

Technische Universität Graz

Masterarbeit

**Heuristiken für das optimale Data-Arrangement-Problem in einem
Baum**

Rostislav Staněk*

Begutachterin:

Ao.Univ.-Prof. Dipl.-Ing. Dr.techn. Eranda Dragoti-Cela

17. Januar 2012

Deutsche Fassung:
Beschluss der Curricula-Kommission für Bachelor-, Master- und Diplomstudien vom 10.11.2008
Genehmigung des Senates am 1.12.2008

EIDESSTÄTLICHE ERKLÄRUNG

Ich erkläre an Eides statt, dass ich die vorliegende Arbeit selbstständig verfasst, andere als die angegebenen Quellen/Hilfsmittel nicht benutzt, und die den benutzten Quellen wörtlich und inhaltlich entnommene Stellen als solche kenntlich gemacht habe.

Graz, am

.....
(Unterschrift)

Englische Fassung:

STATUTORY DECLARATION

I declare that I have authored this thesis independently, that I have not used other than the declared sources / resources, and that I have explicitly marked all material which has been quoted either literally or by content from the used sources.

.....
date

.....
(signature)

Inhaltsverzeichnis

Vorwort	6
Danksagung	7
Kurzfassung	8
Abstract	9
1 Einführung	10
1.1 Problemstellung	10
1.1.1 Grundlegende Definitionen und Sätze	10
1.2 Praktische Anwendungen	15
2 Weitere Definitionen und Bemerkungen	16
3 Untere Schranken	21
3.1 Knotengrad-Schranke	22
4 Heuristiken	27
4.1 Randomisierte Heuristiken	27
4.1.1 Normales-Arrangement-Verfahren	27
4.1.2 Methode der zufälligen Arrangements	28
4.1.3 Methode der zufälligen zusammenhängenden Arrange- ments	31
4.1.4 Erwartungswert des Zielfunktionswertes eines zufälli- gen Arrangements	32
4.1.5 Varianz des Zielfunktionswertes eines zufälligen Arran- gements für einen Graphen mit einer festen Dichte	40
4.2 Greedy- und andere einfache Heuristiken	60
4.2.1 Greedy-Heuristik	60
4.2.2 Breitensuche- und Tiefensuche- basierte Greedy- Verfahren	64
4.2.3 Heuristische Methode der minimalen Schnitte in be- schränkten Mengen	66
4.3 Lokale Suche	75
4.3.1 Austausch-Heuristik	75
4.3.2 Shift-Flip-Heuristik	79

5	Spezialfälle	81
5.1	Spezialfall: G vollständig	81
5.1.1	G vollständig und $n = b$	81
5.1.2	G vollständig und $n < b$	82
5.2	Spezialfall: $d = n \geq 2$	83
5.3	Spezialfall: $d = n - 1 \geq 2$	83
5.4	Spezialfall: Sterne	85
5.5	Spezialfall: erweiterte Sterne	87
6	Numerische Ergebnisse	89
6.1	Kurze Beschreibung benutzter Testinstanzen	89
6.2	Kurze Beschreibung numerischer Ergebnisse	90
7	Kurze Zusammenfassung erreichter Ergebnisse	91
A	Anhang: Abbildungen, Tabellen und Bemerkungen	94
A.1	Allgemeine Abbildungen und Tabellen	94
A.2	Zusammenfassende Tabellen	98
	Literatur	101

Abbildungsverzeichnis

1.1	Beispiel 1.1: Graph	11
1.2	Beispiel 1.1: eine lineare Anordnung der Knoten	11
1.3	Beispiel 1.2: ein Arrangement der Knoten	13
1.4	Beispiel 1.2: ein Arrangement der Knoten – Zielfunktionswert – Berechnung	14
1.5	VLSI	15
2.1	Arrangement-Baum – Skizze mit Blätter-Indexen	17
2.2	Bemerkung 2.2: ein erweiterter Stern	18
2.3	Bemerkung 2.2: Optimallösung für $d = 4$	19
2.4	Bemerkung 2.3: Graph	20
2.5	Bemerkung 2.3: Optimallösung für $d = 5$	20
3.1	Beispiel 3.1: Knotengrad-Schranke – Graph G'_1	22
3.2	Beispiel 3.1: Knotengrad-Schranke – optimales Arrangement ϕ_1 für den Graphen G'_1	23
4.1	Empirischer prozentueller Anteil der optimalen Lösungen für $n = 5$ und $d = 3$ auf $G(5, p)$ mit $p = \frac{x}{100}$, wobei $x \in \mathbb{N}$ und $0 \leq x \leq 100$	30
4.2	Empirischer prozentueller Anteil der optimalen Lösungen für $n = 9$ und $d = 3$ auf $G(5, p)$ mit $p = \frac{x}{100}$, wobei $x \in \mathbb{N}$ und $0 \leq x \leq 100$	30
4.3	Empirischer prozentueller Anteil der optimalen (in Grau) bzw. der optimalen zusammenhängenden (in Rot) Lösungen für $n =$ 5 und $d = 3$ auf $G(5, p)$ mit $p = \frac{x}{100}$, wobei $x \in \mathbb{N}$ und $0 \leq$ $x \leq 100$	32
4.4	Beispiel 4.2: Graph	36
4.5	Beispiel 4.3: Verteilung der Zielfunktionswerte für $n = 5, d = 3$ und $d(G) = 0, 70$	39
4.6	Lemma 58: Graph $G = (V, E)$, wobei $ V = n$ und $ E = 1$	43
4.7	Beispiel 4.5: Varianz des Zielfunktionswertes $\mathbb{V}_z(G, d, \phi)$ eines zufälligen Arrangements für $n = 5, d = 3$ und verschiedene Werte von m (bzw. Dichten $d(G)$)	57
4.8	Beispiel 4.6: Arrangement ϕ_f	59
4.9	Beispiel 4.7: Graph	63
4.10	Beispiel 4.7: Greedy-Heuristik für $d = 2$ – Arrangement	64
4.11	Beispiel 4.8: Graph	67
4.12	Beispiel 4.8: Optimallösung für $d = 4$	67
4.13	Beispiel 4.9: Graph	74
4.14	Beispiel 4.9: Lösung für $d = 2$	74
4.15	Beispiel 4.9: Optimallösung für $d = 2$	75

4.16	Beispiel 4.10: normales Arrangement für $d = 3$	77
4.17	Beispiel 4.10: Lösung für $d = 3$	78
5.1	Beispiel 5.1: minimaler Schnitt bezüglich seiner Kapazität im Graphen G	85
5.2	Beispiel 5.1: Optimallösung für $d = 4$	85
A.1	Optimale lineare Anordnung der Knoten aus dem Beispiel 1.1 (Zielfunktionswert ist 10)	94
A.2	Optimales Arrangement aus dem Beispiel 1.2 (Zielfunktionswert ist 20)	94
A.3	Empirischer prozentueller Anteil der optimalen (in Grau) bzw. der optimalen zusammenhängenden (in Rot) Lösungen für unterschiedliche Werte von n und unterschiedliche Ordnungen der Arrangement-Bäume d auf $G(n, p)$	95
A.4	Verteilung der Zielfunktionswerte für unterschiedliche Werte von n , unterschiedliche Ordnungen des Arrangement-Baumes d und unterschiedliche Dichten $d(G)$	96
A.5	Varianz des Zielfunktionswertes $\mathbb{V}z(G, d, \phi)$ eines zufälligen Arrangements für verschiedene Werte von n und m (bzw. von $d(G)$)	97

Tabellenverzeichnis

1.1	Beispiel 1.1: Zielfunktionswert zur Anordnung bei Abb. 1.2	12
1.2	Beispiel 1.2: Zielfunktionswert zum Arrangement in Abb. 1.3	14
3.1	Beispiel 3.1: s_v für alle Knoten	26
4.1	Beispiel 4.1: Zielfunktionswerteverteilung	28
4.2	Beispiel 4.1: Wahrscheinlichkeit, dass für dieses Beispiel ein Optimum mithilfe der Methode der zufälligen Arrangements getroffen wird	29
4.3	Beispiel 4.2: alle unterschiedlichen Arrangements abgesehen von Knotenpermutationen	37
4.4	Beispiel 4.2: alle möglichen Beiträge von Blätterpaaren und die Wahrscheinlichkeiten, dass sie mit einer Kante verbunden sind	38
4.5	Beispiel 4.5: Varianz des Zielfunktionswertes $\mathbb{V}z(G, d, \phi)$ eines zufälligen Arrangements für $n = 5$, $d = 3$ und verschiedene Werte von m (bzw. Dichten $d(G)$)	56
4.6	Beispiel 4.6: Arrangement ϕ	58
4.7	Beispiel 4.6: Arrangement ϕ_f	58
4.8	Beispiel 4.7: Zielfunktionswerteverteilung	64

5.1	Beispiel 5.1: alle Schnitte $\emptyset \neq X \subset V$ und derer Kapazitäten	84
A.1	Komplette Enumeration – Liste der komplett enumerierten Instanzen	98
A.2	Zusammenfassung – erreichte Ergebnisse für Testinstanzen	100

Algorithmenverzeichnis

4.1	Greedy-Heuristik – Algorithmus 1	61
4.2	Greedy-Heuristik – Algorithmus 2	62
4.3	Breitensuche- und Tiefensuche- basierte Greedy-Verfahren, in den alle Knoten als Startknoten benutzt werden und das beste daraus resultierende Arrangement ausgewählt wird	65
4.4	Heuristische Methode der minimalen Schnitte in beschränkten Mengen – allgemeiner Algorithmus	69
4.5	MinSPBM – Heuristik	70
4.6	Austausch-Heuristik – Algorithmus 1	76
4.7	Austausch-Heuristik – Algorithmus 2	79

Vorwort

Wenn wir zurück an die Geschichte der Informatik oder jeder anderen Wissenschaft denken, entsteht früher oder später die Frage, wohin sie eigentlich gehört und wo genau ihre Grenzen sind. Sicherlich gibt es immer einen Kern, den die meisten als (in diesem Fall) „Informatik“ tatsächlich bezeichnen würden, dann gibt es aber jene Randgebiete, die mehr oder weniger einem anderen Fach nahe liegen. Und genau zu einem solchen Grenzgebiet gehören die kombinatorischen Optimierungsprobleme – eine sehr rasche Entwicklung dieses Gebiet wurde erst nach dem 2. Weltkrieg durch Informatik ermöglicht und obwohl die Informatik selbst in Übermengen die Resultate dieser Wissenschaft ausnutzt, gehören die kombinatorischen Optimierungen auch mit einem gleich großen Teil zur Mathematik.

Eine gute im Gleichgewicht stehende Arbeit in solchen Grenzgebieten zu schreiben ist oft schwer, da genügend Übersicht in beiden Richtungen nötig ist; an der Grenze zur Mathematik könnte einerseits das Problem entstehen, dass wir die Mathematik zu „angewandt“ ansehen würden, sie im Dienst der Informatik als untergeordnet betrachten würden und dabei den mathematischen Hintergrund nicht richtig verstehen würden, andererseits gibt es hier die Gefahr, Informatik nur als Rechenwerkzeug, oder Rechenhilfe zu benutzen, ohne im Geringsten an die Brauchbarkeit zu denken.

Und genau eine gute Arbeit an dieser Grenze war mein Ziel. Natürlich war es nicht möglich, das Thema völlig zu erschöpfen und ich musste die schwere Aufgabe lösen, aus der großen Auswahl von Möglichkeiten, die dieses Thema anbietet, nur diejenigen Bruchteile zu wählen, die eine geschlossene Arbeit bilden würden und die an der schon erwähnten Grenze bleiben würden. Viele Teilgebiete konnte ich nicht einmal berühren und viele nur zum Teil ausarbeiten. Da dieses konkrete Problem noch kaum untersucht worden ist, hatte ich dabei eine relativ freie Hand. Ich habe zuerst versucht, einige problemspezifische Eigenschaften zu erläutern, dann einige eher theoretische Ergebnisse – vor allem die Knotengrad-Schranke und die mathematische Analyse der randomisierten Heuristiken – zu präsentieren und letztendlich habe ich einige Algorithmen vorstellen können, die vor allem praktisch anwendbar sein sollen. Um Vergleiche der vorgestellten Methoden untereinander und auch mit anderen Methoden zu ermöglichen, habe ich eine repräsentative Menge von Instanzen erstellt, an denen ich die entwickelten Algorithmen getestet habe.

Möge die Arbeit also sowohl zum theoretisch-mathematischen Wissen beigetragen haben, als auch für die Informatik praktische Algorithmen vorgestellt haben, dann hat sie ihre Aufgabe erfüllt.

Danksagung

An dieser Stelle möchte ich noch vier Personen meinen speziellen Dank aussprechen: In erster Reihe ist es meine Begutachterin, Frau Ao.Univ.-Prof. Dipl.-Ing. Dr.techn. Eranda Dragoti-Cela, die mir dieses Thema vorgeschlagen hat und die mit ihren Vorschlägen und Korrekturen zu der Arbeit wesentlich beigetragen hat. Sie hat mir auch durch ihre hervorragende Vorlesung Kombinatorische Optimierung 2 zu dem Basiswissen auf dem Gebiet der NP-schweren Probleme ganz wesentlich geholfen.

Weiter möchte ich meinen Dank Ao.Univ.-Prof. Dipl.-Ing. Dr.techn. Clemens Heuberger aussprechen, der ermöglicht hat, dass ich alle numerischen Tests an dem Rechner des Instituts für Optimierung und Diskrete Mathematik (Math B) der TU Graz machen durfte. Weiter möchte ich Dipl.-Ing. Dr.techn. Univ.-Doz. Johannes Hatzl nennen, der mir durch seine Lehrveranstaltung Kombinatorische Optimierung 1 die Türen zu diesem Gebiet öffnete und der auch mein Interesse an diesem Teil des mathematisch-informatischen Grenzgebiets weckte.

Bedanken möchte ich mich auch bei Professor Jordi Petit i Silvestre, der mir seine Testinstanzen und Programme aus seiner Arbeit [Pet03] zur Verfügung gestellt hat. Bei ihm bedanke ich mich auch für die Inspiration, die dann zur Knotengrad-Schranke geführt hat, und die er in [Pet98] auf Linear-Arrangement-Problem (LAP) angewendet hat.

Zuletzt möchte ich gerne meinen Dank ebenfalls allen denjenigen ausdrücken, die bewusst oder unbewusst dazu beigetragen haben, dass diese Arbeit entstehen konnte.

Kurzfassung

Diese Arbeit beschäftigt sich mit dem *Data-Arrangement-Problem in vollständigen Bäumen einer gegebenen Ordnung* (DAPBaum). Das Ziel des DAPBaum ist es, die Knoten eines gegebenen ungerichteten (ungewichteten) Graphen G in den Blättern eines vollständigen Baumes T einer gegebenen Ordnung d so einzubetten, dass die Summe der Abstände zwischen je zwei Blättern von T , die einer Kante in G entsprechen, minimiert wird. Dieses Problem ist ähnlich wie das *Linear-Arrangement-Problems* (LAP) ein Spezialfall des gut untersuchten *Graphen-Einbettungs-Problems* (GEP), das stets NP-schwer ist.

Das Ziel dieser Arbeit ist es, einige *Heuristiken* für dieses Problem einzuführen, deren Performance anhand einer *Klasse von Testinstanzen* zu vergleichen, und somit eine erste Grundlage für weitere Untersuchungen dieses Problems zu schaffen.

Nach einer kurzen *Einführung in die Problematik* werden zuerst einige *problemspezifische Eigenschaften* erläutert. Danach wird eine *untere Schranke* definiert, die als Verallgemeinerung einer ähnlichen unteren Schranke für das LAP gesehen werden kann.

In weiterer Folge werden einige *Heuristiken* für das DAPBaum präsentiert und deren *Performance bei Zufallsgraphen* untersucht. Im Rahmen dieser Untersuchungen wird auch der *Erwartungswert* und die *Varianz* des Zielfunktionswertes eines zufälligen Arrangements ermittelt. Danach werden *Greedy-Heuristiken* und Verfahren, die die *lokale Suche* als Hauptidee benutzen, präsentiert und getestet.

Letztendlich werden einige *polynomiell lösbare Spezialfälle* behandelt.

Alle Methoden werden anhand mehrerer *Beispiele* veranschaulicht. Zu der Arbeit gehört auch eine *Menge von Testinstanzen*, die gut vergleichen lässt, für welche Instanzen welches Lösungsverfahren besser geeignet ist. Diese Performance-Analyse soll auch die eventuelle zukünftige praktische Anwendung der Resultate erleichtern.

Schlagwörter: Einbettung; Einbettungsproblem; Arrangement; Heuristiken; Greedy; Lokale Suche; Kombinatorische Optimierung; Zufallsgraph.

Abstract

This master thesis deals with the *data arrangement problem on complete d -ary trees* (*DAPTree*). The goal of the *DAPTree* is the embedding of the nodes of a given undirected (unweighted) graph G onto the leaves of a given complete d -ary tree T , so as to minimize the overall sum of the distances between any two leaves of T which correspond to an edge of G . Similarly as the *linear arrangement problem* (*LAP*) the *DAPBaum* is an *NP-hard* special case of the well investigated *graph embedding problem* (*GEP*).

In this thesis we present some *heuristics* for the *DAPTree* and analyse their performance over a *set of test instances*, while hoping that these investigations could serve as a first basis for further research work on the *DAPTree*.

In the first chapter we *state the problem* and describe some *problem specific properties*. Then we introduce a *lower bound* which is a generalization of an already known lower bound for the *LAP*.

In the next chapter we introduce some *heuristics* for the *DAPTree* and test their *performance on a class of random graphs*. As an important result of this thesis we give a closed formula for the *expected value* and the *variance* of the objective function value of a random arrangement over a special class of graphs. Further we present some *Greedy-like heuristics* and some *local search* heuristics.

At the end we focus on some *polynomially solvable special cases* of the problem.

All heuristic approaches are illustrated by *examples*. We have also generated a *set of test instances* which allows us to compare the performance of the proposed heuristics on different types of test instances. This kind of analysis could be relevant in case of a practical application of the presented heuristics.

Keywords: Embedding; Embedding problem; Arrangement; Heuristic; Greedy; Local search; Combinatorial optimization; Random graph.

1 Einführung

1.1 Problemstellung

Wenn wir über das Data-Arrangement-Problem sprechen, denken wir an die Klasse der sogenannten kombinatorischen Optimierungsprobleme. Informell gesagt soll man aus einer endlichen Menge der zulässigen Lösungen eine gewissen vordefinierten Kriterien entsprechende optimale Lösung auswählen. Nur für eine bestimmte Klasse solcher Probleme gibt es einen effizienten Algorithmus (sofern $P \neq NP$). Für die anderen Probleme, zu denen auch das Data-Arrangement-Problem gehört, sucht man mindestens eine „möglichst gute“ Lösung, die man „schnell“ berechnen kann und die eine garantierte Güte hat. Es gibt aber auch Probleme, für die man weder eine exakte, noch eine entsprechende approximative Lösung finden kann. Dann setzt man Heuristiken an, die uns eine Lösung liefern, deren Güte aber nur empirisch bestimmbar ist. Um sie abschätzen zu können, muss man dazu repräsentative optimale Lösungen haben, oder wie es meistens der Fall ist, eine gute Schranke bestimmen können, die uns eine sinnvolle Information über den Unterschied zwischen den jeweiligen Zielfunktionswerten geben würde. Und gerade eine solche Untersuchung von Heuristiken und Schranken ist Ziel der folgenden Arbeit.

1.1.1 Grundlegende Definitionen und Sätze

Definition 1.1 (Layout). *Gegeben sei ein ungerichteter Graph $G = (V, E)$ mit Knotenmenge $|V|$, Kantenmenge $|E|$ und mit $|V| = n$. Ein **Layout** von G ist eine injektive Funktion $\varphi : V \rightarrow \{1, \dots, n\}$, die die Knoten von G in ein äquidistantes Gitter verteilt.*

Im *Minimum-Linear-Arrangement-Problem (MinLAP)* versucht man ein bzgl. der Funktion $\sum_{(u,v) \in E} |\varphi(u) - \varphi(v)|$ optimales Layout zu finden.

MINIMUM-LINEAR-ARRANGEMENT-PROBLEM (MINLAP)

Gegeben: Ungerichteter Graph $G = (V, E)$.

Gesucht: Ein Layout, das

$$LA(G, \varphi) = \sum_{(u,v) \in E} |\varphi(u) - \varphi(v)| \text{ minimiert.}$$

Mit Worten beschrieben, suchen wir eine Einbettung der Knoten des Graphen in ein eindimensionales äquidistantes Gitter, sodass die Gesamtsumme der Abstände über jene Gitterpunkte, die den Kanten des ursprünglichen Graphen entsprechen, minimiert wird.

Als Veranschaulichung betrachten wir folgendes Beispiel:

Beispiel 1.1. *Gegeben sei folgender Graph:*

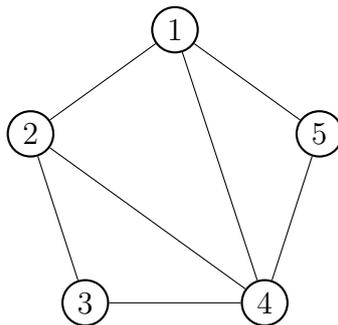


Abbildung 1.1: *Graph*

Eine Möglichkeit, wie man die Knoten 1 – 5 einbetten könnte, ist:

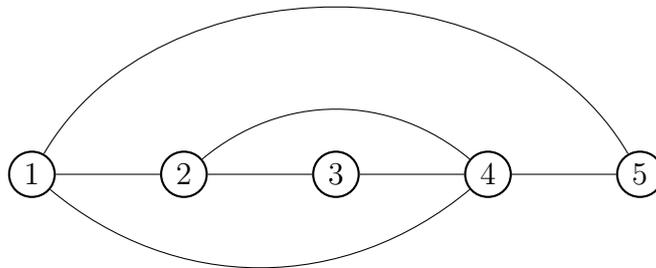


Abbildung 1.2: *Eine lineare Anordnung der Knoten*

Der Zielfunktionswert ist dann aus Tabelle 1.1 ablesbar:

Kante (u, v)	Summand $ \varphi(u) - \varphi(v) $
(1, 2)	1
(2, 3)	1
(3, 4)	1
(4, 5)	1
(5, 1)	4
(1, 4)	3
(2, 4)	2
Insgesamt:	13

Tabelle 1.1: Zielfunktionswert zur Anordnung bei Abb. 1.2

Eine Optimallösung können wir in Abb. A.1 im Anhang sehen.

Das *MinLAP* ist ein intensiv untersuchtes Problem, siehe z.B. die Arbeit von CAPRARA ET AL [CLSG10] und die Arbeiten von PETIT [Pet98] und [Pet03], um einige Literaturquellen zu nennen, insbesondere solche, die sich intensiv mit Heuristiken befassen.

Wenn die Knoten des Graphen in den Blättern eines Baumes der Ordnung d eingebettet werden, so enthält man das *Minimale Data-Arrangement-Problem in einem Baum* (*MinDAPBaum*).

Definition 1.2 (Arrangement-Baum, Arrangement). Gegeben sei ein ungerichteter Graph $G = (V, E)$ mit Knotenmenge $|V|$, Kantenmenge $|E|$ und mit $|V| = n$. Sei weiter $d \in \mathbb{N}$, $2 \leq d \leq n$. Ein vollständiger Baum T der Ordnung d und der Höhe $\lceil \log_d n \rceil$ heißt **Arrangement-Baum**. Ein **Arrangement** von G auf T ist eine injektive Funktion $\phi : V \rightarrow B$, wobei die Menge B aus Blättern des Arrangement-Baumes T besteht.

MINIMUM-DATA-ARRANGEMENT-PROBLEM IN EINEM BAUM (MINDAPBAUM)

Gegeben: Ungerichteter Graph $G = (V, E)$ mit $|V| = n$ und $d \in \mathbb{N}, d \geq 2$.

Gesucht: Ein Arrangement ϕ , das minimal bezüglich

$$\sum_{(u,v) \in E} d_T(\phi(u), \phi(v)) \text{ ist,}$$

wobei $d_T(b_i, b_j)$ die Anzahl der Kanten auf dem Weg zwischen Blättern b_i und b_j im Arrangement-Baum T der Ordnung d ist.

Das Problem wird im folgenden Beispiel veranschaulicht:

Beispiel 1.2. Wenn wir wieder den Graphen aus Abb. 1.1 in Beispiel 1.1 betrachten, kann eine Zuordnung für $d = 3$ folgendermaßen aussehen:

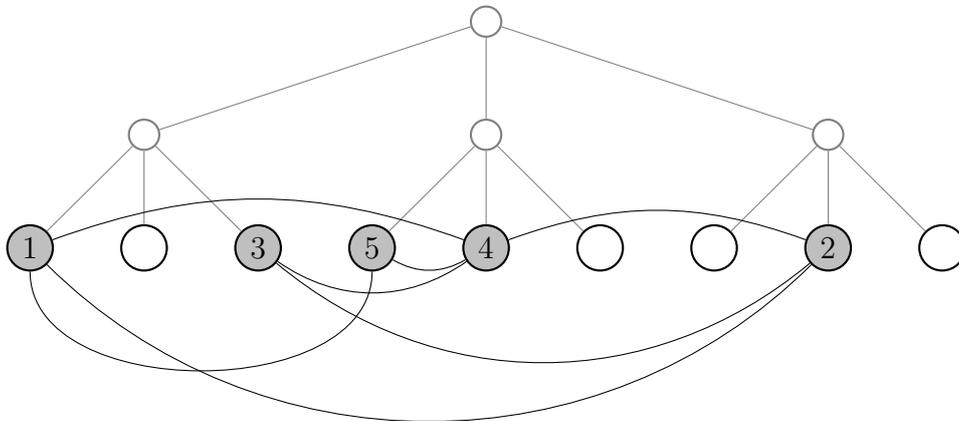


Abbildung 1.3: Ein Arrangement der Knoten

Nun wird der Zielfunktionswert berechnet: Für das Knotenpaar $(1, 2)$ ist die Länge des Weges gleich 4 (siehe Abb. 1.4, in Rot), für die Kante $(4, 5)$ ist die Länge des entsprechenden Weges 2 (Abb. 1.4, blau) usw.

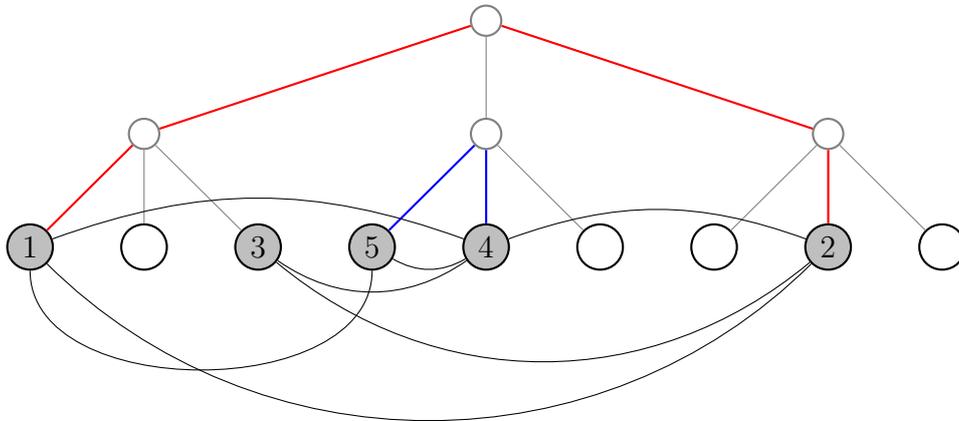


Abbildung 1.4: Ein Arrangement der Knoten – Zielfunktionswert – Berechnung

Zusammengefasst:

Kante (u, v)	Summand $d_T(\phi(u) - \phi(v))$
(1, 2)	4
(2, 3)	4
(3, 4)	4
(4, 5)	2
(5, 1)	4
(1, 4)	4
(2, 4)	4
Insgesamt:	26

Tabelle 1.2: Zielfunktionswert zum Arrangement in Abb. 1.3

Die Optimallösung können wir in Abb. A.2 im Anhang sehen.

Es gibt auch eine allgemeinere Version dieses Problems, in der auch Kantengewichte des Inputgraphen berücksichtigt werden:

DAS GEWICHTETE MINIMUM-DATA-ARRANGEMENT-PROBLEM IN EINEM BAUM (GEWICHTETES MINDAPBAUM)

Gegeben: Ungerichteter Graph $G = (V, E)$ mit $|V| = n$, Gewichte $c : E \rightarrow \mathbb{R}$ und $d \in \mathbb{N}$, $d \geq 2$.

Gesucht: Ein Arrangement ϕ , das minimal bezüglich

$$\sum_{(u,v) \in E} c(u,v) d_T(\phi(u), \phi(v)) \text{ ist,}$$

wobei $d_T(b_i, b_j)$ die Anzahl der Kanten auf dem Weg zwischen Blättern b_i und b_j im Arrangement-Baum T der Ordnung d ist.

Zum Schluss möchten wir noch einen wichtigen Satz erwähnen (siehe auch die Bemerkung über die Komplexität des Problems in Abschnitt 1.1):

Satz 1.1. *Das MinDAPBaum ist NP-vollständig.*

Beweis: Siehe LUCZAK und NOBLE [LN02]. □

1.2 Praktische Anwendungen

Obwohl sich diese Arbeit mit den praktischen Anwendungen ihrer Resultate im Prinzip nicht beschäftigt, möchten wir an dieser Stelle mindestens die wichtigsten Anwendungen erwähnen:

- Da das *MinDAPBaum* ein Spezialfall des *Graph-Einbettung-Problems* (*GEP*) ist, finden wir Anwendungen im *VLSI-design* (*VLSI* steht für „*Very-large-scale integration*“) [MT98]:

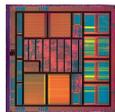


Abbildung 1.5: *VLSI* (Abbildungsquelle: [Bis11])

- Eine weitere wichtige Anwendung findet dieses Problem bei der Data-Verteilung an Speicherplätze, die durch eine Struktur in der Form eines verteilten Folders verbunden werden [LN02].
- Weiters möchten wir anmerken, dass *MinDAPBaum* eng mit anderen relevanten *NP-schweren* Problemen zusammenhängt, z.B. mit dem *minimalen Schnitt-Problem in beschränkten Mengen (MinSPBM)* – siehe auch Abschnitt 4.2.3. Somit sind Ergebnisse und Anwendungen für das *MinDAPBaum* auch auf andere Probleme transferierbar und umgekehrt.

2 Weitere Definitionen und Bemerkungen

In diesem Kapitel führen wir einige Definitionen und Notationen zu *Arrangement-Bäumen* und zum *DAPBaum* an.

Sei also $G = (V, E)$ ein **ungerichteter Graph**, wobei die Knotenmenge $V := V(G)$ und die Kantenmenge $E := E(G)$ ist. Die **Anzahl der Knoten** $|V|$ werden wir mit n und die **Anzahl der Kanten** $|E|$ mit m bezeichnen. Die Menge der **Nachbarn** des Knotens $v \in V(G)$ wird als $\Gamma(v)$ geschrieben, wobei $\Gamma(v) = \{w \in V(G) : (v, w) \in E(G)\}$ gilt. O.B.d.A. können wir annehmen, dass der Graph G *schleifenfrei* ist. Auch wenn wir vom *vollständigen Graphen* sprechen, meinen wir, dass er *schleifenfrei* ist. Ein vollständiger Graph mit n Knoten wird mit K_n bezeichnet.

Sei weiter T ein Arrangement-Baum (T vollständig) der Ordnung $d \geq 2$ und der Höhe $h = \lceil \log_d n \rceil$. Die **Wurzel** werden wir mit r bezeichnen. Eine Menge $V_i \subseteq V$ aller Knoten v mit der Eigenschaft, dass die Anzahl der Kanten auf dem $v - r - \text{Weg}$ gleich i ist, heißt **i-te Ebene des Arrangement-Baumes** T . Für die Anzahl der Blätter ($|V_h|$), die wir als b bezeichnen werden, gilt dann:

$$b = d^h. \tag{1}$$

Wenn wir sagen, dass das *Arrangement-Baum* **indexiert** ist, heißt das, dass die Blätter einen Index zugewiesen bekommen, sodass die Indizes von links nach rechts beginnend mit 1 aufsteigend sortiert sind (siehe Abb. 2.1, in Rot).

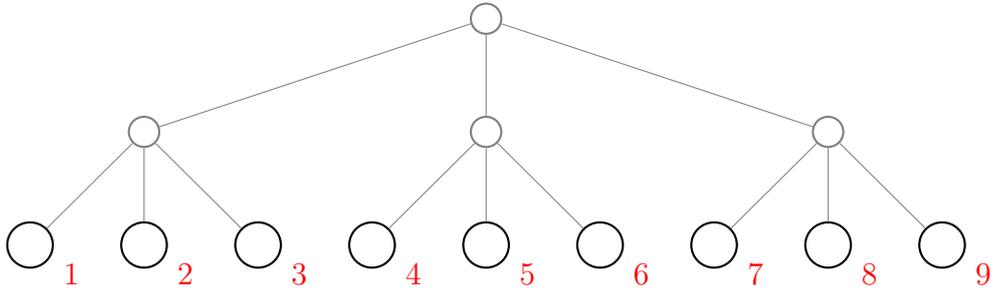


Abbildung 2.1: Arrangement-Baum – Skizze mit Blätter-Indexen

Wir bezeichnen den **Zielfunktionswert** des *MinDAPBaum* mit Input G , d und Arrangement ϕ (siehe genaue Problemstellung in Abschnitt 1.1.1) als $z(G, d, \phi)$.

Formulieren wir nun eine allgemein gültige Bemerkung:

Bemerkung 2.1. Sei $G = (V, E)$ ein Graph und $d \in \mathbb{N}$, $d \geq 2$ die Ordnung eines Arrangement-Baumes T . Dann gibt es $\frac{b!}{(b-n)!}$ unterschiedliche Arrangements der Knoten von G auf T , wobei b die Anzahl der Blätter von T ist. Wir werden

$$c := \frac{b!}{(b-n)!} \quad (2)$$

schreiben.

Die genaue Anzahl hängt von der Ordnung des Arrangement-Baumes ab. Da $d \leq n$ gilt, können wir klarerweise auch $n \leq b \leq n^2$ schreiben, woraus folgt:

$$n! \leq c < (n^2)! \quad (3)$$

Im besten Fall (nach der Wahl von d) wächst die Anzahl der Arrangements also mit der *Fakultät von der Knotenanzahl*; komplette Enumeration ist also schon für ziemlich kleine Instanzen unmöglich ($\Omega(n!)$, $O(n^2!)$). Wie wir aber schon in Abschnitt 1.1 erwähnt haben, ist es nützlich, eine Optimallösung für mindestens einige Instanzen zu haben, um die Güte der Heuristiken und der unteren Schranken abschätzen zu können. Zu diesem Zweck werden in Kapitel 5 einige Spezialfälle behandelt. Um aber allgemeinere Vergleichsdaten zu bekommen, wurde eine Menge der vollständig enumerierbaren Instanzen inklusive der Optimallösungen erstellt (unter *vollständig enumerierbar* verstehen wir solche Instanzen, die in einem heute üblichen Rechner in höchstens 48 Stunden durch vollständige Enumeration gelöst werden können). Sie sind in Tabelle A.1 zusammengefasst. Der Graph *KE_Muster*

entspricht dem Graphen aus Abb. 1.1 in Beispiel 1.1. Die Nummern im Namen deuten die Knotenanzahl im Graphen. Um weitere Informationen über gewählte Testgraphen zu erfahren, siehe die ausführlichere Beschreibung in Abschnitt 6.1. Alle numerischen Ergebnisse sind dann im Abschnitt 6.2 und in Tabelle A.2 zusammengefasst.

Zum Schluss dieses Kapitels möchten wir noch zwei interessante Bemerkungen erwähnen.

Definition 2.1 (Zusammenhängendes Arrangement, Zusammenhängende Optimallösung). Sei $G = (V, E)$ ein Graph, $d \in \mathbb{N}$, $d \geq 2$ die Ordnung des Arrangement-Baumes T , ϕ ein Arrangement von G auf T und T ein indexierter Arrangement-Baum für G der Ordnung d ($d \in \mathbb{N}$, $d \geq 2$). Das Arrangement ϕ heißt **zusammenhängend**, wenn es eine natürliche Zahl $k \in \mathbb{N}$ gibt, sodass es für alle Blätter b_i , $k \leq i \leq k + n$ einen Knoten v_j mit $\phi(v_j) = b_i$ gibt. Ein zusammenhängendes optimales Arrangement heißt eine **zusammenhängende Optimallösung**.

Und nun die Bemerkung:

Bemerkung 2.2. Es gibt Instanzen des MinDAPBaum, die keine zusammenhängende Optimallösung besitzen.

Beweis: Wir betrachten einen sogenannten erweiterten Stern (für die Definition von einem erweiterten Stern siehe in Abschnitt 5.5), der dem Graphen aus der Testinstanz *SF_erwStern* (siehe Tabelle A.2 und die ausführlichere Beschreibung der Testinstanzen in Abschnitt 6.1) entspricht:

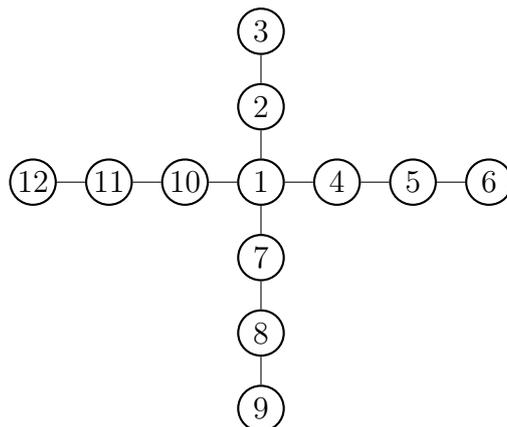


Abbildung 2.2: Ein erweiterter Stern

Für $d = 4$ sehen wir eine Optimallösung in Abb. 2.3.

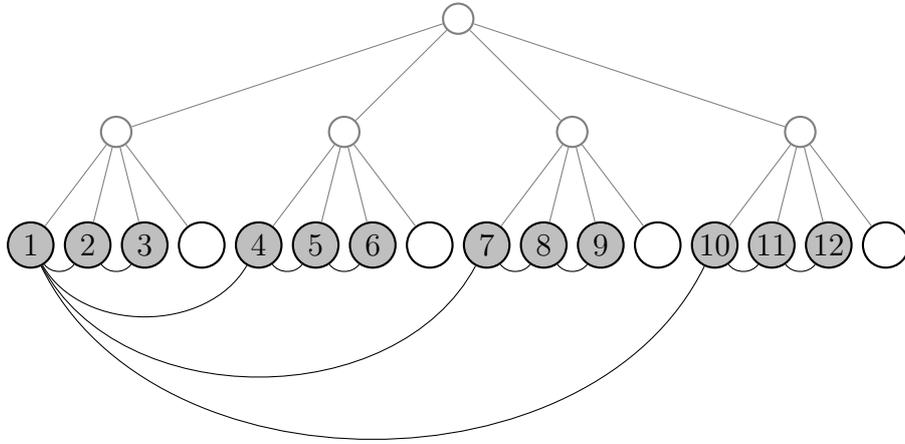


Abbildung 2.3: *Optimallösung für $d = 4$*

Es ist klar, dass diese Lösung optimal ist (siehe den Spezialfall *Erweiterte Sterne* in Abschnitt 5.5). Jeder Sternstrahl $((2, 3), (4, 5, 6), (7, 8, 9)$ und $(10, 11, 12))$ ist an einen Unterbaum angeknüpft und alle diese Teile sind „zu lang“, um zwei davon demselben Unterbaum zuweisen zu können. Um die Optimalität also nicht zu verletzen, darf man das Arrangement nur flippen (siehe Definition 4.4 und Bemerkung 4.2) und man darf auch innerhalb eines Unterbaumes natürlich Änderungen vornehmen. Mit solchen Operationen kann man aber nie eine zusammenhängende Lösung erstellen.

Die erweiterten Sterne werden als eine Klasse von Spezialfällen des *MinDAPBaum* in Abschnitt 5.5 betrachtet. \square

Daraus folgt also, dass man eine Lösung nicht so suchen kann, dass man zunächst ein Intervall von n Blättern fixiert, dann alle bijektiven Abbildungen von V auf dieses Intervall durchsucht und schließlich über alle möglichen Intervalle der Länge n iteriert. Diese Beobachtung wäre in dieser Arbeit nicht so wichtig, wenn sie dem besseren Verständnis der Erklärung der zweiten Bemerkung nicht dienen würde.

Bemerkung 2.3. *Es gibt Instanzen des MinDAPBaum, die folgende Eigenschaften erfüllen*

- (i) G ist unzusammenhängend und besteht aus z Zusammenhangskomponenten G_1, \dots, G_z .
- (ii) *Es gibt kein optimales Arrangement $\phi(G)$, sodass es Indices $0 = i_0 < i_1 < \dots < i_{z-1} < i_z = b$ gibt, für die $\{\phi(v) | v \in V(G_j)\} \subseteq \{b_i | i_{j-1} < i \leq i_j\}$ für alle $1 \leq j \leq z$ gilt und $\phi(G)$ auch bezüglich G_j optimal ist.*

Beweis: Betrachten wir den Graphen aus Abb. 2.4, in dem eine Zusammenhangskomponente in Schwarz und die zweite in Rot gezeichnet ist (es handelt sich wieder um 2 erweiterte Sterne (siehe in Abschnitt 5.5)):

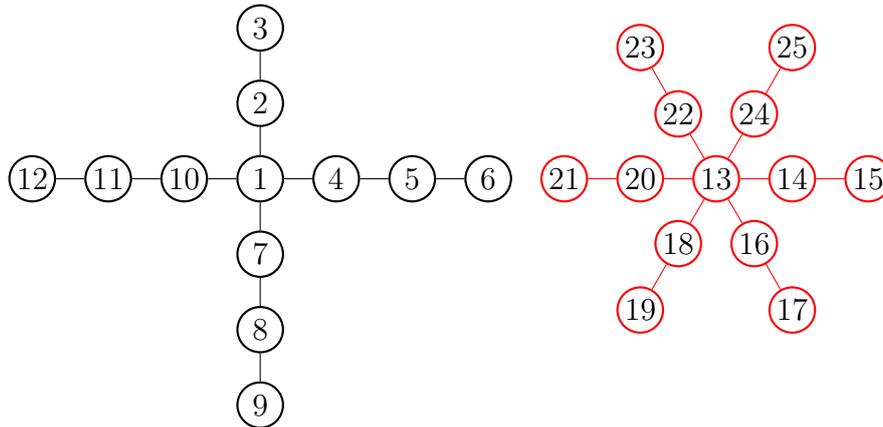


Abbildung 2.4: Graph

Wir betrachten die in Abb. 2.5 dargestellte Lösung für $d = 5$:

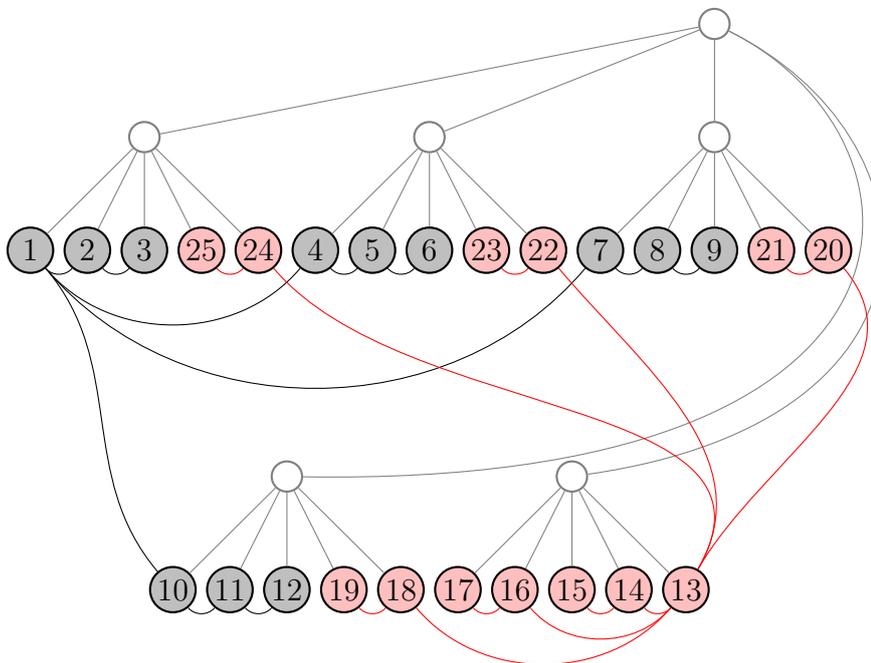


Abbildung 2.5: Optimallösung für $d = 5$

Es ist klar, dass für den optimalen Zielfunktionswert

$$\min_{\phi} z(G, d, \phi) \geq \min_{\phi} z(G_1, d, \phi) + \min_{\phi} z(G_2, d, \phi) \quad (4)$$

gelten muss, wobei G_1 , bzw. G_2 der durch die Knotenmenge der ersten Zusammenhangskomponente, bzw. der zweiten Zusammenhangskomponente induzierter Teilgraph ist und unter der Annahme, dass die Höhen der entsprechenden Arrangement-Bäume gleich sind, d.h. $h_G = h_{G_1} = h_{G_2}$.

Die in Abb. 2.5 gezeigte Lösung erfüllt die Ungleichung (4), die gleichzeitig eine untere Schranke darstellt, mit Gleichheit und ist also optimal (für beide Zusammenhangskomponenten, die erweiterte Sterne sind, ist diese Lösung optimal, siehe in Abschnitt 5.5). Im Beweis zu Bemerkung 2.2 haben wir für die erste Zusammenhangskomponente auch gezeigt, dass sie keine zusammenhängende Lösung besitzt und da die Anzahl der Blätter des Arrangement-Baumes $b = n$, besteht keine Möglichkeit, dass eine optimale Lösung durch Zusammensetzen der Optimallösungen beider Zusammenhangskomponenten entstehen kann (mindestens ein Knoten der zweiten Zusammenhangskomponente muss unter den der ersten Zusammenhangskomponente eingebettet sein).

Somit ist die in Abb. 2.5 dargestellte Lösung optimal. Jede Optimallösung muss ebenfalls den obigen Zielfunktionswert in (4) erreichen, was zwangsweise zu einer Zerstückelung der Intervalle der Indices jener Blätter führt, zu denen die Knoten einer Zusammenhangskomponente zugeordnet werden. D.h. für dieses Beispiel gilt tatsächlich Punkt (ii) aus der Bemerkung. \square

Aus dieser Bemerkung folgt also, dass man auch unzusammenhängende Graphen in den im weiteren Text beschriebenen Methoden betrachten muss.

3 Untere Schranken

Eine der ersten Fragen, die man sich stellen kann, wenn man ein *NP-vollständiges Kombinatorisches Optimierungsproblem* betrachtet, ist: „Wie kann man eine Lösung approximieren? Wie gut mag diese Approximation sein? Und wie kann man den Zielfunktionswert beschränken?“ Auf die erste Frage wird ausführlich in Kapitel 4 eingegangen. Die Zweite folgt dann aus der theoretischen Analyse und oft (wie auch in dieser Arbeit) aus dem Vergleich mit der Optimallösung, bzw. mit den unteren Schranken, denen dieses Kapitel gewidmet ist.

Wie kann man also den Zielfunktionswert beschränken? Im schlimmsten Fall führt der $\phi(u)$ - $\phi(v)$ -Weg für jede Kante (u, v) über die Wurzel. Daraus ergibt sich folgende Bemerkung:

Bemerkung 3.1. Für jeden Graphen $G = (V, E)$ und für jedes Arrangement ϕ gilt:

$$z(G, d, \phi) \leq 2hm, \quad (5)$$

wobei h die Höhe des Arrangement-Baumes für G ist und $m = |E|$.

Genauso muss die Länge des $\phi(u)$ - $\phi(v)$ -Weges für jede Kante (u, v) mindestens 2 sein:

Bemerkung 3.2. Für jeden Graphen $G = (V, E)$ und für jedes Arrangement ϕ gilt:

$$z(G, d, \phi) \geq 2m, \quad (6)$$

wobei h die Höhe des Arrangement-Baumes für G ist und $m = |E|$.

Das Optimum liegt also irgendwo zwischen $2m$ und $2hm$. Nicht viel komplizierter ist die sog. *Knotengrad-Schranke*, die zum ersten Mal prinzipiell von PETIT in [Pet98] für das *MinLAP* eingeführt wurde und die im folgenden Abschnitt auf unser Problem erweitert wird.

3.1 Knotengrad-Schranke

Für diese untere Schranke, die wir als $KGS(G, d)$ bezeichnen werden, nehmen wir an, dass unser (imaginäres) Arrangement ϕ für jeden Knoten v lokal optimal ist. D.h., die Nachbarknoten $\Gamma(v) = \{v_1, \dots, v_{|\Gamma(v)|}\}$ von v sind in unserem Arrangement so plaziert, dass das Arrangement für den Graphen $G'_v = (V'_v, E'_v)$ mit $V'_v = V$ und $E'_v = \{(v, v_1), (v, v_2), \dots, (v, v_{|\Gamma(v)|})\}$ optimal ist.

Beispiel 3.1. Betrachten wir wieder den Graphen aus Abb. 1.1 in Beispiel 1.1 und sei nun $v = 1$. In Abb. 3.1 kann man den Graphen G'_1 in Rot sehen.

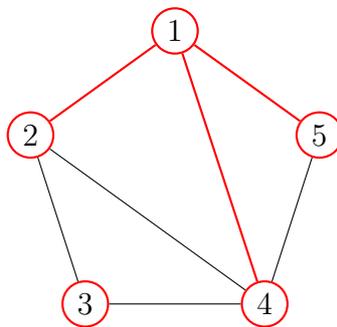


Abbildung 3.1: *Knotengrad-Schranke* – Graph G'_1

Für den Graphen G'_1 und die Ordnung des Arrangement-Baumes $d = 3$ ist dann leicht, das optimale Arrangement ϕ_1 zu finden (siehe auch Abschnitt 5.4):

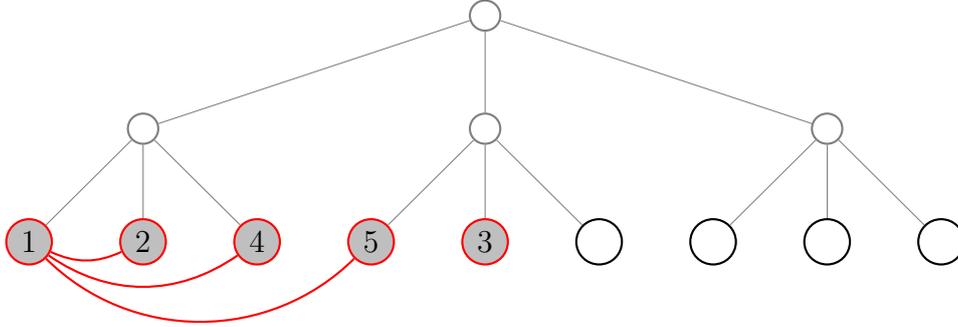


Abbildung 3.2: Knotengrad-Schranke – optimales Arrangement ϕ_1 für den Graphen G'_1

Der Zielfunktionswert dieses lokal optimalen Arrangements ist also 8.

Nun reicht es uns, für alle restlichen Knoten $v \in V$ diesen Wert zu berechnen. Wenn wir den Zielfunktionswert des Optimalen Arrangements ϕ_v für $v \in V$ mit s_v bezeichnen, gilt offensichtlich

$$KGS(G, d) = \frac{1}{2} \sum_{v \in V} s_v \leq z(G, d, \phi). \quad (7)$$

Die Ungleichung 7 lässt sich folgendermaßen begründen: Der Beitrag eines beliebigen Knotens v in den Zielfunktionswert $z(G, d, \phi)$ ist stets nicht kleiner als s_v und jede Kante (v, w) von G wird sowohl in s_v , als auch in s_w berücksichtigt.

Wählen wir also einen Knoten $v \in V$ und bilden das optimale Arrangement ϕ_v für den dazugehörigen Graphen G' . Die Konstruktion von diesem optimalen Arrangement ϕ_v ist ausführlich in Abschnitt 5.4 beschrieben. Den Knoten v muss man an das erste Blatt anknüpfen und die Nachbarn müssen nacheinander den folgenden Blättern zugewiesen werden. Die restlichen Knoten können dann beliebig platziert werden.

Nun teilen wir die Knoten aus $\phi(\Gamma(v)) \cup \{\phi(v)\}$ in Mengen $V_{v,j}$, $j = 0, 1, \dots$ so, dass die Länge des $\phi(v) - \phi(v_i)$ -Weges für alle Knoten in jeder konkreten Menge gleich $2j$ ist, und nehmen zuerst an, dass $\Gamma(v) > d - 1$ gilt.

Sei p der Größte Index der Mengen $V_{v,j}$, sodass $\emptyset \neq V_{v,p} \subseteq \phi(\Gamma(v)) \cup \{\phi(v)\}$ gilt.

Für die Kardinalität der Mengen $V_{v,j}$ gilt $|V_{v,j}| = (d-1)d^{j-1}$ für $1 \leq j < p$, $|V_{v,0}| = 1$ für $j = 0$ und $|V_{v,p}| = r$ für $j = p$, wobei $0 \leq \Gamma(v) \leq (d-1)d^{p-1}$.

Wir unterteilen die Mengen $V_{v,j}$ in 3 Gruppen. Die erste Gruppe besteht aus $V_{v,0}$, die dritte aus $V_{v,p}$ und die zweite Gruppe enthält die restlichen Mengen. Dann gilt

$$|\Gamma(v)| = \sum_{i=1}^{p-1} (d-1)d^{i-1} + r, \quad (8)$$

wobei $\sum_{i=1}^{p-1} (d-1)d^{i-1}$ (bzw. r) die gesamte Kardinalität der Mengen aus der zweiten (bzw. dritten) Gruppe darstellt. Die Parameter p und r müssen dabei folgende Eigenschaften erfüllen:

$$\sum_{i=1}^{p-1} (d-1)d^{i-1} \leq |\Gamma(v)| \quad (9)$$

und

$$r = |\Gamma(v)| - \sum_{i=1}^{p-1} (d-1)d^{i-1}. \quad (10)$$

Aus der Ungleichung (9) folgt (es handelt sich um eine *geometrische Folge*)

$$\sum_{i=1}^{p-1} (d-1)d^{i-1} = d^{p-1} - 1 \leq |\Gamma(v)|. \quad (11)$$

Für p erhalten wir also:

$$p = \lfloor \log_d (|\Gamma(v)| + 1) \rfloor + 1 \quad (12)$$

und für r erhalten wir aus (8) und (11)

$$r = |\Gamma(v)| - d^{p-1} + 1. \quad (13)$$

Aus p und r können wir nun den Wert von s_v berechnen. Dazu brauchen wir aber noch das folgende Lemma:

Lemma 3.1. *Sei $p \in \mathbb{N}$, $p-1 \geq 1$ und $d \in \mathbb{N}$, wobei $d \geq 2$. Dann gilt*

$$\sum_{i=1}^{p-1} d^{i-1} i = \frac{((d-1)p - d)d^{p-1} + 1}{(d-1)^2}. \quad (14)$$

Beweis: Der Beweis folgt mittels Induktion. Für den Fall $p - 1 = 1$ gilt die Aussage trivialerweise. Untersuchen wir nun den Fall p :

$$\begin{aligned} \sum_{i=1}^p d^{i-1}i &= \sum_{i=1}^{p-1} d^{i-1}i + d^{p-1}p \\ &= \frac{((d-1)p - d)d^{p-1} + 1}{(d-1)^2} + d^{p-1}p. \end{aligned} \quad (15)$$

Nach einfachen Umformungen erhalten wir

$$\sum_{i=1}^p d^{i-1}i = \frac{((d-1)(p+1) - d)d^p + 1}{(d-1)^2}. \quad (16)$$

□

Nun können wir, die Gleichung (8) folgend, schreiben:

$$s_v = \sum_{i=1}^{p-1} (d-1)d^{i-1}2i + 2rp. \quad (17)$$

Aus Lemma 3.1 folgt dann (die Bedingungen an p und d sind klarerweise erfüllt)

$$s_v = 2(d-1) \frac{((d-1)p - d)d^{p-1} + 1}{(d-1)^2} + 2rp \quad (18)$$

und eingesetzt aus (13) und nach leichten Umformungen

$$s_v = 2 \left(p(|\Gamma(v)| + 1) - \frac{d^p - 1}{d-1} \right). \quad (19)$$

Am Anfang haben wir angenommen, dass $\Gamma(v) > d - 1$ gilt; untersuchen wir nun den Fall $0 \leq \Gamma(v) \leq d - 1$. In diesem Fall können wir offensichtlich

$$s_v = 2\Gamma(v) \quad (20)$$

schreiben. Da wir aber in diesem Fall aus (12) $p = 1$ bekommen und

$$2 \left(1(|\Gamma(v)| + 1) - \frac{d^1 - 1}{d-1} \right) = 2\Gamma(v) \quad (21)$$

gilt, brauchen wir die Knoten nach dem Knotengrad nicht zu unterscheiden und wenn wir (19) in (7) einsetzen, erhalten wir die Gleichung für die *Knotengrad-Schranke*.

Satz 3.1. Sei $G = (V, E)$ ein ungerichteter Graph und $d \in \mathbb{N}$, $d \geq 2$ die Ordnung des Arrangement-Baumes T . Dann gilt für jedes Arrangement ϕ

$$KGS(G, d) = \sum_{\substack{v \in V \\ \Gamma(v) > d-1}} \left(p(|\Gamma(v)| + 1) - \frac{d^p - 1}{d - 1} \right) + \sum_{\substack{v \in V \\ \Gamma(v) \leq d-1}} 2\Gamma(v) \leq z(G, d, \phi), \quad (22)$$

wobei

$$p = \lfloor \log_d (|\Gamma(v)| + 1) \rfloor + 1. \quad (23)$$

Beweis. Siehe die Herleitung oben. \square

Betrachten wir nochmals das Beispiel 3.1:

Beispiel 3.1 (Fortsetzung). Für $v = 1$ erhalten wir für die Mengen $V_{v,j}$:

$$\begin{aligned} V_{v,0} &= \{1\} \\ V_{v,1} &= \{2, 4\} \\ V_{v,2} &= \{5\} \end{aligned} \quad (24)$$

und für die Koeffizienten p und r

$$\begin{aligned} p &= \lfloor \log_3 3 + 1 \rfloor + 1 = 2, \\ r &= 3 - 3^{2-1} + 1 = 1. \end{aligned} \quad (25)$$

s_1 kann dann mithilfe der Formel (19) berechnet werden

$$s_1 = 2 \left(2(3 + 1) - \frac{3^2 - 1}{3 - 1} \right) = 2(8 - 4) = 8. \quad (26)$$

Analog werden die Werte s_2 , s_3 , s_4 und s_5 berechnet. Zusammengefasst:

v	s_v
1	8
2	8
3	4
4	12
5	4
Insgesamt:	36

Tabelle 3.1: s_v für alle Knoten

Insgesamt erhalten wir dann

$$KGS(G, d) = \frac{1}{2} \sum_{v \in V} s_v = \frac{36}{2} = 18. \quad (27)$$

Der optimale Zielfunktionswert ist $z(G, d, \phi) = 20$ und die Optimallösung finden wir in Abb. A.1.

Die Zeitkomplexität dieser Berechnung lässt sich leicht bestimmen:

Bemerkung 3.3. Sei $G = (V, E)$ ein Graph, $d \in \mathbb{N}$, $d \geq 2$ und seien die Knotengrade aller Knoten $v \in V$ bereits bekannt. Dann kann man die Knotengrad-Schranke $KGS(G, d)$ in $O(n)$ Zeit bestimmen.

4 Heuristiken

Im letzten Kapitel haben wir einige *untere Schranken* betrachtet. Nun wollen wir die *Heuristiken* näher untersuchen. Sie liefern uns einerseits eine mehr oder wenig gute Lösung, andererseits können die Zielfunktionswerte dieser Lösungen auch als *obere Schranken* betrachtet werden. Zuerst werden *Randomisierte Heuristiken* (Abschnitt 4.1), dann *Greedy- und andere einfache Heuristiken* (Abschnitt 4.2) und später *lokale Suche* (Abschnitt 4.3) näher eingegangen.

4.1 Randomisierte Heuristiken

4.1.1 Normales-Arrangement-Verfahren

Die ganz erste Idee, die man bei einer Optimierungsaufgabe bekommen kann ist, eine beliebige Lösung zu wählen und hoffen, sie möge ausreichend gut sein. Schauen wir nun ein Verfahren, das zwar nicht randomisiert ist, das aber im weiteren Text öfter genutzt wird, näher an.

Definition 4.1 (Normales Arrangement). Gegeben sei ein ungerichteter Graph $G = (V, E)$ mit $|V| = n$ und $d \in \mathbb{N}$, $d \geq 2$ die Ordnung des Arrangement-Baumes T , wobei $V = \{v_1, v_2, \dots, v_n\}$ (die Knoten sind bereits sortiert). Sei weiters T ein indexierter Arrangement-Baum für G mit der Blättermenge $B = \{b_1, b_2, \dots, b_b\}$. Ein Arrangement $\phi : V \rightarrow B$ heißt **normales Arrangement**, wenn $\phi(v_i) = b_i$ gilt.

Wir können natürlich erwarten, dass das *normale Arrangement* einen besseren Zielfunktionswert besitzen wird, als ein ganz zufälliges, da vor allem

bei den der Praxis entnommenen Aufgaben die Knoten oft schon „der Graphentopologie gemäß“ vorsortiert sind (d.h., dass z.B. für jede Zusammenhangskomponente die Knotennummern ein Intervall bilden).

4.1.2 Methode der zufälligen Arrangements

Die Hauptidee ist, k zufällige Arrangements zu erstellen und das beste davon auszuwählen. Auf den ersten Blick könnte diese Methode als schlecht eingestuft werden, es gibt aber Graphen, für die sie gut funktioniert. Schauen wir zuerst an einem Beispiel, wie die Zielfunktionswerte aller Lösungen verteilt sind.

Beispiel 4.1. *Betrachten wir wieder den Graphen aus Abb. 1.1 in Beispiel 1.1 und des Arrangement-Baume mit $d = 3$ (wie in Beispiel 1.2). Dann gilt $h = \lceil \log_d n \rceil = \lceil \log_3 5 \rceil = 2$.*

Die Optimallösung besitzt den Zielfunktionswert 20. Wie viele Arrangements liefern diesen Zielfunktionswert? Und wie sieht die Verteilung der möglichen Zielfunktionswerte des Problems aus? Diese Information wird aus Tabelle 4.1 ablesbar:

Zielfunktionswert	Arrangements	Prozentueller Anteil
20	432	2,857143 %
22	1836	12,142857 %
24	7020	46,428571 %
26	5184	34,285714 %
28	648	4,285714 %
Insgesamt Arrangements:		15120

Tabelle 4.1: Zielfunktionswerteverteilung

Wenn wir also z.B. für dieses Beispiel 100 zufällige Arrangements erstellen und das beste daraus wählen, ist die Wahrscheinlichkeit, dass ein optimales gefunden sein wird, gleich $1 - (1 - 0,02857143)^{100} \approx 94,49\%$.

Allgemeiner kann die Entwicklung dieser Wahrscheinlichkeit in Abhängigkeit von k in Tabelle 4.2 beobachtet werden. Die Wahrscheinlichkeit, dass ein zufällig gewähltes Arrangement optimal ist, beträgt $\frac{432}{15120} = \frac{1}{35}$.

k	$B(k, \frac{1}{35})$	k	$B(k, \frac{1}{35})$
1	2,857143 %	60	82,43483 %
5	13,49238 %	80	90,16284 %
10	25,16432 %	100	94,49082 %
20	43,99621 %	200	99,69649 %
40	68,63575 %		

Tabelle 4.2: *Wahrscheinlichkeit, dass für dieses Beispiel ein Optimum mithilfe der Methode der zufälligen Arrangements getroffen wird*

Im letzten Beispiel haben wir 100 Arrangements gebraucht, um eine optimale Lösung mit der Wahrscheinlichkeit 94,49 % zu finden (wobei wir aber natürlich nicht feststellen können, ob wir sie tatsächlich gefunden haben). Angenommen, es werden zufällig immer unterschiedliche Arrangements erstellt, dann brauchten wir ungefähr $\frac{1}{150}$ aller Möglichkeiten zu enumerieren.

Wie aber verhält sich diese Methode für größere Instanzen? Und für welche kann man bessere und für welche schlechtere Ergebnisse erwarten? Dazu schauen wir die Abbildungen 4.1 und 4.2 an. Die Werte auf der x -Achse entsprechen den Wahrscheinlichkeiten, dass zwei Knoten in einem Graphen verbunden werden – es wurden für jeden diesen Wert $x \in \{0,00; 0,01; \dots, 1,00\}$ 100 zufällige Graphen erstellt (in denen je zwei beliebige Knoten mit der Wahrscheinlichkeit x mit einer Kante verbunden wird). Auf der y -Achse wird dann prozentuell der durchschnittliche Anteil optimaler Lösungen an der Lösungsmenge in diesen Graphen gezeigt. Um solche Tests aber besser beschreiben zu können, definieren wir zuerst einen **Zufallsgraphen**, den in dieser Form zuerst ERDŐS und RÉNYI in [ER59] definiert haben.

Definition 4.2 (Zufallsgraph). *Gegeben sei eine Knotenanzahl $n \in \mathbb{N}$ und eine Wahrscheinlichkeit $p \in \mathbb{R}$, wobei $0 \leq p \leq 1$. Ein Graph n Knoten, in dem je zwei Knoten mit der Wahrscheinlichkeit p durch eine Kante verbunden sind, heißt **Zufallsgraph**. Wir werden ihn $G(n, p)$ bezeichnen.*

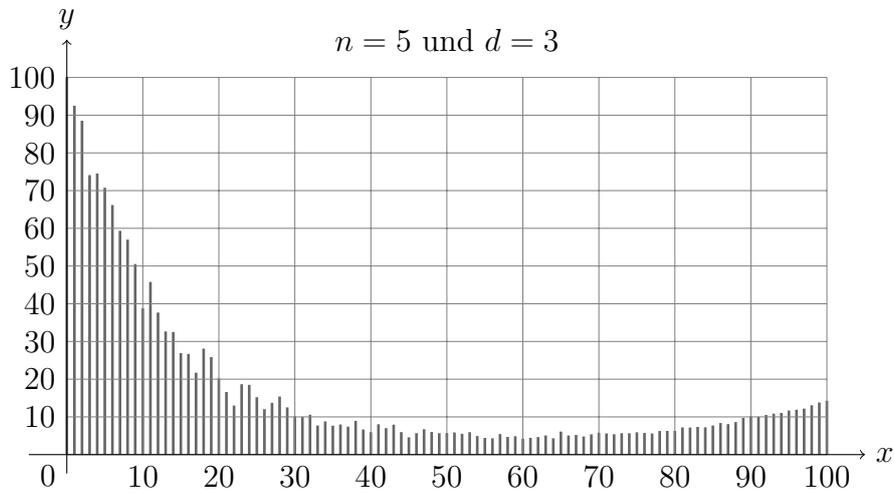


Abbildung 4.1: *Empirischer prozentueller Anteil der optimalen Lösungen für $n = 5$ und $d = 3$ auf $G(5, p)$ mit $p = \frac{x}{100}$, wobei $x \in \mathbb{N}$ und $0 \leq x \leq 100$; es wurden 100 Tests für jede Dichte durchgeführt*

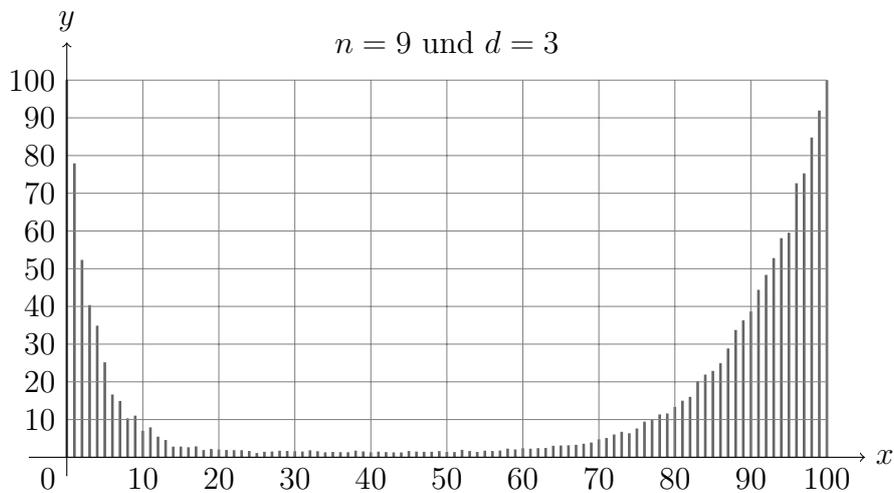


Abbildung 4.2: *Empirischer prozentueller Anteil der optimalen Lösungen für $n = 9$ und $d = 3$ auf $G(5, p)$ mit $p = \frac{x}{100}$, wobei $x \in \mathbb{N}$ und $0 \leq x \leq 100$; es wurden 100 Tests für jede Dichte durchgeführt*

Ohne genaue mathematische Beweise durchzuführen, können wir 2 Vermutungen beobachten – der Anteil der optimalen Lösungen hängt stark:

1. von der Dichte (siehe auch die Definition 4.3) und

2. von der Wahl der Ordnung des Arrangement-Baumes ab.

Wenn ein Graph keine Kanten besitzt, sind klarerweise alle Arrangements optimal. Folgende triviale Bemerkung erklärt uns auch, warum in Abb. 4.2 der Anteil für vollständige Graphen 100 % ist.

Bemerkung 4.1. Sei $G = (V, E)$ ein ungerichteter vollständiger Graph mit $|V| = n$ und $d \in \mathbb{N}$, $d \geq 2$ die Ordnung des Arrangement-Baumes. Sei weiters b Anzahl der Blätter eines Arrangement-Baumes T für G . Wenn $n = b$, dann ist jedes Arrangement ϕ von G auf T optimal.

Wenn ein Graph nahezu vollständig ist und die anderen Annahmen Bemerkung 4.1 gelten, gibt es in dem Graphen viele Symmetrien, die die steigende Tendenz des Anteils der optimalen Lösungen mit wachsender Graphendichte erklärt. An der anderen Seite kann man bei ganz dünnen Graphen viele Zusammenhangskomponenten vermuten, die einerseits symmetrisch in sich und andererseits symmetrisch gegenüber einander sein können.

In Tabelle A.3 gibt es in Grau weitere Untersuchungen dieser Art in der Klasse $G(n, p)$ für verschiedene Werte von n und verschiedene Ordnungen der Arrangement-Bäume.

Zusammengefasst eignet sich diese Methode gut für ganz dünne, oder unter den Bedingungen der Bemerkung 4.1 für ganz dichte Graphen. Trotzdem sind die Ergebnisse in Tabelle A.2 überraschend schlecht vor allem im Vergleich zum *Normalen-Arrangement-Verfahren* (Abschnitt 4.1.1) – extrem z.B. für die Graphen *Pet03_airfoil1* und *Pet03_big*). Diese Beobachtung führt uns zur nächsten Heuristik.

4.1.3 Methode der zufälligen zusammenhängenden Arrangements

Wie wir aus den Ergebnissen in Tabelle A.2 beobachten können, sind die Zielfunktionswerte des *Normalen-Arrangement-Verfahrens* deutlich besser als diejenigen, die aus der *Methode der zufälligen Arrangements* stammen. Dieses Verhalten kann uns zur Vermutung führen, dass es empirisch gesehen besser wäre, nur *zusammenhängende zufällige Arrangements* zu generieren und das Risiko bewusst in Kauf nehmen, dass eine Instanz überhaupt keine zusammenhängende Lösung besitzen muss. Wir haben zwar gezeigt, dass man solche Instanzen konstruieren kann (siehe Bemerkung 2.2), in Praxis treten sie aber recht selten vor.

Um diese Methode bewerten zu können, schauen wir die Abb. 4.3 an. Sie ist ähnlich wie diejenigen Graphen, die bei der *Methode der zufälligen Arrangements* besprochen wurden – in Rot wurden die Anteile der optimalen zusammenhängenden Lösungen an der Menge aller zusammenhängenden

Lösungen gezeichnet. Sollte eine Instanz keine solche Lösung besitzen, wurde der Anteil 0 % angerechnet.

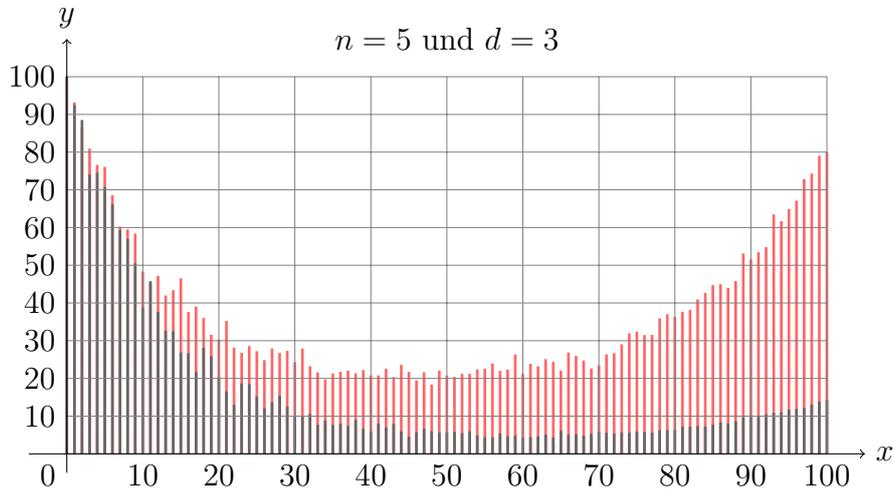


Abbildung 4.3: Empirischer prozentueller Anteil der optimalen (in Grau) bzw. der optimalen zusammenhängenden (in Rot) Lösungen für $n = 5$ und $d = 3$ auf $G(5, p)$ mit $p = \frac{x}{100}$, wobei $x \in \mathbb{N}$ und $0 \leq x \leq 100$; es wurden 100 Tests für jede Dichte durchgeführt

Empirisch gesehen, ist die Wahrscheinlichkeit, eine optimale Lösung mit dieser Methode zufällig zu generieren, viel höher. Vor allem bei dichten Graphen erweist sich diese Methode deutlich besser, da sie weniger von der Ordnung des Arrangement-Baumes abhängt.

Ergebnisse für Zufallsgraphen wurden in Tabelle A.3 in Rot zusammengefasst. (Für den Fall $n = 4$ und $d = 2$ ist keine spezielle Berechnung dieses Verfahrens notwendig, da alle Arrangements zusammenhängend sind.)

4.1.4 Erwartungswert des Zielfunktionswertes eines zufälligen Arrangements

Wenn wir vom *Erwartungswert des Zielfunktionswertes eines zufälligen Arrangements* sprechen, handelt es sich offensichtlich um keine Heuristik, denen sich diese Arbeit widmet. Da uns aber die Abhängigkeit dieses Wertes von der *Dichte des Graphen G* (siehe die Definition 4.3) eine gute Vorstellung vom Zielfunktionswert eines zufälligen Arrangements vermitteln könnte, besteht ein Zusammenhang zur *Methode der zufälligen Arrangements* in Abschnitt 4.1.2.

Da alle Arrangements mit gleicher Wahrscheinlichkeit auftreten, kann der Erwartungswert definitionsgemäß leicht geschrieben werden (siehe auch Bemerkung 2.1):

$$\mathbb{E}z(G, d, \phi) = \sum_{\phi} z(G, d, \phi) \frac{1}{c}. \quad (28)$$

Laut Problemstellung (siehe in Abschnitt 1.1.1) und nach leichter Umformung erhalten wir

$$\mathbb{E}z(G, d, \phi) = \sum_{\phi} \frac{1}{c} \sum_{(u,v) \in E} d_T(\phi(u), \phi(v)), \quad (29)$$

wobei $d_T(b_i, b_j)$ die Anzahl der Kanten auf dem Weg zwischen Blättern b_i und b_j im *Arrangement-Baum* T der *Ordnung* d ist. Nun können wir diesen Ausdruck umformen und für den *Erwartungswert* erhalten wir

$$\mathbb{E}z(G, d, \phi) = \frac{1}{c} \sum_{b_i, b_j \in B} d_T(b_i, b_j) \sum_{(u,v) \in E} \sum_{\substack{\phi: \\ \phi(u)=b_i \text{ und } \phi(v)=b_j}} 1 \quad (30)$$

wobei die Menge B aus Blättern des *Arrangement-Baumes* T besteht.

Um die Abhängigkeit des *Erwartungswertes* von der *Dichte des Graphen* G bestimmen zu können, brauchen wir noch eine Definition, ein Lemma und ein Korollar.

Definition 4.3 (Dichte eines Graphen). Die *Dichte* $d(G)$ eines **Graphen** $G = (V, E)$ sei als

$$d(G) = \begin{cases} 0 & \text{für } |V| = 1 \\ \frac{2|E|}{|V|(|V|-1)} & \text{für } |V| > 1 \end{cases} \quad (31)$$

definiert.

Lemma 4.1. Sei T ein (vollständiger) *Arrangement-Baum* der *Ordnung* $d \in \mathbb{N}$, $d \geq 2$, und der *Höhe* $h \in \mathbb{N}$. Sei weiter $b = d^h$ die Anzahl der Blätter des *Arrangement-Baumes* T . Dann gilt:

$$\sum_{i=1}^b \sum_{j=1}^b d_T(b_i, b_j) = 2b \left((h+1)b - \frac{db-1}{d-1} \right). \quad (32)$$

Beweis: Offensichtlich gilt aus Symmetrie-Gründen

$$\sum_{j=1}^b d_T(b_{i_1}, b_j) = \sum_{j=1}^b d_T(b_{i_2}, b_j) \text{ für alle } 1 \leq i_1 \leq h \text{ und } 1 \leq i_2 \leq h, \quad (33)$$

woraus

$$\sum_{i=1}^b \sum_{j=1}^b d_T(b_i, b_j) = b \sum_{j=1}^b d_T(b_1, b_j) \quad (34)$$

folgt. Nun können wir, ähnlich wie in Abschnitt 3.1, die Blätter b_1, \dots, b_b in Mengen B_j , $j \in \{0, \dots, h\}$, so teilen, dass $B_j = \{b_i \mid d_T(b_1, b_i) = 2j\}$ gilt. Für die Kardinalität dieser Mengen erhalten wir klarerweise $|B_j| = (d-1)d^{j-1}$. Daraus folgt also

$$\begin{aligned} \sum_{j=1}^b d_T(b_1, b_j) &= \sum_{i=1}^h (d-1)d^{i-1}2i \\ &= 2(d-1) \sum_{i=1}^h d^{i-1}i. \end{aligned} \quad (35)$$

Für $h \geq 2$ folgt aus der Gleichung (16) und für $h = 1$ trivialerweise

$$\sum_{j=1}^b d_T(b_1, b_j) = \frac{((d-1)(h+1) - d)d^h + 1}{(d-1)^2} \quad (36)$$

und nach Umformungen (mit $d^h = b$) folgt

$$\sum_{j=1}^b d_T(b_1, b_j) = \frac{(h+1)b}{d-1} - \frac{db-1}{(d-1)^2}. \quad (37)$$

Eingesetzt in (35) und dann in (34) erhalten wir die Aussage. \square

Korollar 4.1. *Sei T ein (vollständiger) Arrangement-Baum der Ordnung $d \in \mathbb{N}$, $d \geq 2$, und der Höhe $h \in \mathbb{N}$. Sei weiter $b = d^h$ die Anzahl der Blätter des Arrangement-Baumes T und B die Menge, die der Blätter des Arrangement-Baumes T . Dann gilt:*

$$\sum_{\substack{b_i, b_j \in B \\ i < j}} d_T(b_i, b_j) = b \left((h+1)b - \frac{db-1}{d-1} \right) \quad (38)$$

Beweis: Diese Aussage folgt direkt aus Lemma 4.1, in dem aber alle Blätterpaare in den Summen 2 mal vorkommen. \square

Und nun können wir die zwei inneren Summen in (30) anders ausdrücken: Im Graphen G stehen uns m Kanten zur Verfügung. In den inneren Summen sind dazu die Blätter b_i und b_j bereits fixiert und wir müssen nur alle möglichen Arrangements zählen, die dieser Situation entsprechen. Dafür

brauchen wir $n - 2$ Knoten in den $b - 2$ Blättern einbetten; diese Knoten können aber auch permutiert werden. Zusammengefasst erhalten wir

$$\sum_{(u,v) \in E} \sum_{\substack{\phi: \\ \phi(u)=b_i \text{ und } \phi(v)=b_j}} 1 = m \binom{b-2}{n-2} (n-2)! \quad (39)$$

Eingesetzt in (30) und mit Verwendung von 2:

$$\begin{aligned} \mathbb{E}z(G, d, \phi) &= \frac{(b-n)!}{b!} \sum_{b_i, b_j \in B} d_T(b_i, b_j) m \binom{b-2}{n-2} (n-2)! \\ &= m \frac{(b-n)!}{b!} \binom{b-2}{n-2} (n-2)! \sum_{b_i, b_j \in B} d_T(b_i, b_j) \\ &= m \frac{(b-n)!}{b!} \frac{(b-2)!}{(b-n)!(n-2)!} (n-2)! \sum_{b_i, b_j \in B} d_T(b_i, b_j) \\ &= \frac{m}{b(b-1)} \sum_{b_i, b_j \in B} d_T(b_i, b_j) \end{aligned} \quad (40)$$

Durch Anwendung des Lemma 4.1 erhalten wir

$$\mathbb{E}z(G, d, \phi) = \frac{2m}{b-1} \left((h+1)b - \frac{bd-1}{d-1} \right). \quad (41)$$

Satz 4.1. *Sei $G = (V, E)$ ein ungerichteter Graph, wobei $|E| = m$, und $d \in \mathbb{N}$, $d \geq 2$ die Ordnung des Arrangement-Baumes T ist. Dann gilt für den Erwartungswert des Zielfunktionswertes $\mathbb{E}z(G, d, \phi)$ eines zufälligen Arrangements ϕ*

$$\mathbb{E}z(G, d, \phi) = \frac{2m}{b-1} \left((h+1)b - \frac{bd-1}{d-1} \right), \quad (41)$$

wobei h die Höhe und b die Anzahl der Blätter des Arrangement-Baumes T sind.

Beweis: Siehe die Herleitung oben. □

Da wir die Abhängigkeit des Erwartungswertes von der Dichte $d(G)$ untersuchen wollen, können wir den Ausdruck in (41) nur umformen und folgendes Korollar erhalten.

Korollar 4.2. *Sei $G = (V, E)$ ein ungerichteter Graph, wobei $|V| = n$ und $d(G)$ die Dichte des Graphen G ist, und $d \in \mathbb{N}$, $d \geq 2$ die Ordnung des*

Arrangement-Baumes T . Dann hängt der Erwartungswert des Zielfunktionswertes $\mathbb{E}z(G, d, \phi)$ eines zufälligen Arrangements ϕ von der Dichte $d(G)$ linear ab und es gilt:

$$\mathbb{E}z(G, d, \phi) = d(G) \frac{n(n-1)}{b-1} \left((h+1)b - \frac{bd-1}{d-1} \right), \quad (42)$$

wobei h die Höhe und b die Anzahl der Blätter des Arrangement-Baumes T sind.

Beweis: Für den Fall $|V| = n = 1$ gilt die Aussage, da dann der Erwartungswert gleich 0 ist. Wir nehmen also an, dass $|V| = n > 1$. Dann gilt

$$\frac{2m}{(b-1)} = \frac{2m}{n(n-1)} \frac{n(n-1)}{(b-1)} = d(G) \frac{n(n-1)}{(b-1)}. \quad (43)$$

Die Linearität folgt direkt aus (42). □

Zum Schluss mögen wir noch 2 Beispiele anschauen – das erste Beispiel (Beispiel 4.2) veranschaulicht die obige Herleitung, das zweite (Beispiel 4.3) unterlegt dann die Theorie durch empirische Versuche.

Beispiel 4.2. Betrachten wir einen vollständigen Graphen K_3 mit drei Knoten:

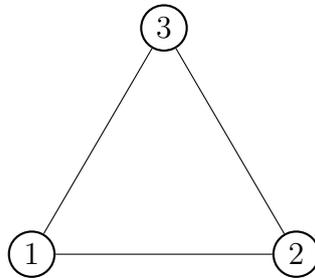


Abbildung 4.4: Graph

Da der Graph K_3 vollständig ist, gibt es für die Ordnung des Arrangement-Baumes $d = 2$ nur 4 unterschiedliche Arrangements ϕ_1, ϕ_2, ϕ_3 und ϕ_4 (die anderen entstehen nur durch eine Permutation der Knotennummern und besitzen denselben Zielfunktionswert $z(G, d, \phi)$), die gleich wahrscheinlich mit der Wahrscheinlichkeit $\mathbb{P}[\phi = \phi_i] = \frac{1}{4}$, wobei $\phi \in \{\phi_1, \phi_2, \phi_3, \phi_4\}$ auftreten (siehe Tabelle 4.3).

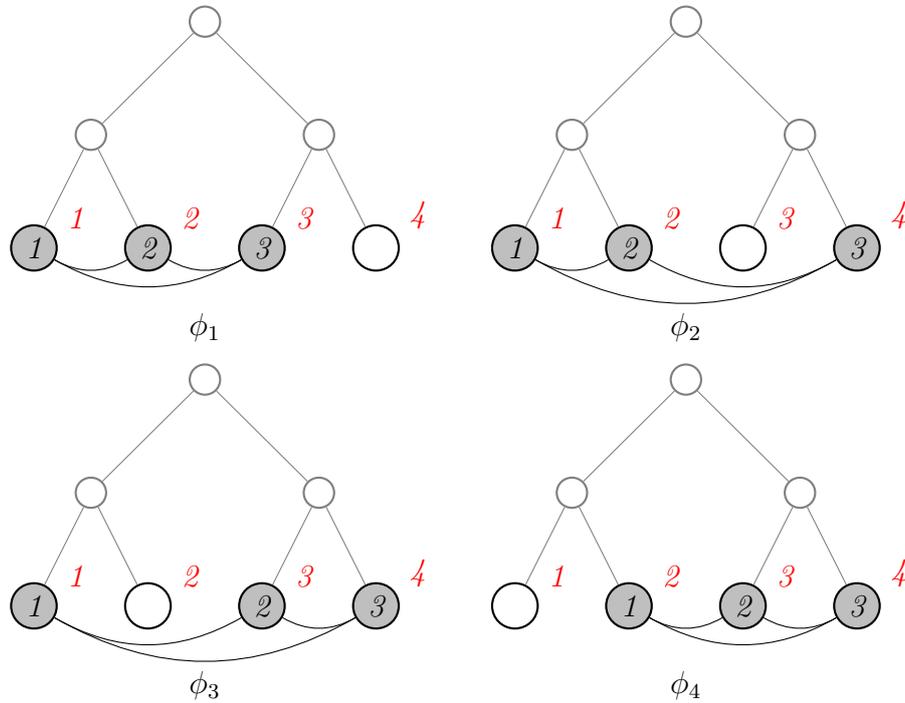


Tabelle 4.3: Alle unterschiedlichen Arrangements abgesehen von Knotenpermutationen; die Blätter-Indices sind in Rot

Im Prinzip können alle Blätterpaare durch eine Kante im Graphen K_3 verbunden werden und somit zu dem Zielfunktionswert beitragen. Da uns 4 Blätter zur Verfügung stehen, gibt es $\frac{4(4-1)}{2} = 6$ Paare, die mit der Länge des Verbindungsweges in dem Arrangement-Baum zu dem Zielfunktionswert beitragen können.

Wenn wir Abb. 4.3 anschauen und z.B. die Blätter b_1 und b_2 betrachten, dann wird die Länge ihres Verbindungsweges $d(b_1, b_2)$ zu dem Zielfunktionswert in den zwei obigen Fällen beitragen. Da wir aber insgesamt vier möglichen Arrangements haben (wie schon oben erwähnt, entstehen die anderen nur durch Permutation der Knotennummern), ist die Wahrscheinlichkeit, dass sich dieser Verbindungsweg mit seiner Länge an dem Zielfunktionswert beteiligt $\frac{1}{2}$.

Die möglichen Beiträge und die Wahrscheinlichkeiten, dass sich ein Blätterpaar (b_i, b_j) an dem Zielfunktionswert tatsächlich beteiligt, können aus Tabelle 4.3 abgelesen werden und sind in Tabelle 4.4 zusammengefasst.

Blätterpaar	Beitrag	Wahrscheinlichkeit
(b_1, b_2)	2	$\frac{1}{2}$
(b_1, b_3)	4	$\frac{1}{2}$
(b_1, b_4)	4	$\frac{1}{2}$
(b_2, b_3)	4	$\frac{1}{2}$
(b_2, b_4)	4	$\frac{1}{2}$
(b_3, b_4)	2	$\frac{1}{2}$
Erwartungswert:		10

Tabelle 4.4: Alle möglichen Beiträge von Blätterpaaren und die Wahrscheinlichkeiten, dass sie mit einer Kante verbunden sind

Der Erwartungswert lässt sich dann leicht berechnen:

$$\mathbb{E}z(G, d, \phi) = \frac{1}{2}2 + \frac{1}{2}4 + \frac{1}{2}4 + \frac{1}{2}4 + \frac{1}{2}4 + \frac{1}{2}2 = 10. \quad (44)$$

Da der Zielfunktionswert $z(G, d, \phi) = 10$ für alle ϕ , stimmt der Erwartungswert mit dem arithmetischen Mittelwert aller Zielfunktionswerte überein.

Beispiel 4.3. Schauen wir nun wieder unser Musterbeispiel mit dem Graphen aus Abb. 1.1 und mit der Ordnung des Arrangement-Baumes $d = 3$ an. Die Verteilung der Zielfunktionswerte ist in Tabelle 4.1 zusammengefasst und in Abb. 4.5 in Schwarz veranschaulicht. In Grau sind dann empirische Ergebnisse abgebildet, die die prozentuelle Verteilung der Zielfunktionswerte im arithmetischen Mittelwert über 10000 Zufallsgraphen für $n = 5$, $d = 3$ und $d(G) = \frac{2*7}{5(5-1)} = \frac{7}{10} = 0,70$ ($m = 7$) zeigen.

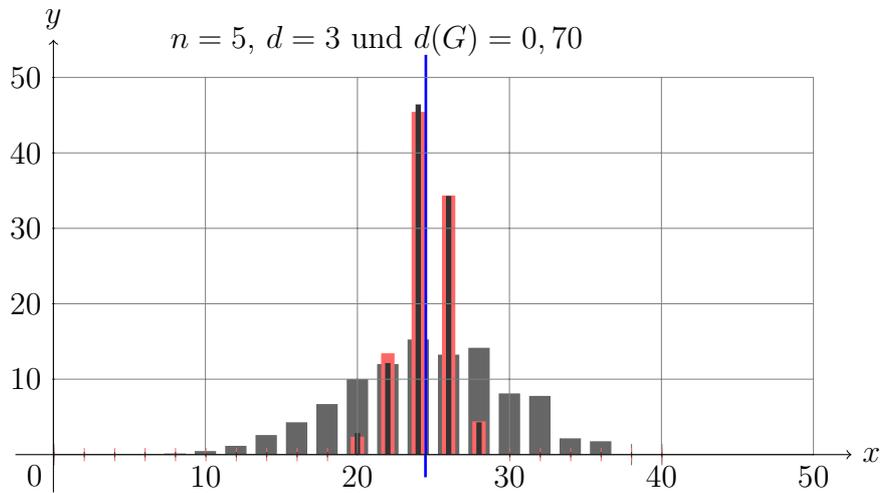


Abbildung 4.5: Verteilung der Zielfunktionswerte für $n = 5$, $d = 3$ und $d(G) = 0,70$; auf der x -Achse befinden sich die Zielfunktionswerte und auf der y -Achse der prozentuelle Anteil der Arrangements mit diesem Zielfunktionswert an der Menge aller Arrangements

- in Grau: die empirische prozentuelle Verteilung der Zielfunktionswerte im arithmetischen Mittelwert über 10000 Zufallsgraphen in Grau
- in Rot: die empirische prozentuelle Verteilung der Zielfunktionswerte im arithmetischen Mittelwert über 10000 Zufallsgraphen mit fester Dichte der Graphen
- in Schwarz: die Verteilung der Zielfunktionswerte des Graphen in Abb. 1.1, wobei $d = 3$
- in Blau: der berechnete Erwartungswert

Schon optisch stimmen die Werte überein. Aus Tabelle 4.1 können wir den arithmetischen Mittelwert für unser Beispiel leicht bestimmen:

$$\bar{z}(G, d, \phi) = \frac{432 * 20 + 1836 * 22 + 7020 * 24 + 5184 * 26 + 648 * 28}{15120} = 24,5. \quad (45)$$

Der Mittelwert, der aus den erstellten Zufallsgraphen berechnet wurde, ist 24,476200 (auf 6 Nachkommastellen gerundet).

Letztendlich können wir auch den Erwartungswert aus Satz 4.1

$$\mathbb{E}z(G, d, \phi) = \frac{2 * 7}{9 - 1} \left((2 + 1)9 - \frac{3^{2+1} - 1}{3 - 1} \right) = 24,5 \quad (46)$$

leicht berechnen. Er stimmt mit den zwei oben gezeigten Werten auch überein und wird in Abb. 4.5 in Blau abgebildet.

An dieser Stelle sollten wir aber noch ein wichtiges Phänomen erwähnen: Warum ist die Varianz der empirischen Ergebnisse (in Abb. 4.5 in Grau) und der durch die komplette Enumeration berechneten Anteile (in Abb. 4.5 in Schwarz) so unterschiedlich? Eigentlich ist bei den Zufallsgraphen nicht sicher, dass die generierten Graphen wirklich immer die Dichte $d(G) = 0,70$ besitzen. Diese Dichte ist nur der Mittelwert der Dichten der erstellten Graphen. Für den Erwartungswert spielt diese Tatsache keine Rolle, da die Herleitung für beide Fälle gleich ist, für die Varianz (siehe in Abschnitt 4.1.5) ist die Unterscheidung aber schon wichtig. Die empirische prozentuelle Verteilung der Zielfunktionswerte im arithmetischen Mittelwert über 10000 Zufallsgraphen mit fester Dichte kann für unser Beispiel in Abb. 4.5 in Rot gesehen werden. Die Ergebnisse stimmen dann sowohl angesichts des Erwartungswertes, als auch angesichts der Varianz überein.

Weitere Vergleiche empirischer Berechnungen mit den Erwartungswerten siehe in Tabelle A.4 im Anhang. Sie lassen uns vermuten, dass die Verteilung der *Normalverteilung* nahe stehen könnte. Auffällig sind auch die nach links (bzw. nach rechts) verlegte Maxima bei dünneren (bzw. dichteren) Graphen.

4.1.5 Varianz des Zielfunktionswertes eines zufälligen Arrangements für einen Graphen mit einer festen Dichte

Wie schon in Abschnitt 4.1.4 erwähnt, ist der Erwartungswert keine Heuristik, allerdings gibt es Gründe, warum wir ihn hier untersucht haben. Dieselben Argumente können wir auch verwenden, um die Untersuchung der *Varianz* zu begründen.

Zuerst müssen wir entscheiden, ob wir die *Varianz des Zielfunktionswertes eines zufälligen Arrangements für einen Zufallsgraphen*, oder die *Varianz des Zielfunktionswertes eines zufälligen Arrangements für einen Graphen mit einer festen Dichte* bestimmen wollen. Der Unterschied wurde schon in Beispiel 4.3 demonstriert. Da wir uns in dieser Arbeit aber vor allem mit *Heuristiken* beschäftigen, ist im Zusammenhang mit den *Randomisierten Heuristiken* (siehe in Abschnitt 4.1) für uns die zweite Variante sinnvoller. (Unsere Instanz besteht ja aus einem gegebenen Graphen G , der eine feste Dichte $d(G)$ besitzt und aus der *Ordnung des Arrangement-Baumes* $d \leq 2$.)

Zuerst aber noch einige Lemmas und ein Korollar.

Lemma 4.2. *Sei T ein (vollständiger) Arrangement-Baum der Ordnung $d \in \mathbb{N}$, $d \geq 2$, und der Höhe $h \in \mathbb{N}$. Sei weiter B die Menge der Blätter des*

Arrangement-Baumes T , wobei $b = |B| = d^h$ die Kardinalität dieser Menge ist. Dann gilt für jedes $\frac{l}{2} \in \mathbb{N}$, $1 \leq \frac{l}{2} \leq h$

$$|\{(b_i, b_j) \in B^2 | b_i \neq b_j \text{ und } d_T(b_i, b_j) = l\}| = \frac{bd^{\frac{l}{2}-1}(d-1)}{2}. \quad (47)$$

Beweis: Damit der Weg zwischen den Blättern b_i und b_j im Arrangement-Baum T die Länge l besitzt, muss er durch genau einen Knoten in der Ebene $e = h - \frac{l}{2}$ führen, wobei uns insgesamt d^e solcher Knoten zur Verfügung stehen. Wenn wir also von einem Knoten an der e -ten Ebene ausgehen, haben wir $\binom{d}{2}$ Möglichkeiten, zwei Unterbäume zu wählen. Jeder von diesen Unterbäumen besitzt dabei genau d^{h-e-1} Blätter. Daraus ergibt sich also:

$$\begin{aligned} |\{(b_i, b_j) \in B^2 | b_i \neq b_j \text{ und } d_T(b_i, b_j) = l\}| &= d^e \binom{d}{2} (d^{h-e-1})^2 \\ &= d^{h-\frac{l}{2}} \binom{d}{2} (d^{h-(h-\frac{l}{2})-1})^2 \\ &= \frac{d^{h+\frac{l}{2}-1}(d-1)}{2} \\ &= \frac{bd^{\frac{l}{2}-1}(d-1)}{2}. \end{aligned} \quad (48)$$

□

Korollar 4.3. Sei T ein (vollständiger) Arrangement-Baum der Ordnung $d \in \mathbb{N}$, $d \geq 2$, der Höhe $h \in \mathbb{N}$. Sei weiter B die Menge der Blätter des Arrangement-Baumes T mit $b = |B| = d^h$ und $\frac{l}{2} \in \mathbb{N}$, $1 \leq \frac{l}{2} \leq h$. Für je zwei Blätter $b_i, b_j \in B$, $b_i \neq b_j$, ist die Wahrscheinlichkeit, dass die Länge des Verbindungsweges zwischen den Blättern b_i und b_j gleich l ist, folgendermaßen gegeben:

$$\mathbb{P}[d_T(b_i, b_j) = l] = \frac{d-1}{b-1} d^{\frac{l}{2}-1}. \quad (49)$$

Beweis: Es ist klar, dass

$$\mathbb{P}[d_T(b_i, b_j) = l] = \frac{|\{(b_i, b_j) \in B^2 | b_i \neq b_j \text{ und } d_T(b_i, b_j) = l\}|}{|\{(b_i, b_j) \in B^2 | b_i \neq b_j\}|} \quad (50)$$

gilt.

Für den Nenner erhalten wir klarerweise

$$|\{(b_i, b_j) \in B^2 | b_i \neq b_j\}| = \frac{b(b-1)}{2}. \quad (51)$$

Wenn wir nun den Zähler aus Lemma 4.2 entnehmen und ihn zusammen mit (51) in (50) einsetzen, erhalten wir

$$\begin{aligned}\mathbb{P}[d_T(b_i, b_j) = l] &= \frac{\frac{bd^{\frac{l}{2}-1}(d-1)}{2}}{\frac{b(b-1)}{2}} \\ &= \frac{d-1}{b-1} d^{\frac{l}{2}-1}.\end{aligned}\tag{52}$$

□

Lemma 4.3. *Sei $h \in \mathbb{N}$ und $d \in \mathbb{N}$, wobei $d \geq 2$. Dann gilt*

$$\sum_{i=1}^h d^{i-1} i^2 = \frac{d^h}{d-1} \left(h^2 - \frac{2}{d-1} h + \frac{d+1}{(d-1)^2} \right) - \frac{d+1}{(d-1)^3}.\tag{53}$$

Beweis: Aus der Formel für die Partialsumme der geometrischen Reihe wissen wir, dass

$$\begin{aligned}\sum_{i=1}^h d^i &= d + d^2 + \dots + d^h \\ &= 1 + d + d^2 + \dots + d^h - 1 \\ &= \frac{1 - d^{h+1}}{1 - d} - 1\end{aligned}\tag{54}$$

gilt. Nun können wir beide Seiten der Gleichung nach d ableiten und

$$\begin{aligned}\sum_{i=1}^h i d^{i-1} &= \frac{-(h+1)d^h(1-d) - (1-d^{h+1})(-1)}{(1-d)^2} \\ &= \frac{1}{d-1} \left((h+1)d^h - \frac{d^{h+1} - 1}{d-1} \right)\end{aligned}\tag{55}$$

erhalten. Wenn wir nun beide Seiten wieder nach d ableiten, können wir nach Umformungen

$$\sum_{i=1}^h i(i-1)d^{i-2} = \frac{d^h}{d-1} \left(\frac{(h+1)h}{d} - 2\frac{h+1}{d-1} \right) + 2\frac{d^{h+1} - 1}{(d-1)^3}\tag{56}$$

schreiben.

Offensichtlich gilt

$$\sum_{i=1}^h d^{i-1} i^2 = d \sum_{i=1}^h i(i-1)d^{i-2} + \sum_{i=1}^h i d^{i-1}.\tag{57}$$

Wenn wir nun aus (55) und (56) einsetzen und den Ausdruck umformen, erhalten wir die Aussage. □

Lemma 4.4. Sei $G = (V, E)$ ein ungerichteter Graph, wobei $|V| = n$, $n \geq 2$ und $|E| = m = 1$, und $d \in \mathbb{N}$, $d \geq 2$, die Ordnung des Arrangement-Baumes T ist. Dann gilt für die Varianz des Zielfunktioswertes $\mathbb{V}z(G, d, \phi)$ eines zufälligen Arrangements ϕ

$$\mathbb{V}z(G, d, \phi) = 4 \left(\frac{d}{(d-1)^2} - h^2 \frac{b}{(b-1)^2} \right), \quad (58)$$

wobei h die Höhe und b die Anzahl der Blätter des Arrangement-Baumes T sind.

Beweis: Für den Fall $|V| = n$, wobei $n \geq 2$, und $|E| = 1$ haben wir im Prinzip nur eine Möglichkeit, wie der Graph $G = (V, E)$ aussehen kann: Es gibt zwei Knoten, o.B.d.A 1 und 2, die mit einer Kante verbunden sind und alle anderen Knoten sind isoliert (siehe Abb. 4.6).



Abbildung 4.6: Graph $G = (V, E)$, wobei $|V| = n$ und $|E| = 1$

Da wir auch die Anzahl aller Arrangements (Bemerkung 2.1) und den Erwartungswert des Zielfunktioswertes eines zufälligen Arrangements ϕ (siehe Satz 4.1) kennen, ist die Varianz mithilfe der Wahrscheinlichkeiten, dass zwei Blätter im Arrangement-Baum T durch einen Weg der Länge l ($\frac{l}{2} \in \mathbb{N}$, $1 \leq \frac{l}{2} \leq h$) verbunden sind, leicht zu bestimmen:

$$\mathbb{V}z(G, d, \phi) = \frac{1}{c} \sum_{i=1}^h 2 \frac{1}{2} b d^{i-1} (d-1) \frac{(b-2)!}{((b-2)-(n-2))!} (2i)^2 - (\mathbb{E}z(G, d, \phi))^2, \quad \text{wobei} \quad (59)$$

$c = \frac{b!}{(b-n)!}$ die Anzahl aller Arrangements darstellt. Dabei werden zuerst die Knoten 1 und 2 auf Blätter b_i und b_j arrangiert, wobei $i, j \in \mathbb{N}$ und $1 \leq i < j \leq b$, (es gibt für jedes Blätterpaar 2 Möglichkeiten), dann mit der Anzahl der Blätter b_1 und b_2 , die durch einen Weg der Länge $2i$ im Arrangement-Baum T verbunden sind (siehe Lemma 4.2), multipliziert und letztendlich müssen alle Arrangements der restlichen Knoten an die restlichen Blätter berücksichtigt werden ($\frac{(b-2)!}{((b-2)-(n-2))!}$). Der Rest folgt den Verschiebungssatz.

Da $c = \frac{b!}{(b-n)!}$, folgt

$$\mathbb{V}z(G, d, \phi) = 4 \frac{d-1}{b-1} \sum_{i=1}^h d^{i-1} i^2 - (\mathbb{E}z(G, d, \phi))^2. \quad (60)$$

Nun können wir Lemma 4.3 und Satz 4.1 anwenden und

$$\begin{aligned} \mathbb{V}z(G, d, \phi) = & 4 \frac{d-1}{b-1} \left(\frac{d^h}{d-1} \left(h^2 - \frac{2}{d-1}h + \frac{d+1}{(d-1)^2} \right) - \frac{d+1}{(d-1)^3} \right) - \\ & \left(\frac{1}{b-1} \left((h+1)b - \frac{bd-1}{d-1} \right) \right)^2 \end{aligned} \quad (61)$$

erhalten.

Nach einfachen Umformungen können wir

$$\mathbb{V}z(G, d, \phi) = 4 \left(\frac{d}{(d-1)^2} - h^2 \frac{b}{(b-1)^2} \right) \quad (62)$$

schreiben. \square

Wie wir sehen, hängt die *Varianz* des Zielfunktionswertes $\mathbb{V}z(G, d, \phi)$ eines zufälligen Arrangements für einen Graphen $G = (V, E)$ mit nur einer Kante von der Knotenanzahl (unter der Annahme, dass $|V| \geq 2$) nicht ab. Die Frage, ob es auch für Graphen mit einer größeren Kantenanzahl so ist, wird in Satz 4.2 ausführlich beantwortet und sie lautet nein; der Grund dafür ist, dass man in diesen Fällen o.B.d.A keine Grundform des Graphen festlegen kann.

Nun aber das wichtigste Ergebnis dieses Abschnitts.

Satz 4.2. *Sei $G = (V, E)$ ein ungerichteter Graph, wobei $|V| = n$ und $|E| = m$, und $d \in \mathbb{N}$, $d \geq 2$ die Ordnung des Arrangement-Baumes T ist. Dann gilt für die Varianz des Zielfunktionswertes $\mathbb{E}z(G, d, \phi)$ eines zufälligen Arrangements ϕ*

$$\mathbb{V}z(G, d, \phi) = \begin{cases} 0 & \text{für } h = \lceil \log_d n \rceil = 1, \\ t_1 t_2 t_3 & \text{für } h = \lceil \log_d n \rceil > 1, \text{ wobei} \end{cases} \quad (63)$$

t_i , $1 \leq i \leq 3$, folgendermaßen gegeben ist

$$t_1 = \frac{4}{(b-3)(b-2)(n+1)}, \quad (64)$$

$$t_2 = \left(\frac{d}{(d-1)^2} - h^2 \frac{b}{(b-1)^2} \right), \quad (65)$$

$$t_3 = \left(2(n+3-2b)m^2 + (b-1)((n+1)b - 4n)m \right) \quad (66)$$

und h die Höhe und b die Anzahl der Blätter des Arrangement-Baumes T sind.

Beweis: Zuerst werden wir den Fall $h = \lceil \log_d n \rceil = 1$ beweisen. Da $h = 1$ gilt, folgt direkt auch, dass $d_T(b_i, b_j) = 2$ für alle $b_i, b_j \in B$, wobei B die Menge aller Blätter des Arrangement-Baumes T ist, gilt. D.h. $z(G, d, \phi) = 2mh$ für alle ϕ und daraus folgt auch die Aussage.

Wir nehmen also an, dass $h = \lceil \log_d n \rceil > 1$.

Zuerst müssen wir einen Spezialfall beweisen. Für $n = 1$ gilt $m = 0$ und offensichtlich auch $\mathbb{V}z(G, d, \phi) = 0$. Da $d \geq 2$ und $h > 1$ angenommen wird, folgt direkt auch $b \geq 4$ und in (63), (64), (65) und (66) kann eingesetzt werden und nach einer leichten Berechnung erhalten wir auch 0.

Den Rest, d.h. $h = \lceil \log_d n \rceil > 1$ und $n \geq 2$, werden wir anhand der mathematischen Induktion nach der Kantenanzahl beweisen.

Für den Fall $|E| = 0$ gilt die Aussage offensichtlich, da wenn der Graph keine Kanten besitzt, ist der Zielfunktionswert $z(G, d, \phi) = 0$ für alle ϕ und deswegen gilt auch $\mathbb{V}z(G, d, \phi) = 0$. Durch Einsetzen in (63), (64), (65) und (66) erhalten wir auch $\mathbb{V}z(G, d, \phi) = 0$.

Sei also die Gültigkeit für $|E| = m$ angenommen. Betrachten wir nun den Fall $|E| = m + 1$. O.B.d.A. können wir annehmen, dass die neue Kante zwischen den Knoten 1 und 2 liegt. Um die Schreibweise einfacher zu machen, werden wir den Zielfunktionswert des Arrangements ϕ bezüglich des alten Graphen (mit m Kanten), bzw. bezüglich des neuen Graphen (mit $m + 1$ Kanten) als $z_m(\phi)$, bzw. $z_{m+1}(\phi)$ bezeichnen. Auch werden wir $\mathbb{E}z(m)$ (bzw. $\mathbb{E}z(1)$, bzw. $\mathbb{E}z(m+1)$) für $\mathbb{E}z(G, d, \phi)$ und $\mathbb{V}z(m)$ (bzw. $\mathbb{V}z(1)$, bzw. $\mathbb{V}z(m+1)$) für $\mathbb{V}z(G, d, \phi)$ unter der Annahme, dass $|E| = m$ (bzw. $|E| = 1$, bzw. $|E| = m + 1$) schreiben.

Nun können wir aus der Definition die Varianz $\mathbb{V}z(m + 1)$ für die neue Kantenanzahl ausdrücken (dabei benutzen wir, dass der Erwartungswert linear von m abhängt (siehe Satz 4.1)):

$$\begin{aligned}
\mathbb{V}z(m + 1) &= \frac{1}{c} \sum_{\phi} \left(z_{m+1}(\phi) - \mathbb{E}z(m + 1) \right)^2 \\
&= \frac{1}{c} \sum_{\phi} \left(z_{m+1}(\phi) - (m + 1)\mathbb{E}z(1) \right)^2 \\
&= \frac{1}{c} \sum_{\phi} \left(z_m(\phi) + d_T(\phi(1), \phi(2)) - (m\mathbb{E}z(1) + \mathbb{E}z(1)) \right)^2 \\
&= \frac{1}{c} \sum_{\phi} \left(z_m(\phi) - \mathbb{E}z(m) + d_T(\phi(1), \phi(2)) - \mathbb{E}z(1) \right)^2.
\end{aligned} \tag{67}$$

Nach leichter algebraischer Umformung erhalten wir

$$\begin{aligned}
\mathbb{V}z(m+1) &= \frac{1}{c} \sum_{\phi} \left[(z_m(\phi) - \mathbb{E}z(m))^2 + \right. \\
&\quad \left. 2(z_m(\phi) - \mathbb{E}z(m))(d_T(\phi(1), \phi(2)) - \mathbb{E}z(1)) + \right. \\
&\quad \left. (d_T(\phi(1), \phi(2)) - \mathbb{E}z(1))^2 \right] \\
&= \frac{1}{c} \left[\sum_{\phi} (z_m(\phi) - \mathbb{E}z(m))^2 + \right. \\
&\quad \left. 2 \sum_{\phi} \left((z_m(\phi) - \mathbb{E}z(m))(d_T(\phi(1), \phi(2)) - \mathbb{E}z(1)) \right) + \right. \\
&\quad \left. \sum_{\phi} (d_T(\phi(1), \phi(2)) - \mathbb{E}z(1))^2 \right]. \tag{68}
\end{aligned}$$

Diese drei Summen werden wir nun Separat behandeln:

- Die erste Summe können wir leicht aus der Definition der Varianz ausdrücken:

$$\sum_{\phi} (z_m(\phi) - \mathbb{E}z(m))^2 = c\mathbb{V}z(m). \tag{69}$$

- Den Ausdruck in der dritten Summe können wir noch einmal algebraisch umformen:

$$\begin{aligned}
\sum_{\phi} (d_T(\phi(1), \phi(2)) - \mathbb{E}z(1))^2 &= \sum_{\phi} (d_T(\phi(1), \phi(2)))^2 - \\
&\quad 2 \sum_{\phi} d_T(\phi(1), \phi(2))\mathbb{E}z(1) + \\
&\quad \sum_{\phi} (\mathbb{E}z(1))^2. \tag{70}
\end{aligned}$$

Auch nun werden die Summen separat behandelt:

- Für die erste Summe dieses Ausdrucks können wir den Erwartungswert der Quadrate des Zielfunktionswertes eines zufälligen Arrangements benutzen und dann den Verschiebungssatz anwenden:

$$\begin{aligned}
\sum_{\phi} (d_T(\phi(1), \phi(2)))^2 &= c\mathbb{E}(z(1))^2 \\
&= c(\mathbb{V}z(1) + (\mathbb{E}z(1))^2). \tag{71}
\end{aligned}$$

- Da $\mathbb{E}z(1)$ eine Konstante ist, können wir für die zweite Summe trivialerweise

$$\begin{aligned}\sum_{\phi} d_T(\phi(1), \phi(2)) \mathbb{E}z(1) &= \mathbb{E}z(1) \sum_{\phi} d_T(\phi(1), \phi(2)) \\ &= \mathbb{E}z(1) c \mathbb{E}z(1) \\ &= c (\mathbb{E}z(1))^2\end{aligned}\tag{72}$$

schreiben.

- Die dritte Summe ist offensichtlich

$$\sum_{\phi} (\mathbb{E}z(1))^2 = c (\mathbb{E}z(1))^2.\tag{73}$$

Eingesetzt in (70) erhalten wir also

$$\begin{aligned}\sum_{\phi} (d_T(\phi(1), \phi(2)) - \mathbb{E}z(1))^2 &= c (\mathbb{V}z(1) + (\mathbb{E}z(1))^2 - \\ &\quad 2 (\mathbb{E}z(1))^2 + (\mathbb{E}z(1))^2) \\ &= c \mathbb{V}z(1).\end{aligned}\tag{74}$$

- Nun bleibt also nur die zweite Summe übrig. Auch sie kann noch umgeformt werden:

$$\begin{aligned}\sum_{\phi} \left((z_m(\phi) - \mathbb{E}z(m)) (d_T(\phi(1), \phi(2)) - \mathbb{E}z(1)) \right) &= \\ \sum_{\phi} z_m(\phi) d_T(\phi(1), \phi(2)) - \sum_{\phi} z_m(\phi) \mathbb{E}z(1) - \\ \sum_{\phi} \mathbb{E}z(m) d_T(\phi(1), \phi(2)) + \sum_{\phi} \mathbb{E}z(m) \mathbb{E}z(1).\end{aligned}\tag{75}$$

Und wegen der Linearität des Erwartungswertes bezüglich der Kantenanzahl (siehe Satz 4.1) und nachdem alle Konstanten vor die Summen gestellt werden, erhalten wir

$$\begin{aligned}\sum_{\phi} \left((z_m(\phi) - \mathbb{E}z(m)) (d_T(\phi(1), \phi(2)) - \mathbb{E}z(1)) \right) &= \\ \sum_{\phi} z_m(\phi) d_T(\phi(1), \phi(2)) - \mathbb{E}z(1) \sum_{\phi} z_m(\phi) - \\ m \mathbb{E}z(1) \sum_{\phi} d_T(\phi(1), \phi(2)) + m (\mathbb{E}z(1))^2 \sum_{\phi} 1.\end{aligned}\tag{76}$$

Auch diesmal werden wir alle vier Summen separat behandeln:

– Die letzte Summe ist trivial:

$$\sum_{\phi} 1 = c. \quad (77)$$

– Auch die vorletzte Summe kann leicht ausgedrückt werden:

$$\sum_{\phi} d_T(\phi(1), \phi(2)) = c\mathbb{E}z(1). \quad (78)$$

– Die zweite Summe ist nur die Summation aller Zielfunktionswerte des alten Graphen (mit m Kanten):

$$\begin{aligned} \sum_{\phi} z_m(\phi) &= c\mathbb{E}z(m) \\ &= cm\mathbb{E}z(1). \end{aligned} \quad (79)$$

– Nun müssen wir die erste Summe ausdrücken, wofür wir die Arrangements ϕ in Mengen $P_i = \{\phi \mid d_T(\phi(1), \phi(2)) = 2i\}$, wobei $i \in \mathbb{N}$ und $1 \leq i \leq h$, aufteilen. Es gilt offensichtlich

$$\sum_{\phi} z_m(\phi) d_T(\phi(1), \phi(2)) = \sum_{i=1}^h \sum_{\substack{\phi: \\ d_T(\phi(1), \phi(2))=2i}} z_m(\phi) d_T(\phi(1), \phi(2)). \quad (80)$$

Und nun werden wir die innere Summe in (80) ausdrücken:

Für $d_T(\phi(1), \phi(2))$ können wir $2i$ schreiben.

Um in der Summe auch $z_m(\phi)$ behandeln zu können, brauchen wir den Erwartungswert des Zielfunktionswertes eines zufälligen Arrangements des alten Graphen (mit m Knoten) unter der Annahme, dass $d_T(\phi(1), \phi(2)) = 2i$, wobei $i \in \mathbb{N}$ und $1 \leq i \leq h$, gilt. Nun teilen wir die Menge aller Blätterpaare, aus der wir das Blätterpaar $(\phi(1), \phi(2))$ ausnehmen, in 2 disjunkte Mengen. Dafür definieren wir zuerst

$$B_p := \{(b_i, b_j) \mid b_i, b_j \in B \text{ und } b_i < b_j\} \setminus \{(\phi(1), \phi(2))\}. \quad (81)$$

Die zwei Mengen, \widetilde{B}_p und \overline{B}_p , werden dann folgendermaßen definiert:

$$(i) \quad \widetilde{B}_p := \{(b_i, b_j) \mid i, j \notin \{\phi(1), \phi(2)\}\}, \quad (82)$$

$$(ii) \quad \overline{B}_p := B_p \setminus \widetilde{B}_p. \quad (83)$$

Und nun müssen wir bestimmen, wie oft ein möglicher Wert der Länge des Weges zwischen zwei Blättern in der Summe in (80) vorkommt. Es ist klar, dass

$$(i) \quad |\widetilde{B}_p| = \frac{(b-2)(b-3)}{2}, \quad (84)$$

$$(ii) \quad |\overline{B}_p| = 2(b-2) \quad (85)$$

gilt. Uns stehen aber nicht alle Blätter zur Verfügung, da im Allgemeinen $n \neq b$ angenommen werden muss und so werden wir zwei neue Zahlen $\left(|\widetilde{B}_p|_n\right)$ und $\left(|\overline{B}_p|_n\right)$ definieren:

$$(i) \quad |\widetilde{B}_p|_n := \frac{(n-2)(n-3)}{2}, \quad (86)$$

$$(ii) \quad |\overline{B}_p|_n := 2(n-2). \quad (87)$$

Nun müssen wir die Erwartungswerte der Länge des Verbindungsweges zwischen den Blättern eines Blätterpaares der Menge \widetilde{B}_p , bzw. \overline{B}_p bestimmen. Die Menge \overline{B}_p können wir wieder in zwei disjunkte Mengen teilen:

$$(ii_1) \quad \overline{B}_{p,1} = \{(b_i, b_j) \in \overline{B}_p | b_i = \phi(1) \text{ oder } b_j = \phi(1)\}, \quad (88)$$

$$(ii_2) \quad \overline{B}_{p,2} = \{(b_i, b_j) \in \overline{B}_p | b_i = \phi(2) \text{ oder } b_j = \phi(2)\}. \quad (89)$$

Offensichtlich gilt:

$$\mathbb{E} [d_T(b_i, b_j) | (b_i, b_j) \in \overline{B}_p] = 2\mathbb{E} [d_T(b_i, b_j) | (b_i, b_j) \in \overline{B}_{p,1}]. \quad (90)$$

Wir können also

$$\begin{aligned} \mathbb{E} [d_T(b_i, b_j) | (b_i, b_j) \in \overline{B}_p] = \\ 2 \frac{1}{2(b-2)} \sum_{i=1}^b d_T(\phi(1), b_i) - d_T(\phi(1), \phi(2)) \end{aligned} \quad (91)$$

schreiben. Um die Summe berechnen zu können, teilen wir die Blätter b_1, b_2, \dots, b_b in Mengen B_i , $i = 0, 1, \dots, h$ so, dass $B_i = \{b | d_T(\phi(1), b) = 2i\}$ gilt. Die Kardinalität dieser Mengen wächst geometrisch und ist $|B_i| = (d-1)d^{i-1}$. Es gilt also

$$\sum_{i=1}^b d_T(\phi(1), b_i) = \sum_{i=1}^h (d-1)d^{i-1}2i = 2(d-1) \sum_{i=1}^h d^{i-1}i. \quad (92)$$

Jetzt können wir Lemma 3.1 oder noch besser direkt die Gleichung (15) aus dem Beweis dieses Lemmas anwenden

$$\sum_{i=1}^b d_T(\phi(1), b_i) = 2(d-1) \frac{((d-1)(h+1) - d)d^h + 1}{(d-1)^2}. \quad (93)$$

und da $d^h = b$ auch

$$\begin{aligned} \sum_{i=1}^b d_T(\phi(1), b_i) &= 2 \frac{((d-1)(h+1) - d)b + 1}{d-1} \\ &= 2 \left((h+1)b - \frac{db-1}{d-1} \right) \end{aligned} \quad (94)$$

schreiben. Da diese Summe im weiteren Verlauf des Beweises öfter vorkommt, werden wir

$$w := (h+1)b - \frac{db-1}{d-1} \quad (95)$$

definieren und eingesetzt in (94)

$$\sum_{i=1}^b d_T(\phi(1), b_i) = 2w \quad (96)$$

schreiben. Nun können wir in (91) einsetzen und für $d_T(\phi(1), \phi(2))$ klarerweise $2i$ schreiben (siehe auch (80)):

$$\mathbb{E} [d_T(b_i, b_j) | (b_i, b_j) \in \overline{B_p}] = \frac{1}{b-2} (2w - 2i) = 2 \frac{w-i}{b-2}. \quad (97)$$

Nun müssen wir den Erwartungswert auch für $\widetilde{B_p}$ untersuchen. Es gilt:

$$\begin{aligned} \mathbb{E} [d_T(b_i, b_j) | (b_i, b_j) \in \widetilde{B_p}] &= \\ &= \frac{1}{\frac{(b-2)(b-3)}{2}} \\ &= \left(\sum_{\substack{b_i, b_j \in B \\ i < j}} d_T(b_i, b_j) - 2(b-2) \mathbb{E} [d_T(b_i, b_j) | (b_i, b_j) \in \overline{B_p}] - 2i \right), \end{aligned} \quad (98)$$

wobei $2(b-2)\mathbb{E} [d_T(b_i, b_j)|(b_i, b_j) \in \overline{B_p}]$ der Summe der Längen der Verbindungswege aller Blätterpaare in der Menge $\overline{B_p}$ im Arrangement-Baum T entspricht (siehe auch (85)). Nun können wir die Formell (97) benutzen und einsetzen:

$$\begin{aligned} \mathbb{E} [d_T(b_i, b_j)|(b_i, b_j) \in \widetilde{B_p}] &= \\ \frac{1}{\frac{(b-2)(b-3)}{2}} \left(\sum_{\substack{b_i, b_j \in B \\ i < j}} d_T(b_i, b_j) - 2(b-2)2\frac{w-i}{b-2} - 2i \right) &= \\ \frac{2}{(b-2)(b-3)} \left(\sum_{\substack{b_i, b_j \in B \\ i < j}} d_T(b_i, b_j) - 4w + 2i \right). \end{aligned} \quad (99)$$

Eingesetzt aus Korollar 4.1 und aus (95) erhalten wir:

$$\begin{aligned} \mathbb{E} [d_T(b_i, b_j)|(b_i, b_j) \in \widetilde{B_p}] &= \frac{2}{(b-2)(b-3)} (bw - 4w + 2i) \\ &= 2\frac{bw - 4w + 2i}{(b-2)(b-3)}. \end{aligned} \quad (100)$$

Letztendlich müssen wir dann den ganzen Ausdruck mit der Anzahl der Arrangements, für die $d_T(\phi(1), \phi(2)) = 2i$, wobei $i \in \mathbb{N}$ und $1 \leq i \leq h$, gilt, ausmultiplizieren. Für die Anzahl solcher Arrangements können wir Korollar 4.3 und Bemerkung 2.1 benutzen und klarerweise $\mathbb{P}[d_T(b_i, b_j) = l] = \frac{d-1}{b-1}d^{i-1}c$ schreiben.

Jetzt können wir den Ausdruck für die innere Summe in (80) zusammenstellen, wozu wir noch u.a. (86), (87), (97) und (100) benutzen:

$$\begin{aligned} \sum_{\substack{\phi: \\ d_T(\phi(1), \phi(2))=2i}} z_m(\phi)d_T(\phi(1), \phi(2)) &= \\ \frac{d-1}{b-1}d^{i-1}c \frac{m}{\frac{(n-2)(n-3)}{2} + 2(n-2)} &= \\ \left(2(n-2)2\frac{w-i}{b-2} + \frac{(n-2)(n-3)}{2}2\frac{bw-4w+2i}{(b-2)(b-3)} \right). \end{aligned} \quad (101)$$

Wenn wir jetzt in (80) einsetzen, müssen wir diesen Ausdruck so bearbeiten, dass in der Summe (bzw. in den Summen) nur i, i^2

und d^{i-1} bleiben und nach einfachen algebraischen Umformungen erhalten wir:

$$\begin{aligned} \sum_{\phi} z_m(\phi) d_T(\phi(1), \phi(2)) = & \\ & 4c \frac{d-1}{(b-1)(b-2)(b-3)} \frac{m}{n+1} \cdot \\ & \left(2((n+3) - 2b) \sum_{i=i}^h d^{i-1} i^2 + w((n+1)b - 4n) \sum_{i=1}^h d^{i-1} i \right). \end{aligned} \quad (102)$$

Jetzt können wir Lemmas 4.3 und 3.1 anwenden und nach Umformungen erhalten wir (dabei wird berücksichtigt, dass $d^h = b$ gilt)

$$\begin{aligned} \sum_{\phi} z_m(\phi) d_T(\phi(1), \phi(2)) = & \\ & 4c \frac{m}{(b-1)(b-2)(b-3)(n+1)} \cdot \\ & \left(2((n+3) - 2b)q + ((n+1)b - 4n)w^2 \right), \text{ wobei} \quad (103) \\ & q = b \left(h^2 - \frac{2}{d-1}h + \frac{d+1}{(d-1)^2} \right) - \frac{d+1}{(d-1)^2} \text{ und} \\ & w = (h+1)b - \frac{db-1}{d-1}. \end{aligned}$$

Nun können wir (103), (79), (78) und (77) in (76) einsetzen und

$$\begin{aligned} \sum_{\phi} \left((z_m(\phi) - \mathbb{E}z(m)) (d_T(\phi(1), \phi(2)) - \mathbb{E}z(1)) \right) = & \\ & 4c \frac{m}{(b-1)(b-2)(b-3)(n+1)} \cdot \quad (104) \\ & \left(2((n+3) - 2b)q + ((n+1)b - 4n)w^2 \right) - \\ & \mathbb{E}z(1)cm\mathbb{E}z(1) - m\mathbb{E}z(1)c\mathbb{E}z(1) + m(\mathbb{E}z(1))^2 c, \end{aligned}$$

erhalten. Nach einer leichten Umformung:

$$\begin{aligned} \sum_{\phi} \left((z_m(\phi) - \mathbb{E}z(m)) (d_T(\phi(1), \phi(2)) - \mathbb{E}z(1)) \right) = \\ c \left(4 \frac{m}{(b-1)(b-2)(b-3)(n+1)} \right. \\ \left. \left(2((n+3) - 2b)q + ((n+1)b - 4n)w^2 \right) - m(\mathbb{E}z(1))^2 \right). \end{aligned} \quad (105)$$

Und nun können wir (69), (105) und (74) in (68) einsetzen und

$$\begin{aligned} \mathbb{V}z(m+1) = \\ \frac{1}{c} \left[c\mathbb{V}z(m) + 2c \left(4 \frac{m}{(b-1)(b-2)(b-3)(n+1)} \right. \right. \\ \left. \left. \left(2((n+3) - 2b)q + ((n+1)b - 4n)w^2 \right) - m(\mathbb{E}z(1))^2 \right) + \right. \\ \left. c\mathbb{V}z(1) \right], \end{aligned} \quad (106)$$

erhalten. Diesen Ausdruck können wir noch umformen:

$$\begin{aligned} \mathbb{V}z(m+1) = \\ \mathbb{V}z(m) + \mathbb{V}z(1) - 2m(\mathbb{E}z(1))^2 + \\ 8 \frac{m}{(b-1)(b-2)(b-3)(n+1)} \cdot \\ \left(2((n+3) - 2b)q + ((n+1)b - 4n)w^2 \right). \end{aligned} \quad (107)$$

Jetzt haben wir aber an der rechten Seite nur bekannte Ausdrücke: Für $\mathbb{V}z(m)$ können wir die Induktionsannahme nutzen ($n \geq 2$ wird angenommen), für $\mathbb{V}z(1)$ verwenden wir Lemma 4.4 und für $\mathbb{E}z(1)$ Satz 4.1. Alle

anderen Variablen sind bekannt. Wir erhalten also

$$\begin{aligned}
\mathbb{V}z(m+1) = & \\
& \frac{4}{(b-3)(b-2)(n+1)} \left(\frac{d}{(d-1)^2} - h^2 \cdot \frac{b}{(b-1)^2} \right) \\
& \left(2(n+3-2b)m^2 + (b-1)((n+1)b-4n)m \right) + \\
& 4 \left(\frac{d}{(d-1)^2} - h^2 \frac{b}{(b-1)^2} \right) - \\
& 2m \left(\frac{2}{b-1} \left((h+1)b - \frac{bd-1}{d-1} \right) \right)^2 + \\
& \frac{8m}{(b-1)(b-2)(b-3)(n+1)} \\
& \left(2((n+3)-2b)q + ((n+1)b-4n)w^2 \right). \tag{108}
\end{aligned}$$

Diesen Ausdruck können wir nun durch einfache algebraische Umformungen umwandeln und

$$\begin{aligned}
\mathbb{V}z(m+1) = & \\
& \frac{4}{(b-3)(b-2)(n+1)} \left(\frac{d}{(d-1)^2} - h^2 \frac{b}{(b-1)^2} \right) \cdot \\
& \left(2(n+3-2b)(m+1)^2 + (b-1)((n+1)b-4n)(m+1) \right) \tag{109}
\end{aligned}$$

erhalten. \square

Ähnlich wie bei dem Erwartungswert, können wir die Varianz auch auf die Dichte $d(G)$ beziehen.

Korollar 4.4. *Sei $G = (V, E)$ ein ungerichteter Graph mit $|V| = n$ und Dichte $d(G)$. Sei $d \in \mathbb{N}$, $d \geq 2$, die Ordnung des Arrangement-Baumes T . Dann hängt die Varianz des Zielfunktionswertes $\mathbb{E}z(G, d, \phi)$ eines zufälligen Arrangements ϕ von der Dichte $d(G)$ quadratisch ab und es gilt:*

$$\mathbb{V}z(G, d, \phi) = \begin{cases} 0 & \text{für } h = \lceil \log_d n \rceil = 1, \\ T_1 \left(T_2(d(G))^2 + T_2 d(G) \right) & \text{für } h = \lceil \log_d n \rceil > 1, \text{ wobei} \end{cases} \tag{110}$$

T_i , $1 \leq i \leq 3$, folgendermaßen gegeben ist

$$T_1 = \frac{2n(n-1)}{(b-3)(b-2)(n+1)} \left(\frac{d}{(d-1)^2} - h^2 \frac{b}{(b-1)^2} \right), \tag{111}$$

$$T_2 = (n+3-2b)n(n-1), \tag{112}$$

$$T_3 = (b-1)((n+1)b-4n) \tag{113}$$

und h die Höhe und b die Anzahl der Blätter des Arrangement-Baumes T sind.

Beweis: Der Fall $h = \lceil \log_d n \rceil = 1$ folgt direkt aus Satz 4.2.

Wenn $|V| = n = 1$, ist offensichtlich $\mathbb{V}z(G, d, \phi) = 0$ und in die Formel kann unter der Annahme, dass $h = \lceil \log_d n \rceil > 1$ gilt und da $d \geq 2$ angenommen wird, auch $b \geq 4$ gilt, eingesetzt werden.

Für die restlichen Fälle machen wir eine triviale Umformung des Ausdrucks (31) aus der Definition 4.3 und wir können

$$m = \frac{n(n-1)}{2}d(G) \quad (114)$$

schreiben und dann in Satz 4.2 einsetzen.

Da die Ausdrücke (111), (112) und (113) bezüglich $d(G)$ konstant sind, folgt die quadratische Abhängigkeit direkt aus (110). \square

In Abschnitt 4.1.2 haben wir einige relativ schwache und mathematisch fade Aussagen über Verwendbarkeit dieser Methode in Abhängigkeit von den Parametern getroffen. Durch numerische Beobachtungen konnten wir vermuten, dass sich diese Methode für ganz dünne Graphen und für ganz dichte (unter der Annahme, dass $n \approx b$ gilt) Graphen eignet. Die in diesem Abschnitt ermittelten Formeln über die Varianz eines zufälligen Arrangements bestätigen diese Vermutung.

Die bewiesene Formel werden wir nun anhand des Beispiel 4.4 illustrieren.

Beispiel 4.4. *Schauen wir nochmals unser Beispiel mit dem Graphen aus Abb. 1.1 an und für die Ordnung des Arrangement-Baumes wählen wir wieder $d = 3$. Aus Tabelle 4.1 können wir die Verteilung der Zielfunktionswerte aller Arrangements ϕ auslesen und in Abb. 4.5 in Schwarz graphisch ansehen. Da der Erwartungswert $\mathbb{E}z(G, d, \phi) = 24,5$ (siehe (46) in Beispiel 4.3), können wir aus Tabelle 4.1 die Varianz leicht bestimmen:*

$$\begin{aligned} s^2 &= \frac{1}{15120} (432 * (20 - 24,5)^2 + 1836 * (22 - 24,5)^2 + \\ &\quad 7020 * (24 - 24,5)^2 + 5184 * (26 - 24,5)^2 + 648 * (28 - 24,5)^2) \quad (115) \\ &= 2,75. \end{aligned}$$

Aus Satz 4.2 können wir nun die Varianz des Zielfunktionswertes eines zufälligen Arrangements eines Graphen mit denselben Parametern berechnen

(es gilt $h = 2$ und $b = 9$):

$$\begin{aligned} \mathbb{V}z(G, d, \phi) &= \frac{4}{(9-3)(9-2)(5+1)} \left(\frac{3}{(3-1)^2} - 2^2 \frac{9}{(9-1)^2} \right) \\ &\quad \left(2(5+3-2*9)7^2 + (9-1)((5+1)9-4*5)7 \right) \quad (116) \\ &= 2,75. \end{aligned}$$

Die Werte stimmen genau überein.

Viel interessanter, als die im Beispiel 4.5 vorgeführte Demonstration der Formel, ist die Abhängigkeit der Varianz $\mathbb{V}z(G, d, \phi)$ des Zielfunktionswertes eines zufälligen Arrangements von der Kantenanzahl (bzw. von der Dichte) wenn n und d konstant gewählt werden. Um sie zu veranschaulichen, werden wir noch ein Beispiel zeigen.

Beispiel 4.5. Sei die Knotenanzahl $n = 5$ und die Ordnung des Arrangement-Baumes $d = 3$ konstant gegeben. Offensichtlich gilt $h = 2$ und $b = 9$. Die maximale Kantenanzahl ist $\frac{5(5-1)}{2} = 10$; die in Betracht kommende Werte von m sind also $0, \dots, 10$. In Tabelle 4.5 können wir die Varianzen des Zielfunktionswertes $\mathbb{V}z(G, d, \phi)$ eines zufälligen Arrangements sehen, die aus Satz 4.2 berechnet wurden. In Abb. 4.7 können wir dieselben Werte graphisch in Rot sehen, wobei zur besseren Veranschaulichung der quadratischen Abhängigkeit der Varianz des Zielfunktionswertes eines zufälligen Arrangements von der Kantenanzahl auch reelle Werte für die Kantenanzahl in die Formel in Satz 4.2 eingesetzt wurden ($m \in \mathbb{R}$).

m	$d(G)$	$\mathbb{V}z(G, d, \phi)$	m	$d(G)$	$\mathbb{V}z(G, d, \phi)$
0	$\frac{0}{10} = 0$	0	6	$\frac{3}{5} = 0,6$	$\frac{19}{7} \approx 2,7143$
1	$\frac{1}{10} = 0,1$	$\frac{3}{4} = 0,75$	7	$\frac{7}{10} = 0,7$	$\frac{11}{4} = 2,75$
2	$\frac{1}{5} = 0,2$	$\frac{29}{21} \approx 1,3810$	8	$\frac{4}{5} = 0,8$	$\frac{8}{3} \approx 2,6667$
3	$\frac{3}{10} = 0,3$	$\frac{53}{28} \approx 1,8929$	9	$\frac{9}{10} = 0,9$	$\frac{69}{28} \approx 2,4643$
4	$\frac{2}{5} = 0,4$	$\frac{16}{7} \approx 2,2857$	10	$\frac{10}{10} = 1$	$\frac{15}{7} \approx 2,1429$
5	$\frac{1}{2} = 0,5$	$\frac{215}{84} \approx 2,5595$			

Tabelle 4.5: Varianz des Zielfunktionswertes $\mathbb{V}z(G, d, \phi)$ eines zufälligen Arrangements für $n = 5$, $d = 3$ und verschiedene Werte von m (bzw. Dichten $d(G)$)

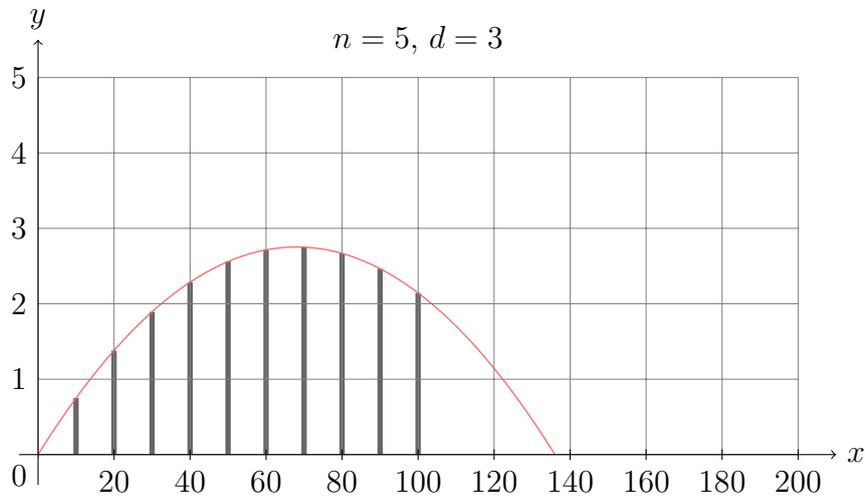


Abbildung 4.7: Varianz des Zielfunktionswertes $\mathbb{V}z(G, d, \phi)$ eines zufälligen Arrangements für $n = 5, d = 3$ und verschiedene Werte von m (bzw. Dichten $d(G)$); auf der x -Achse ist die Dichte und auf der y -Achse die Varianz abgebildet

- in Grau: gemessene Werte über 10000 Zufallsgraphen mit entsprechender Dichte
- in Rot: die berechneten Werte, wobei für die Kantenanzahl in Satz 4.2 auch reelle Werte eingesetzt wurden

Zum Schluss dieses Abschnitts werden wir noch 2 Bemerkungen erwähnen, die man zwar direkt aus Satz 4.2, oder aus Korollar 4.4 ableiten kann und die nur 2 Spezialfälle darstellen, die aber zum besseren Verständnis der Problematik dienen können. Aus diesem Grund werden wir auch auf die (recht technischen) direkten Herleitungen aus dem oben erwähnten Satz und Korollar verzichten und alternative Beweise zeigen.

Zuerst müssen wir aber noch **Flips** einführen, die dann vor allem in Abschnitt 4.3.2 Gebrauch finden.

Definition 4.4 (Flip, Flip an der e -ten Ebene, flippen). Sei $G = (V, E)$ ein ungerichteter Graph mit $|V| = n, d \in \mathbb{N}, d \geq 2$, die Ordnung des Arrangement-Baumes T und $\phi : V \rightarrow B$ ein Arrangement von G auf T , wobei die Menge B aus Blättern des Arrangement-Baumes T besteht. Gegeben sei weiters $e \in \mathbb{N}, 0 \leq e \leq h$, und die Höhe $h = \lceil \log_d n \rceil$ des Arrangement-Baumes T , sowie $l, k \in \mathbb{N}$, für die $1 \leq l < k \leq d$ gilt.

Sei f eine bijektive Funktion $f : B \rightarrow B$, die folgendermaßen definiert ist:

$$f(b_i) = \begin{cases} b_{i+(k-l)d^{h-(e+1)}} & \text{für } (l-1)d^{h-(e+1)} < i \leq ld^{h-(e+1)} \\ b_{i-(k-l)d^{h-(e+1)}} & \text{für } (k-1)d^{h-(e+1)} < i \leq kd^{h-(e+1)} \\ b_i & \text{sonst} \end{cases} . \quad (117)$$

Ein Arrangement $\phi_f : V \rightarrow B$, wobei $\phi_f = f \circ \phi$, heißt ein **Flip** des Arrangements ϕ , oder ein **Flip** des Arrangements ϕ **an der e -ten Ebene**. Wir sagen auch, dass wir das Arrangement ϕ **flippen**.

Der Sinn des *Flips* wird aus dem folgenden Beispiel klar.

Beispiel 4.6. Betrachten wir wieder den Graphen aus Abb. 1.1 in Beispiel 1.1 und die Ordnung des Arrangement-Baumes $d = 3$ (wie in Beispiel 1.2). Für die Höhe des Arrangement-Baumes T gilt $h = \lceil \log_d n \rceil = \lceil \log_3 5 \rceil = 2$. Betrachten wir nun das Arrangement ϕ , das in Abb. 1.3 abgebildet ist und für das gilt:

Kante v	Summand $\phi(v)$
1	b_1
2	b_8
3	b_3
4	b_5
5	b_4

Tabelle 4.6: Arrangement ϕ

Sei weiters $e = 1$, $l = 2$ und $k = 3$. Dann sieht das Arrangement ϕ_f folgendermaßen aus:

Kante v	Summand $\phi(v)$
1	b_1
2	b_4
3	b_3
4	b_8
5	b_7

Tabelle 4.7: Arrangement ϕ_f

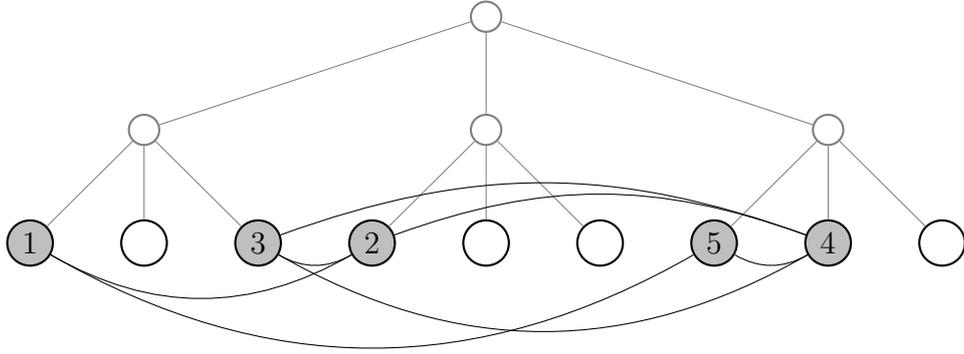


Abbildung 4.8: Arrangement ϕ_f

Bemerkung 4.2. Sei $G = (V, E)$ ein ungerichteter Graph mit $|V| = n$, $d \in \mathbb{N}$, $d \geq 2$ die Ordnung des Arrangement-Baumes T und $\phi : V \rightarrow B$ ein Arrangement von G auf T . Sei weiters ϕ_f ein Arrangement, das aus dem Arrangement ϕ durch einen Flip, oder durch mehrere nacheinander durchgeführten Flips entstanden ist. Dann gilt

$$z(G, d, \phi_f) = z(G, d, \phi). \quad (118)$$

Beweis: Es ist klar, dass

$$z(G, d, \phi) = 2a_h h + 2a_{h-1}(h-1) \dots 2a_1, \text{ wobei } a_h, a_{h-1}, \dots, a_1 \in \mathbb{N} \quad (119)$$

gilt. Dabei stellt jeder Koeffizient a_i , $1 \leq i \leq h$, die Anzahl der Kanten $(u, v) \in E$ dar, für die $d_T(\phi(u), \phi(v)) = 2i$ gilt. Es gibt also eine gewisse Anzahl der Kanten $(u, v) \in E$, für die die Länge des Weges zwischen den Blättern $\phi(u)$ und $\phi(v)$ gleich $2h$ ist, eine gewisse Anzahl der Kanten $(u, v) \in E$, für die die Länge gleich $2(h-1)$ ist usw. Durch einen Flip ändern sich aber klarerweise diese Werte nicht, da bei einem Flip nur in einem inneren Knoten an der e -ten Ebene des Arrangement-Baumes T der k -te Unterbaum mit dem l -ten Unterbaum vertauscht wird. Für einen Flip gilt also die Aussage, woraus auch die Gültigkeit für eine Reihe von Flips folgt. \square

Und nun die zwei Bemerkungen.

Bemerkung 4.3. Sei $G = (V, E)$ ein ungerichteter Graph, wobei $|V| = n$, und $d \in \mathbb{N}$, $d \geq 2$ die Ordnung des Arrangement-Baumes T . Dabei gelte $n = b$, wobei b die Anzahl der Blätter des Arrangement-Baumes T darstellt, und $d(G) = 1$. Dann gilt für die Varianz des Zielfunktionswertes $\mathbb{E}z(G, d, \phi)$ eines zufälligen Arrangements ϕ

$$\mathbb{V}z(G, d, \phi) = 0. \quad (120)$$

Beweis. Da der Graph G vollständig ist, werden bei jedem Arrangement ϕ alle Blätterpaare durch die Länge ihres Verbindungsweges im Arrangement-Baum T zu dem Zielfunktionswert beitragen und alle Arrangements besitzen also denselben Zielfunktionswert $z(G, d, \phi) = \mathbb{E}z(G, d, \phi)$ für alle ϕ . \square

Die Zweite Bemerkung ist weniger offensichtlich.

Bemerkung 4.4. Sei $G = (V, E)$ ein ungerichteter Graph, wobei $|V| = n$, und $d \in \mathbb{N}$, $d \geq 2$ die Ordnung des Arrangement-Baumes T . Dabei gelte $n - 1 = b$, wobei b die Anzahl der Blätter des Arrangement-Baumes T darstellt, und $d(G) = 1$. Dann gilt für die Varianz des Zielfunktionswertes $\mathbb{E}z(G, d, \phi)$ eines zufälligen Arrangements ϕ

$$\mathbb{V}z(G, d, \phi) = 0. \quad (121)$$

Beweis. Wir bilden ein zufälliges Arrangement ϕ von G auf T . Da $n - 1 = b$ gilt, gibt es genau ein Blatt b_i , für das $\phi(v) \neq b_i$ für alle $v \in V$ gilt. Offensichtlich können wir durch mehrere Flips ein Arrangement ϕ_f erstellen, sodass $\phi_f(v) \neq b_1$ für alle $v \in V$, ohne den Zielfunktionswert zu verändern. Da bei diesem Arrangement ϕ_f auch alle Blätterpaare (b_i, b_j) , wobei $b_i, b_j \in B$ und $i \neq 1$ und $j \neq 1$, wobei B die Menge aller Blätter des Arrangement-Baumes T ist, durch die Länge ihres Verbindungsweges im Arrangement-Baum T zu dem Zielfunktionswert beitragen, folgt aus Bemerkung 4.2 auch in diesem Fall $z(G, d, \phi) = \mathbb{E}z(G, d, \phi)$ für alle ϕ . \square

Weitere Beispiele, die auch diese 2 Bemerkungen graphisch veranschaulichen siehe in Tabelle A.5 (für Bemerkung 4.3 ist das linke untere und für Bemerkung 4.4 das rechte untere Beispiel relevant).

4.2 Greedy- und andere einfache Heuristiken

Im letzten Abschnitt haben wir die *randomisierten Heuristiken* untersucht – sie können uns zwar eine Vorstellung von dem Zielfunktionswert (bzw. von seiner oberen Schranke) geben, sie ignorieren aber völlig die Struktur des Graphen. Die in diesem Abschnitt behandelten Verfahren sind etwas sofistizierter. Am Anfang behandeln wir kurz eine *Greedy-Heuristik*, später dann alle Heuristiken, die mithilfe der Struktur des Graphen eine Lösung bilden, die dann nicht mehr, wie etwa bei der *lokalen Suche* verbessert wird.

4.2.1 Greedy-Heuristik

Der einfachste Ansatz, der der Struktur des Graphen gemäß eine Lösung sucht, ist die *Greedy-Heuristik*. Sie liefert für viele wichtigen *NP-vollständigen*

Probleme, wie z.B. etwa für das Bin-Packing-Problem, eine recht gute Lösung (siehe z.B. in KORTE ET AL [KV08]). Das und auch ihre Einfachheit ist der Grund, warum sie oft die erste Heuristik ist, die wir zu implementieren versuchen, wenn wir ein neues Problem lösen sollen. Natürlich gibt es auch für dieses Problem viele *Greedy-Ansätze*, im Prinzip können wir aber entweder ein *Arrangement* ϕ bilden (d.h. den nach einer Regel vorsortierten Knoten nacheinander die Blätter zuweisen), oder umgekehrt mit der *Umkehrfunktion* zum *Arrangement* ϕ arbeiten (d.h. die Blätter nacheinander auf die Knoten abbilden). Dabei minimieren (1. Variante) / maximieren (2. Variante) wir in jedem Schritt den Zuwachs des Zielfunktionswertes. Algorithmus 4.1 und Algorithmus 4.2 zeigen beide Varianten im Pseudocode. Dabei bezeichnet ϕ^{-1} die **Umkehrfunktion zum Arrangement** ϕ . In Algorithmus 4.2 nutzen wir außerdem, dass wir die Blätter nacheinander besetzen, d.h. wir brauchen nur n Blätter zu betrachten.

Input: $G = (V, E)$ ungerichteter Graph, wobei $V = \{v_1, \dots, v_n\}$ nach einer Regel vorsortiert; $d \geq 2$; sei weiter $|V| = n$ und $|E| = m$

Output: Arrangement ϕ

```

1:  $h := \lceil \log_d n \rceil$  und  $b := h^d$ ;
2: for  $i := 1$  to  $n$  do
3:    $z_{min}^+ := 2hm + 1$ ;
4:   for  $j := 1$  to  $b$  do
5:     if  $\nexists v \in V : \phi(v) = b_j$  then
6:        $z_b^+ := 0$ ;
7:       for  $q := 1$  to  $i$  do
8:         if  $v_q \in \Gamma(v_i)$  then
9:            $z_b^+ := z_b^+ + d_t(b_j, \phi(v_q))$ ;
10:        end if
11:       end for
12:       if  $z_b^+ < z_{min}^+$  then
13:          $z_{min}^+ := z_b^+$ ;
14:          $b_{min} := b_j$ ;
15:       end if
16:     end if
17:   end for
18:    $\phi(v_i) := b_{min}$ ;
19: end for
20: return  $\phi$ ;

```

Algorithmus 4.1: Greedy-Heuristik – Algorithmus 1

Input: $G = (V, E)$ ungerichteter Graph, wobei $V = \{v_1, \dots, v_n\}$; $d \geq 2$; B sei die Menge aller Blätter des Arrangement-Baumes T der Höhe $\lceil \log_d n \rceil$;
 sei weiter $|V| = n$ und $|E| = m$

Output: Arrangement ϕ

```

1: for  $i := 1$  to  $n$  do
2:    $z_{max}^+ := -1$ ;
3:   for  $j := 1$  to  $n$  do
4:     if  $\nexists b \in B : \phi^{-1}(b) = v_j$  then
5:        $z_v^+ := 0$ ;
6:       for  $q := 1$  to  $i$  do
7:         if  $\phi^{-1}(b_q) \in \Gamma(v_j)$  then
8:            $z_b^+ := z_b^+ + d_t(b_i, b_q)$ ;
9:         end if
10:      end for
11:      if  $z_v^+ > z_{max}^+$  then
12:         $z_{max}^+ := z_v^+$ ;
13:         $v_{max} := v_j$ ;
14:      end if
15:    end if
16:  end for
17:   $\phi^{-1}(v_i) := v_{max}$ ;
18: end for
19: Bestimme die Funktion  $\phi$  aus der Funktion  $\phi^{-1}$ ;
20: return  $\phi$ ;

```

Algorithmus 4.2: Greedy-Heuristik – Algorithmus 2

Bemerkung 4.5. *Abgesehen von den Kosten, die in Algorithmus 4.1 für die Vorsortierung auffallen, lässt sich Algorithmus 4.1 in $O(n^3 d \log n)$ und Algorithmus 4.2 in $O(n^3 \log n)$ implementieren.*

Beweis: Betrachten wir nun die 2 Algorithmen getrennt:

- Angenommen zuerst, dass wir die Knoten nacheinander arrangieren wollen (Algorithmus 4.1). Wir müssen 3 Schleifen ausführen, wobei eine davon $O(b)$ und zwei davon $O(n)$ Zeit benötigen. Die Bedingung in der Kode-Zeile 5 kann in $O(1)$ implementiert werden (wenn markiert, welche Blätter besetzt und welche unbesetzt sind).

Die Länge eines Weges zwischen 2 Blättern in einem Arrangement-Baum lässt sich in $O(2h) = O(\log b)$ berechnen, da wir in $O(1)$ den Vorgänger eines Knotens im Baum berechnen können. Da aber $b < nd$

und $d \leq n$ gilt, können wir $O(\log b) = O(\log nd) = O(\log n + \log d) = O(\log n)$ schreiben.

Zusammengesetzt erhalten wir $O(n^2 b \log n)$ und da $b < n$, folgt die angegebene Laufzeit klarerweise.

- Für Algorithmus 4.2 können wir dieselben Argumente benutzen, nur müssen wir dabei beachten, dass alle Schleifen nur $O(n)$ benötigen.

□

Obwohl beide Algorithmen eine ähnliche Zeitkomplexität besitzen und obwohl sie ähnlich aussehen, verhalten sie sich unterschiedlich. Schauen wir nun anhand eines Beispiels Algorithmus 4.1 näher an:

Beispiel 4.7. *Gegeben sei folgender Graph:*

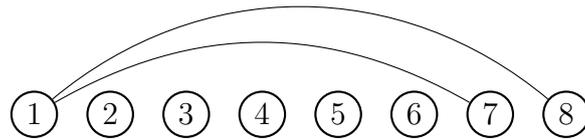


Abbildung 4.9: *Graph*

Sei weiters die Ordnung des Arrangement-Baumes $d = 2$. Es gilt $h = \lceil \log_d n \rceil = \lceil \log_2 8 \rceil = 3$. Nehmen wir Algorithmus 4.1, d.h. wir werden den Knoten nacheinander die Blätter zuweisen. Dabei versuchen wir, den Zielfunktionswertzuwachs zu minimieren. Die Knoten werden in der Reihenfolge genommen, in der sie in Abb. 4.9 nummeriert wurden:

Für den ersten Knoten können wir ein beliebiges Blatt wählen – nehmen wir also das Blatt 1. Dann müssen die Knoten 2 – 6 eingebettet werden und da sie den Zielfunktionswert nicht erhöhen, können wir diesen Knoten die Blätter 2 – 6 zuweisen. Und den restlichen Knoten 7 und 8 werden erst die 2 letzten Blätter eingeordnet. Wir erhalten folgendes Arrangement:

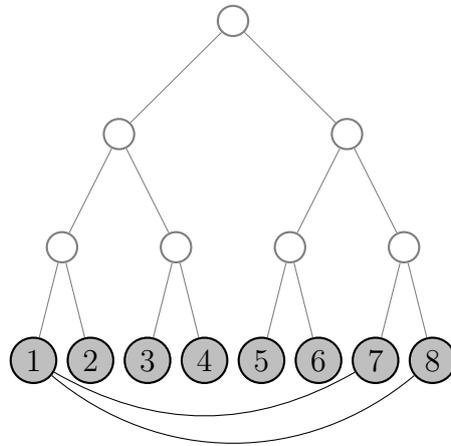


Abbildung 4.10: *Greedy-Heuristik für $d = 2$ – Arrangement*

Schauen wir nun die Tabelle aller Zielfunktionswerte an:

Zielfunktionswert	Arrangements	Prozentueller Anteil
6	3840	9,523810 %
8	9600	23,809524 %
10	15360	38,095238 %
12	11520	28,571429 %
Insgesamt Arrangements: 40320		

Tabelle 4.8: *Zielfunktionswerteverteilung*

Da unser Arrangement den Zielfunktionswert $z(G, d, \phi) = 12$ hat, handelt es sich um eines, das zu den schlechtesten gehört.

Die einfache *Greedy-Heuristik* kann im Fall, dass man Algorithmus 4.1 benutzt und eine zufällige Vorsortierung nimmt, ein beliebig schlechtes Arrangement zurückliefern. In Wirklichkeit hängt die Güte dieser Variante am stärksten von der Vorsortierung der Knoten ab. Eine mögliche Anordnung ergibt sich z.B. aus dem Breitensuche-Algorithmus gestartet von dem Knoten mit dem größten Knotengrad (siehe auch Abschnitt 4.2.2).

4.2.2 Breitensuche- und Tiefensuche- basierte Greedy-Verfahren

Eine weitere einfache Möglichkeit, wie man vorgehen kann, sind die *Breitensuche-* (bzw. die *Tiefensuche-*) *basierten Greedy-Verfahren*, die in jeder Zusammenhangskomponente einen Startknoten wählen und den Graphen in die

Input: $G = (V, E)$ ungerichteter Graph, wobei $V = \{v_1, \dots, v_n\}$; $d \geq 2$; sei weiter $|V| = n$ und $|E| = m$

Output: Arrangement ϕ

- 1: $h := \lceil \log_d n \rceil$ und $b := h^d$;
- 2: Finde alle Zusammenhangskomponenten V_1, \dots, V_q des Graphen G und ordne sie, dabei gelte $V_k = \{v_{k,1}, \dots, v_{k,|V_k|}\}$;
- 3: Berechne die Länge des Verbindungsweges zwischen allen Blätterpaaren (b_i, b_j) , $1 \leq i < j \leq b$;
- 4: $l := 0$;
- 5: **for** $k := 1$ **to** q **do**
- 6: $z_{min}^+ := 2hm + 1$;
- 7: **for** $i := 1$ **to** $|V_k|$ **do**
- 8: Starte die Breitensuche / Tiefensuche aus dem Knoten $v_{k,i}$ und erhalte daraus die Menge $V_{k,i} = \{v_{k,1}^i, \dots, v_{k,|V_k|}^i\}$;
- 9: $z_i^+ := \sum_{(v_{k,x}^i, v_{k,y}^i) \in E(G[V_k])} d_T(b_{l+x}, b_{l+y})$;
- 10: **if** $z_i^+ < z_{min}^+$ **then**
- 11: $z_{min}^+ := z_i^+$;
- 12: **for** $j := 1$ **to** $|V_k|$ **do**
- 13: $\phi(v_{k,j}^i) := l + j$;
- 14: **end for**
- 15: **end if**
- 16: **end for**
- 17: $l := l + |V_k|$;
- 18: **end for**
- 19: **return** ϕ ;

Algorithmus 4.3: Breitensuche- und Tiefensuche- basierte Greedy-Verfahren, in den alle Knoten als Startknoten benutzt werden und das beste daraus resultierende Arrangement ausgewählt wird

Breite (bzw. in die *Tiefe*) durchsuchen. Die Knoten werden dann in der daraus resultierenden Reihenfolge den Blättern zugeordnet. Die Wahl des Startknotens und der Reihenfolge der Zusammenhangskomponenten ist dabei zufällig. Eine andere Möglichkeit ist, in jeder Zusammenhangskomponente immer den Knoten mit dem *größten Knotengrad* als *Startknoten* zu wählen.

Vom Vorteil ist natürlich, dass alle Knoten jeder Zusammenhangskomponente nebeneinander eingebettet sind und für die meisten Instanzen liefert dieses Verfahren in einer guten Laufzeit relativ akzeptierbare Ergebnisse. Am schlechtesten verhält sich die *Breitensuche*-Variante für diejenige Graphen, die *kettenartige Unterstrukturen* und *erweiterte Sterne* (siehe Ab-

schnitt 5.5) als Teilgraphen aufweisen; für solche Graphen eignet sich dagegen die *Tiefensuche*-Variante besser.

Natürlich können wir aber auch andere Auswahlkriterien als den Knotengrad für den Startknoten benutzen. Eine Möglichkeit wäre, in jeder Zusammenhangskomponente alle Möglichkeiten auszuprobieren und dann das beste Arrangement zu wählen (die Reihenfolge, in der die Zusammenhangskomponenten angegriffen werden, muss dabei schon vorher festgelegt worden sein). Die Laufzeit wird stets polynomiell sein.

Schauen wir nun Algorithmus 4.3, wobei $G[V]$ den durch die Knotenmenge V **induzierten Teilgraphen** darstellt. In Algorithmus 4.3 werden zuerst alle Zusammenhangskomponenten gefunden, diese Suche ändert die Zeitkomplexität nicht.

Bemerkung 4.6. *Algorithmus 4.3 lässt sich in $O(nm(n+m) + n^2d^2 \log n)$ implementieren.*

Beweis: Eine Breiten-, bzw. Tiefensuche in einem Graphen lässt sich in $O(n+m)$ implementieren. Diese Breiten-, bzw. Tiefensuche müssen wir n -mal durchführen. Dabei müssen wir für jede Kante die Länge des Verbindungsweges zwischen zwei Blättern berechnen, um den Zielfunktionswert bestimmen zu können.

Wenn wir am Anfang des Algorithmus die Längen der Wege zwischen allen Blätterpaaren im Arrangement-Baum T berechnen und speichern, brauchen wir für diese Operation $O(b^2 \log n) = O(n^2d^2 \log n)$ (um die Länge des Verbindungsweges zwischen zwei Blättern im Arrangement-Baum T zu berechnen, brauchen wir $O(2h) = O(h) = O(\log b) = O(\log nd) = O(\log n + \log d) = O(\log n)$ Zeit).

Um am Anfang alle Zusammenhangskomponenten zu finden, brauchen wir nur eine lineare Zeit bezüglich n und m [KV08].

Insgesamt ist die Laufzeit also $O(nm(n+m) + n^2d^2 \log n)$. □

4.2.3 Heuristische Methode der minimalen Schnitte in beschränkten Mengen

Wenn wir darüber nachdenken, was für den größten Zielfunktionswertzuwachs sorgt, sind es diejenigen Kanten, deren Endknoten im *Arrangement-Baum* über Wurzel verbunden sind. Bei fast vollständigen Graphen gibt es natürlich immer viele solche Kanten, es gibt aber (vor allem aus der Praxis genommene) Instanzen, die eine gewisse Struktur aufweisen, die wir ausnutzen können.

Beispiel 4.8. *Betrachten wir den Graphen aus Abb. 4.11.*

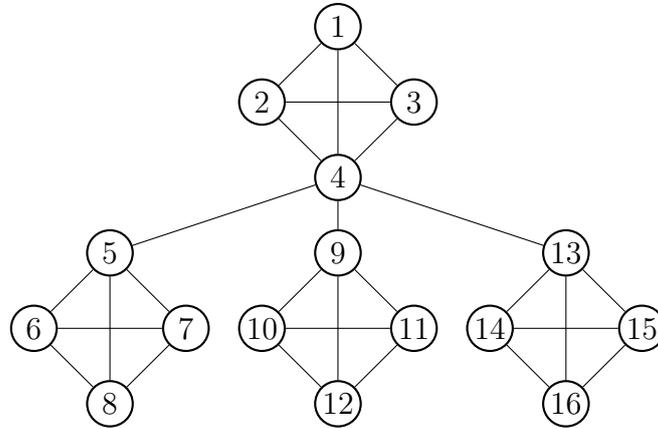


Abbildung 4.11: Graph

Wenn wir weiter die Ordnung des Arrangement-Baumes $d = 4$ annehmen, gilt $h = \lceil \log_d n \rceil = \lceil \log_4 16 \rceil = 2$. Es ist klar, wie eine Optimallösung aussehen muss: Alle Kanten werden in den Zielfunktionswert entweder mit 2, oder mit 4 angerechnet. Wir wollen also die Anzahl der Kanten minimieren, die mit 4 angerechnet werden. Eine von den 4 stark zusammenhängenden Komponenten ($\{1, 2, 3, 4\}$, $\{5, 6, 7, 8\}$, $\{9, 10, 11, 12\}$ und $\{13, 14, 15, 16\}$) zu teilen, würde mindestens 3 solche Kanten verursachen und ein solches Arrangement kann also nie besser sein, als ein Arrangement, das jeder dieser Clique einen Unterbaum, wie es in Abb. 4.12 veranschaulicht wird, zuordnet.

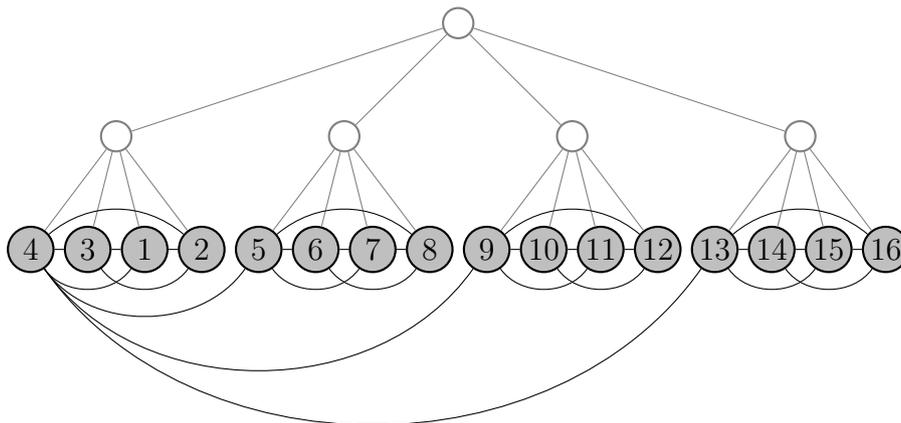


Abbildung 4.12: Optimallösung für $d = 4$

In der Optimallösung von Beispiel 4.8 haben wir versucht, die Anzahl der Kanten, die mit dem größten Wert an dem Zielfunktionswert beteiligt sind, zu minimieren. Das ist auch die Hauptidee der in diesem Abschnitt dargestellten Methode. Mit anderen Worten werden wir zuerst die Knoten fixieren, die dem linken Unterbaum zugeordnet werden indem wir einen minimalen Schnitt suchen, der den Graphen in $\frac{n}{d}$ und $n - \frac{n}{d}$ Knoten teilt (sofern $b = n$). Wenn wir dies getan haben, machen wir dasselbe für den Restgraphen bis alle Knoten zugeordnet werden und wir erhalten d Unterbäume. Dann können wir den Algorithmus Rekursiv für alle Unterbäume neu starten.

Sollte $b \neq n$, so können wir die ersten $b - n$ Blätter frei lassen und den Rest wie oben beschrieben behandeln, denn wie in Abschnitt 4.1.3 bereits erwähnt, ist es empirisch gesehen ventünftig, sich auf zusammenhängende Arrangements einzuschränken.

Ein Problem dieses Verfahrens ist, dass das genaue Herausfinden eines solchen Schnittes allerdings *NP-schwer* ist. Trotzdem gibt es eine Heuristik, die uns erlaubt, die oben genannte Idee umzusetzen und die dann zu den besten Ergebnissen für dieses Problem führen wird: Selbst für fast alle Testinstanzen mit bekannter Optimallösung hat diese Methode eine Optimallösung geliefert (siehe Kapitel 6 und Tabelle A.2)!

Bevor wir den Algorithmus beschreiben, widmen wir uns zuerst dem *minimalen Schnitt-Problem in beschränkten Mengen (MinSPBM)*.

Definition 4.5 (Schnitt, Kapazität eines Schnittes). Sei $G = (V, E)$ ein ungerichteter Graph. Eine Menge $X \subseteq V$ heißt **Schnitt**. Die Anzahl aller Kanten $(u, v) \in E$, sodass $u \in X$ und $v \in V \setminus X$ gilt, werden wir **Kapazität eines Schnittes** bezeichnen.

DAS MINIMALE SCHNITT-PROBLEM IN BESCHRÄNKTEN MENSCHEN (MINSPBM)

Gegeben: Ungerichteter Graph $G = (V, E)$ und $u, o \in \mathbb{N}$.

Gesucht: Ein minimaler Schnitt $X \subseteq V(G)$ bezüglich seiner Kapazität, das die Nebenbedingung $u \leq |X| \leq o$ erfüllt.

Satz 4.3. *Das MinSPBM ist NP-schwer.*

Beweis: Siehe BODLAENDER [Bod87]. □

Eine Möglichkeit, das *MinSPBM* heuristisch zu lösen, bieten lokale Suchverfahren, wie z.B. die in Algorithmus 4.5 beschriebene Methode.

Input: $G = (V, E)$ ungerichteter Graph; $d \in \mathbb{N}$ Ordnung des Arrangement-Baumes T , $d \geq 2$; sei weiter $|V| = n$

Output: Arrangement ϕ von G auf T

- 1: $h := \lceil \log_d n \rceil$ und $b := h^d$;
- 2: **if** Die Höhe des Arrangement-Baumes $h = 1$ **then**
- 3: Erstelle das Arrangement ϕ zufällig;
- 4: **else**
- 5: Setze die Anzahl der freien Blätter $b_{frei} := b - n$;
- 6: **for** $i := 1$ **to** d **do**
- 7: **if** $b_{frei} \geq \frac{b}{d}$ **then**
- 8: $\phi^{-1}(l) := \text{unbesetzt}$, $(i-1)\frac{b}{d} \leq l \leq i\frac{b}{d}$;
- 9: $b_{frei} := b_{frei} - \frac{b}{d}$;
- 10: **else**
- 11: Finde einen minimalen Schnitt $X \subseteq V(G)$ im Graphen G , der die Nebenbedingung $\frac{b}{d} - b_{frei} \leq |X| \leq \frac{b}{d}$ erfüllt;
- 12: Löse das Problem für $G[X]$ und einen Arrangement-Baum der Ordnung d , der Höhe $h-1$ und mit $\frac{b}{d}$ Blättern; sei ϕ_X die Lösung dieses Problems
- 13: Bestimme aus ϕ_X die Umkehrfunktion ϕ_X^{-1} ;
- 14: **for** $j := 1$ **to** $\frac{b}{d}$ **do**
- 15: $\phi^{-1}((i-1)\frac{b}{d} + j) := \phi_X^{-1}(j)$;
- 16: **end for**
- 17: $G := G[G \setminus X]$;
- 18: $b_{frei} := b_{frei} - (\frac{b}{d} - |X|)$;
- 19: **end if**
- 20: **end for**
- 21: Bestimme die Funktion ϕ aus der Funktion ϕ^{-1} ;
- 22: **end if**

Algorithmus 4.4: Heuristische Methode der minimalen Schnitte in beschränkten Mengen – allgemeiner Algorithmus

Input: G ungerichteter Graph; $u, o \in \mathbb{N}$
Output: Schnitt $X \subseteq V(G)$, der die Bedingung $u \leq |X| \leq o$ erfüllt

- 1: **if** $u > o$ **or** $u > |V(G)|$ **then**
- 2: **return** Diese Instanz besitzt keine zulässige Lösung;
- 3: **else if** $u = |V(G)|$ **or** $o \geq |V(G)|$ **then**
- 4: **return** G ;
- 5: **else**
- 6: Bilde einen zufälligen Schnitt $X \subset V(G)$, wobei $u \leq |X| \leq o$;
- 7: $X_{wert} := |\{e = (u, v) | u \in X \text{ und } v \notin X\}|$;
- 8: **for** $k := u$ **to** o **do**
- 9: Bilde einen zufälligen Schnitt $X' \subset V(G)$, wobei $|X'| = k$;
- 10: $X'_{wert} := |\{e = (u, v) | u \in X' \text{ und } v \notin X'\}|$;
- 11: **repeat**
- 12: **for all** $u \in X'$ **do**
- 13: **for all** $v \in V(G) \setminus X'$ **do**
- 14: $\delta^{V(G) \setminus X'}(u) := |\{e = (u, u') | \text{wobei } u' \notin X'\}|$;
- 15: $\delta^{X'}(u) := |\{e = (u, u') | u' \in X'\}|$;
- 16: $\delta^{X'}(v) := |\{e = (v, v') | v' \in X'\}|$;
- 17: $\delta^{V(G) \setminus X'}(v) := |\{e = (v, v') | v' \notin X'\}|$;
- 18: $gewinn := \delta^{V(G) \setminus X'}(u) - \delta^{X'}(u) + \delta^{X'}(v) - \delta^{V(G) \setminus X'}(v)$;
- 19: **if** $(u, v) \in E(G)$ **then**
- 20: $gewinn := gewinn - 2$;
- 21: **end if**
- 22: **if** $gewinn > 0$ **then**
- 23: $X' := X' \setminus \{u\} \cup \{v\}$;
- 24: $X'_{wert} := X'_{wert} - gewinn$;
- 25: **end if**
- 26: **end for**
- 27: **end for**
- 28: **until** Kein besserer Schnitt wurde gefunden
- 29: **if** $X'_{wert} < X_{wert}$ **then**
- 30: $X := X'$;
- 31: $X_{wert} = X'_{wert}$;
- 32: **end if**
- 33: **end for**
- 34: **return** X ;
- 35: **end if**

Algorithmus 4.5: MinSPBM – Heuristik

Mit Worten beschrieben: Zuerst werden die trivialen Fälle behandelt.

Dann wird ein zufälliger Schnitt X erstellt, der als die bis jetzt beste gefundene Lösung genommen wird, die dann im Laufe des Algorithmus eventuell ersetzt werden kann. Dann wird für jede zulässige Größe des Schnittes k ($u \leq k \leq o$) eine zufällige Startinstanz X' erstellt, die die Bedingung $|X| = k$ erfüllt. Diese Lösung wird dann verbessert, indem man diejenigen Knotenpaare u, v , wobei $u \in X$ und $v \notin X$, sucht, die eine Zielfunktionswertverbesserung durch ihren Austausch ermöglichen. Eine solche Lösung wird für alle $u \leq k \leq o$ konstruiert und dann die beste daraus gewählt.

Die Zeitkomplexitätsanalyse dieses Algorithmus ist recht einfach, sie wird allerdings erst bei der Zeitkomplexitätsanalyse des Algorithmus 4.4 durchgeführt, da man dann gewisse Annahmen über die Parameter u und o machen kann. Hier sollte auch erwähnt werden, dass wir diesen Algorithmus so verbessern können, indem wir anstatt nur mit den Knoten auch mit den Knotenpaaren in X' und $V(G) \setminus X'$ arbeiten würden – wir würden also Knotenpaare austauschen. Im Allgemeinen könnten wir auch mit 3 und mehreren Knoten arbeiten, die Zeitkomplexität wird aber dadurch viel schlechter und die dadurch gewonnene Zielfunktionsverbesserung relativ gering.

Und jetzt zu unserem Problem. Schauen wir nun kurz Algorithmus 4.4 an, wobei ϕ^{-1} die Umkehrfunktion zum Arrangement ϕ ist und $G[V]$ den durch die Knotenmenge V induzierten Teilgraphen darstellt. Das Problem wird an d Unterprobleme geteilt, bis man zu den trivialen Instanzen kommt.

Algorithmus 4.4 nutzt Algorithmus 4.5 als Subroutine.

Satz 4.4. *Algorithmus 4.4 unter Verwendung von Algorithmus 4.5 lässt sich in $O(n^5 d^3)$ implementieren.*

Beweis: Um unseren Graphen wie beschrieben teilen zu können, müssen wir im allgemeinen Fall zuerst einen minimalen Schnitt in beschränkten Mengen mit höchstens $\frac{b}{d}$ Knoten im Graphen G finden. In unserem Algorithmus nehmen die Grenzen u und o immer solche Werte an, dass $u \leq o \leq \frac{b}{d}$ gilt (für die 0. Ebene). Weiters gilt auch (für die 0. Ebene) $o - u \leq \frac{b}{d}$. Schauen wir nun den ersten Schnitt näher an: Die äußerste Schleife in Algorithmus 4.5 wird höchstens $\frac{b}{d}$ mal wiederholt. Für die Anzahl der Durchführungen der zwei inneren Schleifen muss dann $O\left(\frac{b(d-1)b}{d}\right)$ angerechnet werden. Da für das Optimum der Subroutine $0 \leq \text{MinSPBM}_{\text{Optimum}} \leq \frac{b(d-1)b}{d}$ gilt (als Schranke wird die maximale Anzahl der Kanten in einem zulässigen Schnitt genommen), können wir die Anzahl der Verbesserungen an dem Zielfunktionswert mit $O\left(\frac{b(d-1)b}{d}\right)$ beschränken (repeat-Schleife). Wir brauchen also insgesamt $O\left(\frac{b}{d} \left(\frac{b(d-1)b}{d}\right)^2\right)$, um den ersten minimalen beschränkten Schnitt

mithilfe des Algorithmus 4.5 zu approximieren. Für die ersten $d-1$ Schritte an der 0. Ebene brauchen wir also insgesamt

$$O\left(\frac{b}{d}\left(\frac{b(d-1)b}{d}\right)^2 + \frac{b}{d}\left(\frac{b(d-2)b}{d}\right)^2 + \cdots + \frac{b}{d}\left(\frac{b(d-(d-1))b}{d}\right)^2\right). \quad (122)$$

(Die Kardinalität der Menge X' bleibt immer gleich, die obere Schranke für die Anzahl der Knoten des Restgraphen sinkt dagegen immer um die Anzahl der Blätter an dem vorigen Unterbaum, d.h. um $\frac{b}{d}$.)

Umgeformt erhalten wir

$$O\left(\left(\frac{b}{d}\right)^5 ((d-1)^2 + (d-2)^2 + \cdots + (1)^2)\right) \quad (123)$$

für die 0. Ebene. Da aber

$$(d-1)^2 + (d-2)^2 + \cdots + (1)^2 = \frac{(d-1)((d-1)+1)(2(d-1)+1)}{6} \quad (124)$$

gilt, können wir für die 0. Ebene

$$O\left(\left(\frac{b}{d}\right)^5 \frac{2d^3 - 3d^2 + d}{6}\right) = O\left(b^5 \left(\frac{1}{3d^2} - \frac{1}{2d^3} + \frac{1}{6d^5}\right)\right) = O\left(\frac{b^5}{d^2}\right) \quad (125)$$

schreiben.

In der nächsten Ebene gibt es dann d Probleme mit jeweils höchstens $\frac{b}{d}$ Knoten. Für alle Ebenen gilt also

$$O\left(\frac{b^5}{d^2} + d\frac{\left(\frac{b}{d}\right)^5}{d^2} + d^2\frac{\left(\frac{b}{d^2}\right)^5}{d^2} + \cdots + d^{h-1}\frac{\left(\frac{b}{d^{h-1}}\right)^5}{d^2} + n\right). \quad (126)$$

(Die letzte Ebene wird als Spezialfall in Algorithmus 4.4 betrachtet und kann in $O(n)$ Zeit behandelt werden.) Da aber

$$d^h \frac{\left(\frac{b}{d^h}\right)^5}{d} = b \frac{\left(\frac{b}{b}\right)^5}{d} = \frac{b}{d} \quad (127)$$

gilt, erhalten wir

$$O\left(\frac{b^5}{d^2} + d\frac{\left(\frac{b}{d}\right)^5}{d^2} + d^2\frac{\left(\frac{b}{d^2}\right)^5}{d^2} + \cdots + d^h\frac{\left(\frac{b}{d^h}\right)^5}{d^2} - \frac{b}{d} + n\right). \quad (128)$$

Durch weitere Umformungen erhalten wir nacheinander:

$$O\left(\frac{b^5}{d^2}\left(1 + d\left(\frac{1}{d}\right)^5 + d^2\left(\frac{1}{d^2}\right)^5 + \dots + d^h\left(\frac{1}{d^h}\right)^5\right) - \frac{b}{d} + n\right) = \quad (129)$$

$$O\left(\frac{b^5}{d^2}\left(1 + \frac{1}{(d^4)^1} + \frac{1}{(d^4)^2} + \dots + \frac{1}{(d^4)^h}\right) - \frac{b}{d} + n\right). \quad (130)$$

Nun können wir die Formel für die Partialsumme einer geometrischen Folge anwenden:

$$1 + \frac{1}{(d^4)^1} + \frac{1}{(d^4)^2} + \dots + \frac{1}{(d^4)^h} = \frac{\left(\frac{1}{d^4}\right)^h - 1}{\frac{1}{d^4} - 1} = \frac{\frac{1}{b^4} - 1}{\frac{1}{d^4} - 1} = \frac{(b^4 - 1)d^4}{b^4(d^4 - 1)}. \quad (131)$$

Und eingesetzt in (130):

$$O\left(\frac{b^5}{d^2} \frac{(b^4 - 1)d^4}{b^4(d^4 - 1)} - \frac{b}{d} + n\right) = O\left(\frac{b^5}{d^2} + n\right). \quad (132)$$

Nun ist wichtig zu bemerken, dass die Anzahl der Blätter in dem Arrangement Baum höchstens nd ist ($b < nd$), woraus wir

$$O\left(\frac{(nd)^5}{d^2} + n\right) = O(n^5 d^3) \quad (133)$$

erhalten. □

Hier sollten wir auch erwähnen, dass wir eine sehr grobe Abschätzung gemacht haben: Z.B. wird sich die *repeat*-Schleife im Alg 4.5 kaum so oft wiederholen, da es dafür u.a. einen zulässigen Schnitt mit dem Zielfunktionswert 0 geben müsste. Auch die inneren Schleifen werden am meisten nicht bis zum letzten Knoten durchgelaufen - die reale Laufzeit, die man bei realen Instanzen erwarten kann, wird also deutlich besser sein. Die Ergebnisse dieses Algorithmus (siehe in Tabelle A.2) sind aber so gut, dass wir auch ein wenig schlechtere, aber stets polynomielle Laufzeit in Kauf nehmen können.

Zum Schluss möchten wir noch anhand eines Beispiels illustrieren, dass Algorithmus 4.4 kein Optimum finden muss, auch dann, wenn wir das *Min-SPBM* exakt lösen könnten.

Beispiel 4.9. *Betrachten wir den Graphen aus Abb. 4.13 und des Arrangement-Baumes mit $d = 2$. Dann gilt $h = \lceil \log_d n \rceil = \lceil \log_2 7 \rceil = 3$. Die Instanz entspricht der Testinstanz *KE_duenner7* mit der Wahl $d = 2$ (siehe die ausführlichere Beschreibung der Testinstanzen in Abschnitt 6.1).*

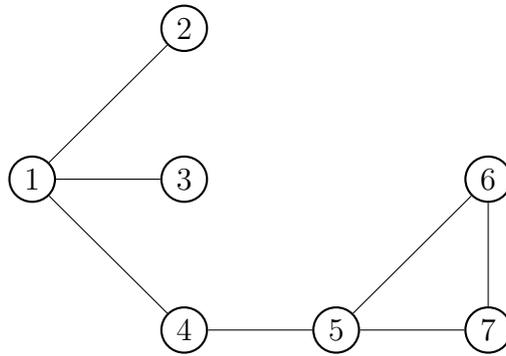


Abbildung 4.13: Graph

Im ersten Schritt müssen wir einen minimalen Schnitt X mit $3 \leq |X| \leq 4$ finden. Da der Graph zusammenhängend ist, muss der Schnitt mindestens eine Kante überschneiden – wir können also $X = \{1, 2, 3, 4\}$ ($V(G \setminus X) = \{5, 6, 7\}$) mit der Sicherheit wählen, dass wir einen optimalen Schnitt gefunden haben. Wenn wir den induzierten Teilgraphen $G[X]$ in zwei Hälften teilen wollen, überschneiden wir immer 2 Kanten, es ist also egal, welchen Schnitt wir wählen. Genau dieselbe Situation entsteht für den induzierten Teilgraphen $G[V(G) \setminus X]$. Das Arrangement, das wir also mithilfe der heuristischen Methode der minimalen beschränkten Schnitte finden (und wir haben dabei das MinSPBM immer exakt gelöst), sieht folgendermaßen aus:

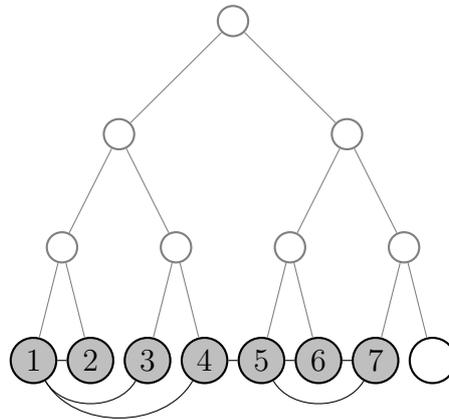


Abbildung 4.14: Lösung für $d = 2$

Diese Lösung hat den Zielfunktionswert gleich $z(G, d, \phi) = 26$. Die Optimallösung hat aber den Zielfunktionswert $z(G, d, \phi_{\text{Optimum}} = 24)$ und sieht z.B. folgendermaßen aus:

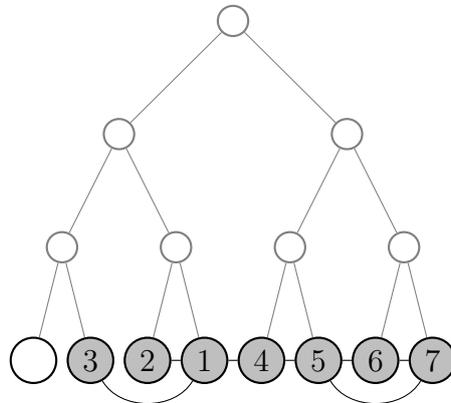


Abbildung 4.15: *Optimallösung für $d = 2$*

Der „Fehler“ im Bsp. 4.9 wurde durch eine „falsche“ Wahl eines minimalen Schnittes bei der ersten Teilung getan. Wir könnten aber auch Beispiele konstruieren, die zwar einen eindeutigen minimalen beschränkten Schnitt haben, derer Optimallösung aber einen anderen nehmen muss. Die Minimalitätsbedingung für die minimalen beschränkten Schnitte ist also weder notwendig, noch hinreichend.

4.3 Lokale Suche

Nachdem wir in Abschnitt 4.2 eine Heuristik unterusucht haben, die an der Grenze zu der *lokalen Suche* steht, werden wir uns nun mit diesem Gebiet näher beschäftigen. Die Hauptidee der lokalen Suchverfahren ist, mit lokalen Veränderungen einen Gewinn an dem Zielfunktionswert zu erzielen.

4.3.1 Austausch-Heuristik

Diese Heuristik ist mit der *heuristischen Methode der minimalen Schnitte in beschränkten Mengen* (siehe Abschnitt 4.2.3) eng verwandt. Allerdings erweist sich die Qualität der von diesem Verfahren gefundenen Lösungen um eine Spur schlechter bei derselben, oder vergleichbaren Zeitkomplexität. Auf der anderen Seite ist diese Methode leichter zu implementieren.

Die Hauptidee ist, iterativ Paare von Knoten zu finden, sodass das Vertauschen der ihnen zugewiesenen Blätter eine Verbesserung des Zielfunktionswertes impliziert. Es ist aber natürlich wichtig, auch die unbesetzten Blätter behandeln zu können und deswegen definieren wir einen Graphen $G^k = (V^k, E^k)$, wobei $V^k = V \cup \{v_{n+1}, \dots, v_b\}$ und $E^k = E$. (Dabei nehmen wir an, dass $V = \{v_1, \dots, v_n\}$.) Dann ist $z(G, d, \phi) = z(G^k, d, \phi^k)$ für jedes

Arrangement $\phi^k : V^k \rightarrow B$, wenn $\phi(v) = \phi^k(v_k)$ für alle $v_k \in V$ gilt und B die Menge der Blätter des Arrangement-Baumes T ist.

Input: $G = (V, E)$ ungerichteter Graph, wobei $V = \{v_1, \dots, v_n\}$; $d \geq 2$; sei weiter $|V| = n$

Output: Arrangement ϕ

- 1: $h := \lceil \log_d n \rceil$ und $b := h^d$;
- 2: Erstelle $G^k = (V^k, E^k)$, sodass $V^k = V \cup \{v_{n+1}, \dots, v_b\}$ und $E^k = E$;
- 3: Erstelle ϕ^k von G^k auf T zufällig;
- 4: **repeat**
- 5: **for** $i := 1$ **to** $b - 1$ **do**
- 6: **for** $j := i + 1$ **to** b **do**
- 7: Finde die Knoten $u_1^k, u_2^k \in V^k$, sodass $\phi^k(u_1^k) = b_i$ und $\phi^k(u_2^k) = b_j$ gilt;
- 8: Bilde ein neues Arrangement $\phi_{neu}^k(v^k) = \begin{cases} b_j & \text{für } v^k = u_1^k, \\ b_i & \text{für } v^k = u_2^k, \\ \phi^k(v^k) & \text{sonst} \end{cases}$
- 9: **if** $z(G^k, d, \phi_{neu}^k) < z(G^k, d, \phi^k)$ **then**
- 10: $\phi^k := \phi_{neu}^k$;
- 11: **end if**
- 12: **end for**
- 13: **end for**
- 14: **until** Kein besseres Arrangement wurde gefunden
- 15: Erstelle $\phi : V \rightarrow B$ so, dass $\phi(v) = \phi^k(v) \forall v \in V$;
- 16: **return** ϕ ;

Algorithmus 4.6: Austausch-Heuristik – Algorithmus 1

Bemerkung 4.7. *Algorithmus 4.6 lässt sich in $O(n^2 d^2 m^2 (\log n)^2)$ implementieren.*

Beweis: Aus Bemerkungen 3.1 und 3.2 ist es klar, dass es höchstens zu $O(2hm - 2m) = O(hm) = O((\log b)m) = O(\log(nd)m)$ Verbesserungen kommen kann. Die Länge des Weges $d_T(b_i, b_j)$ zwischen zwei Blättern b_i und b_j zu finden, dauert $O(2h) = O(\log b)$ Zeit. Um einen Zielfunktionswert zu berechnen, brauchen wir also $O((\log b)m) = O((\log(nd))m)$ Zeit. Dies wird höchstens $O(b^2) = O(n^2 d^2)$ geschehen und zusammengefasst erhalten wir die Gesamtlaufzeit

$$O(n^2 d^2 m^2 (\log nd)^2) = O(n^2 d^2 m^2 (\log n + \log d)^2) = O(n^2 d^2 m^2 (\log n)^2). \quad (134)$$

□

Inwieweit eine Lösung vom Optimum entfernt ist, hängt natürlich stark von der Wahl des zufälligen Arrangements ϕ am Anfang. Auch wenn wir die Bedingung in der Kode-Zeile 9 in Algorithmus 4.6 als $z(G^k, d, \phi_{neu}^k) \leq z(G^k, d, \phi^k)$ definieren würden und bei Gleichheit einen Austausch mit einer vordefinierten Wahrscheinlichkeit machen würden, gibt es Instanzen und Arrangements ϕ , die die obige Heuristik nicht löst.

Ein einfacher Weg, die Qualität der von Algorithmus uns ermittelten Arrangements zu verbessern, wäre es, alle Tripel von Knoten, dann alle Quadrupel von Knoten usw. zu permutieren. Dabei ist die Verbesserung des Zielfunktionswertes relativ gering im Vergleich zu dem erheblichen Wachstum der Laufzeit (immer um eine Potenz von n und d – d.h. für alle Tripel würden wir $O(n^3 d^3 m^2 (\log n)^2)$, für alle Quadrupel $O(n^4 d^4 m^2 (\log n)^2)$ usw.).

Beispiel 4.10. *Betrachten wir wieder den Graphen aus Abb. 1.1 in Beispiel 1.1 und des Arrangement-Baume mit $d = 3$ (wie in Beispiel 1.2). Dann gilt $h = \lceil \log_d n \rceil = \lceil \log_3 5 \rceil = 2$. Wenn wir am Anfang für das Arrangement ϕ das normale Arrangement (siehe Definition 4.1) verwenden, so fangen wir mit dem Zielfunktionswert $z(G, d, \phi) = 22$ an.*

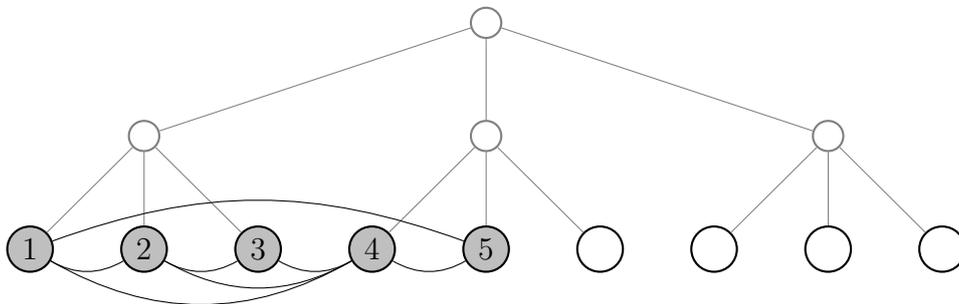


Abbildung 4.16: Normales Arrangement für $d = 3$

Die erste Möglichkeit, durch den Austausch eines Knotenpaares einen Gewinn zu erzielen, ist für $i = 1$ und $j = 4$; wir erhalten das Arrangement in Abb. 4.17.

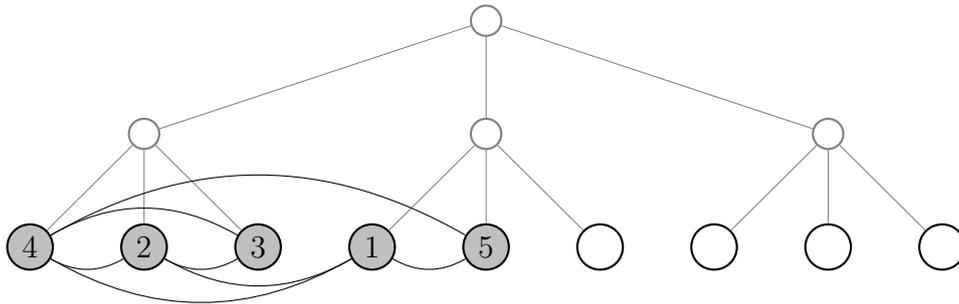


Abbildung 4.17: Lösung für $d = 3$

Diese Lösung können wir nicht mehr verbessern (es handelt sich sogar um eine Optimallösung). Der Zielfunktionswert ist $z(G, d, \phi) = 20$.

In Beispiel 4.10 haben wir eine typische Anwendungsart dieser Heuristik gesehen – wir können bereits bekannte Lösungen verbessern. Sehr gut eignet sich dieser Algorithmus vor allem dazu, eine Lösung einer *randomisierten Heuristik* (siehe Kapitel 4.1) weiter zu verbessern.

Wie wir schon erwähnt haben, ist die Laufzeitkomplexität dieses Algorithmus relativ schlecht – eine interessante und relativ einfache Möglichkeit, sie zu verbessern, ist folgende: Wir starten mit einer *zusammenhängenden Lösung*, die z.B. mithilfe des *Normales-Arrangement-Verfahrens* (siehe Abschnitt 4.1.1), oder mittels der *Methode der zufälligen zusammenhängenden Arrangements* (siehe Abschnitt 4.1.3) generiert wird. Nun vertauschen wir keine Paare in der Blättermenge, sondern wirkliche Knotenpaare v_i und v_j tauschen wir die Blätter $\phi(v_i)$ und $\phi(v_j)$ aus.

Bemerkung 4.8. *Algorithmus 4.7 lässt sich in $O(n^2 m^2 (\log n)^2)$ implementieren.*

Beweis: Aus Bemerkungen 3.1 und 3.2 ist es klar, dass es höchstens zu $O(2hm - 2m) = O(hm) = O((\log b)m) = O(\log(nd)m)$ Verbesserungen kommen kann. Die Länge des Weges $d_T(b_i, b_j)$ zwischen zwei Blättern b_i und b_j zu finden, dauert $O(2h) = O(\log b)$ Zeit. Um einen Zielfunktionswert zu berechnen, brauchen wir also $O((\log b)m) = O((\log(nd))m)$ Zeit. Dies wird höchstens $O(n^2)$ geschehen und zusammengefasst erhalten wir die Gesamtlaufzeit

$$O(n^2 m^2 (\log nd)^2) = O(n^2 m^2 (\log n + \log d)^2) = O(n^2 m^2 (\log n)^2). \quad (135)$$

□

Input: $G = (V, E)$ ungerichteter Graph, wobei $V = \{v_1, \dots, v_n\}$; $d \geq 2$; sei weiter $|V| = n$

Output: Arrangement ϕ

- 1: Erstelle zusammenhängendes Arrangement ϕ zufällig;
- 2: **if** $n = 1$ **then**
- 3: **return** ϕ ;
- 4: **end if**
- 5: **repeat**
- 6: **for** $i := 1$ **to** $n - 1$ **do**
- 7: **for** $j := i + 1$ **to** n **do**
- 8: Bilde ein neues Arrangement $\phi_{neu}(v) = \begin{cases} \phi(v_j) & \text{für } v = v_i, \\ \phi(v_i) & \text{für } v = v_j, \\ \phi(v) & \text{sonst} \end{cases}$
- 9: **if** $z(G, d, \phi_{neu}) < z(G, d, \phi)$ **then**
- 10: $\phi := \phi_{neu}$;
- 11: **end if**
- 12: **end for**
- 13: **end for**
- 14: **until** Kein besseres Arrangement wurde gefunden
- 15: **return** ϕ ;

Algorithmus 4.7: Austausch-Heuristik – Algorithmus 2

Unser Laufzeitgewinn im Vergleich zu Algorithmus 4.6 hängt also von der konkreten *Ordnung des Arrangement-Baumes* d ab.

In Abschnitt 4.1.3 haben wir gesehen, dass es sich lohnt, sich nur mit *zusammenhängenden Lösungen* zu beschäftigen. Dasselbe gilt auch für die *Austausch-Heuristik*: empirisch gemessen, liefert Algorithmus 4.7 nur um eine Spur schlechtere Lösungen, als Algorithmus 4.6. Die Laufzeitverbesserung ist jedoch vor allem bei einer großen *Ordnung des Arrangement-Baumes* nicht gering im Vergleich zu n . Neben der *heuristischen Methode der minimalen Schnitte in beschränkten Mengen* (siehe Abschnitt 4.2.3) ist die *Austausch-Heuristik* empirisch das beste Verfahren, das in dieser Arbeit vorgestellt wird.

4.3.2 Shift-Flip-Heuristik

Als letzte Heuristik wollen wir noch die *Shift-Flip-Heuristik* vorstellen. Sie gehört zwar nicht zu den besten und auch ihre Zuordnung zu der *lokalen Suche* ist spekulativ, die Hauptidee dieses Verfahren ist aber problemspezifisch und zeigt schön einige Eigenschaften dieses Problems.

Die Operation *Flip* haben wir bereits definiert (siehe Definition 4.4). Es fehlt uns noch die *Shift*-Operation.

Definition 4.6 (Shift, Shift um k , shiften). Sei $G = (V, E)$ ein ungerichteter Graph mit $|V| = n$, $d \in \mathbb{N}$, $d \geq 2$, die Ordnung des Arrangement-Baumes T und $\phi : V \rightarrow B$ ein Arrangement von G auf T , wobei die Menge B aus Blättern des Arrangement-Baumes T besteht. Gegeben sei weiters $k \in \mathbb{N}$. Ein Arrangement $\phi_k : V \rightarrow B$, sodass

$$\phi_k(v) := b_{((i-1)+k) \bmod b+1}, \text{ wobei } \phi(v) = b_i \text{ und } b = |B|, \quad (136)$$

heißt ein **Shift** des Arrangements ϕ , oder ein **Shift** des Arrangements ϕ **um k** . Wir sagen auch, dass wir das Arrangement ϕ **shiften**.

Der eigentliche Sinn dieser Operation ist es, ein $0 \leq k \leq b - 1$ so zu finden, dass $z(G, d, \phi_k)$ minimiert wird.

Der Algorithmus startet mit einem zufälligen Arrangement ϕ . Dann wiederholen sich immer zwei Operationen:

1. Es wird ein $0 \leq k \leq b - 1$ so gefunden, dass $z(G, d, \phi_k)$ minimiert wird. Für weitere Berechnungen wird dieses Arrangement benutzt ($\phi := \phi_k$). Wenn es mehrere solche k gibt, die den Zielfunktionswert minimieren, dann wird aus diesen ein zufälliges k gewählt.
2. Es wird ein zufälliges *Flip* durchgeführt.

Die Operation *Shift* verringert den Zielfunktionswert und die Operation *Flip* ermöglicht, dass die Knoten permutiert werden, ohne den Zielfunktionswert zu verändern (siehe Bemerkung 4.2).

Eine andere Variante dieses Algorithmus wäre, ein $k \geq 1$ nur dann zu wählen, wenn diese Wahl eine Verbesserung des Zielfunktionswertes bringen würde (d.h. wenn der Zielfunktionswert nicht verbessert werden kann, würde die Operation *Shift* mit $k = 0$ durchgeführt werden). Diese Variante verkleinert aber die Möglichkeit, viele unterschiedliche Arrangements zu generieren und empirisch gesehen ist sie schlechter, als die ursprüngliche.

Als sinnvolles Terminierungskriterium könnte z.B. eine obere Schranke für die Anzahl der *Shift-Flip-Zyklen*, die keine Verbesserung bringen und direkt nacheinander folgen, dienen. Alternativ kann auch die Laufzeit des Algorithmus beschränkt werden, bzw. können wir den Algorithmus nach einer gewissen vorher definierten Zeit, in der es zu keiner Verbesserung des Zielfunktionswertes kommt, beenden.

Eine mathematische Analyse dieses Verfahrens könnte z.B. den *Erwartungswert* der Anzahl der nötigen *Shift-Flip-Zyklen*, um eine Verbesserung

des Zielfunktionswertes zu erzielen, ermitteln. Diese Analyse würde aber den Rahmen dieser Arbeit sprengen. Die empirischen Ergebnisse dieser Methode sind vergleichsweise eher schlecht. Sie kann höchstens zum Nachiterieren, oder als Hilfsmethode in Kombination mit einem anderen Verfahren verwendet werden. Eine interessante Möglichkeit wäre z.B., abwechselnd die *Austausch-Heuristik* (siehe Abschnitt 4.3.1) und die *Shift-Flip-Heuristik* anzuwenden. Wenn eine dieser Methoden in einem *lokalen Minimum* stehen bleiben würde, könnte die andere gestartet werden. Vom Vorteil ist dabei, dass die Lösung in beiden Methoden sehr unterschiedlich verbessert wird und eine Methode könnte dabei der anderen helfen, den Weg aus einem *lokalen Minimum* zu finden. Die empirischen Ergebnisse dieser Kombination sind sogar vergleichbar mit der *heuristischen Methode der minimalen Schnitte in beschränkten Mengen* (siehe Abschnitt 4.2.3), allerdings unter einer wesentlich schlechteren Laufzeit.

5 Spezialfälle

Nachdem wir einige Algorithmen unteruscht haben, die an das *MinDAPBaum* angewendet werden können, werden wir uns nun einigen Spezialfällen widmen – vor allem aber denen, die in einigen Beweisen vertreten wurden. Es handelt sich also meistens um ganz spezielle Instanzen, die eher keine praktische Anwendungen in der angewandten Mathematik finden können, die aber einige Eigenschaften dieses Problems erläutern und einem besseren Verständnis des obigen Textes dienen.

5.1 Spezialfall: G vollständig

Bei solchen Instanzen werden wir noch 2 Typen unterscheiden.

5.1.1 G vollständig und $n = b$

Diese Instanzen sind trivial. Da es bei jedem Arrangement ϕ zu jedem Blatt b_i einen Knoten v_i gibt, sodass $\phi(v_i) = b_i$ gilt und da alle Knotenpaare v_i und v_j mit einer Kante (v_i, v_j) verbunden sind, werden alle Wegelängen zwischen allen Blätterpaaren in dem Zielfunktionswert eingerechnet. Aus Korollar 4.1 erhalten wir also

$$z(G, d, \phi) = b \left((h + 1)b - \frac{db - 1}{d - 1} \right) \text{ für jedes } \phi. \quad (137)$$

Diese Tatsache folgt auch aus Satz 4.1 und aus Bemerkung 4.3:

$$\begin{aligned}\mathbb{E}z(G, d, \phi) &= \frac{2m}{b-1} \left((h+1)b - \frac{bd-1}{d-1} \right) \\ &= b \left((h+1)b - \frac{db-1}{d-1} \right),\end{aligned}\tag{138}$$

$$\mathbb{V}z(G, d, \phi) = 0.\tag{139}$$

Da die Varianz des Zielfunktionswertes eines zufälligen Arrangements gleich 0 ist, gilt

$$\mathbb{E}z(G, d, \phi) = z(G, d, \phi) \text{ für jedes } \phi.\tag{140}$$

Die Gleichung (139) wird für diesen Spezialfall auch in Abb. 4.2 illustriert.

5.1.2 G vollständig und $n < b$

Bevor wir diesen Fall näher eingehen, brauchen wir noch eine Bemerkung, die direkt aus der Struktur des indexierten Arrangement-Baumes folgt.

Bemerkung 5.1. *Sei T ein indexierter Arrangement Baum T mit Blättern $\{b_1, \dots, b_b\} = B$. Dann gilt für alle i, j, k , wobei $1 \leq i < j < k \leq b := |B|$*

$$d_T(b_i, b_j) \leq d_T(b_i, b_k).\tag{141}$$

Betrachten wir nun unser Problem. Wir wissen, dass G vollständig ist. Wir müssen also nur eine Menge der Blätter $\tilde{B} = \{\tilde{b}_1, \dots, \tilde{b}_n\} \subset B$, wobei $|\tilde{B}| = |V|$ auswählen, an die wir unsere Knoten arrangieren. Offensichtlich gilt $z(G, d, \phi_1) = z(G, d, \phi_2)$ für alle ϕ_1 und ϕ_2 mit dem gleichen Wertebereich, $\{\phi_1(v_1), \dots, \phi_1(v_n)\} = \{\phi_2(v_1), \dots, \phi_2(v_n)\} = \tilde{B}$. Wir schreiben

$$z(G, d, \tilde{B}) := z(G, d, \phi_1) = z(G, d, \phi_2).\tag{142}$$

Da dann die Längen der Wege aller Blätterpaare b_1 und b_2 , wobei $b_1, b_2 \in \tilde{B}$ eingerechnet werden, folgt aus Bemerkung 5.1, dass es für jede oben definierte Menge $\tilde{B} \subset B$, die den Wert $z(G, d, \tilde{B})$ minimiert ein *zusammenhängendes* Arrangement $\phi_{\tilde{B}}$ gibt, sodass $\tilde{B} = \{\phi_{\tilde{B}}(v_1), \dots, \phi_{\tilde{B}}(v_n)\}$.

Um also eine Optimallösung zu finden, reicht es, das optimale Intervall $[k, l]$ mit $1 \leq k \leq l \leq b$ und $k - l = n$ zu finden und das Arrangement ϕ folgendermaßen zu definieren:

$$\phi(v_i) := b_{i+k}.\tag{143}$$

Es reicht also, nur den Zielfunktionswert des in (143) definierten Arrangements für alle $k = 1, \dots, b - n$ zu berechnen und das beste Arrangement daraus zu wählen.

Bemerkung 5.2. Sei $G = (V, E)$ ein vollständiger Graph, $d \geq 2$ die Ordnung des Arrangement-Baumes T und es gelte $n < b$, wobei b die Anzahl der Blätter eines Arrangement-Baumes T darstellt. Dann lässt sich eine Optimallösung des MinDAPBaum in $O(ndm \log n)$ berechnen.

Beweis: Um einen Zielfunktionswert zu berechnen, brauchen wir $O(2mh) = O(mh) = O(m \log b)$ Zeit. Diesen müssen wir $b - n$ -mal ausrechnen und brauchen also insgesamt $O((b - n)m \log b)$. Da aber $b < nd$ gilt, erhalten wir $O((nd - n)m \log nd) = O(ndm \log nd) = O(ndm(\log n + \log d)) = O(ndm \log n)$. \square

5.2 Spezialfall: $d = n \geq 2$

In diesem Fall sind offensichtlich alle Arrangements ϕ optimal, da die Länge des Weges zwischen allen Blätterpaaren gleich 2 ist. Für den Zielfunktionswert gilt dann $z(G, d, \phi) = 2m$. Dieser Wert stellt dabei auch eine allgemeine untere Schranke dar (siehe Bemerkung 3.2).

In Tabelle A.2 gibt zwei Instanzen, die diesem Spezialfall entsprechen: *KE_duenner7* mit $d = 7$ und *KE_dichter7* mit $d = 7$.

5.3 Spezialfall: $d = n - 1 \geq 2$

Da $d = n - 1$ gilt, gilt auch $d < n < d^2$ und es muss offensichtlich $h = 2$ sein. Daraus folgt auch $2 \leq d_T(b_i, b_j) \leq 4$ für alle Blätter b_i und b_j , wobei $b_i \neq b_j$. Dabei können wir höchstens $n - 1$ Knoten des Graphen G auf die Blätter eines Unterbaumes arrangieren. Wenn der Graph G also zusammenhängend ist, werden in dem Zielfunktionswert immer Summanden mit dem Wert 4 auftreten und derer Anzahl wollen wir minimieren.

Es reicht uns also, die Knoten des Graphen G in zwei Mengen so zu teilen ($\emptyset \neq X \subset V$ und $V \setminus X$), dass die Anzahl der Kanten $(u, v) \in E$, wobei $u \in X$ und $v \in V \setminus X$ minimiert wird, zu finden:

$$\min_{\emptyset \neq X \subset V} |\{(u, v) \in E \mid u \in X \text{ und } v \in V \setminus X\}| \quad (144)$$

Dafür gibt es aber effektive und gut untersuchte Algorithmen – siehe z.B. [KV08].

Wenn wir dann $X = \{v_{X,1}, \dots, v_{X,|X|}\}$ und $V \setminus X = \{v_{V \setminus X,1}, \dots, v_{V \setminus X,|V \setminus X|}\}$ schreiben, ist das optimale Arrangement folgendermaßen definiert:

$$\phi(v) = \begin{cases} b_i & \text{für } v = v_{X,i} \in X, \\ b_{\frac{b}{d}+j} & \text{für } v = v_{V \setminus X,j} \in V \setminus X. \end{cases} \quad (145)$$

Bemerkung 5.3. Sei $G = (V, E)$ ein Graph, wobei $|V| = n \geq 3$, $d \geq 2$ die Ordnung des Arrangement-Baumes T und gelte dabei $d = n - 1$. Dann lässt sich eine Optimallösung des MinDAPBaum in $O(mn + n^2 \log n)$ berechnen.

Beweis: Um einen minimalen Schnitt $\emptyset \neq X \subset V$ im Graphen $G = (V, E)$ bezüglich seiner Kapazität zu finden, brauchen wir $O(mn + n^2 \log n)$ Zeit (siehe z.B. [KV08]).

Das Arrangement kann dann in $O(n)$ Zeit erstellt werden. □

Den Lösungsansatz mögen wir noch anhand eines Beispielles veranschaulichen.

Beispiel 5.1. Betrachten wir wieder den Graphen aus Abb. 1.1 in Beispiel 1.1 und die Ordnung des Arrangement-Baumes $d = 5 - 1 = 4$.

In Tabelle 5.1 sehen wir alle möglichen Schnitte $\emptyset \neq X \subset V$.

X	Kapazität	X	Kapazität
{1}	3	{1, 2, 3}	4
{2}	3	{1, 2, 4}	4
{3}	2	{1, 2, 5}	4
{4}	4	{1, 3, 4}	5
{5}	2	{1, 3, 5}	5
{1, 2}	4	{1, 4, 5}	3
{1, 3}	5	{2, 3, 4}	3
{1, 4}	5	{2, 3, 5}	5
{1, 5}	3	{2, 4, 5}	5
{2, 3}	3	{3, 4, 5}	4
{2, 4}	5	{1, 2, 3, 4}	2
{2, 5}	5	{1, 2, 3, 5}	4
{3, 4}	4	{1, 2, 4, 5}	2
{3, 5}	4	{1, 3, 4, 5}	3
{4, 5}	4	{2, 3, 4, 5}	3

Tabelle 5.1: Alle Schnitte $\emptyset \neq X \subset V$ und derer Kapazitäten

Eine der Möglichkeiten, die sich als minimaler Schnitt bietet, ist die Menge $X = \{1, 2, 4, 5\}$. Diesen Schnitt können wir in Abb. 5.1 sehen – die X -Menge ist in Blau gezeichnet, in Rot wurden dann die Kanten, die im Schnitt liegen, bzw. der Schnitt selbst gekennzeichnet.

In Abb. 5.2 können wir dann die Optimallösung sehen. Für den Zielfunktionswert erhalten wir

$$z(G, d, \phi) = 18. \tag{146}$$

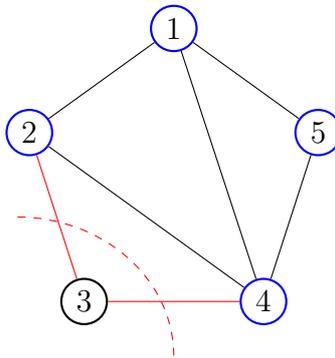


Abbildung 5.1: Minimaler Schnitt bezüglich seiner Kapazität im Graphen G ; die X -Menge ist in Blau gezeichnet, in Rot sind dann die Kanten, die im Schnitt liegen, bzw. der Schnitt selbst

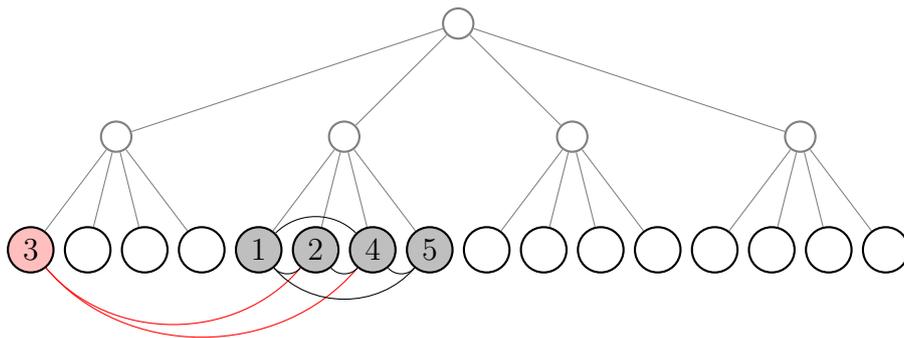


Abbildung 5.2: Optimallösung für $d = 4$; in Rot ist der Knoten mit einem minimalen Knotengrad und alle Kanten, die mit dem Wert 4 zu dem Ziel-funktionswert beitragen, gekennzeichnet

In Tabelle A.2 ist dieser Spezialfall durch den Graphen $SF_zufaellig50$ vertreten.

5.4 Spezialfall: Sterne

Definition 5.1 (Stern, Zentralknoten, Strahl). Ein ungerichteter zusammenhängender Graph $G = (V, E)$, der folgende eigenschaften erfüllt:

- (i) $\exists! v_z \in V : \Gamma(v_z) \geq 2$,
- (ii) $\forall v \in V \setminus \{v_z\} : \Gamma(v) = 1$,

heißt **Stern**. Der Knoten v_z heißt **Zentralknoten**, jeder anderer Knoten $v \in V \setminus \{v_z\}$ heißt **Strahl**.

Wir haben dieses Problem als Subproblem bereits im Abschnitt 3.1 lösen müssen.

Beschäftigen wir uns nun mit dem *Zentralknoten*. Nehmen wir an, dass wir das optimale Arrangement ϕ bereits kennen. Dann gibt es auch ein Blatt b_i , für das $\phi(v_z) = b_i$ gilt. Nun können wir offensichtlich immer durch mehrere nacheinander durchgeführten Flips ein Arrangement ϕ_f erstellen, sodass $\phi_f(v_z) = b_1$ ist. Dabei ändert sich der Zielfunktionswert des Arrangements nicht (siehe Bemerkung 4.2):

$$z(G, d, \phi) = z(G, d, \phi_f). \quad (147)$$

Wir können also o.B.d.A. annehmen, dass $\phi(v_z) = b_1$. Dann müssen wir nur mehr alle *Strahlen* arrangieren. Da alle die gleiche Rolle bezüglich des Zielfunktionswertes spielen, können wir jeden weiteren Strahl genau so platzieren, dass er den Zielfunktionswertzuwachs minimiert. Dabei können wir Bemerkung 5.1 verwenden und jedem Strahl ein weiteres Blatt zuweisen. Wenn wir o.B.d.A. annehmen, dass $v_z = v_1$, können wir

$$\phi(v_i) = b_i \quad (148)$$

definieren.

Den Zielfunktionswert des Arrangements ϕ können wir aus (19) ausdrücken:

$$z(G, d, \phi) = 2 \left(pn - \frac{d^p - 1}{d - 1} \right), \quad (149)$$

wobei

$$p = \lfloor \log_d n \rfloor + 1. \quad (150)$$

Satz 5.1. Sei $G = (V, E)$ ein Stern mit $|V| = n$, $V = \{v_1, \dots, v_n\}$ und $\Gamma(v_1) \geq 2$. Sei weiter $d \in \mathbb{N}$, $d \geq 2$ die Ordnung des Arrangement-Baumes T mit der Blättermenge $B = \{b_1, \dots, b_b\}$, wobei b die Kardinalität dieser Menge ist. Dann sieht eine Optimallösung des MinDAPBaum folgendermaßen gegeben

$$\phi(v_i) = b_i \text{ für alle } 1 \leq i \leq n \quad (151)$$

und hat den Zielfunktionswert

$$z(G, d, \phi) = 2 \left(pn - \frac{d^p - 1}{d - 1} \right), \quad (152)$$

wobei

$$p = \lfloor \log_d n \rfloor + 1. \quad (153)$$

Beweis: Siehe die Herleitung oben. □

In Tabelle A.2 ist dieser Spezialfall durch den Graphen *SF-Stern* mit der Wahl $d = 3$ vertreten.

5.5 Spezialfall: erweiterte Sterne

Definition 5.2 (Stern, Zentralknoten, Strahl, Blatt). Sei $G = (V, E)$ ein ungerichteter zusammenhängender azyklischer Graph und seien die Mengen V_1 , V_2 und V_z folgendermaßen definiert:

$$(i) \quad V_1 := \{v \in V \mid \Gamma(v) = 1\}$$

$$(ii) \quad V_2 := \{v \in V \mid \Gamma(v) = 2\}$$

$$(iii) \quad V_z := \{v \in V \mid \Gamma(v) \geq 3\}$$

Falls $|V_z| = 1$ gilt, dann heißt der Graph G **Erweiterter Stern**. Der Knoten $v_z \in V_z$ heißt **Zentralknoten**, jede Knotenmenge $S \subset V \setminus \{v_z\}$, für die gilt, dass $S \cup \{v_z\}$ einen v_z - v_1 -Weg, wobei $v_1 \in V_1$ bildet, heißt **Strahl** und jeder Knoten $v \in V_1$ heißt **Blatt**.

Wir werden uns nicht mit allgemeinen *erweiterten Sternen* beschäftigen, sondern nur mit solchen, die weitere Eigenschaften erfüllen und die im obigen Text Verwendung finden.

Sei nun $\tilde{S} := S_1 \cup S_2 \cup \dots \cup S_{|\tilde{S}|}$ die Menge der Strahlen und gelte dabei $S_1 = \arg \min_{S \in \tilde{S}} |S|$, dann werden wir weiter annehmen, dass

$$(i) \quad \lfloor \frac{d}{2} \rfloor \leq |S_1| \leq d - 1, \tag{154}$$

$$(ii) \quad \lfloor \frac{d}{2} \rfloor + 1 \leq |S| \leq d, \text{ wobei } S \in \tilde{S} \setminus \{S_1\} \text{ und} \tag{155}$$

$$(iii) \quad h = \lceil \log_d n \rceil = 2. \tag{156}$$

Diese Eigenschaften erfüllt z.B. der Graph in Abb. 2.2 für $d = 4$.

Da die Höhe $h = 2$, können die Summanden im Zielfunktionswert

$$z(G, d, \phi) = \sum_{(u,v) \in E} d_T(\phi(u), \phi(v)) \tag{157}$$

entweder den Wert 2, oder 4 annehmen (die Annahme (156)). Die Kantenanzahl m ist dabei konstant und wir können

$$z(G, d, \phi) = 4q + 2(m - q) = 2q + 2m \tag{158}$$

schreiben, wobei q die Anzahl der Blätterpaare, die mit dem Wert 4 zu dem Zielfunktionswert beitragen, ist. Wir wollen also die Anzahl der Kanten $(u, v) \in E$ minimieren, für die $d_T(\phi(u), \phi(v)) = 4$ gilt. Genauso, wie im Abschnitt 5.4 können wir o.B.d.A

$$\phi(v_z) = b_1 \quad (159)$$

festlegen. Nun werden wir ein Arrangement konstruieren, sodass $q = |\tilde{S}| - 1$ und auch zeigen, dass q keinen kleineren Wert annehmen kann.

Seien die Knoten in jedem *Strahl* S_i von dem Zentralknoten nummeriert:

$$S_i = \{s_{i,1}, \dots, s_{i,|S_i|}\}. \quad (160)$$

Und bilden wir das Arrangement ϕ folgendermaßen:

$$\begin{aligned} \phi(v_z) &= b_1, \\ \phi(s_{1,j}) &= b_{1+j} \text{ für } 1 \leq j \leq |S_1| \text{ und} \\ \phi(s_{i,j}) &= b_{d(i-1)+j} \text{ für } 2 \leq i \leq |\tilde{S}| \text{ und } 1 \leq j \leq |S_i|. \end{aligned} \quad (161)$$

Der *kürzeste Strahl* wird ganz auf dem linken Unterbaum mit dem *Zentralknoten* arrangiert und leistet also an dem Wert q keinen Beitrag. Die *anderen Strahlen* werden auch immer zur gänze auf einem Unterbaum arrangiert, nur die Verbindung zu dem *Zentralknoten* erhöht q um 1. Daraus erhalten wir $q = |\tilde{S}| - 1$ und aus (158) können wir auch

$$\begin{aligned} z(G, d, \phi) &= 4 \left(|\tilde{S}| - 1 \right) + 2 \left(m - |\tilde{S}| + 1 \right) \\ &= 2 \left(|\tilde{S}| + m - 1 \right) \end{aligned} \quad (162)$$

schreiben.

Wir müssen nun argumentieren, dass es kein besseres Arrangement bezüglich des Zielfunktionswertes gibt. Dadurch, dass wir 2 *Strahlen* und den *Zentralknoten* auf einen Unterbaum arrangieren würden, können wir den Zuwachs des Zielfunktionswertes nicht vermeiden, da eine solche Lösung die Annahmen (154) und (155) nicht ermöglichen. Um aber den Zielfunktionswert zu verkleinern, müsste es mehrere Strahlen geben, die q nicht erhöhen würden. Das kann aber nicht gemacht werden, da höchstens ein Strahl gemeinsam mit dem *Zentralknoten* arrangiert wird und bei allen anderen Strahlen trägt mindestens die Verbindung zu dem Rest des *erweiterten Sternes* mit dem Wert 1 zu q . Unsere Lösung ist also optimal.

Somit haben wir folgenden Satz bewiesen.

Satz 5.2. Sei $G = (V, E)$, $|V| = n$ und $|E| = m$, ein erweiterter Stern mit Strahlen $S_1, \dots, S_{|\tilde{S}|}$, $|S_1| \leq \dots \leq |S_{|\tilde{S}|}|$, wobei \tilde{S} die Menge der Strahlen ist und $S_i = \{s_{i,1}, \dots, s_{i,|S_i|}\}$, $1 \leq i \leq |\tilde{S}|$, gilt. Sei v_z der Zentralknoten und gegeben sei weiter die Ordnung des Arrangement-Baumes $d \in \mathbb{N}$, $d \geq 2$, und seien dazu folgende Bedingungen erfüllt:

$$(i) \lfloor \frac{d}{2} \rfloor \leq |S_1| \leq d - 1, \quad (163)$$

$$(ii) \lfloor \frac{d}{2} \rfloor + 1 \leq |S| \leq d, \text{ wobei } S \in \tilde{S} \setminus \{S_1\} \text{ und} \quad (164)$$

$$(iii) h = \lceil \log_d n \rceil = 2. \quad (165)$$

Dann ist eine Optimallösung des MinDAPBaum folgendermaßen gegeben

$$\phi(v) = \begin{cases} b_1 & \text{für } v_i = v_z \\ b_{1+j} & \text{für } v = s_{1,j} \in S_1 \text{ mit } 1 \leq j \leq |S_1| \\ b_{d(i-1)+j} & \text{für } v = s_{i,j} \in S_i \text{ mit } 2 \leq i \leq |\tilde{S}| \text{ und } 1 \leq j \leq |S_i| \end{cases} \quad (166)$$

und hat den Zielfunktionswert

$$z(G, d, \phi) = 2 \left(|\tilde{S}| + m - 1 \right). \quad (167)$$

Beweis: Siehe die Herleitung oben. □

Beispiel 5.2. Betrachten wir nun den Graphen aus Abb. 2.2 und die Ordnung des Arrangement-Baumes $d = 4$.

Die Lösung kann einfach mithilfe des Satzes 5.2 gebildet werden und wir können sie in Abb. 2.3 sehen.

In Tabelle A.2 ist dieser Spezialfall durch den Graphen *SF_erwStern* mit der Wahl $d = 4$ vertreten.

6 Numerische Ergebnisse

6.1 Kurze Beschreibung benutzter Testinstanzen

Es gibt leider keine Bibliothek mit Standardinstanzen für dieses Problem (so, wie es z.B. für das *TSP* der Fall ist). Um aber doch gewissermaßen standardisierte Testinstanzen zu betrachten, wurden bei allen Tests die Testinstanzen aus [Pet03] verwendet. Diese Graphen sind meistens sehr dünn besetzt, wofür

es bei dem *Linear-Arrangement-Problem* gute Gründe gibt. Für die Analyse der Heuristiken und anderer Ergebnisse für das *MinDAPBaum* werden allgemeinere Testdaten gebraucht.

Die in dieser Arbeit benutzte Testinstanzen sind in mehrere Gruppen unterteilt:

1. Kleine Instanzen, die man komplett enumerieren kann. (Namen der Graphen in Form: *KE_* + Dichtangabe + Anzahl der Knoten.)
2. Instanzen mit zufälligen Graphen, in denen zwei Knoten mit einer vorgegebenen Wahrscheinlichkeit durch eine Kante verbunden werden. (Namen der Graphen in Form: *KW_zufaellig* + Buchstabe + Wahrscheinlichkeit, dass eine Kante existiert.)
3. Spezielle Instanzen, zu denen man eine Optimallösung in polynomieller Zeit konstruieren kann. (Namen der Graphen in Form: *SF_* + Charakteristik.)
4. Instanzen mit Graphen, die in [Pet03] vorgestellt wurden. Unter diesen Graphen befinden sich auch mehrere, die aus der Praxis genommen wurden (wie etwa Verbindungsschema eines Computer-Netzes) und zwei, die im Vergleich zu anderen relativ groß sind (*Pet03_big* und *Pet03_wave*). (Namen der Graphen wurden von [Pet03] übernommen und tragen das Präfix *Pet03_*.)

6.2 Kurze Beschreibung numerischer Ergebnisse

Schauen wir nun kurz die Ergebnisse der durchgeführten Tests an. Sie sind in Tabelle A.2 zusammengefasst.

Wenn wir zuerst mit der am Anfang vorgestellten *Knotengrad-Schranke* (Abschnitt 3.1) beginnen, sehen wir, dass ihre Werte vor allem bei den kleinen Testinstanzen, zu denen wir eine Optimallösung kennen, nur relativ wenig vom Optimum entfernt sind. Natürlich ist schwer, diese Ergebnisse zu beurteilen, da wir im Allgemeinen nicht wissen, wie gut die von den vorgestellten Heuristiken gefundenen Lösungen sind. Aber auch bei der Testinstanz *Pet03_bintree10* ist die Spanne zwischen der *Knotengrad-Schranke* und dem Zielfunktionswert der besten gefundenen Lösung relativ klein. Hier können wir eher die Güte der Heuristik vermuten, die diese Lösung geliefert hat (*Tiefensuche-basiertes Greedy-Verfahren* (Abschnitt 4.2.2)), da dieser Zielfunktionswert deutlich besser ist, als die Zielfunktionswerte der Lösungen, die von anderen Heuristiken stammen.

Dann haben wir uns den *randomisierten Heuristiken* (Abschnitt 4.1) gewidmet. Hier können wir feststellen, dass die *Methode der zufälligen zusammenhängenden Arrangements* (Abschnitt 4.1.3) eindeutig bessere Lösungen als die anderen in Abschnitt 4.1 vorgestellten Heuristiken gefunden hat. Sogar auch bei der Testinstanz *SF_ervStern*, bei der wir wissen, dass sie keine zusammenhängende Optimallösung besitzt (siehe Beweis zu Bemerkung 2.2), war diese Heuristik besser als die *Methode der zufälligen Arrangements* (Abschnitt 4.1.2).

Wenn wir den *Erwartungswert des Zielfunktionswertes eines zufälligen Arrangements* (Abschnitt 4.1.4) anschauen, sehen wir, dass die Werte mit den Ergebnissen der *Methode der zufälligen Graphen* (Abschnitt 4.1.2) übereinstimmen. Die *Varianz des Zielfunktionswertes eines zufälligen Arrangements für einen Graphen mit einer festen Dichte* (Abschnitt 4.1.5) gibt uns dann eine gute Vorstellung über die Verteilung der Zielfunktionswerte.

Dann haben wir die *Greedy-Heuristik* (Abschnitt 4.2.1) getestet. Wir sehen, dass sie sich im Vergleich zu anderen getesteten Verfahren eher schlecht verhält.

Alle vorgestellten *Breitensuche- und Tiefensuche- basierten Greedy-Verfahren* (Abschnitt 4.2.2) haben untereinander vergleichbare Ergebnisse, allerdings erweist sich das *Tiefensuche- basierte Greedy-Verfahren* um eine Spur besser; deutlich besser ist es bei der Testinstanz *Pet03_bintree*, bei der wir sogar vermuten, dass diese Heuristik eine Optimallösung geliefert hat.

Die *Heuristische Methode der minimalen Schnitte in beschränkten Mengen* (Abschnitt 4.2.3) ist eindeutig das beste vorgestellte Verfahren bezüglich des Zielfunktionswertes. Es hat zwar eine schlechtere Zeitkomplexität, als die getesteten *Breitensuche- und Tiefensuche- basierten Greedy-Verfahren* (Abschnitt 4.2.2), die Zielfunktionswerte sind aber deutlich besser.

Zum Schluss haben wir noch die *Austausch-Heuristik* (Abschnitt 4.3.1) und die *Shift-Flip-Heuristik* (Abschnitt 4.3.2) getestet. Beide Algorithmen haben sich als eher langsam erwiesen, wobei die *Austausch-Heuristik* relativ gute Lösungen geliefert hat. Die *Shift-Flip-Heuristik* war empirisch gesehen das schlechteste vorgestellte Verfahren.

7 Kurze Zusammenfassung erreichter Ergebnisse

In den ersten zwei Kapiteln (Kapitel 1 und Kapitel 2) haben wir das Problem eingeführt und einige seiner Grundeigenschaften untersucht. Dabei haben wir das Problem hinsichtlich seiner Komplexität eingestuft und unter anderem

auch gezeigt, dass wir o.B.d.A nicht annehmen können, dass der zugrunde liegende Graph zusammenhängend ist.

Im weiteren Kapitel (Kapitel 3) haben wir eine untere Schranke definiert. Da wir nur eine untere Schranke gefunden haben, ist es schwer, ihre Güte zu beurteilen; wir können sie nur mit von den vorgestellten Heuristiken erzeugten Lösungen (siehe Kapitel 4) vergleichen. Obwohl die Spanne zwischen der unteren Schranke und den Lösungen der Heuristiken im Allgemeinen nicht gering ist, können wir sicher behaupten, dass wir eine sinnvolle untere Schranke gefunden haben; diese Vermutung wird vor allem anhand der Testinstanzen, bei denen wir die Optimallösung kennen (Instanzen *KE_duenner7*, oder die Instanz *KE_erwStern*) bestätigt.

Weiter haben wir die *randomisierten Heuristiken* (siehe Abschnitt 4.1) analysiert. Die geschlossenen Formeln für den *Erwartungswert* (Abschnitt 4.1.4) und für die Varianz des Zielfunktionswertes eines zufälligen Arrangements (Abschnitt 4.1.5) zählen zu den wichtigsten Resultaten dieser Arbeit. Wir haben geschlossene Formeln gefunden, die eine Anweisung geben, wann es sich lohnt, die *Methode der zufälligen Arrangements* (siehe Abschnitt 4.1.2) zu verwenden. Die theoretischen Ergebnisse stimmen auch mit den empirischen Ergebnissen der *Methode der zufälligen Arrangements* überein (siehe Kapitel 6).

In weiteren zwei Abschnitten (*Greedy- und andere einfache Heuristiken* (Abschnitt 4.2) und lokale Suche (Abschnitt 4.3) haben wir praktisch orientierte Algorithmen vorgestellt. Zuerst war es die *Greedy-Heuristik* (Abschnitt 4.2.1). Dann waren es die *Breitensuche- und Tiefensuche- basierten Greedy-Verfahren* (Abschnitt 4.2.2) mit mehreren einfach implementierbaren Algorithmen. Diese Verfahren haben sich für einige Testinstanzen als sehr gut erwiesen (z.B. für die Testinstanz *Pet03_bintree10*, derer Graph einem vollständigen Binärbaum entspricht).

Dann haben wir die *heuristische Methode der minimalen Schnitte in beschränkten Mengen* (siehe Abschnitt 4.2.3) vorgestellt. Sie hat sich im Vergleich zu anderen gezeigten Methoden als hervorragend erwiesen. Bei fast allen Testinstanzen hat sie die beste gefundene Lösung geliefert, die manchmal deutlich besser als die Lösungen aller anderen behandelten Verfahren ist. Dabei bietet diese Methode die Möglichkeit, verbessert zu sein, da sie ein *NP-schweres* Problem heuristisch als Subroutine lösen muss; die Güte dieser Methode hängt von der Güte der in der Subroutine gefundenen Lösungen ab. Vergleichbar war dieses Verfahren nur mit der im nächsten Abschnitt vorgestellten *Austausch-Heuristik* (siehe Abschnitt 4.3.1), bei der wir aber beim Testen am meisten eine schlechtere Laufzeit gemessen haben.

Bei allen an die Anwendung orientierten Algorithmen wurde eine Beschreibung in Form eines Pseudocode erstellt. Wie bereits geschrieben, emp-

fehlen können wir vor allem die *heuristische Methode der minimalen Schnitte in beschränkten Mengen* (siehe Abschnitt 4.2.3).

Weiter haben wir kurz einige eher offensichtliche *Spezialfälle* behandelt (siehe Kapitel 5). Sie sollen uns vor allem ermöglichen, die Güte der vorgestellten *unteren Schranke* und der beschriebenen *Heuristiken* besser empirisch bewerten zu können.

Zum Schluss wurden alle empirischen Ergebnisse kurz zusammengefasst, beschrieben und untereinander verglichen. Getestet wurden folgende Methoden, bzw. Ergebnisse: *Knotengrad-Schranke* (Abschnitt 3.1), *Normales-Arrangement-Verfahren* (Abschnitt 4.1.1), *Methode der zufälligen Arrangements* (Abschnitt 4.1.2), *Methode der zufälligen zusammenhängenden Arrangements* (Abschnitt 4.1.3), *Erwartungswert des Zielfunktionswertes eines zufälligen Arrangements* (Abschnitt 4.1.4), *Varianz des Zielfunktionswertes eines zufälligen Arrangements für einen Graphen mit einer festen Dichte* (Abschnitt 4.1.5), *Greedy-Heuristik* (Abschnitt 4.2.1), *Breitensuche-basiertes Greedy-Verfahren* (Abschnitt 4.2.2), *Tiefensuche-basiertes Greedy-Verfahren* (Abschnitt 4.2.2), *Heuristische Methode der minimalen Schnitte in beschränkten Mengen* (Abschnitt 4.2.3), *Austausch-Heuristik* (Abschnitt 4.3.1) und *Shift-Flip-Heuristik* (Abschnitt 4.3.2).

Wie bereits im Vorwort erwähnt, ist das Thema durch diese Arbeit keinesfalls erschöpft. Es gibt Bereiche, die überhaupt nicht berührt worden sind und die zu sicherlich interessanten Ergebnissen führen könnten – wir können z.B. die *Polyedertheorie* oder *Approximationsalgorithmen* nennen. Andererseits gibt es Bereiche, die nur am Rande dieser Arbeit erwähnt wurden: Etwa könnte man weitere Spezialfälle, die in polynomieller Laufzeit lösbar sind, untersuchen. Aus den Ergebnissen in Tabelle A.2 können wir z.B. vermuten, dass das *Tiefensuche-basiertes Greedy-Verfahren* (siehe Abschnitt 4.2.2) für vollständige Binärbäume eine Optimallösung findet. Diese Vermutung konnten wir im Rahmen dieser Arbeit leider nicht bestätigen.

Möge also diese Arbeit mindestens in den berührten Gebieten eine gute Grundlage für weitere Untersuchungen geschafft haben und einige praktisch anwendbare Algorithmen gezeigt haben, dann hat sie ihre Aufgabe erfüllt.

A Anhang: Abbildungen, Tabellen und Bemerkungen

A.1 Allgemeine Abbildungen und Tabellen

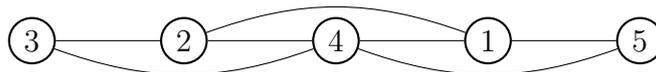


Abbildung A.1: *Optimale lineare Anordnung der Knoten aus dem Beispiel 1.1 (Zielfunktionswert ist 10)*

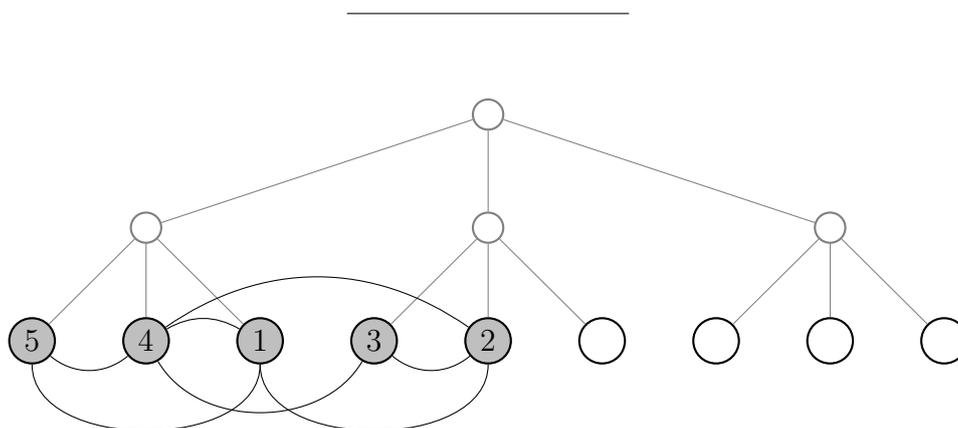
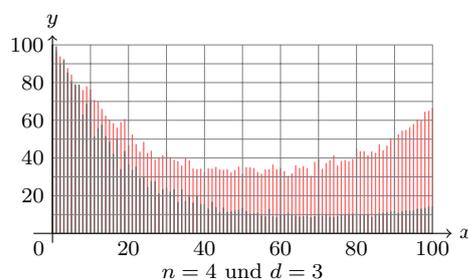
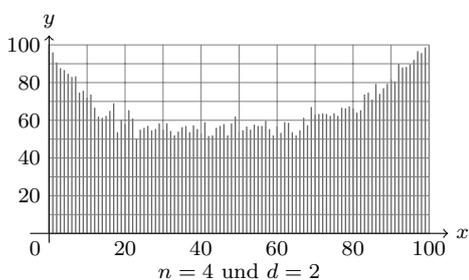


Abbildung A.2: *Optimales Arrangement aus dem Beispiel 1.2 (Zielfunktionswert ist 20)*



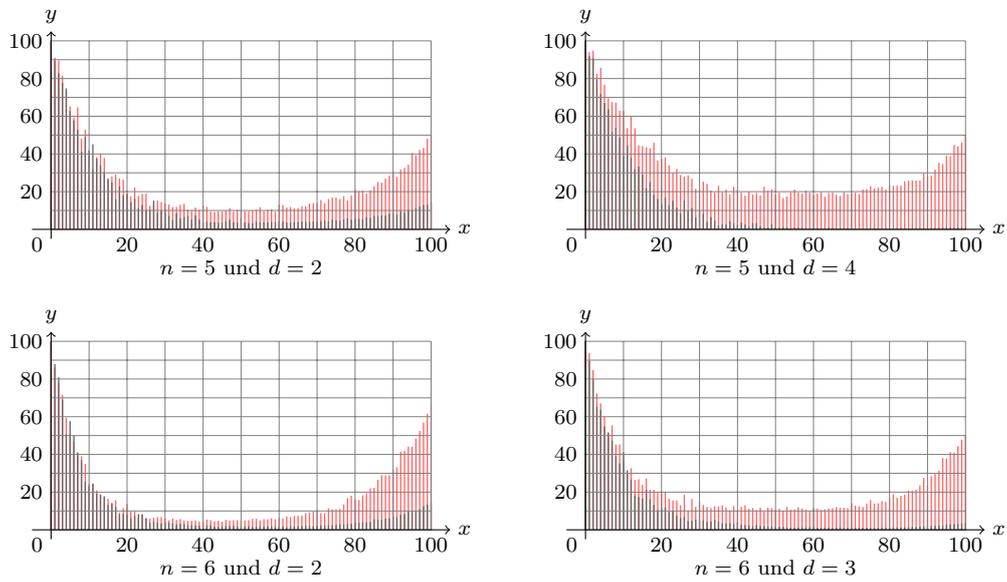
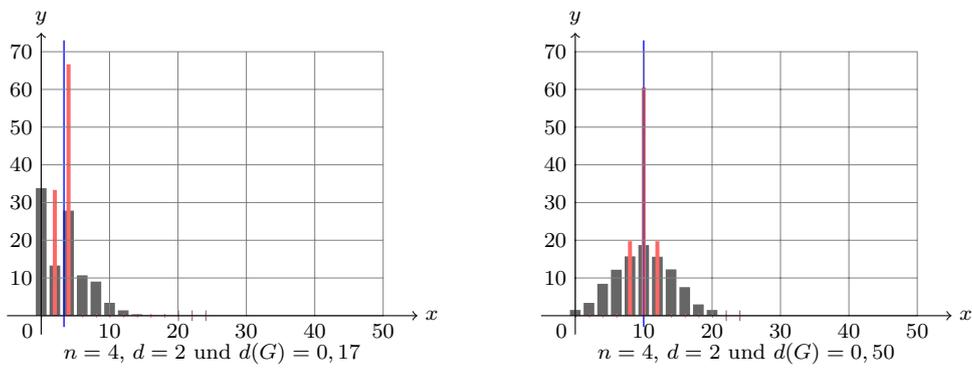


Abbildung A.3: Empirischer prozentueller Anteil der optimalen (in Grau) bzw. der optimalen zusammenhängenden (in Rot) Lösungen für unterschiedliche Werte von n und unterschiedliche Ordnungen der Arrangement-Bäume d auf $G(n, p)$; es wurden 100 Tests für jede Dichte durchgeführt



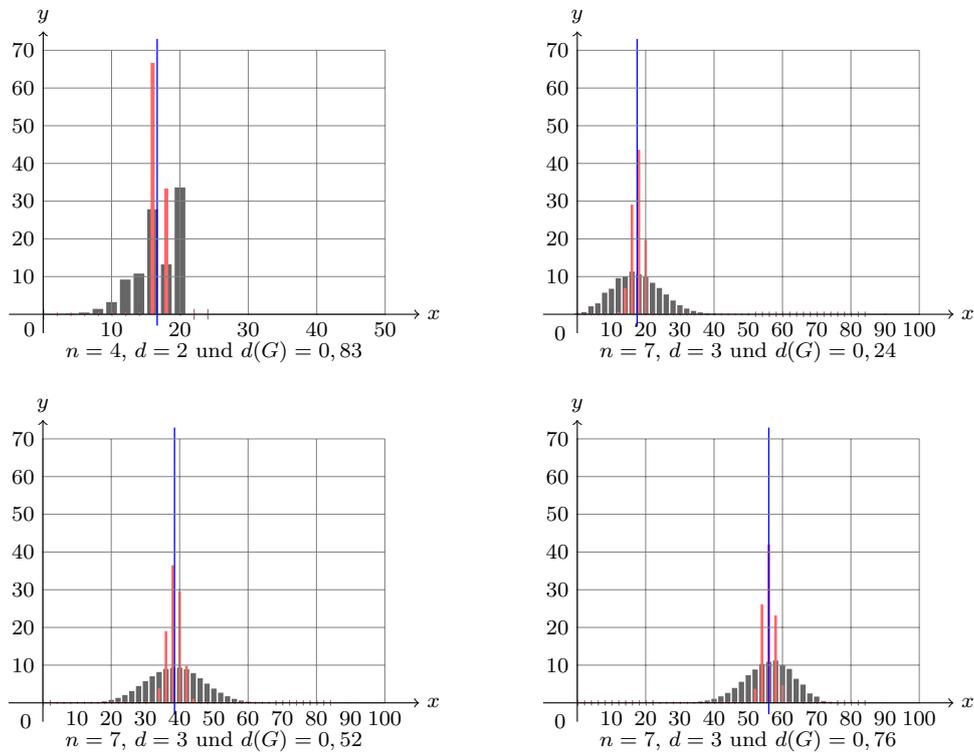


Abbildung A.4: Verteilung der Zielfunktionswerte für unterschiedliche Werte von n , unterschiedliche Ordnungen des Arrangement-Baumes d und unterschiedliche Dichten $d(G)$; auf der x -Achse befinden sich die Zielfunktionswerte und auf der y -Achse der prozentuelle Anteil an der Menge aller Arrangements

- in Grau: die empirische prozentuelle Verteilung der Zielfunktionswerte im arithmetischen Mittelwert über 10000 Zufallsgraphen in Grau
- in Rot: die empirische prozentuelle Verteilung der Zielfunktionswerte im arithmetischen Mittelwert über 10000 Zufallsgraphen mit fester Dichte der Graphen
- in Blau: der berechnete Erwartungswert

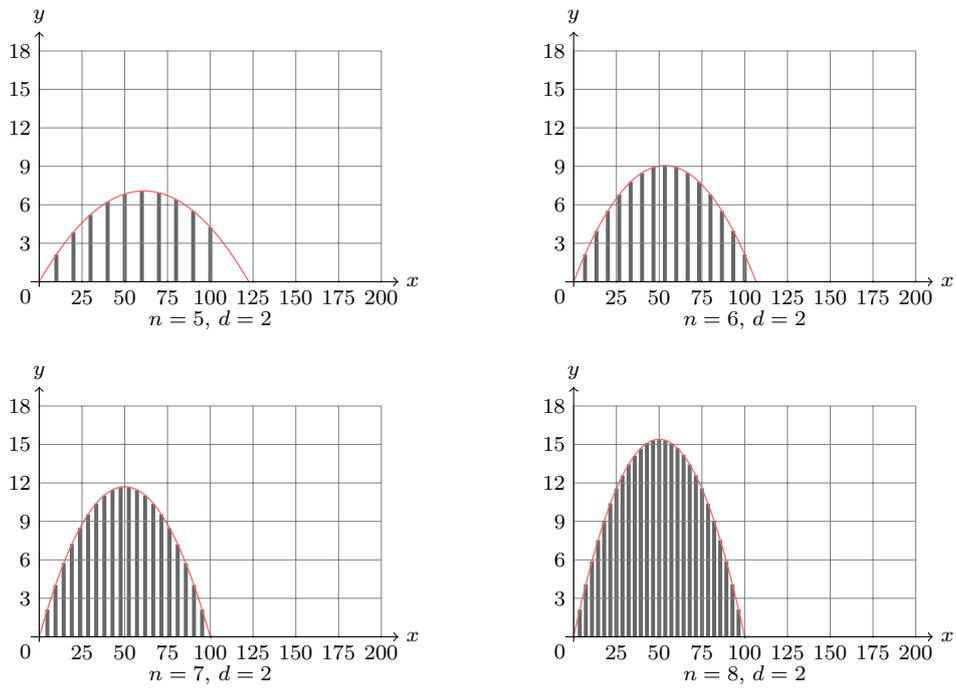


Abbildung A.5: Varianz des Zielfunktionswertes $\mathbb{V}z(G, d, \phi)$ eines zu-fälligen Arrangements für verschiedene Werte von n und m (bzw. von $d(G)$); auf der x -Achse ist die Dichte (prozentuell) und auf der y -Achse die Varianz abgebildet

- in Grau: die gemessenen Werte über 10000 Zufallsgraphen mit entsprechender Dichte
- in Rot: die berechneten Werte, wobei für die Kantenanzahl in Satz 4.2 auch reelle Werte eingesetzt wurden

A.2 Zusammenfassende Tabellen

Tabelle A.1 – Komplett enumerierbare Instanzen						
Graph	d	n	m	Optimallösung	ZFW	
KE_Muster	3	5	7	5-4-1-3-2-*.***	20	
KE_duenner7	2	7	7	7-6-5-4-3-1-2-*	24	
KE_duenner7	3	7	7	7-6-5-4-3-1-2-*.***	18	
KE_duenner7	4	7	7	7-6-5-4-3-2-1-*.***.***	16	
KE_duenner7	5	7	7	7-6-5-4-3-2-1-*.***.***.***.***	16	
KE_duenner7	6	7	7	7-6-5-1-*.***-4-3-2-*.***.***.***.***.***.***.***.***	16	
KE_duenner7	7	7	7	7-6-5-4-3-2-1	14	
KE_dichter7	2	7	14	7-6-5-4-3-2-1-*	62	
KE_dichter7	3	7	14	7-6-4-5-*.***-3-2-1	44	
KE_dichter7	4	7	14	7-6-5-4-3-2-1-*.***.***.***	40	
KE_dichter7	5	7	14	7-6-5-4-3-2-1-*.***.***.***.***.***	38	
KE_dichter7	6	7	14	7-6-5-4-3-1-2-*.***.***.***.***.***.***.***.***.***.***	34	
KE_dichter7	7	7	14	7-6-5-4-3-2-1	28	
KE_Gitter9	2	9	12	9-8-7-4-6-5-3-2-1-*.***.***.***	54	
KE_Gitter9	3	9	12	9-8-7-6-5-4-3-2-1	36	
KE_Gitter9	4	9	12	9-8-7-6-5-4-2-1-3-*.***.***.***	34	
KE_duenner10	2	10	11	10-7-9-5-8-6-4-1-3-2-*.***.***.***	46	
KE_duenner10	4	10	11	10-9-7-5-8-6-4-1-3-2-*.***.***.***	30	
KE_dichter10	2	10	26	10-9-8-5-4-3-2-1-7-6-*.***.***.***	134	
KE_dichter10	4	10	26	10-9-8-5-7-6-*.***-4-3-2-1-*.***.***	80	

Tabelle A.1: *Komplette Enumeration – Liste der komplett enumerierten Instanzen; in der Spalte „Optimallösung“ befindet sich immer nur die Liste der Blätter (eine Zahl markiert den Knotenindex des angeknüpften Knoten, ein Stern heißt, dass das Blatt unbesetzt bleibt)*

Tabelle A.2 – Zusammenfassung																	
Graph	d	n	m	OL	KGS	NA	ZG	ZZG	E	V	G	BSZ	TSZ	HMMSBM	AH	SFH	
KE_Muster	3	5	7	20	18	22	20	20	24,5	2,8	22	20	20	20	20	20	
KE_duenner7	2	7	2	24	21	26	24	24	34,0	10,4	26	24	24	24	26	24	
KE_duenner7	3	7	2	18	16	20	18	18	24,5	3,8	20	18	18	18	18	18	
KE_duenner7	4	7	2	16	14	16	16	16	25,2	3,7	16	16	16	16	16	16	
KE_duenner7	5	7	2	16	14	18	18	16	25,7	3,4	18	18	18	18	18	16	
KE_duenner7	6	7	2	16	14	18	18	16	26,0	3,1	18	16	16	16	16	16	
KE_duenner7	7	7	2	14	14	14	14	14	14,0	0,0	14	14	14	14	14	14	
KE_dichter7	2	7	14	62	56	64	62	62	68,0	10,4	64	62	66	62	62	62	
KE_dichter7	3	7	14	44	42	44	44	44	49,0	4,0	44	44	46	44	44	44	
KE_dichter7	4	7	14	40	35	42	40	40	50,4	5,4	42	40	44	40	40	40	
KE_dichter7	5	7	14	38	30	40	40	38	51,3	5,8	38	38	38	38	38	38	
KE_dichter7	6	7	14	34	28	36	38	34	52,0	5,6	34	34	34	34	34	34	
KE_dichter7	7	7	14	28	28	28	28	28	28,0	0,0	28	28	28	28	28	28	

Fortsetzung an der nächsten Seite ...

Tabelle A.2 – Zusammenfassung – Fortsetzung

Graph	d	n	m	OL	KGS	NA	ZG	ZZG	E	V	G	BSZ	TSZ	HMMSBM	AH	SFH
KE_Gitter9	2	9	12	54	40	58	62	56	78,4	31,4	58	62	54	54	54	54
KE_Gitter9	3	9	12	36	30	36	36	36	42,0	6,2	36	36	36	36	36	36
KE_Gitter9	4	9	12	34	25	36	36	34	43,2	5,8	36	36	34	34	34	34
KE_duenner10	2	10	11	46	34	62	54	48	71,9	30,7	60	56	46	46	48	48
KE_duenner10	4	10	11	30	22	34	32	30	39,6	5,7	34	32	30	30	30	30
KE_dichter10	2	10	26	134	118	140	150	134	169,9	47,1	136	134	140	134	134	136
KE_dichter10	4	10	26	80	74	82	86	80	93,6	8,7	80	80	86	80	80	80
KW_zufaelligA5	7	500	6126	–	21504	40774	46738	40522	46986,4	4517,2	38070	39990	40518	36392	36518	39780
KW_zufaelligB5	7	500	6175	–	21700	41204	47124	40886	47362,3	4552,6	38358	40222	40816	36570	36870	40504
KW_zufaelligA15	7	500	18654	–	84924	124204	142714	123766	143076,2	13231,4	119752	123254	124042	117348	117516	124010
KW_zufaelligB15	7	500	18887	–	86322	125500	144464	125316	144863,3	13386,8	121384	124626	125386	118666	119174	125044
KW_zufaelligA25	7	500	31254	–	160524	207686	239158	207686	239718,2	21286,2	203046	206856	207712	199806	200200	207458
KW_zufaelligB25	7	500	31114	–	159684	206620	238134	206588	238644,4	21200,6	201778	205850	206552	198944	199068	206506
KW_zufaelligA35	7	500	43605	–	234630	289994	333908	289584	334450,4	28491,4	284438	288968	289716	281458	281916	289994
KW_zufaelligB35	7	500	43595	–	234570	290000	333834	289650	334373,7	28485,9	284588	288862	289652	281502	281602	290000
KW_zufaelligA45	7	500	56653	–	312918	376680	433734	376316	434528,5	35360,6	371212	375824	376474	368306	368630	376604
KW_zufaelligB45	7	500	55627	–	306762	369694	426026	369460	426659,1	34484,1	364486	368858	369702	361318	361750	369646
KW_zufaelligA55	7	500	68320	–	382920	454306	523376	454142	524014,4	40856,7	448710	453224	453876	446038	446338	454306
KW_zufaelligB55	7	500	68701	–	385206	456812	526276	456536	526936,7	41025,9	451350	455894	456454	448314	448936	456812
KW_zufaelligA65	7	500	81279	–	460795	540234	622664	540098	623409,9	46246,3	535146	539482	540146	532718	533110	540234
KW_zufaelligB65	7	500	81172	–	460159	539620	621798	539254	622589,2	46204,9	534380	538760	539478	532046	532352	539620
KW_zufaelligA75	7	500	93347	–	548776	620442	715076	620622	715971,5	50588,7	616240	619842	620570	613984	614282	620442
KW_zufaelligB75	7	500	93399	–	549192	621312	715644	620878	716370,3	50606,0	616030	620392	620938	614182	614538	621312
KW_zufaelligA85	7	500	106047	–	650376	705130	812572	705052	813380,5	54453,6	701458	704718	704864	699976	700088	705130
KW_zufaelligB85	7	500	106111	–	650888	705676	812948	705388	813871,4	54471,3	701914	705162	705472	700330	700584	705286
KW_zufaelligA95	7	500	118499	–	749992	788218	907730	787964	908887,3	57541,3	786042	787888	788062	785158	785336	788214
KW_zufaelligB95	7	500	118606	–	750848	788782	908742	788638	909708,0	57564,9	786674	788496	788652	785996	786080	788642
SF_zufaellig50	99	100	4950	10098	10000	10098	19648	10098	19701,0	192,1	10098	10098	10098	10098	10098	10098
SF_Stern	3	20	19	94	66	94	94	94	99,4	27,1	94	94	94	94	94	94
SF_erwStern	4	12	11	28	23	30	32	30	39,6	6,0	32	32	30	30	30	30
Pet03_randomA1	7	1000	4974	–	14006	35988	37952	35680	38150,6	3711,5	32204	35032	35120	29574	29792	35002
Pet03_randomA2	7	1000	24738	–	96266	178872	189270	178334	189740,5	17974,3	171354	177498	178924	166598	166966	178052
Pet03_randomA3	7	1000	49820	–	244920	360072	381548	359372	382119,4	34960,1	350714	358698	359846	343664	344384	359368
Pet03_randomA4	7	1000	8177	–	26710	59134	62466	58842	62717,6	6075,5	54702	58128	58810	51290	51514	58048
Pet03_randomG4	7	1000	8173	–	26697	59124	62296	58840	62686,9	6072,6	40294	39452	40402	31838	34218	56006
Pet03_hc10	7	1024	5120	–	14336	26204	39034	36824	39270,4	3820,4	26280	35020	34452	25892	25896	26192
Pet03_mesh33x33	7	1089	2112	–	4224	8294	16044	15298	16199,0	1582,4	8326	10478	10076	8224	8182	8286
Pet03_bintree10	7	1023	1022	–	2044	7384	7752	7288	7838,7	766,6	7384	6488	2856	3030	3272	6436
Pet03_3elt	7	4720	13722	–	27694	68636	132340	120844	132654,2	10584,7	73754	71042	86290	52420	–	68576
Pet03_airfoil1	7	4253	12289	–	24792	60750	118422	107822	118801,0	9479,3	64224	63170	78482	46768	–	60466
Pet03_crack	7	10240	30380	–	69741	277928	293238	287434	293691,4	23436,6	168802	158460	193856	138178	–	269060
Pet03_whitaker3	7	9800	28989	–	58482	154188	279798	273056	280244,2	22363,3	153490	147924	191430	126198	–	153272
Pet03_small	3	5	8	24	22	26	24	24	28,0	2,7	24	24	24	24	24	24
Pet03_big	7	15606	45878	–	92511	261418	442852	442412	443514,6	35396,6	286180	264862	317434	186158	–	260680
Pet03_wave	7	156317	1059331	–	3299657	8223278	14474822	13238898	14477541,7	823647,5	9203558	8150052	9050366	–	–	8110512
Pet03_c1y	7	828	1749	–	4075	10224	13294	12326	13414,8	1309,8	8778	10426	9454	7508	7576	9652
Pet03_c2y	7	980	2102	–	4822	12218	16010	15044	16122,3	1574,3	10540	12132	11058	8724	9344	11592
Pet03_c3y	7	1327	2844	–	6436	16810	21678	20930	21813,5	2130,6	17786	14652	15230	12216	12556	16612
Pet03_c4y	7	1366	2915	–	6442	17168	22212	21438	22358,1	2183,9	14884	17320	15490	12460	12762	16872
Pet03_c5y	7	1202	2577	–	5750	15112	19450	18730	19612,2	1915,5	13452	15170	13618	10946	11020	14520

Fortsetzung an der nächsten Seite ...

Tabelle A.2 – Zusammenfassung – Fortsetzung																
Graph	d	n	m	OL	KGS	NA	ZG	ZZG	E	V	G	BSZ	TSZ	HMSBM	AH	SFH
Pet03_gd95c	7	62	144	–	320	474	780	608	818,5	94,4	446	460	492	410	398	450
Pet03_gd96a	7	1096	1676	–	3800	12056	12736	12092	12854,9	1256,4	9514	11106	7990	7768	7788	11050
Pet03_gd96b	7	111	193	–	491	1030	1062	928	1097,1	127,5	744	716	772	682	654	696
Pet03_gd96c	7	65	125	–	250	574	680	542	710,5	82,3	426	448	448	378	420	392
Pet03_gd96d	7	180	228	–	555	1028	1258	1188	1296,0	151,6	1060	1126	876	838	854	958

Tabelle A.2: Zusammenfassung – erreichte Ergebnisse für Testinstanzen; fett sind diejenigen unteren Schranken bzw. Heuristiken gekennzeichnet, die für den jeweiligen Graphen den Zielfunktionswert der optimalen Lösung darstellen, sofern dieser bekannt ist; sonst werden die besten Ergebnisse fett markiert

- KGS = Knotengrad-Schranke
- NA = Normales-Arrangement-Verfahren
- ZG = Methode der zufälligen Arrangements für $k = 1000$
- ZZG = Methode der zufälligen zusammenhängenden Arrangements für $k = 1000$
- E = Erwartungswert des Zielfunktionswertes eines zufälligen Arrangements unter der gegebenen Knoten- und Kantenanzahl und der Ordnung des Arrangement-Baumes gerundet auf 1 Nachkommastelle
- V = Varianz des Zielfunktionswertes eines zufälligen Arrangements für einen Graphen mit einer festen Dichte unter der gegebenen Knoten und Kantenanzahl und der Ordnung des Arrangement-Baumes gerundet auf 1 Nachkommastelle,
- G = Greedy-Heuristik – 2. Variante des Algorithmus
- BS = Breitensuche- basiertes Greedy-Verfahren, in dem alle Knoten als Startknoten benutzt werden und das beste daraus resultierende Arrangement ausgewählt wird
- TSZ = Tiefensuche- basiertes Greedy-Verfahren, in dem alle Knoten als Startknoten benutzt werden und das beste daraus resultierende Arrangement ausgewählt wird
- HMSBM = Heuristische Methode der minimalen Schnitte in beschränkten Mengen,
- AH = Austausch-Heuristik – 2. Variante des Algorithmus
- SFH = Shift-Flip-Heuristik – gestoppt wurde der Algorithmus, wenn es 3 Tage zu keiner Zielfunktionswertverbesserung gekommen ist

Literatur

- [Bis11] BISHOP, Peter J.: *Diopsis.jpg*.
<http://en.wikipedia.org/wiki/File:Diopsis.jpg>, 2011
- [Bod87] BODLAENDER, H.L.: The maximum cut and minimum cut into bounded sets problems on cographs / Department of Information and Computing Sciences, Utrecht University. 1987 (RUU-CS-87-12). – Forschungsbericht
- [CLSG10] CAPRARA, Alberto ; LETCHFORD, Adam N. ; SALAZAR-GONZÁLEZ, Juan-José: Decorous Lower Bounds for Minimum Linear Arrangement. In: *INFORMS Journal on Computing* (2010), Mai
- [ER59] ERDŐS, P. ; RÉNYI, A.: On random graphs, I. In: *Publicationes Mathematicae (Debrecen)* 6 (1959), 290 – 297.
http://www.renyi.hu/~p_erdos/1959-11.pdf
- [KV08] KORTE, Bernhard ; VYGEN, Jens: *Kombinatorische Optimierung Theorie und Algorithmen*. Springer-Verlag Berlin Heidelberg, 2008. – Deutsche Ausgabe
- [LN02] LUCZAK, M.J. ; NOBLE, S.D.: Optimal Arrangement of Data in a Tree Directory. In: *Discrete Applied Mathematics* 121 (2002), September
- [MT98] MEINEL, Christoph ; THEOBALD, Thorsten: *Algorithmen und Datenstrukturen im VLSI-Design*. Springer-Verlag Berlin Heidelberg, 1998
- [Pet98] PETIT I SILVESTRE, Jordi: Approximation Heuristics and Benchmarkings for the MinLA Problem. In: BATTITI, Roberto (Hrsg.) ; BERTOSSI, Alan A. (Hrsg.): “*Algorithms and Experiments*” (*ALEX98*) – *Building bridges between theory and applications*, 1998, S. 112–128
- [Pet03] PETIT, Jordi: Experiments on the Minimum Linear Arrangement Problem. In: *Journal of Experimental Algorithmics (JEA)* 8 (2003)