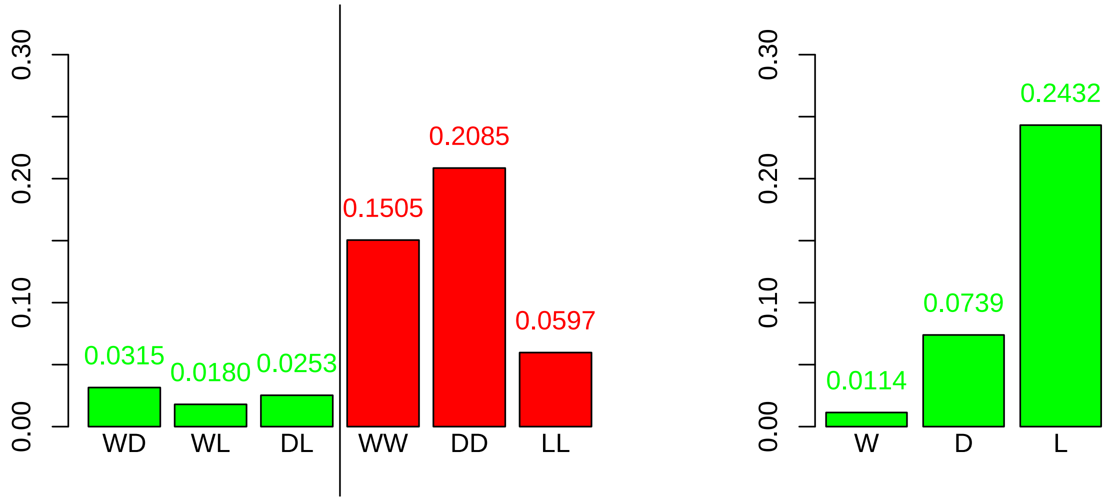


(c) Endgames with 5 pieces (including kings)

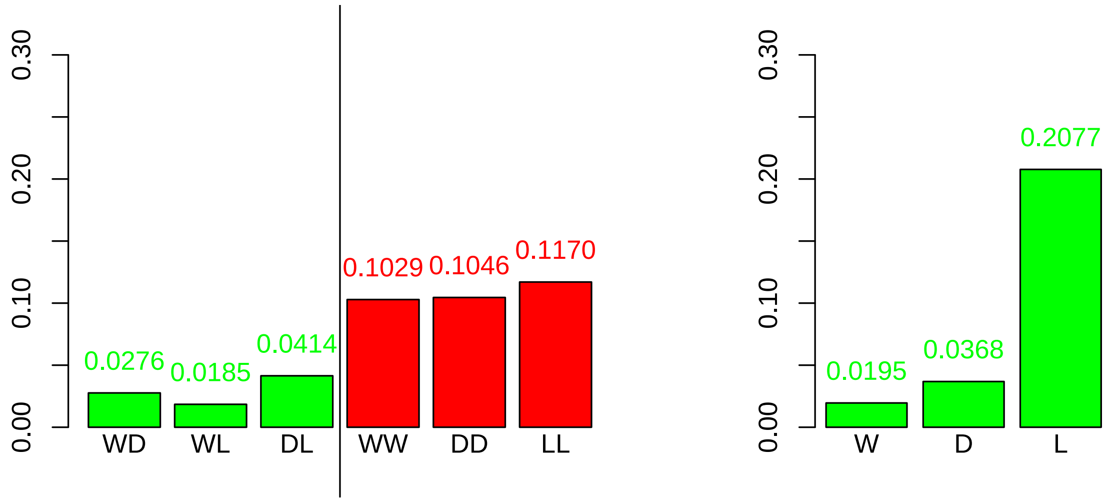
Figure F.4.2: Relative frequencies of different types of chess positions in endgames with 3, 4, or 5 pieces (including kings) with at least three feasible moves.



(a) Endgames with 3 pieces (including kings)



(b) Endgames with 4 pieces (including kings)



(c) Endgames with 5 pieces (including kings)

Figure F.4.3: Relative frequencies of different types of chess positions in endgames with 3, 4, or 5 pieces (including kings) with only two (left) or one (right) feasible moves.

F.5 Monte Carlo Samples for Endgames with 6 Pieces

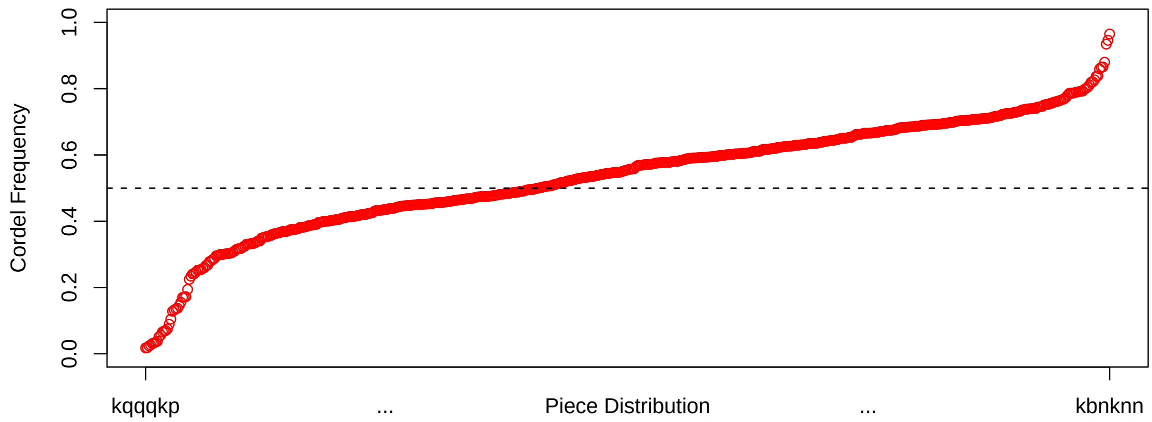
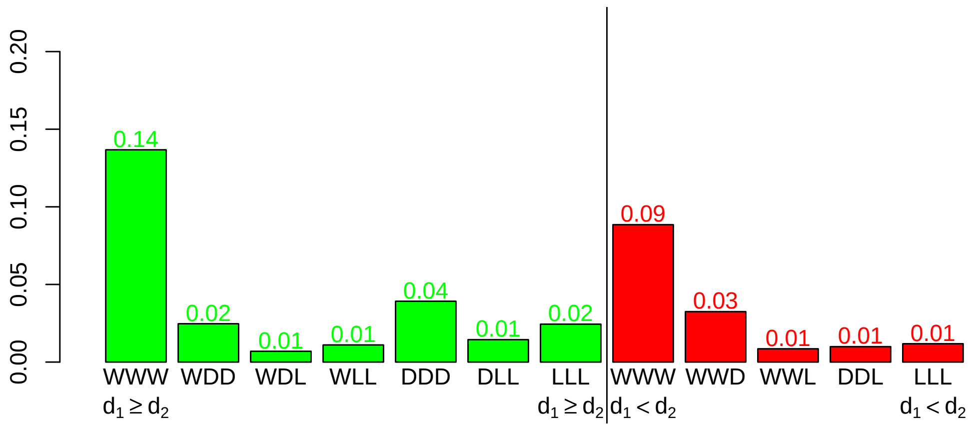
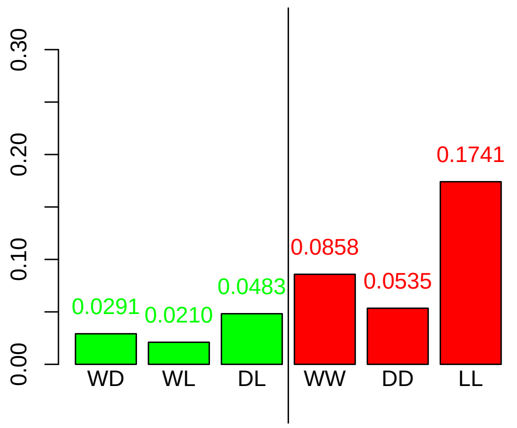


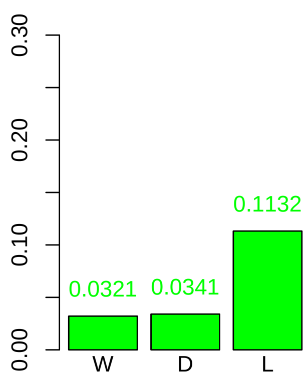
Figure F.5.1: Cordel Frequencies for the 515 different piece configurations.



(a) Endgames with at least 3 feasible moves.



(b) Endgames with exactly 2 feasible moves.



(c) Endgames with exactly 1 feasible move.

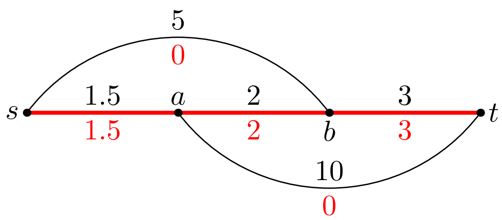
Figure F.5.2: Relative frequencies of different types of chess positions in endgames with 6 pieces (including kings) with only two (left) or one (right) feasible moves.

Appendix G

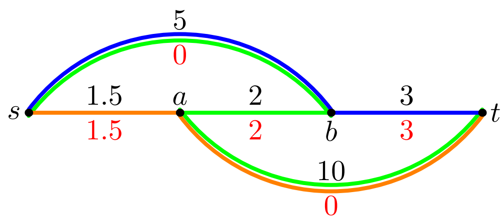
Two Detailed Shortest Path Examples for the Penalty Method

Refers to page 28.

Example G.1
We consider the shortest $s - t$ path problem on the undirected graph shown in Figure G.1-(a) where the path $s - a - b - t$ (marked in red) is the optimal solution.



(a) Optimal solution $s - a - b - t$.



(b) Remaining simple $s - t$ paths (paths without cycles): $s - b - t$, $s - a - t$, and $s - b - a - t$.

Figure G.1: Graph with weights $w(\cdot)$ (values in black) and penalties $p(\cdot)$ (values in red) for each edge.

Again we use a weight vector w representing the weights of the edges in order to transform the optimization problem in the requested form $\min_{B \in S} w'B$.

$$w = [\underbrace{1.5}_{=w(s-a)}, \underbrace{5}_{=w(s-b)}, \underbrace{2}_{=w(a-b)}, \underbrace{10}_{=w(a-t)}, \underbrace{3}_{=w(b-t)}]$$

Thus the optimal path $s - a - b - t$ is represented by

$$B^{(0)} = [\underbrace{1}_{(s-a)}, 0, \underbrace{1}_{(a-b)}, 0, \underbrace{1}_{(b-t)}].$$

Therewith we can specify the canonical penalty vector which is

$$p = [\underbrace{1.5}_{w(s-a)}, 0, \underbrace{2}_{w(a-b)}, 0, \underbrace{3}_{w(b-t)}].$$

The canonical penalty vector p is also shown in Figure G.1 represented by the red values.

Now we consider all $s-t$ paths without cycles which are $s-a-b-t$, $s-b-t$, $s-a-t$, and $s-b-a-t$ (c.f. Figure G.1-(a) and (b)). With the penalty vector p we can compute the penalized part $p(\cdot)$ for each of the four paths. Together with the weights $w(\cdot)$ we can compute $f_\varepsilon(\cdot) = w(\cdot) + \varepsilon \cdot p(\cdot)$ depending on ε. The following chart shows for each of the four chosen paths it's weight $w(\cdot)$, it's punished part $p(\cdot)$ and it's penalized value $f_\varepsilon(\cdot)$.

path P	$w(P)$	$p(P)$	$f_\varepsilon(P)$
$s-a-b-t$	6.5	6.5	$6.5 + 6.5 \cdot \varepsilon$
$s-b-t$	8	3	$8 + 3 \cdot \varepsilon$
$s-a-t$	11.5	1.5	$11.5 + 1.5 \cdot \varepsilon$
$s-b-a-t$	17	2	$17 + 2 \cdot \varepsilon$

As we can see, the penalized values $f_\varepsilon(\cdot)$ are linear in ε for each path. Figure G.2 shows these linear functions. The bold lines indicate for each ε the smallest punished value which implicates that the related path is a penalty alternative for this penalty parameter ε.

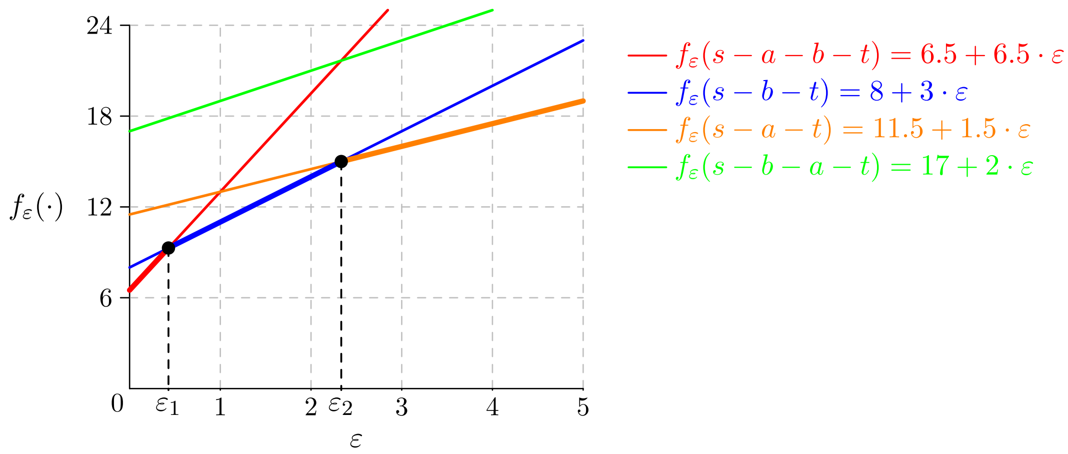


Figure G.2: Penalized values $f_\varepsilon(P)$ of the four paths $s-a-b-t$, $s-b-t$, $s-a-t$, and $s-b-a-t$

We summarize without proof:

1. *For each ε in the interval $\left[0, \varepsilon_1 = \frac{3}{7} \approx 0.43\right]$ the optimal solution $s-a-b-t$ is an ε-penalty alternative.*

2. *For each ε in the interval $\left[\varepsilon_1 = \frac{3}{7} \approx 0.43, \varepsilon_2 = \frac{7}{3} \approx 2.33\right]$ $s-b-t$ is an ε-penalty alternative.*

3. *For each ε in the interval $\left[\varepsilon_2 = \frac{7}{3} \approx 2.33, \infty\right]$ $s-a-t$ is an ε-penalty alternative.*

   Note that the intervall $\left[\varepsilon_2 = \frac{7}{3} \approx 2.33, \infty\right]$ covers each $\varepsilon \geq \frac{7}{3}$ as well as $\varepsilon = \infty$ as penalty parameter.

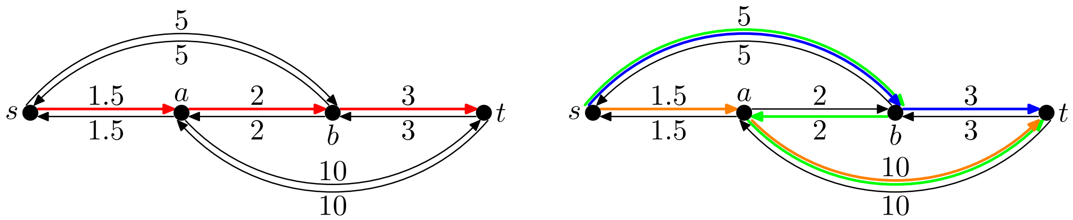
4. *There exists no $0 \leq \varepsilon \leq \infty$ for which the path $s-b-a-t$ is an ε-penalty alternative. Furthermore it can be shown that the unconsidered paths which contain cycles (e.g. $s-b-a-s-b-t$) are for no $0 \leq \varepsilon \leq \infty$ penalty alternative, too.*

Thus we can decompose the interval $[0,\infty]$ into three subintervals $[0,\varepsilon_1]$, $[\varepsilon_1,\varepsilon_2]$, $[\varepsilon_2,\infty]$ and for each of these three intervals we have one representative which is optimal for every ε in the subinterval.

In Section 2.3 we will have a deeper look at this interval representation. Besides we will show how to compute the so called threshold parameters $\varepsilon_1 = \frac{3}{7}$ and $\varepsilon_2 = \frac{7}{3}$ which provide the interval decomposition.

The next Example G.2 points out that the penalty alternatives also depend on the graph representation.

Example G.2
*In contrast to Example G.1 we now represent the graph shown in Figure G.1-(a) as a **directed** graph. Therefore we represent each undirected edge $\{u,v\}$ as the directed edge (u,v) and the directed edge in the opposite direction whereas both edges have the same weight as $\{u,v\}$. Therewith we get the directed graph shown in Figure G.3. Of course, $s-a-b-t$ is still the shortest $s-t$ path, because we changed only the representation of the optimization problem.*



(a) Optimal solution $s-a-b-t$.

(b) Remaining simple $s-t$ paths: $s-b-t$, $s-a-t$, and $s-b-a-t$.

Figure G.3: Representation of the graph from Example G.1 as directed graph. Again the red marked path $s-a-b-t$ is the shortest $s-t$ path.

Again we use a weight vector w representing the weights of the edges. In fact this is the same weight vector as in Example G.1 but with a copy of each weight for the edge in opposite direction.

$$w = [\underbrace{1.5, 1.5}_{(s-a),(a-s)}, \underbrace{5, 5}_{(s-b),(b-s)}, \underbrace{2, 2}_{(a-b),(b-a)}, \underbrace{10, 10}_{(a-t),(t-a)}, \underbrace{3, 3}_{(b-t),(t-b)}]$$

With

$$B^{(0)} = [\underbrace{1}_{(s-a)}, 0, 0, 0, \underbrace{1}_{(a-b)}, 0, 0, 0, \underbrace{1}_{(b-t)}, 0]$$

we get the canonical penalty vector

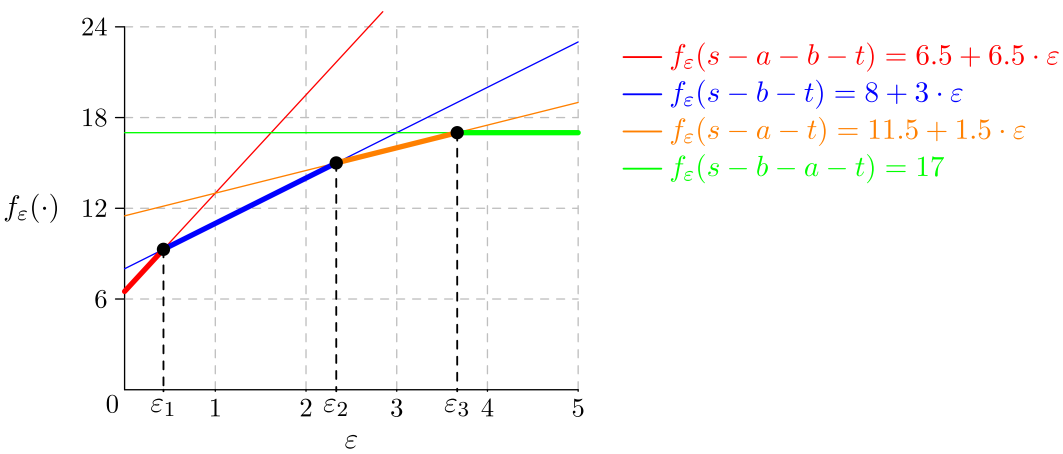
$$p = [\underbrace{1.5}_{=w(s-a)}, 0, 0, 0, \underbrace{2}_{=w(a-b)}, 0, 0, 0, \underbrace{3}_{=w(b-t)}, 0].$$

We see that the edge $b-a$, for example, is not punished, while the edge in opposite direction $a-b$ is punished. Hence, the punished value of the path $s-b-a-t$ is now 0 whereas it was 2 in Example G.1 where both directions of each edge were penalized.

Altogether we get the following chart where distinctions from Example G.1 are emphasized in bold print.

path P	$w(P)$	$p(P)$	$f_\varepsilon(P)$
$s-a-b-t$	6.5	6.5	$6.5+6.5\cdot\varepsilon$
$s-b-t$	8	3	$8+3\cdot\varepsilon$
$s-a-t$	11.5	1.5	$11.5+1.5\cdot\varepsilon$
$s-b-a-t$	17	**0**	$17+\mathbf{0}\cdot\varepsilon$

Again we plotted the linear functions $f_\varepsilon(\cdot)$ of the four paths in order to determine the penalty alternatives for each penalty parameter ε. Once more the bold lines indicate the penalized functional value $f_\varepsilon(\cdot)$ which indicates the corresponding penalty alterntive.



Without proof we summarize:

1. *For each ε in the interval $\left[0,\varepsilon_1=\frac{3}{7}\approx 0.43\right]$ the optimal solution $s-a-b-t$ is an ε-penalty alternative.*

2. *For each ε in the interval $\left[\varepsilon_1=\frac{3}{7}\approx 0.43,\varepsilon_2=\frac{7}{3}\approx 2.33\right]$ $s-b-t$ is an ε-penalty alternative.*

3. *For each ε in the interval $\left[\varepsilon_2=\frac{7}{3}\approx 2.33,\varepsilon_3=\frac{11}{3}\approx 3.67\right]$ $s-a-t$ is an ε-penalty alternative.*

4. *In contrast to Example G.1 where we examined the representation as undirected graph, now the path $s-b-a-t$ becomes an ε-penalty alternative. It is optimal for each ε in the interval $\left[\varepsilon_3=\frac{11}{3}\approx 3.67,\infty\right]$.*

5. *Once more it can be shown that the unconsidered paths containing cycles (e.g. $s-b-a-s-b-t$) are for no $0\leq\varepsilon\leq\infty$ penalty alternatives.*

Thus we saw that the penalty alternatives can depend on the graph representation. We can prevent this by adjusting the canonical penalty vector. In this example with the representation as directed graph we could modify the penalty vector p *a bit by penalizing also the edges in opposite direction. By doing so and penalizing with*

$$\overline{p} = [\underbrace{1.5}_{=w(s-a)}, \underbrace{1.5}_{=w(a-s)}, 0, 0, \underbrace{2}_{=w(a-b)}, \underbrace{2}_{=w(b-a)}, 0, 0, \underbrace{3}_{=w(b-t)}, \underbrace{3}_{=w(b-t)}].$$

we would get the same penalty alternatives as computed in Example G.1.

Appendix H

Examples for General Σ-Type Problems

Referes to pages 22, 33, 29, and 110.
We already mentioned examples for Σ-type problems on page 22: the shortest path problem, the minimum spanning tree problem, the assignment problem, the traveling salesperson problem, the binary knapsack problem and the sequence alignment problem. In this section we present some more examples for general Σ-type problems (please remember that Σ-type problems are general Σ-type problems as well).

Unbounded and b-Bounded Knapsack Problem

Consider a weight vector[1] $w \in \mathbb{R}^n_{\geq 0}$ and a vector of values $v \in \mathbb{R}^n_{\geq 0}$. Thus we have n items $(w_1, v_1), \ldots, (w_n, v_n)$ with weights w_i and values v_i. Furthermore let $C \geq 0$ be a given knapsack capacity.

Then a knapsack problem is a problem of the following type.

$$\begin{aligned} \max \sum_{i=1}^{n} v_i x_i & \\ \text{subject to } \sum_{i=1}^{n} w_i x_i \leq C & \\ x_i \in \{0, 1, \ldots, b\} & \qquad \text{for } i = 1, \ldots, n \end{aligned}$$

We differentiate the following three different types of knapsack problems:

(i) In the case $b = 1$ we call the problem above **0-1-knapsack problem or binary knapsack problem [KP]**.

(ii) In the case $b = \infty$, where we have no upper bound for x_i, we call the problem **unbounded knapsack problem [UKP]**.

(iii) In the case $1 < b < \infty$ we call the problem **b-bounded knapsack problem [BKP(b)]**.

[1]Note that the item weights w_i are not the weight function in our definition of general Σ-type problem.

Thus the unbounded knapsack problem is a general Σ-type problem: maximize $v(B)$ over all $B \in S$, where $S \subseteq \mathbb{N}^n$ contains all collections of items which fulfill the capacity constraint $w(B) \leq C$.
Analogously the b-bounded knapsack problem, where we have at most b copies of each item, is a general Σ-type problem which differs only in the definition of S which is now a subset of $\{0, 1, \ldots, b\}^n$.

Transportation Problem

Assume a set of n sources and m sinks, each source i has a certain supply $s_i \in \mathbb{R}$ of goods and each sink j has a certain demand $d_i \in \mathbb{R}$ with $s_1 + \cdots + s_n = d_1 + \cdots + d_m$. For each path (i, j) from a source i to a sink j transportation costs c_{ij} per unit arise. The aim is to transport the goods from the sources to the sinks, such that each sink gets as much goods as it needs and such that the total transportation costs are minimal.

$$
\begin{aligned}
\min_x \; & \sum_{i=1}^{n} \sum_{j=1}^{m} c_{ij} \cdot x_{ij} && && \text{(TP)} \\
\text{s.t.} \; & \sum_{j=1}^{m} x_{ij} = s_i && \text{for } i = 1, \ldots, n \\
& \sum_{i=1}^{n} x_{ij} = d_i && \text{for } j = 1, \ldots, m \\
& x_{ij} \geq 0 && \text{for } i = 1, \ldots, n \text{ and } j = 1, \ldots, m
\end{aligned}
$$

Here the weight function w assigns to each path (i, j) the transportation costs c_{ij}. The set of feasible solutions $S \subseteq \mathbb{N}^n$ contains all transportation plans which satisfy the demands without exceeding the supplies.

Network Flow Problem

Consider a finite directed graph $G = (V, E)$ with a non-negative capacity $c(e)$ for every edge $e \in E$, let s and t denote two distinguished vertices, the source and sink respectively.
A feasible flow is a real-valued function $f : E \to [0, \infty)$, which fulfills two properties.

1. **Capacity constraint:** $0 \leq f(e) \leq c(e)$ for all edges $e \in E$.

2. **Flow conservation:** The sum of the flows into the vertex should be the sum of the flows out of the vertex.We write

$$\sum_{e=(\overline{v},v)\in E} f(e) = \sum_{e=(v,\overline{v})\in E} f(e) \quad \text{for every vertex } v \in V \backslash \{s, t\}\,.$$

Then we consider the following maximization problem called the network flow problem

$$\max_{f \text{feasible flow}} w(f) := \sum_{e=(s,\overline{v})} f(e) - \sum_{e=(\overline{v},s)} f(e) = \sum_{e=(\overline{v},t)} f(e) - \sum_{e=(t,\overline{v})} f(e)\,.$$

Appendix I

Proofs to Section 2.3

I.1 Proof of Lemma 2.3.5

Refers to Lemma 2.3.5 on page 37.

Lemma (Properties of Penalty Alternatives, Schwarz 2003)
The following two statements hold.

(i) Every penalty alternative P is optimal for all penalty parameters ε in a non-empty optimality interval $I_P = [\varepsilon_L, \varepsilon_R]$, $\varepsilon_L, \varepsilon_R \in \mathbb{R} \cup \{\infty\}$ and for no other parameters. The case $\varepsilon_L = \varepsilon_R$ is allowed. We call P an interval representative of I_P.

(ii) If P and P' are two penalty alternatives and I_P and $I_{P'}$ their optimality intervals, then there are only three cases possible.

a) $I_P = I_{P'}$, iff $w(P) = w(P')$ and $p(P) = p(P')$.

b) $I_P \cap I_{P'} = \emptyset$.

c) $I_P \cap I_{P'} = \{\bar{\varepsilon}\}$, this means the intersection contains only a single epsilon. This happens if I_P and $I_{P'}$ are neighboring intervals.

Proof (cf. [Sch 2003, p. 16-17]).

(i) Assume $P \in S$ is optimal for the penalty parameters $\varepsilon_L, \varepsilon_R \in \mathbb{R} \cup \{\infty\}$ with $\varepsilon_L < \varepsilon_R$.

Case 1: $\boldsymbol{\varepsilon_R < \infty}$. In this case we have

$$w(P) + \varepsilon_L \cdot p(P) \le w(B) + \varepsilon_L \cdot p(B) \qquad \text{for all } B \in S\,, \tag{I.1}$$
$$w(P) + \varepsilon_R \cdot p(P) \le w(B) + \varepsilon_R \cdot p(B) \qquad \text{for all } B \in S\,. \tag{I.2}$$

For an intermediate value $\varepsilon \in (\varepsilon_L, \varepsilon_R)$ we multiply (I.1) by $\frac{\varepsilon_R - \varepsilon}{\varepsilon_R - \varepsilon_L} > 0$ and (I.2) by $\frac{\varepsilon - \varepsilon_L}{\varepsilon_R - \varepsilon_L} > 0$. Adding up these two inequalities leads to

$$\begin{aligned}
&\left(\frac{\varepsilon_R - \varepsilon}{\varepsilon_R - \varepsilon_L} + \frac{\varepsilon - \varepsilon_L}{\varepsilon_R - \varepsilon_L}\right) \cdot w(P) + \left(\frac{\varepsilon_R - \varepsilon}{\varepsilon_R - \varepsilon_L} + \frac{\varepsilon - \varepsilon_L}{\varepsilon_R - \varepsilon_L}\right) \cdot p(P) \\
\le &\left(\frac{\varepsilon_R - \varepsilon}{\varepsilon_R - \varepsilon_L} + \frac{\varepsilon - \varepsilon_L}{\varepsilon_R - \varepsilon_L}\right) \cdot w(B) + \left(\frac{\varepsilon_R - \varepsilon}{\varepsilon_R - \varepsilon_L} + \frac{\varepsilon - \varepsilon_L}{\varepsilon_R - \varepsilon_L}\right) \cdot p(B) \quad \text{for all } B \in S
\end{aligned}$$

which is equivalent to

$$w(P) + \varepsilon \cdot p(P) \le w(B) + \varepsilon \cdot p(B) \qquad \text{for all } B \in S\,. \tag{I.3}$$

Hence, P is also optimal for the intermediate value $\varepsilon \in (\varepsilon_L, \varepsilon_R)$ and thus for all values $\varepsilon \in [\varepsilon_L, \varepsilon_R]$.

Case 2: $\boldsymbol{\varepsilon_R = \infty}$. In this case we have

$$w(P) + \varepsilon_L \cdot p(P) \le w(B) + \varepsilon_L \cdot p(B) \qquad \text{for all } B \in S\,, \tag{I.4}$$
$$p(P) \le p(B) \qquad \text{for all } B \in S\,. \tag{I.5}$$

For a value $\varepsilon > \varepsilon_L$ we multiply (I.5) by $\varepsilon - \varepsilon_L > 0$ and add (I.4), getting

$$w(P) + \varepsilon \cdot p(P) \le w(B) + \varepsilon \cdot p(B) \qquad \text{for all } B \in S\,. \tag{I.6}$$

Hence, P is also optimal for all values $\varepsilon > \varepsilon_L$.

(ii) Assume that $P, P' \in S$ are both optimal for two different parameters ε_L and ε_R with $\varepsilon_L < \varepsilon_R$.

Case 1: $\boldsymbol{\varepsilon_R < \infty}$. In this case we have

$$w(P) + \varepsilon_L \cdot p(P) = w(P') + \varepsilon_L \cdot p(P') \quad \text{and} \tag{I.7}$$
$$w(P) + \varepsilon_R \cdot p(P) = w(P') + \varepsilon_R \cdot p(P')\,. \tag{I.8}$$

Subtracting (I.7) from (I.8) we get

$$\begin{aligned} & (\varepsilon_R - \varepsilon_L) \cdot p(P) = (\varepsilon_R - \varepsilon_L) \cdot p(P') && \mid : (\varepsilon_R - \varepsilon_L) \neq 0 \\ \Rightarrow\quad & p(P) = p(P') \\ \Rightarrow\quad & w(P) = w(P') \\ \Rightarrow\quad & f_\varepsilon(P) = f_\varepsilon(P') && \text{for all } \varepsilon \ge 0 \qquad \text{(I.9)} \end{aligned}$$

Case 2: $\boldsymbol{\varepsilon_R = \infty}$. In this case we have likewise (I.7) and directly $p(P) = p(P')$. Hence, (I.9) is also true for $\varepsilon_R = \infty$.

In both cases P and P' are optimal for the same parameters ε and must have the same optimality interval. ■

I.2 Proof of Theorem 2.3.10

Refers to Theorem 2.3.10 on page 40.

Theorem (Althöfer, Berger, Schwarz [ABS 2002])
Let $w : E \to \mathbb{R}$ be a real-valued function and $p : E \to \mathbb{R}_+$ a positive real valued function on E. Let $B^{(\varepsilon)}$ be defined according to Definition 2.2.2 for $\varepsilon \in \mathbb{R}_+$. The following four statements hold:

(i) $p\left(B^{(\varepsilon)}\right)$ is weakly monotonically decreasing in ε.

(ii) $w\left(B^{(\varepsilon)}\right)$ is weakly monotonically increasing in ε.

(iii) $w\left(B^{(\varepsilon)}\right) - p\left(B^{(\varepsilon)}\right)$ is weakly monotonically increasing in ε.

(iv) $\left(B^{(\varepsilon)}\right) + \varepsilon \cdot p\left(B^{(\varepsilon)}\right)$ is weakly monotonically increasing in ε.

Proof (cf. [Sch 2003, p. 10]). Let δ and ε be two arbitrary nonnegative penalty parameters with $0 \leq \delta < \varepsilon$.

Because of the definition of $B^{(\delta)}$ and $B^{(\varepsilon)}$ the following inequalities hold.

(i) In the case $\varepsilon < \infty$ we have

$$w\left(B^{(\varepsilon)}\right) + \varepsilon \cdot p\left(B^{(\varepsilon)}\right) \leq w\left(B^{(\delta)}\right) + \varepsilon \cdot p\left(B^{(\delta)}\right), \tag{I.10}$$
$$w\left(B^{(\varepsilon)}\right) + \delta \cdot p\left(B^{(\varepsilon)}\right) \geq w\left(B^{(\delta)}\right) + \delta \cdot p\left(B^{(\delta)}\right). \tag{I.11}$$

Subtracting (I.11) from (I.10) we get

$$(\varepsilon - \delta) \cdot p\left(B^{(\varepsilon)}\right) \leq (\varepsilon - \delta) \cdot p\left(B^{(\delta)}\right) \qquad | : (\varepsilon - \delta) > 0$$
$$\Leftrightarrow \qquad p\left(B^{(\varepsilon)}\right) \leq p\left(B^{(\delta)}\right). \tag{I.12}$$

In the case $\varepsilon = \infty$ inequality (I.12) follows directly from the definition of $B^{(\infty)}$.

(ii) Subtracting (I.12) multiplied with δ from (I.10) we get

$$w\left(B^{(\varepsilon)}\right) \geq w\left(B^{(\delta)}\right). \tag{I.13}$$

(iii) Subtracting (I.12) from (I.13) we get

$$w\left(B^{(\varepsilon)}\right) - p\left(B^{(\varepsilon)}\right) \geq w\left(B^{(\delta)}\right) - p\left(B^{(\delta)}\right).$$

(iv) With (I.11) and $\varepsilon > \delta \geq 0$ we have

$$w\left(B^{(\delta)}\right) + \delta \cdot p\left(B^{(\delta)}\right) \leq w\left(B^{(\varepsilon)}\right) + \delta \cdot p\left(B^{(\varepsilon)}\right) \leq w\left(B^{(\varepsilon)}\right) + \varepsilon \cdot p\left(B^{(\varepsilon)}\right)$$
$$\Rightarrow \quad w\left(B^{(\varepsilon)}\right) + \varepsilon p\left(B^{(\varepsilon)}\right) \geq w\left(B^{(\delta)}\right) + \delta p\left(B^{(\delta)}\right).$$

■

Appendix J

Experimentally Observed Cordel Frequencies under the Penalty Selection Rule

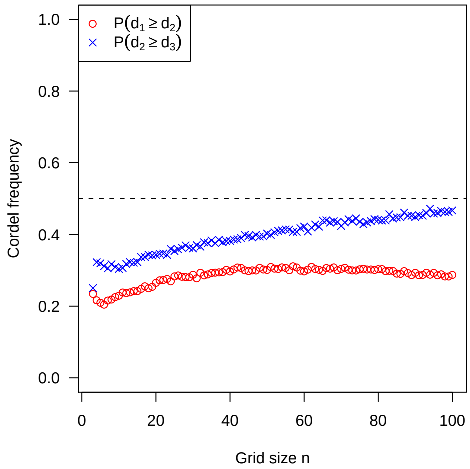
Overview of the **Cordel frequencies** presented in this chapter:

Shortest Path Problem	Grid Graphs (refers to p. 71)	quadratic $n \times n$ grids	Figure J.1.1, p. 212
		rectangular $n \times 2n$ grids	Figure J.1.2, p. 213
	Trellises (refers to p. 73)	quadratic $n \times n$ trellis	Figure J.2.1, p. 214
		rectangular $m{\times}2m$ trellis	Figure J.2.3, p. 216
		rectangular $2n \times n$ trellis	Figure J.2.5, p. 218
	Real Road Networks (refers to p. 75)		Figure J.3.1, p. 220
MST Problem	Grid Graphs (refers to p. 81)	quadratic $n \times n$ grids	Figure J.4.1, p. 224
		rectangular $n \times 2n$ grids	Figure J.4.3, p. 226
Knapsack Problems	b-bounded Knapsack Problem (refers to p. 82)	$b = 1$	Figure J.5.1, p. 228
		$b = 10$	Figure J.5.2, p. 229
		$b = 20$	Figure J.5.3, p. 230
		$b = \infty$	Figure J.5.4, p. 231

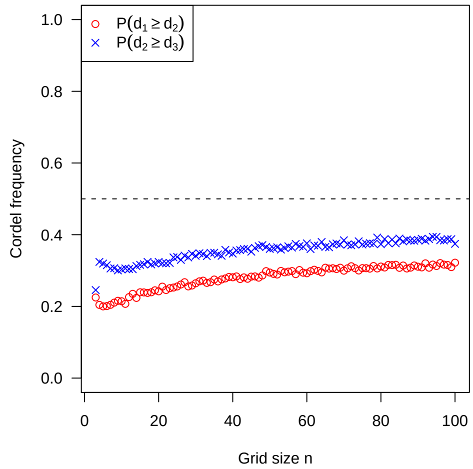
Overview of the **adjusted Cordel frequencies** presented in this chapter:

Shortest Path Problem	Trellises (refers to p. 86)	quadratic $n \times n$ trellis	Figure J.2.2, p. 215
		rectangular $m{\times}2m$ trellis	Figure J.2.4, p. 217
		rectangular $2n \times n$ trellis	Figure J.2.6, p. 219
MST Problem	Grid Graphs (refers to p. 88)	quadratic $n \times n$ grids	Figure J.4.2, p. 225
		rectangular $n \times 2n$ grids	Figure J.4.4, p. 227

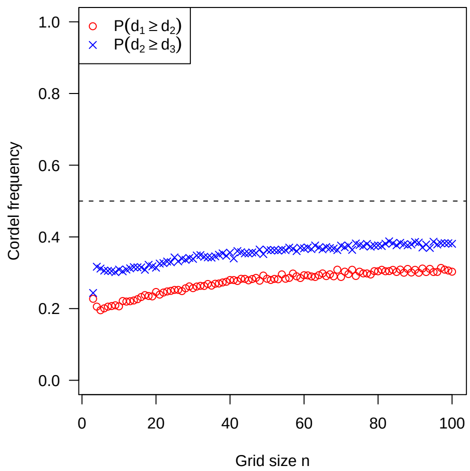
J.1 Shortest Path Problem in Grid Graphs



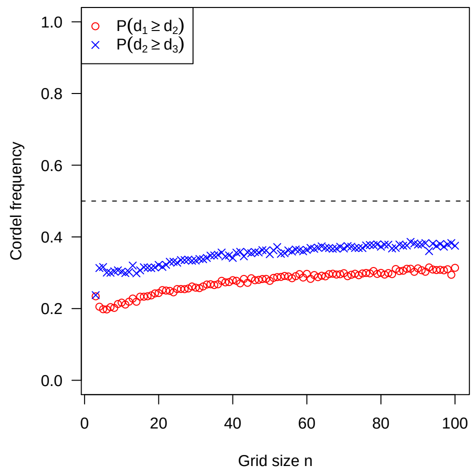
(a) $g = 2$



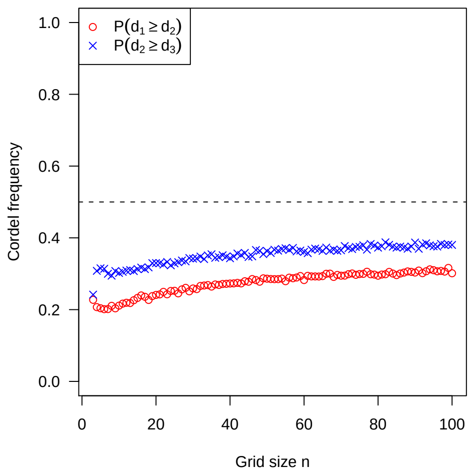
(b) $g = 3$



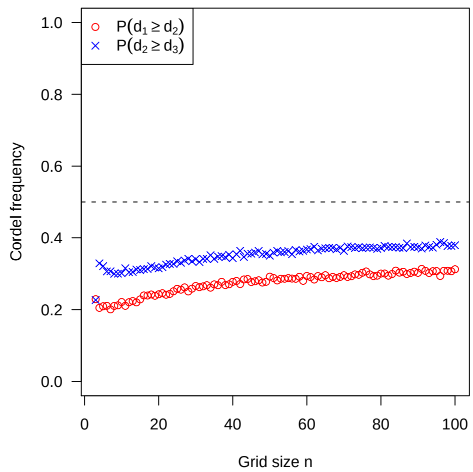
(c) $g = 4$



(d) $g = 5$

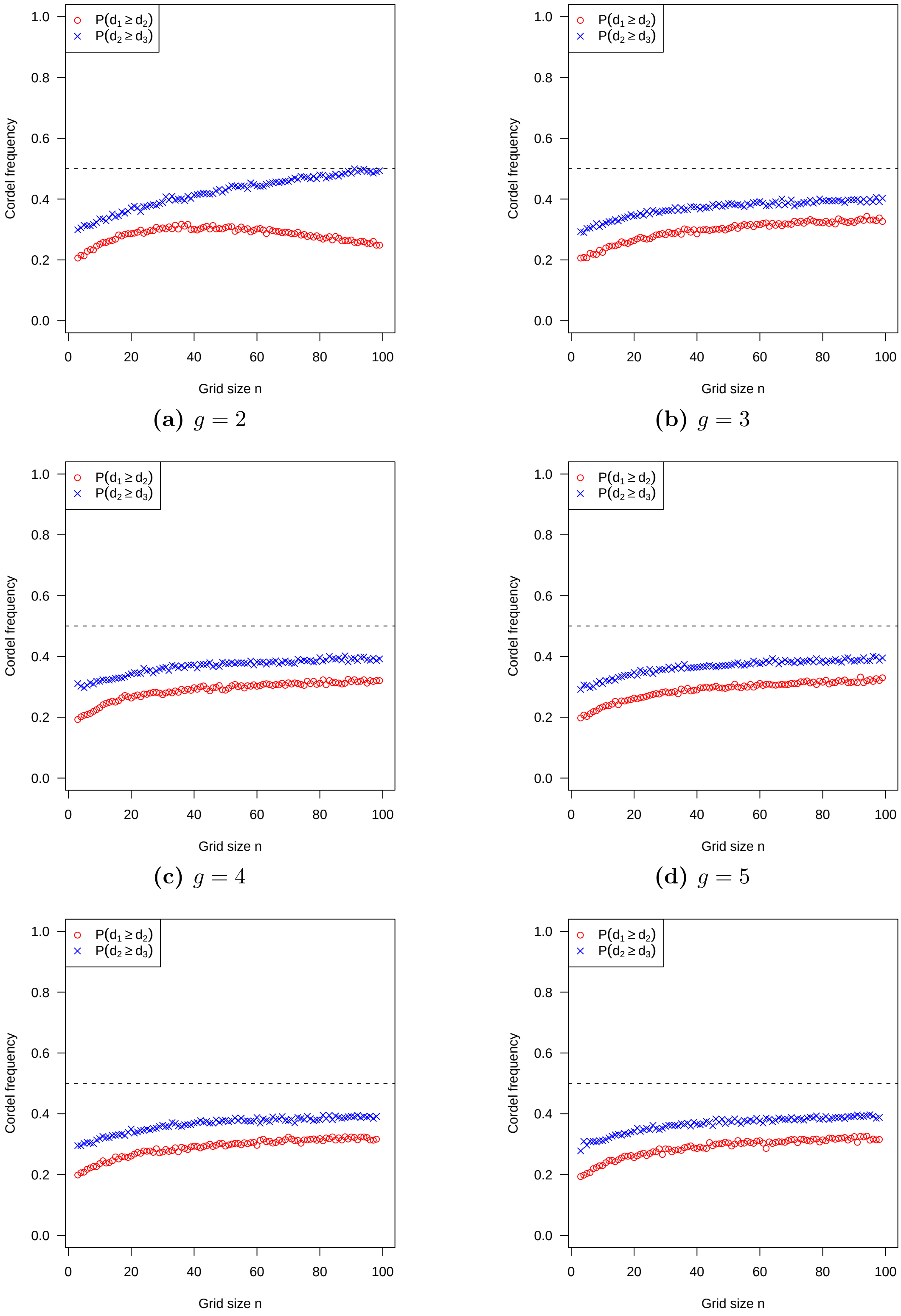


(e) $g = 6$



(f) $g = 7$

Figure J.1.1: Cordel frequencies for **quadratic** $\boldsymbol{n \times n}$ grids.



(a) $g = 2$

(b) $g = 3$

(c) $g = 4$

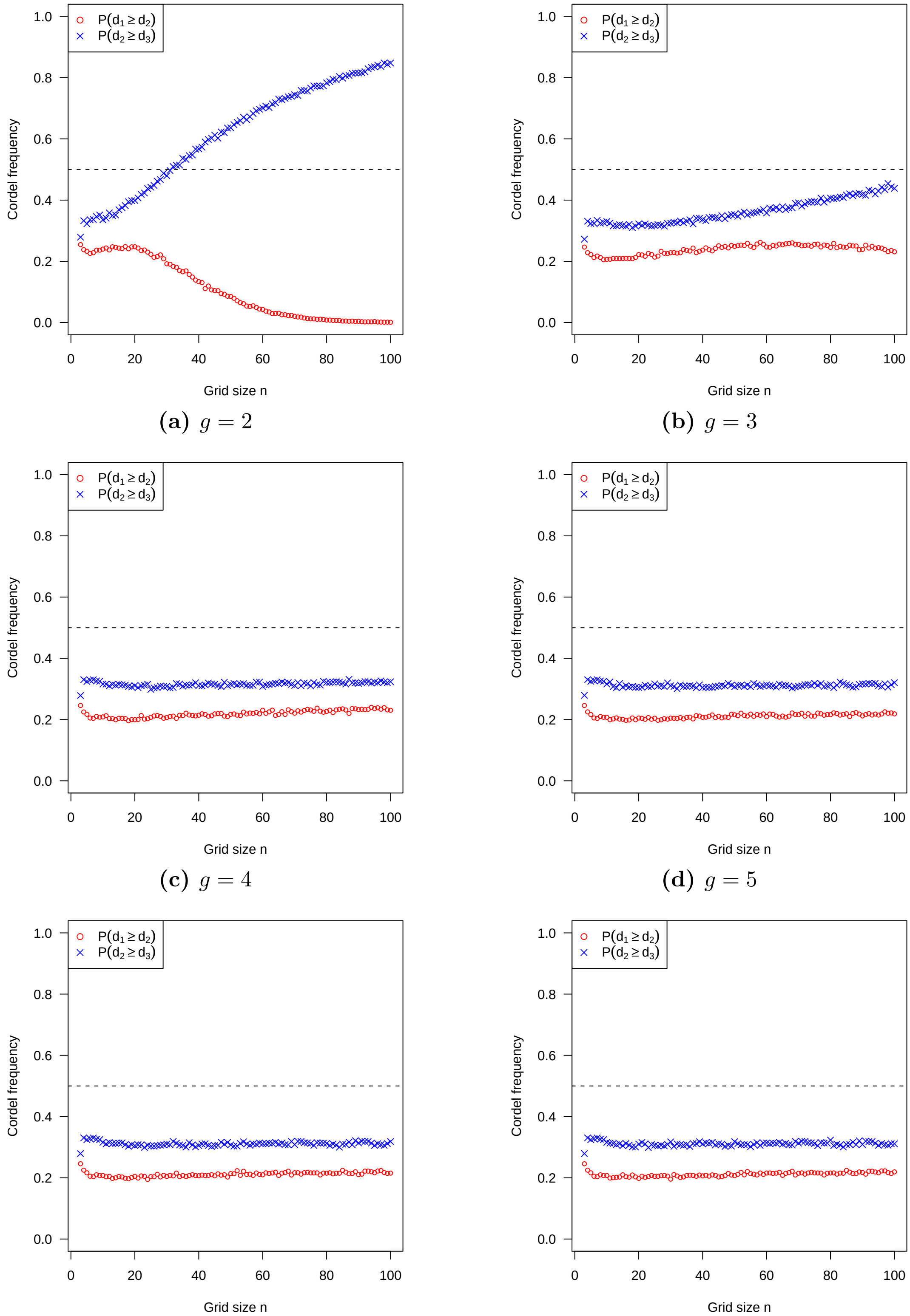
(d) $g = 5$

(e) $g = 6$

(f) $g = 7$

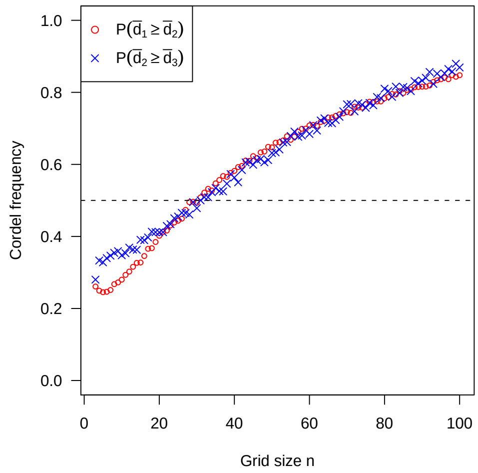
Figure J.1.2: Cordel frequencies for **rectangular** $\boldsymbol{n \times 2n}$ grids.

J.2 Shortest Path Problem in Trellis Graphs

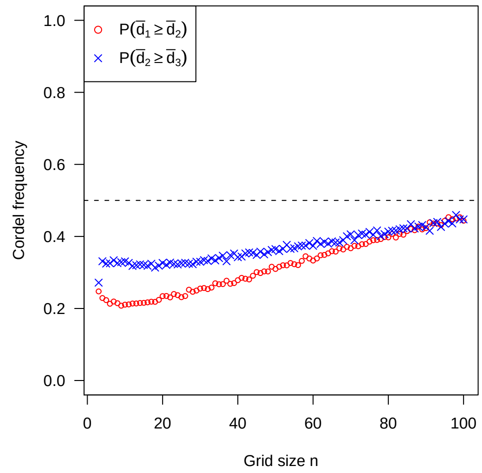


(a) $g = 2$ (b) $g = 3$ (c) $g = 4$ (d) $g = 5$ (e) $g = 6$ (f) $g = 7$

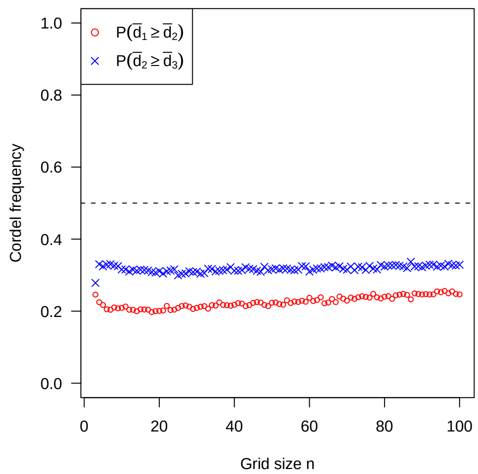
Figure J.2.1: Cordel frequencies for **quadratic** $\boldsymbol{n \times n}$ trellis graphs.



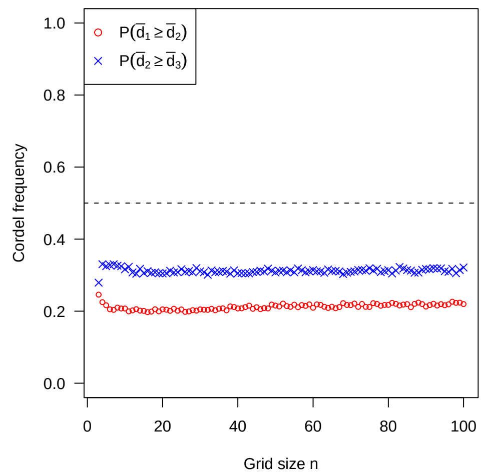
(a) $g = 2$



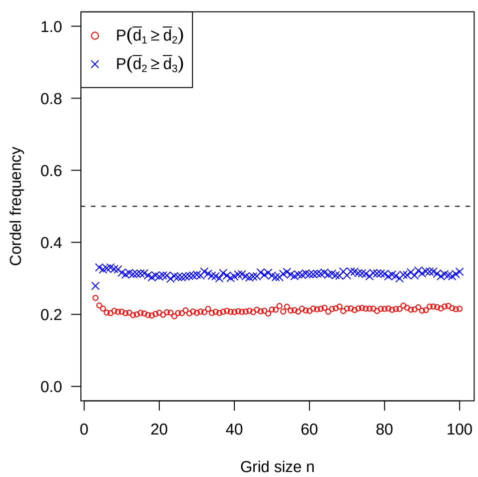
(b) $g = 3$



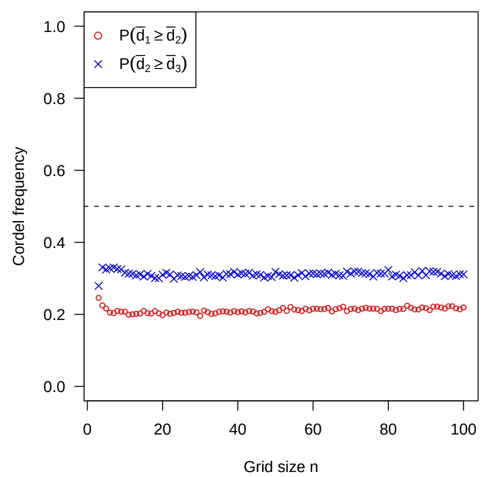
(c) $g = 4$



(d) $g = 5$

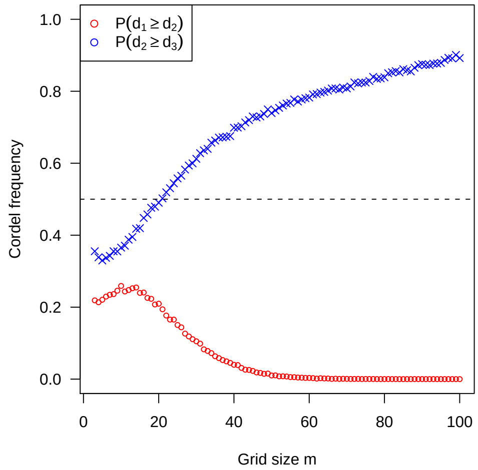


(e) $g = 6$

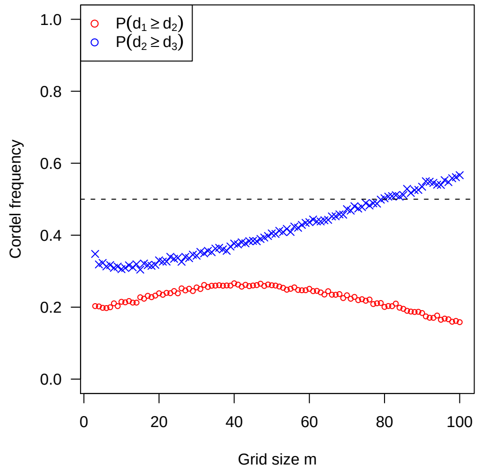


(f) $g = 7$

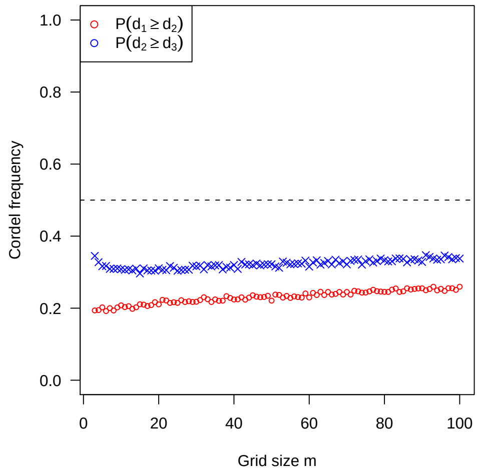
Figure J.2.2: Adjusted Cordel frequencies for **quadratic** $\boldsymbol{n \times n}$ trellis graphs.



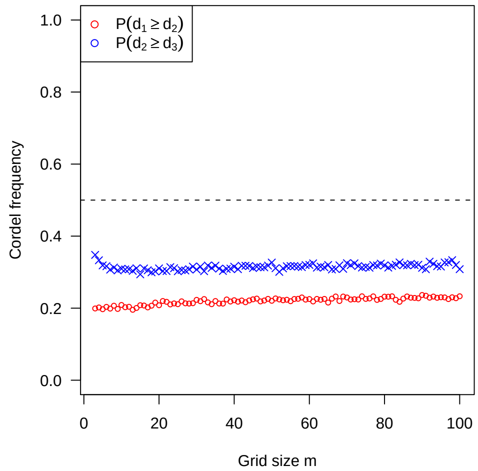
(a) $g = 2$



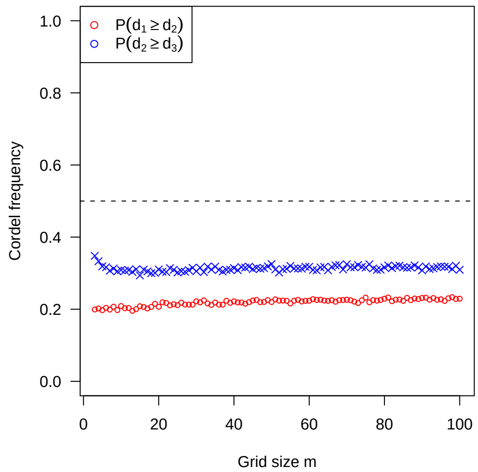
(b) $g = 3$



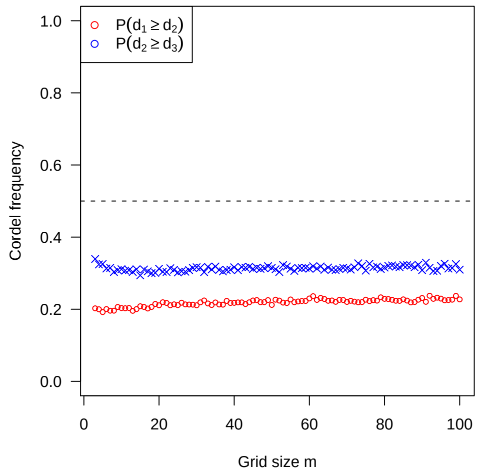
(c) $g = 4$



(d) $g = 5$

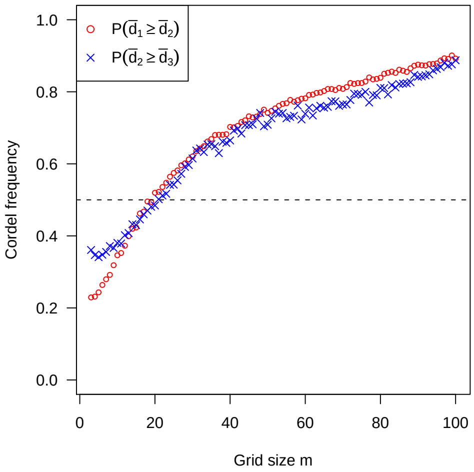


(e) $g = 6$

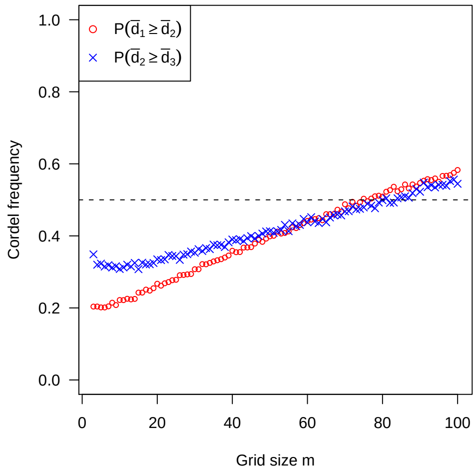


(f) $g = 7$

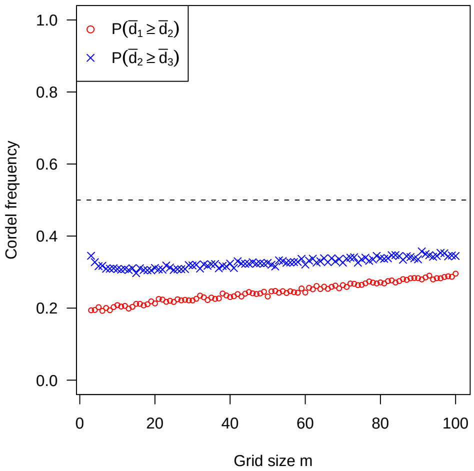
Figure J.2.3: Cordel frequencies for **rectangular $m \times 2m$** trellis graphs.



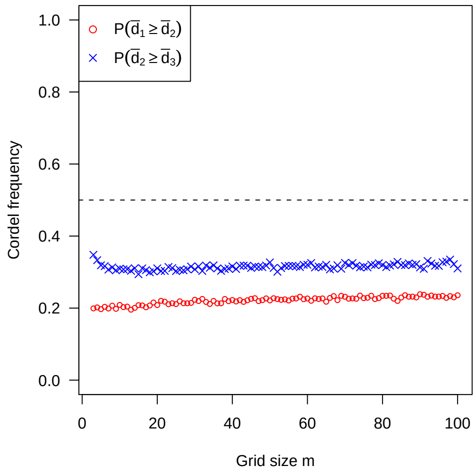
(a) $g = 2$



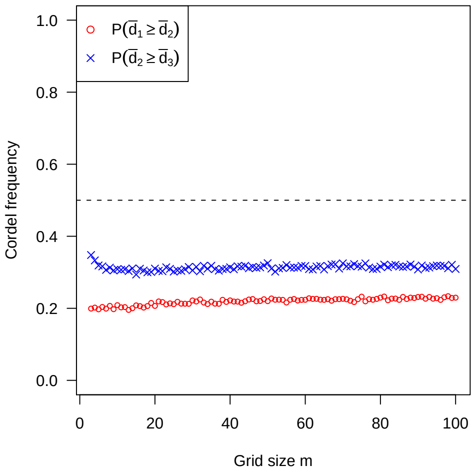
(b) $g = 3$



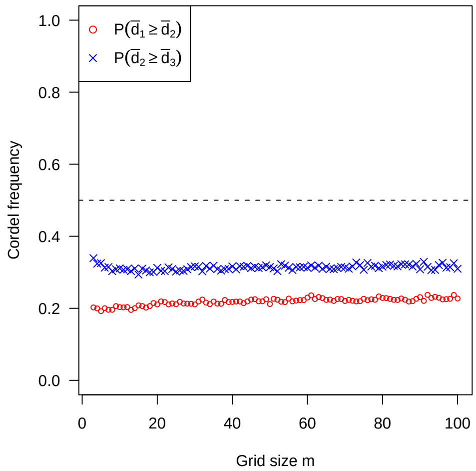
(c) $g = 4$



(d) $g = 5$

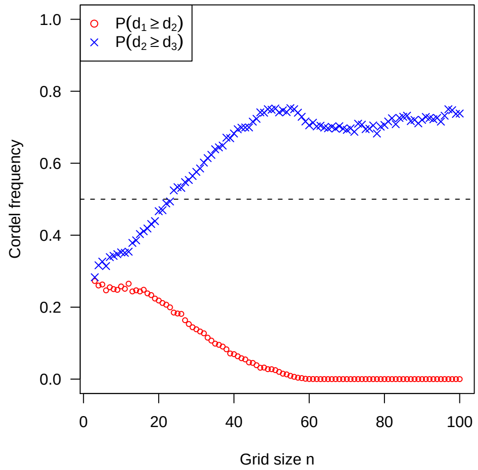


(e) $g = 6$

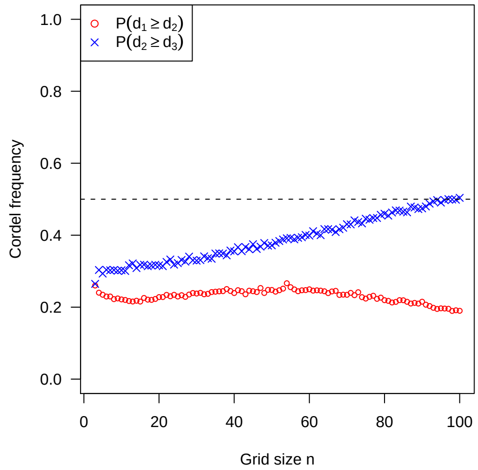


(f) $g = 7$

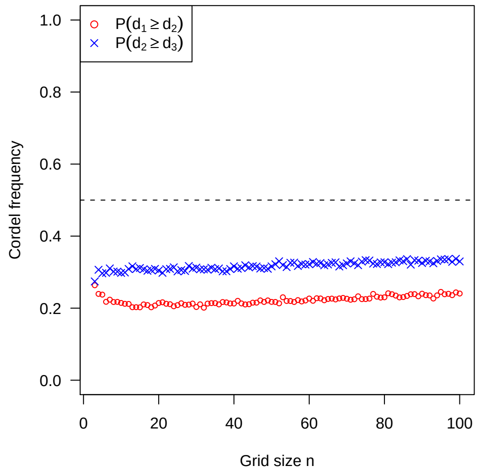
Figure J.2.4: Adjusted Cordel frequencies for **rectangular** $\boldsymbol{m \times 2m}$ trellis graphs.



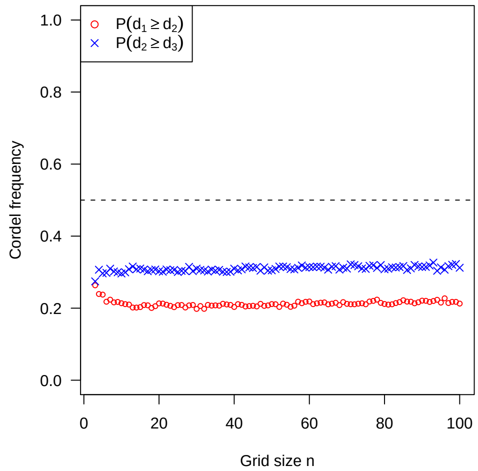
(a) $g = 2$



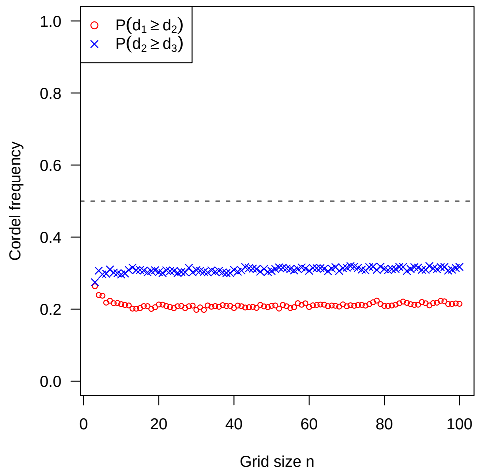
(b) $g = 3$



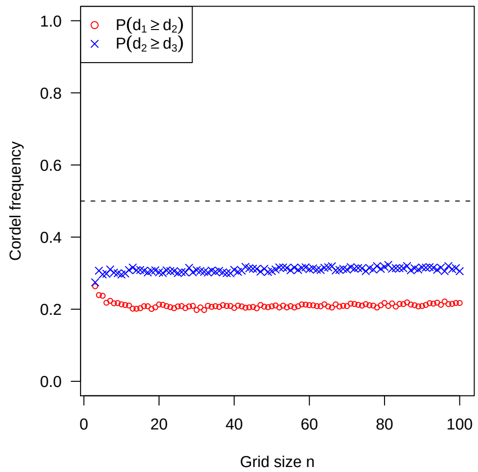
(c) $g = 4$



(d) $g = 5$

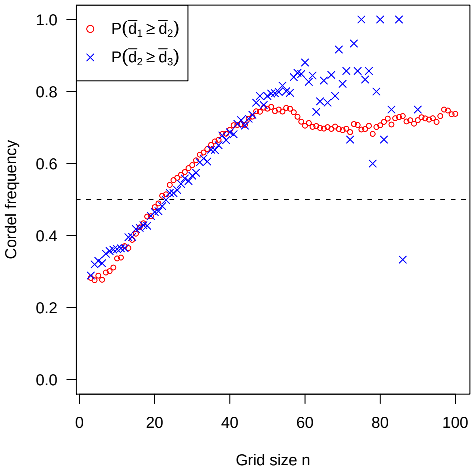


(e) $g = 6$

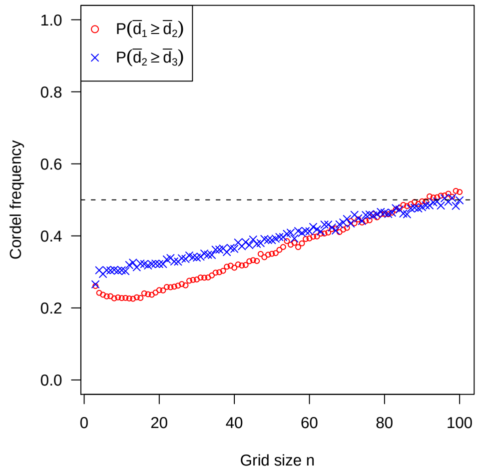


(f) $g = 7$

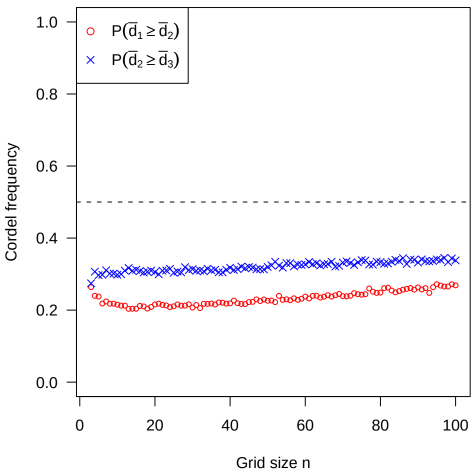
Figure J.2.5: Cordel frequencies for **rectangular $2n \times n$** trellis graphs.



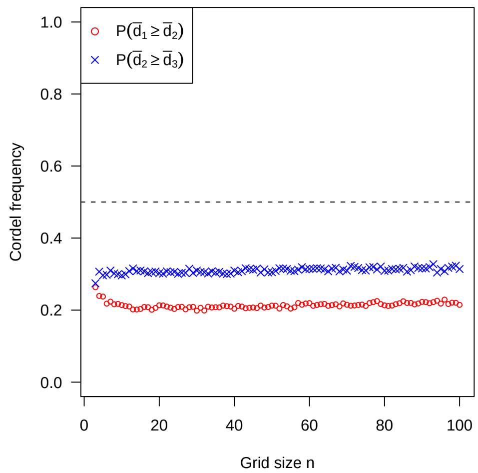
(a) $g = 2$



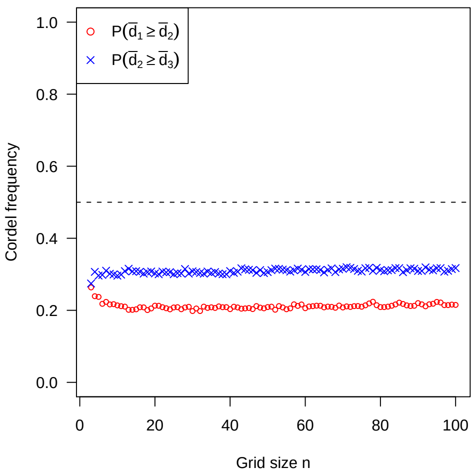
(b) $g = 3$



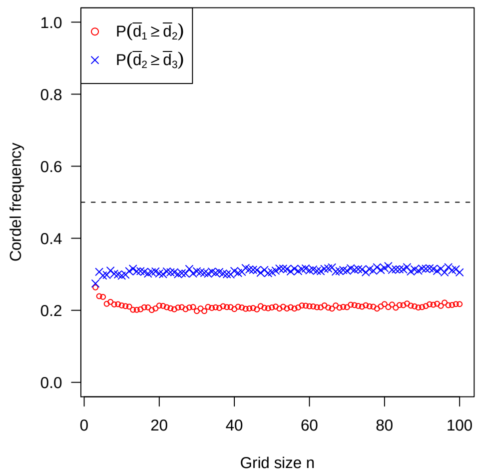
(c) $g = 4$



(d) $g = 5$



(e) $g = 6$



(f) $g = 7$

Figure J.2.6: Adjusted Cordel frequencies for **rectangular** $2n \times n$ trellis graphs.

J.3 Shortest Path Problem in Real Road Networks

Refers to page 76.

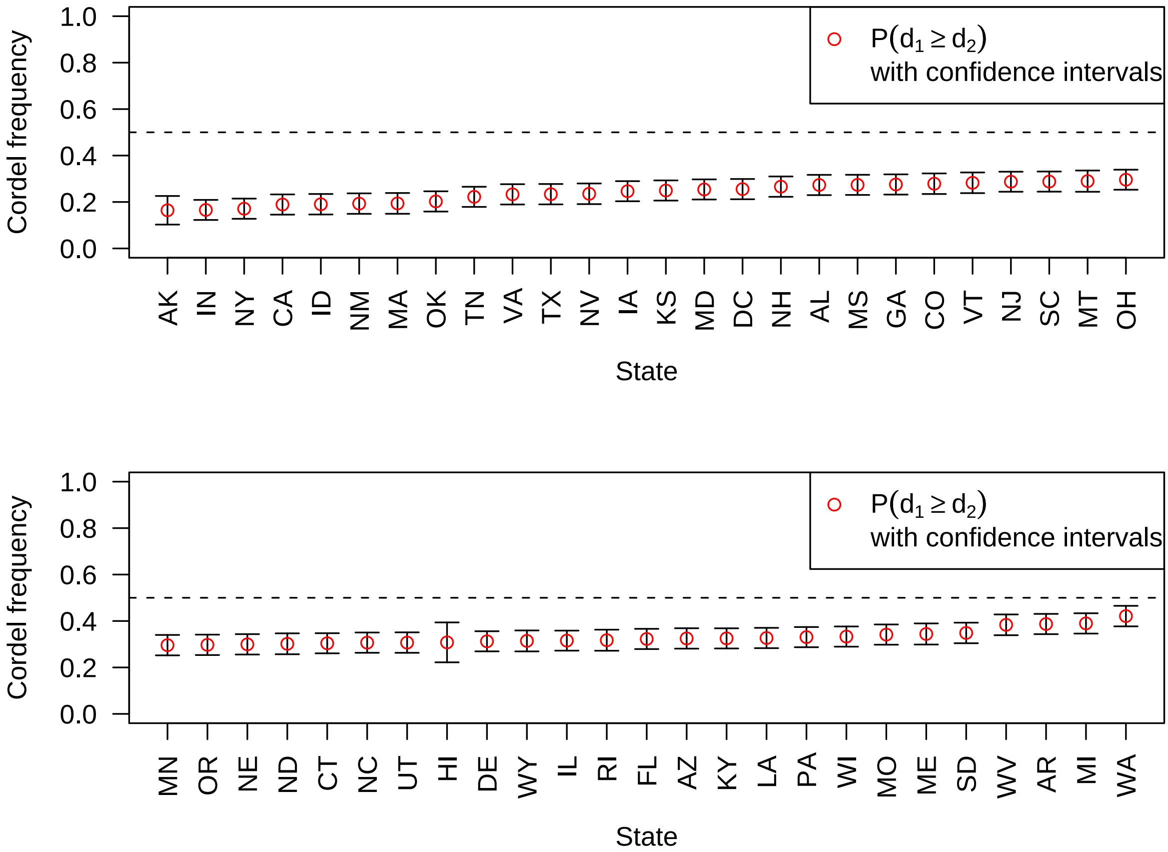


Figure J.3.1: Cordel Frequencies in TIGER/Line® roadmaps of the US-States.

List of Abbreviations and Size of the Considered Road Networks

Refers to pages 75, 76, and 76.

		with Shape Points		without Shape Points	
Abbr.	US-State	# vertices	# edges	# vertices	# edges
AK	Alaska	69082	156200	49929	117894
AL	Alabama	566843	1322974	401250	991788
AR	Arkansas	483175	1126072	321120	801962
AZ	Arizona	545111	1331654	385611	1012654
CA	California	1613325	3978298	1218395	3188438
CO	Colorado	448253	1078590	305932	793948
CT	Connecticut	153011	374636	120344	309302
DC	District of Columbia	9559	29818	8467	27634

continued on next page

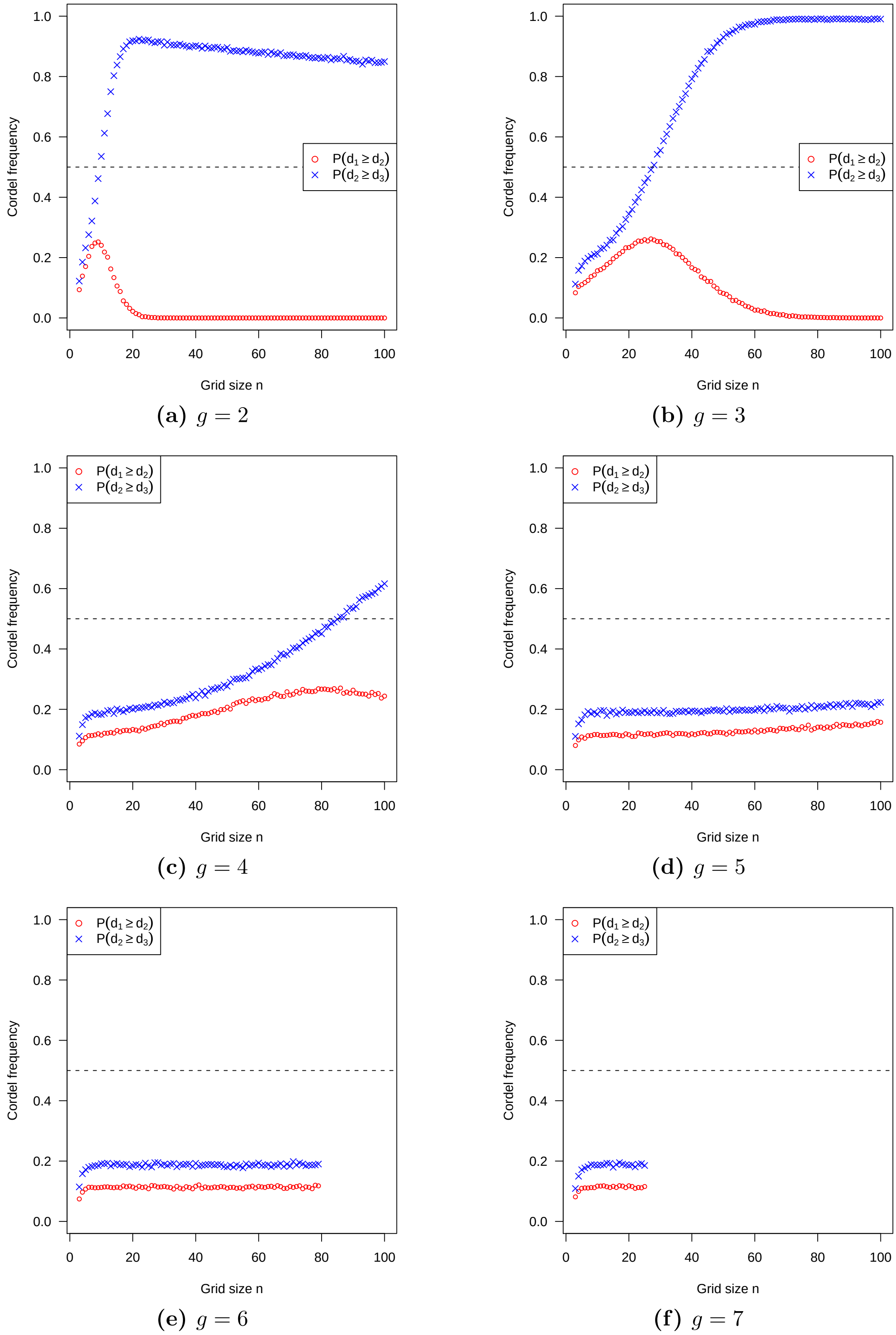
Abbr.	US-State	with Shape Points # vertices	# edges	without Shape Points # vertices	# edges
DE	Delaware	49109	121024	38461	99728
FL	Florida	1048506	2661102	830318	2224726
GA	Georgia	738879	1739780	541214	1344450
HI	Hawaii	64892	153618	49881	123596
IA	Iowa	390002	1004538	241505	707544
ID	Idaho	271450	637522	171223	437068
IL	Illinois	793336	2025634	551005	1540972
IN	Indiana	497458	1259500	363495	991574
KS	Kansas	474015	1214782	292385	851522
KY	Kentucky	467967	1051990	325685	767426
LA	Louisiana	413574	998508	308112	787584
MA	Massachusetts	308401	770328	253046	659618
MD	Maryland	265912	635248	213226	529876
ME	Maine	194505	429842	132512	305856
MI	Michigan	673534	1690174	487964	1319034
MN	Minnesota	547028	1340886	356717	960264
MO	Missouri	675407	1615784	443091	1151152
MS	Mississippi	413250	966612	288522	717156
MT	Montana	317905	721872	189699	465460
NC	North Carolina	887630	2019692	656421	1557274
ND	North Dakota	210801	521804	124350	348902
NE	Nebraska	308157	784016	179211	526124
NH	New Hampshire	116920	266830	78588	190166
NJ	New Jersey	330386	872072	272564	756428
NM	New Mexico	467529	1134168	330013	859136
NV	Nevada	261155	622086	163162	426100
NY	New York	716215	1794902	516230	1394932
OH	Ohio	676058	1685744	478919	1291466
OK	Oklahoma	540981	1328430	366020	978508
OR	Oregon	536236	1256334	382158	948178
PA	Pennsylvania	874843	2176592	626214	1679334
RI	Rhode Island	53658	138426	45827	122764
SC	South Carolina	463652	1107198	347746	875386
SD	South Dakota	212313	519244	120339	335296
TN	Tennessee	583484	1352160	415769	1016730
TX	Texas	2073870	5168318	1586204	4192986
UT	Utah	248730	591526	166468	427002
VA	Virginia	630639	1429618	479827	1127994
VT	Vermont	97975	215116	62679	144524
WA	Washington	575860	1350098	431176	1060730

continued on next page

		with Shape Points		without Shape Points	
Abbr.	US-State	# vertices	# edges	# vertices	# edges
WI	Wisconsin	519157	1270872	359481	951520
WV	West Virginia	300146	657716	198708	454840
WY	Wyoming	253077	608028	166242	434358

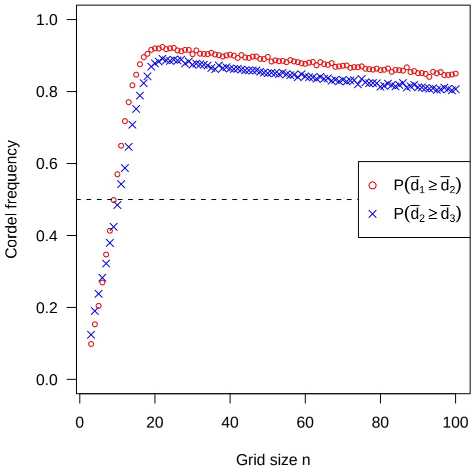
Table J.3.2: Number of vertices and number of edges of the road networks from [DIMACS] before and after the cutting-out of shape points.

J.4 Minimum Spanning Tree Problem

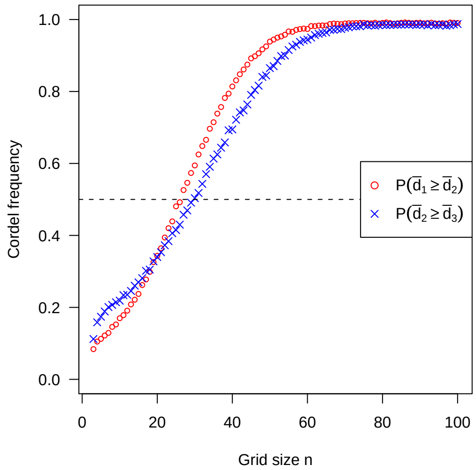


(a) $g = 2$ (b) $g = 3$ (c) $g = 4$ (d) $g = 5$ (e) $g = 6$ (f) $g = 7$

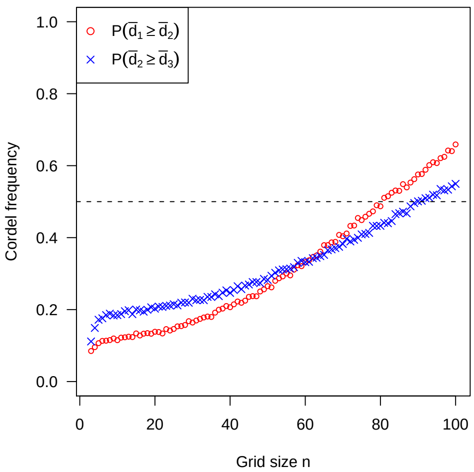
Figure J.4.1: Cordel frequencies for **quadratic** $\boldsymbol{n \times n}$ grids.



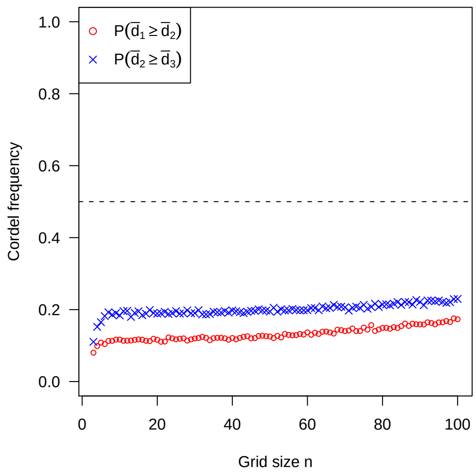
(a) $g = 2$



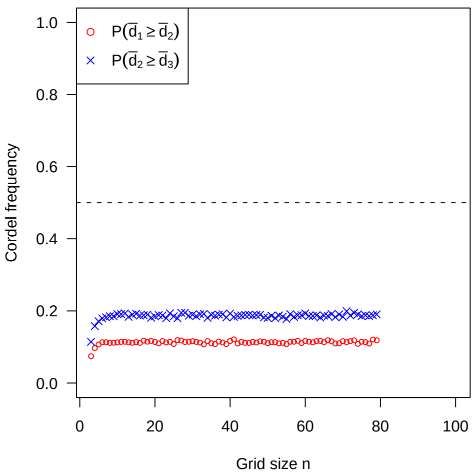
(b) $g = 3$



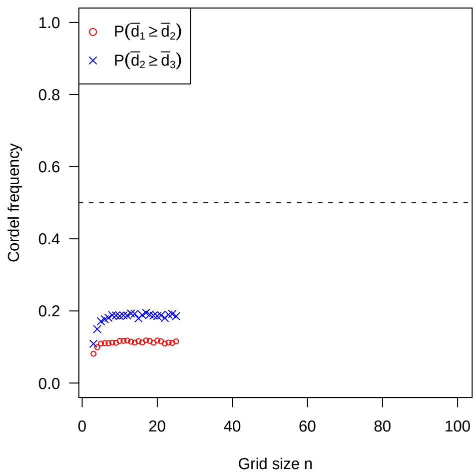
(c) $g = 4$



(d) $g = 5$

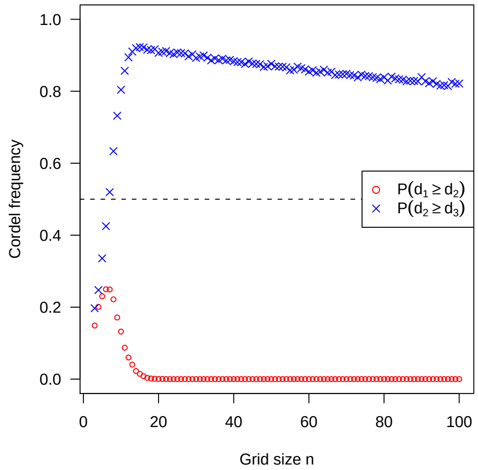


(e) $g = 6$

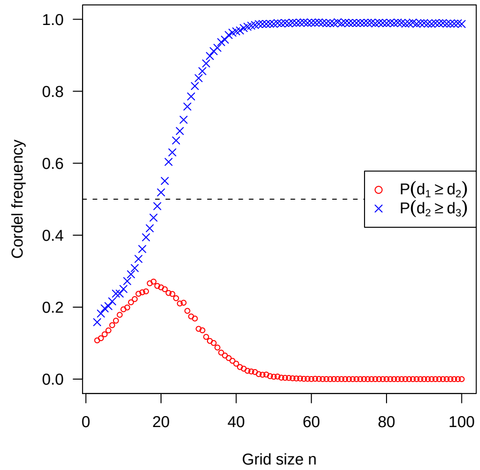


(f) $g = 7$

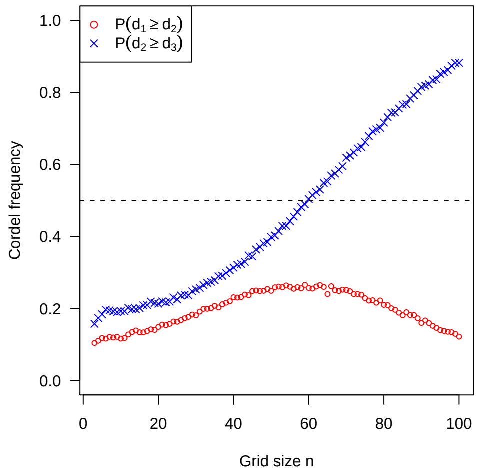
Figure J.4.2: Adjusted Cordel frequencies for **quadratic** $\boldsymbol{n \times n}$ grids.



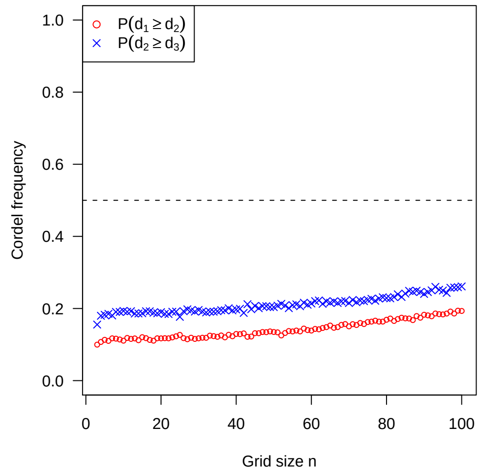
(a) $g = 2$



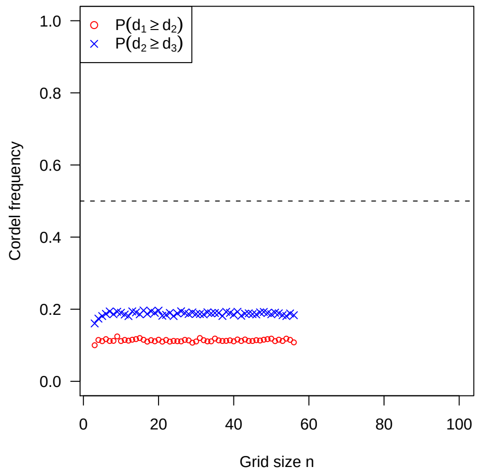
(b) $g = 3$



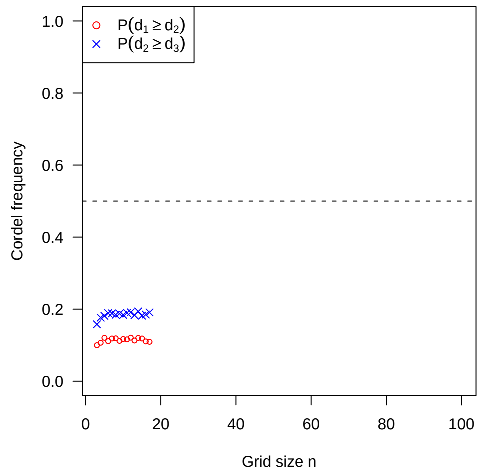
(c) $g = 4$



(d) $g = 5$

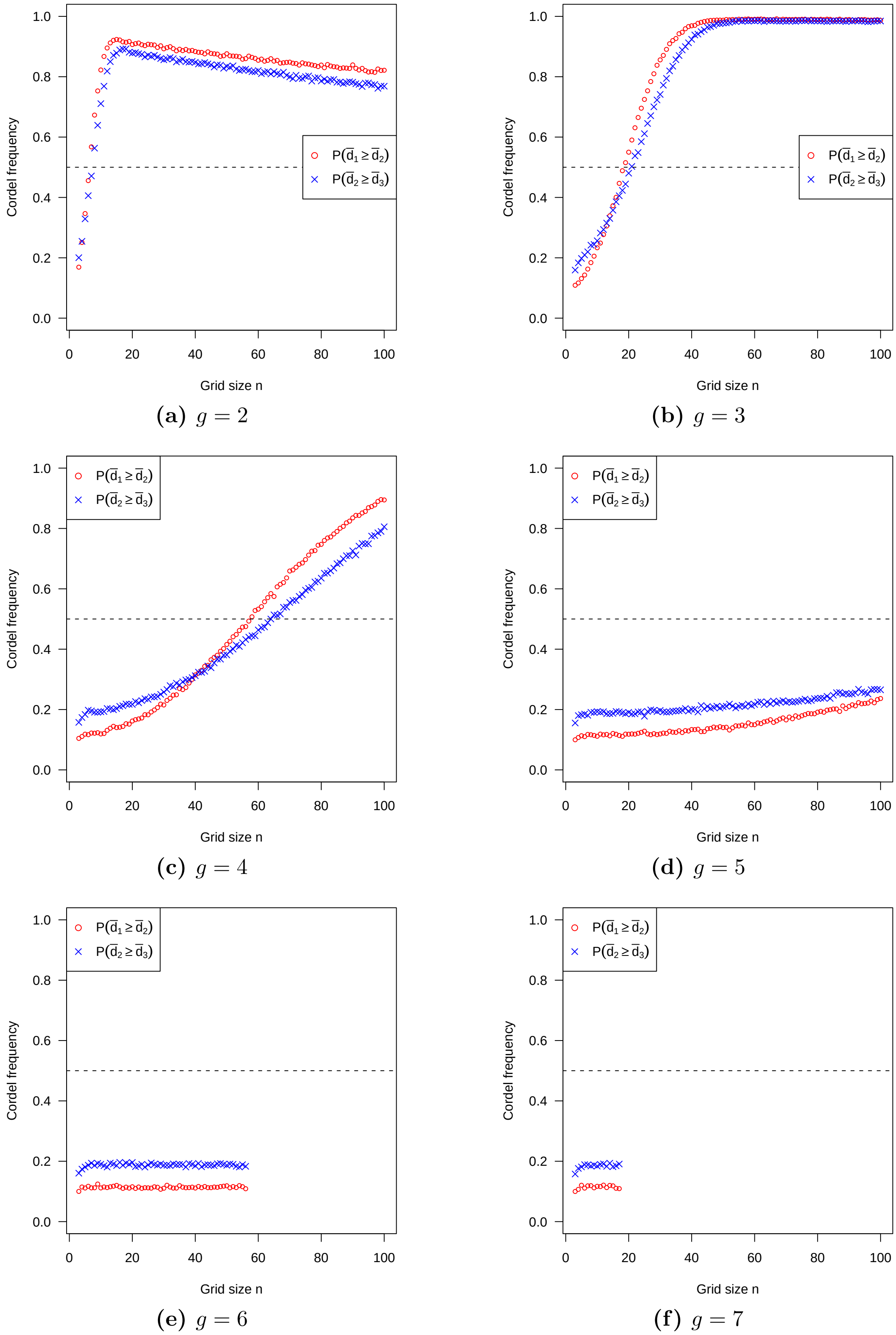


(e) $g = 6$



(f) $g = 7$

Figure J.4.3: Cordel frequencies for **rectangular** $\boldsymbol{n \times 2n}$ grids.



(a) $g = 2$ (b) $g = 3$ (c) $g = 4$ (d) $g = 5$ (e) $g = 6$ (f) $g = 7$

Figure J.4.4: Adjusted Cordel frequencies for **rectangular** $\boldsymbol{n \times 2n}$ grids.

J.5 Knapsack Problems

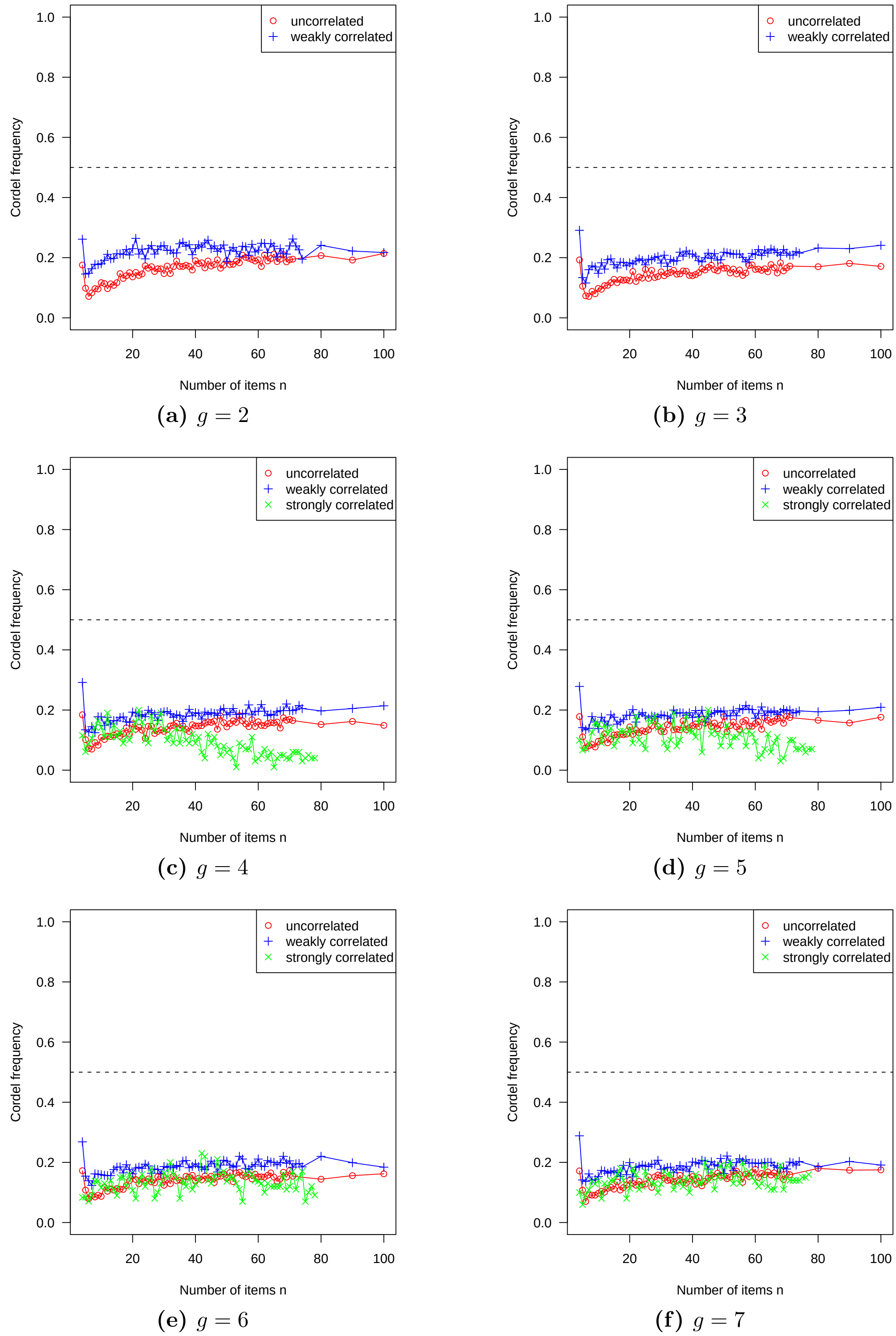
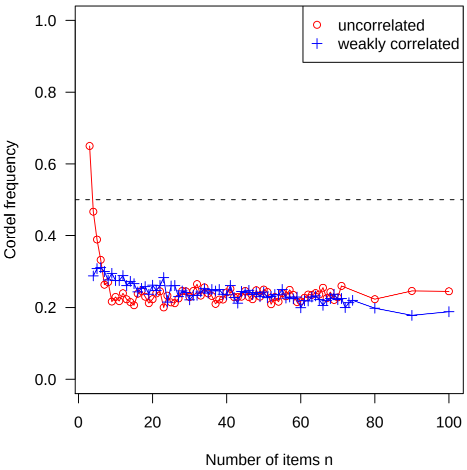
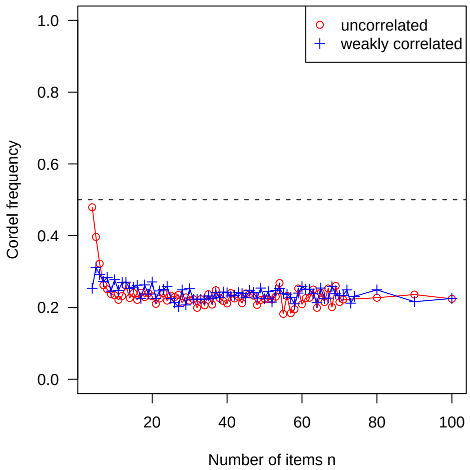


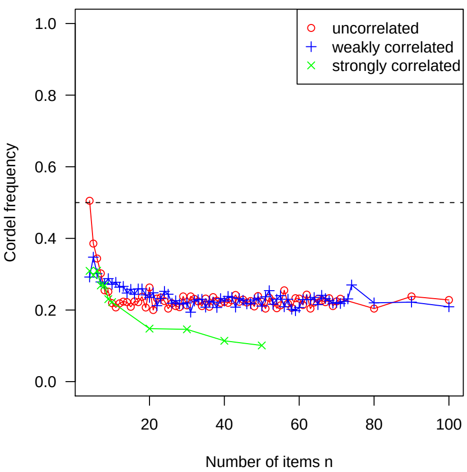
Figure J.5.1: Cordel frequencies for the **binary knapsack problem ($b = 1$)**.



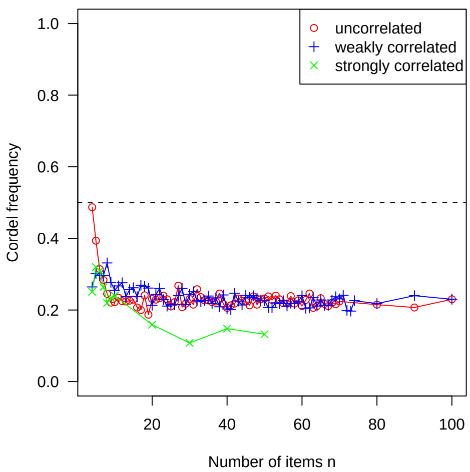
(a) $g = 2$



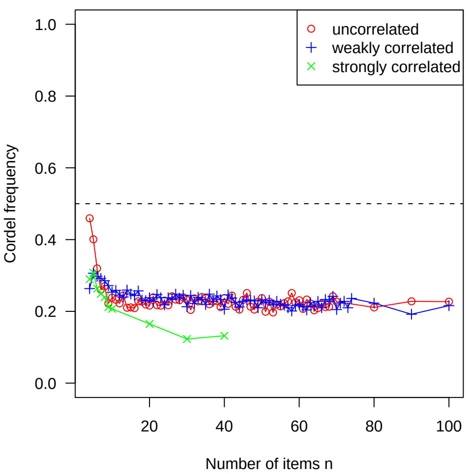
(b) $g = 3$



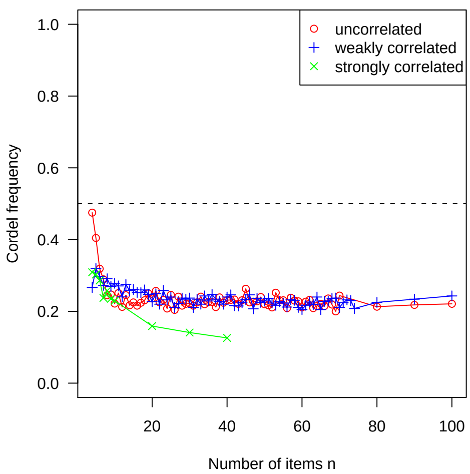
(c) $g = 4$



(d) $g = 5$

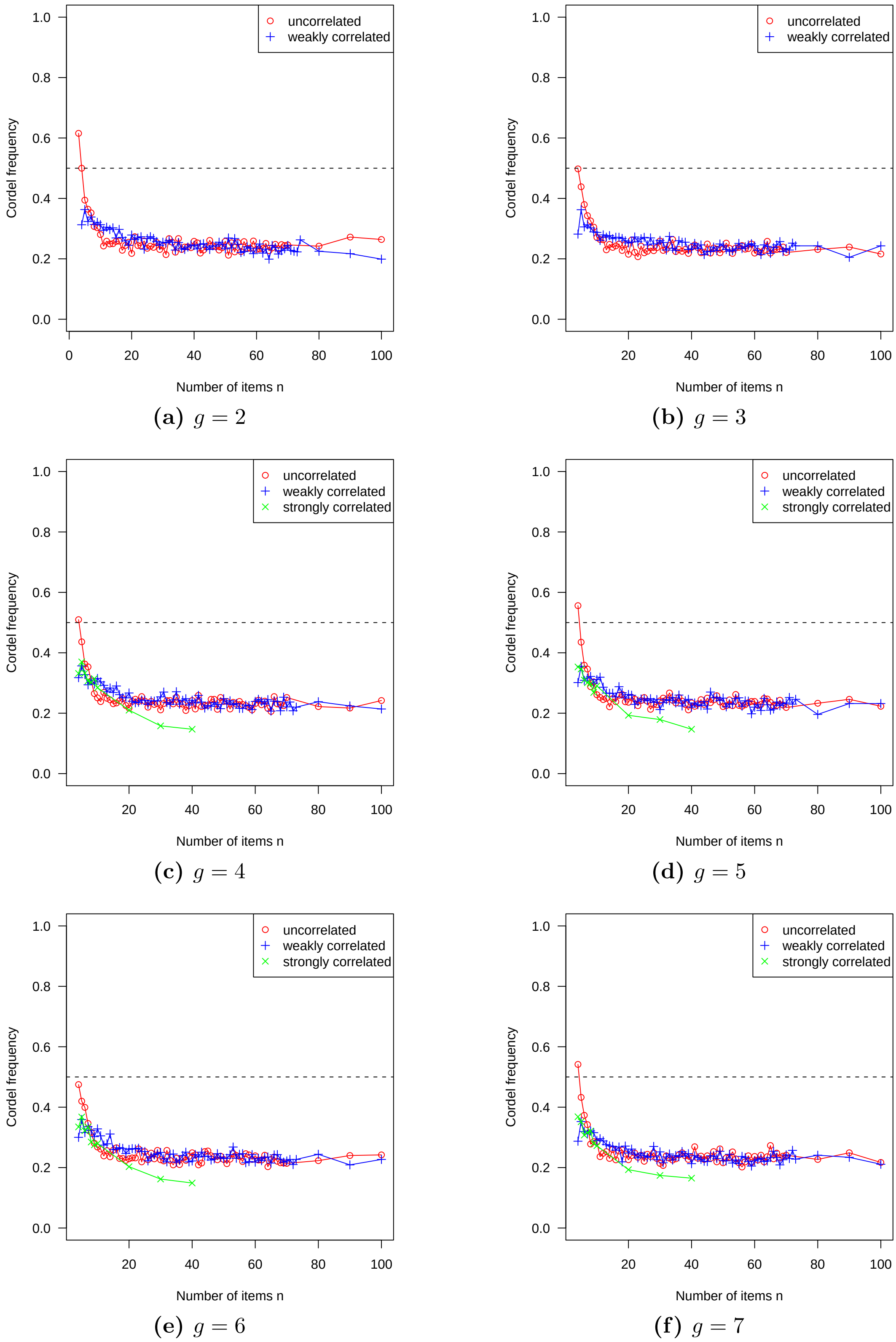


(e) $g = 6$



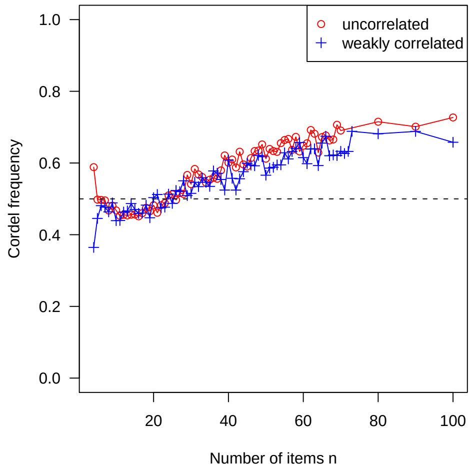
(f) $g = 7$

Figure J.5.2: Cordel frequencies for the **bounded knapsack problem with $b = 10$**.

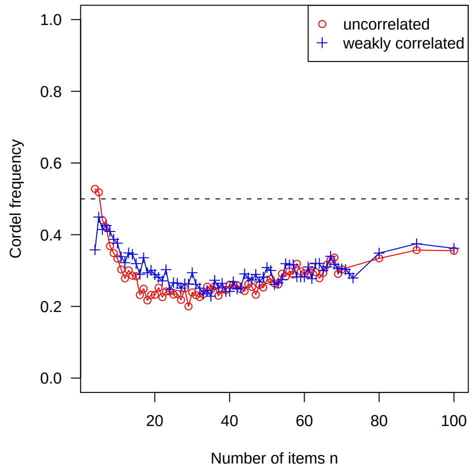


(a) $g = 2$ (b) $g = 3$ (c) $g = 4$ (d) $g = 5$ (e) $g = 6$ (f) $g = 7$

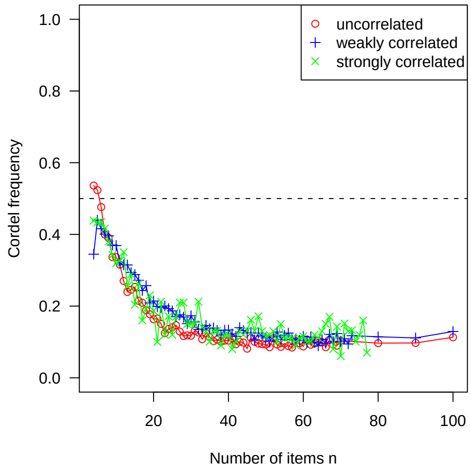
Figure J.5.3: Cordel frequencies for the **bounded knapsack problem with $b = 20$**.



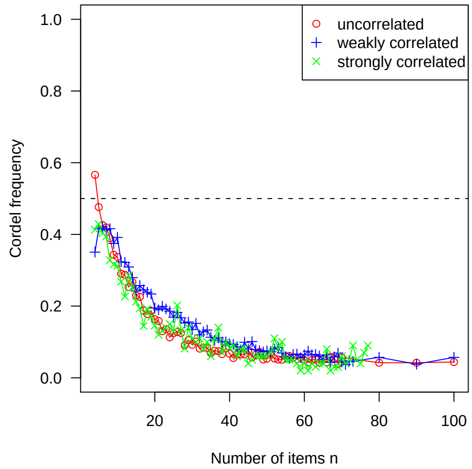
(a) $g = 2$



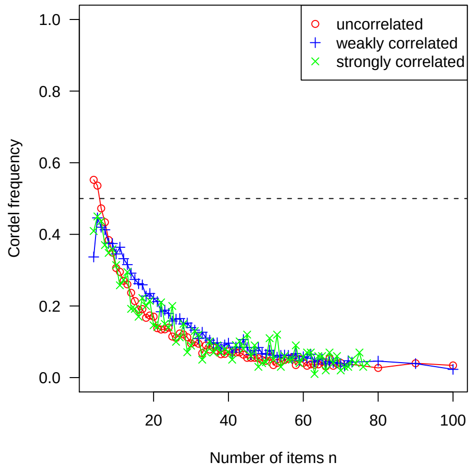
(b) $g = 3$



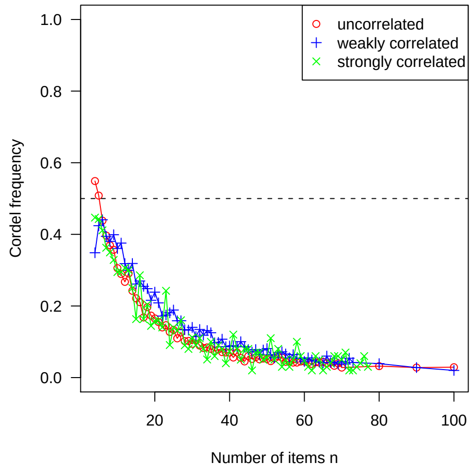
(c) $g = 4$



(d) $g = 5$



(e) $g = 6$



(f) $g = 7$

Figure J.5.4: Cordel frequencies for the **unbounded knapsack problem** ($\boldsymbol{b = \infty}$).

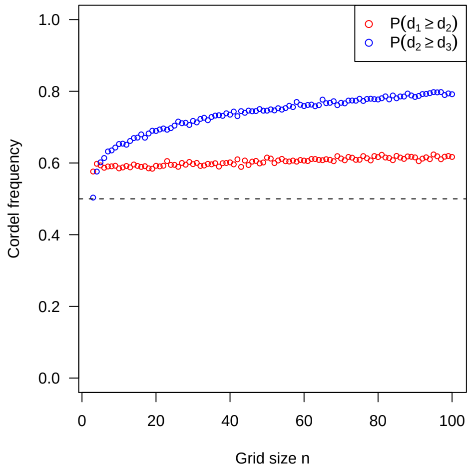
Appendix K

Experimentally Observed Cordel Frequencies under the Best Solutions Rule

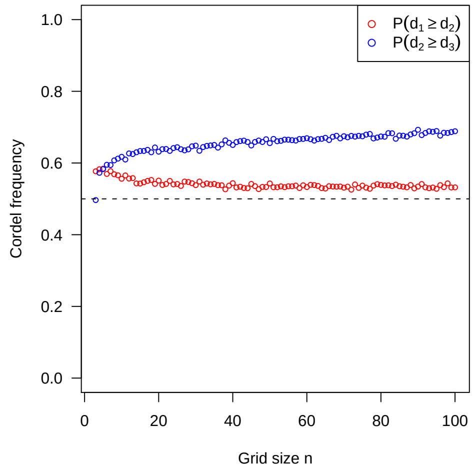
Overview of the **Cordel frequencies** presented in this chapter:

Shortest Path Problem	Grid Graphs (refers to p. 92)	quadratic $n \times n$ grids	Figure K.1.1, p. 234
		rectangular $n \times 2n$ grids	Figure K.1.2, p. 235
MST Problem	Grid Graphs (refers to p. 93)	quadratic $n \times n$ grids	Figure K.2.1, p. 236
		rectangular $n \times 2n$ grids	Figure K.2.2, p. 237

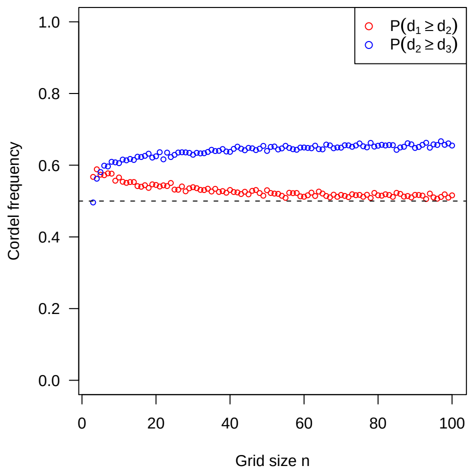
K.1 Shortest Path Problem in Grid Graphs



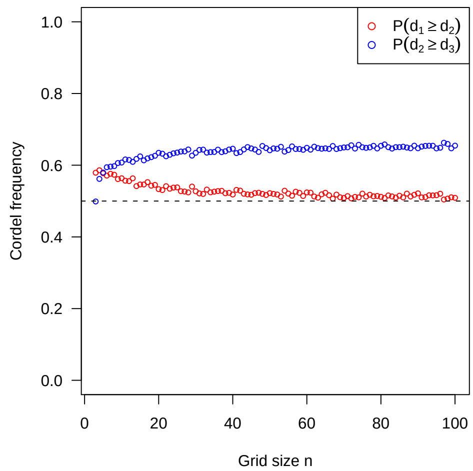
(a) $g = 2$



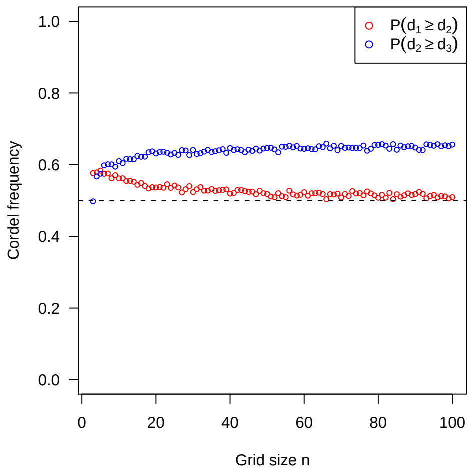
(b) $g = 3$



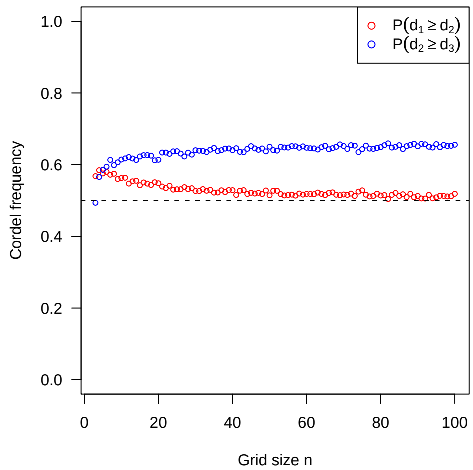
(c) $g = 4$



(d) $g = 5$

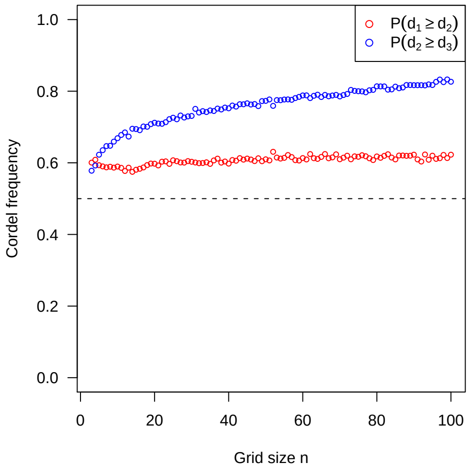


(e) $g = 6$

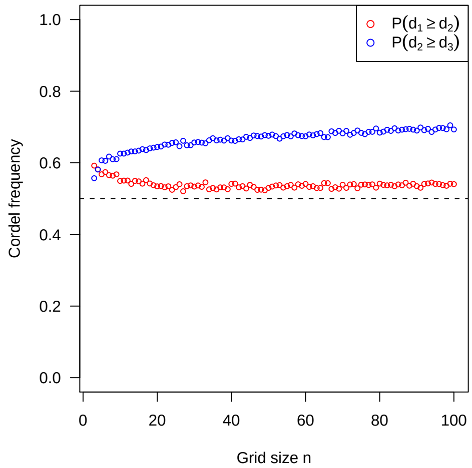


(f) $g = 7$

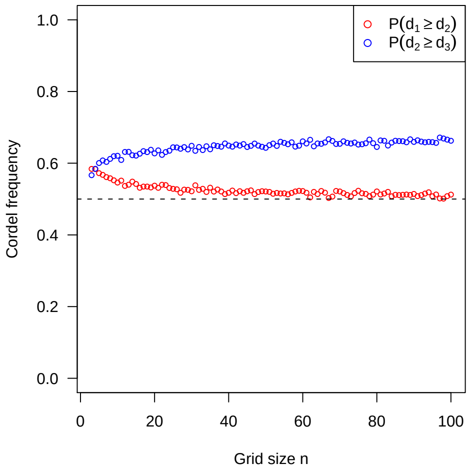
Figure K.1.1: Cordel frequencies for **quadratic** $\boldsymbol{n \times n}$ grids.



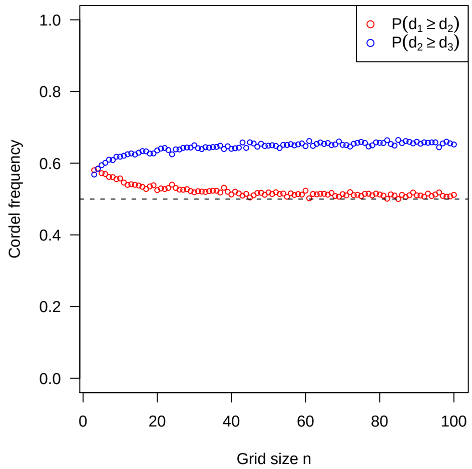
(a) $g = 2$



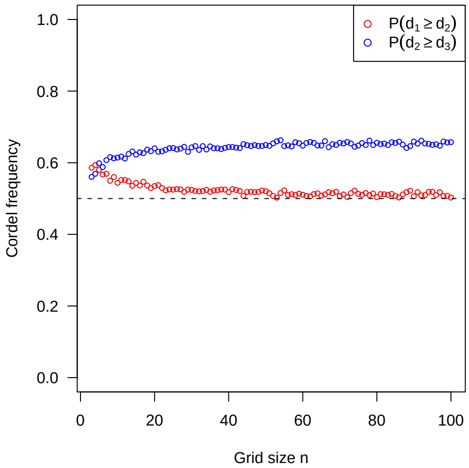
(b) $g = 3$



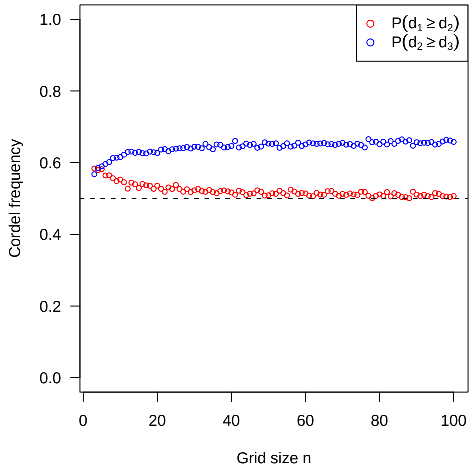
(c) $g = 4$



(d) $g = 5$



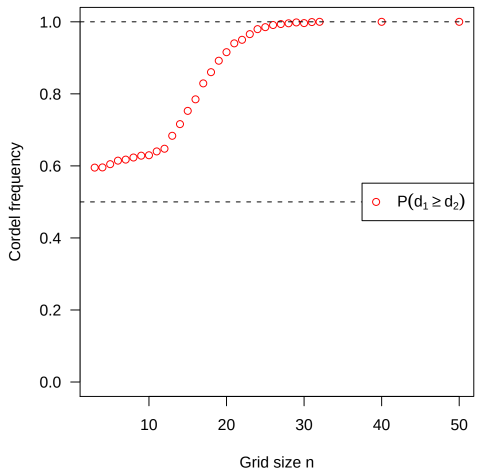
(e) $g = 6$



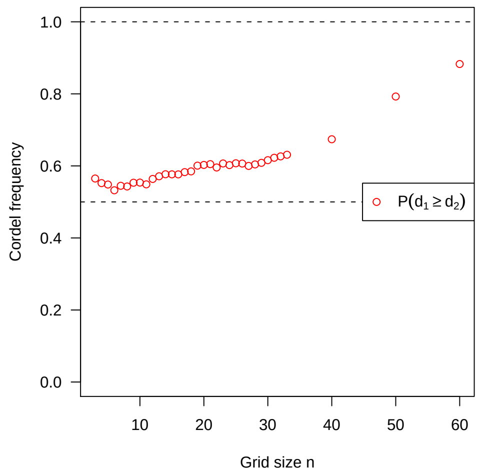
(f) $g = 7$

Figure K.1.2: Cordel frequencies for **rectangular** $\boldsymbol{n \times 2n}$ grids.

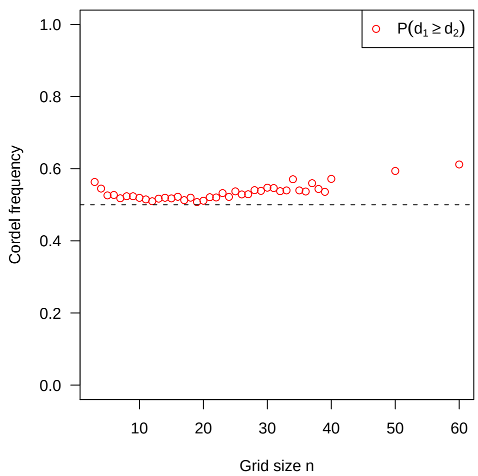
K.2 Minimum Spanning Tree Problem



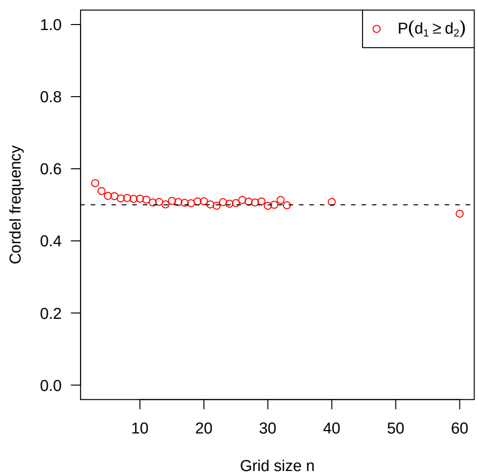
(a) $g = 2$



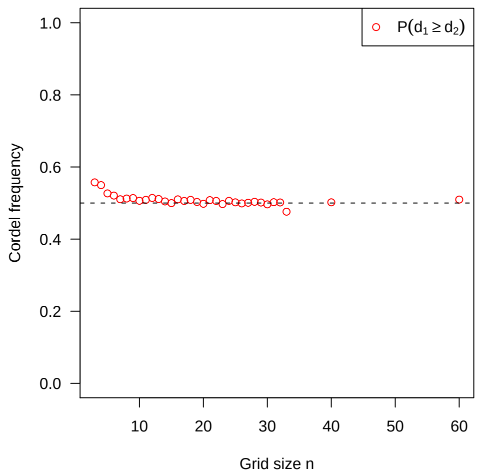
(b) $g = 3$



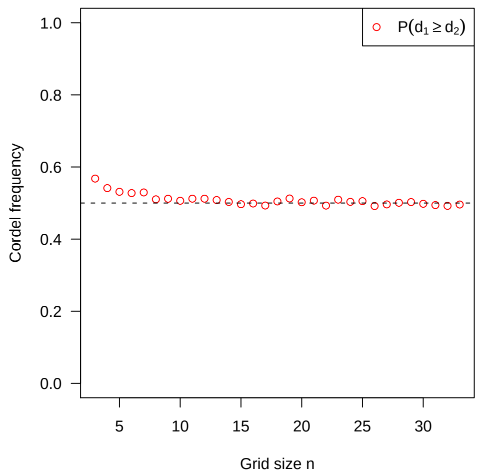
(c) $g = 4$



(d) $g = 5$

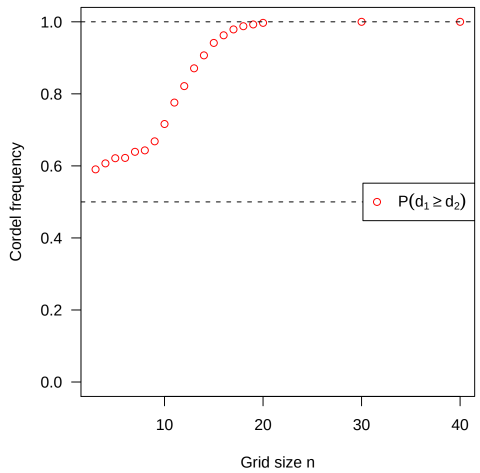


(e) $g = 6$

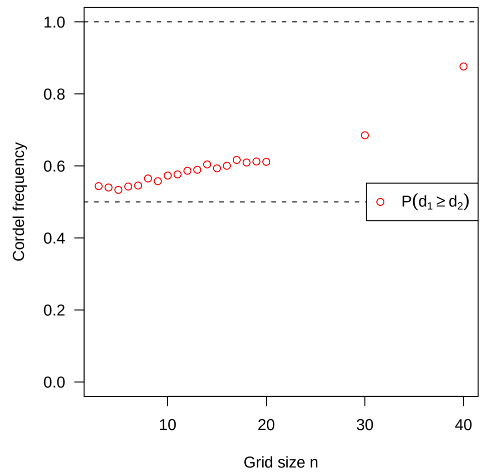


(f) $g = 7$

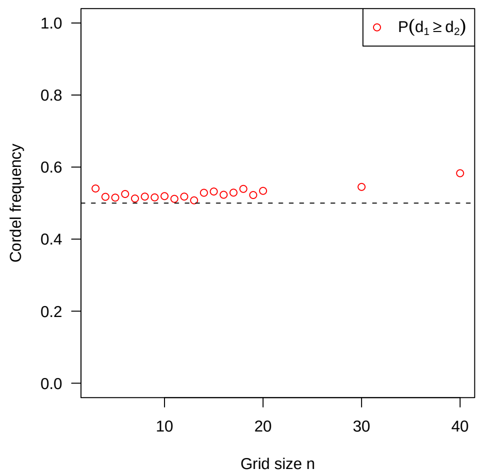
Figure K.2.1: Cordel frequencies for **quadratic $n \times n$** grids.



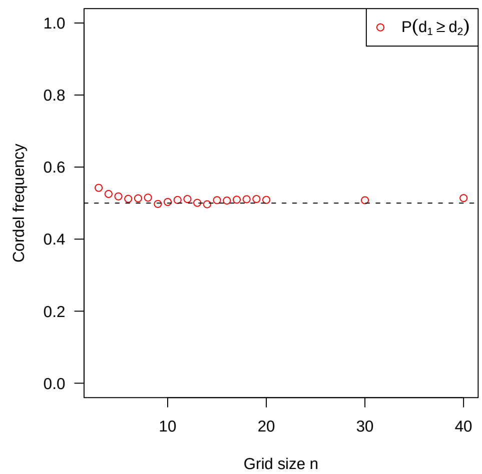
(a) $g = 2$



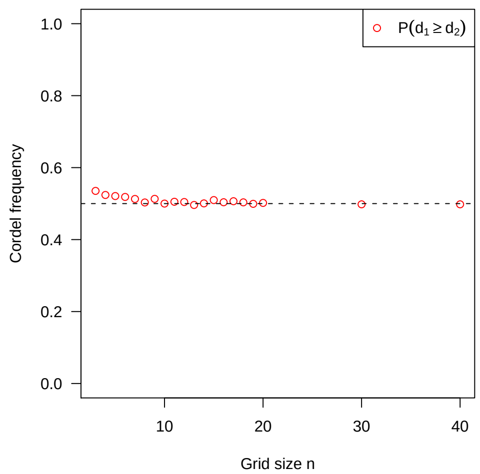
(b) $g = 3$



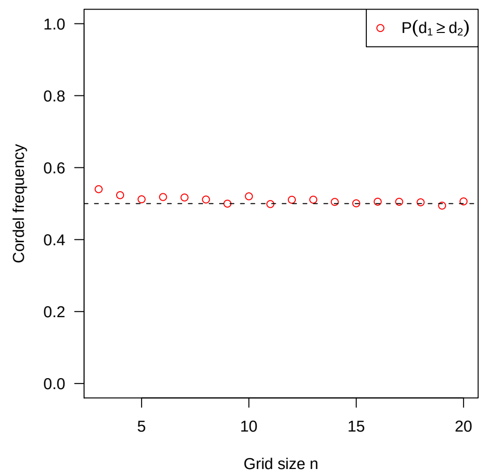
(c) $g = 4$



(d) $g = 5$



(e) $g = 6$



(f) $g = 7$

Figure K.2.2: Cordel frequencies for **rectangular $n \times 2n$** grids.

Appendix L

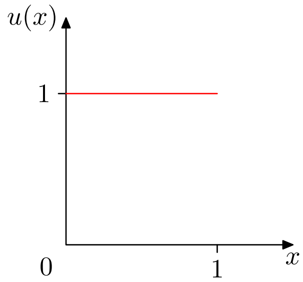
Considered Distributions

Distributions on [0, 1]

- uniform distribution:

$$u(x) := \begin{cases} 1 & \text{for } 0 \leq x \leq 1 \\ 0 & \text{for } x < 0 \text{ or } x > 1 \end{cases}$$

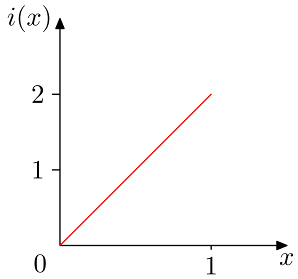
$$U(x) := \begin{cases} 0 & \text{for } 0 < x \\ x & \text{for } 0 \leq x \leq 1 \\ 1 & \text{for } x > 1 \end{cases}$$



- increasing distribution:

$$i(x) := \begin{cases} 2x & \text{for } 0 \leq x \leq 1 \\ 0 & \text{for } x < 0 \text{ or } x > 1 \end{cases}$$

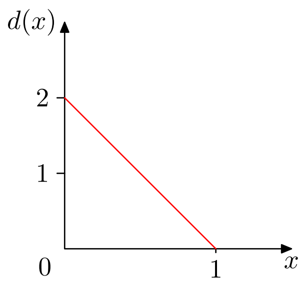
$$I(x) := \begin{cases} 0 & \text{for } 0 < x \\ x^2 & \text{for } 0 \leq x \leq 1 \\ 1 & \text{for } x > 1 \end{cases}$$



- decreasing distribution:

$$d(x) := \begin{cases} 2 - 2x & \text{for } 0 \leq x \leq 1 \\ 0 & \text{for } x < 0 \text{ or } x > 1 \end{cases}$$

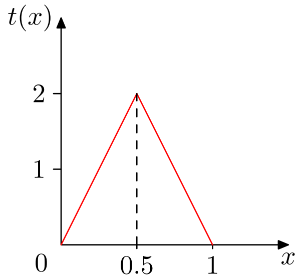
$$D(x) := \begin{cases} 0 & \text{for } 0 < x \\ 2x - x^2 & \text{for } 0 \leq x \leq 1 \\ 1 & \text{for } x > 1 \end{cases}$$



- triangular distribution:

$$t(x) := \begin{cases} 4x & \text{for } 0 \le x \le \frac{1}{2} \\ 4 - 4x & \text{for } \frac{1}{2} \le x \le 1 \\ 0 & \text{for } x < 0 \text{ and } x > 1 \end{cases}$$

$$T(x) := \begin{cases} 0 & \text{for } x < 0 \\ 2x^2 & \text{for } 0 \le x \le \frac{1}{2} \\ -2x^2 + 4x - 1 & \text{for } \frac{1}{2} \le x \le 1 \\ 1 & \text{for } x > 1 \end{cases}$$

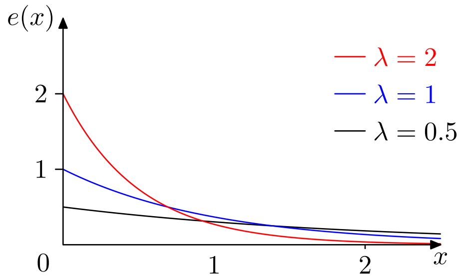


Distributions on $[0, \infty)$

- exponential distribution with parameter $\lambda > 0$:

$$e_\lambda(x) := \begin{cases} \lambda \mathrm{e}^{-\lambda x} & \text{for } x \le 0 \\ 0 & \text{for } x < 0 \end{cases}$$

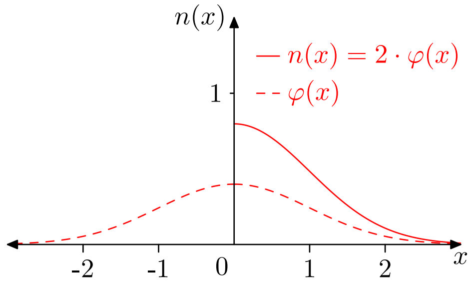
$$E_\lambda(x) := \begin{cases} 1 - \mathrm{e}^{-\lambda x} & \text{for } x \ge 0 \\ 0 & \text{for } x < 0 \end{cases}$$



- "right half" of the standard normal distribution (cf. standard normal distribution on the next page):

$$\overline{\varphi}(x) := \begin{cases} 2 \cdot \varphi(x) = \frac{2}{\sqrt{2\pi}} \mathrm{e}^{-\frac{x^2}{2}} & \text{for } 0 \le x \\ 0 & \text{for } x < 0 \end{cases}$$

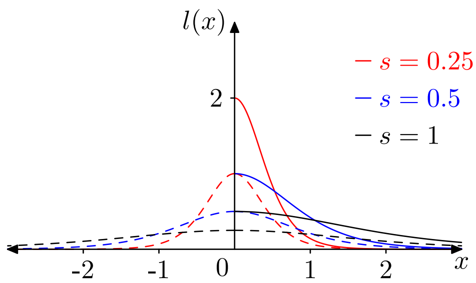
$$\overline{\Phi}(x) := \begin{cases} 2 \cdot \Phi(x) - 1 & \text{for } x \ge 0 \\ 0 & \text{for } x < 0 \end{cases}$$



- "right half" of the logistic distribution with expected value $\mu = 0$ and scale parameter $s > 0$ (cf. logistic distribution on the next page):

$$\overline{l}_s(x) := \begin{cases} 2 \cdot l(x) = \frac{2\mathrm{e}^{-x/s}}{s\left(1+\mathrm{e}^{-x/s}\right)^2} & \text{for } 0 \le x \\ 0 & \text{for } x < 0 \end{cases}$$

$$\overline{L}_s(x) := \begin{cases} 2 \cdot L(x) - 1 & \text{for } 0 \le x \\ 0 & \text{for } x < 0 \end{cases}$$

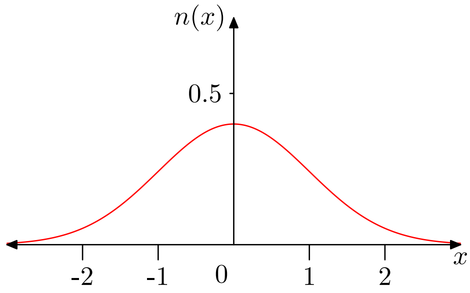


Distributions on $(-\infty, \infty)$

- standard normal distribution:

$$\varphi(x) := \frac{1}{\sqrt{2\pi}} \, \mathrm{e}^{-\frac{x^2}{2}}$$

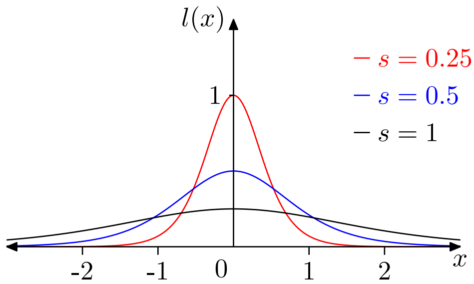
$$\Phi(x) := \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{x} \mathrm{e}^{-\frac{x^2}{2}} \, \mathrm{d}x$$



- logistic distribution with expected value $\mu = 0$ and scale parameter $s > 0$:

$$l_s(x) := \frac{\mathrm{e}^{-x/s}}{s\,(1+\mathrm{e}^{-x/s})^2}$$

$$L_s(x) := \frac{\mathrm{e}^{-x/s}}{(1+\mathrm{e}^{-x/s})^2}$$



Appendix M

Maple Worksheets

M.1 Best Solutions Rule for Optimization Problems with Density f

$\mathbf{P}(\mathbf{d}_1 \geq \mathbf{d}_2)$ for Minimization Problems

 # *Uniform Distribution on* [0, 1] :

$f := x \to piecewise(0 \leq x \leq 1, 1, 0)$:

$F := x \to piecewise(x < 0, 0, 0 \leq x \leq 1, x, 1 < x, 1)$:

$simplify\left(n \cdot (n-1) \cdot (n-2) \cdot \int_{-\infty}^{\infty} (1 - F(x))^{n-3} \cdot f(x) \cdot \int_{-\infty}^{x} f(y) \cdot F(2 \cdot y - x) \, \mathrm{d}y \, \mathrm{d}x\right) \text{assuming}(n :: integer, n > 2)$

$$\frac{1}{2} \tag{1}$$

 # *Increasing Distribution on* [0, 1] :

$f := x \to piecewise(0 \leq x \leq 1, 2\,x, 0)$:

$F := x \to piecewise\left(x < 0, 0, 0 \leq x \leq 1, x^2, 1 < x, 1\right)$:

$simplify\left(n \cdot (n-1) \cdot (n-2) \cdot \int_{-\infty}^{\infty} (1 - F(x))^{n-3} \cdot f(x) \cdot \int_{-\infty}^{x} f(y) \cdot F(2 \cdot y - x) \, \mathrm{d}y \, \mathrm{d}x\right) \text{assuming}(n :: integer, n > 2)$

$$\frac{7}{12} \tag{2}$$

 # *Decreasing Distribution on* [0, 1] :

$f := x \to piecewise(0 \leq x \leq 1, 2 - 2\,x, 0)$:

$F := x \to piecewise\left(x < 0, 0, 0 \leq x \leq 1, 2\,x - x^2, 1 < x, 1\right)$:

$simplify\left(n \cdot (n-1) \cdot (n-2) \cdot \int_{-\infty}^{\infty} (1 - F(x))^{n-3} \cdot f(x) \cdot \int_{-\infty}^{x} f(y) \cdot F(2 \cdot y - x) \, \mathrm{d}y \, \mathrm{d}x\right) \text{assuming}(n :: integer, n > 2)$

$$\frac{1}{4} \, \frac{4\,n - 7}{-3 + 2\,n} \tag{3}$$

 # *Centered Triangular Distribution on* [0, 1] :

$f := x \to piecewise\left(0 \leq x \leq \frac{1}{2}, 4\,x, \frac{1}{2} < x \leq 1, 4 - 4\,x, 0\right)$:

$F := x \to piecewise\left(x < 0, 0, 0 \leq x \leq \frac{1}{2}, 2x^2, \frac{1}{2} < x \leq 1, -2\,x^2 + 4\,x - 1, 1 < x, 1\right)$:

$simplify\left(n \cdot (n-1) \cdot (n-2) \cdot \int_{-\infty}^{\infty} (1 - F(x))^{n-3} \cdot f(x) \cdot \int_{-\infty}^{x} f(y) \cdot F(2 \cdot y - x) \, \mathrm{d}y \, \mathrm{d}x\right) \text{assuming}(n :: integer, n > 2)$

$$-\frac{1}{12} \, \frac{10\,2^{-n}\,n - 14\,n - 12\,2^{-1-n} + 21}{-3 + 2\,n} \tag{4}$$

$Int1 := simplify\left(\int_{0}^{\frac{1}{2}} 4\,c \cdot \left(1 - 2\,c^2\right)^{n-3} \int_{\frac{c}{2}}^{c} 4\,b \cdot 2 \cdot (2\,b - c)^2 \, \mathrm{d}b \, \mathrm{d}c\right) \text{assuming}(n :: integer, n > 2)$

$$-\frac{7}{24} \, \frac{-2 + 2^{-n}\,n^2 + 2\,2^{-n} + 2^{-n}\,n}{n\left(-3\,n + 2 + n^2\right)} \tag{5}$$

$\mathbf{P}\left(\mathbf{d}_1 \geq \mathbf{d}_2\right)$ for Maximization Problems

 # Uniform Distribution on $[\mathbf{0}, \mathbf{1}]$ **:**

$f := x \rightarrow piecewise(0 \leq x \leq 1, 1, 0)$:

$F := x \rightarrow piecewise(x < 0, 0, 0 \leq x \leq 1, x, 1 < x, 1)$:

$simplify\left(n \cdot (n-1) \cdot (n-2) \cdot \int_0^{\infty} F(x)^{n-3} \cdot f(x) \cdot \int_x^{\infty} f(y) \cdot (1 - F(2 \cdot y - x))\, \mathrm{d}y\, \mathrm{d}x\right)$ assuming$(n :: integer, n > 2)$

$$\frac{1}{2} \tag{1}$$

 # Increasing Distribution on $[\mathbf{0}, \mathbf{1}]$ **:**

$f := x \rightarrow piecewise(0 \leq x \leq 1, 2\,x, 0)$:

$F := x \rightarrow piecewise\left(x < 0, 0, 0 \leq x \leq 1, x^2, 1 < x, 1\right)$:

$simplify\left(n \cdot (n-1) \cdot (n-2) \cdot \int_0^{\infty} F(x)^{n-3} \cdot f(x) \cdot \int_x^{\infty} f(y) \cdot (1 - F(2 \cdot y - x))\, \mathrm{d}y\, \mathrm{d}x\right)$ assuming$(n :: integer, n > 2)$

$$\frac{1}{4}\,\frac{4\,n-7}{-3+2\,n} \tag{2}$$

 # Decreasing Distribution on $[\mathbf{0}, \mathbf{1}]$ **:**

$f := x \rightarrow piecewise(0 \leq x \leq 1, 2 - 2\,x, 0)$:

$F := x \rightarrow piecewise\left(x < 0, 0, 0 \leq x \leq 1, 2\,x - x^2, 1 < x, 1\right)$:

$simplify\left(n \cdot (n-1) \cdot (n-2) \cdot \int_0^{\infty} F(x)^{n-3} \cdot f(x) \cdot \int_x^{\infty} f(y) \cdot (1 - F(2 \cdot y - x))\, \mathrm{d}y\, \mathrm{d}x\right)$ assuming$(n :: integer, n > 2)$

$$\frac{7}{12} \tag{3}$$

 # Centered Triangular Distribution on $[\mathbf{0}, \mathbf{1}]$ **:**

$f := x \rightarrow piecewise\left(0 \leq x \leq \frac{1}{2}, 4\,x, \frac{1}{2} < x \leq 1, 4 - 4\,x, 0\right)$:

$F := x \rightarrow piecewise\left(x < 0, 0, 0 \leq x \leq \frac{1}{2}, 2x^2, \frac{1}{2} < x \leq 1, -2\,x^2 + 4\,x - 1, 1 < x, 1\right)$:

$simplify\left(n \cdot (n-1) \cdot (n-2) \cdot \int_0^{\infty} F(x)^{n-3} \cdot f(x) \cdot \int_x^{\infty} f(y) \cdot (1 - F(2 \cdot y - x))\, \mathrm{d}y\, \mathrm{d}x\right)$ assuming$(n :: integer, n > 2)$

$$\frac{1}{12}\,\frac{-10\,2^{-n}\,n + 14\,n - 21 + 12\,2^{-1-n}}{-3+2\,n} \tag{4}$$

 # Exponential Distribution:

$f := (x, \lambda) \rightarrow piecewise\left(x \geq 0, \lambda \cdot \mathrm{e}^{-\lambda \cdot x}, 0\right)$:

$F := (x, \lambda) \rightarrow piecewise\left(x \geq 0, 1 - \mathrm{e}^{-\lambda \cdot x}, 0\right)$:

$simplify\left(n \cdot (n-1) \cdot (n-2) \cdot \int_0^{\infty} F(x, \lambda)^{n-3} \cdot f(x, \lambda) \cdot \int_x^{\infty} f(y, \lambda) \cdot (1 - F(2 \cdot y - x, \lambda))\, \mathrm{d}y\, \mathrm{d}x\right)$ assuming$(n :: integer, n > 2, \lambda > 0)$

$$\frac{2}{3} \tag{5}$$

$\mathbf{E(d_1)}$ und $\mathbf{E(d_2)}$ for Minimization Problems

Uniform Distribution on [0, 1] :

$f := x \rightarrow piecewise(0 \le x \le 1, 1, 0)$:

$F := x \rightarrow piecewise(x < 0, 0, 0 \le x \le 1, x, 1 < x, 1)$:

$$E_1 := simplify\left(n \cdot (n-1) \cdot \int_{-\infty}^{\infty} (1 - F(x))^{n-2} \cdot f(x) \cdot \int_{-\infty}^{x} F(y)\, dy\, dx\right) \text{assuming}(n :: integer, n > 2)$$

$$\frac{1}{n+1} \tag{1}$$

$$E_2 := simplify\left(n \cdot (n-1) \cdot (n-2) \int_{-\infty}^{\infty} (1 - F(x))^{n-3} \cdot f(x) \cdot \int_{-\infty}^{x} F(y) \cdot f(y) \cdot (x-y)\, dy\, dx\right)$$
assuming$(n :: integer, n > 2)$

$$\frac{1}{n+1} \tag{2}$$

$$simplify\left(\frac{E_1}{E_1 + E_2}\right)$$

$$\frac{1}{2} \tag{3}$$

Increasing Distribution on [0, 1] :

$f := x \rightarrow piecewise(0 \le x \le 1, 2\,x, 0)$:

$F := x \rightarrow piecewise(x < 0, 0, 0 \le x \le 1, x^2, 1 < x, 1)$:

$$E_1 := simplify\left(n \cdot (n-1) \cdot \int_{-\infty}^{\infty} (1 - F(x))^{n-2} \cdot f(x) \cdot \int_{-\infty}^{x} F(y)\, dy\, dx\right) \text{assuming}(n :: integer, n > 2)$$

$$\frac{1}{3}\, n\, (n-1)\, \mathrm{B}\left(\frac{5}{2}, n-1\right) \tag{4}$$

$$E_2 := simplify\left(n \cdot (n-1) \cdot (n-2) \int_{-\infty}^{\infty} (1 - F(x))^{n-3} \cdot f(x) \cdot \int_{-\infty}^{x} F(y) \cdot f(y) \cdot (x-y)\, dy\, dx\right)$$
assuming$(n :: integer, n > 2)$

$$\frac{1}{10}\, n\, (n-1)\, (n-2)\, \mathrm{B}\left(n-2, \frac{7}{2}\right) \tag{5}$$

$$simplify\left(\frac{E_1}{E_1 + E_2}\right)$$

$$\frac{10\, \mathrm{B}\left(n-1, \frac{5}{2}\right)}{10\, \mathrm{B}\left(n-1, \frac{5}{2}\right) + 3\, \mathrm{B}\left(n-2, \frac{7}{2}\right) n - 6\, \mathrm{B}\left(n-2, \frac{7}{2}\right)} \tag{6}$$

Decreasing Distribution on [0, 1] :

$f := x \rightarrow piecewise(0 \le x \le 1, 2 - 2\,x, 0)$:

$F := x \rightarrow piecewise(x < 0, 0, 0 \le x \le 1, 2\,x - x^2, 1 < x, 1)$:

$$E_1 := simplify\left(n \cdot (n-1) \cdot \int_{-\infty}^{\infty} (1 - F(x))^{n-2} \cdot f(x) \cdot \int_{-\infty}^{x} F(y)\, dy\, dx\right) \text{assuming}(n :: integer, n > 2)$$

$$\frac{2\,n}{4\,n^2 - 1} \tag{7}$$

$$E_2 := simplify\left(n\cdot(n-1)\cdot(n-2)\int_{-\infty}^{\infty}(1-F(x))^{n-3}\cdot f(x)\cdot\int_{-\infty}^{x}F(y)\cdot f(y)\cdot(x-y)\,dy\,dx\right) \text{ assuming}(n :: integer, n > 2)$$

$$\frac{4\,(n-1)\,n}{8\,n^3-12\,n^2-2\,n+3} \quad \textbf{(8)}$$

$$simplify\left(\frac{E_1}{E_1+E_2}\right)$$

$$\frac{2\,n-3}{4\,n-5} \quad \textbf{(9)}$$

 # *Centered Triangular Distribution on* [0, 1] — *no simple formulas have been found* :

$$f := x \to piecewise\left(0 \le x \le \frac{1}{2}, 4\,x, \frac{1}{2} < x \le 1, 4-4\,x, 0\right) :$$

$$F := x \to piecewise\left(x < 0, 0, 0 \le x \le \frac{1}{2}, 2x^2, \frac{1}{2} < x \le 1, -2\,x^2+4\,x-1, 1 < x, 1\right) :$$

$$E_1 := simplify\left(n\cdot(n-1)\cdot\int_{-\infty}^{\infty}(1-F(x))^{n-2}\cdot f(x)\cdot\int_{-\infty}^{x}F(y)\,dy\,dx\right)\text{assuming}(n :: integer, n > 2) :$$

$$E_2 := simplify\left(n\cdot(n-1)\cdot(n-2)\int_{-\infty}^{\infty}(1-F(x))^{n-3}\cdot f(x)\cdot\int_{-\infty}^{x}F(y)\cdot f(y)\cdot(x-y)\,dy\,dx\right) \text{ assuming}(n :: integer, n > 2) :$$

$$simplify\left(\frac{E_1}{E_1+E_2}\right) :$$

 # *Exponential Distribution:*

$$f := (x, \lambda) \to piecewise\left(x \ge 0, \lambda\cdot e^{-\lambda\cdot x}, 0\right) :$$

$$F := (x, \lambda) \to piecewise\left(x \ge 0, 1-e^{-\lambda\cdot x}, 0\right) :$$

$$E_1 := simplify\left(n\cdot(n-1)\cdot\int_{-\infty}^{\infty}(1-F(x,\lambda))^{n-2}\cdot f(x,\lambda)\cdot\int_{-\infty}^{x}F(y,\lambda)\,dy\,dx\right)\text{assuming}(n :: integer, n > 2, \lambda > 0)$$

$$\frac{1}{\lambda\,(n-1)} \quad \textbf{(10)}$$

$$E_2 := simplify\left(n\cdot(n-1)\cdot(n-2)\int_{-\infty}^{\infty}(1-F(x,\lambda))^{n-3}\cdot f(x,\lambda)\cdot\int_{-\infty}^{x}F(y,\lambda)\cdot f(y,\lambda)\cdot(x-y)\,dy\,dx\right)\text{assuming}(n :: integer, n > 2, \lambda > 0)$$

$$\frac{1}{\lambda\,(n-2)} \quad \textbf{(11)}$$

$$simplify\left(\frac{E_1}{E_1+E_2}\right)$$

$$\frac{n-2}{2\,n-3} \quad \textbf{(12)}$$

$E(d_1)$ and $E(d_2)$ for Maximization Problems

 # *Uniform Distribution on* [0, 1] :

$f := x \rightarrow piecewise(0 \le x \le 1, 1, 0)$:

$F := x \rightarrow piecewise(x < 0, 0, 0 \le x \le 1, x, 1 < x, 1)$:

$E_1 := simplify\left(n \cdot (n-1) \cdot \int_{-\infty}^{\infty} F(x)^{n-2} \cdot f(x) \cdot \int_{x}^{\infty} f(y) \cdot (y-x)\, \mathrm{d}y\, \mathrm{d}x\right)$ assuming$(n :: integer, n > 2)$

$$\frac{1}{n+1} \tag{1}$$

$E_2 := simplify\left(n \cdot (n-1) \cdot (n-2) \int_{-\infty}^{\infty} F(x)^{n-3} \cdot f(x) \cdot \int_{x}^{\infty} (1 - F(y)) \cdot f(y) \cdot (y-x)\, \mathrm{d}y\, \mathrm{d}x\right)$ assuming$(n :: integer, n > 2)$

$$\frac{1}{n+1} \tag{2}$$

$simplify\left(\frac{E_1}{E_1 + E_2}\right)$

$$\frac{1}{2} \tag{3}$$

 # *Increasing Distribution on* [0, 1] :

$f := x \rightarrow piecewise(0 \le x \le 1, 2\,x, 0)$:

$F := x \rightarrow piecewise\left(x < 0, 0, 0 \le x \le 1, x^2, 1 < x, 1\right)$:

$E_1 := simplify\left(n \cdot (n-1) \cdot \int_{-\infty}^{\infty} F(x)^{n-2} \cdot f(x) \cdot \int_{x}^{\infty} f(y) \cdot (y-x)\, \mathrm{d}y\, \mathrm{d}x\right)$ assuming$(n :: integer, n > 2)$

$$\frac{2\,n}{4\,n^2 - 1} \tag{4}$$

$E_2 := simplify\left(n \cdot (n-1) \cdot (n-2) \int_{-\infty}^{\infty} F(x)^{n-3} \cdot f(x) \cdot \int_{x}^{\infty} (1 - F(y)) \cdot f(y) \cdot (y-x)\, \mathrm{d}y\, \mathrm{d}x\right)$ assuming$(n :: integer, n > 2)$

$$\frac{4\,(n-1)\,n}{8\,n^3 - 12\,n^2 - 2\,n + 3} \tag{5}$$

$simplify\left(\frac{E_1}{E_1 + E_2}\right)$

$$\frac{2\,n-3}{4\,n-5} \tag{6}$$

Decreasing Distribution on [**0, 1**] - *no simple formulas have been found* :

$f := x \rightarrow piecewise(0 \le x \le 1, 2 - 2x, 0)$:

$F := x \rightarrow piecewise(x < 0, 0, 0 \le x \le 1, 2x - x^2, 1 < x, 1)$:

$E_1 := simplify\left(n \cdot (n-1) \cdot \int_{-\infty}^{\infty} F(x)^{n-2} \cdot f(x) \cdot \int_{x}^{\infty} f(y) \cdot (y - x)\, \mathrm{d}y\, \mathrm{d}x\right) \text{assuming}(n :: integer, n > 2)$:

$E_2 := simplify\left(n \cdot (n-1) \cdot (n-2) \int_{-\infty}^{\infty} F(x)^{n-3} \cdot f(x) \cdot \int_{x}^{\infty} (1 - F(y)) \cdot f(y) \cdot (y - x)\, \mathrm{d}y\, \mathrm{d}x\right) \text{assuming}(n :: integer, n > 2)$:

Centered Triangular Distribution on [**0, 1**] - *no simple formulas have been found* :

$f := x \rightarrow piecewise\left(0 \le x \le \frac{1}{2}, 4x, \frac{1}{2} < x \le 1, 4 - 4x, 0\right)$:

$F := x \rightarrow piecewise\left(x < 0, 0, 0 \le x \le \frac{1}{2}, 2x^2, \frac{1}{2} < x \le 1, -2x^2 + 4x - 1, 1 < x, 1\right)$:

$E_1 := simplify\left(n \cdot (n-1) \cdot \int_{-\infty}^{\infty} F(x)^{n-2} \cdot f(x) \cdot \int_{x}^{\infty} f(y) \cdot (y - x)\, \mathrm{d}y\, \mathrm{d}x\right) \text{assuming}(n :: integer, n > 2)$:

$E_2 := simplify\left(n \cdot (n-1) \cdot (n-2) \int_{-\infty}^{\infty} F(x)^{n-3} \cdot f(x) \cdot \int_{x}^{\infty} (1 - F(y)) \cdot f(y) \cdot (y - x)\, \mathrm{d}y\, \mathrm{d}x\right) \text{assuming}(n :: integer, n > 2)$:

Exponential Distribution:

$f := (x, \lambda) \rightarrow piecewise(x \ge 0, \lambda \cdot \mathrm{e}^{-\lambda \cdot x}, 0)$:

$F := (x, \lambda) \rightarrow piecewise(x \ge 0, 1 - \mathrm{e}^{-\lambda \cdot x}, 0)$:

$E_1 := simplify\left(n \cdot (n-1) \cdot \int_{-\infty}^{\infty} F(x, \lambda)^{n-2} \cdot f(x, \lambda) \cdot \int_{x}^{\infty} f(y, \lambda) \cdot (y - x)\, \mathrm{d}y\, \mathrm{d}x\right) \text{assuming}(n :: integer, n > 2, \lambda > 0)$

$$\frac{1}{\lambda} \tag{7}$$

$E_2 := simplify\left(n \cdot (n-1) \cdot (n-2) \int_{-\infty}^{\infty} F(x, \lambda)^{n-3} \cdot f(x, \lambda) \cdot \int_{x}^{\infty} (1 - F(y, \lambda)) \cdot f(y, \lambda) \cdot (y - x)\, \mathrm{d}y\, \mathrm{d}x\right) \text{assuming}(n :: integer, n > 2, \lambda > 0)$

$$\frac{1}{2\lambda} \tag{8}$$

$simplify\left(\frac{E_1}{E_1 + E_2}\right)$

$$\frac{2}{3} \tag{9}$$

M.2 Best Solutions Rule for Random Σ-Type Problems

$P\left(d_1 > d_2\right)$ for Random Σ-Type Problems

Uniform Distribution on [0, 1]:
$f := x \to piecewise(0 \le x \le 1, 1, 0):$
$F := x \to piecewise(x < 0, 0, 0 \le x \le 1, x, 1 < x, 1):$
$P_u := simplify\left(n \cdot (n-1) \cdot \int_0^{\infty} (1 - F(x))^{n-2} \cdot \left(F(x) - F\left(\frac{x}{2}\right)\right) \cdot f(x)\, \mathrm{d}x\right)$ assuming$(n :: integer, n > 2)$

$$\frac{1}{2} \tag{1}$$

Increasing Distribution on [0, 1]:
$f := x \to piecewise(0 \le x \le 1, 2x, 0):$
$F := x \to piecewise\left(x < 0, 0, 0 \le x \le 1, x^2, 1 < x, 1\right):$
$P_i := simplify\left(n \cdot (n-1) \cdot \int_0^{\infty} (1 - F(x))^{n-2} \cdot \left(F(x) - F\left(\frac{x}{2}\right)\right) \cdot f(x)\, \mathrm{d}x\right)$ assuming$(n :: integer, n > 2)$

$$\frac{3}{4} \tag{2}$$

Decreasing Distribution on [0, 1]:
$f := x \to piecewise(0 \le x \le 1, 2 - 2x, 0):$
$F := x \to piecewise\left(x < 0, 0, 0 \le x \le 1, 2x - x^2, 1 < x, 1\right):$
$P_d := simplify\left(n \cdot (n-1) \cdot \int_0^{\infty} (1 - F(x))^{n-2} \cdot \left(F(x) - F\left(\frac{x}{2}\right)\right) \cdot f(x)\, \mathrm{d}x\right)$ assuming$(n :: integer, n > 2)$

$$\frac{1}{4} \frac{4n-3}{2n-1} \tag{3}$$

Centered Triangular Distribution on [0, 1]:
$f := x \to piecewise\left(0 \le x \le \frac{1}{2}, 4x, \frac{1}{2} < x \le 1, 4 - 4x, 0\right):$
$F := x \to piecewise\left(x < 0, 0, 0 \le x \le \frac{1}{2}, 2x^2, \frac{1}{2} < x \le 1, -2x^2 + 4x - 1, 1 < x, 1\right):$
$P_t := simplify\left(n \cdot (n-1) \cdot \int_0^{\infty} (1 - F(x))^{n-2} \cdot \left(F(x) - F\left(\frac{x}{2}\right)\right) \cdot f(x)\, \mathrm{d}x\right)$ assuming$(n :: integer, n > 2)$

$$-\frac{1}{4} \frac{-6n + 4\,2^{-1-n} + 3}{2n-1} \tag{4}$$

Exponential Distribution:
$f := (x, \lambda) \to \lambda \cdot e^{-\lambda \cdot x}:$
$F := (x, \lambda) \to 1 - e^{-\lambda \cdot x}:$
$P_e := simplify\left(n \cdot (n-1) \cdot \int_0^{\infty} (1 - F(x, \lambda))^{n-2} \cdot \left(F(x, \lambda) - F\left(\frac{x}{2}, \lambda\right)\right) \cdot f(x, \lambda)\, \mathrm{d}x\right)$ assuming$(n :: integer, n > 2, \lambda > 0)$

$$\frac{n-1}{2n-1} \tag{5}$$

Expected Values $E(d_1)$ and $E(d_2)$ for Random Σ-Type Problems

 # *Uniform Distribution on* [0, 1] :

$f := x \to piecewise(0 \le x \le 1, 1, 0)$:

$F := x \to piecewise(x < 0, 0, 0 \le x \le 1, x, 1 < x, 1)$:

$E_1 := simplify\left(n \cdot \int_0^{\infty} x \cdot (1 - F(x))^{n-1} \cdot f(x)\, dx\right) \text{assuming}(n :: integer, n > 2)$

$$\frac{1}{n+1} \tag{1}$$

$E_2 := simplify\left(n \cdot (n-1) \cdot \int_0^{\infty} (1 - F(x))^{n-2} \cdot f(x) \int_0^{\frac{x}{2}} y \cdot f(y)\, dy\, dx\right) \text{assuming}(n :: integer, n > 2)$

$$\frac{1}{4\,(n+1)} \tag{2}$$

$simplify\left(\frac{E_1}{E_1 + E_2}\right)$

$$\frac{4}{5} \tag{3}$$

 # *Increasing Distribution on* [0, 1] :

$f := x \to piecewise(0 \le x \le 1, 2\,x, 0)$:

$F := x \to piecewise\left(x < 0, 0, 0 \le x \le 1, x^2, 1 < x, 1\right)$:

$E_1 := simplify\left(n \cdot \int_0^{\infty} x \cdot (1 - F(x))^{n-1} \cdot f(x)\, dx\right) \text{assuming}(n :: integer, n > 2)$

$$n\, \mathrm{B}\left(n, \frac{3}{2}\right) \tag{4}$$

$E_2 := simplify\left(n \cdot (n-1) \cdot \int_0^{\infty} (1 - F(x))^{n-2} \cdot f(x) \int_0^{\frac{x}{2}} y \cdot f(y)\, dy\, dx\right) \text{assuming}(n :: integer, n > 2)$

$$\frac{1}{12}\, n\,(n-1)\, \mathrm{B}\left(\frac{5}{2}, n-1\right) \tag{5}$$

$simplify\left(\frac{E_1}{E_1 + E_2}\right)$

$$\frac{12\, \mathrm{B}\left(n, \frac{3}{2}\right)}{12\, \mathrm{B}\left(n, \frac{3}{2}\right) + \mathrm{B}\left(n-1, \frac{5}{2}\right) n - \mathrm{B}\left(n-1, \frac{5}{2}\right)} \tag{6}$$

 # *Decreasing Distribution on* [0, 1] :

$f := x \to piecewise(0 \le x \le 1, 2 - 2\,x, 0)$:

$F := x \to piecewise\left(x < 0, 0, 0 \le x \le 1, 2\,x - x^2, 1 < x, 1\right)$:

$$E_1 := simplify\left(n\cdot\int_0^{\infty} x\cdot(1-F(x))^{n-1}\cdot f(x)\,dx\right) \text{assuming}(n :: integer, n > 2)$$

$$\frac{1}{2n+1} \quad \textbf{(7)}$$

$$E_2 := simplify\left(n\cdot(n-1)\cdot\int_0^{\infty}(1-F(x))^{n-2}\cdot f(x)\int_0^{\frac{x}{2}} y\cdot f(y)\,dy\,dx\right) \text{assuming}(n :: integer, n > 2)$$

$$\frac{1}{2}\ \frac{n}{4n^2-1} \quad \textbf{(8)}$$

$$simplify\left(\frac{E_1}{E_1+E_2}\right)$$

$$\frac{2(2n-1)}{5n-2} \quad \textbf{(9)}$$

 # Centered Triangular Distribution on* [0, 1] − *no simple formulas have been found :

$$f := x \to piecewise\left(0 \le x \le \frac{1}{2}, 4x, \frac{1}{2} < x \le 1, 4-4x, 0\right) :$$

$$F := x \to piecewise\left(x < 0, 0, 0 \le x \le \frac{1}{2}, 2x^2, \frac{1}{2} < x \le 1, -2x^2+4x-1, 1 < x, 1\right) :$$

$$E_1 := simplify\left(n\cdot\int_0^{\infty} x\cdot(1-F(x))^{n-1}\cdot f(x)\,dx\right) \text{assuming}(n :: integer, n > 2) :$$

$$E_2 := simplify\left(n\cdot(n-1)\cdot\int_0^{\infty}(1-F(x))^{n-2}\cdot f(x)\int_0^{\frac{x}{2}} y\cdot f(y)\,dy\,dx\right) \text{assuming}(n :: integer, n > 2) :$$

 # Exponential Distribution:

$$f := (x, \lambda) \to \lambda\cdot e^{-\lambda\cdot x} :$$

$$F := (x, \lambda) \to 1 - e^{-\lambda\cdot x} :$$

$$E_1 := simplify\left(n\cdot\int_0^{\infty} x\cdot(1-F(x,\lambda))^{n-1}\cdot f(x,\lambda)\,dx\right) \text{assuming}(n :: integer, n > 2, \lambda > 0)$$

$$\frac{1}{n\lambda} \quad \textbf{(10)}$$

$$E_2 := simplify\left(n\cdot(n-1)\cdot\int_0^{\infty}(1-F(x,\lambda))^{n-2}\cdot f(x,\lambda)\int_0^{\frac{x}{2}} y\cdot f(y,\lambda)\,dy\,dx\right) \text{assuming}(n :: integer, n > 2, \lambda > 0)$$

$$\frac{n}{(4n^2-4n+1)\lambda} \quad \textbf{(11)}$$

$$simplify\left(\frac{E_1}{E_1+E_2}\right)$$

$$\frac{4n^2-4n+1}{5n^2-4n+1} \quad \textbf{(12)}$$

Bibliography

[ABN 1992] B. C. Arnold, N. Balakrishnan, and H. N. Nagaraja. *A First Course in Order Statistics*. John Wiley & Sons, Inc., 1992.

[ABS 2002] I. Althöfer, F. Berger, and S. Schwarz. *Generating True Alternatives with a Penalty Method*. Tech. Rep., 2002. `http://www.minet.uni-jena.de/Math-Net/reports/shadows/02-04report.html`.

[ABS 2012] I. Althöfer, E. Bleicher, and L. Schreiber. *Endgames statistics on Oskar Cordel's "Three Moves Law"*. Report in preparation. 2012.

[Alt 2009] I. Althöfer. *Gedanken zum 3-Züge-Gesetz von Oskar Cordel*. 2009. Available since May 2009, Retrieved at 06-06-2011.

`http://www.althofer.de/zu-cordel.pdf`.

[AMO 1993] R. K. Ahuja, T. L. Magnanti, and J. B. Orlin. *Network flows: theory, algorithms and applications*. Prentice-Hall, 1993.

[Ber 2000] F. Berger. *K alternative paths instead of k shortest paths*. Diploma thesis, Faculty of Mathematics and Computer Science, Friedrich-Schiller-University Jena, Germany, 2000.

[BJS 1990] M. S. Bazaraa, J. J. Jarvis, and H. D. Sherali. *Linear Programming and Network Flows*. John Wiley & Sons, Inc., 1990.

[Ble 2005] E. Bleicher. *Online-Query of the Nalimov Tablebases*. available since 2005. Retrieved at 06-06-2011.

`http://www.k4it.de/index.php?topic=egtb&lang=en`.

[Cor 1913] O. Cordel. *Theorie und Praxis des Schachspiels, II. Band*. A. Stein's Verlagsbuchhandlung, Kaiser Wilhelm-Straße 52, Potsdam, 1913.

[Dij 1959] E. W. Dijkstra. *A note on two problems in connexion with graphs*. Numerische Mathematik, Vol. 1, pp. 269–271, 1959.

[DIMACS] *9th DIMACS Implementation Challenge - Shortest Paths, United States Road Networks (TIGER/Line)*. Available since October 2005. Retrieved at 06-06-2011.

`http://www.dis.uniroma1.it/~challenge9/data/tiger/`.

[Dör 2009] M. Dörnfelder. *Penalty Methods in Discrete Optimization: On the Maximum Number of Threshold Parameters*. Doctoral Dissertation, Faculty of Mathematics and Computer Science, Friedrich-Schiller-University Jena, Germany, 2009.

[FIDE] FIDE - World Chess Federation. *Laws of Chess.* Valid since 1 July 2009.
`http://www.fide.com/fide/handbook.html`.

[HK 2000] H. W. Hamacher and K. Klamroth. *Lineare und Netzwerk-Optimierung: ein bilinguales Lehrbuch = Linear and Network Optimization: A bilingual textbook.* Vieweg, 2000.

[Jun 2005] D. Jungnickel. *Graphs, Networks and Algorithms.* Springer, second edition Ed., 2005.

[Lan 2004] C. Langton. *Intuitive Guide to Principles of Communications: Tutorial 23b - Trellis coded Modulation (TCM).* 2004.
`http://www.complextoreal.com/chapters/tcm.pdf`.

[Law 1972] E. L. Lawler. *A Procedure for Computing the K Best Solutions to Discrete Optimization Problems and Its Application to the Shortest Path Problem.* Managment Science, Vol. 18, No. 7, pp. 401–405, 1972.

[Les 2008] A. M. Lesk. *Introduction to Bioinformatics.* Oxford University Press, third edition Ed., 2008.

[Maple 2008] *Maple 12.0.* Maplesoft, a division of Waterloo Maple Inc., Waterloo, Ontario, 2008.

[Matlab 2008] *Matlab Version R2008b.* The MathWorks Inc., Natick, Massachusetts, 2008.

[MT 1990] S. Martello and P. Toth. *Knapsack Problems: Algorithms and Computer Implementations.* John Wiley & Sons, Inc., 1990.

[Mur 1968] K. G. Murty. *An Algorithm for Ranking all the Assignments in Order of Increasing Cost.* Operations Research, Vol. 16, pp. pp. 682–687, 1968.

[Sam 2005] J. Sameith. *Generating Alternative Solutions for Discrete Optimization Problems with Uncertain Data.* Doctoral Dissertation, Faculty of Mathematics and Computer Science, Friedrich-Schiller-University Jena, Germany, 2005. 3-Hirn-Verlag, Lage.

[Sch 2003] S. Schwarz. *On the Generation of Multiple Candidate Solutions for Discrete Optimization Problems.* Doctoral Dissertation, Faculty of Mathematics and Computer Science, Friedrich-Schiller-University Jena, Germany, 2003.

[Ung 1982] G. Ungerboeck. *Channel coding with multilevel/phase signals.* IEEE Transactions on Information Theory, Vol. 28, No. 1, pp. 55 – 67, January 1982.

[USCB] Census Bureau, U.S. *Technical Documentation of the UA Census 2000 Tiger/Line® Files.* Available since 2002. Retrieved at 06-06-2011.
`http://www.census.gov/geo/www/tiger/tigerua/ua2ktgr.pdf`.

Nomenclature

$G = (V, E)$	Graph with edge set E and set of vertices V
$\operatorname{lex\,min}$	Lexicographical minimization. The returned value is minimal concerning the first argument. Amongst all these solutions it is also minimal concerning the second argument.
$\mathcal{P}(E)$	Power set of E
S	Set of feasible solutions Sometimes also used for selection rules.

Classification of Moves

Refers to Chapter 1.

D	Draw
L	Loss
W	Win

Chess Pieces

Refers to Chapter 1.

In Forsyth-Edwards Notation (FEN) upper case letters are used for white pieces while lower case letters represent black pieces.

B, b	Bishop
K, k	King
N, n	Knight
P, p	Pawn
Q, q	Queen
R, r	Rook

The Generalized Cordel Property for Chess

Refers to Chapter 1.

CF(k)	Cordel Frequency for chess positions with exactly k chess pieces incl. kings, cf. Definition 1.3.3 (page 12)
(GeCoP)	Generalized Cordel Property, cf. Definition 1.2.8 (page 7) for chess under the best moves rule and Definition 1.2.10 (page 8) for chess with an arbitrary selection rule
m_1, m_2, m_3	Three moves chosen by a selection rule S, cf. Definition 1.2.11 (page 8)

The Generalized Cordel Property for Optimization Problems

Refers to Section 1.4 and Chapter 2, 3, and 4.

CF(T, S, R)	Cordel Frequency for a specific optimization problem T, a selection rule S, and a rule R for generating random instances of T, cf. Definition 1.4.4 (page 19)
CF(n)	Cordel Frequency, where n is a size parameter of the considered optimization problem
d_i	Difference of the functional values of x_1 and x_2, $d_i := \lvert f(x_i) - f(x_{i+1}) \rvert$
$\overline{d}_i$	Adjusted difference for the computation of the adjusted Cordel frequency, cf. Equations (3.1) and (3.2) (page 86)
(GeCoP)	Generalized Cordel Property, cf. Definition 1.4.1 (page 18)
x_1, x_2, x_3	Three feasible solutions chosen by a selection rule S, cf. Definition 1.2.11 (page 18)

The Penalty Method

Refers to Chapter 2 and 3.

$B^{(0)}$	An optimal solution
$B^{(\varepsilon)}$	ε-penalty alternative, ε-alternative
ε	Penalty parameter
$\varepsilon_0, \varepsilon_1, \varepsilon_2, \ldots$	Threshold parameters
$f_\varepsilon(B)$	Penalized value/weight of B
I_P	Optimality interval of the penalty alternative P
p	Penalty vector
$p(B)$	Penalized part of B
$P^{(i)}$	i-th penalty alternative
$P^{(0)}, P^{(1)}, \ldots, P^{(k-1)}$	k best penalty alternatives
w	Weight-vector
$w(B)$	Weight of B
$w^{(\varepsilon)}$	Penalized weight

The Generalized Cordel Property for Optimization Problems with Density *f*

Refers to Chapter 5.

$\mathrm{CF}_{f,\max}(n)$	Cordel frequency of a maximization problem with exactly n feasible solutions and density f
$\mathrm{CF}_{f,\min}(n)$	Cordel frequency of a minimization problem with exactly n feasible solutions and density f
f	Probability density function, describing the distribution of the functional values of the optimization problem
$V_1, \ldots V_n$	n independent random variables with density f representing the functional values of an optimization problem with exactly n feasible solutions and density f
$V_{1:n}, \ldots V_{1:n}$	Order statistics of $V_1, \ldots V_n$

Ehrenwörtliche Erklärung

Hiermit erkläre ich,

- dass mir die geltende Promotionsordnung der Fakultät für Mathematik und Informatik der Friedrich-Schiller-Universität Jena bekannt ist,
- dass ich die Dissertation selbst angefertigt habe, keine Textabschnitte oder Ergebnisse eines Dritten oder eigener Prüfungsarbeiten ohne Kennzeichnung übernommen und alle von mir benutzten Hilfsmittel, persönlichen Mitteilungen und Quellen in meiner Arbeit angegeben sind,
- dass alle Personen, die mich bei der Auswahl und Auswertung des Materials sowie bei der Herstellung des Manuskripts unterstützt haben in der Danksagung auf Seite vii genannt werden,
- dass die Hilfe eines Promotionsberaters nicht in Anspruch genommen wurde und dass Dritte weder unmittelbar noch mittelbar geldwerte Leistungen von mir für Arbeiten erhalten haben, die im Zusammenhang mit dem Inhalt der vorgelegten Dissertation stehen,
- dass ich die Dissertation noch nicht als Prüfungsarbeit für eine staatliche oder andere wissenschaftliche Prüfung eingereicht habe und
- dass ich die gleiche, eine in wesentlichen Teilen ähnliche oder eine andere Abhandlung nicht bei einer anderen Hochschule als Dissertation eingereicht habe.

Jena, 29.07.2012 Lisa Schreiber

Tabellarischer Lebenslauf

Persönliche Daten

Name	Lisa Schreiber
Anschrift	Kunitzer Straße 12c 07745 Jena
geboren am	30.07.1986
in	Jena

Schulbildung

1992 - 1996	Staatlich Grundschule „Heinrich Heine“ Jena
1996 - 1998	Staatliches Gymnasium „Am Anger“ Jena
1998 - 2000	Staatliches Gymnasium „Adolf Reichwein“ Jena
2000 - 2004	Mathematisch-naturwissenschaftlich-technischer Spezialschulteil am staatlichen Gymnasium „Carl Zeiss“ Jena Abschluss: Abitur (1,0)

Studium

10/2004 - 02/2009	Studium der Mathematik an der Friedrich-Schiller-Universität Jena Abschluss: Diplom (1,1)
04/2009 - 07/2012	Promotion am Lehrstuhl für Mathematische Optimierung von 04/2009 bis 03/2011 gefördert durch ein Landesgraduiertenstipendium

Berufstätigkeit

04/2011 - 01/2012	Wissenschaftliche Mitarbeiterin am Lehrstuhl für Mathematische Optimierung
seit 02/2012	Optiksystemdesignerin bei der Carl Zeiss AG