

Gebiete optimal aufteilen

**OR–Verfahren für die Gebietsaufteilung als
Anwendungsfall gleichmäßiger Baumzerlegung**

Zur Erlangung des akademischen Grades eines

Doktors der Wirtschaftswissenschaften

(Dr. rer. pol.)

von der Fakultät für Wirtschaftswissenschaften
der Universität Fridericiana zu Karlsruhe

genehmigte

Dissertation

von

Dipl.-Math. Michael Schröder

Tag der mündlichen Prüfung:

6. Juli 2001

Referent:

Prof. Dr. G. Hammer

Korreferent:

Prof. Dr. W. Gaul

Karlsruhe 2001

Meiner lieben Frau Beate,
für Deine Freundschaft.

Mein großer Dank gilt meinem Doktorvater Herrn Prof. Dr. Gerald Hammer. Mehrfach hat sein Weitblick dem Gang der vorliegenden Arbeit entscheidende Richtungsimpulse gegeben. Meine Doktorandenzeit an seinem Lehrstuhl wird mir in bester Erinnerung bleiben.

Herr Prof. Dr. Wolfgang Gaul hat freundlicherweise das Korreferat übernommen. Auch ihm gilt mein herzlicher Dank.

Mit Dr. Eric Fromm blicke ich auf mehrere gemeinsame Assistentenjahre zurück. Selbstverständliche gegenseitige Unterstützung hat vieles einfacher gemacht. Auch Dr. Michael Löffler hat mich in vielem begleitet. Beiden möchte ich meinen Dank aussprechen.

Mein Bruder Andreas hat sich als Nicht-Mathematiker der Herausforderung gestellt, die Arbeit kritisch zu lesen. Ihm, sowie meinen Eltern, möchte ich hier danken — stellvertretend für die vielen Personen aus Familie und Freundeskreis, deren Anteilnehmen mir das Projekt „Dissertation“ erst möglich gemacht hat.

Inhaltsverzeichnis

I	OR–Modelle für Probleme der Gebietsaufteilung	1
1	Einleitung	3
1.1	Problemstellung und Ziele der Arbeit	3
1.2	Aufbau der Arbeit	5
2	Gebietsaufteilung im Anwendungskontext	9
2.1	Wahlkreiseinteilung	10
2.1.1	Vorschläge zur Wahlkreiseinteilung mit OR–Methoden	11
2.1.1.1	Vereinigte Staaten	12
2.1.1.2	Kanada	13
2.1.1.3	Neuseeland	14
2.1.1.4	Italien	14
2.1.1.5	Wales	15
2.1.1.6	Überblick ausgewählter OR–Studien	15
2.1.2	Wahlkreiseinteilung in Deutschland	15
2.1.2.1	Regelung durch das Bundeswahlgesetz	17
2.1.2.2	Änderungen der Wahlkreiseinteilung	17
2.1.2.3	Übertragbarkeit der OR–Studien aus anderen Ländern	18
2.2	Verkaufsgebiete und Betreuungsgebiete für Vertreter	19
2.2.1	Kriterien für die Einteilung von Verkaufs- und Vertretergebieten.	19
2.2.1.1	Organisationsbezogene Kriterien	20
2.2.1.2	Geographische Kriterien	21
2.2.1.3	Tätigkeitsbezogene Kriterien	22
2.2.2	OR–Literatur zu Verkaufs- und Vertretergebieten	23
2.3	Andere Anwendungen	24
2.3.1	Einzugsgebiete von sozial–sanitären Einrichtungen und Schulen	24
2.3.1.1	Sozial–sanitäre Bezirke	24
2.3.1.2	Schulbezirke	26
2.3.2	Zuständigkeitsbezirke von Serviceeinrichtungen	26

2.3.2.1	Straßenräumung im Winter	27
2.3.2.2	Abfall-Entsorgung	28
2.3.2.3	Server des Rettungswesens	29
2.3.3	Clusterbildung für regional bezogene Daten	29
2.3.4	Überblick ausgewählter Studien	30
3	Grundmodelle für die Gebietsaufteilung	33
3.1	Annahmen in den Grundmodellen	34
3.1.1	Generelle Voraussetzungen	34
3.1.2	Topologische Voraussetzungen	36
3.2	Modell-Typen	37
3.2.1	Location-allocation Modelle	37
3.2.2	Set-partitioning Modelle	38
3.2.3	Heuristische Verfahren	39
3.2.4	In dieser Arbeit untersuchte Modelle	40
3.3	GAMOR: Ein System zur Berechnung von Gebietsaufteilungen	40
3.3.1	Realisierung von GAMOR	41
3.3.2	Ein Beispiel für die Verwendung von GAMOR	42
3.3.2.1	Definition der Problemstellung	42
3.3.2.2	Berechnungsbeispiele mit GAMOR	44
II	Gleichmäßige Baumzerlegung	47
4	Grundlagen	49
4.1	Das allgemeine Baumzerlegungsproblem	50
4.1.1	Bezeichnungen	50
4.1.1.1	Notation	50
4.1.1.2	Einige Klassen von Bäumen	52
4.1.2	Problemstellung der Baumzerlegung	53
4.1.3	Lösung mit dynamischer Programmierung	55
4.1.3.1	Lösung des allgemeinen BZP	55
4.1.3.2	Lösung des allgemeinen p -BZP	58
4.1.4	Lösung mit Spaltenerzeugung	61
4.1.4.1	BZP als lineares Programm und Spaltenerzeugung	61
4.1.4.2	Limited Column Generation nach Shaw	63
4.1.4.3	p -BZP als 0-1-Programm	66
4.2	Modelle für gleichmäßige Baumzerlegung	67
4.2.1	Ein Klassifikationsschema für Baumzerlegungsprobleme	67

4.2.2	Probleme der gleichmäßigen Baumzerlegung	70
4.3	Komplexität von Problemen der gleichmäßigen Baumzerlegung	72
4.3.1	\mathcal{NP} -schwere Probleme	72
4.3.2	Polynomial lösbare Probleme	75
4.4	Komplexität verwandter Problemstellungen	79
5	Polynomiale Verfahren	85
5.1	Vorbereitungen	85
5.1.1	Das Problem WEAK SUBSET SUM (WSS)	86
5.1.1.1	Definition von WSS	86
5.1.1.2	Ein Verfahren zur Lösung von WSS	86
5.1.1.3	WSS mit mehreren Bedingungen.	89
5.1.2	Rucksackprobleme mit Präzedenzbedingungen	90
5.1.2.1	Definitionen	91
5.1.2.2	Algorithmen für TKP	91
5.1.3	Das Problem TREE WEAK SUBSET SUM (TWSS)	94
5.1.3.1	Definition von TWSS	95
5.1.3.2	Lösung von TWSS	95
5.2	Lösung von $T/L, U/\cdot/\exists$	98
5.2.1	Vorüberlegungen	98
5.2.2	Lösung im Fall $L \leq n(U - L)$	99
5.2.3	Lösung im Fall $L > n(U - L)$	99
5.2.4	Zusammenfassende Darstellung des Verfahrens	101
5.3	Lösung von $T/L, U/p/\exists$	101
5.3.1	Lösung im Fall $L > p(U - L)$	103
5.3.2	Lösung im Fall $L \leq p(U - L)$	104
5.3.3	Zusammenfassende Darstellung des Verfahrens	106
5.4	Lösung von $T/Z_{\text{ent}}, U_z - L_z = \text{const}/\cdot/\exists$	108
5.4.1	Der Fall $v \in Z$	109
5.4.2	Der Fall $v \notin Z$	110
5.4.3	Zusammenfassende Darstellung des Verfahrens	111
5.5	Lösung von $\ \cdot\ _\infty$ -Problemen	111
6	Ein Branch&Bound-Verfahren für $T/\cdot/p/\ \cdot\ _\infty$	115
6.1	Darstellung des Verfahrens	115
6.1.1	Der Suchbaum	117
6.1.1.1	Aufbau des Suchbaumes	117
6.1.1.2	Das Ausloten und das Reduzieren von Knoten	118
6.1.1.3	Heuristische obere Schranke	119

6.1.1.4	Untersuchung eines Knotens	119
6.1.2	Einige Definitionen	120
6.1.3	Kriterien für das Reduzieren	121
6.1.4	Das Ausloten: Test einer unteren Schranke	122
6.1.4.1	Ausloten des Wurzelknotens	123
6.1.4.2	Ausloten beliebiger Knoten	123
6.1.5	Heuristik zur Berechnung einer oberen Schranke	126
6.2	Rechenergebnisse	128
6.2.1	Testprobleme	128
6.2.1.1	Zufällige Bäume	128
6.2.1.2	Zufällige Eckengewichte	129
6.2.1.3	Menge der Testinstanzen	130
6.2.2	Testergebnisse	130
6.2.2.1	Gleichverteilte Eckengewichte	133
6.2.2.2	Logarithmisch-normalverteilte Eckengewichte	133

III Gebietsaufteilung mit gleichmäßiger Baumzerlegung 135

7	Modelle und Verfahren für ebene Probleme	137
7.1	Ein klassisches Zuordnungsmodell für die Gebietsaufteilung	138
7.1.1	KGE und Zentren als Punkte der Ebene	138
7.1.2	Das Zuordnungsmodell nach Hess et al.	140
7.2	Transportmodell und Splitauflösung	143
7.2.1	Das Transportmodell	143
7.2.2	Splitauflösung und Splitadjazenz	145
7.2.3	Eine einfache Heuristik zur Splitauflösung	146
7.2.4	Optimale Splitauflösung durch Baumzerlegung	149
7.2.4.1	Gleichmäßige Baumzerlegung der Splitadjazenz	150
7.2.4.2	Komplexität optimaler Splitauflösung	152
7.2.5	Praktikable Verfahren zur optimalen Splitauflösung	153
7.2.5.1	Vorbemerkungen	154
7.2.5.2	Rekursionsgleichungen für die optimale Zerlegung von T^S	155
7.2.5.3	Aufwand der optimalen Zerlegung von T^S	158
7.2.5.4	Balancierte Bezirke mit optimaler Zerlegung von T^S	158
7.2.5.5	Worst-case Abschätzung der maximalen Abweichung	161
7.2.5.6	Minimierung der maximalen Abweichung mit polynomialem Aufwand	162

7.2.6	Partielle Splitauflösung bei beschränkter Abweichung	164
7.2.6.1	Sukzessive Toleranzervergrößerung	166
7.2.6.2	Die local assignment / deassignment Heuristiken	168
7.2.6.3	Rechenergebnisse	169
7.3	Lagrange–Relaxation und Subgradienten–Verfahren	171
7.3.1	Das Verfahren von Zoltners und Sinha	171
7.3.2	Anschauliche Interpretation des Verfahrens	173
7.3.3	Beziehung zur Vorgehensweise in 7.2	175
8	Modelle und Verfahren für Probleme auf Netzwerken	181
8.1	Topologische Ausgangssituation	181
8.1.1	KGE als Ecken eines Graphen	181
8.1.2	Distanzmessung über kürzeste Wege	183
8.1.3	Modellierung der Zusammenhangsbedingung	184
8.1.3.1	Schnittformulierung des Zusammenhangs	184
8.1.3.2	Flußformulierung des Zusammenhangs	185
8.1.3.3	Verstärkte Zusammenhangsformulierungen	186
8.2	Diskussion der Übertragbarkeit von Verfahren für ebene Probleme	189
8.2.1	Transportmodell bei KW–Distanz	189
8.2.2	Lagrange–Relaxation bei KW–Distanz	191
8.2.3	Zusammenfassung	193
8.3	Gebietsaufteilung durch gleichmäßige Baumzerlegung	194
8.3.1	Gerüst–zusammenhängende balancierte Bezirke	194
8.3.2	Auswahl des Gerüsts	195
8.3.2.1	Verwendung des Minimalgerüsts	195
8.3.2.2	Verwendung der Optimalbasis eines Flußproblems	195
8.3.3	Berechnung von Senken für das Flußproblem MF	199
8.3.3.1	Eine Rundungsheuristik zur Bestimmung optimaler Senken	199
8.3.3.2	Berechnung optimaler Senken mit dynamischer Pro- grammierung	201
8.3.4	Zusammenfassende Darstellung und Beispiele	202
8.3.4.1	Zusammenfassung	202
8.3.4.2	Praktische Umsetzung	202
8.3.4.3	Beispiele	203
8.3.5	Vorteile gegenüber herkömmlichen Verfahren	206

9	Verfahren zur Zentrenbestimmung (simultane Standortplanung)	207
9.1	Optimale Zentren bei gegebener Gebietsaufteilung	208
9.1.1	Berechnung des Medians bei ebenen Problemen	209
9.1.1.1	Quadriert-euklidische Distanzen	209
9.1.1.2	Euklidische Distanzen	209
9.1.2	Berechnung des Medians bei Netzwerk-Topologie	210
9.2	Bestimmung von Zentren mit dem location-allocation Prinzip	210
9.2.1	Das GEOLINE-Modell	210
9.2.2	Konvergenz des location-allocation Prinzips	211
9.2.3	Abhängigkeit von den Startzentren	212
9.3	Eine Lagrange-Heuristik zur Bestimmung optimaler Zentren	212
9.3.1	Ein kapazitiertes warehouse location Problem	213
9.3.1.1	Formulierung des Modells	213
9.3.1.2	Lösungsverfahren für WLP	214
9.3.2	Eine Lagrange-Heuristik zur Lösung von WLP	214
9.3.2.1	Lagrange-Relaxation von WLP	214
9.3.2.2	Darstellung der Heuristik	215
9.3.2.3	Anmerkungen zur Implementierung	216
9.3.3	Rechenergebnisse	217
9.3.3.1	Ergebnisse der Lagrange-Heuristik	218
9.3.3.2	Qualität der Schranken	220
9.3.3.3	Iterationenzahl bei Berechnung der oberen Schranke	222
	Zusammenfassung und Ausblick	225
	Glossar graphentheoretischer Begriffe	229
	Literaturverzeichnis	235

Teil I

OR–Modelle für Probleme der Gebietsaufteilung

Kapitel 1

Einleitung

1.1 Problemstellung und Ziele der Arbeit

Das Problem der Aufteilung eines geographischen Gebietes in kleinere Einheiten mit Hilfe von Methoden des Operations Research (OR) zu behandeln, hat eine Tradition, die mehrere Jahrzehnte zurückreicht. Sie ist motiviert durch Aufgaben wie

- Wahlkreiseinteilung,
- Verkaufsgebietsgestaltung,
- Abgrenzung der räumlichen Zuständigkeit von Serviceeinrichtungen,

die sich in der Praxis regelmäßig stellen und die in der Regel eine große Anzahl möglicher Lösungen zulassen. Die daraus resultierende Komplexität des Entscheidungsproblems macht Rechnerunterstützung wünschenswert, und es existieren — nicht überraschend — viele Vorschläge in der OR-Literatur für geeignete Modelle und Verfahren. Die vorliegende Arbeit knüpft hier an und hat im Kern das Ziel, aufzuzeigen, daß sich aus dem Studium sog. *gleichmäßiger Baumzerlegungen* Verfahren für die Gebietsaufteilung gewinnen lassen, die unter Gesichtspunkten wie Mächtigkeit und Effizienz leistungsfähiger als herkömmliche Methoden sind.

Dabei wird das Thema auf breiter Basis behandelt; im einzelnen sind unsere Ziele die folgenden:

1. *Darstellung des Anwendungskontexts.* Die unterschiedlichen Anwendungssituationen, in denen ein Gebietsaufteilungsproblem mit Hilfe von OR-Methoden behandelt werden kann, sollen besprochen werden. Gemeinsamkeiten und Unterschiede dieser Anwendungen sollen deutlich werden.
2. *Unabhängigkeit von einer speziellen Anwendung.* Durch Konzentration auf die Gemeinsamkeiten der Anwendungsbereiche sollen Modelle untersucht werden,

die anwendungsübergreifend einsetzbar sind. Wir geben damit hochaggregierten Modellen, die wenige wesentliche Aspekte enthalten, den Vorzug.

Wahlkreiseinteilung und Verkaufsgebietsgestaltung haben eng verwandte Eigenschaften. Grob gesagt, geht es darum, Bezirke zu bilden, die „vernünftige“ geographische Eigenschaften haben und die darüberhinaus „fair“, also gleich groß, sind. Ob „Größe“ sich dabei auf die Bevölkerungszahl oder auf ein Maß wie Umsatzpotential oder Arbeitslast bezieht, spielt aus Sicht eines OR-Modells in der Regel keine Rolle. Die OR-Literatur zu Problemen der Gebietsaufteilung enthält viele Beispiele des Austauschs von Modellen zwischen verschiedenen Anwendungsbereichen. So wurde das bekannte GEOLINE-Modell von Hess et al. [50] ursprünglich für die Wahlkreiseinteilung entworfen, dann für Verkaufsgebiete übernommen (Hess und Samuels [51]) und erweitert (Fleischmann und Paraschis [35]), und schließlich in den vergangenen Jahren wieder (mit Ergänzungen) für die Wahlkreiseinteilung benutzt (George et al. [42], Hojati [52]).

3. *Analyse bekannter Modelle zur Gebietsaufteilung.* Die Eigenschaften von Modellen und Verfahren des Operations Research werden häufig vor allem an Hand praktischer Beispiele demonstriert. Dies trifft auch für das Problem der Gebietsaufteilung zu. Daneben ist jedoch das tiefere Verständnis solcher Verfahren ein anzustrebendes Ziel. In dieser Arbeit soll dazu beigetragen werden, fokussiert auf sog. *location-allocation* Modelle für die Gebietsaufteilung.

Nahezu alle Arbeiten im OR zur Gebietsaufteilung gehen von einer endlichen Menge an kleinsten Gebietseinheiten (KGE) aus, und gruppieren diese zu Bezirken. Location-allocation bedeutet, zunächst für jeden zu bildenden Bezirk ein Zentrum festzulegen (location) und dann simultan diesen Zentren die KGE zuzuordnen (allocation). Im allgemeinen soll die Zuordnung zu einem nahegelegenen Zentrum erfolgen, gleichzeitig aber sollen balancierte, d.h. möglichst gleichgroße, Bezirke gebildet werden. Neben location-allocation Modellen wurden andere vorgeschlagen (siehe 3.2), die in dieser Arbeit nur eine untergeordnete Rolle spielen.

4. *Entwicklung neuer Verfahren zur Gebietsaufteilung, basierend auf der Zerlegung baumförmiger Graphen.* Schwerpunkt unserer Arbeit ist die Entwicklung von Algorithmen, mit denen sich Lösungen für die untersuchten Modelle berechnen lassen. Dafür arbeiten wir mit einem heuristischen Ansatz, der unseres Wissens bisher nur von Maravalle und Simeone [66] im Rahmen der Gebietsaufteilung benutzt wurde. Er besteht in der Lösung von Zerlegungsproblemen für baumförmige Graphen. Bedingt durch die Baumstruktur sind diese Probleme

aus algorithmischer Sicht häufig „gutartig“. Unser Ziel ist es, dieses Kapital zu nutzen.

Die aus Anwendungssicht wichtige Forderung nach geographisch zusammenhängenden Bezirken erfordert zur modellmäßigen Erfassung den Bezug auf einen geeigneten Nachbarschaftsgraphen der KGE. In location-allocation Modellen muß dann mit diesem Graph gearbeitet werden, was ihre Lösung in der Regel zu einem \mathcal{NP} -schweren Problem macht. Hier kann die Einschränkung auf einen baumförmigen Graphen zu einem polynomial lösbaeren Problem führen.

5. *Eingehendes Studium der gleichmäßigen Baumzerlegung.* Die Untersuchung des Problems der gleichmäßigen Baumzerlegung soll nicht ausschließlich im Hinblick auf die Nutzbarmachung in Algorithmen für Gebietsaufteilungsmodelle geschehen. Der Verfasser ist der Ansicht, daß die gleichmäßige Baumzerlegung ein eigenständig zu sehendes, interessantes Forschungsgebiet ist, und möchte auch dazu einen Beitrag leisten.

Gleichmäßige Baumzerlegung wird in unserer Arbeit als die Partition eines eckengewichteten baumförmigen Graphen in Teilbäume möglichst gleichen Gewichts verstanden. Für dieses Problem lassen sich verschiedene Modelle angeben (vgl. Tab. 4.1, S. 73); es hat neben der Gebietsaufteilung viele andere Anwendungen, vor allem in der Informatik.

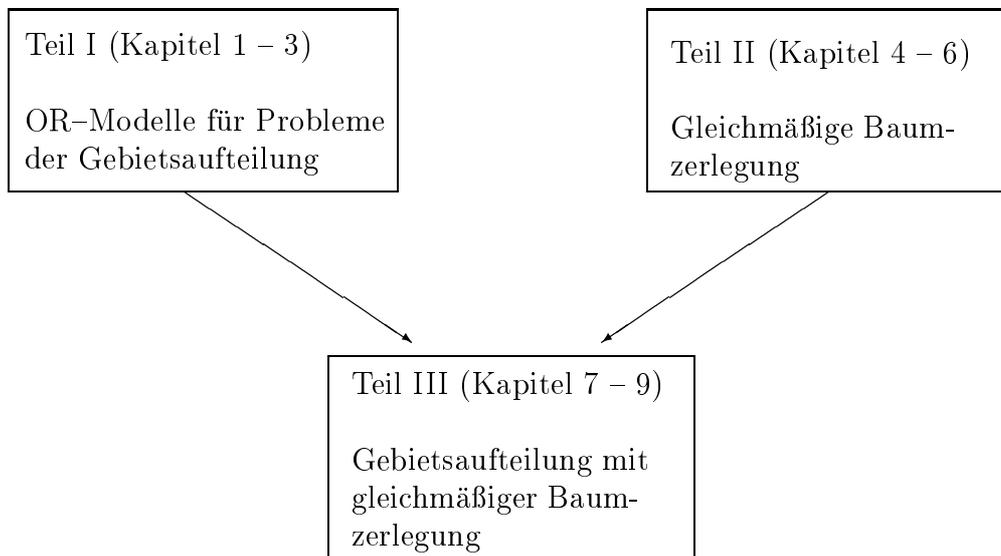
6. *Implementierung bekannter und neuer Algorithmen in einem Softwaresystem.* Die entwickelten Verfahren sollen getestet und mit herkömmlichen Verfahren verglichen werden. Dazu wurde das Softwaresystem GAMOR programmiert, das eine ganze Reihe solcher Verfahren zur Verfügung stellt.

„GAMOR“ ist Akronym von „Gebiete aufteilen mit Operations Research-Methoden“. Das implementierte System umfaßt neben den Algorithmen eine einfache Schnittstelle zur Problemspezifikation und grafische Ausgabemöglichkeiten.

1.2 Aufbau der Arbeit

Unsere Arbeit ist in drei Teile gegliedert. Teil I behandelt Anwendungen und grundlegende Modellbildungen des OR für die Gebietsaufteilung. Ferner wird das Softwaresystem GAMOR vorgestellt. Teil II untersucht das Problem gleichmäßiger Baumzerlegung; vorwiegend aus algorithmischer Sicht. Teil III führt das Material der ersten beiden Teile gedanklich zusammen. Modelle für die Gebietsaufteilung, hauptsächlich

vom location–allocation Typ, werden analysiert und neue Verfahren, basierend auf gleichmäßiger Baumzerlegung, werden vorgeschlagen.



Übersicht von Teil I:

Kapitel 2 stellt Anwendungen der Gebietsaufteilung dar. Dabei werden solche Anwendungen berücksichtigt, die tatsächlich in der OR–Literatur untersucht worden sind. Die meiste Aufmerksamkeit hat die Einteilung von Wahlkreisen und die Gestaltung von Verkaufsgebieten gefunden. Insbesondere für Wahlkreise stellen wir auch die bundesdeutsche Situation dar, da es unseres Wissens keine entsprechende Arbeit aus dem Operations Research gibt. Schwerpunkt des Kapitels ist es, wesentliche Kriterien, die eine anwendungsübergreifende Bedeutung haben, herauszuarbeiten.

Kapitel 3 behandelt grundlegende Modelle für die Gebietsaufteilung und demonstriert beispielhaft die Verwendung von GAMOR zur Berechnung von Lösungen solcher Modelle. Grundlegende Modelle beschränken sich auf die in Kapitel 2 als wesentlich erkannten Kriterien und erreichen dadurch eine hohe Allgemeinheit. Wir skizzieren die verschiedenen Modell–Typen, die in der Literatur zu finden sind; im weiteren Verlauf der Arbeit (Teil III) werden Modelle vom location–allocation Typ untersucht.

Übersicht von Teil II:

Kapitel 4 führt in die Problemstellung der gleichmäßigen Baumzerlegung ein. Zunächst wird Baumzerlegung in allgemeiner Form untersucht, der Schwerpunkt

liegt auf geeigneten Algorithmen. Danach wird auf Modelle für das Ziel der gleichmäßigen Zerlegung eingegangen, die meisten davon sind bereits in der Literatur zu finden. Ein Schwerpunkt ist die Diskussion von Fragen der Problemkomplexität, da ihr aus algorithmischer Sicht eine wichtige Bedeutung zukommt.

Kapitel 5 entwickelt Algorithmen mit polynomialem Aufwand für Baumzerlegungsprobleme, in denen die maximale Abweichung von einem Zielwert für das Gewicht der Teilbäume minimiert werden soll. Unseres Wissens war es bisher nicht bekannt, daß diese Problemstellungen polynomial lösbar sind. Aus Anwendungssicht ist die maximale Abweichung jedoch ein wichtiger Parameter.

Kapitel 6 präsentiert ein Branch&Bound–Verfahren für eines der in Kapitel 5 behandelten Probleme, da der polynomiale Algorithmus zu dessen Lösung eine Laufzeitabschätzung von hohem Grad aufweist. Das Branch&Bound–Verfahren wird durch Testrechnungen als sehr effizient erwiesen. Dies macht es insbesondere praktikabel für die Verwendung in Algorithmen für die Gebietsaufteilung.

Übersicht von Teil III:

Kapitel 7 untersucht ein klassisches Zuordnungsmodell für Gebietsaufteilungsprobleme, deren Topologie durch Punkte der Ebene charakterisiert ist. Für dieses Modell haben sich zwei Lösungswege etabliert. Der erste ist das Lösen der linearen Relaxation (ein Transportmodell) mit anschließendem Runden. Wir verwenden gleichmäßige Baumzerlegung, um dieses Runden in einem optimalen Sinn durchzuführen. Der zweite, gemeinhin als besser eingeschätzte Weg, ist die Verwendung einer Heuristik vom Lagrangen Typ. Wir arbeiten heraus, daß diese, mehr auf Erfahrung beruhende Einschätzung einer tiefergehenden Analyse nicht standhält.

Kapitel 8 behandelt Modelle mit Netzwerk–Topologie. In ihnen kann die Forderung nach geographisch zusammenhängenden Bezirken abgebildet werden. Jedoch resultiert dies im allgemeinen in schwierigen Optimierungsproblemen. Mit Hilfe von gleichmäßiger Baumzerlegung läßt sich jedoch ein effizienter Algorithmus zur heuristischen Behandlung des Problems konstruieren.

Kapitel 9 beschäftigt sich mit der Berechnung von Zentren für die Bezirke in Gebietsaufteilungen. Dies ist der location Schritt in location–allocation Verfahren. Schwerpunkt des Kapitels ist die Verbesserung einer von Hojati [52] vorgeschlagenen Lagrange–Heuristik zur Bestimmung möglichst guter Zentren.

Kapitel 2

Gebietsaufteilung im Anwendungskontext

Die Untersuchung von Modellen für die Gebietsaufteilung im Operations Research ist durch einen Kreis ganz unterschiedlicher Anwendungen motiviert. Sie sollen in diesem Kapitel diskutiert werden. Dabei dienen folgende Überlegungen als Leitlinien:

1. Zunächst ist festzustellen, daß die überwiegende Zahl der Arbeiten, die Gebietsaufteilung zum Gegenstand haben, jeweils auf einen der Anwendungsbereiche fokussieren. Selten findet man eine Diskussion des „Gebietsaufteilungsproblems an sich“. Die Tendenz im Operations Research, daß Modellbildungen aus ihrem Anwendungskontext gelöst und als grundlegende Modelle etabliert und erforscht werden, ist im Bereich der Gebietsaufteilung bisher nur wenig zu beobachten.
2. Auf der anderen Seite erweist es sich, daß die jeweiligen Entscheidungssituationen typische gemeinsame Eigenschaften aufweisen, trotz der ganz unterschiedlichen Anwendungen. Folglich sind die entwickelten OR-Modelle ähnlich, zum Teil sogar direkt von einer Anwendung auf die andere übertragbar.
3. In dieser Arbeit wird bei der Untersuchung der Entscheidungssituation „Gebietsaufteilung“ von einem konkreten Anwendungsbereich abstrahiert. In Anbetracht von Punkt zwei erscheint dies prinzipiell gerechtfertigt.¹ Auf der anderen Seite sollen und können die Anwendungen nicht ignoriert werden. Im wesentlichen muß sich ihre Darstellung aber auf das beschränken, was sich in den Arbeiten mit OR-Hintergrund niedergeschlagen hat. So kann die Darstellung des Themas Wahlkreisgestaltung nicht mit der Kompetenz des Verfassungsrechtlers

¹Für eine detaillierte Begründung siehe Seite 33.

erfolgen, ebensowenig wie die Planung von Verkaufsgebieten vom Standpunkt des Marketing-Experten aus referiert werden kann.

Aus diesen Überlegungen ergeben sich drei Ziele für das vorliegende Kapitel:

1. Die Anwendungen der Gebietsaufteilung sollen so ausführlich besprochen werden, daß ein „plastischer“ Eindruck vom Problemkontext unserer Arbeit entsteht. Dem vor allem an der Praxis interessierten Leser soll es dadurch möglich sein, das Thema der Arbeit zu überblicken.
2. Die anwendungsübergreifende Perspektive unserer Arbeit soll begründet werden, indem deutlich gemacht wird, daß die für die jeweiligen Anwendungen essentiellen Kriterien dieselben, oder zumindest einander ähnlich sind.
3. Durch Literatur-Überblicke soll ein Einstieg zu den jeweiligen Anwendungsbereichen gegeben werden.

2.1 Wahlkreiseinteilung

Die Bildung von Wahlkreisen ist ein in vielen Demokratien verwendetes Mittel für die direkte Wahl der Abgeordneten. Die Festlegung der Wahlkreiseinteilung ist dabei in der Regel dem Gesetzgeber vorbehalten, da sie ein zentrales Moment der Demokratie darstellt. Die Dauerhaftigkeit einer vorgenommenen Einteilung ist jedoch begrenzt. Veränderungen der Abgeordnetenzahl oder Veränderungen in der Bevölkerungsverteilung machen eine Anpassung der Wahlkreisgrenzen oder eine vollständige Neuabgrenzung notwendig. Das Problem der Einteilung des Wahlgebietes in Wahlkreise tritt somit regelmäßig auf.

Seit etwa 1960 werden Vorschläge für rechnergestützte Verfahren zur Wahlkreisabgrenzung gemacht. Neben der Überzeugung, daß sich die gesetzlich vorgegebenen Anforderungen an die Wahlkreiseinteilung gut in einer mathematischen Modellbildung einfangen lassen, dürften vor allem zwei Motive dafür ausschlaggebend sein:

1. In den Vereinigten Staaten, aus denen ein großer Teil der Vorschläge stammt, gibt es immer wieder harte politische Auseinandersetzungen über Wahlkreiseinteilungen, die in juristische Verfahren bis hinauf zum Supreme Court münden.

Im Raum steht dabei das Problem des sogenannten *gerrymandering*,² womit im

²Der Begriff entstand aus Zusammenziehung des Namens von Elbridge Gerry (1744-1814), seinerzeit Gouverneur von Massachusetts, mit dem Wort „salamander“. Um seine Wiederwahl zu sichern, setzte Gerry eine Wahlkreiseinteilung durch, die einen salamanderförmigen Wahlkreis enthielt. Dieser Wahlkreis umfaßte die „Hochburgen“ von Gerrys Anhängern (di Cortona et al. [29, S. 141]).

weitesten Sinne der Versuch bezeichnet wird, durch geeignete Wahlkreisgrenzen den Wahlausgang zu beeinflussen.³ Durch die Verwendung „neutraler“ Verfahren, die keine demographischen Daten einbeziehen, soll gerrymandering verhindert werden. Ein Rechenverfahren, das ohne Bezug auf solche Daten abläuft, könnte die Rolle eines über jeden Verdachts der Parteinahme erhabenen Schlichters einnehmen.

2. Die Wahlkreisgestaltung muß bevölkerungsstatistische Daten einbeziehen. Da diese heute in elektronischer Form vorliegen, ist es naheliegend, für das Wahlkreisproblem Rechner einzusetzen. Insbesondere wenn das Datenmaterial sehr umfangreich ist, wird die Handhabung der Aufgabe dadurch vereinfacht. In natürlicher Weise stellt sich damit die Frage nach entscheidungsunterstützenden Verfahren.

Im nächsten Abschnitt gehen wir auf Länder ein, für die Vorschläge zur Wahlkreiseinteilung mit Operations Research Verfahren in der Literatur zu finden sind. Für Deutschland ist dem Autor keine solche Arbeit bekannt, wir gehen aber der Frage nach, ob die Modelle auch auf die deutsche Situation anwendbar wären. Einen Überblick zu rechtlichen Fragen der Wahlkreisgestaltung in einer Vielzahl westlicher Demokratien gibt ein Gutachten des Max-Planck-Instituts für ausländisches öffentliches Recht und Völkerrecht [39].

2.1.1 Vorschläge zur Wahlkreiseinteilung mit OR-Methoden

Eine Zusammenstellung und umfassende Verarbeitung der OR-Literatur zum Wahlkreisproblem bis 1996 bietet Teil III der Monographie „Evaluation and Optimization of Electoral Systems“ von di Cortona et al. [29]. Darin wird das Problem länderübergreifend studiert; das ist möglich, denn die sich aus gesetzlichen und juristischen Vorgaben ergebenden Kriterien sind in den unterschiedlichen Demokratien ähnlich.⁴ Ziel der folgenden Darstellung ist es, solche Kriterien für verschiedene Demokratien beispielhaft aufzuzeigen, und damit die Anwendungsseite stärker zu betonen, als dies bei di Cortona et al. [29] der Fall ist.

³Unterschieden werden: 1. *partisan gerrymander*, 2. *bipartisan gerrymander* und 3. *racial gerrymander* (siehe Mehrotra et al. [71]). Das erste ist Manipulation der Wahlkreisgrenzen zum Nutzen einer Partei, die zweite Variante ist die abgesprochene Aufteilung des Wahlgebiets mit dem Ziel, die Amtsinhaber zu schützen, das dritte die Schwächung oder Stärkung des Gewichts von Minderheiten.

⁴Es ist anzumerken, daß nicht in allen Demokratien das Wahlkreisproblem auftritt; zum Beispiel haben Israel und die Niederlande ein reines Verhältniswahlrecht und deshalb keine Wahlkreiseinteilung.

2.1.1.1 Vereinigte Staaten

Die amerikanische Verfassung enthielt ursprünglich zwar Vorgaben, wie die Anzahl der Kongreß-Abgeordneten auf die Mitgliedsstaaten zu verteilen sei, jedoch nicht darüber, wie die Wahl der Abgeordneten in den Bundesstaaten stattfinden sollte. Zum Teil verwendeten die Bundesstaaten Wahlkreise, die je einen Abgeordneten wählten (*single-member districts*), zum Teil wurden alle Abgeordneten von allen Bürgern gewählt, wobei jeder eine der Anzahl der Abgeordneten entsprechende Stimmzahl hatte (*multi-member districts*) (Mehrotra et al. [71]). Durchgesetzt hat sich jedoch die erste Vorgehensweise. In moderner Zeit werden im Zehnjahres-Rhythmus die Bevölkerungsstatistiken neu erhoben. Unter Berücksichtigung dieser Daten wird gegebenenfalls die Anzahl der Abgeordneten der einzelnen Bundesstaaten neu festgelegt, was nachfolgend eine Neueinteilung der Wahlkreise nötig macht.

Zum Teil sind die Vorgaben, die bei einer solchen Einteilung zu berücksichtigen sind, gesetzlich festgelegt,⁵ zum Teil beruhen sie auf Grundsatzurteilen des Supreme Court. Folgende Kriterien werden in der Literatur genannt (siehe zum Beispiel Altman [5]):

1. Das *one person one vote* Prinzip fordert, daß alle Stimmen gleiches Gewicht haben. Erreicht wird dies durch Wahlkreise mit gleicher Bevölkerungszahl. Die Auslegung des Prinzips ist infolge eines Urteils aus den sechziger Jahren heute sehr streng, die Abweichung von der mittleren Größe beträgt in den meisten Bundesstaaten weniger als ein Prozent.
2. Das Kriterium der *contiguity* fordert geographisch zusammenhängende Wahlkreise.
3. Das „Beachten traditioneller Grenzen“ fordert eine Übereinstimmung von Wahlkreisgrenzen mit solchen von Gemeinden und Bezirken (*counties*).
4. Vor allem durch die Problematik des gerrymandering ist die Forderung nach Kompaktheit (*compactness*) der Wahlkreise motiviert. Während die Idee dieses Kriteriums, daß die Form von Wahlkreisen rund und geschlossen, anstatt lang und dünn oder mäanderförmig sein soll, einleuchtend erscheint, ist die Umsetzung der Forderung Gegenstand intensiver Diskussion. Bis heute gibt es keine Übereinstimmung in der Frage, was Kompaktheit genau bedeutet, und wie sie

⁵Da die Kriterien nicht in der Verfassung verankert sind, wurden entsprechende Gesetze immer wieder vom Kongreß verabschiedet. Es ist jedoch umstritten, ob der Kongreß dazu berechtigt ist, beziehungsweise, ob er die Anwendung der Gesetze erzwingen darf (Altman [5]).

gemessen werden kann.⁶

Aufbauend auf diesen Kriterien wurde eine Vielzahl von Operations Research Verfahren zur Wahlkreiseinteilung vorgeschlagen. Die Modelle haben aufgrund des dritten Kriteriums eine diskrete Struktur, das heißt, es werden geographische Einheiten wie *counties*, oder auch kleinere Einheiten mit statistischer Bedeutung, zu Wahlkreisen zusammengefaßt. Dabei werden hinsichtlich der Anzahl an Wahlberechtigten gleichgroße Bezirke gebildet, die sowohl die Zusammenhangsforderung beachten, als auch eine geeignet operationalisierte Kompaktheit aufweisen (siehe Tabelle 2.1 auf Seite 16).

2.1.1.2 Kanada

Das kanadische Wahlrecht sieht zunächst eine Aufteilung der Anzahl der Sitze im Parlament (*House of Commons*) auf die Provinzen vor und nachfolgend die Bildung einer entsprechenden Anzahl an Wahlkreisen in den Provinzen. Alle zehn Jahre ist basierend auf den jeweils aktuellen Zensusdaten die Wahlkreiseinteilung zu überprüfen.

Für den Wahlkreis-Zuschnitt sind folgende Kriterien maßgeblich (o.V. [39]):

1. Wahlrechtsgleichheit soll durch gleiche Bevölkerungszahlen in den Wahlkreisen gewährleistet werden. Es ist das Ziel möglichst gleichgroßer Wahlkreise anzustreben. Eine Abweichung von maximal 25 Prozent vom Mittel, in Ausnahmefällen 50 Prozent, ist zulässig, wenn dies zur Erfüllung der unter 2. genannten Forderungen nötig ist.
2. Die Wahlkreiseinteilung muß beachten: *communities of interest*, *communities of identity*, *historical patterns* und eine handhabbare geographische Größe der Wahlkreise.

Vorschläge aus der OR-Literatur zur Wahlkreiseinteilung in Kanada finden sich in Bourjolly et al. [18] und Hojati [52]. Sie beziehen sich nicht auf die Einteilung der Wahlkreise einer gesamten Provinz, sondern auf regional begrenzte Problemstellungen. Während Bourjolly et al. [18] neben dem Kriterium der Bevölkerungsgleichheit und einem Kompaktheitsmaß auch eine Modellierung für Interessengleichheit in die Methode einbeziehen, entspricht das Modell von Hojati [52] weitgehend den OR-Modellen für die Wahlkreiseinteilung in den Vereinigten Staaten.

⁶Horn und Hampton [53], Niemi et al. [80] und Young [113] diskutieren eine Vielzahl von Kompaktheitsmaßen.

2.1.1.3 Neuseeland

Das Neuseeländische Recht für die Wahl des Parlaments wurde in Folge eines Referendums im Jahr 1993 von einem Mehrheitswahlsystem angelsächsischer Prägung in ein gemischtes Verhältniswahlsystem nach deutschem Vorbild umgewandelt. Infolgedessen wurde die Anzahl der Wahlkreise um etwa ein Drittel auf ungefähr 60 verringert, was einen vollständigen Neuzuschnitt nötig machte.

Die folgenden Kriterien werden beim Wahlkreiszuschnitt beachtet (George et al. [42]):

1. Bevölkerungsgleichheit in allen Wahlkreisen; Abweichungen bis zu fünf Prozent sind zulässig, um weiteren Forderungen zu genügen.
2. Keine Überschneidung mit natürlichen Barrieren wie Flußläufen oder Gebirgszügen.
3. Beachtung gewachsener Verkehrs- und Kommunikationsrouten.
4. Beachtung von *communities of interest*.
5. Möglichst geringe Veränderung zuvor bestehender Wahlkreisgrenzen.

Darüber müssen die Kriterien des geographischen Zusammenhangs und der Kompaktheit berücksichtigt werden, sie ergeben sich jedoch nur implizit aus dem *Electoral Act of New Zealand*.

Anpassungen der Wahlkreisgrenzen an die aktuelle Bevölkerungsentwicklung müssen alle fünf Jahre vorgenommen werden. Somit tritt das Problem der Wahlkreiseinteilung regelmäßig auf. Seine Komplexität ist bedingt durch die Tatsache, daß die Wahlkreise auf Basis von sehr kleinen statistischen Raumeinheiten (sog. *meshblocks*) gebildet werden, die jeweils höchstens 200 Einwohner umfassen. In Neuseeland sind ungefähr 35,000 solcher *meshblocks* definiert.

Aufgrund des Problemumfangs wird eine erste Wahlkreiskarte mit Unterstützung von OR-Methoden erstellt. Ausgehend davon wird durch einen manuellen Planungsprozeß und Diskussion innerhalb der Wahlkreiskommission die Wahlkreiseinteilung schließlich festgelegt (George et al. [42]).

2.1.1.4 Italien

In Italien werden 75 Prozent der 630 Parlamentsabgeordneten in Einmannwahlkreisen direkt gewählt, die restlichen 25 Prozent nach dem Verhältniswahlrecht. Wegen des Prinzips des gleichen Wertes der Stimmen muß die Wahlkreiskarte regelmäßig hinsichtlich der Bevölkerungsentwicklung überprüft werden (o.V. [39, S. 655]).

In Ricca und Simeone [92] werden alternative Wahlkreiseinteilungen für verschiedene Regionen Italiens hinsichtlich der Kriterien Bevölkerungsgleichheit, Kompaktheit und Übereinstimmung mit Verwaltungsgrenzen bewertet. Dabei werden nur geographisch zusammenhängende Wahlkreise gebildet.

2.1.1.5 Wales

Die Bildung von Wahlkreisen in Wales für Wahlen auf europäischer Ebene wird von Nygreen [81] studiert. Das (recht einfach strukturierte) Entscheidungsproblem besteht darin, die 38 bestehenden nationalen Parlamentswahlkreise zu vier Europa-Wahlkreisen zusammenzufassen; diese müssen geographisch zusammenhängend sein und der Zielsetzung gleicher Bevölkerungszahl soweit wie möglich entsprechen.

Um die Bildung von „bananenförmigen“ Wahlkreisen zu vermeiden, berücksichtigt die Studie zusätzlich ein einfaches Kompaktheitskriterium.

2.1.1.6 Überblick ausgewählter OR–Studien

Eine Auswahl der Vorschläge, das Wahlkreisproblem mit Hilfe von Operations Research Verfahren zu lösen, ist in Tabelle 2.1 aufgeführt. Angegeben ist, welche Kriterien in der jeweiligen Studie einbezogen wurden. Das Kriterium „traditionelle Grenzen“ umfaßt die Beachtung von bestehenden Verwaltungsgrenzen und / oder geographischen Barrieren. Kompaktheit ist in den USA politisch motiviertes Kriterium (Verhindern von gerrymandering), in anderen Ländern hat es eher praktische Bedeutung zur Berechnung einer Einteilung mit „vernünftig“ geformten Wahlkreisen. Unter „demographische Daten“ wird die Berücksichtigung von Informationen über die Bevölkerungsstruktur, z.B. hinsichtlich Rassenzugehörigkeit oder sozialer Schicht, zusammengefaßt.

2.1.2 Wahlkreiseinteilung in Deutschland

Im folgenden werden die Rahmenbedingungen und das Vorgehen bei der Wahlkreiseinteilung für die Wahl des Deutschen Bundestages dargestellt. Da dem Autor keine OR–Studie bekannt ist, die auf die deutsche Situation Bezug nimmt, soll ein Vergleich mit den zuvor geschilderten Verhältnissen in anderen Ländern klären, inwieweit die dafür entwickelten OR–Ansätze übertragbar sind.⁷

⁷Dieser Abschnitt beruht auf einer Studienarbeit, die im Sommer 2000 am Lehrstuhl für Anwendungen des Operations Research, Universität Karlsruhe, durchgeführt wurde.

Studie	Land	Bevölkerungs- gleichheit	geographischer Zusammenhang	traditionelle Grenzen	Kompaktheit	demograph. Daten
Hess et al. 1965 [50]	USA	+	+	-	+	-
Garfinkel&Nemhauser 1970 [41]	USA	+	+	-	+	-
Helbig et al. 1972 [49]	USA	+	+	+	+	-
Bodin 1973 [17]	USA	+	+	-	-	-
Smith et al. 1975 [106]	Neuseeland	+	+	-	+	-
Bourjolly et al. 1981 [18]	Kanada	+	+	-	+	+
Nygreen 1988 [81]	Wales	+	+	+	+	-
Hojati 1996 [52]	Kanada	+	-	-	+	-
George et al. 1997 [42]	Neuseeland	+	+	+	+	-
Ricca&Simeone [92]	Italien	+	+	+	+	-
Mehrotra et al. 1998 [71]	USA	+	+	+	+	-
Cirincione et al. 2000 [24]	USA	+	+	+	+	+

Tabelle 2.1: Ausgewählte OR-Studien zur Wahlkreiseinteilung; „+“ kennzeichnet die jeweils berücksichtigten Kriterien.

2.1.2.1 Regelung durch das Bundeswahlgesetz

Im deutschen Wahlsystem, das eine Mehrheits- und eine Verhältniswahlkomponente hat, wird die Hälfte der Bundestagsabgeordneten direkt gewählt. Dazu ist Deutschland in 299 Wahlkreise eingeteilt, diese Einteilung wird durch das Wahlkreisneueinteilungsgesetz von 1998 [43] festgelegt.⁸

Die Kriterien für die Wahlkreiseinteilung ergeben sich aus dem Bundeswahlgesetz. Sie sind dort wie folgt festgelegt (§3 Abs. 1 BWG, zitiert nach [20, 21]):

1. Die Ländergrenzen sind einzuhalten.
2. Die Zahl der Wahlkreise in den einzelnen Ländern muß deren Bevölkerungsanteil soweit wie möglich entsprechen.
3. Die Bevölkerungszahl eines Wahlkreises soll von der durchschnittlichen Bevölkerungszahl der Wahlkreise nicht um mehr als 15 vom Hundert nach oben oder unten abweichen; beträgt die Abweichung mehr als 25 vom Hundert, ist eine Neuabgrenzung vorzunehmen.
4. Der Wahlkreis soll ein zusammenhängendes Gebiet bilden.
5. Die Grenzen der Gemeinden, Kreise und kreisfreien Städte sollen nach Möglichkeit eingehalten werden.

Die unterschiedliche Gewichtung der Kriterien ergibt sich aus der jeweiligen Formulierung („muß“, „soll“, „soll nach Möglichkeit“). Aufgrund der ersten beiden Bestimmungen ist die Wahlkreiseinteilung in jedem Bundesland gesondert vorzunehmen, zuvor muß aufgrund der Bevölkerungszahlen die Anzahl der Wahlkreise jedes Bundeslandes festgelegt werden. Die zulässigen Grenzen für die Abweichung von der exakten Bevölkerungsgleichheit sind explizit angegeben. Dabei gibt es einen Toleranzbereich zwischen 15 und 25 Prozent, der zur Berücksichtigung der anderen Kriterien genutzt werden kann, soweit diese als vorrangig angesehen werden.

2.1.2.2 Änderungen der Wahlkreiseinteilung

Aufgrund von regionalen Veränderungen in den Bevölkerungszahlen ist es notwendig, die bestehende Wahlkreiseinteilung regelmäßig zu überprüfen. Diese Aufgabe hat in Deutschland die vom Bundespräsidenten ernannte ständige Wahlkreiskommission. Sie erstattet in jeder Legislaturperiode einen Bericht, gemäß §3, Abs. 5, Satz 3 BWG:

⁸Die vollständige Neueinteilung 1998 wurde nötig, weil die Anzahl der Abgeordneten ab der 15. Wahlperiode (Beginn voraussichtlich 2002) auf 598 festgelegt ist, während sie gegenwärtig 656 beträgt (jeweils ohne Überhangmandate) und es folglich bis zur Bundestagswahl von 1998 insgesamt 328 Wahlkreise gab.

„Die Wahlkreiskommission hat die Aufgabe, über Änderungen der Bevölkerungszahlen im Wahlgebiet zu berichten und darzulegen, ob und welche Änderungen der Wahlkreiseinteilung sie im Hinblick darauf für erforderlich hält. Sie kann in ihrem Bericht auch aus anderen Gründen Änderungsvorschläge machen.“

Änderungen der Wahlkreiseinteilung müssen per Gesetz vom Bundestag beschlossen werden. Ausgangspunkt ist dabei der Bericht der Wahlkreiskommission. Bis Mitte der neunziger Jahre war die Neigung des Gesetzgebers, den Änderungsvorschlägen zu folgen, eher gering. Seit der Bundestagswahl von 1994, bei der in erheblichem Umfang Überhangmandate anfielen, und aufgrund des Umstandes, daß dies durch große Ungleichgewichte in der Bevölkerungszahl der Wahlkreise mitbedingt war, wird dem Problem der Wahlkreiseinteilung eine erhöhte Aufmerksamkeit zuteil. So wird die erst seit 1998 bestehende Einteilung in 299 Wahlkreise schon in der laufenden Legislaturperiode Änderungen erfahren; davon sind möglicherweise mehr als 50 Wahlkreise betroffen (FAZ vom 10.11.2000 [36]).

Aus den Berichten der Wahlkreiskommission [109, 110] läßt sich die Tendenz erschließen, die Auswirkungen der nötigen Änderungen auf die Wahlkreiseinteilung so gering wie möglich zu halten. Selbst bei Hinzukommen oder Wegfall eines Wahlkreises in einem Bundesland, wird angestrebt, mit Änderungen in einem regional begrenzten Gebiet auszukommen. Die plausible Vorstellung, daß bei Veränderung der Anzahl an Wahlkreisen eine vollständige Neueinteilung im betroffenen Bundesland vorgeschlagen würde, trifft nicht zu. Ein Grund hierfür kann zum Beispiel darin gesehen werden, daß ein persönliches Verhältnis von Wahlkreis–Abgeordneten zu „ihrer“ Bevölkerung besteht, das möglichst nicht zerrissen werden soll. Nach Schreiber [95, S. 110] „sind Stabilität und Kontinuität der Wahlkreiseinteilung Voraussetzungen für die Effektivität der politischen Arbeit der Parteien und ihrer Funktionsträger“.

2.1.2.3 Übertragbarkeit der OR–Studien aus anderen Ländern

Die durch das Bundeswahlgesetz definierten Kriterien, die bei der Wahlkreiseinteilung zu beachten sind, entsprechen denen in Demokratien, für die Operations Research Modelle zum Wahlkreisproblem vorgeschlagen worden sind. Deshalb ist die Übertragbarkeit solcher Modelle grundsätzlich positiv einzuschätzen.

Auf der anderen Seite ist hervorzuheben, daß viele der OR–Verfahren dazu entworfen sind, eine Wahlkreiseinteilung „von Null ausgehend“ zu berechnen. Dies steht im Konflikt mit der Praxis in Deutschland, notwendige Änderungen durch möglichst geringe Modifikationen der Wahlkreiskarte zu realisieren. Daher erscheint es für die deutsche Situation angebracht, vor allem solche Verfahren in Betracht zu ziehen,

welche zur Modifikation einer existierenden Einteilung dienen. Insbesondere die in di Cortona et al. [29] ausführlich diskutierten, auf lokaler Suche basierenden Ansätze, wären hierbei in Betracht zu ziehen. Die in Teil III entwickelten Verfahren folgen jedoch der Mehrzahl der OR-Studien und berechnen Gebietsaufteilungen vollständig neu.

2.2 Verkaufsgebiete und Betreuungsgebiete für Vertreter

Die Planungsaufgabe der Einteilung von Verkaufsgebieten tritt in allen Unternehmen auf, die mit einem Verkaufs-Außendienst operieren und den einzelnen Außendienstmitarbeitern Kunden(-regionen) exklusiv zuordnen. Skiera [103, Kap. 1.1] stellt heraus, daß es sich um eine (hinsichtlich Kosten und Umsatz) bedeutsame und regelmäßig auftretende Planungsaufgabe handelt. Andererseits ist die im allgemeinen unüberschaubare Vielzahl der möglichen Lösungen ein starkes Hindernis bei der rein manuellen Bearbeitung des Problems. Deshalb wurden im Operations Research etliche Modelle und Verfahren vorgeschlagen, um Entscheidungsunterstützung zu leisten. Einige haben sich etabliert und dürften auch in der Praxis zum Einsatz kommen.

Die Fragestellung tritt jedoch nicht nur im Zusammenhang mit Verkaufstätigkeit auf. Auch die Betreuung von Kunden, organisatorischen Untereinheiten oder technischen Einrichtungen wird häufig durch Mitarbeiter geleistet, die in einem exklusiv zugeordneten geographischen Gebiet tätig sind. Die OR-Literatur behandelt ebenfalls das Problem der Einteilung solcher Betreuungsgebiete. Da oft vergleichbare Kriterien angelegt werden wie bei der Einteilung von Verkaufsgebieten, behandeln wir die Anwendungsbereiche gemeinsam. Entscheidend ist für uns der Aspekt, daß es sich um Gebiete handelt, die von einem „Vertreter“ (im weitesten Sinn) betreut werden, wobei dies mit vor-Ort-Tätigkeit, also dem Bereisen seines Gebietes verbunden ist.

2.2.1 Kriterien für die Einteilung von Verkaufs- und Vertretergebieten.

Die wesentlichen, in der Literatur zur Einteilung von Verkaufs- und Vertretergebieten genannten Aspekte untergliedern wir im folgenden in organisationsbezogene, geographische und tätigkeitsbezogene Kriterien.

2.2.1.1 Organisationsbezogene Kriterien

Anzahl der Gebiete. Häufig werden Vertretergebiete für eine gegebene Mitarbeiterzahl geplant. Dann ist eine feste Anzahl an Gebieten zu erstellen; viele Modelle der Literatur berücksichtigen diesen Aspekt. Jedoch wird auch oft auf die Interdependenz der Planungsaufgaben „Größe der Verkaufsmannschaft“ und „Gebietsaufteilung“ hingewiesen. Folglich sind auch Modelle vorgeschlagen worden, die von einer variablen Anzahl an Gebieten ausgehen.

Kleinste geographische Einheiten. Vertretergebiete werden in der Regel nicht auf der Basis einzelner (Kunden-)Adressen eingeteilt. Vielmehr werden zunächst kleine geographische Einheiten (KGE) gebildet und dann durch Zusammenfassung der KGE zu Gebieten eine Einteilung erzielt. Die dadurch gegebene organisatorische Vereinfachung hat zwei Aspekte.

1. Die Komplexität des Problems wird verringert, wenn es auf der Basis von KGE gelöst wird. Außerdem wird dadurch eine diskrete Struktur aufgeprägt, die in entscheidungsunterstützenden Verfahren ausgenutzt werden kann.
2. Daten auf der Basis von KGE sind oft einfacher zu gewinnen, als auf der Basis von Einzeladressen. Dies betrifft zum Beispiel Entfernungen oder Umsatzpotentiale.

Der Aspekt der Datenbeschaffung hat entscheidende Bedeutung, insbesondere beim Einsatz von OR-Verfahren. Durch die Wahl der KGE wird der Aufwand der Datenbeschaffung stark beeinflusst. Skiera [103] diskutiert für die Verkaufsgebietseinteilung in Deutschland die Vor- und Nachteile von Postleiteinheiten⁹ und politischen Gemeinden oder Kreisen als Basis für die Definition der KGE. Er kommt zum Ergebnis, „die Einteilung auf der Basis von Postleiteinheiten [...] langfristig als vorteilhafter einzustufen“ (Skiera [103, S. 15]). Ähnliche Überlegungen dürften für Vertretergebiete anderer Art gelten.

Exklusive Zuordnung. Die Forderung, jede KGE einem Gebiet exklusiv zuzuordnen, ist ebenfalls aus organisatorischen Überlegungen motiviert.

1. Es wird Transparenz über Zuständigkeiten erzielt.

⁹Mit *Postleiteinheiten* bezeichnet werden die 95 Postleitregionen (zwei führende Stellen der Postleitzahl), die 718 Postleitbereiche (drei führende Stellen) und die 8279 Postleitzahlengebiete (gesamte Postleitzahl) (siehe Skiera [103, S. 13]).

2. Konkurrenzeffekte innerhalb der Verkaufsmannschaft werden vermieden, wenn nur ein Außendienstmitarbeiter für die Kunden einer KGE zuständig ist. Außerdem werden so langfristige Kundenbeziehungen ermöglicht.
3. Auf der Basis von KGE anfallende Daten können direkt einem Mitarbeiter zugeordnet werden. Dies ist zum Beispiel bei der Leistungsbeurteilung von Nutzen.

Zusammenfassung. Das Problem der Einteilung von Verkaufs- und Vertretergebieten hat aufgrund von organisatorischen Überlegungen eine einfache Struktur: Eine endliche Menge von KGE ist in Teilmengen (die jeweils einem Gebiet entsprechen) aufzuteilen, jede KGE muß in genau einer Teilmenge enthalten sein. Diese kombinatorische Struktur macht das Problem gut für den Einsatz von OR-Verfahren zugänglich.

2.2.1.2 Geographische Kriterien

Kriterien, die die Geographie möglicher Vertretergebiete betreffen, sind vor allem dadurch motiviert, daß die Vertreter vor Ort tätig sind und ihr Gebiet bereisen. Es resultieren folgende Anforderungen (Zoltners und Sinha [115, S. 1242]):

1. Die Gebiete sollen geographisch zusammenhängend sein.
2. Geographische Gegebenheiten müssen berücksichtigt werden, zum Beispiel Hindernisse wie Flußläufe oder Gebirgrücken.
3. Verträglichkeit mit dem Verlauf von Fernstraßen soll gegeben sein.¹⁰

Vertreterstandorte. Da die Mitarbeiter ihr Gebiet bereisen, werden sie ihren Standort zweckmäßigerweise innerhalb des Gebiets haben. Soll eine neue Einteilung vorgenommen werden, ist zu klären, ob bisherige Standorte beizubehalten sind, oder neue Standorte simultan geplant werden sollen. Die Erreichbarkeit des Betreuungsgebietes kann im Grunde genommen nicht ohne Bezug auf den Standort bewertet werden. Die OR-Modelle zur Einteilung von Vertretergebieten beziehen deshalb in aller Regel Standorte ein, entweder feststehend oder als Planungsgröße.

¹⁰Damit ist wohl gemeint, daß die Anzahl der Schnittpunkte der Gebietsgrenzen mit einer solchen Straße klein sein soll und daß innerhalb eines Gebietes verlaufende Fernstraßen sich nach Möglichkeit auch dort kreuzen.

Entfernungen, Reisezeiten und Kompaktheit. Während es vor einigen Jahren noch üblich war, Entfernungen zwischen KGE mittels einer einfachen Formel aus Koordinaten abzuleiten, und daher auch diskutiert wurde, welches Distanzmaß aus Anwendungssicht angemessen ist (Cloonan [25]), kann heute davon ausgegangen werden, daß detaillierte Informationen über Reisezeiten in elektronischer Form verfügbar sind. Damit läßt sich ein realistisches Maß verwenden, wenn Gebiete einzuteilen sind, die bereist werden müssen.

In dieser Hinsicht sind die Modelle der Literatur jedoch zumeist universell verwendbar. Sie beziehen in der Regel Größen d_{iv} ein, welche die Entfernung oder auch Reisezeit von Standort i zu KGE v angeben. Es ist der Wahl des Anwenders überlassen, wie er diese Werte bestimmt.

Der Wunsch nach gut zu bereisenden Gebieten wird durch das Kriterium der *Kompaktheit* zum Ausdruck gebracht. Im Zusammenhang mit Vertretergebieten wird der Begriff in der Regel so verstanden, daß die Reisezeit innerhalb jedes Gebietes klein sein soll. Typischerweise ist ein Maß der Form $\sum_v d_{iv}$ zu minimieren, summiert wird dabei über die KGE des i -ten Gebietes.

2.2.1.3 Tätigkeitsbezogene Kriterien

Balancierung. Die geographischen Anforderungen an Vertretergebiete beziehen sich vor allem auf das Bereisen. Letzteres ist jedoch nur Mittel zum Zweck, die eigentliche Tätigkeit der Mitarbeiter stellt Verkaufen, Beratung, Wartung o.ä. dar. Im Hinblick auf diese Aufgaben ist in vielen Fällen das Kriterium der *Balancierung* wesentlich. Es bezeichnet keine Eigenschaft eines einzelnen Gebietes, sondern bezieht sich auf den Wunsch nach einer Gleichartigkeit der Gebiete hinsichtlich der auszuübenden Tätigkeit. Die Motivation dafür ist das Ziel einer Gleichbehandlung der Mitarbeiter. So sollen Verkaufsgebiete ein ähnliches Umsatzpotential aufweisen, damit alle Verkäufer gleiche Chancen haben. Genauso soll die Arbeitsbelastung, zum Beispiel gemessen als Anzahl erforderlicher Besuche, gleichmäßig auf die Gebiete verteilt sein. Dieses Kriterium wird auch bei nicht verkaufsbezogenen Tätigkeiten angewendet.

Abstrahiert man von der konkreten Fragestellung, so ist unter Balancierung die gleichmäßige Verteilung eines oder mehrerer Aktivitätsmaße über die Gebiete zu verstehen. Typischerweise wird dieses Maß aus KGE-bezogenen Werten aggregiert, in der Regel durch Addition. Bezeichnet man es als „Größe“ der Gebiete, entspricht der Balancierung die Schaffung möglichst gleichgroßer Gebiete.

Maximierung des Deckungsbeitrages. Im Zusammenhang mit Verkaufsgebieten hat die bisher in Theorie und Praxis vorherrschende Erstellung balancierter

Gebietseinteilungen immer wieder Kritik erfahren, verstärkt in jüngerer Zeit (Zoltners und Sinha [115, S. 1253], Skiera [104]). So führt Skiera [104, S. 724] aus, daß „Gleichartigkeit an sich kein Ziel darstellen kann“ und die „offensichtlich ökonomisch sinnvollere Vorgehensweise“ die Maximierung des Deckungsbeitrages ist. Die Umsetzung dieses Programms führt zu Optimierungsmodellen, die sich deutlich von denen mit Balance-Bedingung unterscheiden (siehe Drexl und Haase [31], Skiera und Albers [105] und Wartenberg [112] für solche Modelle).¹¹ Die vorliegende Arbeit fokussiert auf Balance-orientierte Modelle. Auch wenn für Verkaufsgebiete das Kriterium der Balancierung in Frage zu stellen ist, wird es für nicht verkaufsbezogene Tätigkeiten seine Bedeutung behalten. Darüberhinaus ist zu fragen, inwieweit die Deckungsbeitrags-maximierende Verkaufsgebietseinteilung die traditionelle Vorgehensweise ersetzen wird. Das Kriterium der Balancierung kann daher für die Gestaltung von Vertretergebieten nach wie vor als wesentlich eingeschätzt werden.

2.2.2 OR-Literatur zu Verkaufs- und Vertretergebieten

Ein Überblick der Literatur soll die vorangegangene Darstellung der Kriterien bei der Einteilung von Verkaufs- und Vertretergebieten ergänzen. Tabelle 2.3 wurde nicht mit dem Ziel einer vollständigen Auflistung aller Studien erstellt, vielmehr fanden die nach Einschätzung des Autors wichtigen Arbeiten Aufnahme in die Übersicht. Einen älteren Überblick ähnlichen Aufbaus geben Howick und Pidd [54]. Darüber hinaus stellt Wartenberg [112, Tab. 3.3] synoptisch für die meisten der Studien das jeweils benutzte Optimierungsmodell dar.

Unser Ziel ist es, hervorzuheben, welche Berücksichtigung die Kriterien des vorangegangenen Abschnittes in den Arbeiten finden. Tabelle 2.2 stellt die Kriterien und ihre in Tabelle 2.3 benutzten Ausprägungen zusammen. Ein „+“/„–“-Eintrag bedeutet, daß das jeweilige Kriterium in der Studie berücksichtigt /nicht berücksichtigt ist. Beim Kriterium der Balance wird zusätzlich durch „mult“ das simultane Balancieren mehrerer Aktivitätsmaße angezeigt.

Die Spalte „Bemerkungen“ in Tabelle 2.3 enthält Hinweise auf einen gängigen Namen des Modells und ggf. auf das Kapitel in dieser Arbeit, in welchem das Modell näher beschrieben und untersucht wird.

¹¹Die wesentliche Zutat solcher Modelle ist die Definition einer geeigneten, üblicherweise nichtlinearen, Response-Funktion, mit der der Umsatz in Abhängigkeit von Besuchszeit, Erreichbarkeit, Marketingmaßnahmen usw. auf Basis der KGE geschätzt wird. Die Summe dieser Funktion über alle KGE bildet die zu maximierende Zielfunktion des Modells.

Art	Kriterium	Ausprägungen in Tabelle 2.3
organisations- bezogen	Anzahl der Gebiete	fix / variabel
	Exklusive Zuordnung der KGE	+ / -
geographisch	Kompaktheit	+ / -
	geograph. Zusammenhang	+ / -
	Standorte	fix / variabel
tätigkeits- bezogen	Balance	mult / + / -
	Deckungsbeitrag / Umsatz	+ / -

Tabelle 2.2: Kriterien bei der Einteilung von Vertretergebieten

2.3 Andere Anwendungen

Die meisten Anwendungen von OR-Studien zum Thema Gebietsaufteilung stammen aus den Bereichen Wahlkreiseinteilung und Verkaufsgebietgestaltung. Jedoch sind auch Arbeiten mit anderen Anwendungen zu finden. Diese werden im folgenden besprochen.

2.3.1 Einzugsgebiete von sozial–sanitären Einrichtungen und Schulen

2.3.1.1 Sozial–sanitäre Bezirke

Bei der Bildung sozial–sanitärer Verwaltungsbezirke werden territoriale Einheiten (z.B. Gemeinden) zu Bezirken zusammengefaßt, um einen definierten Zusammenhang zwischen den vorhandenen medizinischen Versorgungseinrichtungen (Krankenhäuser, Zahnkliniken) und der Bevölkerung zu schaffen. Dabei nennt die Literatur (siehe Tabelle 2.4 auf Seite 32) vor allem folgende Kriterien:

1. Die Bevölkerungszahl jedes Bezirks muß innerhalb vorgegebener Schranken liegen.
2. Die Kapazität der Einrichtungen muß in jedem Bezirk zur medizinischen Versorgung der Bevölkerung ausreichen.
3. Jeder Bezirk soll sich über ein geographisch zusammenhängendes Gebiet erstrecken.
4. Die Erreichbarkeit der Einrichtungen für die Bevölkerung soll möglichst gut sein.

Studie	Anwendung	org.-bezogen		Kriterien			Bemerkung		
		Anz.	exkl.	geographisch	tätigk.-bezogen				
		f	+	Komp.	Zush.	Std.orte	Balan.	Deckb.	
Hess&Samuels 1971 [51]	Verkaufsgebiete (pharmazeut. Industrie) Vertretergebiete (technischer Service)	f	+	+	-	v	+	-	GEOLINE siehe 9.2.1
Easingwood 1973 [32]	Vertreter für Betreuung Einzelhandel	f	+	+	+	f	+	-	manuelles Verfah- ren
Shanker et al. 1975 [98]	Verkaufsgebiete	v	+	+	?	?	+	+	
Segal&Weinberger 1977 [97]	Vertretergebiete (technischer Service)	f	+	+	-	f	+	-	
Glaze&Weinberg 1979 [44]	Verkaufsgebiete	f	+	+	-	v	-	+	TAPS
Zoltners 1979 [114]	Verkaufsgebiete	f	?	+	+	v	+ / mult	+	drei verschied. Modelle
Marlin 1981 [68]	Versicherungsinspektoren	f	+	+	-	f	+	-	
Ronen 1983 [93]	Kundenbetreuung ohne Verkauf	f	+	+	-	f	+	-	
Zoltners&Sinha 1983 [115]	Verkaufsgebiete	f	+	+	+	f	mult	-	siehe 7.3.1
Fleischmann&Paraschis 1988 [35]	Vertreter für Betreuung Einzelhandel	f	+	+	-	v	+	-	siehe 7.2.6
Skiera&Albers 1994 [105]	Verkaufsgebiete	f	+	+	+	f	-	+	COSTA
Wartenberg 1997 [112]	Verkaufsgebiete	f	+	+	+	f	-	+	
Drexl&Haase 1999 [31]	Verkaufsgebiete	v	+	+	+	v	-	+	

Tabelle 2.3: Literatur zur Einteilung von Verkaufs- und Vertretergebieten; zu den Spalteneinträgen vgl. Tab. 2.2.

Die angelegten Kriterien sind denen bei Wahlkreis- und Verkaufsgebietseinteilung zu vergleichen. Kompaktheit als Maß für gute Erreichbarkeit und Balance der Bezirksgröße sind neben geographischem Zusammenhang die wesentlichen Zielsetzungen.

2.3.1.2 Schulbezirke

Durch die Festlegung von Einzugsbereichen von Schulen wird eine Zuordnung von Wohngebieten zu Schulen geschaffen. Dadurch wird neben vereinfachter Verwaltung erreicht, daß Schüler aus einer Wohnnachbarschaft in der Regel auch dieselbe Schule besuchen. Daneben ist im Rahmen von Entwicklungsprogrammen die simultane Planung von Schulstandorten und -kapazitäten mit den Einzugsbereichen zu leisten. Schulbezirke teilen also eine gegebene Region in kleinere Einheiten auf. Daher ist bei der Planung solcher Einzugsbereiche ein Gebietsaufteilungsproblem zu lösen, wobei folgende Kriterien maßgeblich sind (zu Literatur siehe Tabelle 2.4, Seite 32):

1. Die maximale oder durchschnittliche Entfernung bzw. Wegzeit der Schüler zu ihrer Schule soll möglichst klein sein.
2. Kapazitätsschranken sind zu beachten, um eine gleichmäßige Auslastung der Schulen zu erreichen.
3. Das Rassengleichgewicht soll gewahrt werden. Diese Zielsetzung wird in Studien aus den Vereinigten Staaten genannt.
4. Geographische Gegebenheiten, wie Zugehörigkeit zum gleichen Verwaltungsbezirk oder andererseits Trennung durch Flußläufe sind zu berücksichtigen.

Auch hier sind die angelegten Kriterien den bisher diskutierten ähnlich.

2.3.2 Zuständigkeitsbezirke von Serviceeinrichtungen

Ein Teil der Serviceeinrichtungen des öffentlichen Bereichs erbringen ihren Dienst nicht an einem festen Ort, sondern vielmehr verteilt über eine geographische Region. Ahituv und Berman [3] bezeichnen dies als *distributed service networks*. Beispiele dafür sind Straßenräumung im Winter, Abfall-Entsorgung oder das Rettungswesen.

Da solche Dienste in der Regel mit einer Flotte von Fahrzeugen (im folgenden *Server* genannt) operieren, kann es, je nach Arbeitsweise des Dienstes, erforderlich sein, die Region in Zuständigkeitsbezirke für die Server aufzuteilen. Jeder Server ist dann in einem abgegrenzten Gebiet für die Erbringung des Dienstes zuständig.

Für alle drei genannten Beispiele gibt es Arbeiten aus dem Operations Research, die sich mit der Gebietsaufteilung für solche Server beschäftigen. Im folgenden wird

auf die dabei angelegten Kriterien eingegangen. Grundsätzlich ist zu sagen, daß alle Arbeiten die starke Interdependenz der Gebietsaufteilung mit anderen Entscheidungen betonen, so zum Beispiel der Standort- oder Tourenplanung.

2.3.2.1 Straßenräumung im Winter

Zunächst wird man im Zusammenhang mit der Straßenräumung im Winter ein Problem der Tourenplanung sehen. Dies betrifft jedoch vor allem die operative Ebene. Van Oudheusden et al. [83] betonen daneben das (übergeordnete) Problem der Bildung von Zuständigkeitsbezirken (*districting*). Der wesentliche Gesichtspunkt ist, daß diese Bezirke unabhängig von einander operieren: in jedem ist mindestens ein Fahrzeug-Depot angesiedelt und die Routenplanung erfolgt ausschließlich auf Bezirksebene.

Innerhalb der Bezirke sind verschiedene Aufgaben zu erfüllen, zum Beispiel vorbeugendes Streuen und Streuen bei bereits bestehender Eisglätte; sie erfordern unterschiedliche Salzmengen und daher unterschiedliche Touren. Deshalb kann sich die Einteilung der Bezirke nicht ausschließlich an den Touren orientieren. Van Oudheusden et al. [83] führen folgende Kriterien an:

1. Die Bezirke sollen balanciert (hinsichtlich Gesamtstraßenlänge) und kompakt sein.
2. Die Bezirke sollen die Planung guter Touren erlauben.
3. Die Bezirke sollen robust sein, das heißt, die verschiedenen auszuführenden Aufgaben gleichermaßen unterstützen.
4. Die Überlappung benachbarter Bezirke soll möglichst gering sein.

Bei der Planung von Bezirken von politischen Gemeinden als Flächeneinheiten auszugehen, wird dem Problem nicht gerecht. Vielmehr sollte die Zusammenfassung von elementaren Zyklen (*elemental cycles*) des Straßennetzes¹² zu Bezirken vorgenommen werden. Überlappung von Bezirken findet dann nur auf Straßenabschnitten statt, an denen zwei Bezirke aneinandergrenzen. Die Planung guter Touren wird möglich, da die Straßenabschnitte jedes Bezirkes weitgehend am Stück durchfahren werden können (eulerscher Graph).

¹²Es handelt sich hier um einen graphentheoretischen Begriff, der von einem (eingebetteten) ebenen eulerschen Graphen ausgeht. Die Flächen dieses Graphen können schachbrettartig eingeteilt werden; die elementaren Zyklen sind dann die begrenzenden Kanten der Flächen derjenigen Farbe, zu welchen nicht die äußere Fläche gehört.

Wird nur der Prozeß der Zusammenfassung der elementaren Zyklen zu Bezirken betrachtet, sind die angelegten Kriterien weitgehend in Übereinstimmung mit denen in anderen Anwendungen: Kompaktheit, Balance und geographischer Zusammenhang. Hinzu kommt, daß in die Bewertung der Bezirkseinteilung eine einfache Abschätzung der benötigten Anzahl an Streufahrzeugen einfließt (Muyldermans et al. [78]).

2.3.2.2 Abfall-Entsorgung

Nach Hanafi et al. [47] ist bei der Planung der städtischen Müllentsorgung zunächst die Bildung von Sektoren (*sectors*) vorzunehmen, in einem zweiten Schritt wird die Fahrtroute der Müllwagen innerhalb der Sektoren festgelegt. Im Hinblick auf die Kosten ist es vorteilhaft, Sektoren mit möglichst balancierten Routen zu bilden; das sind solche mit möglichst ausgewogenem Arbeitsaufwand bei der Entsorgung.

Drei Ziele werden daher betrachtet (Hanafi et al. [47]):

1. Der maximale zeitliche Aufwand aller Sektoren soll möglichst klein sein.
2. Der zeitliche Aufwand in jedem Sektor soll einem Zielwert möglichst nahe kommen.
3. Die Anzahl der Zusammenhangskomponenten jedes Sektors soll möglichst klein sein. Damit wird angestrebt, daß ein Sektor während des Einsammelns der Abfälle möglichst am Stück durchfahren werden kann, beziehungsweise nur selten verlassen werden muß.

Ein Sektor wird durch eine Menge von Straßenzügen gebildet, in denen an einem festen Wochentag der Abfall entsorgt wird. Es ist durchaus möglich, daß ein Straßenzug zu mehreren Sektoren gehört, also mehrfach in einer Woche angefahren wird. Hierdurch ist ein Unterschied zu anderen Anwendungen der Gebietsaufteilung gegeben: Die Sektoren insgesamt bilden keine Partition der Menge der Straßenzüge, sondern überdecken diese; im allgemeinen mehrfach. Wird jedoch ein fester Wochentag betrachtet, kann jeder Straßenzug nur zu einem Sektor gehören.

Das Problem der Sektorbildung bei der Abfallentsorgung hat also Parallelen zu anderen Gebietsaufteilungsproblemen, es werden Einheiten zu Sektoren zusammengefaßt und Kriterien der Balance und des geographischen Zusammenhangs sind wesentlich. Andererseits unterscheidet sich seine Struktur von Problemen mit echtem Aufteilungscharakter (das heißt Bildung einer Partition) und damit auch von den in der vorliegenden Arbeit untersuchten Modellen.

2.3.2.3 Server des Rettungswesens

Auch im Zusammenhang mit Diensten des Rettungswesens wird der Begriff des *districting* gebraucht (Swersey [107, Kap. 7]). Hier bezeichnet er die Bildung von Gebieten (sog. *primary response areas*; Baker et al. [7]), in welchen organisatorisch zusammengefaßte mobile Einheiten im Rettungswesen auf eintretende Unfälle, Brände usw. reagieren. Durch Gebietsaufteilung wird eine primäre Zuständigkeit der Server für die Ereignisse festgelegt.

Wesentliche Kenngrößen im Rettungswesen sind zum Beispiel die durchschnittliche bzw. die maximale erwartete Antwortzeit auf eintretende Ereignisse oder die Wahrscheinlichkeit, daß beim Eingang einer Meldung kein Server verfügbar ist. Ein wichtiges Kriterium ist auch die gleichmäßige Verteilung der Arbeitslast (*workload*) auf die Server.

Die Planung von *primary response areas* dient (neben anderen Zielen) dazu, die erwartete Antwortzeit zu minimieren und die Arbeitslast gleichmäßig aufzuteilen (Baker et al. [7]). Dies kann als Kompaktheits- und Balance-Zielsetzung interpretiert werden. Gegenüber den bisher vorgestellten Anwendungen der Gebietsaufteilung ist jedoch zu sagen, daß geographische Eigenschaften der Gebiete weniger berücksichtigt werden, und daß auf der anderen Seite die Modelle für das Rettungswesen in der Regel stochastisch sind.¹³ Deshalb ist ein deutlicher Unterschied zu den Modellen unserer Arbeit gegeben.

2.3.3 Clusterbildung für regional bezogene Daten

Ziel der Clusteranalyse ist die Zusammenfassung von Daten zu Klassen, die möglichst ähnliche Daten enthalten, und die untereinander möglichst verschieden sind. Wenn die Daten einen regionalen Bezug haben, zum Beispiel geographische Daten sind, und der „Ort“ eines Datums Berücksichtigung findet, kann die Clusteranalyse als Problem der Gebietsaufteilung im weiteren Sinne aufgefaßt werden.

Tatsächlich sind der Literatur zur Clusteranalyse Begriffe wie *regional clustering* (Maravalle und Simeone [66]) oder *contiguity-constrained clustering* (Gordon [45], Murtagh [75], Murtagh [76]) zu finden, denen die Vorstellung der Clusterbildung von Daten mit Bezug auf einen (geographischen) Ort oder auf eine Nachbarschaftstruktur zugrundeliegt. Jedoch handelt es sich häufig nicht um die Bildung geographischer Gebiete im engeren Sinne, der Ort eines Datums kann einfach ein Punkt in einem

¹³So wird das Eintreten von Ereignissen in kleinen Flächeneinheiten oft als Poisson-Prozeß modelliert und die Verfügbarkeit der Server über ein Warteschlangen-Modell erfaßt (siehe zum Beispiel Filippi und Romanin-Jacur [34]).

Koordinatensystem sein, zum Beispiel in der Bildverarbeitung der „Ort“ eines Bildpunktes (Pixel).

Dieser Unterschied zu den zuvor beschriebenen Anwendungen könnte jedoch außer acht gelassen werden, sobald man sich auf die Modellebene begibt. Wesentlicher erscheint die Abgrenzung auf Basis der Ziel-Kriterien.

1. Die Zusammenfassung ähnlicher Daten zu Clustern ist dem Ziel kompakter Bezirke dann vergleichbar, wenn in die Messung der Ähnlichkeit ein Wert für die räumliche Entfernung eingeht.¹⁴
2. Die Nebenbedingung, zusammenhängende Bezirke zu bilden, entspricht derjenigen für Cluster beim *contiguity-constrained* Clustering.
3. Das Balance-Ziel der Bildung gleichgroßer Bezirke fordert *äußere* Ähnlichkeit der Bezirke, dagegen sollen Cluster aus Daten gebildet werden, die homogen sind; es wird somit *innere* Ähnlichkeit angestrebt.

Vor allem aufgrund des letzten Unterschiedes fällt das Clustern von räumlichen Daten nicht unter die in dieser Arbeit untersuchten (Modelle für) Gebietsaufteilungsprobleme. Auch wenn, wie in Maravalle und Simeone [66], die Cluster aus geographischen Gebietseinheiten gebildet werden, und somit Gebieten entsprechen, ist mit der inneren Ähnlichkeit eine grundsätzlich andere Zielsetzung gegeben.¹⁵

2.3.4 Überblick ausgewählter Studien

In Tabelle 2.4 ist eine Auswahl aus der OR-Literatur zu den in den vorangehenden Abschnitten vorgestellten Anwendungen der Gebietsaufteilung aufgeführt.

Aufgrund des breiten Anwendungsspektrums gibt es keinen einheitlichen Kriterienkatalog in diesen Arbeiten. Dennoch lassen sich in vielen Fällen Kriterien identifizieren, die aus unserer Sicht zu den Grundkriterien der Gebietsaufteilung gehören.

1. Die Menge der kleinsten Gebietseinheiten ist in Bezirke aufzuteilen (Partition).
2. Ein Maß für die Ausdehnung oder die geographische Geschlossenheit eines Bezirkes oder für die Nähe der Einheiten eines Bezirkes untereinander geht ein. (Kompaktheit)

¹⁴Überlicherweise wird ein solches Maß beim Clustern regional-bezogener Daten berücksichtigt, neben einem Maß der Ähnlichkeit der „eigentlichen“ Datenvektoren (siehe Murtagh [76]).

¹⁵Interessant ist jedoch, daß der in den Teilen II und III dieser Arbeit verfolgte Ansatz, Verfahren für die Gebietsaufteilung auf der Basis von Baumzerlegungsalgorithmen zu entwickeln, von Maravalle und Simeone [66] zuerst für das *regional clustering* benutzt wurde. Somit ist trotz Unterschieden in den Zielsetzungen eine verfahrensmäßige Verwandtschaft zur Clusterbildung gegeben.

3. Geographischer Zusammenhang der Bezirke wird gefordert.
4. Die Größe der Bezirke hinsichtlich eines anwendungsspezifischen Aktivitätsmaßes wird bewertet oder unterliegt Bedingungen. Im weitesten Sinne bezeichnen wir dies als Balance, häufig handelt es sich dabei auch um Kapazitätsrestriktionen.¹⁶

Tabelle 2.4 gibt wieder, inwieweit die in den Studien verwendeten Kriterien für die Gebietsaufteilung diese Grundkriterien umfassen.¹⁷ Damit wird das Verhältnis der Studien zur Sichtweise des Gebietsaufteilungsproblems in der vorliegenden Arbeit deutlich, da wir uns an einfachen Modellen für die Grundkriterien orientieren. Dies wird im folgenden Kapitel entwickelt.

¹⁶Die Kapazität kann für verschiedene Serviceeinrichtungen unterschiedlich sein, dadurch ist die Bildung verschieden großer Bezirke möglich. Trotzdem auch Kapazitätsrestriktionen unter dem Stichwort „Balance“ zu fassen, erscheint angemessen, da es sich um Bedingungen an die Bezirksgröße handelt.

¹⁷Der Eintrag „relax“ bedeutet, daß ein Boolesches Kriterium in abgeschwächter Form beachtet wird.

Studie	Anwendung	Partition	Kompaktheit	geographischer Zusammenhang	Balance / Kapazität
Andria et al. 1979 [6]	Soz.-san. Bezirke	+	+	+	+
Minciardi et al. 1981 [74]	Soz.-san. Bezirke	+	-	+	+
Ogryczak&Malczewski 1987 [82]	Soz.-san. Bezirke	relax	+	+	+
Palermo et al. 1977 [84]	Schulbezirke	+	+	-	+
Ferland&Guénette 1990 [33]	Schulbezirke	+	+	+	+
Muyldermans et al. 1999 [78, 77]	Bezirke für Straßenräumung	+	+	+	+
Hanafi et al. 1999 [47]	Bezirke für Abfallentsorgung	-	+	relax	+
Baker et al. 1989 [7]	Bezirke für Ambulanzfahrzeuge	+	+	relax	+
Simchi-Levi&Berman 1990 [101]	allg. Service-Bezirke (stoch.)	+	-	+	-
Filippi&Romanin-Jacur 1996 [34]	allg. Service-Bezirke (stoch.)	+	-	-	-
Maravalle&Simeone 1995 [66]	Clusterbildung	+	-	+	-
Maravalle et al. 1997 [67]	Clusterbildung	+	-	+	-

Tabelle 2.4: Ausgewählte OR-Studien zu verschiedenen Anwendungen der Gebietsaufteilung

Kapitel 3

Grundmodelle für die Gebietsaufteilung

Die Untersuchung des Problems der Gebietsaufteilung fokussiert in der vorliegenden Arbeit auf grundlegende Modellbildungen, also solche, die wenige wesentliche Aspekte des Aufteilen eines geographischen Gebietes in kleinere Einheiten einbeziehen. Nicht jede Situation, vor die sich der Praktiker gestellt sieht, wird sich mit solchen Modellen abbilden lassen. Eine Vielzahl zusätzlicher Bedingungen können in der jeweils vorliegenden Entscheidungssituation gegeben sein, die in den diskutierten Modellbildungen nicht berücksichtigt sind. Dennoch lassen mehrere Gründe die Untersuchung solcher hochaggregierten Modelle lohnenswert erscheinen.

1. Die in den Modellen eingefangenen Aspekte können eine *ausreichend gute Approximation* der realen Situation sein. Dies wird belegt durch viele Arbeiten, in denen mit solchen Modellen Entscheidungsunterstützung für reale Probleme geleistet worden ist.¹
2. Die Modelle können zum *Erzeugen einer Anfangslösung* eingesetzt werden, die von Hand, oder mit Hilfe speziell angepaßter Verfahren, verbessert wird; dabei fließen die zunächst unberücksichtigten Aspekte der Planungssituation ein.
3. Die Allgemeinheit der Modelle sichert ein *breites Anwendungsspektrum*. Es wurde im vorangehenden Kapitel dargestellt. So unterschiedliche Problemstellungen wie Wahlkreisgestaltung oder das Festlegen von Gebieten für Kundenbetreuer lassen sich mit demselben OR-Modell erfassen.²

¹In der Praxis übernommen wurden zum Beispiel Wahlkreise, die nach dem in George et al. [42] vorgestellten Modell berechnet wurden, oder Verkaufsgebiete gemäß dem Modell in Fleischmann und Paraschis [35]. Auf letzteres gehen wir in 7.2.6 ein.

²So wurde das GEOLINE-Modell ursprünglich für Wahlkreiseinteilung entworfen (Hess et al.

4. Die Grundmodelle können als *Prototypen* für die Entwicklung reichhaltigerer Modelle, die zusätzliche Aspekte der Praxissituation einbeziehen, verwendet werden. In aller Regel werden sich die Variablen und Beziehungen des Grundmodells auch im verfeinerten Modell wiederfinden.
5. Die Analyse der Modelle *erhellt die zugrundeliegende Entscheidungssituation* „Gebietsaufteilung“. Dies kann dem Entscheider wertvolle Informationen über das zu lösende Problem geben.³

Im folgenden wird auf Voraussetzungen und „Bausteine“ der im dritten Teil der Arbeit untersuchten Modelle für die Gebietsaufteilung eingegangen. Aufgrund der Beschränkung auf wesentliche Aspekte sprechen wir von *Grundmodellen*. Sie integrieren solche Kriterien, für die in Kapitel 2 eine anwendungsübergreifende Bedeutung erkennbar wurde.

Anschließend wird in Abschnitt 3.3 dargestellt, wie sich Probleme der Gebietsaufteilung im Softwaresystem GAMOR repräsentieren und Lösungen dafür berechnen lassen. In GAMOR sind Verfahren, die auf den Grundmodellen basieren, implementiert. Dies umfaßt sowohl etablierte Algorithmen aus der Literatur, als auch die im Verlauf der Teile II und III dieser Arbeit entwickelten Verfahren.

3.1 Annahmen in den Grundmodellen

3.1.1 Generelle Voraussetzungen

Kleinste Gebietseinheiten (KGE). Wir gehen davon aus, daß das aufzuteilende Gebiet als eine Menge V kleinster Gebietseinheiten (KGE) gegeben ist, und daß durch Zusammenfassen dieser KGE das Gebiet in Bezirke aufgeteilt wird. Obwohl die KGE flächenhafte Gebilde sind, werden sie zumeist als punktförmig lokalisiert angenommen. (Diese topologischen Voraussetzungen diskutieren wir im nächsten Abschnitt.)

Zentren. Jeder Bezirk ist durch einen geographischen Ort, sein *Zentrum*, charakterisiert. Die Modelle des Kapitels 7 setzen gegebene Zentren voraus; Verfahren zur Bestimmung von Zentren(-orten), zum Beispiel im Fall bekannter Bezirke, sind Gegenstand von Kapitel 9. Wir bezeichnen die Menge der Zentren mit I .

[50]), dann für Verkaufsgebiete übernommen (Hess und Samuels [51]) und erweitert (Fleischmann und Paraschis [35]), und schließlich in den vergangenen Jahren wieder (mit Ergänzungen) für die Wahlkreiseinteilung benutzt (George et al. [42], Hojati [52]).

³Hier sei als Beispiel auf Satz 7.13 vorgegriffen, der eine a priori Aussage macht über die maximale Verletzung der exakten Gleichheit der Bezirksgrößen; letztere ist in der Regel nicht realisierbar.

Anzahl Bezirke. Alle im weiteren Verlauf diskutierten Modelle gehen von einer vorgegebenen Anzahl zu bildender Bezirke aus. Wir bezeichnen diese Anzahl durchgängig mit p . Die Menge der Zentren ist häufig $I = \{1, \dots, p\}$.

Eindeutige Zuordnung der KGE. Jede KGE soll genau einem Bezirk zugewiesen werden. Dazu werden Bezirke $B_i, i \in I$, gebildet, die eine Partition der Menge V darstellen, also

$$\emptyset \neq B_i \subset V, \quad B_i \cap B_j = \emptyset, \quad i, j \in I, i \neq j, \quad \text{und} \quad \bigcup_{i \in I} B_i = V.$$

Balancierung. Die zu bildenden Bezirke sollen hinsichtlich eines Größen-Attributes möglichst gleich groß sein. Dieses Attribut wird additiv aggregiert aus entsprechenden Attributen der KGE. Wir gehen also davon aus, daß zu jeder KGE $v \in V$ ein positives Attribut $w_v \in \mathbb{R}_{>0}$ gehört, und daß die Größe des Bezirks $B_i \subset V$ durch

$$w(B_i) = \sum_{v \in B_i} w_v$$

gegeben ist. Mit

$$\mu = \frac{1}{p} \sum_{v \in V} w_v$$

bezeichnen wir die mittlere Größe der Bezirke. Im Idealfall sollte jeder Bezirk eine Größe von μ haben. Diese Zielsetzung bezeichnen wir als *Balance*-Bedingung. Sie läßt sich in der Regel nicht gleichzeitig mit der eindeutigen Zuordnung aller KGE erreichen. Deshalb streben die Modelle danach, ein möglichst hohes Maß an Balance der Bezirksgrößen zu erzielen.

Zusammenhang. Jeder Bezirk soll aus einer geographisch zusammenhängenden Landfläche bestehen. Ein Modell, das diese Forderung explizit berücksichtigt, benötigt Informationen darüber, welche KGE benachbart sind. Wird das Benachbartsein mittels eines Graphen⁴ $G = (V, E)$, dessen Ecken die KGE sind, beschrieben, so muß das Modell diesen Graphen einbeziehen. In den einfacheren Modellen des Kapitels 7 ist dies nicht realisiert. Zusammenhängende Bezirke können deswegen nicht zugesichert werden, oft werden sie jedoch indirekt als Folge von Kompaktheit erreicht. Modelle, die den Graphen G explizit berücksichtigen, werden in Kapitel 8 vorgestellt.

⁴Zu graphentheoretischen Begriffen in dieser Arbeit siehe S. 229.

Kompaktheit. Die Zielsetzung einer kompakten Gebietsaufteilung wird modelliert, indem die Minimierung eines distanzbasierten Maßes angestrebt wird. Ist d_{iv} die Entfernung zwischen der KGE v und dem Zentrum des Bezirkes B_i , so soll die gewichtete totale Entfernung

$$\sum_{i=1}^p \sum_{v \in B_i} w_v d_{iv}$$

möglichst klein sein.

3.1.2 Topologische Voraussetzungen

Gebietsaufteilungsprobleme „leben“ in der Ebene, beziehen Orte und Entfernungen ein und haben deshalb eine topologische Komponente. Im Hinblick auf die Modellbildungen ist es dabei sinnvoll, zwei verschiedene Grundstrukturen zu unterscheiden: Ebene Modelle und Netzwerk-Modelle.⁵

Im ersten Fall werden KGE und Zentren als Punkte der Ebene betrachtet und die Entfernung zwischen ihnen mittels Entfernungsmessung in der Ebene bestimmt. Hierfür kommt zum Beispiel die euklidische Distanz in Betracht.

In Netzwerk-Modellen entsprechen die KGE den Ecken eines Graphen, dessen Kanten positive Werte als Längen zugeordnet sind. In der Regel ist der Graph in die Ebene eingebettet und Kanten verlaufen zwischen benachbarten KGE. Die Entfernung zwischen zwei KGE wird gemessen als Länge des kürzesten Weges im Graphen, der diese KGE verbindet. Jedes Zentrum ist ebenfalls mit einer Ecke des Graphen assoziiert und entspricht somit dem Ort einer KGE. Dadurch wird es möglich, Entfernungen zwischen Zentren und KGE zu bestimmen.

Aus Sicht des Praktikers mag der semantische Unterschied zwischen diesen beiden topologischen „settings“ nicht sehr bedeutsam sein. Auf die Modelleigenschaften wirkt er sich aber deutlich aus. Dies wird bereits deutlich, wenn man sich vor Augen führt, daß der Graph zur Entfernungsmessung in Netzwerk-Modellen mit dem Nachbarschaftsgraph der KGE übereinstimmt (siehe den Stichpunkt „Zusammenhang“ in 3.1.1). Deshalb untersuchen wir ebene Modelle und Netzwerk-Modelle in gesonderten Kapiteln (siehe Tabelle 3.1).

⁵Im (viel umfangreicheren) Forschungsgebiet der Standortplanung mit OR-Methoden wird in der Regel dieselbe Unterscheidung zwischen ebenen Modellen und Netzwerk-Modellen vorgenommen (siehe z.B. Domschke und Drexl [30]); dies vor allem, weil sich jeweils ganz unterschiedliche Strukturen ergeben und auch unterschiedliche Verfahren erforderlich sind.

	Modelle in der Ebene (Kapitel 7)	Modelle auf Netzwerken (Kapitel 8)
KGE, Zentren	Punkte in der Ebene	Ecken des Nachbarschaftsgraphen der KGE
Distanzmessung	z.B. euklidisch oder quadriert-euklidisch	kürzeste Wege im Graphen
Zusammenhang	nicht berücksichtigt	Teil des Modells

Tabelle 3.1: Topologische Modelleigenschaften

3.2 Modell-Typen

Die Modellbildungen im OR für Probleme der Gebietsaufteilung sind überwiegend Optimierungsmodelle. Dabei lassen sich im wesentlichen drei Typen identifizieren: *location-allocation* Ansätze, *set-partitioning* Ansätze und heuristische Verfahren, die keine Methoden der mathematischen Programmierung einbeziehen. Alle drei Typen werden im folgenden kurz charakterisiert und jeweils Stärken und Schwächen benannt. Die in Teil III behandelten Modelle sind überwiegend dem *location-allocation* Typ zuzuordnen.

3.2.1 Location-allocation Modelle

Location-allocation Modelle orientieren sich an einer Sicht des Gebietsaufteilungsproblems, die verbal folgendermaßen wiedergegeben werden kann:

Bestimme p Zentren(orte) und ordne diesen kleinste Gebietseinheiten eindeutig zu, so daß zusammenhängende Bezirke mit einem möglichst hohen Maß an Kompaktheit und Balance erzielt werden.

Das Problem der Festlegung von Zentren und das Zuordnen von KGE zu diesen Zentren sind zwei Schritte zur Bestimmung einer Gebietsaufteilung. Diese Schritte können nacheinander, ggf. wiederholt, durchlaufen werden, oder auch in *einem* Modell zusammengefaßt sein (siehe Kapitel 9).

Unter Umständen ist jedoch das Festlegen von Zentren nicht erforderlich, zum Beispiel wenn diese den Heimatorten von Vertretern entsprechen und ein Umziehen nicht in Frage kommt. Dann ist es vor allem der Zuordnungsschritt, der modellmäßig erfaßt werden muß.

Der wesentliche Vorteil des *location-allocation* Ansatzes ist die Möglichkeit, auch sehr große Probleme mit vielen KGE zu behandeln. Dies zeigen zum Beispiel George

et al. [42], die Wahlkreise für Neuseeland auf der Basis von 35000 KGE erstellen. Sie führen das Zuordnungsproblem im Kern auf das Lösen eines Netzwerkfluß-Problems zurück, für welches die genannte Größenordnung leicht zu beherrschen ist.

Unter den in Abschnitt 3.1 aufgeführten Voraussetzungen läßt sich das Problem der Gebietsaufteilung gut mit dem location-allocation Ansatz erfassen. Es sind auch Erweiterungen denkbar, wie zum Beispiel das Balancieren mehrerer Attribute (Zoltners und Sinha [115]).

Dennoch ist der Ansatz nicht sehr flexibel. Um ein lineares Programm in den die Zuordnungsaufgabe beschreibenden Variablen

$$x_{iv} = \begin{cases} 1 & \text{KGE } v \text{ wird Zentrum } i \text{ zugeordnet,} \\ 0 & \text{sonst} \end{cases}$$

zu erhalten, müssen die einbezogenen Größen sich auf lineare Terme in diesen Variablen abbilden lassen. Dies schränkt zum Beispiel die Verwendung von Kompaktheitsmaßen ein.⁶

Trotz seiner eingeschränkten Flexibilität bauen viele Arbeiten, die Gebietsaufteilungsprobleme mit Optimierungsmethoden lösen, auf dem location-allocation Ansatz auf (zum Beispiel Fleischmann und Paraschis [35], Hess et al. [50], Hess und Samuels [51], Hojati [52], Marlin [68], Ronen [93], Zoltners und Sinha [115]). Auch unter den einschränkenden Annahmen des Abschnitts 3.1 lassen sich somit realitätsnahe Modelle formulieren.

3.2.2 Set-partitioning Modelle

Die set-partitioning Modellen zugrundeliegende Sicht der Gebietsaufteilung lautet:

Generiere (viele) Kandidaten-Bezirke als Teilmengen der KGE und bestimme eine „gute“ Partition der Gesamtmenge der KGE in solche Teilmengen.

Wiederum sind zwei Verfahrensschritte zu erkennen, zunächst das Erzeugen von Bezirken (d.h. KGE-Teilmengen), die als Kandidaten für eine gute Gebietsaufteilung in Frage kommen, und dann das Lösen eines set-partitioning Problems aufbauend auf diesen Teilmengen. Beide Schritte können nacheinander ausgeführt werden (Garfinkel und Nemhauser [41], Nygreen [81]) oder simultan (Mehrotra et al. [71]).

Der Vorteil des set-partitioning Ansatzes ist seine größere Flexibilität gegenüber den location-allocation Modellen. Durch Realisierung eines geeigneten Generators

⁶So sind Maße, die Durchmesser oder Umfang der Bezirke berücksichtigen, nicht durch lineare Terme in den x_{iv} auszudrücken.

für die Kandidatenbezirke lassen sich vielfältige Anforderungen an gute Bezirke abbilden.⁷

Auf der anderen Seite sind mit set-partitioning Modellen aufgrund der stark anwachsenden kombinatorischen Komplexität nur kleine Problemstellungen lösbar. Die in der Literatur gelösten Beispiele haben sämtlich weniger als 100 KGE. Trotzdem kann die Anzahl der Kandidatenbezirke schon sehr groß sein, und somit ein rechnerisch aufwendiges set-partitioning Problem zu lösen sein.⁸ Hierin ist ein entscheidender Nachteil des Ansatzes zu sehen.

3.2.3 Heuristische Verfahren

Mit heuristischen Verfahren sind an dieser Stelle Methoden gemeint, die nicht auf dem Lösen von mathematischen Programmen beruhen. In Anlehnung an Ricca und Simeone [92] führen wir die wichtigsten Ideen auf:

- Beim *Eat-up* Ansatz wird ein Bezirk nach dem anderen, ausgehend von jeweils einer KGE am Rand des (noch) aufzuteilenden Gebietes, durch sukzessives Hinzufügen benachbarter KGE gebildet. Ein Bezirk ist fertig, wenn er eine ausreichende Größe hat. (Ein Beispiel einer solchen Heuristik ist in Mehrotra et al. [71] zu finden.)
- Beim *Zusammenfassen von Teilbezirken* wird zunächst jede KGE als ein Teilbezirk aufgefaßt. Dann werden in jedem Schritt benachbarte Teilbezirke vereinigt (wenn dadurch kein zu großer Teilbezirk entsteht); das Verfahren bricht ab, wenn die gewünschte Anzahl an Bezirken gebildet ist. (Als Beispiel sei Deckro [28] genannt.)
- *Multi-kernel growth* ist mit dem location-allocation Ansatz verwandt. Zunächst werden einige KGE als „Samen“ ausgewählt; danach wird um jeden Samen ein Bezirk gebildet, indem KGE nach wachsender Entfernung hinzugefügt werden, bis der Bezirk die angestrebte Größe hat.
- Verfahren der *lokalen Suche* verändern eine Gebietsaufteilung durch Austausch von einzelnen KGE zwischen benachbarten Bezirken. Dabei wird eine Verbesserung der Aufteilung hinsichtlich der einbezogenen Zielsetzung(en) angestrebt.

⁷Zum Beispiel wird von Garfinkel und Nemhauser [41] ein Kompaktheitsmaß verwendet, das den quadrierten Durchmesser von Bezirken einbezieht.

⁸In Mehrotra et al. [71] werden für ein Problem mit 51 KGE und sechs Bezirken insgesamt 802 Kandidaten-Bezirke erzeugt, das heißt, es wird bereits ein set-partitioning Problem mittlerer Größe gelöst.

Gängige Erweiterungen der lokalen Suche wie *Simulated Annealing* oder *Tabu-Suche* kommen auch hier in Betracht (di Cortona et al. [29, S. 179]).

Wesentlicher Vorteil heuristischer Ansätze ist ihre große Flexibilität hinsichtlich der Integration und Berücksichtigung anwendungsspezifischer Kriterien. Außerdem ist ihre Implementierung in einer Programmiersprache in der Regel recht einfach auszuführen. Ein Rückgriff auf einen guten Solver für lineare Programme ist im Gegensatz zu den Verfahren, die auf mathematischer Programmierung beruhen, nicht nötig.

Auf der anderen Seite läßt sich das Verhalten solcher Heuristiken zumeist schlecht analysieren. Wie „gut“ die erzeugte Gebietsaufteilung sein wird, läßt sich a priori kaum feststellen. Qualitätsaussagen a posteriori werden in der Regel durch Erzeugen und Vergleichen *vieler* Lösungen gewonnen. Folglich können nur relative Qualitäten gemessen werden.

3.2.4 In dieser Arbeit untersuchte Modelle

In Teil III dieser Arbeit werden Modelle untersucht, die konzeptionell auf dem location-allocation Ansatz beruhen. Folgende Überlegungen begründen diese Entscheidung.

1. Alle in Abschnitt 3.1 dargestellten Modell-Aspekte lassen sich abbilden.
2. Große Probleme, denen sehr viele KGE zugrundeliegen, lassen sich mit dem location-allocation Ansatz handhaben.
3. Die Modelle sind aufgrund ihrer Struktur (mathematisches Programm) einer Analyse gut zugänglich.
4. Die vorzustellenden Verfahren sind ein Anwendungsfall der Methoden gleichmäßiger Baumzerlegung, die in Teil II entwickelt werden.

3.3 GAMOR: Ein System zur Berechnung von Gebietsaufteilungen

Um die in Teil III darzustellenden Verfahren zur Berechnung von Gebietsaufteilungen testen zu können, wurde das Softwaresystem GAMOR⁹ entwickelt. Neben der Implementierung der Algorithmen umfaßt es einfache Dateiformate zur Spezifikation von Gebietsaufteilungsproblemen; zusätzlich zur Ausgabe des Ergebnisprotokolls in

⁹GAMOR ist Akronym für „Gebiete aufteilen mit Operations Research-Methoden“.

Textform ist eine graphische Ausgabe zur Visualisierung der berechneten Aufteilung möglich.

Bezogen auf Ein- und Ausgabemöglichkeiten wurde damit bei weitem nicht der Funktionsumfang kommerziell verfügbarer Softwaresysteme zur Bearbeitung regional bezogener Daten realisiert. Diese *Geographischen Informationssysteme*¹⁰ bieten neben den sehr weit entwickelten Möglichkeiten zur Darstellung geographischer Daten in Form von thematischen Karten zum Teil auch Funktionen zur Berechnung von Gebietsaufteilungen. Hier dürften jedoch einfache Heuristiken vorherrschend sein. In GAMOR dagegen liegt der Schwerpunkt auf der Methodenseite.

3.3.1 Realisierung von GAMOR

Bei der Realisierung von GAMOR waren die folgenden Zielsetzungen wesentlich.

1. Das System soll einen einfachen Zugriff auf die implementierten Algorithmen erlauben. Insbesondere soll es möglich sein, auch viele Probleme in Folge ohne manuelle Eingriffe zu berechnen; dies ist zum Test von Verfahren in der Regel erforderlich.
2. Das System soll sehr einfach um neue Algorithmen erweitert werden können. Dies ist notwendig, da die Entwicklung von Verfahren und ihre Implementierung Hand in Hand gehen. Insbesondere soll es möglich sein, Verfahren aus bestehenden Teilen nach dem Baukastenprinzip zusammensetzen zu können.

Aus diesen Anforderungen resultierten die folgenden Entscheidungen.

1. Die Steuerung von GAMOR erfolgt über Kommandoparameter und Parameterdateien. Dadurch steht unter dem Betriebssystem UNIX ohne Implementierungsaufwand eine Kontrollsprache in Form von Shellskripten oder auch der Sprache *Perl* (Wall et al. [111]) zur Verfügung. Die Realisierung einer Fensterumgebung als Benutzerschnittstelle wurde demgegenüber für entbehrlich gehalten.
2. Die Implementierung wurde in der objektorientierten Sprache *Eiffel* (Meyer [73]) vorgenommen. Durch die durchdachten Konzepte dieser Sprache wird eine sichere Wiederverwendung fertiger Softwareteile in sehr vorteilhafter Weise unterstützt. Vor allem in späteren Phasen der Arbeit an GAMOR konnte davon profitiert werden; die Implementierung von OR-Verfahren nach dem Baukastenprinzip und in kürzester Zeit war durchaus möglich.

¹⁰Eine Marktübersicht und Beurteilung der um das Jahr 1996 in Deutschland verfügbaren Systeme gibt Wartenberg [112].

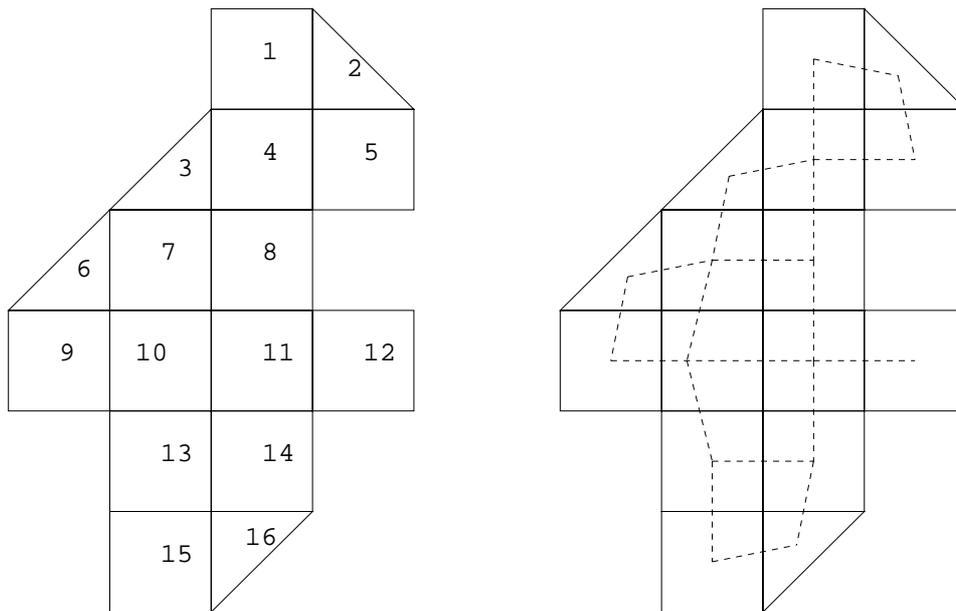


Abbildung 3.1: Kleinste Gebietseinheiten des Tafelbeispiels und ihr Nachbarschaftsgraph

3.3.2 Ein Beispiel für die Verwendung von GAMOR

Im folgenden wird anhand eines „Tafelbeispiels“ die Verwendung von GAMOR demonstriert. Dazu wird erläutert, wie das gegebene Problem sich darstellen läßt, und wie sich ganz verschiedene Gebietsaufteilungen ergeben können, abhängig davon, welcher Algorithmus ausgewählt wird.

3.3.2.1 Definition der Problemstellung

Abbildung 3.1 zeigt das aufzuteilende „Gebiet“. Die 16 KGE sind Dreiecke oder Quadrate. In GAMOR ist der Umriß der KGE in Form von Polygonzügen definierbar. Desweiteren kann zu jeder KGE eine Nummer, ein Name und die Lage eines Zentrums,¹¹ das zum Beispiel zur Entfernungsmessung benutzt wird, angegeben werden. Für unser Beispiel wurde folgende Datei verwendet.

```
ANZAHL = 16
{ NUMMER = 1 NAME = "A-Dorf"
  { (3000,1000) (4000,1000) (4000,2000) (3000,2000) } }
{ NUMMER = 2 NAME = "B-Dorf"
  { (4000,1000) (5000,2000) (4000,2000) } }
{ NUMMER = 3 NAME = "C-Dorf"
```

¹¹Zentren von KGE haben mit den Bezirkszentren (siehe 3.1.1) nur indirekt zu tun. Das Zentrum einer KGE ist der Ort, an welchem sie punktförmig lokalisiert angenommen wird.

```

    { (3000,2000) (3000,3000) (2000,3000) } }
  { NUMMER = 4
    { (3000,2000) (4000,2000) (4000,3000) (3000,3000) } }
  { NUMMER = 5
    { (4000,2000) (5000,2000) (5000,3000) (4000,3000) } }
  { NUMMER = 6
    { (2000,3000) (2000,4000) (1000,4000) } }
  { NUMMER = 7
    { (2000,3000) (3000,3000) (3000,4000) (2000,4000) } }
  { NUMMER = 8
    { (3000,3000) (4000,3000) (4000,4000) (3000,4000) } }
  { NUMMER = 9
    { (1000,4000) (2000,4000) (2000,5000) (1000,5000) } }
  { NUMMER = 10 ZENTRUM = (2250,4500)
    { (2000,4000) (3000,4000) (3000,5000) (2000,5000) } }
  { NUMMER = 11
    { (3000,4000) (4000,4000) (4000,5000) (3000,5000) } }
  { NUMMER = 12
    { (4000,4000) (5000,4000) (5000,5000) (4000,5000) } }
  { NUMMER = 13
    { (2000,5000) (3000,5000) (3000,6000) (2000,6000) } }
  { NUMMER = 14
    { (3000,5000) (4000,5000) (4000,6000) (3000,6000) } }
  { NUMMER = 15
    { (2000,6000) (3000,6000) (3000,7000) (2000,7000) } }
  { NUMMER = 16
    { (3000,6000) (4000,6000) (3000,7000) } }

```

Das Format ist weitgehend selbsterklärend. Wird für eine KGE kein Zentrum vorgegeben, so wird von GAMOR der Schwerpunkt der durch den Polygonzug umschlossenen Fläche als Zentrum angenommen.¹² Das „Koordinatensystem“ in GAMOR ist vereinbarungsgemäß ein 10000×10000 Quadrat.

In Abbildung 3.1 wird rechts der Nachbarschaftsgraph der KGE gezeigt; die gebrochenen Strecken entsprechen den Kanten und verbinden die Zentren der KGE. Dieser Graph wird in einer eigenen Datei definiert, sie hat folgende Gestalt.

ECKENZAHL=16	11 12 ; 1000	10 13 ; 1000
KANTENZAHL=21	11 14 ; 1000	13 14 ; 1000
1 2 ; 849.837	14 16 ; 849.837	13 15 ; 1000
1 4 ; 1000	3 4 ; 849.837	15 16 ; 849.837
2 5 ; 849.837	3 7 ; 849.837	6 7 ; 849.837
4 5 ; 1000	7 8 ; 1000	6 9 ; 849.837
4 8 ; 1000	7 10 ; 1000	9 10 ; 1000
8 11 ; 1000	10 11 ; 1000	

Zu jeder Kante ist ihre Länge angegeben. Diese kann, anders als hier, von der euklidischen Entfernung der Zentren abweichen. So können zum Beispiel Straßendistan-

¹²Siehe 7.1.1 für eine Formel zur Schwerpunktberechnung.

zen oder Fahrtzeiten in mit GAMOR berechneten Gebietsaufteilungen berücksichtigt werden.

Die Definition des Nachbarschaftsgraphen in einer separaten Datei ist sinnvoll, da durchaus mehrere, unterschiedlich bewertete, Graphen zu einer Datei mit KGE-Daten gehören können.

In einer dritten Datei wird nun das eigentliche Gebietsaufteilungsproblem definiert.

```
KARTENDATEI = "beispiel"
ADJAZENZ_NUMMER = 1
ANZAHL_BEZIRKE = 5
ELEMFLATTRIBUTE = { 201; 102; 103; 201; 202; 103; 201; 202;
                    203; 201; 200; 201; 200; 202; 203; 100 }
```

Die ersten beiden Zeilen dienen der Auswahl von KGE-Daten und des ersten der zugehörigen Nachbarschaftsgraphen. Danach wird die Anzahl der zu bildenden Bezirke angegeben. In den letzten beiden Zeilen werden die zu den KGE gehörenden Größenattribute definiert.¹³ Sie sind hier so gewählt, daß sie, bis auf eine kleine Störung, proportional zur Fläche der KGE sind.

Die Problemstellung unseres Tafelbeispiels lautet damit:

Teile das Gebiet aus Abbildung 3.1 in fünf Bezirke auf, die möglichst gleiche Fläche haben und möglichst kompakt sind.

3.3.2.2 Berechnungsbeispiele mit GAMOR

(Die folgenden Berechnungsbeispiele dienen nur der Demonstration von GAMOR; es soll nicht der Versuch gemacht werden, die Ideen der in Teil III entwickelten und dargestellten Algorithmen zu erläutern.)

In einem ersten Versuch rufen wir GAMOR auf:

```
gamor -p tafelbsp -v1 -n5 -d2
```

Der Kommandoparameter `-v` wählt das Verfahren; hier das mit der Nummer Eins. Es ist das klassische GEOLINE-Verfahren von Hess und Samuels [51] (siehe 9.2.1). Der Parameter `-n` legt die maximale Anzahl der Iterationen fest. Der Parameter `-d2` legt fest, daß Entfernungen über kürzeste Wege im Nachbarschaftsgraphen gemessen werden sollen.¹⁴

¹³In GAMOR werden die KGE als *Elementarflächen* bezeichnet. Diese Benennung verwendet Hanssmann [48]. Wie das Beispiel zeigt, läßt sie sich nicht schön abkürzen und wird deswegen in dieser Arbeit nicht verwendet.

¹⁴Dies ist im GEOLINE-Modell ursprünglich nicht vorgesehen (es werden quadriert-euklidische Distanzen verwendet); jedoch ist es eine naheliegende Modifikation.

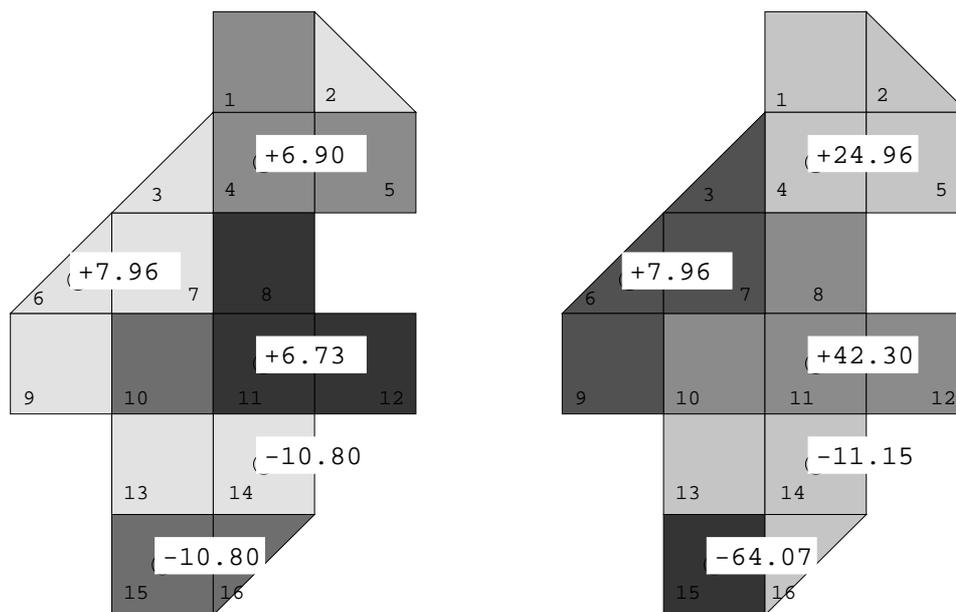


Abbildung 3.2: Gebietsaufteilungen berechnet mit dem GEOLINE-Verfahren und mit dem Verfahren von Zoltners und Sinha [115].

Die berechnete Gebietsaufteilung ist im linken Teil der Abbildung 3.2 zu sehen. Die gebildeten Bezirke sind $B_1 = \{1, 4, 5\}$, $B_2 = \{3, 6, 7, 9\}$, $B_3 = \{8, 11, 12\}$, $B_4 = \{2, 13, 14\}$ und $B_5 = \{10, 15, 16\}$. Die hell unterlegten Zahlen geben die prozentuale Abweichung von der mittleren Bezirksgröße an. Das Maximum beträgt absolut 10.8%. Unbefriedigend ist, daß zwei Bezirke (B_4 und B_5) nicht zusammenhängend sind.

Daher machen wir einen zweiten Versuch. Nun verwenden wir Verfahren Nummer Zwei, es wurde von Zoltners und Sinha [115] vorgeschlagen (siehe 7.3). Das Ergebnis des Aufrufs

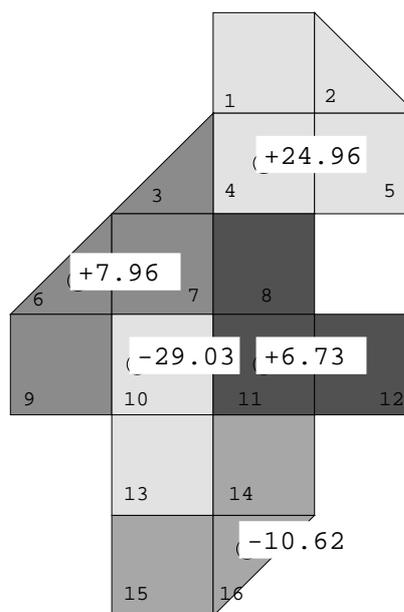
```
gamor -p tafelbsp -v2 -n5 -d2
```

ist im rechten Teil von Abbildung 3.2 dargestellt. Es wurden die Bezirke $B_1 = \{1, 2, 4, 5\}$, $B_2 = \{3, 6, 7, 9\}$, $B_3 = \{8, 10, 11, 12\}$, $B_4 = \{13, 14, 16\}$ und $B_5 = \{15\}$ berechnet. Nun sind die Bezirke zusammenhängend, (dies ist eine Folge des Verfahrens, vgl. Satz 8.1 auf S. 192), jedoch von sehr ungleicher Größe. Die maximale Abweichung beträgt absolut ungefähr 64% (Bezirk B_5).

Ein dritter Versuch mit dem auf gleichmäßiger Baumzerlegung beruhenden Verfahren nach Algorithmus 8.2 (S. 203) bringt ein weiteres Ergebnis. Die Gebietsaufteilung wird berechnet mit

```
gamor -p tafelbsp -v7 -m5 -d2
```

(Der Parameter `-m` legt eine Maximalzahl an Iterationen fest.) Diese Lösung vereinbart den Vorteil zusammenhängender Bezirke mit einer mäßigen Abweichung der Bezirksgrößen vom Mittel. Die Bezirke sind $B_1 = \{1, 2, 4, 5\}$, $B_2 = \{3, 6, 7, 9\}$, $B_3 = \{8, 11, 12\}$, $B_4 = \{10, 13\}$, $B_5 = \{14, 15, 16\}$.



Teil II

Gleichmäßige Baumzerlegung

Kapitel 4

Grundlagen

Nachdem in Teil I das Problem der Gebietsaufteilung aus Anwendungssicht dargestellt und Aspekte grundlegender Modellbildungen diskutiert wurden, wird in den folgenden Kapiteln mit der Untersuchung der *gleichmäßigen Baumzerlegung* die Basis für neue Algorithmen zur Gebietsaufteilung geschaffen.

Dabei ist die Untersuchung breit angelegt. Manche Ergebnisse gehen über das in späteren Kapiteln benötigte hinaus. Der Autor ist jedoch der Überzeugung, daß sich dieser „Umweg“ lohnt; aus zwei Gründen:

1. Das Problem der gleichmäßigen Baumzerlegung ist in der Literatur häufig untersucht worden, und dies völlig losgelöst vom Kontext der Gebietsaufteilung. Es existiert somit das Baumzerlegungsproblem als eigenständige Fragestellung, und zu diesem Forschungsfeld soll in Teil II beigetragen werden. Er ist so angelegt, daß die bisher aufgeworfenen Fragen nur noch den Hintergrund des Gedankengangs bilden, man kann die Kapitel 4 bis 6 auch für sich allein lesen.
2. Je nach persönlicher Interessenlage wird der Leser die Untersuchungen zum Problem der gleichmäßigen Baumzerlegung als eher „theoretisch“ empfinden, und schnell wieder zur Gebietsaufteilung zurück kommen wollen. Hierzu ist zu sagen, daß mit der späteren Anwendung der Baumzerlegungsverfahren in Teil III das Ziel verfolgt wird, dem Thema „gleichmäßige Baumzerlegung“ ein Anwendungsfeld hinzuzufügen, das bisher wenig wahrgenommen wurde. Dazu ist es aber nötig, zunächst die Fragestellung der gleichmäßigen Baumzerlegung in einer gewissen Gründlichkeit aufgearbeitet zu haben.

In diesem Kapitel wird die Thematik grundlegend angegangen. Zunächst werden das allgemeine Baumzerlegungsproblem und geeignete Lösungsverfahren diskutiert. In 4.2 werden Modelle für das gleichmäßige Zerlegen von Bäumen vorgestellt, viele davon sind zuvor in der Literatur untersucht worden. Abschnitt 4.3 erweitert be-

kannte Ergebnisse zur aus algorithmischer Sicht wichtigen Komplexitätsfrage. Die Sätze in 4.4 haben ergänzenden Charakter, hier geht es um Verallgemeinerungen des Baumzerlegungsproblems auf die Partitionierung von nicht zyklischen Graphen.

4.1 Das allgemeine Baumzerlegungsproblem

Zunächst ist es erforderlich, Begriffe und Notation einzuführen, die mit Bäumen im Zusammenhang stehen.

4.1.1 Bezeichnungen

Bäume sind Graphen¹ mit einer speziellen Struktur: Ist V die Eckenmenge und E die Kantenmenge, so ist der Graph $G = (V, E)$ ein *Baum* genau dann, wenn G zusammenhängend ist und $|E| = |V| - 1$ gilt. Eine äquivalente Definition für Bäume fordert: G ist zusammenhängend und zyklischfrei. Dies impliziert zum Beispiel, daß es in einem Baum mit $|V| \geq 2$ stets mindestens zwei Ecken vom Grad eins gibt, wir bezeichnen solche Ecken als *Endecken*. Für den Grad einer Ecke v schreiben wir im folgenden $\text{grad}(v)$.

4.1.1.1 Notation

Sei $T = (V, E)$ ein Baum. Ein *Teilbaum* T' von T ist ein nichtleerer zusammenhängender Subgraph von T , also selbst wieder ein Baum. Wir schreiben hierfür $T' \triangleleft T$. Ein Teilbaum T' ist eindeutig durch seine Eckenmenge $V(T') \neq \emptyset$ bestimmt. Häufig schreiben wir kurz $v \in T'$ anstelle von $v \in V(T')$. Ist V' Eckenmenge eines Teilbaumes von T , so schreiben wir oft $T(V')$ für diesen, von V' *induzierten*, Teilbaum.

Pfade. Der (eindeutig bestimmte) *Pfad* $P[u, v]$ von u nach v in einem Baum T ist die kürzeste Eckenfolge ($u = u_1, u_2, \dots, u_k = v$) so daß u_i und u_{i+1} in T adjazent sind, $i = 1, \dots, k - 1$.² Wir betrachten auch den *offenen* Pfad $P(u, v) := P[u, v] - \{u, v\}$ und die sinngemäß definierten Pfade $P[u, v)$ und $P(u, v]$. Man beachte, daß $P(u, v)$, $P[u, v)$ und $P(u, v]$ leer sein können.

Wurzelbäume. Ein *Wurzelbaum* ist ein Baum, in dem eine Ecke $v_0 \in V$ als *Wurzel* ausgezeichnet ist. Somit kann man durch Festlegung einer Wurzel einen jeden Baum zum Wurzelbaum machen. Wurzelbäume haben die für Algorithmen wichtige Eigenschaft, daß auf der Eckenmenge eine Ordnung besteht. Sie hängt von der Wurzel v_0

¹Zu graphentheoretischen Begriffen in dieser Arbeit siehe S. 229.

²Kantenrichtungen spielen hierbei keine Rolle.

ab und ist wie folgt definiert:

$$v \preceq_{v_0} u \iff v \in P[u, v_0].$$

Die Wurzel v_0 ist das minimale Element bezüglich dieser Ordnung, die maximalen Elemente werden als *Blätter* bezeichnet. Ein Wurzelbaum hat also stets mindestens ein Blatt.

Da man sich Wurzelbäume üblicherweise als an der Wurzel „aufgehängt“ vorstellt, bedeutet $v \prec_{v_0} u$ anschaulich, daß v „über“ u im Baum liegt. Der *Vorgänger* $\text{pre}_{v_0}(v)$ von $v \neq v_0$ ist die Ecke direkt über v , d.h. das erste Element von $P(v, v_0)$.

Der in v wurzelnde Teilbaum $T_v^{v_0}$ ist durch die Eckenmenge $\{u \in V : v \preceq_{v_0} u\}$ charakterisiert. Die Ecken $u \in T_v^{v_0} - \{v\}$ heißen *Nachfolger* von v . Für die direkten Nachfolger von v verwenden wir die Bezeichnung *Söhne*, dies sind die Ecken in $S_v^{v_0} := \{u \in V : v = \text{pre}_{v_0}(u)\}$. Die Blätter sind also die Ecken ohne Söhne.

Die bezüglich \preceq_{v_0} kleinste Ecke v in einem Teilbaum T' von T ist eindeutig bestimmt, wir schreiben $v = \min_{\prec_{v_0}} V(T')$ und sprechen von v als der Wurzel von T' . Die direkten Nachfolger eines Teilbaumes T' sind wie folgt definiert:

$$S^{v_0}(T') := \left(\bigcup_{v \in T'} S_v^{v_0} \right) - V(T').$$

Hat in einem textlichen Zusammenhang unserer Arbeit ein Baum eine feste Wurzel, so verzichten wir bei den auf sie bezugnehmenden Größen auf ihre Nennung, d.h. wir schreiben $v \prec u$, $\text{pre}(v)$, T_v , S_v und $S(T')$.

In einem Wurzelbaum besteht eine eineindeutige Beziehung zwischen den Kanten und den von der Wurzel verschiedenen Ecken. Dabei korrespondiert zu jeder solchen Ecke diejenige Kante, die sie mit ihrem Vorgänger verbindet. Ist also $v \in V - \{v_0\}$, so bezeichnen wir mit $e(v)$ die zu v gehörende Kante.³

Kontraktion, Quotienten. Eine in Algorithmen auf Graphen häufig benötigte Operation ist die *Kontraktion* von Kanten. Für einen Wurzelbaum T bedeutet die Kontraktion von $e(v)$, daß

- v und $e(v)$ aus T entfernt werden,⁴

³Man beachte, daß wir keinen Bezug auf eine mögliche Kantenrichtung nehmen. Insbesondere fordern wir nicht, daß Kantenrichtung und die Ordnung \preceq in einem Wurzelbaum in Beziehung zueinander stehen.

⁴Gewöhnlich wird die Kontraktion einer Kante eines Graphen so definiert, daß diese Kante und ihre inzidenten Ecken entfernt und durch eine *neue* Ecke ersetzt werden. In unserer Definition übernimmt $\text{pre}(v)$ die Rolle der neuen Ecke. Der algorithmische Vorteil ist, daß keine neuen „Eckennamen“ berücksichtigt werden müssen.

- die Kanten zwischen v und seinen Söhnen nun mit $\text{pre}(v)$ inzidieren. (Folglich sind Ecken aus S_v nun Element von $S_{\text{pre}(v)}$.)

Der aus der Kontraktion von $e = e(v)$ hervorgehende Wurzelbaum heißt *Quotient* von T nach e , wir bezeichnen ihn mit T/e oder auch (da e durch v eindeutig bestimmt ist) mit T/v . Häufig werden mehrere Kanten kontrahiert. Ist also $V' \subseteq V - \{v_0\}$, so ist T/V' der Wurzelbaum, der aus T durch Kontraktion aller Kanten aus $\{e(v) : v \in V'\}$ hervorgeht. (Die Reihenfolge hat dabei keinen Einfluß auf das Ergebnis.) Insbesondere ist auf diese Weise das Resultat $T/P[u, v]$ der Kontraktion eines Pfades $P[u, v]$ mit $u \succeq v$ definiert. Die triviale Quotientenbildung T/\emptyset hat das Ergebnis T .

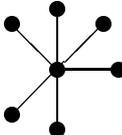
4.1.1.2 Einige Klassen von Bäumen

Für Bäume mit einem speziellen Aufbau werden besondere Bezeichnungen verwendet. In unserer Arbeit sind die folgenden Baum-Klassen von Interesse.

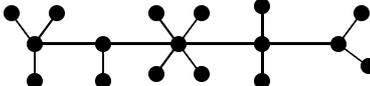
Ketten: Alle Ecken haben einen Grad von höchstens zwei.

Eine Kette: 

Sterne: (Höchstens) eine Ecke hat einen Grad größer als eins.

Ein Stern: 

Sternketten, „*Raupen*“: Der Baum enthält eine Kette und jede Kante inzidiert mit (mindestens) einer Ecke dieser Kette.

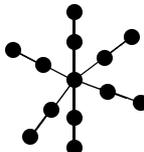
Eine Raupe: 

Spinnen: (Höchstens) eine Ecke hat einen Grad größer als zwei.

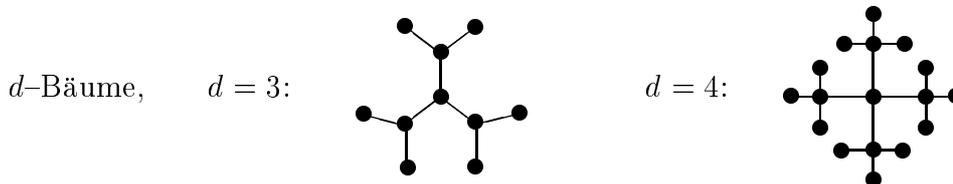
l -*Spinnen*, $l \geq 1$: Eine Spinne mit „Beinen“ der Länge l . Genauer:

$$\text{grad}(v) > 2 \text{ und } \text{grad}(u) = 1 \implies |P(v, u)| = l.$$

Die 1-Spinnen sind also gerade die Sterne.

Eine 2-Spinne: 

d-Bäume: Alle Ecken haben den Grad *d* oder den Grad eins. Zusätzlich gibt es eine „zentrale“ Ecke *v*, so daß $|P[v, u]| = \text{const}$ für alle Ecken *u* mit Grad eins.



4.1.2 Problemstellung der Baumzerlegung

Es sei ein Baum $T = (V, E)$ und eine Menge \mathcal{T} von Teilbäumen von T gegeben. Die Elemente von \mathcal{T} bezeichnen wir als *zulässige* Teilbäume. Eine (*Baum-*)*Zerlegung* von T ist eine Menge $\pi = \{T'_1, \dots, T'_s\} \subseteq \mathcal{T}$ zulässiger Teilbäume, so daß $V(T'_1), \dots, V(T'_s)$ eine Partition der Eckenmenge V bilden:

$$V = V(T'_1) \cup \dots \cup V(T'_s) \quad V(T'_i) \cap V(T'_j) = \emptyset \text{ für } 1 \leq i < j \leq s.$$

Wir bezeichnen mit $\mathcal{P}(T, \mathcal{T})$ die Menge aller solcher Zerlegungen.

Das Optimierungsziel bei der Baumzerlegung ist es, eine „günstigste“ Zerlegung zu bestimmen, in folgendem Sinne: Jeder Teilbaum $T' \in \mathcal{T}$ sei mit „Kosten“ cT' bewertet. Die Kosten der Zerlegung $\pi = \{T'_1, \dots, T'_s\}$ ergeben sich durch Addition,

$$c\pi := \sum_{i=1}^s cT'_i.$$

Die Aufgabe lautet nun: Bestimme $\pi \in \mathcal{P}(T, \mathcal{T})$ mit $c\pi$ kleinstmöglich.

Bei den Kosten cT' wird man zunächst an reelle Werte denken. Im Hinblick auf die Anwendungen im Teil III müssen wir hier jedoch verallgemeinern. Wir setzen lediglich voraus, daß die Kosten einem geordneten Monoid entstammen. Dies ist wie folgt definiert:

Definition 4.1

Das Tripel $(M, +, \leq)$ ist ein *geordnetes Monoid*, falls gilt:

- i. M ist nichtleere Menge,
- ii. '+' ist assoziative und kommutative Verknüpfung auf M ,
- iii. ' \leq ' ist eine lineare Ordnung⁵ auf M , und

⁵Andere Bezeichnungen: totale Ordnung oder vollständige Ordnung. Sie ist charakterisiert durch die Axiome (gültig für alle $m_1, m_2, m_3 \in M$): *Reflexivität* ($m_1 \leq m_1$), *Antisymmetrie* ($m_1 \leq m_2 \wedge m_2 \leq m_1 \Rightarrow m_1 = m_2$), *Transitivität* ($m_1 \leq m_2 \wedge m_2 \leq m_3 \Rightarrow m_1 \leq m_3$) und *Vergleichbarkeit* ($m_1 \leq m_2 \vee m_2 \leq m_1$).

iv. für alle $m_1, m_2, m_3 \in M$ mit $m_1 \leq m_2$ gilt $m_1 + m_3 \leq m_2 + m_3$.

Damit können wir die im Baumzerlegungsproblem nötigen Operationen durchführen: Berechnung der Kosten einer Partition als Summe der Kosten ihrer Teilbäume und Vergleich von zwei Partitionen hinsichtlich ihrer Kosten.

Bemerkung 4.2 (Komplexität der Monoid-Operationen)

Aus algorithmischer Sicht ist wichtig, welcher Rechenaufwand mit den Operationen Addition zweier Elemente und Vergleich zweier Elemente bezüglich der Ordnung in einem Monoid M verbunden ist. Die uns interessierenden Monoide (siehe 7.2.5) haben die Eigenschaft, daß diese Operationen mit konstantem Aufwand $O(1)$ durchführbar sind.

Darüberhinaus ist der Aufwand für die Berechnung der Funktion $c : \mathcal{T} \rightarrow M$ wichtig. Für die von uns betrachteten Monoide ist er typischerweise linear in der Eckenzahl des Urbild-Teilbaumes. Mit anderen Worten: Die Komplexität der Berechnung von cT' , $T' \in \mathcal{T}$, ist $O(|V(T')|)$.

Die folgende Definition faßt die Ausführungen zusammen.

Definition 4.3 (Allgemeines Baumzerlegungsproblem (BZP))

Gegeben seien

- ein Baum $T = (V, E)$,
- eine Menge \mathcal{T} von Teilbäumen von T ,
- ein geordnetes Monoid $(M, +, \leq)$,
- eine Bewertungsfunktion $c : \mathcal{T} \rightarrow M$.

Problem: Entscheide, ob $\mathcal{P}(T, \mathcal{T}) \neq \emptyset$ ist, und bestimme in diesem Fall eine Zerlegung $\pi = \{T'_1, \dots, T'_s\} \in \mathcal{P}(T, \mathcal{T})$, die $c\pi = \sum_{i=1}^s cT'_i$ minimiert.

Häufig tritt das allgemeine Baumzerlegungsproblem mit einer Beschränkung der Anzahl von Teilbäumen in der gesuchten Partition auf. Es wird gefordert, daß $|\pi| = p$ ist, mit einem $p \in \{1, \dots, |V|\}$. Wir bezeichnen die Menge aller solcher Zerlegungen mit $\mathcal{P}_p(T, \mathcal{T})$.

Definition 4.4 (Allgemeines p -Baumzerlegungsproblem (p -BZP))

Gegeben sei eine Instanz des allgemeinen Baumzerlegungsproblems und zusätzlich ein $p \in \{1, \dots, |V|\}$.

Problem: Entscheide, ob $\mathcal{P}_p(T, \mathcal{T}) \neq \emptyset$ ist, und bestimme in diesem Fall eine Zerlegung $\pi = \{T'_1, \dots, T'_p\} \in \mathcal{P}_p(T, \mathcal{T})$, die $c\pi = \sum_{i=1}^p cT'_i$ minimiert.

Die Definitionen 4.3 und 4.4 sind in dieser Form bisher wohl nicht in der Literatur zu finden. Jedoch wurden ähnliche Probleme untersucht; zum Beispiel das Packen

von Teilbäumen (Aghezzaf et al. [1], Barany et al. [9]; siehe auch Magnanti und Wolsey [65, Kap. 7]), oder das allgemeine Baumzerlegungsproblem mit reellwertiger Kostenfunktion c (Shaw [99]; vgl. 4.1.4).

4.1.3 Lösung mit dynamischer Programmierung

4.1.3.1 Lösung des allgemeinen BZP

Zur Lösung des allgemeinen Baumzerlegungsproblems bietet sich die Methode der dynamischen Programmierung an⁶. Sie ist ein gerne gewählter Ansatz für Optimierungsprobleme, denen eine Baumstruktur zugrundeliegt.⁷ Ihre Idee ist, die Optimallösung für das betrachtete Problem aus den Lösungen von Teilproblemen zusammenzusetzen. Voraussetzung ist demnach, daß das sogenannte *Optimalitätsprinzip* gilt: Es gibt Optimallösungen des Gesamtproblems, die aus Optimallösungen von Teilproblemen aufgebaut sind. Daß dies für unser allgemeines Baumzerlegungsproblem der Fall ist, werden wir im folgenden sehen.

Die Teilprobleme des allgemeinen BZP bestehen in der Aufgabe, Teilbäume von T optimal zu zerlegen. Hierbei müssen nicht alle Teilbäume betrachtet werden. Andererseits ist es nötig, die zu untersuchenden Teilbäume in eine Bearbeitungsreihenfolge zu bringen. Beides wird erreicht, indem durch Auszeichnung einer (beliebigen) Ecke $v_0 \in V$ als Wurzel der Baum T zu einem Wurzelbaum gemacht wird. Die dadurch definierten Teilbäume $T_v^{v_0}$, $v \in V$, werden nach abnehmender Ordnung \prec_{v_0} der Ecken v bearbeitet, mit anderen Worten, von den Blättern zur Wurzel.

Zur genauen Darstellung der Vorgehensweise führen wir einige Bezeichnungen ein. Wir gehen dabei von einer festgelegten Wurzel v_0 in T aus und beziehen uns nicht mehr explizit auf sie. Es sei

$$\mathcal{T}_v := \{T' \in \mathcal{T} : v \in T' \text{ und } v \preceq u \text{ für alle } u \in T'\} \quad (4.1)$$

die Menge der zulässigen Teilbäume, die in v wurzeln. Die \mathcal{T}_v , $v \in V$, bilden also eine Partition von \mathcal{T} . Ferner sei

$$\overline{\mathcal{T}}_v := \bigcup_{u \in T_v} \mathcal{T}_u.$$

Dies ist die Menge der zulässigen Teilbäume von T_v . Offenbar gilt $\overline{\mathcal{T}}_{v_0} = \mathcal{T}$.

Das Teilproblem für den Teilbaum T_v lautet nun:

Entscheide, ob $\mathcal{P}(T_v, \overline{\mathcal{T}}_v)$ nicht leer ist und bestimme in diesem Fall eine günstigste Zerlegung von T_v .

⁶Eine ausführliche Darstellung der dynamischen Programmierung gibt Ibaraki [55, Kap. 6,7]

⁷Aghezzaf und Wolsey [2] und Barany et al. [9] lösen dem allgemeinen BZP ähnliche Problemstellungen mit dynamischer Programmierung.

Um die erste Möglichkeit, daß es keine Zerlegung von T_v in zulässige Teilbäume gibt, in einer kompakten Formulierung der dynamischen Programmierung berücksichtigen zu können, erweitern wir das Monoid M um ein Symbol ∞ mit den Eigenschaften

- i. $m \neq \infty$ für alle $m \in M$,
- ii. $m < \infty$ für alle $m \in M$, und
- iii. $m + \infty = \infty$ für alle $m \in M \cup \{\infty\}$.

Die optimalen Kosten zu einem Teilproblem sind damit gegeben durch

$$C(v) := \begin{cases} \infty & \text{falls } \mathcal{P}(T_v, \overline{\mathcal{T}}_v) = \emptyset, \\ \min\{c\pi : \pi \in \mathcal{P}(T_v, \overline{\mathcal{T}}_v)\} & \text{sonst.} \end{cases} \quad (4.2)$$

Das allgemeine Baumzerlegungsproblem (für T) besteht in der Berechnung von $C(v_0)$ und, falls $C(v_0) < \infty$, in der Bestimmung eines entsprechenden $\pi \in \mathcal{P}(T, \mathcal{T})$.

Dazu werden die $C(v)$ in der folgenden Weise im Baum aufsteigend berechnet:

Proposition 4.5 (Rekursion für das allgemeine BZP)

Es sei $v \in V$ und für alle $u \in T_v$, $u \neq v$ sei $C(u)$ bereits berechnet. Dann gilt die Rekursionsgleichung

$$C(v) = \min\{cT^v + \sum_{u \in S(T^v)} C(u) : T^v \in \mathcal{T}_v\}. \quad (4.3)$$

Beweis. Die Aussage folgt aus den Eigenschaften des Monoids M , insbesondere iv., in Definition 4.1. □

Proposition 4.5 bringt das Optimalitätsprinzip zum Ausdruck: Eine optimale Zerlegung von T ist aus optimalen Zerlegungen von Teilbäumen von T zusammengesetzt. Es ergibt sich unmittelbar das folgende Verfahren zur Berechnung einer optimalen Zerlegung π^* .

Algorithmus 4.1 Lösung des allgemeinen Baumzerlegungsproblems

Input: allgemeines BZP gemäß Definition 4.3

Output: optimale Zerlegung π^* oder „keine zulässige Zerlegung“

- 1: **procedure** Rechne vorwärts **begin**
- 2: **for all** $v \in V$ von den Blättern zur Wurzel **do**
- 3: Berechne $C(v)$ gemäß (4.3)
- 4: **if** $C(v) < \infty$ **then** $T^*(v)$ sei minimierend in (4.3) **end if**
- 5: **end for**
- 6: **end**

```

7: procedure Rechne rückwärts begin
8:   if  $C(v_0) < \infty$  then
9:      $\pi' := \{T^*(v_0)\}; \pi^* := \emptyset$ 
10:    while  $\pi' \neq \emptyset$  do
11:      Wähle  $T' \in \pi'$ 
12:       $\pi^* := \pi^* \cup \{T'\}; \pi' := \pi' - \{T'\} \cup \{T^*(v) : v \in S(T')\}$ 
13:    end while
14:    gib_aus  $\pi^*$ 
15:  else
16:    gib_aus „keine zulässige Zerlegung“
17:  end if
18: end

```

Bemerkung 4.6 (Komplexität des allgemeinen Baumzerlegungsproblems)

Der Aufwand zur Lösung des allgemeinen BZP besteht darin, $|V|$ -mal das Optimierungsproblem (4.3) zu lösen. Dessen Eigenschaften sind somit ausschlaggebend für die Komplexität eines Baumzerlegungsproblems. Ist zum Beispiel $|\mathcal{T}|$ polynomial in der Größe der Problembeschreibung, so ist das allgemeine BZP polynomial lösbar. Dies ist insbesondere der Fall, wenn \mathcal{T} explizit in der Problembeschreibung gegeben ist.

Ein anderer interessanter Fall polynomialer Lösbarkeit sind Ketten, wie die nächste Proposition zeigt. Dagegen ist das allgemeine BZP für Sterne \mathcal{NP} -schwer. (Proposition 4.8)

Proposition 4.7 (Polynomialität des allgemeinen BZP für Ketten)

Es gelte Bemerkung 4.2 über die Komplexität der Operationen bezüglich des Monoids M . Ferner sei für jeden Teilbaum T' von T mit Aufwand $O(|V(T')|)$ die Zugehörigkeit zu \mathcal{T} entscheidbar. Dann ist der Aufwand zur Lösung des allgemeinen BZP für Ketten $O(|V|^3)$.

Beweis. Wir können annehmen, daß eine Ecke vom Grad eins als Wurzel ausgezeichnet ist. Dann ist $|\mathcal{T}_v| \leq |V(T_v)| \leq |V|$ für alle $v \in V$, daher sind in (4.3) $O(|V|)$ Teilbäume T' zu betrachten. Für jeden dieser Teilbäume kann die Summe $cT' + \sum_{u \in S(T')} C(u)$ mit Aufwand $O(|V(T')|)$ berechnet werden. \square

Proposition 4.8 (Allgemeines BZP für Sterne)

Das allgemeine BZP ist für die Baumklasse der Sterne \mathcal{NP} -schwer.

Beweis. Wir geben eine Transformation des (\mathcal{NP} -schweren) Rucksackproblems⁸ auf ein BZP mit sternförmigem Baum T .

Durch $c_i > 0$, $w_i > 0$, $i = 1, \dots, n$ und $b \geq \max_i w_i$ sei eine Instanz des Rucksackproblems

$$\max \left\{ \sum_{i \in I} c_i : \sum_{i \in I} w_i \leq b, I \subseteq \{1, \dots, n\} \right\}$$

gegeben. Sei T ein Stern mit $n + 1$ Ecken $0, 1, \dots, n$, wobei 0 die zentrale Ecke sei. Setze $w_0 := 0$ und

$$\mathcal{T} = \{T' \triangleleft T : \sum_{i \in T'} w_i \leq b\}.$$

Für $T' \in \mathcal{T}$ gibt es die beiden Möglichkeiten $T' = (\{i\}, \emptyset)$ für ein $i \in \{1, \dots, n\}$ oder $0 \in T'$. Das Monoid M seien die reellen Zahlen \mathbb{R} mit der gewöhnlichen Addition und Ordnung. Für $T' \in \mathcal{T}$ setze

$$cT' := \begin{cases} 0 & T' = (\{i\}, \emptyset) \text{ mit } i \in \{1, \dots, n\}, \\ -\sum_{i \in T'} c_i & 0 \in T'. \end{cases}$$

Sei T_0^* der Teilbaum in einer optimalen Zerlegung von T , der die Ecke 0 enthält. Es ist einfach zu sehen, daß die Indizes der zu 0 adjazenten Ecken in T_0^* eine optimale Lösung des Rucksackproblems bilden. \square

4.1.3.2 Lösung des allgemeinen p -BZP

Wir wenden uns nun der Lösung des p -Baumzerlegungsproblems zu. Hierbei ist die Vorgehensweise analog zu den bisherigen Ausführungen. Jedoch vergrößert die zusätzliche Bedingung, daß nur Zerlegungen von T in p Teilbäume zulässig sind, den Umfang der zu betrachtenden Teilprobleme. Sei

$$\bar{q}_v := \min\{p, |V(T_v)|\}$$

für $v \in V$; für $1 \leq q \leq \bar{q}_v$ definieren wir

$$C(v, q) := \begin{cases} \infty & \text{falls } \mathcal{P}_q(T_v, \overline{T}_v) = \emptyset, \\ \min\{c\pi : \pi \in \mathcal{P}_q(T_v, \overline{T}_v)\} & \text{sonst.} \end{cases}$$

Dies ist der optimale Lösungswert für das Zerlegen von T_v in q Teilbäume.

⁸Rucksackprobleme gehören zu den bekanntesten Problemen der Kombinatorischen Optimierung. Eine umfassende Darstellung bringt die Monographie von Martello und Toth [70]. Die neuesten Entwicklungen lassen sich zum Beispiel aus Martello et al. [69] entnehmen.

Proposition 4.9 (Rekursion für das allgemeine p -BZP)

Sei $v \in V$ und für $u \in T_v$, $u \neq v$ und $q' \in \{1, \dots, \bar{q}_u\}$ sei $C(u, q')$ bereits berechnet.

Dann ist

$$C(v, q) = \min \left\{ cT' + \sum_{u \in S(T')} C(u, q'_u) : T' \in \mathcal{T}_v, \sum_{u \in S(T')} q'_u = q - 1, 1 \leq q'_u \leq \bar{q}_u \right\}. \quad (4.4)$$

Bei der Lösung des Optimierungsproblems (4.4) sind nun nicht mehr nur die $T' \in \mathcal{T}_v$ zu betrachten, sondern es ist zusätzlich eine optimale „Aufteilung“ der Teilbaum-Anzahl $q - 1$ auf die unterhalb von T' wurzelnden Teilbäume zu finden.

Algorithmus 4.2 Lösung des allgemeinen p -Baumzerlegungsproblems

Input: allgemeines p -BZP gemäß Definition 4.4

Output: optimale Zerlegung π^* oder „keine zulässige Zerlegung“

```

1: procedure Rechne vorwärts begin
2:   for all  $v \in V$  von den Blättern zur Wurzel do
3:     for all  $q \in \{1, \dots, \bar{q}_v\}$  do
4:       Berechne  $C(v, q)$  gemäß (4.4)
5:       if  $C(v, q) < \infty$  then
6:          $T^*(v, q)$  und  $\{q_u^*(v, q) : u \in S(T^*(v, q))\}$  seien minimierend in (4.4)
7:       end if
8:     end for
9:   end for
10: end

11: procedure Rechne rückwärts begin
12:   if  $C(v_0, p) < \infty$  then
13:      $\pi' := \{(T^*(v_0, p), p)\}$ ;  $\pi^* := \emptyset$ 
14:     while  $\pi' \neq \emptyset$  do
15:       Wähle  $(T', q') \in \pi'$ 
16:        $\pi^* := \pi^* \cup \{T'\}$ 
17:        $v' := \min_{\prec} V(T')$ ;  $\pi' := \pi' - \{(T', q')\} \cup \{T^*(u, q_u^*(v', q')) : u \in S(T')\}$ 
18:     end while
19:     gib_aus  $\pi^*$ 
20:   else
21:     gib_aus „keine zulässige Zerlegung“
22:   end if
23: end

```

Beim Rückwärtsrechnen werden in der Hilfsmenge π' gegenüber Algorithmus 4.1 nun Paare (T', q') festgehalten, wobei T' ein Teilbaum in der optimalen Partition π^*

ist und q' die Anzahl Teilbäume, in die $T_{v'}$ zerlegt werden muß (v' die Wurzel von T').

Bemerkung 4.10 (Komplexität des allgemeinen p -BZP)

Auch für das allgemeine p -Baumzerlegungsproblem gilt: Ist $|\mathcal{T}|$ polynomial in der Problembeschreibung, so ist das p -BZP polynomial. Dies erfordert allerdings, daß zu jedem T' im Optimierungsproblem (4.4) die optimalen q'_u , $u \in S(T')$, über eine weitere dynamische Programmierung berechnet werden. Wir skizzieren dies kurz:

Optimale q'_u bei festem T' in (4.4): Sei $T' \in \mathcal{T}_v$ fest in (4.4). Sei $S(T') = \{u_1, \dots, u_k\}$. Zur Vereinfachung schreiben wir \bar{q}_i für \bar{q}_{u_i} . Wir haben das Optimierungsproblem

$$\min \sum_{i=1}^k C(u_i, q_i) \quad \text{unter} \quad \sum_{i=1}^k q_i = q - 1, \quad 1 \leq q_i \leq \bar{q}_i$$

zu lösen. Dazu definieren wir

$$\tilde{C}(l, q') := \min \left\{ \sum_{i=1}^l C(u_i, q_i) : \sum_{i=1}^l q_i = q', \quad 1 \leq q_i \leq \bar{q}_i \right\}$$

für $1 \leq l \leq k$ und $l \leq q' \leq q - 1 - (k - l)$. Gesucht ist somit $\tilde{C}(k, q - 1)$. Es läßt sich mittels der Rekursionsformel

$$\tilde{C}(l, q') = \min \{ C(u_l, q'') + \tilde{C}(l - 1, q' - q'') : q'' \text{ durchläuft alle relevanten Werte} \}$$

berechnen. Sie ist verankert in

$$\tilde{C}(1, q') = C(u_1, q'), \quad 1 \leq q' \leq q - 1 - (k - 1).$$

Die optimalen q_i werden durch Rückwärtsrechnung bestimmt.

Der Aufwand für die Berechnung von $\tilde{C}(k, q - 1)$ ist $O(qk) = O(p|V|)$. Somit läßt sich (4.4) polynomial lösen, wenn $|\mathcal{T}_v|$ polynomial ist. Im Fall von Ketten ist $|S(T')| \leq 1$,⁹ deshalb ist (4.4) besonders einfach lösbar:

Proposition 4.11 (Polynomialität des allgemeinen p -BZP für Ketten)

Unter den Voraussetzungen wie in Proposition 4.7 ist der Aufwand zur Lösung allgemeinen p -BZP für Ketten $O(p|V|^3)$.

Beweis. Der Aufwand zur Lösung von (4.4) ist $O(|V(T_v)|^2)$, für jeden Teilbaum $T' \in \mathcal{T}_v$ muß cT' berechnet werden und $|S(T')| \leq 1$. Ferner muß (4.4) $O(p|V|)$ -mal gelöst werden. \square

⁹Wir können o.B.d.A. annehmen, daß eine Endecke als Wurzel gewählt ist.

Bemerkung 4.12 (Allgemeines p -BZP für Sterne)

Ist ein spezielles BZP \mathcal{NP} -schwer, so gilt dies auch für das zugehörige p -BZP, in welchem die zulässigen Zerlegungen auf solche mit p Teilbäumen beschränkt sind. Somit folgt aus Proposition 4.8 unmittelbar, daß das allgemeine p -BZP für Sterne \mathcal{NP} -schwer ist.

4.1.4 Lösung mit Spaltenerzeugung

Die Entwicklung von Algorithmen für kombinatorische Optimierungsprobleme, die auf der Formulierung eines solchen Problems als (gemischt-ganzzahliges) lineares Programm beruhen, hat sich als außerordentlich erfolgreicher Ansatz erwiesen. Daher ist es naheliegend, auch für das allgemeine Baumzerlegungsproblem diese Vorgehensweise zu untersuchen. Die im folgenden dargestellten Überlegungen beziehen sich zum großen Teil auf die Arbeit von Shaw [99], und ähnliche Darstellungen in Shaw [100] und Cho und Shaw [23]. Implizit ist dieses Material bereits in den umfassenderen Arbeiten von Barany et al. [9] und Aghezzaf et al. [1] enthalten (siehe auch Magnanti und Wolsey [65, Kap. 7]).

4.1.4.1 BZP als lineares Programm und Spaltenerzeugung

Zunächst ist die Voraussetzung zu machen, daß das Monoid M in Definition 4.3 die Menge der reellen Zahlen \mathbb{R} ist, ausgestattet mit der üblichen Addition und Ordnung. Für jeden Teilbaum $T' \in \mathcal{T}$ sei $a(T') \in \{0, 1\}^n$, $n = |V|$, der Inzidenzvektor von $V(T')$ in V , also

$$a(T')_v = \begin{cases} 1 & v \in V(T'), \\ 0 & \text{sonst,} \end{cases} \quad \text{für } v \in V.$$

Mit Hilfe der 0–1–Matrix $A = (a(T') : T' \in \mathcal{T})$, deren Spalten durch die $a(T')$ gebildet werden, läßt sich das allgemeine BZP in der Form eines set–partitioning Problems¹⁰ schreiben:

$$\min \sum_{T'} c_{T'} x_{T'} \quad \text{unter} \quad Ax = 1, \quad x \in \{0, 1\}^{|\mathcal{T}|}. \quad (4.5)$$

Die zulässigen Zerlegungen von T sind eindeutig durch die zulässigen Lösungen x dieses Modells charakterisiert.

Die Matrix A ist eine sogenannte *Knoten–Teilbaum–Inzidenzmatrix*, also eine Matrix, deren Spalten Ecken–Inzidenzvektoren von Teilbäumen eines (festen) Baumes sind. Set–partitioning (und set–packing) Probleme mit einer solchen Matrix haben

¹⁰Siehe zum Beispiel Nemhauser und Wolsey [79, S. 6] für eine Einführung grundlegender kombinatorischer Optimierungsprobleme

die hier wesentliche Eigenschaft, daß sie durch ihre lineare Relaxation vollständig beschrieben sind.

Satz 4.13 ([65, Th. 7.1])

Ist A eine Knoten–Teilbaum–Inzidenzmatrix, so sind alle Extrempunkte des Polyeders $P_A = \{x \in \mathbb{R}^t : Ax \leq 1, x \geq 0\}$ ganzzahlig. ($t = |\mathcal{T}|$ ist die Anzahl der Spalten von A .)

Die Aussage überträgt sich unmittelbar auf die Seite $\{x \in \mathbb{R}^t : Ax = 1, x \geq 0\}$ von P_A und wir erhalten

Korollar 4.14

Das lineare Programm

$$\min \sum_{\mathcal{T}} cT^l x_{T^l} \quad \text{unter} \quad Ax = 1, x \geq 0 \quad (4.6)$$

löst das allgemeine BZP.

Es stellt sich nun die Frage, ob diese theoretisch interessante und ansprechende Aussage algorithmisch ausgenutzt werden kann. Zunächst ist klar, daß eine naive Lösung von (4.6) zum Beispiel mit dem Simplex–Verfahren nicht in Frage kommt, denn dieses lineare Programm hat im allgemeinen sehr viele Spalten. Typischerweise ist für ein BZP die Menge \mathcal{T} der zulässigen Teilbäume nicht explizit gegeben, sondern durch gewisse Bedingungen charakterisiert. Dann ist ein exponentielles Anwachsen von $|\mathcal{T}|$ in der Größe der Problembeschreibung der Normalfall.

Es ist eine übliche Vorgehensweise, lineare Programme mit sehr vielen, implizit beschriebenen Spalten, über ein Spaltenerzeugungsverfahren zu lösen. Der Gedanke dabei ist, daß nur wenige der Spalten zu einer optimalen Basis gehören und daß es dann genügt, ein eingeschränktes lineares Programm mit diesen Spalten zu lösen. Wir stellen die Vorgehensweise am Beispiel des linearen Programms (4.6) dar.

Das Verfahren der Spaltenerzeugung. Sei A' eine Matrix, deren Spalten eine (i. allg. kleine) Teilmenge der Spalten von A sind und \mathcal{T}' die diesen Spalten entsprechenden Teilbäume. Es wird das *Restricted Master Problem* (RMP)

$$\min \sum_{\mathcal{T}'} cT^l x_{T^l} \quad \text{unter} \quad A'x = 1, x \geq 0 \quad (4.7)$$

zum *Master Problem* (MP) (4.6) gelöst. Seien $\lambda_v, v \in V$, optimale Dualvariablen zu (4.7). Genau dann ist die optimale Lösung von RMP optimal für MP, wenn

$$cT^l - \sum_{v \in V} a(T^l)_v \lambda_v \geq 0 \quad \text{für alle } T^l \in \mathcal{T} - \mathcal{T}'$$

gilt. Diese Bedingung bedeutet, daß die reduzierten Kosten aller in RMP unberücksichtigten Spalten nichtnegativ sind. Um sie zu verifizieren, wird das *Subproblem* (SP)

$$\gamma^* := \min\{cT' - \sum_{v \in T'} \lambda_v : T' \in \mathcal{T}\} \quad (4.8)$$

gelöst. Ist $\gamma^* \geq 0$, so ist MP gelöst, andernfalls wird der Inzidenzvektor des in SP optimalen Teilbaumes den Spalten von A' hinzugefügt und dann RMP reoptimiert.

Die Vorgehensweise, das BZP mit Hilfe von (4.6) und Spaltenerzeugung zu lösen, erscheint attraktiv. Denn das Problem, eine optimale *Baumzerlegung* zu finden, ist damit im wesentlichen zurückgeführt auf mehrfaches Lösen von (4.8), die Bestimmung *eines* optimalen Teilbaumes; hinzu kommen Iterationen des Simplex-Verfahrens, die unproblematisch sind.

Konsequent umgesetzt wird diese Idee von D. X. Shaw. Er verfeinert das dargestellte Spaltenerzeugungsverfahren, indem er die spezielle Struktur der Matrix A für das BZP ausnutzt, und entwickelt ein sog. *limited column generation* Verfahren. Dieses Verfahren wird in mehreren Publikationen dargestellt (Cho und Shaw [23], Shaw, [99], Shaw, [100]) und dabei algorithmisch für die Lösung spezieller BZPe zum Einsatz gebracht. Deshalb soll hier näher darauf eingegangen werden.

Es sei jedoch vorweggenommen, was die Erkenntnis daraus sein wird: Limited column generation nach Shaw ist nichts anderes als die Dynamische Programmierung gemäß Algorithmus 4.1. Diese Tatsache, die aus einem einfachen Zusammenhang zwischen den $C(v)$ in Algorithmus 4.1 und den Dualvariablen λ_v in RMP folgt, geht zum Beispiel aus Magnanti und Wolsey [65, Seite 578, Bemerkung (ii)] hervor.

Warum Shaw seinen Lesern den naheliegenden Algorithmus 4.1 vorenthält und sie den „Umweg“ über (4.6) und das limited column generation Verfahren führt, bleibt unklar. Die Äquivalenz der Verfahren an sich ist ein interessanter Punkt, der Erwähnung verdient. Shaw vermittelt jedoch den Eindruck, mit limited column generation sei algorithmisch eine Neuentwicklung erfolgt.¹¹ Dies ist nicht der Fall.

4.1.4.2 Limited Column Generation nach Shaw

Wir folgen der Darstellung in Shaw [100]. Zunächst wird die Matrix A' als Einheitsmatrix initialisiert. Es wird somit die Voraussetzung gemacht, daß für alle $v \in V$ die Teilbäume ($\{v\}, \emptyset$) zulässig sind. (Dies kann ggf. durch Festlegung von sehr hohen Kosten für diese Teilbäume erreicht werden.) Die optimalen Dualvariablen in RMP

¹¹Zum Beispiel in Cho und Shaw [23, S. 318]: *Our computational experiments indicate that the Limited Column Generation is a very efficient approach for these models.* Man beachte auch den Titel der Arbeit Shaw [100].

sind dann durch $\lambda_v = c(\{v\}, \emptyset)$ gegeben. Ferner wird der Baum T durch Auszeichnung einer Wurzel geordnet.

Shaws Überlegung ist es, in den Iterationen des Spaltenerzeugungsverfahrens beim Lösen des Subproblems (4.8) nicht alle Spalten zu den Teilbäumen in \mathcal{T} zu betrachten, sondern nur zu solchen, die in einer festen Ecke v wurzeln. Das heißt, (4.8) wird ersetzt durch

$$\gamma_v^* := \min\{cT' - \sum_{v \in T'} \lambda_v : T' \in \mathcal{T}_v\}. \quad (4.9)$$

(\mathcal{T}_v ist in (4.1) definiert, man beachte $\mathcal{T}_v \neq \emptyset$.) Dies bezeichnet Shaw als die *locally most violated* (LMV) reduzierten Kosten. Sie werden für alle Ecken $v \in V$ von den Blättern zur Wurzel berechnet, wobei jeweils die Matrix A' aktualisiert und RMP neu gelöst wird. Dabei ändert sich nur der Wert der einen Dualvariable λ_v ,

$$\lambda_v := \lambda_v + \gamma_v^*. \quad (4.10)$$

Es ist also nicht nötig, zur Optimierung von RMP ein explizites Simplex-Verfahren heranzuziehen.

Algorithmus 4.3 Limited column generation, [100, S. 373]

- 1: $A' := I$
 - 2: $\lambda_v := c(\{v\}, \emptyset)$, $v \in V$
 - 3: **for all** $v \in V$ von den Blättern zur Wurzel in T **do**
 - 4: $\gamma_v^* := \min\{cT' - \sum_{v \in T'} \lambda_v : T' \in \mathcal{T}_v\}$ /* siehe (4.9) */
 - 5: **if** $\gamma_v^* < 0$ **then**
 - 6: ersetze die zu $(\{v\}, \emptyset)$ gehörende Spalte von A' durch den Inzidenzvektor eines optimalen Teilbaumes in der Berechnung von γ_v^* .
 - 7: $\lambda_v := \lambda_v + \gamma_v^*$
 - 8: **end if**
 - 9: **end for**
-

Satz 4.15 ([100, Th. 1])

Algorithmus 4.3 löst das lineare Programm (4.6).

Der Beweis ist recht einfach durch vollständige Induktion zu führen. Für uns von Interesse ist die bereits angekündigte Äquivalenz der Algorithmen 4.1 und 4.3, die sich aus dem folgenden Satz ergibt.

Satz 4.16

Nach der Iteration zu $v \in V$ in den Algorithmen 4.1 (Vorwärtsrechnung) und 4.3 gilt

$$\lambda_v = C(v) - \sum_{u \in S_v} C(u). \quad (4.11)$$

Die Berechnung der LMV reduzierten Kosten (4.9) ist äquivalent zur Rekursionsgleichung (4.3) der dynamischen Programmierung.

Beweis. Ist v ein Blatt, so ist $\mathcal{T}_v = \{(\{v\}, \emptyset)\}$ und damit $\gamma_v^* = 0$. Somit gilt $\lambda_v = c(\{v\}, \emptyset)$ am Ende der Iteration zu v in Algorithmus 4.3. Andererseits ist offensichtlich $C(v) = c(\{v\}, \emptyset)$.

Sei v kein Blatt und (4.11) gelte für alle $v' \succ v$. Algorithmus 4.3 stehe am Beginn der Iteration zu v . Dann ist für $T' \in \mathcal{T}_v$

$$\begin{aligned} \sum_{v' \in T'} \lambda_{v'} &= c(\{v\}, \emptyset) + \sum_{\substack{v' \in T' \\ v' \neq v}} \left(C(v') - \sum_{u \in S_{v'}} C(u) \right) \\ &= c(\{v\}, \emptyset) + \sum_{v' \in S_v \cap V(T')} C(v') - \sum_{u \in S(T') - S_v} C(u) \\ &= c(\{v\}, \emptyset) + \sum_{v' \in S_v} C(v') - \sum_{u \in S(T')} C(u), \end{aligned}$$

wobei wir die letzte Gleichung durch Addition und Subtraktion von $\sum_{v' \in S_v - V(T')} C(v')$ erhalten. Wir setzen nun zur Abkürzung $\zeta_v := c(\{v\}, \emptyset) + \sum_{v' \in S_v} C(v')$ und erhalten damit

$$\sum_{v' \in T'} \lambda_{v'} = \zeta_v - \sum_{u \in S(T')} C(u).$$

Somit geht (4.9) über in

$$\gamma_v^* = -\zeta_v + \min \left\{ cT' + \sum_{u \in S(T')} C(u) : T' \in \mathcal{T}_v \right\},$$

was bis auf die Konstante $-\zeta_v$ das Optimierungsproblem (4.3) ist. Dies zeigt die zweite Behauptung des Satzes. Es ist also $\gamma_v^* = -\zeta_v + C(v)$. Offenbar ist $\zeta_v \geq C(v)$ (man beachte $(\{v\}, \emptyset) \in \mathcal{T}_v$ in (4.3)) und damit $\gamma_v^* \leq 0$.

Ist γ_v^* negativ, so erhalten wir (4.11) durch Einsetzen von γ_v^* in die Aktualisierungsvorschrift (4.10), die in Zeile 7 von Algorithmus 4.3 ausgeführt wird.

$$\lambda_v := \lambda_v + \gamma_v^* = c(\{v\}, \emptyset) - \zeta_v + C(v) = C(v) - \sum_{v' \in S_v} C(v')$$

Andernfalls wird λ_v nicht verändert und ist

$$\lambda_v = c(\{v\}, \emptyset) + 0 = c(\{v\}, \emptyset) - \zeta_v + C(v) = C(v) - \sum_{v' \in S_v} C(v').$$

□

4.1.4.3 p -BZP als 0–1-Programm

Wir stellen die Frage, ob sich die Überlegung, BZP als lineares Programm mit Spaltenerzeugung zu lösen, auch auf das allgemeine p -BZP übertragen läßt. Wie in Abschnitt 4.1.4.1 machen wir die Voraussetzung, daß das Monoid M in Definition 4.4 durch \mathbb{R} gegeben ist. Mit der Matrix A erhalten wir die folgende Formulierung von p -BZP als *set-partitioning* Problem mit Zusatzbedingung:

$$\min \sum_{\mathcal{T}} cT'x_{T'} \quad \text{unter} \quad Ax = 1, 1x = p, x \in \{0, 1\}^{|\mathcal{T}|}. \quad (4.12)$$

Wegen der Zusatzbedingung gilt hier jedoch kein Korollar 4.14 entsprechender Satz. Vielmehr hat der Polyeder

$$P_{A,p} = \{x \in \mathbb{R}^t : Ax = 1, 1x = p, x \geq 0\}$$

im allgemeinen nichtganzzahlige Extrempunkte.

Beispiel 4.1

Sei T eine Kette mit drei Ecken. Die Matrix

$$A = \begin{pmatrix} 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 1 \end{pmatrix}$$

ist eine Knoten–Teilbaum–Inzidenzmatrix für T . Der Polyeder $\{Ax = 1, x \geq 0\}$ hat nur die ganzzahligen Extrempunkte (man beachte Satz 4.13)

$$(1, 0, 0, 0, 0), \quad (0, 1, 0, 0, 1) \quad \text{und} \quad (0, 0, 1, 1, 1).$$

Dagegen hat $P_{A,2}$ die Extrempunkte

$$(0, 1, 0, 0, 1) \quad \text{und} \quad \left(\frac{1}{2}, 0, \frac{1}{2}, \frac{1}{2}, \frac{1}{2}\right).$$

Aus dem Beispiel folgt, daß das lineare Programm

$$\min \sum_{\mathcal{T}} cT'x_{T'} \quad \text{unter} \quad Ax = 1, 1x = p, x \geq 0 \quad (4.13)$$

im allgemeinen nur eine untere Schranke für den optimalen Wert von (4.12) liefert. Für die Berechnung einer optimalen Lösung von (4.12) kann ein Verzweungsverfahren (Branch&Bound) eingesetzt werden. In der Regel wird es nötig sein, dabei die zu den Knoten des Suchbaumes gehörenden linearen Relaxationen der Form (4.13) nicht explizit, sondern implizit mit Spaltenerzeugung zu lösen. Diese Technik ist unter dem

Namen *Branch&Price* bekannt, siehe zum Beispiel Barnhart et al. [10]. Als Besonderheit sei erwähnt, daß ein Verzweigen über die beiden Äste $x_{T'} = 0$ und $x_{T'} = 1$ für eine Variable $x_{T'}$ mit gebrochenem Wert in der Optimallösung von (4.13) ungeeignet ist und andere Verzweigungsschemata herangezogen werden müssen.¹²

4.2 Modelle für gleichmäßige Baumzerlegung

Nachdem wir das Baumzerlegungsproblem in einer allgemeinen Form eingeführt und untersucht haben, wenden wir uns der gleichmäßigen Baumzerlegung zu. Dabei handelt es sich um spezielle Baumzerlegungsprobleme, die sich fast durchgängig als Probleme vom Typ BZP oder p -BZP herausstellen werden.

Die Idee der gleichmäßigen Baumzerlegung ist einfach. Der Baum T ist mit nicht-negativen Eckengewichten versehen; Teilbäume erhalten ein Gewicht, das als Summe der Gewichte ihrer Ecken definiert ist. T soll so zerlegt werden, daß alle Teilbäume möglichst gleichschwer sind.

Wir gehen von einem eckengewichteten Baum $T = (V, E, w)$ aus. Die Abbildung $w : V \rightarrow \mathbb{R}_{\geq 0}$ ordnet jeder Ecke $v \in V$ ein nichtnegatives Gewicht w_v zu. In additiver Weise läßt sich damit jedem Teilbaum $T' \triangleleft T$ das Gewicht

$$wT' = \sum_{v \in V(T')} w_v$$

zuschreiben. Es sollen Zerlegungen $\pi = \{T'_1, \dots, T'_s\}$ bestimmt werden, für welche die Gewichte wT'_1, \dots, wT'_s möglichst gleich groß sind.

Die Quantifizierung des Terms „möglichst gleich groß“ kann auf ganz unterschiedliche Weise vorgenommen werden. Die uns wesentlich erscheinenden Modellbildungen sollen im folgenden vorgestellt werden (siehe 4.2.2). Viele davon sind auch in der Literatur zu finden.

4.2.1 Ein Klassifikationsschema für Baumzerlegungsprobleme

Um zu einer übersichtlichen und einheitlichen Benennung der Modelle zu kommen, führen wir ein Klassifikationsschema ein. Es ist an die von Problemen der Ablaufpla-

¹²Die Bedingung $x_{T'} = 0$ im Spaltenerzeugungs-Subproblem (4.8) zu berücksichtigen, heißt, den Teilbaum T' unter den möglichen Lösungen von (4.8) auszuschließen. Wäre gerade er die Optimallösung, müßte die nächstbeste Lösung bestimmt werden. Dadurch wird die Struktur des Problems (4.8) zerstört. (Das Berechnen der k -besten Lösung eines Optimierungsproblems ist im allgemeinen deutlich schwieriger, als die Bestimmung der Optimallösung.)

nung¹³ bekannte Notation angelehnt. Das Schema besteht aus vier Positionen in der Form

Position 1 / Position 2 / Position 3 / Position 4

deren Bedeutung im folgenden ausgeführt wird.

Position 1 ist der Typ des zu zerlegenden Graphen. Im wesentlichen behandeln wir das Zerlegen von Bäumen, doch lassen sich die Modelle zumeist auch für das Zerlegen eines zusammenhängenden Graphen in zusammenhängende Subgraphen formulieren. Beispiele für Einträge sind

T Ein Baum ist in Teilbäume zu zerlegen.

G Ein zusammenhängender Graph ist in zusammenhängende volle Subgraphen¹⁴ zu zerlegen.

P Ein zusammenhängender *planarer* Graph ist zu zerlegen wie bei **G**.

Position 2 sind gegebenenfalls Restriktionen, die die Menge \mathcal{T} der zulässigen Teilbäume charakterisieren. Beispiele für Einträge sind

L Das Gewicht aller Teilbäume (bzw. Subgraphen) in der Zerlegung ist nach unten beschränkt.

U Wie **L**, Beschränkung nach oben.

Zent Eine Teilmenge $Z \subset V$ ist als *Zentren* ausgezeichnet. Jeder Teilbaum (bzw. Subgraph) muß genau ein Zentrum enthalten.¹⁵

2-Zent Wie **Zent**, jedoch bilden die Menge Z der Zentren und die restlichen Ecken $V - Z$ eine 2-Färbung des Baumes (bzw. Graphen); alle Kanten verlaufen zwischen Z und $V - Z$.

Ist in der Problemstellung keine Beschränkung an die Menge der zulässigen Teilbäume vorhanden, kennzeichnen wir dies durch einen „·“ in Position 2.

Position 3 sind gegebenenfalls Restriktionen, welche die zulässigen Zerlegungen beschreiben. Der Eintrag p bedeutet, daß nur Zerlegungen mit p (nichtleeren) Teilbäumen (bzw. Subgraphen) zulässig sind. Die Abwesenheit dieser Restriktion wird durch einen „·“ kenntlich gemacht.

¹³Für *Scheduling-Probleme* siehe zum Beispiel Blazewicz et al. [16] und Pinedo [89].

¹⁴Ein *voller* Subgraph ist durch eine Teilmenge der Ecken und alle zwischen diesen Ecken verlaufenden Kanten definiert.

¹⁵Diese und die folgende Restriktion ist durch die Anwendungen in der Gebietsaufteilung motiviert.

Position 4 ist die Zielsetzung des Modells, das heißt, die Angabe der Zielfunktion. Beispiele für Einträge sind

minimax Das Gewicht des schwersten Teilbaumes (bzw. Subgraphen) in der Zerlegung ist zu minimieren.

maximin Das Gewicht des leichtesten Teilbaumes (bzw. Subgraphen) ist zu maximieren.

Δ Die Differenz zwischen dem Gewicht des schwersten und dem des leichtesten Teilbaumes ist zu minimieren.

$\|\cdot\|_\rho$ Sei μ ein vorgegebener Wert, den die Gewichte der Teilbäume (bzw. Subgraphen) in der Zerlegung möglichst gut approximieren sollen. Dazu definieren wir zu einer Zerlegung $\pi = \{T'_1, \dots, T'_s\}$ den Vektor $\delta(\pi, \mu) := (wT'_1 - \mu, \dots, wT'_s - \mu)$. Seine Komponenten sind die Differenzen zwischen tatsächlichem und gewünschtem Gewicht der Teilbäume. Zielsetzung ist nun, eine Zerlegung zu finden, welche für gegebenes $1 \leq \rho \leq \infty$ die ρ -Norm von δ minimiert. Dabei ist die ρ -Norm eines Vektors $x = (x_1, \dots, x_s)$ wie üblich definiert durch

$$\|x\|_\rho = \|(x_1, \dots, x_s)\|_\rho = \begin{cases} (\sum_{i=1}^s |x_i|^\rho)^{1/\rho} & 1 \leq \rho < \infty, \\ \max_{i=1 \dots s} |x_i| & \rho = \infty. \end{cases}$$

Für $\rho = 1$ ergibt sich zum Beispiel die Minimierung der totalen (absoluten) Abweichung der Teilbaum-Gewichte von μ , für $\rho = \infty$ die Minimierung der maximalen Abweichung von μ .

$\max |\pi|$ Die Anzahl der Teilbäume in der Zerlegung ist zu maximieren.

$\min |\pi|$ Die Anzahl der Teilbäume ist zu minimieren.

\exists Es handelt sich um ein Entscheidungsproblem, das heißt, es ist gefragt, ob eine Zerlegung mit den durch Position 2 und 3 spezifizierten Eigenschaften existiert.

Bezug auf mehrere Problemstellungen. Nimmt eine Aussage auf mehrere Problemstellungen Bezug, kann dies natürlich dadurch geschehen, daß sie alle in der soeben eingeführten Notation aufgezählt werden. Doch kann dies umständlich sein, wenn es sich um viele Probleme handelt. Deshalb lassen wir die Möglichkeit zu, an den Positionen 1 bis 4 Alternativen anzugeben, die in Klammern [...] gesetzt sind. So ist zum Beispiel $[P, T] / \cdot / [\cdot, p] / \exists$ gleichbedeutend mit der Aufzählung der vier (trivialen) Entscheidungsprobleme $P / \cdot / \cdot / \exists$, $P / \cdot / p / \exists$, $T / \cdot / \cdot / \exists$ und $T / \cdot / p / \exists$.

4.2.2 Probleme der gleichmäßigen Baumzerlegung

Die folgende Darstellung von Problemen der gleichmäßigen Baumzerlegung dient einem dreifachen Ziel.

1. Eine kurze verbale Beschreibung soll die jeweilige Fragestellung verdeutlichen.
2. Es soll gezeigt werden, daß die Probleme bis auf wenige Ausnahmen allgemeine Baumzerlegungsprobleme im Sinne von Abschnitt 4.1 sind.
3. Wir geben Hinweise, ob und wo das jeweilige Problem in der Literatur auftaucht.

Eine Übersicht der Problemstellungen gibt Tabelle 4.1 auf Seite 73.

Als Bemerkung zum zweiten Punkt sei gesagt, daß die auftretenden Monoide (vgl. Definition 4.1) (ausschließlich) $M = (\mathbb{R}, +, \leq)$ und $M = (\mathbb{R}, \max, \leq)$ sind. Das erste ist die Menge der reellen Zahlen, versehen mit ihrer gewöhnlichen Addition und Ordnung. Das zweite Monoid ist die Menge der reellen Zahlen, wobei das Ergebnis der (Monoid-)Addition zweier Zahlen als ihr Maximum in \mathbb{R} definiert ist; die Ordnung ist die gewöhnliche Ordnung in \mathbb{R} .

Maximin/Minimax-Probleme. Der Baum $T = (V, E, w)$ soll so in p Teilbäume zerlegt werden, daß das Gewicht des leichtesten Teilbaumes maximiert bzw. das Gewicht des schwersten Teilbaumes minimiert wird. In unserer Klassifikation sind dies die Probleme $T/\cdot/p/\text{maximin}$ und $T/\cdot/p/\text{minimax}$. Es sind allgemeine p -Baumzerlegungsprobleme (siehe Definition 4.4): Setze $\mathcal{T} = \{T' : T' \triangleleft T\}$, $M = (\mathbb{R}, \max, \leq)$ und $cT' = -wT$ bzw. $cT' = +wT$. Problem $T/\cdot/p/\text{maximin}$ wird untersucht in Frederickson [38] und in Perl und Schach [87]. $T/\cdot/p/\text{minimax}$ ist Gegenstand von Becker et al. [13] und von Perl und Vishkin [88] und wird in Frederickson [38] erwähnt.

Minimierung der Abweichung von Zielgewichten. Das erste Problem dieser Art ist $T/\cdot/\cdot/\|\cdot\|_\rho$, das heißt, die Minimierung der ρ -Norm des Vektors $\delta = (wT'_1 - \mu, \dots, wT'_s - \mu)$ der Abweichungen der Teilbaumgewichte von einem Zielwert μ . Es handelt sich um ein allgemeines Baumzerlegungsproblem (siehe Definition 4.3). Hierzu setzen wir $\mathcal{T} = \{T' : T' \triangleleft T\}$ und im Fall $\rho < \infty$ $M = (\mathbb{R}, +, \leq)$, $cT' = |wT' - \mu|^\rho$. (Damit wird die ρ -te Potenz der Norm minimiert, was äquivalent ist zur Minimierung der Norm selbst.) Im Fall $\rho = \infty$ ist $M = (\mathbb{R}, \max, \leq)$ und $cT' = |wT' - \mu|$ die erforderliche Definition. Während Problem $T/\cdot/\cdot/\|\cdot\|_\rho$ in der Literatur unseres Wissens keine Erwähnung findet, gibt es einige Arbeiten, die sich mit seiner p -BZP Variante $T/\cdot/p/\|\cdot\|_\rho$ befassen. Hierbei wird $\mu = wT/p$ angenommen,

es soll also die ρ -Norm der Abweichungen vom mittleren Gewicht der p Teilbäume minimiert werden. Messe [72] befaßt sich mit $T/\cdot/p/\|\cdot\|_1$ für die Baumklasse der Ketten. De Simone et al. [102] erweitern dies auf Sterne, Ketten, Raupen und 2-Spinnen. Das Problem $T/\cdot/p/\|\cdot\|_\infty$ für Ketten behandeln Liverani et al. [63].

Aus Anwendungssicht interessant ist ferner $T/\text{Zent}/\cdot/\|\cdot\|_\rho$. Ist $Z \subset V$ die Menge der Zentren, so ist $\mathcal{T} = \{T' : T' \triangleleft T, |V(T') \cap Z| = 1\}$ zu definieren, um das Problem in die Form eines BZP zu bringen. Weiter kann nun angenommen werden, daß für jedes Zentrum $z \in Z$ ein spezifisches Zielgewicht μ_z vorgegeben ist. Das heißt, es ist die ρ -Norm von $\delta = (wT'_z - \mu_z)_{z \in Z}$ über alle Zerlegungen $\{T'_z : z \in Z\}$ (wobei $z \in T'_z$) zu minimieren. Daraus ergibt sich unmittelbar die Wahl von cT' , $T' \in \mathcal{T}$. Problem $T/\text{Zent}/\cdot/\|\cdot\|_\rho$ ist bislang nicht untersucht worden.

Minimierung der Spannweite. In Problem $T/\cdot/p/\Delta$ ist das Ziel, den Baum T so in p Teilbäume zu zerlegen, daß die Differenz zwischen größtem und kleinstem Teilbaumgewicht minimiert wird. Im Gegensatz zu den anderen in diesem Abschnitt präsentierten Problemen, ist dies kein allgemeines Baumzerlegungsproblem. Der Grund ist, daß sich die Zielfunktion nicht als „Summe“ der Teilbaumgewichte in einem Monoid darstellen läßt. Lucertini et al. [64] behandeln $T/\cdot/p/\Delta$ für die Baumklasse der Ketten.

Probleme mit gewichtsbeschränkten Teilbäumen. Sind in einem Problem der gleichmäßigen Baumzerlegung Schranken für die Gewichte der Teilbäume vorgegeben, so erhält man häufig anstelle von Optimierungsproblemen Entscheidungsprobleme. So ist bei $T/L/p/\exists$ die Frage, ob es eine Zerlegung des Baumes in p Teilbäume gibt, die alle ein vorgegebenes Mindestgewicht haben. Es ist also ein p -BZP mit der Menge zulässiger Teilbäume $\mathcal{T} = \{T' : T' \triangleleft T, wT' \geq L\}$. Ist die Anzahl der Teilbäume nicht vorgegeben, so ist es interessant, sie zu maximieren. Dies ist Problem $T/L/\cdot/\max|\pi|$, das ein BZP ist, wie die Wahl $M = (\mathbb{R}, +, \leq)$ und $cT' = -1$ zeigt. $T/L/\cdot/\max|\pi|$ ist einfach zu lösen; zum Beispiel wird in Perl und Schach [87] auf dieses Problem eingegangen.

In einer Variante mit Zentren können individuelle Schranken L_z vorgegeben werden. In Problem $T/\text{Zent}, L_z/\cdot/\exists$ ist also die Frage, ob es eine Zerlegung von T gibt, in welcher jeder Teilbaum genau ein Zentrum z enthält und ein Gewicht von mindestens L_z hat. Für das BZP definieren wir $\mathcal{T} = \{T' : T' \triangleleft T, wT' \geq L_z \text{ mit } \{z\} = V(T') \cap Z\}$,

Analog zu den Problemen mit unteren Gewichtsschranken sind die Probleme $T/U/p/\exists$, $T/U/\cdot/\min|\pi|$ und $T/\text{Zent}, U_z/\cdot/\exists$ definiert, in denen die Teilbäume ein vorgegebenes Gewicht nicht überschreiten dürfen. Die Referenz zu $T/U/\cdot/\min|\pi|$ ist Kundu und Misra [61].

Ein großer Teil von Kapitel 5 betrifft die Lösung von Problemen, in welchen untere *und* obere Gewichtschränken vorgegeben sind. Das einfachste Problem ist $T/L, U/\cdot/\exists$: Gibt es eine Zerlegung von T , in welcher alle Teilbäume ein Gewicht zwischen oberer und unterer Schranke haben? Dies ist ein BZP wie die Festlegung $\mathcal{T} = \{T' : T' \triangleleft T, L \leq wT' \leq U\}$ zeigt. Das Problem findet sich unseres Wissens nicht in der Literatur. Die p -BZP Variante der Fragestellung, $T/L, U/p/\exists$, wird in Lucertini et al. [64] für den Fall von Ketten untersucht. Weiter läßt sich auch eine Variante unter der Berücksichtigung von Zentren definieren, dies ist das Problem $T/Zent, L_z, U_z/\cdot/\exists$, in welchem die Schranken vom jeweiligen Zentrum abhängen können.

4.3 Komplexität von Problemen der gleichmäßigen Baumzerlegung

Im folgenden bringen wir Ergebnisse zur Komplexität der Probleme der gleichmäßigen Baumzerlegung. Es wird sich dabei herausstellen, daß die Grenze zwischen polynomial lösbaren und \mathcal{NP} -schweren (und damit wahrscheinlich nicht polynomial lösbaren) Problemen quer durch die Problemstellungen aus Tabelle 4.1 hindurch verläuft. Probleme, in denen ein vorgegebenes Teilbaumgewicht bestmöglich zu approximieren ist, sind, außer im Fall der Maximumnorm, \mathcal{NP} -schwer. (Dieser Ausnahmefall wird in Kapitel 5 eingehend untersucht.) Die meisten anderen Probleme sind bekanntermaßen polynomial lösbar. Eine Zusammenfassung aller Ergebnisse bietet Tabelle 4.2 auf Seite 80.

4.3.1 \mathcal{NP} -schwere Probleme

Das bisher einzige Nachweis eines \mathcal{NP} -schweren Problems der gleichmäßigen Baumzerlegung stammt von De Simone et al. [102]. Die Autoren zeigen, daß $T/\cdot/p/\|\cdot\|_1$ für die Klasse der 2-Spinnen \mathcal{NP} -schwer ist. Dazu wird eine Transformation des Referenzproblems PARTITION (Garey und Johnson [40, S. 223]) verwendet.

Im folgenden verallgemeinern wir dieses Resultat, indem wir die Konstruktion etwas abwandeln, und zeigen so für eine Reihe weiterer Probleme der gleichmäßigen Baumzerlegung, daß sie auf 2-Spinnen \mathcal{NP} -schwer sind. Auf der anderen Seite stellt sich heraus, daß alle diese Probleme für 1-Spinnen, also Sterne, polynomial sind. Dies ist mit anderen Polynomialitäts-Resultaten Gegenstand des nächsten Abschnitts.

Satz 4.17

Für jedes $1 \leq \rho < \infty$ sind die Probleme $T/\cdot/\cdot/\|\cdot\|_\rho$, $T/\cdot/p/\|\cdot\|_\rho$ und $T/2-Zent/\cdot/\|\cdot\|_\rho$

Bezeichnung	BZP/ p -BZP	Zulässige Teilbäume T'	Monoid M	Kosten cT'	ggf. Literatur
Maximin-/Minimax-Probleme					
$T/\cdot/p/\max\min$	p -BZP	$T' \triangleleft T$	(\mathbb{R}, \max, \leq)	$-wT'$	[38, 87]
$T/\cdot/p/\min\max$	p -BZP	$T' \triangleleft T$	(\mathbb{R}, \max, \leq)	wT'	[13, 88]
Abweichung von Zielgewicht(en)					
$T/\cdot/p/\ \cdot\ _\rho$	BZP	$T' \triangleleft T$	$\rho < \infty: (\mathbb{R}, +, \leq)$	$\rho < \infty: wT' - \mu ^\rho$	
$T/\cdot/p/\ \cdot\ _\rho$	p -BZP	"	$\rho = \infty: (\mathbb{R}, \max, \leq)$	$\rho = \infty: wT' - \mu $	
$T/\text{Zent}/\cdot/\ \cdot\ _\rho$	BZP	$T' \triangleleft T, V(T') \cap Z = 1$	"	" $(\mu = wT/p)$	[72, 102]
			"	$\rho < \infty: wT' - \mu_z ^\rho, \{z\} = V(T') \cap Z$	
				$\rho = \infty: wT' - \mu_z $	
Spannweite					
$T/\cdot/p/\Delta$	—	$T' \triangleleft T$	—	—	[64]
Gewichtsbeschränkte Teilbäume					
$T/[L, U]/p/\exists$	p -BZP	$T' \triangleleft T, wT'[\geq L, \leq U]$	—	—	
$T/L/\cdot/\max \pi $	BZP	"	$(\mathbb{R}, +, \leq)$	-1	[87]
$T/U/\cdot/\min \pi $	BZP	"	"	+1	[61]
$T/L, U/\cdot/\exists$	BZP	$T' \triangleleft T, L \leq wT' \leq U$	—	—	
$T/L, U/p/\exists$	p -BZP	"	—	—	
$T/\text{Zent}, [L_z, U_z]/\cdot/\exists$	BZP	$T' \triangleleft T, wT'[\geq L_z, \leq U_z]$ mit $\{z\} = V(T') \cap Z$	—	—	
$T/\text{Zent}, L_z, U_z/\cdot/\exists$	BZP	$T' \triangleleft T, L_z \leq wT' \leq U_z$ mit $\{z\} = V(T') \cap Z$	—	—	

Tabelle 4.1: Problemstellungen der gleichmäßigen Baumzerlegung.

für die Baum-Klasse der 2-Spinnen \mathcal{NP} -schwer.

Beweis. Wir führen den Beweis durch Transformation von PARTITION auf die im Satz genannten Problemstellungen. Eine Instanz von PARTITION ist gegeben durch n positive Zahlen w_1, \dots, w_n ; es ist zu entscheiden, ob eine Teilmenge $I \subset N := \{1, \dots, n\}$ existiert, für die $wI = \sum_{i \in I} w_i = wN/2 = \frac{1}{2} \sum_{i \in N} w_i$ gilt.

Zu einer solche Instanz definieren wir eine 2-Spinne T mit Eckengewichten wie folgt. Ecke 0 sei „im Zentrum“ und dazu adjazent seien die Ecken $1, \dots, n$. Adjazent zu $i \in N$ sei jeweils die Ecke i' . Wir setzen $w_0 := 1$, Ecke i bekommt Gewicht w_i , und für i' setzen wir $w_{i'} := wN/2 + 1 - w_i/2$.

Das Gewicht von T ist somit $wT = 1 + wN + (n-1)wN/2 + n = (n+1)(1+wN/2)$. Für Problem $T/\cdot/p/\|\cdot\|_\rho$ setzen wir $p := n+1$, daraus ergibt sich $\mu = 1 + wN/2$. Für Problem $T/\cdot/\cdot/\|\cdot\|_\rho$ wählen wir μ gerade mit diesem Wert. Für Problem $T/2\text{-Zent}/\|\cdot\|_\rho$ setzen wir $Z := \{0\} \cup \{i' : i \in N\}$ und $\mu_z = \mu, z \in Z$.

Nun zeigen wir zweierlei; aus beidem zusammen folgt die Behauptung der Proposition.

1. Wenn die Antwort für das PARTITION-Problem positiv ist, dann gibt es eine Zerlegung π^* von T , die für jedes der drei Probleme zulässig ist, mit Wert $c\pi^* = (\sum_{i \in N} |w_i/2|^\rho)^{1/\rho}$.
2. Gibt es eine Zerlegung π von T mit $c\pi = (\sum_{i \in N} |w_i/2|^\rho)^{1/\rho}$, so gibt es ein $I \subset N$ mit $wI = wN/2$.

Zu 1.: Sei $I \subset N$ mit $wI = wN/2$. Wir definieren Teilbäume T'_0, T'_1, \dots, T'_n durch $V(T'_0) = \{0\} \cup I$ und

$$V(T'_i) = \begin{cases} \{i'\} & i \in I, \\ \{i, i'\} & i \in N - I, \end{cases} \quad \text{für } i \in N.$$

Es folgt $wT'_0 = 1 + wN/2 = \mu$ und $wT'_i = \mu \pm w_i/2, i \in N$. Daraus ergibt sich für $\pi^* = \{T'_0, T'_1, \dots, T'_n\}$ der Wert $c\pi^* = (\sum_{i \in N} |w_i/2|^\rho)^{1/\rho}$. Wegen $|\pi^*| = p$ und der Tatsache, daß $|T'_i \cap Z| = 1$ für $i = 0, 1, \dots, n$, ist π^* für alle drei Baumzerlegungsprobleme zulässig.

Zu 2.: Sei umgekehrt π eine Zerlegung mit $c\pi = (\sum_{i \in N} |w_i/2|^\rho)^{1/\rho}$. Sei $T'_0 \in \pi$ der Teilbaum, der die Ecke 0 enthält. Setze $N_0 := \{i \in N : i' \in T'_0\}$. Wir werden zeigen, daß $N_0 = \emptyset$ ist. Dann folgt, daß es Teilbäume T'_i mit $i' \in T'_i$ für $i \in N$ in der Zerlegung π gibt und daß für diese $wT'_i = \mu \pm w_i/2$ ist. Hieraus ergibt sich

$$(c\pi)^\rho \geq |wT'_0 - \mu|^\rho + \sum_{i \in N} |wT'_i - \mu|^\rho \geq \sum_{i \in N} \left| \frac{w_i}{2} \right|^\rho,$$

also $wT'_0 = \mu$ und damit ist $I = \{i \in N : i \in T'_0\}$ Lösung des PARTITION-Problems.

Angenommen, $N_0 \neq \emptyset$. Dann ist $wT'_0 \geq 1 + wN_0/2 + |N_0|\mu$, somit

$$|wT'_0 - \mu|^\rho \geq \left| 1 + \frac{wN_0}{2} + (|N_0| - 1)\mu \right|^\rho \geq \left| 1 + \frac{wN_0}{2} \right|^\rho \geq 1 + \sum_{i \in N_0} \left| \frac{w_i}{2} \right|^\rho.$$

Dies führt wegen

$$(c\pi)^\rho \geq |wT'_0 - \mu|^\rho + \sum_{i \in N - N_0} \left| \frac{w_i}{2} \right|^\rho$$

zum Widerspruch zur Voraussetzung über $c\pi$. \square

Die nächste Komplexitätsresultat betrifft die Zerlegung mit Gewichtsschranken auf Bäumen mit Zentren.

Proposition 4.18

Problem $T/2$ -Zent, $L_z, U_z / \cdot / \exists$ ist \mathcal{NP} -vollständig für die Klasse der 2-Spinnen.

Beweis. Das Problem ist offenbar in \mathcal{NP} . Um \mathcal{NP} -Vollständigkeit zu zeigen, transformieren wir wiederum das PARTITION-Problem. Eine Instanz sei gegeben durch $w_1, \dots, w_n > 0$. Die Ecken der 2-Spinne seien wie im Beweis von Proposition 4.17 numeriert. Wir setzen $w_0 = 0$ und $w_{i'} = 0$ für alle i' . Ferner sei w_i das Gewicht der Ecke i , $i \in N$. Sei $Z = \{0\} \cup \{i' : i \in N\}$. Setze $L_0 = U_0 = wN/2$, $L_{i'} = 0$ und $U_{i'} = w_i$.

Es ist nun sehr leicht zu sehen, daß eine Lösung des PARTITION-Problems genau dann existiert, wenn das spezifizierte Baumzerlegungsproblem vom Typ $T/2$ -Zent, $L_z, U_z / \cdot / \exists$ eine zulässige Zerlegung besitzt. \square

Bemerkung 4.19

Der Grund für die Aussage von Proposition 4.18 ist die Möglichkeit, die Schranken L_z, U_z für alle Zentren unabhängig voneinander zu wählen. Im nächsten Kapitel werden wir zeigen, daß das Problem $T/Zent, U_z - L_z = const / \cdot / \exists$ polynomial lösbar ist.

4.3.2 Polynomial lösbare Probleme

Nun wenden wir uns Aussagen über die Polynomialität von Problemen der gleichmäßigen Baumzerlegung zu. Dabei steht zunächst die Aussage der polynomialen Lösbarkeit im Vordergrund. Ein weiterer Schritt ist in einer solchen Situation die Suche nach möglichst effizienten polynomialen Algorithmen. Hierfür verweisen wir auf die entsprechende Literatur.

Daß das allgemeine Baumzerlegungsproblem für Sterne nicht polynomial ist, haben wir in Proposition 4.8 festgestellt. Daher soll zunächst gezeigt werden, daß dies

für die speziellen Probleme der gleichmäßigen Baumzerlegung anders ist. Dies betrifft insbesondere die im letzten Abschnitt auf 2–Spinnen als \mathcal{NP} –schwer erwiesenen Problemstellungen. Die Probleme $T/\text{Zent}/\cdot/\|\cdot\|_\rho$ und $T/\text{Zent}, L_z, U_z/\cdot/\exists$ sind für Sterne trivial lösbar, weil dabei die Zentrenbedingung die möglichen Zerlegungen determiniert.¹⁶ Für die anderen beiden Probleme verallgemeinert das folgende Ergebnis ein Resultat in De Simone et al. [102], betreffend Problem $T/\cdot/p/\|\cdot\|_1$ auf Sternen.

Satz 4.20

Sei $T = (V, E, w \geq 0)$ ein eckengewichteter Stern mit zentraler Ecke 0 und dazu adjazenten Ecken $1, 2, \dots, n$, wobei $oBdA$ $w_1 \leq w_2 \leq \dots \leq w_n$ gelte. In einer Zerlegung von T sei T'_0 der Teilbaum, der die Ecke 0 enthält. Dann gibt es optimale Zerlegungen für die Probleme $T/\cdot/\cdot/\|\cdot\|_\rho$ bzw. $T/\cdot/p/\|\cdot\|_\rho$, $1 \leq \rho \leq \infty$, mit $V(T'_0) = \{0, 1, \dots, k\}$ wobei $k \geq 0$ bzw. $k = n - p + 1$ ist.

Beweis. Der Beweis beruht auf einem Vertauschungsargument. Sei zunächst $1 \leq \rho < \infty$. Wir stützen uns auf die Konvexität der Funktion $\mathbb{R} \ni x \mapsto |x|^\rho$.

Es gilt offenbar: Ist $\varphi : \mathbb{R} \rightarrow \mathbb{R}$ konvex und $A, B, \Delta \in \mathbb{R}$ mit $A \geq B$ und $\Delta \geq 0$, so ist

$$\varphi(A) - \varphi(A - \Delta) \geq \varphi(B) - \varphi(B - \Delta).$$

Sei nun π^* eine optimale Zerlegung und $I := V(T'_0) - \{0\}$. Gibt es ein $i \in I$ und ein $i' \in \{1, \dots, n\} - I$ mit $w_i > w_{i'}$, so sei π' die Zerlegung, in welcher i' zu T'_0 hinzugefügt wird und i einen eigenen Teilbaum bildet. Wir zeigen $c\pi' \leq c\pi^*$. Setze dazu $A := w_0 + wI - \mu$, $B := w_i - \mu$ und $\Delta := w_i - w_{i'}$. Dann ist $A \geq B$ und $\Delta > 0$. Wir erhalten für $\varphi : x \mapsto |x|^\rho$

$$\begin{aligned} (c\pi^*)^\rho - (c\pi')^\rho &= |w_0 + wI - \mu|^\rho + |w_i - w_i + w_{i'} - \mu|^\rho \\ &\quad - |w_0 + wI - w_i + w_{i'} - \mu|^\rho - |w_i - \mu|^\rho \\ &= \varphi(A) + \varphi(B - \Delta) - \varphi(A - \Delta) - \varphi(B) \\ &\geq 0. \end{aligned}$$

Im Fall $\rho = \infty$ folgt $c\pi^* \geq c\pi'$ aus der Tatsache, daß für $A, B, \Delta \in \mathbb{R}$ mit $A \geq B$ und $\Delta \geq 0$ offenbar gilt

$$\max\{|A|, |B - \Delta|\} \geq \max\{|B|, |A - \Delta|\}.$$

□

¹⁶Ist die zentrale Ecke des Sterns Zentrum, gibt es nur eine zulässige Zerlegung, andernfalls gibt es $|Z|$ zulässige Zerlegungen. Siehe auch Korollar 4.22.

Korollar 4.21

Der Lösungsaufwand für $T/\cdot/\cdot/\|\cdot\|_\rho$ bzw. $T/\cdot/p/\|\cdot\|_\rho$ ($1 \leq \rho \leq \infty$) auf Sternen ist $O(n \log n)$ bzw. $O(n)$, wobei $n = |V| - 1$ die Anzahl der Edecken ist.

Beweis. Für Problem $T/\cdot/\cdot/\|\cdot\|_\rho$ dominiert das Sortieren der Edecken nach ihrem Gewicht den Aufwand, und für $T/\cdot/p/\|\cdot\|_\rho$ ist das $(n - p + 1)$ -kleinste Element in der Menge der Gewichte zu bestimmen. \square

Korollar 4.22

Sei T ein Stern wie in Satz 4.20 und $Z \subset V$ eine Menge von Zentren, $0 \notin Z$. In einer optimalen Lösung von $T/\text{Zent}/\cdot/\|\cdot\|_\rho$ enthält T'_0 dasjenige Zentrum $z \in Z$, für das $w_z - \mu_z$ minimal ist.

Beweis. Da T'_0 genau ein Zentrum enthält, liefert das Vertauschungsargument aus dem Beweis von Satz 4.20 die Behauptung, wenn wir $A := w_0 + wI - \mu_z$, $B := w_z - \mu_z$ und $\Delta := (w_z - \mu_z) - (w_{z'} - \mu_{z'})$ setzen. \square

Die Polynomialität der Probleme $T/\cdot/\cdot/\|\cdot\|_\rho$, $T/\cdot/p/\|\cdot\|_\rho$ und $T/\text{Zent}/\cdot/\|\cdot\|_\rho$ für Ketten folgt aus den Propositionen 4.7 und 4.11. Wir zeigen nun, daß aus Korollar 4.21 die Polynomialität dieser Probleme für Raupen abgeleitet werden kann. Für $T/\cdot/p/\|\cdot\|_1$ ist dieses Resultat in De Simone et al. [102] zu finden. Wir übernehmen die dortige Beweisidee.

Proposition 4.23

Für $1 \leq \rho \leq \infty$ sind die Probleme $T/\cdot/\cdot/\|\cdot\|_\rho$, $T/\cdot/p/\|\cdot\|_\rho$ und $T/\text{Zent}/\cdot/\|\cdot\|_\rho$ für Raupen polynomial.

Beweis. Eine Raupe $T = (V, E)$ besteht aus einem Pfad $P = \{v_0, v_1, \dots, v_s\}$ und dazu adjazenten Edecken. Die Idee ist nun, alle Teilbäume, die aus einem festen Abschnitt P' von P und einer beliebigen Teilmenge der dazu adjazenten Edecken bestehen, simultan zu betrachten. Hierzu wird (gedanklich) P' zu einer einzelnen Ecke zusammengezogen, unter Addition der Eckengewichte in P' . Dann entspricht das Optimieren über alle diese Teilbäume einem Zerlegungsproblem auf einem Baum mit Stern-Struktur.

Zur formalen Darstellung dieser Idee wählen wir v_0 als Wurzel von T , so daß $v_0 \prec v_1 \prec \dots \prec v_s$ gilt. Wir behandeln zunächst Problem $T/\cdot/\cdot/\|\cdot\|_\rho$, und untersuchen, wie es mit Algorithmus 4.1 auf Seite 56 gelöst werden kann. Ist $v \in V - P$, so ist v ein Blatt und die Berechnung von $C(v)$ ist trivial. Um für ein $v \in P$ den Wert $C(v)$ zu bestimmen, haben wir in (4.3) über alle $T' \in \mathcal{T}_v$ zu optimieren. Diese Menge teilen wir folgendermaßen auf: Sei für $u \in P$, $u \succeq v$

$$\mathcal{T}_{v,u} := \{T' \in \mathcal{T} : V(T') \cap P = \{u' : v \preceq u' \preceq u\}\}.$$

Offenbar ist

$$\mathcal{T}_v = \bigcup_{\substack{u \in P \\ u \succeq v}} \mathcal{T}_{v,u},$$

und wir können (4.3) schreiben als

$$C(v) = \min_{\substack{u \in P \\ u \succeq v}} \bigcup \left\{ \min \left\{ cT' + \sum_{u' \in S(T')} C(u') : T' \in \mathcal{T}_{v,u} \right\} \right\}.$$

Die innere Minimierung besteht gerade darin, in optimaler Weise eine Teilmenge der zu $P' = \{u' : v \preceq u' \preceq u\}$ adjazenten Blätter „abzuschneiden“. Wie bereits erläutert, kann diese Aufgabe mit Satz 4.20 gelöst werden.

Analog ergibt sich die Polynomialität von $T/\cdot/p/\|\cdot\|_\rho$ indem in (4.4) für \mathcal{T}_v obige Aufteilung verwendet wird. Und auch für $T/Zent/\cdot/\|\cdot\|_\rho$ wird mit dieser Idee Polynomialität bewiesen, wobei auf Korollar 4.22 zurückzugreifen ist. \square

Um zu einer brauchbaren Aufwandsabschätzung für die Probleme in Proposition 4.23 zu kommen, muß eine detailliertere Analyse der auszuführenden Operationen vorgenommen werden, als dies der gegebene Beweis zuläßt. Aus Platzgründen verzichten wir jedoch darauf. Ein großer Teil der Arbeit von De Simone et al. [102] besteht in der Entwicklung eines polynomialen Algorithmus mit guter Laufzeitabschätzung für $T/\cdot/p/\|\cdot\|_1$ auf Raupen.

Nun bringen wir einige andere Resultate über Polynomialität von Problemen aus Tabelle 4.1.

Satz 4.24

Die Probleme $T/L/\cdot/\max|\pi|$ und $T/U/\cdot/\min|\pi|$ sind mit linearem Aufwand lösbar (für beliebige Bäume).

Der Beweis für $T/L/\cdot/\max|\pi|$ ist sehr einfach; man findet ihn zum Beispiel bei Perl und Schach [87]. Für $T/U/\cdot/\min|\pi|$ sei auf Kundu und Misra [61] verwiesen.

Korollar 4.25

Die Probleme $T/[L,U]/p/\exists$ sind mit linearem Aufwand lösbar.

Beweis. Löse $T/L/\cdot/\max|\pi|$ bzw. $T/U/\cdot/\min|\pi|$ und vergleiche p mit dem optimalen Zielfunktionswert. \square

Korollar 4.26

Die Probleme $T/\cdot/p/\text{maximin}$ und $T/\cdot/p/\text{minimax}$ sind polynomial lösbar.

Beweis. Für $T/\cdot/p/\text{maximin}$: Bestimme den optimalen Zielfunktionswert durch Binärsuche über L im Intervall $[0, wT]$ und jeweiligem Lösen von $T/L/p/\exists$.

Eine analoge Vorgehensweise löst $T/\cdot/p/\text{minimax}$ mit Hilfe von $T/U/p/\exists$. \square

Der Aufwand für das Lösen von $T/\cdot/p/\text{maximin}$ beziehungsweise $T/\cdot/p/\text{minimax}$ ist damit $O(|V|\log wT)$. In der Literatur sind verschiedene Algorithmen mit streng polynomialem Aufwand (also ohne Abhängigkeit von $|\log wT|$) für diese Probleme zu finden. So lösen Perl und Schach [87] $T/\cdot/p/\text{maximin}$ mit der sogenannten *shifting* Methode in $O(p^2 \text{rd}(T) + pn)$. (Hierbei ist $\text{rd}(T)$ der Radius des Baumes, also die halbe Länge des längsten Pfades von T .) Von Frederickson [38] wird der Nachweis geführt, daß sich $T/\cdot/p/\text{maximin}$ sogar in $O(n)$ lösen läßt, dabei werden fortgeschrittene Datenstrukturen eingesetzt und der Algorithmus ist sehr kompliziert. Für $T/\cdot/p/\text{minimax}$ entwickeln Becker et al. [13] einen *shifting* Algorithmus mit Laufzeit $O(p^3 \text{rd}(T) + pn)$; verbessert wird dies von Perl und Vishkin [88] auf $O(\text{rd}(T)p(p + \log d) + n)$ (d ist der maximale Eckengrad in T). Auch für dieses Problem hält Frederickson [38] einen Algorithmus mit linearer Laufzeit für realisierbar.

Die nun folgenden Resultate waren unseres Wissens bisher nicht bekannt und gehören zu den wesentlichen Ergebnissen unserer Arbeit. Wir beweisen sie in Kapitel 5.

Satz 4.27

Die Probleme $T/L,U/\cdot/\exists$, $T/L,U/p/\exists$ und $T/Zent,U_z - L_z = \text{const}/\cdot/\exists$ sind polynomial lösbar.

Durch Binärsuche ergibt sich daraus die Polynomialität von Problemstellungen zur Minimierung der maximalen Abweichung vom Zielgewicht der Teilbäume:

Satz 4.28

Die Probleme $T/\cdot/\cdot/\|\cdot\|_\infty$, $T/\cdot/p/\|\cdot\|_\infty$ und $T/Zent/\cdot/\|\cdot\|_\infty$ sind polynomial lösbar.

Die Problemstellungen des Satzes 4.28 sind im Hinblick auf die Anwendungen in der Gebietsaufteilung besonders interessant, denn die maximale Abweichung der Bezirksgrößen von ihrem Mittel ist ein wichtiges Maß. Dieser Umstand unterstreicht die Bedeutsamkeit der hierfür erzielten Polynomialitäts-Resultate, die den Fall $\rho = \infty$ von der bei $\rho < \infty$ gegebenen Situation (siehe Satz 4.17) unterscheiden.

4.4 Komplexität verwandter Problemstellungen

Die Theoreme des vorangehenden Abschnitts betreffen Modelle für das gleichmäßige Zerlegen eines Baumes in Teilbäume. Eine Verallgemeinerung davon sind die entsprechenden Fragestellungen für das Zerlegen eines zusammenhängenden Graphen in zusammenhängende Subgraphen. Wir führen wir an dieser Stelle zur Ergänzung Komplexitätsresultate für solche Zerlegungsprobleme an. Es zeigt sich, daß alle Probleme \mathcal{NP} -schwer werden, wenn von Bäumen zu Graphen mit Zyklen übergegangen

Problem	Komplexität, Referenz				
	allg. Bäume	2–Spinnen	Sterne	Ketten	Raupen
$T/\cdot/p/\text{maximin}$	$O(n)$ bzw. $O(p^2 \text{rd}(T) + pn)$ [38] bzw. [87]				
$T/\cdot/p/\text{minimax}$	$O(n)$ bzw. $O(\text{rd}(T)p(p + \log d) + n)$ [38] bzw. [88]				
$T/\cdot/\cdot/\ \cdot\ _\rho$ $1 \leq \rho < \infty$	(\mathcal{NP} –schwer)	\mathcal{NP} –schwer Satz 4.17	$O(n \log n)$ Kor. 4.21	polynomial Prop. 4.7	polynomial Prop. 4.23
$T/\cdot/p/\ \cdot\ _1$	(\mathcal{NP} –schwer)	\mathcal{NP} –schwer [102]	$O(n)$ [102]	$O(np)$ [102]	$O(n^2p)$ [102]
$T/\cdot/p/\ \cdot\ _\rho$ $1 \leq \rho < \infty$	(\mathcal{NP} –schwer)	\mathcal{NP} –schwer Satz 4.17	$O(n)$ Kor. 4.21	polynomial Prop. 4.11	polynomial Prop. 4.23
$T/(2\text{–})\text{Zent}/\cdot/\ \cdot\ _\rho$ $1 \leq \rho < \infty$	(\mathcal{NP} –schwer)	\mathcal{NP} –schwer Satz 4.17	trivial Kor. 4.22	polynomial Prop. 4.7	polynomial Prop. 4.23
$T/\cdot/\cdot/\ \cdot\ _\infty$	$O(n^5 \log wT)$ Satz 5.23		$O(n \log n)$ Kor. 4.21		
$T/\cdot/p/\ \cdot\ _\infty$	$O(p^3 n^4 \log wT)$ Satz 5.24		$O(n)$ Kor. 4.21	$O(np \log p)$ [63]	
$T/\text{Zent}/\cdot/\ \cdot\ _\infty$	$O(p^2 n^4 \log wT)$ Satz 5.26		trivial Kor. 4.22		
$T/\cdot/p/\Delta$				$O(n^2 p \log n)$ [64]	
$T/L/\cdot/\max \pi $ bzw. $T/U/\cdot/\min \pi $	$O(n)$ [87] bzw. [61]				
$T/[L, U]/p/\exists$	$O(n)$ Satz 4.25				
$T/L, U/\cdot/\exists$	$O(n^4)$, falls $L \leq n(U - L)$; $O(n^5)$ sonst Satz 5.19			$O(n)$ [64]	
$T/L, U/p/\exists$	$O(p^3 n^4)$, falls $L \leq p(U - L)$; $O(pn^4)$ sonst Satz 5.20			$O(n)$ [64]	
$T/\text{Zent},$ $U_z - L_z = \text{const}/\cdot/\exists$	$O(p^2 n^4)$ Satz 5.22				
$T/2\text{–Zent}, L_z, U_z/\cdot/\exists$	(\mathcal{NP} –schwer)	\mathcal{NP} –schwer Prop. 4.18			

Tabelle 4.2: Komplexität von Problemen der gleichmäßigen Baumzerlegung.

wird. Zum Teil läßt sich auch nachweisen, daß die Probleme \mathcal{NP} -schwer im strengen Sinn sind, das heißt, auch bei „kleinen“ Eckengewichten noch \mathcal{NP} -schwer bleiben.

Die Konsequenz daraus ist, daß Modelle für die Gebietsaufteilung, die auf dem Zerlegen des Nachbarschaftsgraphen der KGE in zusammenhängende Subgraphen basieren, im allgemeinen nicht effizient gelöst werden können. Sich auf das Zerlegen eines geeignet gewählten Gerüsts dieses Graphen zurückzuziehen, wie dies zum Beispiel in Abschnitt 8.3 vorgenommen wird, erscheint daher als sinnvoller heuristischer Weg.

Die Verallgemeinerung der gleichmäßigen *Baumzerlegung* auf zusammenhängende Graphen wird in offensichtlicher Weise vorgenommen.

Gegeben sei ein zusammenhängender Graph $G = (V, E)$, dessen Ecken mit Gewichten $w_v \geq 0$, $v \in V$ versehen sind. Eine Zerlegung $\pi = \{G'_1, \dots, G'_p\}$ in zusammenhängende Subgraphen ist gegeben, wenn die $V(G'_i)$ eine Partition der Menge V bilden und alle G'_i zusammenhängend sind. Das Gewicht eines Subgraphen ist $w_{G'_i} = \sum_{v \in V(G'_i)} w_v$. Alle Modelle des Abschnitts 4.2.2 zur Messung der Gleichmäßigkeit (bezüglich der Subgraphengewichte) einer Zerlegung (siehe Tabelle 4.1) lassen sich unmittelbar von Bäumen auf die allgemeinere Situation zusammenhängender Graphen übertragen.

Im folgenden wird auf verschiedene Klassen von Graphen Bezug genommen. *Planare* Graphen sind dadurch charakterisiert, daß sie sich ohne Überschneidungen von Kanten in der Ebene „zeichnen“ lassen. *Bipartite* (oder paare) Graphen haben nur Zyklen gerader Länge, eine äquivalente Charakterisierung solcher Graphen ist durch die Möglichkeit einer 2-Färbung gegeben. Gittergraphen sind Graphen deren Eckmenge die Form $V = \{(i, j) : 1 \leq i \leq k, 1 \leq j \leq l\}$ mit $k, l \geq 1$ hat, wobei die Kanten waagrecht und senkrecht benachbarte Ecken verbinden, das heißt es ist $E = \{((i_1, j_1), (i_2, j_2)) : |i_1 - i_2| + |j_1 - j_2| = 1\}$. Gittergraphen sind planar und bipartit.

In der Notation unseres Klassifikationsschemas (siehe Abschnitt 4.2.1) werden planare Graphen mit P , bipartite Graphen mit B und Gittergraphen mit $Grid$ bezeichnet.

Beispiel 4.2

Problem $P/\cdot/p/\text{maximin}$ ist die Aufgabe, einen planaren Graphen so in p zusammenhängende Subgraphen zu zerlegen, daß das Gewicht des schwersten Subgraphen minimiert wird.

Bei Problem $Grid/\cdot/p/\text{minimax}$ ist ein Gittergraph in p zusammenhängende Subgraphen zu zerlegen und dabei das Gewicht des leichtesten Subgraphen zu maximieren. Dabei wird nicht verlangt, daß die Subgraphen selbst Gittergraphen sind.

Das Entscheidungsproblem, ob ein bipartiter Graph in zusammenhängende Sub-

graphen mit Gewicht zwischen den Schranken L und U zerlegt werden kann, ist mit $B/L, U/ \cdot / \exists$ bezeichnet.

Proposition 4.29

Alle in Tabelle 4.1 aufgeführten Problemstellungen sind für Gittergraphen mit $k = 3$ \mathcal{NP} -schwer.

Beweis. Von Becker et al. [12] wird die Aussage für $Grid/ \cdot / p/ \text{minimax}$ bewiesen. Dazu wird eine Transformation des Problems PARTITION verwendet. Wir zeigen, daß diese Transformation für alle Modelle in Tabelle 4.1 die Zugehörigkeit zur Komplexitätsklasse \mathcal{NP} -schwer liefert.

Sei w_1, \dots, w_n eine Instanz von PARTITION (vgl. Beweis von Satz 4.17). Seien $V = \{(i, j) : 1 \leq i \leq 3, 1 \leq j \leq n\}$ die Ecken eines Gittergraphen mit drei Reihen. Wir weisen den Ecken Gewichte wie folgt zu:

$$w(1, j) = w(3, j) = 0, \quad w(2, j) = w_j, \quad j = 1, \dots, n.$$

Sei $W := \sum_{j=1}^n w_j$. Es gebe eine Teilmenge $J \subset \{1, \dots, n\}$ mit $\sum_{j \in J} w_j = W/2$. Dann sind die beiden Mengen

$$V'_1 = \{(1, j) : j = 1, \dots, n\} \cup \{(2, j) : j \in J\}$$

und

$$V'_2 = \{(2, j) : j \notin J\} \cup \{(3, j) : j = 1, \dots, n\}$$

Eckenmengen zusammenhängender Subgraphen mit Gewicht $W/2$. Gibt es andererseits solche Subgraphen, dann ist die Antwort für vorliegende Instanz von PARTITION positiv.

Man überzeugt sich nun leicht, daß sich durch Lösen jeder beliebigen Problemstellung in Tabelle 4.1 (mit $Grid/ \dots$ anstelle T/ \dots) entscheiden läßt, ob zwei zusammenhängende Subgraphen mit Gewicht $W/2$ existieren. Dabei ist jeweils eine geeignete Instanz zu definieren. Zum Beispiel setzt man in $Grid/ \cdot / p/ \|\cdot\|_p$ $p = 2$ und $\mu = W/2$. Für $Grid/Zent/ \cdot / \|\cdot\|_p$ wählt man eine Ecke aus der ersten Reihe und eine aus der dritten als Zentren. In Problemen mit Schranken setzt man jeweils L und / oder U gleich $W/2$. \square

Aus Proposition 4.29 folgt, daß gleichmäßige Zerlegung in zusammenhängende Subgraphen auch für planare und für bipartite Graphen \mathcal{NP} -schwer ist. Darüberhinaus läßt sich für die meisten der Problemstellungen aus Tabelle 4.1 leicht zeigen, daß sie im Fall bipartiter Graphen sogar \mathcal{NP} -schwer im strengen Sinne sind. Dies beruht auf einem Komplexitätsresultat für folgendes Problem.

Definition 4.30 (Zerlegung in zwei zusammenhängende Subgraphen (2ZS))

Gegeben sei ein zusammenhängender bipartiter Graph $G = (V, E)$ und eine Schranke $U \geq |V|/2$.

Problem: Entscheide, ob es eine Zerlegung von V in zwei Mengen V_1 und V_2 gibt, mit $|V_i| \leq U$, $i = 1, 2$, wobei die von V_1 und V_2 erzeugten Subgraphen zusammenhängend sind.

Satz 4.31

Das Entscheidungsproblem 2ZS ist \mathcal{NP} -vollständig.

Dieses Ergebnis findet sich in der kontinuierlich fortgesetzten Liste der \mathcal{NP} -vollständigen Probleme (Johnson [56]). Es bildet die Grundlage für die nächste Proposition.

Proposition 4.32

Folgende Problemstellungen sind \mathcal{NP} -schwer im strengen Sinn:

$$\begin{array}{llll}
 B/\cdot/p/\text{maximin} & B/\cdot/p/\|\cdot\|_\rho & B/[L, U]/p/\exists & B/L, U/p/\exists \\
 B/\cdot/p/\text{minimax} & B/\text{Zent}/\cdot/\|\cdot\|_\rho & B/L/\cdot/\max|\pi| & B/U/\cdot/\min|\pi| \\
 & B/\cdot/p/\Delta & B/\text{Zent}, [L_z, U_z]/\cdot/\exists & B/\text{Zent}, L_z, U_z/\cdot/\exists
 \end{array}$$

Beweis. Eine Transformation von 2ZS auf jede der Problemstellungen ist sehr einfach, da 2ZS bereits ein Problem der gleichmäßigen Graphenzerlegung ist.

Ist p Teil der Probleminstanz, so setze man $p = 2$. Ist eine obere Schranke beteiligt, so wähle man sie gleich der Schranke U in 2ZS. Eine untere Schranke setze man zu $L := W - U$. Als Zielgewicht wähle man $\mu := |V|/2$.

Die angeführten Probleme mit Zentren haben dieselbe Komplexität wie 2ZS, da man durch sukzessive Wahl aller Eckenpaare als Zentren jede Instanz von 2ZS auf das Lösen einer polynomialen Anzahl Instanzen des Zentrenproblems zurückführen kann.

Für $B/\cdot/p/\Delta$ ist genau dann der optimale Zielfunktionswert kleiner oder gleich $2(U - |V|/2)$ (man beachte, daß nur in zwei Subgraphen zerlegt wird), wenn 2ZS lösbar ist. \square

Kapitel 5

Polynomiale Verfahren

In diesem Kapitel entwickeln wir mehrere Hauptergebnisse des Teils II unserer Arbeit. Sie betreffen die polynomiale Lösbarkeit von gleichmäßigen Baumzerlegungsproblemen, in denen untere *und* obere Gewichtsschranken für die Teilbäume vorgegeben sind, und die verwandten Probleme der Minimierung der maximalen Abweichung von vorgegebenen Zielgewichten.¹

Als Schlüssel wird sich das Lösen einer abgeschwächten Version des SUBSET SUM Problems herausstellen. Unter gewissen Voraussetzungen kann dies mit polynomialem Aufwand geschehen, basiert auf der Tatsache, daß Rucksack-Probleme mit polynomial beschränkten Koeffizienten in der Nebenbedingung polynomial lösbar sind. Da diese Eigenschaft auch für Rucksackprobleme mit baumförmigen sogenannten *Präzedenzbedingungen* besteht, können wir auch ein abgeschwächtes SUBSET SUM Problem mit Baumstruktur polynomial lösen, wenn geeignete Voraussetzungen erfüllt sind. In unseren Verfahren für die gleichmäßige Baumzerlegung bei zweiseitiger Gewichtsbeschränkung ist dies der Fall. So gewinnen wir Algorithmen mit polynomialem Aufwand.

5.1 Vorbereitungen

In diesem Abschnitt leiten wir die grundlegenden Resultate her, auf denen die polynomialen Algorithmen zur gleichmäßigen Baumzerlegung aufgebaut sind.

¹Vgl. hierzu die Sätze 4.27 und 4.28.

5.1.1 Das Problem WEAK SUBSET SUM (WSS)

5.1.1.1 Definition von WSS

SUBSET SUM ist ein bekanntes \mathcal{NP} -vollständiges Problem (Garey und Johnson [40, S. 223]). Es ist wie folgt definiert.

Definition 5.1 (SUBSET SUM Problem)

Gegeben sind n Zahlen $w_1, \dots, w_n \in \mathbb{Z} - \{0\}$ und ein $B \in \mathbb{Z}$.

Problem: Entscheide, ob es eine Menge $\emptyset \neq I \subseteq \{1, \dots, n\}$ gibt mit

$$wI = \sum_{i \in I} w_i = B.$$

Die Optimierungsvariante von SUBSET SUM lautet: Maximiere wI unter der Nebenbedingung $wI \leq U$. Für sie und für die Entscheidungsvariante gilt die folgende Aussage:

Satz 5.2 (Komplexität von SUBSET SUM)

Der Aufwand zur Lösung von SUBSET SUM ist $O(n \sum_i |w_i|)$. Für den Fall, daß alle w_i positiv sind, reduziert sich dies auf $O(n \max_i w_i)$.

Der ersten Aussage des Satzes liegt der wohlbekannte Algorithmus mit dynamischer Programmierung nach Bellman [14] zugrunde. Das zweite Ergebnis ist neuen Datums (Pisinger [90]) und beruht auf einer anders angelegten dynamischen Programmierung. Beide Verfahren lösen SUBSET SUM mit pseudopolynomialem Aufwand.

Wir führen nun eine Abschwächung von SUBSET SUM ein.

Definition 5.3 (WEAK SUBSET SUM Problem (WSS))

Gegeben sind n Zahlen $w_1, \dots, w_n \in \mathbb{Z} - \{0\}$ und Schranken $L, U \in \mathbb{Z}$, mit $L \leq U$.

Problem: Entscheide, ob es eine Menge $\emptyset \neq I \subseteq \{1, \dots, n\}$ gibt mit

$$L \leq wI \leq U. \tag{5.1}$$

5.1.1.2 Ein Verfahren zur Lösung von WSS

WSS kann offenbar durch die Optimierungsvariante von SUBSET SUM gelöst werden. Deshalb gilt Satz 5.2 sinngemäß auch für WSS. Jedoch machen die dahinter stehenden Algorithmen keinen Gebrauch von der Größe der Lücke $U - L$.

Im Falle positiver w_i und $U > L$ kann WSS mit einem Approximationsverfahren für (die Optimierungsvariante von) SUBSET SUM von Kellerer et al. [59, Th. 7,8] gelöst werden. Der Aufwand dafür ist $O(\min\{n/\varepsilon, n + 1/\varepsilon^2 \log(1/\varepsilon)\})$, mit $\varepsilon = (U - L)/U$. Dieser Algorithmus ist für unsere Zwecke allerdings nicht geeignet,

da wir später das WSS mit baumförmigen Präzedenzbedingungen lösen werden; eine Verallgemeinerung des Verfahrens auf diesen Fall ist nicht in direkter Weise möglich.

Deshalb entwickeln wir im folgenden ein anderes, einfacheres Verfahren für WSS. Es kann in 5.1.3 leicht auf den Fall mit Zusatzbedingungen angepaßt werden. Für den Spezialfall von WSS mit positiven w_i erreicht es jedoch nicht die Laufzeitabschätzung des Verfahrens von Kellerer et al. [59].

Die Voraussetzung für unser Verfahren ist $U - L > 0$. Es muß also ein „echtes“ WSS, das kein SUBSET SUM Problem ist, vorliegen. Das Verfahren basiert darauf, die w_i durch Division mit Rest in $w_i = b_i\Delta + c_i$ zu zerlegen. Dabei sind $b_i \in \mathbb{Z}$ und $0 \leq |c_i| < \Delta$ mit $\text{sign}(b_i) = \text{sign}(c_i) = \text{sign}(w_i)$ eindeutig bestimmt. Der Divisor Δ wird so klein gewählt, daß für je zwei Teilmengen $I_1, I_2 \subseteq \{1, \dots, n\}$ stets gilt $|c_{I_1} - c_{I_2}| \leq U - L$. Zum Beispiel ist $\Delta = (U - L)/n$ hinreichend klein.

Es sei $B^+ := \sum_i \max(b_i, 0)$ und $B^- := \sum_i \min(b_i, 0)$. Für $B \in \{B^-, \dots, B^+\}$ betrachten wir die beiden Rucksackprobleme

$$\underline{c}(B) = \min \left\{ \sum_{i \in I} c_i : \sum_{i \in I} b_i = B, I \subseteq \{1, \dots, n\} \right\} \quad (5.2)$$

und

$$\bar{c}(B) = \max \left\{ \sum_{i \in I} c_i : \sum_{i \in I} b_i = B, I \subseteq \{1, \dots, n\} \right\}. \quad (5.3)$$

Ist die Nebenbedingung $\sum_{i \in I} b_i = B$ erfüllbar, so können, weil $0 \leq \bar{c}(B) - \underline{c}(B) \leq U - L$ ist, nur drei Fälle eintreten:

1. $\underline{c}(B) + B\Delta > U$
2. $\bar{c}(B) + B\Delta < L$
3. $\{\underline{c}(B) + B\Delta, \bar{c}(B) + B\Delta\} \cap [L, U] \neq \emptyset$

Im dritten Fall löst die zu $\underline{c}(B)$ beziehungsweise zu $\bar{c}(B)$ gehörende Teilmenge I das WSS, in den anderen beiden Fällen gibt es keine solche Lösung, die gleichzeitig $bI = B$ erfüllt.

Unser Verfahren für WSS besteht nun darin, für alle $B \in \{B^-, \dots, B^+\}$ die Rucksackprobleme (5.2) und (5.3) zu lösen, und jeweils zu prüfen, welcher der drei Fälle vorliegt. Eine Instanz von WSS ist offenbar genau dann zulässig, wenn für mindestens ein B der dritte Fall eintritt.

Zum Lösen der Rucksackprobleme verwenden wir in bekannter Weise dynamische Programmierung. Sei zum Beispiel für (5.2)

$$C(j, B) = \min \left\{ \sum_{i \in I} c_i : \sum_{i \in I} b_i = B, I \subseteq \{1, \dots, j\} \right\}.$$

($C(j, B) = +\infty$, falls kein zulässiges I existiert.) Die $C(j, B)$ werden für $1 \leq j \leq n$ und $B \in \{B^-, \dots, B^+\}$ rekursiv berechnet durch

$$C(1, B) = \begin{cases} c_1 & B = b_1, \\ +\infty & B \neq b_1, \end{cases}$$

und für $j \geq 2$ (mit $b_j^+ = \max(b_j, 0)$, $b_j^- = \min(b_j, 0)$)

$$C(j, B) = \begin{cases} \min \{c_j + C(j-1, B - b_j), C(j-1, B)\} & B \in \{B^- + b_j^+, \dots, B^+ + b_j^-\}, \\ +\infty & \text{sonst.} \end{cases} \quad (5.4)$$

Am Ende ist mit $\underline{c}(B) = C(n, B)$ das Rucksackproblem (5.2) für alle B gelöst. Analog werden die $\bar{c}(B)$ berechnet.

Der Rechenaufwand zur Lösung des WSS ist somit $O(n(B^+ - B^-))$. Da $|b_i| \leq |w_i|/\Delta = n|w_i|/(U - L)$ ist, erhalten wir die erste Aussage des folgenden Satzes.

Satz 5.4 (Komplexität von WSS)

Der Aufwand zur Lösung von WSS ist $O(n^2 \sum_i |w_i|/(U - L))$. Für den Fall, daß alle w_i positiv sind, ist der Aufwand $O(n^2 U/(U - L))$.

Beweis. Die zweite Aussage nutzt die Tatsache aus, daß für nichtnegative b_i und c_i die Rucksackprobleme (5.2) und (5.3) nur für $B \in \{0, \dots, \lfloor U/\Delta \rfloor\}$ gelöst werden müssen. (Für größere B ist stets $\underline{c}(B) + B\Delta > U$). \square

Beim Vergleich der Sätze 5.2 und 5.4 zeigt sich, daß unser Verfahren für WSS im Vorteil gegenüber einer Lösung mittels SUBSET SUM ist, sobald $U - L > n$ beziehungsweise, bei positiven w_i , $U - L > n^2$ ist.² Unter geeigneten Voraussetzungen über die Größe der Lücke $U - L$ hat unser Verfahren streng polynomialen Aufwand:

Korollar 5.5

Ist $p \in \{1, \dots, n\}$ und $\max_i |w_i| \leq p(U - L)$, so löst unser Verfahren WSS mit Aufwand $O(pn^3)$. Sind alle w_i positiv und ist $L \leq p(U - L)$, so ist der Aufwand $O(pn^2)$.

Beweis. Beide Aussagen folgen direkt aus Satz 5.4, man beachte $\sum_i |w_i| \leq n \max_i |w_i|$ und $U/(U - L) = 1 + L/(U - L)$. \square

Die Aussagen von Korollar 5.5³ sind der Schlüssel für unsere polynomialen Algorithmen zur Baumzerlegung mit zweiseitigen Gewichtsschranken. In den nächsten Ab-

²Letzteres beruht auf dem Umstand, daß WSS für positive w_i und $\max_i w_i \leq U/n$ trivial ist.

³Sie widersprechen nicht der Tatsache, daß wir das Lösen von WSS mit Hilfe von Rucksackproblemen vornehmen. Das Rucksackproblem ist zwar \mathcal{NP} -schwer, aber die Instanzen, die unter den Voraussetzungen der Korollars gelöst werden müssen, haben die Eigenschaft, Nebenbedingungskoeffizienten polynomialer Größe aufzuweisen und sind somit polynomial lösbar.

schnitten werden wir zeigen, daß sie bestehen bleiben, wenn wir dem WSS baumförmige Präzedenzbedingungen hinzufügen.

5.1.1.3 WSS mit mehreren Bedingungen.

Der Umstand, daß WSS bei nicht zu kleiner Lückengröße $U - L$ polynomial lösbar ist, ist nicht mehr gewährleistet, wenn mehrere Bedingungen der Form (5.1) gleichzeitig zu beachten sind.

Eine Instanz sei gegeben durch k Paare von Schranken L_j, U_j und kn Zahlen w_{ji} ($j = 1, \dots, k, i = 1, \dots, n$). Das Problem, zu entscheiden, ob es eine Menge $I \subseteq \{1, \dots, n\}$ gibt, mit $L_j \leq \sum_{i \in I} w_{ji} \leq U_j$ für alle j , ist \mathcal{NP} -vollständig, falls $k \geq 2$ ist. Denn diese Fragestellung umfaßt die Entscheidungsvariante des Rucksackproblems, auch wenn die Lücken $U_j - L_j$ beliebig groß gewählt werden.

Einen Spezialfall mit $k = 2$, den wir später benötigen, bringt der folgende Satz. Er behandelt das Problem, gleichzeitig eine Bedingung der Form (5.1) und eine Gleichung zu erfüllen.

Satz 5.6

Gegeben seien

- nichtnegative Zahlen $w_{11}, w_{12}, \dots, w_{1n} \in \mathbb{Z}_{\geq 0}$, höchstens $m \leq n$ davon positiv,
- Schranken $0 \leq L < U$ und
- ganze Zahlen $w_{21}, w_{22}, \dots, w_{2n} \in \mathbb{Z}$.

Es sei $W^- = \sum_{i=1}^n \min(w_{2i}, 0)$ und $W^+ = \sum_{i=1}^n \max(w_{2i}, 0)$. Die Frage, ob eine Menge $I \subseteq \{1, \dots, n\}$ existiert, für die

$$L \leq w_1 I = \sum_{i \in I} w_{1i} \leq U \quad \text{und} \quad w_2 I = \sum_{i \in I} w_{2i} = W$$

ist, kann für alle $W \in \{W^-, \dots, W^+\}$ mit einem Aufwand von

$$O\left(n \frac{mU}{U-L} \sum_{i=1}^n |w_{2i}|\right)$$

entschieden werden.

Beweis. Wir zerlegen die w_{2i} durch Division mit Rest, wie oben die w_i , in $w_{2i} = b_i \Delta + c_i$. ($b_i \in \mathbb{Z}_{\geq 0}$, $0 \leq c_i < \Delta$.) Dabei wählen wir $\Delta = (U - L)/m$. Da höchstens m der c_i positiv sind, ist $|cI_1 - cI_2| \leq U - L$ für alle $I_1, I_2 \subseteq \{1, \dots, n\}$. Die gegebene Problemstellung kann deshalb durch das Lösen der Rucksackprobleme mit doppelter Nebenbedingung

$$\underline{c}(B, W) = \min \{cI : bI = B, w_2 I = W, I \subseteq \{1, \dots, n\}\}$$

und

$$\bar{c}(B, W) = \max \{cI : bI = B, w_2I = W, I \subseteq \{1, \dots, n\}\}$$

für alle $B \in \{0, \dots, \lfloor U/\Delta \rfloor\}$ und $W \in \{W^-, \dots, W^+\}$ entschieden werden.

Diese Rucksackprobleme werden mit dynamischer Programmierung mit einem Aufwand von $O(n \lfloor U/\Delta \rfloor (W^+ - W^-))$ gelöst. (Man betrachte Zwischenlösungen der Form $C(j, B, W)$ und eine Rekursion analog (5.4).) \square

Noch spezieller als Satz 5.6 ist das folgende Ergebnis. Darin sind die w_{2i} Null oder Eins, die w_{1i} dürfen auch negativ sein. Der Lösungsaufwand hat dieselbe Komplexität wie der des einfachen WSS nach Definition 5.3. Auch diesen Satz werden wir später verwenden.

Satz 5.7

Gegeben seien

- ganze Zahlen $w_{11}, w_{12}, \dots, w_{1n} \in \mathbb{Z}$
- Schranken $L < U$ und
- Zahlen $w_{21}, w_{22}, \dots, w_{2n} \in \{0, 1\}$.

Die Frage, ob eine Menge $I \subseteq \{1, \dots, n\}$ existiert, für die

$$L \leq w_1I = \sum_{i \in I} w_{1i} \leq U \quad \text{und} \quad w_2I = \sum_{i \in I} w_{2i} = W$$

ist, kann für $W \in \{0, 1\}$ mit einem Aufwand von

$$O\left(n^2 \sum_{i=1}^n |w_{1i}| / (U - L)\right)$$

entschieden werden.

Beweis. Wie im Beweis von Satz 5.6 werden die w_{1i} zerlegt in $w_{1i} = b_i\Delta + c_i$ ($b_i \in \mathbb{Z}$, $0 \leq |c_i| < \Delta$, $\text{sign}(c_i) = \text{sign}(w_{1i})$), hier mit dem Divisor $\Delta = (U - L)/n$. Es sind die angeführten Rucksackprobleme $\underline{c}(B, W)$ und $\bar{c}(B, W)$ zu lösen, hier jedoch für $B \in \{B^-, \dots, B^+\}$ und $W \in \{0, 1\}$. ($B^- = \sum_i \min(b_i, 0)$, $B^+ = \sum_i \max(b_i, 0)$). Der Aufwand hierfür ist $O(n(B^+ - B^-))$, die durch die zweite Nebenbedingung $w_2I = W$ hinzugefügte Komplexität ist lediglich ein konstanter Faktor. \square

5.1.2 Rucksackprobleme mit Präzedenzbedingungen

Die Herleitung von Satz 5.4 zeigt, daß das Lösen des WSS im wesentlichen auf Rucksackprobleme zurückgeführt wird. Als nächsten Schritt zur Untersuchung des WSS mit Baumstruktur stellen wir in diesem Abschnitt Material über das Rucksackproblem mit Präzedenzbedingungen zusammen.

5.1.2.1 Definitionen

Definition 5.8 (Rucksackproblem mit Präzedenzbedingungen)

Gegeben sind n Gegenstände $1, 2, \dots, n$ und eine Ordnung \prec auf $N = \{1, \dots, n\}$. Gegenstand i hat das „Gewicht“ $b_i \in \mathbb{Z}$ (siehe Bemerkung 5.9) und die Kosten $c_i \in \mathbb{Z}$. Ferner ist $B \in \mathbb{Z}$ die Kapazität des Rucksacks.

Problem: Bestimme $I \subseteq N$, $I \neq \emptyset$, das $\sum_{i \in I} c_i$ minimiert und die Bedingungen

$$\sum_{i \in I} b_i = B \quad \text{und} \quad i \in I, j \prec i \implies j \in I$$

erfüllt.

Bemerkung 5.9

Unsere Definition des Rucksackproblems mit Präzedenzbedingungen unterscheidet sich von der Literatur (Boyd [19], Leensel et al. [62], Park und Park [85], Samphaiboon und Yamada [94]) darin, daß wir auch $b_i < 0$ zulassen und fordern, daß der Rucksack mit Gleichheit gefüllt wird. Insofern ist die Interpretation der b_i als Gewichte und von B als Kapazität nicht ganz treffend. Wir verwenden jedoch der Einfachheit halber die übliche Terminologie.

Das Rucksackproblem mit Präzedenzbedingung ist \mathcal{NP} -schwer im strengen Sinn (Garey und Johnson [40, S. 247]). Für uns interessant ist der spezielle Fall, in dem die Ordnung \prec eine baumförmige Struktur hat. Dann ist das Problem, wie das gewöhnliche Rucksackproblem, in pseudopolynomialer Zeit lösbar. Dieses Resultat wird von Garey und Johnson [40, S. 248] erwähnt.

Definition 5.10 (Baum-Rucksackproblem (TKP))

Gegeben ist die Menge $V = \{1, \dots, n\}$ als Ecken eines Wurzelbaumes $T = (V, E)$. Die entsprechende Ordnung auf V (siehe 4.1.1.1) sei mit \prec bezeichnet. Zu $v \in V$ sind $b_v \in \mathbb{Z}$ Gewicht und $c_v \in \mathbb{Z}$ Kosten von v . Weiter ist die Kapazität B gegeben.

Problem: Löse das Rucksackproblem (im Sinn der Definition 5.8) mit den Gegenständen $v \in V$ und den durch \prec gegebenen Präzedenzbedingungen.

Das TKP⁴ ist also die Aufgabe, einen die Wurzel enthaltenden (nichtleeren) Teilbaum von T mit Gewicht B zu finden, der minimale Kosten hat.

5.1.2.2 Algorithmen für TKP

Für das TKP sind verschiedene pseudopolynomiale Algorithmen vorgeschlagen worden. Sie basieren alle auf dynamischer Programmierung, unterscheiden sich jedoch darin, wie der Baum traversiert wird. Der übliche *bottom-up* Ansatz⁵ führt zu einem

⁴Die englische Bezeichnung *tree knapsack problem* wurde von Johnson und Niemi [57] eingeführt.

⁵Wir haben ihn in Abschnitt 4.1.3 zur Lösung des allgemeinen Baumzerlegungsproblems verwendet.

Aufwand von $O(n\bar{B}^2)$, $\bar{B} = \sum_{v=1}^n |b_v|$. Johnson und Niemi [57] lösen das TKP mit einem sogenannten *left-to-right* Ansatz; mit Aufwand $O(n\bar{C})$, $\bar{C} = \sum_{v=1}^n |c_v|$.

Da wir auf die Lösung des WEAK SUBSET SUM Problems mit Baum-Präzedenz abzielen und ausnutzen wollen, daß die b_v polynomial sein können, ist für unsere Zwecke am besten geeignet der *depth-first* Ansatz von Cho und Shaw [22]. Er führt zu einem Aufwand von $O(n\bar{B})$, dies entspricht dem Aufwand für das gewöhnliche Rucksackproblem. Aufgrund unserer abweichenden Definition des TKP (Bemerkung 5.9) ändern wir das Verfahren von Cho und Shaw [22] leicht ab.

Die folgende Darstellung ist auf das Wesentliche beschränkt und unterscheidet sich (durch die Verwendung rekursiver Prozeduren) deutlich von der bei Cho und Shaw [22]. (Dort findet der Leser ein ausführliches Rechenbeispiel, das gegebenenfalls zur Einsicht in das Verfahren beitragen kann.)

Notation. Im folgenden setzen wir für das TKP voraus, daß die Numerierung der Gegenstände eine Traversierung des Baumes in Tiefensuche induziert. Somit ist $v = 1$ die Wurzel und für jedes $v \in V$ gilt, daß die Gegenstände $u \succ v$ die Nummern $v + 1, \dots, v + l$ haben. Den Gegenstand $\max\{u : u \succeq v\} = v + l$ bezeichnen wir mit $\text{last}(v)$ (auch wenn v ein Blatt ist). Es ist der letzte Gegenstand bei der Tiefensuche-Traversierung von T_v .⁶ Daraus folgt: Sind die Söhne $S_v = \{u_1, \dots, u_k\}$ von v nach wachsender Numerierung sortiert, so ist $V(T_v) = \{v, v + 1, \dots, v + l\} = \{v, u_1, \dots, \text{last}(u_1), \dots, \dots, u_k, \dots, \text{last}(u_k)\}$ und $\text{last}(u_k) = \text{last}(v)$. Ferner ist $\text{last}(1) = n$.

Lösung des TKP. Die dynamische Programmierung berechnet im „Vorwärts“-Teil (Algorithmus 5.1) Zwischenergebnisse $C(v, u, b)$, die den optimalen Zielfunktionswert für das Teil-TKP mit durch $\{1, \dots, u\}$ gegebenen Gegenständen und Kapazität b angeben, unter der Bedingung, daß v in der Lösung enthalten ist. Dabei ist $u \in \{v, \text{last}(u_1), \dots, \text{last}(u_k)\}$ und $b \in \{B^-, \dots, B^+\}$ mit $B^- = \sum_v \min(b_v, 0)$ und $B^+ = \sum_v \max(b_v, 0)$.

Um eine kompakte Darstellung des Verfahrens zu ermöglichen, fügen wir einen Dummy-Gegenstand 0 hinzu mit $0 = \text{pre}(1)$ und $b_0 = c_0 = 0$. Die Prozedur `TKP_vorwärts` initialisiert und startet die Berechnung der $C(v, u, b)$. Am Ende ist mit $C(1, n, B)$ der optimale Zielfunktionswert des TKP ermittelt.

Die Prozedur `vorwärts` berechnet für ihr Argument v mit Söhnen $u_1 < u_2 < \dots < u_k$ sukzessive die Werte $C(v, v, b)$, $C(v, \text{last}(u_1), b)$, \dots , $C(v, \text{last}(u_k), b)$. Während $C(v, v, b)$ in den Zeilen 7 und 8 initialisiert wird, geschieht die Berechnung von

⁶Wir erinnern an die in 4.1.1.1 eingeführte Notation.

Algorithmus 5.1 Rechne vorwärts TKP

```

1: procedure TKP_vorwärts begin
2:    $C(0, 0, b) := +\infty, \quad b \in \{B^-, \dots, B^+\} - \{0\}$ 
3:    $C(0, 0, 0) := 0$ 
4:   vorwärts(1)
5: end

6: procedure vorwärts( $v$ ) begin /* berechnet  $C(v, \text{last}(v), b)$  für alle  $b$  */
7:    $C(v, v, b) := C(\text{pre}(v), v - 1, b - b_v) + c_v, \quad b \in \{B^- + b_v^+, \dots, B^+ + b_v^-\},$ 
8:    $C(v, v, b) := +\infty,$  sonst.
9:   for all  $u \in S_v$  do /* aufsteigend sortiert */
10:     vorwärts( $u$ )
11:      $C(v, \text{last}(u), b) := \min\{C(v, u - 1, b), C(u, \text{last}(u), b)\}, \quad b \in \{B^-, \dots, B^+\}$ 
12:   end for
13: end

```

$C(v, \text{last}(u_i), b)$ in Zeile 11. Dabei sind zwei Fälle möglich: Entweder enthält die optimale Teillösung nicht den Gegenstand u_i , dann sind nur Gegenstände aus $\{1, \dots, \text{last}(u_{i-1}) = u - 1\}$ in der Teillösung, oder aber u_i ist Element der optimalen Teillösung, dann ist deren Wert bereits beim rekursiven Aufruf von **vorwärts**(u_i) in Zeile 10 berechnet worden. (Für $i = 1$ gilt der erste Fall sinngemäß wegen $u_1 - 1 = v$.)

Es ist leicht zu sehen, daß die Aufrufe von **vorwärts** beim Ablauf von Algorithmus 5.1 gerade in der Reihenfolge der Ecken-Numerierung erfolgen. Diese entspricht einer Tiefensuche-Traversierung des Präzedenzen-Baumes und erklärt so die Bezeichnung als *depth first* Ansatz bei Cho und Shaw [22].

Rückwärtsrechnung. Da die Rückwärtsrechnung für die dynamische Programmierung zur Lösung des TKP nicht ganz offensichtlich ist, geben wir den entsprechenden Algorithmus an. Er setzt voraus, daß in den Variablen $x(v, u, b)$ festgehalten wird, welcher Term in Zeile 11 von **vorwärts** das Minimum ergibt. Genauer setzen wir $x(v, u, b) = 0$, falls das Minimum durch $C(v, u - 1, b)$ gegeben ist (u ist nicht in der optimalen Teillösung), und $x(v, u, b) = 1$ im anderen Fall ($C(u, \text{last}(u), b)$ ist das Minimum).

Algorithmus 5.2 berechnet die Menge V' der Gegenstände in der optimalen Lösung, falls das TKP überhaupt eine Lösung besitzt. Die *globale* Variable b enthält das Gewicht des noch zu konstruierenden Teils des Lösungsteilbaumes. Wesentlich ist, daß die Söhne der Ecke v in der Prozedur **rückwärts** nun in der Reihenfolge absteigender Nummerierung bearbeitet werden (Zeile 10).

Algorithmus 5.2 Rechne rückwärts TKP

```

1: procedure TKP_rückwärts begin
2:   if  $C(1, n, B) = +\infty$  then
3:     STOP /* TKP unlösbar */
4:   else
5:      $V' := \emptyset$ ;  $b := B$ ; rückwärts(1)
6:   end if
7: end

8: procedure rückwärts( $v$ ) begin
9:    $V' := V' \cup \{v\}$ 
10:  for all  $u \in S_v$  do /* absteigend sortiert */
11:    if  $x(v, u, b) = 1$  then
12:      rückwärts( $u$ )
13:    end if
14:  end for
15:   $b := b - b_v$ 
16: end

```

Aufwand zur Lösung des TKP. Wie bereits definiert, ist $\bar{B} = \sum_{v=1}^n |b_v| = B^+ - B^-$. Es ist leicht zu sehen, daß die folgende Aufwandsabschätzung gilt:

Proposition 5.11

Die Vorwärtsrechnung mit Algorithmus 5.1 hat Zeit- und Speicheraufwand $O(n\bar{B})$; der Zeitaufwand für die Rückwärtsrechnung mit Algorithmus 5.2 ist $O(n)$. Somit sind Zeit- und Speicheraufwand zur Lösung des TKP $O(n\bar{B})$.

Bemerkung 5.12

Die Vorwärtsrechnung ermittelt den optimalen Zielfunktionswert des TKP nicht nur für die Kapazität B , sondern für alle Kapazitäten $b \in \{B^-, \dots, B^+\}$. (Für andere „rechte Seiten“ ist das TKP auf jeden Fall unlösbar.) Da für jedes solche b die Rückwärtsrechnung in linearer Zeit durchgeführt werden kann, lassen sich die optimalen Lösungen des TKP für alle „interessanten“ Kapazitäten mit derselben Zeitkomplexität $O(n\bar{B})$ ermitteln, wie die Bestimmung der einen Lösung mit rechter Seite B .

5.1.3 Das Problem TREE WEAK SUBSET SUM (TWSS)

Wir führen nun das Konzept der baumförmigen Präzedenzbedingungen und das WEAK SUBSET SUM Problem aus 5.1.1 zusammen. Das dadurch gegebene Problem

TWSS ist der Schlüssel zur Lösung von Baumzerlegungsproblemen mit zweiseitiger Gewichtsbeschränkung. Problem TWSS wurde unseres Wissens in der Literatur noch nicht betrachtet.

5.1.3.1 Definition von TWSS

Definition 5.13 (TREE WEAK SUBSET SUM Problem (TWSS))

Gegeben ist ein gewichteter Wurzelbaum $T = (V, E, w)$ mit Wurzel $v_0 \in V$ und Eckengewichten $w_v \in \mathbb{Z}$ und ferner Schranken $L, U \in \mathbb{Z}$ mit $L \leq U$.

Problem: Entscheide, ob es einen Teilbaum $T' \triangleleft T$ gibt, der die Wurzel enthält und $L \leq wT' \leq U$ erfüllt.

Bemerkung 5.14

Für das WSS (Definition 5.3) haben wir die triviale Voraussetzung $w_v \neq 0$ gemacht. In einem TWSS können dagegen Ecken mit Gewicht 0 auftreten. Weiter ist festzustellen, daß das Gewicht der Wurzel v_0 irrelevant für die Problemstellung ist, denn v_0 muß auf jeden Fall Element der Lösung sein. Somit kann das Gewicht der Wurzel um einen beliebigen Wert verändert werden, wenn gleichzeitig L und U um denselben Wert abgeändert werden.

5.1.3.2 Lösung von TWSS

TWSS kann mittels TKP gelöst werden, indem für alle rechten Seiten

$$B \in \{W^-, \dots, W^+\} \cap [L, U]$$

mit

$$W^- = \sum_v \min(w_v, 0), \quad W^+ = \sum_v \max(w_v, 0),$$

die Zulässigkeit von TKP mit Gewichten w_v geprüft wird; dabei ist die Wahl der Zielfunktionskoeffizienten beliebig. Somit ist TWSS lösbar mit Aufwand $O(n \sum_{v \neq v_0} |w_v|)$. (Man ändere w_{v_0} auf 0 ab.)

Dies ist jedoch ungünstig, sobald $U - L > n$ ist, denn dann liefert der folgende Satz eine bessere Abschätzung. Er entspricht Satz 5.4 für das WSS.

Satz 5.15

Sei $U > L$. Dann ist der Aufwand zur Lösung von TWSS $O(n^2 \sum_{v \neq v_0} |w_v| / (U - L))$, $n = |V|$. Für den Fall, daß alle w_v nichtnegativ sind ($v \neq v_0$), ist der Aufwand $O(n^2 U / (U - L))$.

Beweis. Die Vorgehensweise ist identisch mit derjenigen in der Herleitung von Satz 5.4. Dort werden die Größen $\underline{c}(B)$ und $\bar{c}(B)$ durch das Lösen von Rucksackproblemen mit Aufwand $O(n(B^+ - B^-))$ bestimmt. Hier sind nun analog TKPs zu lösen, dazu wird das Verfahren aus dem vorangehenden Abschnitt verwendet, das denselben Rechenaufwand hat (siehe Bemerkung 5.12). Wie ausgeführt, ist das Gewicht w_{v_0} irrelevant. Daher wird es in der Aussage des Satzes nicht berücksichtigt. \square

Zusammenfassend geben wir das Verfahren zur Lösung von TWSS im Fall $U - L > |V|$ in Pseudo-Code Darstellung an. Der Fall $U - L \leq |V|$ wird, wie oben ausgeführt, als TKP mit den Verfahren aus dem letzten Abschnitt behandelt.

Algorithmus 5.3 Lösung des TWSS für $U - L > |V|$

Input: $T = (V, E, w)$, v_0 , L, U gemäß Definition 5.13

Output: „unzulässig“ oder $T' \triangleleft T$ mit $v_0 \in T'$ und $L \leq wT' \leq U$.

```

1:  $\Delta := (U - L)/|V|$ 
2:  $b_{v_0} = c_{v_0} = 0$ ;  $L := L - w_{v_0}$ ;  $U := U - w_{v_0}$ 
3: für alle  $v \neq v_0$  berechne  $b_v \in \mathbb{Z}, c_v$  mit
      
$$b_v \Delta + c_v = w_v, \quad 0 \leq |c_v| < \Delta, \quad \text{sign}(b_v) = \text{sign}(c_v) = \text{sign}(w_v)$$

4: if  $w_v \geq 0$  für alle  $v \neq v_0$  then
5:    $B^- := 0$ ;  $B^+ := \lfloor U/\Delta \rfloor$ 
6: else
7:    $B^- := \sum_v \min(b_v, 0)$ ;  $B^+ := \sum_v \max(b_v, 0)$ 
8: end if
9: for all  $B \in \{B^-, \dots, B^+\}$  do
10:   $\underline{c}(B) := \min\{cT' : T' \triangleleft T, v_0 \in T', bT' = B\}$ 
11:   $\bar{c}(B) := \max\{cT' : T' \triangleleft T, v_0 \in T', bT' = B\}$ 
12: end for
13: for all  $B \in \{B^-, \dots, B^+\}$  do
14:   if  $L \leq \underline{c}(B) + B\Delta \leq U$  then
15:     gib zu  $\underline{c}(B)$  gehörendes  $T'$  aus; STOP
16:   end if
17:   if  $L \leq \bar{c}(B) + B\Delta \leq U$  then
18:     gib zu  $\bar{c}(B)$  gehörendes  $T'$  aus; STOP
19:   end if
20: end for
21: gib_aus „unzulässig“
```

Der wesentliche Teil des Verfahrens steckt in den Zeilen 9 bis 12 und wird mittels Algorithmus 5.1 in $O(|V|(B^+ - B^-))$ Schritten ausgeführt. (Für die Berechnung von

\bar{c} ist dort in Zeile 11 das \min durch ein \max zu ersetzen.) Im Fall der Lösbarkeit erfolgt die Konstruktion des auszugebenden Teilbaumes T' (Zeilen 15, 18) durch Rückwärtsrechnung gemäß Algorithmus 5.2.

TWSS mit zwei Bedingungen. Die bisherigen Ausführungen haben gezeigt, daß das Lösen von TWSS völlig analog zum Lösen von WSS ist. Daher ist unmittelbar klar, daß auch die Sätze 5.6 und 5.7 direkt übertragen werden können. Wir benötigen dies in den nächsten Abschnitten.

Satz 5.16

Gegeben seien ein Wurzelbaum $T = (V, E, w_1, w_2)$ mit zweifachen Eckengewichten $w_{1v} \in \mathbb{Z}_{\geq 0}$ und $w_{2v} \in \mathbb{Z}$. Höchstens $m \leq |V|$ der w_{1v} seien positiv. Weiter seien $0 \leq L < U$ zwei Schranken und $W^- = \sum_v \min(w_{2v}, 0)$, $W^+ = \sum_v \max(w_{2v}, 0)$.

Die Frage, ob ein Teilbaum $T' \triangleleft T$ existiert, der die Wurzel enthält und für den

$$L \leq w_1 T' \leq U \quad \text{und} \quad w_2 T' = W$$

ist, kann für alle $W \in \{W^-, \dots, W^+\}$ mit einem Aufwand von

$$O\left(|V| \frac{mU}{U-L} \sum_{v \in V} |w_{2v}|\right)$$

entschieden werden.

Satz 5.17

Gegeben seien ein Wurzelbaum $T = (V, E, w_1, w_2)$ mit Wurzel v_0 und zweifachen Eckengewichten $w_{1v} \in \mathbb{Z}$ und $w_{2v} \in \{0, 1\}$. Weiter seien $L < U$ zwei Schranken.

Die Frage, ob ein Teilbaum $T' \triangleleft T$ existiert, der die Wurzel enthält und für den

$$L \leq w_1 T' \leq U \quad \text{und} \quad w_2 T' = W$$

ist, kann für $W \in \{0, 1\}$ mit einem Aufwand von

$$O\left(|V|^2 \sum_{v \in V - \{v_0\}} |w_{1v}| / (U - L)\right)$$

entschieden werden.

Hier ist die Größe von w_{1v_0} unerheblich, da eine Veränderung dieses Wertes bei gleichzeitiger Veränderung von L und U um denselben Betrag ein äquivalentes Problem ergibt.

5.2 Lösung von $T/L, U/ \cdot / \exists$

In diesem und den folgenden Abschnitten werden polynomiale Lösungsverfahren für Probleme der gleichmäßigen Baumzerlegung, bei denen die zulässigen Teilbäume durch untere und obere Gewichtsschranken charakterisiert sind, entwickelt. Das Kernproblem ist dabei die Lösung von Instanzen von TWSS mit polynomialem Aufwand. Wir beginnen in diesem Abschnitt mit der Problemstellung $T/L, U/ \cdot / \exists$ (vgl. 4.2.2).

5.2.1 Vorüberlegungen

In einer Instanz von $T/L, U/ \cdot / \exists$ sind ein eckengewichteter Baum $T = (V, E, w)$, $w : V \rightarrow \mathbb{Z}_{\geq 0}$ und Schranken $U, L \in \mathbb{Z}_{\geq 0}$ mit $U \geq L$ gegeben. Zu entscheiden ist, ob eine Zerlegung von T in zulässige Teilbäume existiert. Ein Teilbaum $T' \triangleleft T$ ist zulässig, wenn sein Gewicht zwischen L und U liegt.

Die Grundstruktur unseres Verfahrens entspricht Algorithmus 4.1 zur Lösung des allgemeinen Baumzerlegungsproblems. Dazu wählen wir eine beliebige Ecke in T als Wurzel aus. Da eine Zerlegung von T in zulässige Teilbäume gesucht ist, definieren wir die Kosten solcher Teilbäume als $cT' := 0$, somit ist

$$C(v) = \begin{cases} 0 & T_v \text{ kann in zulässige Teilbäume zerlegt werden,} \\ \infty & \text{sonst.} \end{cases}$$

Ziel ist nun die Berechnung von $C(v)$ gemäß Gleichung (4.3), wenn alle $C(u)$ für $u \succ v$ bekannt sind.

Beobachtung 1. Genau dann ist $C(v) = 0$, wenn es einen in v wurzelnden zulässigen Teilbaum T' von $T(v)$ gibt, für den $C(u) = 0$ für alle $u \in S(T')$ gilt.

Daraus folgt, daß Ecken $u \in T_v$ mit $C(u) = \infty$ aus T_v „herausdividiert“ werden können, wir also die Betrachtung auf den Baum

$$\tilde{T}_v := T_v / \{u \succ v : C(u) = \infty\}$$

beschränken können.⁷ Dabei ist folgende Regel beachten:

Wird eine Ecke u herausdividiert, so wird ihr Gewicht w_u zum Gewicht ihres Vorgängers addiert, das heißt $w_{\text{pre}(u)} := w_{\text{pre}(u)} + w_u$.

Im folgenden bezeichnen wir einen zulässigen Teilbaum $T' \triangleleft \tilde{T}_v$, der in v wurzelt, als *Lösungsteilbaum*. Außerdem ist für eine Ecke $u \in \tilde{T}_v$ \tilde{S}_u die Menge ihrer Söhne in \tilde{T}_v und \tilde{T}_u der Teilbaum von \tilde{T}_v , der u und alle Nachfolger enthält.

⁷Quotientenbildung wird in 4.1.1.1 eingeführt.

In unserem Verfahren werden die beiden Fälle $L \leq n(U - L)$ und $L > n(U - L)$, $n = |V|$, unterschiedlich behandelt. Der Grund ist, daß die Problemstellung in den beiden Fällen unterschiedliche Eigenschaften hat, die algorithmisch ausgenutzt werden.

5.2.2 Lösung im Fall $L \leq n(U - L)$

Die Frage, ob ein Lösungsteilbaum existiert, wird durch das Lösen eines TWSS auf \tilde{T}_v entschieden. Da alle w_v für Ecken in diesem Baum nach Voraussetzung nichtnegativ sind, können wir zur Aufwandsabschätzung die zweite Aussage in Satz 5.15 heranziehen. Wir erhalten eine Komplexität von $O(|\tilde{T}_v|^2 U / (U - L)) = O(n^2(1 + L/(U - L)))$. Da wir uns im Fall $L \leq n(U - L)$ befinden, ergibt dies einen Aufwand von $O(n^3)$ zur Bestimmung von $C(v)$.

5.2.3 Lösung im Fall $L > n(U - L)$

Wie in 5.2.2 ist unser Ziel, die Bestimmung von $C(v)$ auf das Lösen eines TWSS Problems zurückzuführen. Jedoch kann nicht mit einer unmittelbaren Anwendung von Satz 5.15 eine polynomiale Laufzeit garantiert werden. Wegen $L > n(U - L)$ hilft die zweite Aussage des Satzes nicht weiter. Auch die Summe der w_u , $u \in \tilde{T}_v$ kann zu groß sein, um über die erste Aussage von Satz 5.15 eine polynomiale Laufzeitabschätzung zu erhalten. Deswegen analysieren wir im folgenden die Problemeigenschaften im Fall $L > n(U - L)$ genauer. Dabei wird sich herausstellen, daß durch eine Modifikation des Problems Satz 5.17 anwendbar wird. Damit erhalten wir dann die gewünschte Abschätzung.

Für $u \in \tilde{T}_v$, $u \neq v$, gibt es nach Voraussetzung eine Zerlegung von \tilde{T}_u in zulässige Teilbäume, wir bezeichnen mit q_u deren Anzahl. Da wir notwendige Bedingungen für die Zerlegbarkeit von \tilde{T}_v ableiten wollen, gehen wir davon aus, daß auch \tilde{T}_v in eine Anzahl von q_v zulässigen Teilbäumen zerlegbar sei.

Beobachtung 2. Für alle Ecken $u \in \tilde{T}_v$ ist q_u eindeutig bestimmt.

Beweis. Angenommen, \tilde{T}_u ist sowohl in q' als auch in $q'' > q'$ zulässige Teilbäume zerlegbar. Dann ist $q''L \leq w\tilde{T}_u \leq q'U$. Daraus folgt $(q'' - q')L \leq q'(U - L)$ und daraus $L \leq q'(U - L)$. Nun ist $q' \leq n$, denn jeder Teilbaum muß mindestens eine Ecke enthalten. Somit erhalten wir einen Widerspruch zur Voraussetzung $L > n(U - L)$. \square

Wir setzen nun

$$\Delta_q(u) := q_u - \sum_{u' \in \tilde{S}_u} q_{u'}.$$

Damit erhalten wir eine Abschätzung des Gewichts von w_u .

Hilfssatz 5.18

Für $u \in \tilde{T}_v$ gilt

$$\Delta_q(u)L - n(U - L) \leq w_u \leq \Delta_q(u)L + n(U - L). \quad (5.5)$$

Beweis. Da für \tilde{T}_u eine zulässige Zerlegung in q_u Teilbäume existiert, ist $q_u L \leq w_{\tilde{T}_u} \leq q_u U$. Wir erweitern den linken und mittleren Term dieser Ungleichung zu

$$\Delta_q(u)L + \sum_{u' \in \tilde{S}_u} q_{u'} L \leq w_u + \sum_{u' \in \tilde{S}_u} w_{\tilde{T}_{u'}} \leq q_u U. \quad (5.6)$$

Auch für alle $u' \in \tilde{S}_u$ existiert eine zulässige Zerlegung von $\tilde{T}_{u'}$ in $q_{u'}$ Teilbäume; wir ziehen die Ungleichungen $q_{u'} L \leq w_{\tilde{T}_{u'}} \leq q_{u'} U$ von (5.6) ab und erhalten

$$\Delta_q(u)L - \sum_{u' \in \tilde{S}_u} q_{u'}(U - L) \leq w_u \leq q_u U - \sum_{u' \in \tilde{S}_u} q_{u'} L.$$

Subtraktion und Addition von $q_u L$ auf der rechten Seite ergibt

$$\Delta_q(u)L - \sum_{u' \in \tilde{S}_u} q_{u'}(U - L) \leq w_u \leq \Delta_q(u)L + q_u(U - L).$$

Da jeder Teilbaum in den Zerlegungen von \tilde{T}_u und $\tilde{T}_{u'}$ mindestens eine Ecke enthält, ist sowohl $q_u \leq n$ als auch $\sum_{u' \in \tilde{S}_u} q_{u'} \leq n$ und wir gewinnen die behauptete Abschätzung (5.5). \square

Eine erste Konsequenz aus Hilfssatz 5.18 ist $\Delta_q(u) \geq 0$, weil sonst wegen der Voraussetzung $L > n(U - L)$ die rechte Seite in (5.5) negativ wäre, ein Widerspruch zu $w_u \geq 0$. Wir benötigen nun eine weitere Formel.

Beobachtung 3. Sei T' Lösungsteilbaum. Dann ist $\sum_{u \in T'} \Delta_q(u) = 1$.

Beweis. Da q_v eindeutig bestimmt ist (Beobachtung 2), muß $q_v = 1 + \sum_{u \in \tilde{S}(T')} q_u$ sein. Dies ergibt durch „Aufblähen“ die Behauptung. \square

Sei $u \in \tilde{T}_v$. Da \tilde{T}_u in zulässige Teilbäume zerlegbar ist, gilt Beobachtung 3 sinngemäß auch für den in u wurzelnden Teilbaum in einer solchen Zerlegung. Mit $\Delta_q(u) \geq 0$ folgt daraus $\Delta_q(u) \in \{0, 1\}$:

Beobachtung 4. Für alle $u \in \tilde{T}_v$ gilt $\Delta_q(u) \in \{0, 1\}$.

Wir definieren nun zu jeder Ecke $u \in \tilde{T}_v$, $u \neq v$ zwei Gewichte durch $w_{1u} := w_u - \Delta_q(u)L$ und $w_{2u} := \Delta_q(u)$. Ferner setzen wir $w_{1v} := w_v$ und $w_{2v} := 0$. Falls ein Lösungsteilbaum T' existiert, tritt wegen der Eindeutigkeit von q_v und Beobachtung

3 einer von zwei Fällen ein: Entweder ist $\Delta_q(v) = 0$, dann ist $w_2T' = 1$ und $L - L \leq w_1T' \leq U - L$ oder es ist $\Delta_q(v) = 1$, dann ist $w_2T' = 0$ und $L \leq w_1T' \leq U$. Daher kann die Entscheidung, ob es einen Lösungsteilbaum gibt, auf die Beantwortung folgender Fragen zurückgeführt werden:

- Gibt es $T' \triangleleft \tilde{T}_v$ mit $v \in T'$, $0 \leq w_1T' \leq U - L$ und $w_2T' = 1$?
- Gibt es $T' \triangleleft \tilde{T}_v$ mit $v \in T'$, $L \leq w_1T' \leq U$ und $w_2T' = 0$?

Beides sind TWSS Probleme mit einer zusätzlichen Bedingung, wobei die Voraussetzungen von Satz 5.17 erfüllt sind.⁸ Nach Hilfssatz 5.18 gilt für $u \neq v$ $|w_{1u}| \leq n(U - L)$ und somit $\sum_{u \in \tilde{T}_v, u \neq v} |w_{1u}| \leq n|V(\tilde{T}_v)|(U - L)$. Aus Satz 5.17 erhalten wir eine Aufwandsabschätzung von $O(n|V(\tilde{T}_v)|^3) = O(n^4)$ für die Entscheidung, ob es einen solchen Lösungsteilbaum gibt. Damit ist jedoch $C(v)$ bestimmt.

5.2.4 Zusammenfassende Darstellung des Verfahrens

Zusammenfassend geben wir das Verfahren für $T/L, U/\cdot/\exists$ in Pseudo-Code Darstellung an (Algorithmus 5.4). In jeder Iteration der Hauptschleife (Zeilen 2-29) wird ein Lösungsteilbaum zur aktuellen Ecke v gesucht. In Zeile 5 ist ein TWSS Problem zu lösen, hierfür wird Algorithmus 5.3 verwendet, der Aufwand ist $O(n^3)$. In den Zeilen 14 und 16 sind TWSS Probleme mit einer Zusatz-Bedingung zu lösen, der Aufwand ist nach Satz 5.17 $O(n^4)$, wie oben gezeigt. In Zeile 21 wird ein gefundener Lösungsteilbaum zu einem Teilbaum von T expandiert, indem die in \tilde{T} ausdividierten Ecken wieder eingefügt werden.

Algorithmus 5.4 endet mit der Ausgabe „unzulässig“, oder es können durch Rückwärtsrechnung von der Wurzel zu den Blättern (siehe Algorithmus 4.1) unter den $T^*(v)$, $v \in \tilde{T}$, Teilbäume von T bestimmt werden, die eine zulässige Zerlegung bilden.

Wir fassen das Ergebnis von Abschnitt 5.2 zusammen.

Satz 5.19

Der Aufwand zur Lösung des Problems $T/L, U/\cdot/\exists$ mit Algorithmus 5.4 ist im Fall $L \leq n(U - L)$ $O(n^4)$ und im Fall $L > n(U - L)$ $O(n^5)$, $n = |V|$.

5.3 Lösung von $T/L, U/p/\exists$

In der Problemstellung $T/L, U/p/\exists$ sind ein eckengewichteter Baum $T = (V, E, w)$, $w : V \rightarrow \mathbb{Z}_{\geq 0}$, Schranken $L, U \in \mathbb{Z}$ mit $U \geq L \geq 0$ und ein $p \in \{1, \dots, |V|\}$ gegeben; gefragt ist, ob eine Zerlegung von T in p zulässige Teilbäume, d.h. solche mit Gewicht zwischen L und U , existiert.

⁸Wegen der Eindeutigkeit von q_v kann nur eines der beiden Probleme lösbar sein.

Algorithmus 5.4 Rechne vorwärts $T/L, U/\cdot/\exists$

Input: $T = (V, E, w \rightarrow \mathbb{Z}_{\geq 0}), v_0, L, U \in \mathbb{Z}_{\geq 0}, L \leq U$
Output: „unzulässig“ oder \tilde{T} und zulässige Teilbäume $T^*(v) \triangleleft T, v \in \tilde{T}$

```

1:  $\tilde{T} := T; C(v) := \infty, v \in V$ 
2: for all  $v \in V$  von den Blättern zur Wurzel in  $T$  do
3:    $T' := \emptyset$ 
4:   if  $L \leq n(U - L)$  then
5:     if TWSS für  $\tilde{T}_v, v, L, U$  zulässig then
6:        $T'$  sei Lösung
7:     end if
8:   else /*  $L > n(U - L)$  */
9:      $q_u :=$  Anzahl Teilbäume in zulässiger Zerlegung von  $\tilde{T}_u, u \in \tilde{T}_v, u \neq v$ 
10:     $\Delta_q(u) := q_u - \sum_{u' \in \tilde{s}_u} q_{u'}, u \in \tilde{T}_v, u \neq v$ 
11:     $w_{1u} := w_u - \Delta_q(u)L, u \in \tilde{T}_v, u \neq v$ 
12:     $w_{2u} := \Delta_q(u), u \in \tilde{T}_v, u \neq v$ 
13:     $w_{1v} := w_v; w_{2v} := 0$ 
14:    if  $\exists T' \triangleleft \tilde{T}_v$  mit  $v \in T', 0 \leq w_1 T' \leq U - L, w_2 T' = 1$  then
15:       $T'$  sei Lösung
16:    else if  $\exists T' \triangleleft \tilde{T}_v$  mit  $v \in T', L \leq w_1 T' \leq U, w_2 T' = 0$  then
17:       $T'$  sei Lösung
18:    end if
19:  end if
20:  if  $T' \neq \emptyset$  then
21:     $C(v) := 0; T^*(v) :=$  expandiere( $T'$ )
22:  else
23:    if  $v = v_0$  then
24:      gib aus „unzulässig“
25:    else
26:       $\tilde{T} := \tilde{T}/v$ 
27:    end if
28:  end if
29: end for

```

Im Gegensatz zur Problemstellung des letzten Abschnittes ist $T/L, U/p/\exists$ ein p -BZP, wir greifen deshalb für unser Lösungsverfahren auf die Rekursion (4.4) zurück. Dazu wird eine beliebige Ecke v_0 in V als Wurzel ausgewählt und die Kosten zulässiger Teilbäume T' werden auf $cT' := 0$ gesetzt. Es ist für $v \in V$ und $1 \leq q \leq p$

$$C(v, q) = \begin{cases} 0 & T_v \text{ kann in } q \text{ zulässige Teilbäume zerlegt werden,} \\ \infty & \text{sonst.} \end{cases}$$

Zu berechnen sei $C(v, q)$, bereits bestimmt seien die $C(u, q')$ für alle $u \in T_v$, $u \neq v$ und $1 \leq q' \leq p$. Wir setzen $Q_u := \{q' : C(u, q') = 0\}$. Ecken u mit $Q_u = \emptyset$ sind nicht Kandidaten für Wurzeln von Teilbäumen in zulässigen Zerlegungen und können daher aus T_v herausdividiert werden:

$$\tilde{T}_v := T_v / \{u \in T_v : Q_u = \emptyset\}.$$

Das Gewicht dieser Ecken wird dabei zum Gewicht des jeweiligen Vorgängers addiert. Gesucht ist ein *Lösungsteilbaum* $T' \triangleleft \tilde{T}_v$ zu v und q , also ein Baum mit $v \in T'$, $L \leq wT' \leq U$, für den es zu allen $u \in \tilde{S}(T')$ ein $q_u \in Q_u$ gibt mit $\sum_{u \in \tilde{S}(T')} q_u = q - 1$. Wie im vorangehenden Abschnitt unterscheiden wir für das Lösungsverfahren zwei Fälle. Der Fall $L > p(U - L)$, den wir zuerst betrachten, wird völlig analog behandelt, wie der Fall $L > n(U - L)$ für das $T/L, U/\cdot/\exists$ Problem. Dagegen sind für den Fall $L \leq p(U - L)$ zusätzliche Überlegungen nötig.

5.3.1 Lösung im Fall $L > p(U - L)$

Zunächst stellen wir fest, daß $|Q_u| = 1$ ist für alle $u \in \tilde{T}_v$, $u \neq v$ und daß $|Q_v| \leq 1$ ist. Der Beweis hierfür ist analog zu dem der Beobachtung 2 in 5.2.3. Da uns interessiert, welche Eigenschaften im Fall der Zerlegbarkeit von \tilde{T}_v gegeben sind, nehmen wir $|Q_v| = 1$ an. Somit sind für alle $u \in \tilde{T}_v$ eindeutig bestimmte q_u ($Q_u = \{q_u\}$) gegeben. Es sei

$$\Delta_q(u) := q_u - \sum_{u' \in \tilde{S}_u} q_{u'},$$

wobei \tilde{S}_u die Söhne von u in \tilde{T}_v sind.

Der Beweis von Hilfssatz 5.18 läßt sich ebenfalls übertragen und ergibt hier, daß für alle u in \tilde{T}_v gilt

$$\Delta_q(u)L - p(U - L) \leq w_u \leq \Delta_q(u)L + p(U - L). \quad (5.7)$$

Wie in Abschnitt 5.2.3 schließen wir, daß $\Delta_q(u) \in \{0, 1\}$ ist. Ferner gilt für jeden Lösungsteilbaum T' die Gleichung $\sum_{u \in T'} \Delta_q(u) = 1$. Es folgt, daß entweder $q_v =$

$\sum_{u \in \tilde{S}_v} q_u$ oder $q_v = 1 + \sum_{u \in \tilde{S}_v} q_u$ ist, so daß nur für einen dieser beiden Werte von $q = q_v$ überhaupt $C(v, q) = 0$ sein kann. Die Iteration über q kann in der Berechnung von $C(v, q)$ entfallen.

Wiederum wird die Existenz eines Lösungsteilbaumes mit Satz 5.17 geprüft. Wir setzen wie oben $w_{1u} := w_u - \Delta_q(u)L$ und $w_{2u} := \Delta_q(u)$ für $u \in \tilde{T}_v$, $u \neq v$ und weiter $w_{1v} := w_v$, $w_{2v} := 0$. Folgende Fragen sind nun zu beantworten:

- Gibt es $T' \triangleleft \tilde{T}_v$ mit $v \in T'$, $0 \leq w_1 T' \leq U - L$ und $w_2 T' = 1$?
- Gibt es $T' \triangleleft \tilde{T}_v$ mit $v \in T'$, $L \leq w_1 T' \leq U$ und $w_2 T' = 0$?

Ist die Antwort auf die erste Frage positiv, so ist $C(v, q_v) = 0$ für $q_v = \sum_{u \in \tilde{S}_v} q_u$ und $C(v, q) = \infty$ für alle anderen q . Entsprechendes gilt bei der zweiten Frage für $q_v = 1 + \sum_{u \in \tilde{S}_v} q_u$.

Der Aufwand für die Beantwortung der beiden Fragen ist $O(pn^3)$, $n = |V|$. Dies ergibt sich aus Satz 5.17 und (5.7), da diese Ungleichung $|w_{1u}| \leq p(U - L)$ impliziert.

5.3.2 Lösung im Fall $L \leq p(U - L)$

Im Gegensatz zum Fall $L > p(U - L)$ kann hier durchaus $|Q_u| > 1$ für $u \in \tilde{T}_v$ vorkommen. Um $C(v, q) = 0$ festzustellen, müssen daher neben einem Teilbaum $T' \triangleleft \tilde{T}_v$, $v \in T'$ mit $L \leq w T' \leq U$ auch Zahlen $q_u \in Q_u$ für $u \in \tilde{S}(T')$ bestimmt werden, deren Summe $q - 1$ ergibt. Dadurch wird das TWSS Problem für T' mit einem Auswahlproblem gekoppelt. Durch eine geeignete Konstruktion läßt sich dieses zusammengesetzte Entscheidungsproblem jedoch auf ein TWSS auf einem „großen“ Wurzelbaum \dot{T} mit zweifachen Eckengewichten abbilden. Im folgenden geben wir diese Konstruktion an und weisen dann mit Hilfe von Satz 5.16 einen polynomialen Aufwand für die Bestimmung von $C(v, q)$ nach.

Der zu konstruierende Baum $\dot{T} = (\dot{V}, \dot{E})$ entspricht in seiner Struktur \tilde{T}_v , wobei jede Ecke $u \in \tilde{T}_v$, $u \neq v$ durch eine Kette von $|Q_u|$ Ecken ersetzt ist (siehe Abbildung 5.1). Genauer sei

$$\dot{V} := \{v\} \cup \{\dot{u}_l : u \in \tilde{T}_v, u \neq v, l \in \{1, \dots, |Q_u|\}\}. \quad (5.8)$$

Die Kantenmenge \dot{E} und die Orientierung auf \dot{T} werden festgelegt, indem wir für jede Ecke in \dot{V} den Vorgänger angeben: v ist die Wurzel von \dot{T} und für \dot{u}_l setzen wir

$$\text{pre}(\dot{u}_l) = \begin{cases} \dot{u}_{l-1} & l > 1, \\ [\text{pre}(u)]_{|Q_u|} & l = 1. \end{cases} \quad (5.9)$$

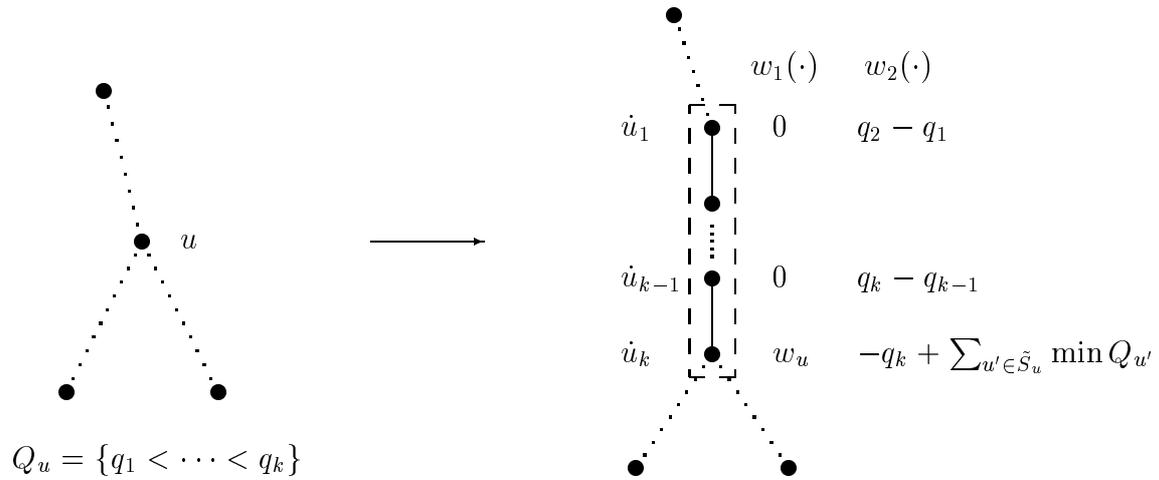


Abbildung 5.1: Konstruktion von \tilde{T} am Beispiel der Ecke u

Jeder Ecke in \tilde{V} werden zwei Gewichte zugewiesen. Das Gewicht w_1 ist bestimmt durch

$$w_1(v) = w_v \quad \text{und} \quad w_1(\hat{u}_l) = \begin{cases} 0 & 1 \leq l < |Q_u|, \\ w_u & l = |Q_u|. \end{cases} \quad (5.10)$$

Weiter ist

$$w_2(v) = 1 + \sum_{u \in \tilde{S}_v} \min Q_u \quad (5.11)$$

und zur Definition von $w_2(\hat{u}_l)$ seien $q_1 < q_2 < \dots < q_k$ die Elemente von Q_u . Wir setzen

$$w_2(\hat{u}_l) = \begin{cases} q_{l+1} - q_l & 1 \leq l < k = |Q_u|, \\ -q_k + \sum_{u' \in \tilde{S}_u} \min Q_{u'} & l = k. \end{cases} \quad (5.12)$$

Beobachtung 1. Die Lösungsteilbäume zu v, q von \tilde{T}_v entsprechen eineindeutig den Teilbäumen $\tilde{T}' \triangleleft \tilde{T}$ mit $v \in \tilde{T}'$, $L \leq w_1 \tilde{T}' \leq U$ und $w_2 \tilde{T}' = q$.

Beweis. Die Aussage folgt direkt aus der Konstruktion von \tilde{T} . Der einem Lösungsteilbaum T' entsprechende Teilbaum \tilde{T}' enthält genau die folgenden Ecken:

- die Ecke v
- \hat{u}_l , $1 \leq l \leq |Q_u|$, für jedes $u \in T'$, $u \neq v$
- $\hat{u}_1, \dots, \hat{u}_{l-1}$ für jedes $u \in \tilde{S}(T')$, wobei das l -kleinste Element von Q_u als q_u ausgewählt ist.⁹

⁹Man beachte, daß im Fall $l = 1$ hier keine Ecken zu \tilde{T}' hinzukommen.

Es ist leicht nachzuprüfen, daß $w_1\dot{T}' = wT'$ und $w_2\dot{T}' = 1 + \sum_{u \in \tilde{S}(T')} q_u = q$ gilt. \square

Das nächste Ergebnis zielt auf die Anwendung von Satz 5.16.

Beobachtung 2. Es ist $|\dot{V}| \leq p|V|$. Höchstens $|V|$ der $w_1(\dot{u})$, $\dot{u} \in \dot{V}$, sind positiv, alle anderen sind Null. Ferner ist $\sum_{\dot{u} \in \dot{V}} |w_2(\dot{u})| = O(p|V|)$.

Beweis. Die ersten beiden Aussagen folgen unmittelbar aus der Konstruktion von \dot{T} . Die dritte Abschätzung erhält man wie folgt. Zunächst ist für $u \in \tilde{T}_v$ mit $k = |Q_u|$

$$\sum_{l=1}^k |w_2(\dot{u}_l)| \leq q_k - q_1 + |\tilde{S}_u|p \leq (|\tilde{S}_u| + 1)p,$$

daraus folgt

$$\begin{aligned} \sum_{\dot{u} \in \dot{V}} |w_2(\dot{u})| &= w_2(v) + \sum_{u \in \tilde{T}_v, u \neq v} \sum_{l=1, \dots, |Q_u|} |w_2(\dot{u}_l)| \\ &\leq 1 + |\tilde{S}_v|p + \sum_{u \in \tilde{T}_v, u \neq v} (|\tilde{S}_u| + 1)p \\ &\leq 2|V|p = O(p|V|). \end{aligned}$$

\square

Aus Satz 5.16 erhalten wir nun, daß für alle $1 \leq q \leq p$ mit Aufwand

$$O(p|V| (|V|U/(U-L)) p|V|)$$

entschieden werden kann, ob ein Lösungsteilbaum zu v und q existiert. Da nach Voraussetzung $U/(U-L) = 1 + L/(U-L) \leq 1 + p$ ist, ergibt dies einen Aufwand von

$$O(p^3 n^3), \quad n = |V|.$$

5.3.3 Zusammenfassende Darstellung des Verfahrens

Eine Pseudo-Code Darstellung des Verfahrens für $T/L, U/p/\exists$ ist Algorithmus 5.5. Seine Grundstruktur entspricht der Vorwärtsrechnung in Algorithmus 4.2.

In jeder Iteration der Hauptschleife (Zeilen 2 bis 36) wird für eine feste Ecke v das Vorhandensein von Lösungsteilbäumen zu allen $q \in \{1, \dots, p\}$ geprüft. Dies geschieht in den Zeilen 5 (Fall $L \leq p(U-L)$ bzw. 19 und 22 (Fall $L > p(U-L)$). Dazu werden jeweils TWSS Probleme mit zusätzlicher Nebenbedingung gemäß den Sätzen 5.16 bzw. 5.17 gelöst.

Der Operator **expandiere** (Zeilen 9, 27) fügt Ecken, die aus \tilde{T} herausdividiert wurden, in T' ein und liefert somit einen Teilbaum von T . Die Formel in Zeile 10 zur

Algorithmus 5.5 Rechne vorwärts $T/L, U/p/\exists$ **Input:** $T = (V, E, w \rightarrow \mathbb{Z}_{\geq 0})$, $v_0, p \in \{1, \dots, |V|\}$, $L, U \in \mathbb{Z}_{\geq 0}$, $L \leq U$ **Output:** „unzulässig“ oder: \tilde{T} und $Q_v \neq \emptyset$ für $v \in \tilde{T}$ mit $p \in Q_{v_0}$; weiter $T^*(v, q) \triangleleft T$ für $v \in \tilde{T}$, $q \in Q_v$; schließlich $q_u^*(v, q)$ für $v \in \tilde{T}$, $q \in Q_v$, $u \in S(T^*(v, q))$

- 1: $\tilde{T} := T$; $C(v, q) := \infty$, $v \in V$, $1 \leq q \leq p$; $Q_v := \emptyset$, $v \in V$
- 2: **for all** $v \in V$ von den Blättern zur Wurzel in T **do**
- 3: **if** $L \leq p(U - L)$ **then**
- 4: ausgehend von \tilde{T}_v konstruiere $\dot{T}(\dot{V}, \dot{E})$, w_1, w_2 gemäß (5.8) bis (5.12)
- 5: **for all** $q \in \{1, \dots, p\}$ mit: $\exists \dot{T}' \triangleleft \dot{T}$ mit $v \in \dot{T}'$, $L \leq w_1 \dot{T}' \leq U$, $w_2 \dot{T}' = q$ **do**
- 6: $C(v, q) := 0$; $Q_v := Q_v \cup \{q\}$
- 7: $V' := \{v\} \cup \{u \in \tilde{T}_v : \dot{u}_l \in \dot{T}' \text{ für } l = |Q_u|\}$
- 8: $T' :=$ Teilbaum von \tilde{T}_v mit der Eckenmenge V'
- 9: $T^*(v, q) :=$ expandiere(T')
- 10: $q_u^*(v, q) := \min Q_u + \sum_{\dot{u}_l \in \dot{T}'} w_2(\dot{u}_l)$, $u \in \tilde{S}(T')$
- 11: **end for**
- 12: **else** /* $L > p(U - L)$ */
- 13: **for all** $u \in \tilde{T}_v$, $u \neq v$ aufsteigend in \tilde{T}_v **do**
- 14: $q_u := Q_u$ /* es gilt $|Q_u| = 1$ */
- 15: $\Delta_q(u) := q_u - \sum_{u' \in \tilde{S}_u} q_{u'}$,
- 16: $w_{1u} := w_u - \Delta_q(u)L$; $w_{2u} := \Delta_q(u)$
- 17: **end for**
- 18: $w_{1v} := w_v$; $w_{2v} := 0$
- 19: **if** $\exists T' \triangleleft \tilde{T}_v$ mit $v \in T'$, $0 \leq w_1 T' \leq U - L$, $w_2 T' = 1$ **then**
- 20: T' sei Lösung
- 21: $q_v := \sum_{u \in \tilde{S}_v} q_u$; $Q_v := \{q_v\}$
- 22: **else if** $\exists T' \triangleleft \tilde{T}_v$ mit $v \in T'$, $L \leq w_1 T' \leq U$, $w_2 T' = 0$ **then**
- 23: T' sei Lösung
- 24: $q_v := 1 + \sum_{u \in \tilde{S}_v} q_u$; $Q_v := \{q_v\}$
- 25: **end if**
- 26: **if** $Q_v \neq \emptyset$ **then**
- 27: $C(v, q_v) := 0$; $T^*(v, q_v) :=$ expandiere(T')
- 28: $q_u^*(v, q_v) := q_u$, $u \in \tilde{S}(T')$
- 29: **end if**
- 30: **end if**
- 31: **if** $v = v_0$ und ($Q_v = \emptyset$ oder $p \notin Q_v$) **then**
- 32: gib aus „unzulässig“
- 33: **else if** $Q_v = \emptyset$ **then**
- 34: $\tilde{T} := \tilde{T}/v$
- 35: **end if**
- 36: **end for**

Berechnung der $q_u^*(v, q)$ ist durch die Konstruktion von \dot{T} begründet (siehe Abbildung 5.1): Für jede Ecke $u \in \tilde{S}(T')$ ist nach Wahl von V' (Zeile 7) die (eventuell leere) Kette mit den Ecken $\dot{u}_1, \dots, \dot{u}_l$ in \dot{T}' enthalten, wobei $0 \leq l < k = |Q_u|$ gilt. Somit hat die Summe in Zeile 10 den Wert

$$q_1 + (q_2 - q_1) + \dots + (q_{l+1} - q_l) = q_{l+1},$$

ist also Element von Q_u . Es ist leicht nachzuprüfen, daß

$$1 + \sum_{u \in \tilde{S}(T')} q_u^*(v, q) = \sum_{\dot{u} \in \dot{T}} w_2(\dot{u}) = q$$

gilt, indem die linke Seite geeignet „aufgebläht“ wird.

Algorithmus 5.5 endet mit der Ausgabe „unzulässig“ oder es kann unter den berechneten $T^*(v, q)$ und $q_u^*(v, q)$ durch Rückwärtsrechnung gemäß Algorithmus 4.2 eine zulässige Zerlegung von T in p Teilbäume bestimmt werden.

Das Ergebnis dieses Abschnitts faßt der folgende Satz zusammen:

Satz 5.20

Der Aufwand zur Lösung des Problems $T/L, U/p/\exists$ mit Algorithmus 5.5 ist $O(p^3 n^4)$ im Fall $L \leq p(U - L)$ und $O(pn^4)$ im Fall $L > p(U - L)$, $n = |V|$.

5.4 Lösung von $T/\text{Zent}, U_z - L_z = \text{const}/\cdot/\exists$

Eine Instanz des Problems $T/\text{Zent}, U_z - L_z = \text{const}/\cdot/\exists$ ist folgendermaßen definiert (vgl. 4.2.2): Gegeben ist ein eckengewichteter Baum $T = (V, E, w)$, $w : V \rightarrow \mathbb{Z}_{\geq 0}$, eine Menge Zentren $Z \subset V$ und Schranken $L_z, U_z \in \mathbb{Z}_{\geq 0}$, wobei $U_z - L_z \geq 0$ nicht von z abhängt. Ein Teilbaum $T' \triangleleft T$ ist zulässig, wenn er genau ein Zentrum $z \in Z$ enthält und sein Gewicht zwischen den zu diesem Zentrum gehörenden Schranken L_z und U_z liegt. Es ist zu entscheiden, ob eine Zerlegung von T in zulässige Teilbäume existiert.

Um Algorithmus 4.1 als Grundstruktur unseres Verfahrens verwenden zu können, setzen wir wie in den vorangegangenen Abschnitten $cT' := 0$ für alle zulässigen T' , folglich ist

$$C(v) = \begin{cases} 0 & T_v \text{ kann in zulässige Teilbäume zerlegt werden,} \\ \infty & \text{sonst.} \end{cases}$$

Ziel ist nun die Berechnung von $C(v)$ gemäß Gleichung (4.3), wenn alle $C(u)$ für $u \succ v$ bekannt sind. Ecken u mit $C(u) = \infty$ werden aus T_v herausdividiert,

$$\tilde{T}_v := T_v / \{u \succ v : C(u) = \infty\}$$

da sie nicht als Wurzeln von Teilbäumen in zulässigen Zerlegungen in Frage kommen. Dabei wird nach folgender Regel verfahren:

Beim Herausdividieren einer Ecke u wird ihr Gewicht dem Gewicht ihres Vorgängers $\text{pre}(u)$ hinzuaddiert. Ferner wird $\text{pre}(u)$ Zentrum, wenn u Zentrum ist.

Die Möglichkeit, daß $\text{pre}(u)$ selbst bereits Zentrum sein kann, führt zur folgenden Überlegung:

Beobachtung 1. Wird bei der Berechnung von \tilde{T}_v eine Kante kontrahiert, für die beide Enden Zentren sind, so ist $C(v) = \infty$.

Wir gehen daher im folgenden davon aus, daß der in Beobachtung 1 genannte Fall nicht eingetreten ist.

Beobachtung 2. Alle Blätter von \tilde{T}_v sind Zentren.

Beweis. Ist u Blatt und nicht in Z , so enthält T_u kein Zentrum. Folglich ist $C(u) = \infty$ und u nicht in \tilde{T}_v enthalten. \square

5.4.1 Der Fall $v \in Z$

Zuerst betrachten wir den Fall, daß bei der Bestimmung von $C(v)$ die Ecke v ein Zentrum ist. Dann ist ein in v wurzelnder Teilbaum T' von \tilde{T}_v gesucht, der kein weiteres Zentrum enthält. Wir bezeichnen daher die Ecken aus $R := \{u \in \tilde{T}_v : P[u, v) \cap Z = \emptyset\}$ als *relevante* Ecken. (Man beachte, daß auch $v \in R$ ist.)

Beobachtung 3. Sei $v \in Z$. Ist $T' \triangleleft \tilde{T}_v$ zulässig und $v \in T'$, so enthält T' nur relevante Ecken.

Mit $T(R)$ bezeichnen wir den Teilbaum von \tilde{T}_v , dessen Eckenmenge R ist.

Beobachtung 4. Sei $v \in Z$. Es ist $C(v) = 0$ genau dann, wenn es einen in v wurzelnden Teilbaum T' von $T(R)$ gibt mit $L_v \leq wT' \leq U_v$.

Aus Beobachtung 4 folgt, daß wir zur Bestimmung von $C(v)$ ein TWSS zu lösen haben. Das nächste Ergebnis ist eine Abschätzung der Größe der darin auftretenden Eckengewichte.

Hilfssatz 5.21

Ist $u \in R$, $u \neq v$, so ist $w_u \leq (U_v - L_v)|Z|$.

Beweis. Für $u' \in \tilde{T}_v$ sei mit $Z_{u'} := Z \cap V(\tilde{T}_{u'})$ die Menge der Zentren in $\tilde{T}_{u'}$ bezeichnet.

Nach Voraussetzung ist u kein Zentrum. Daher kann Z_u disjunkt zerlegt werden in

$$Z_u = \bigcup_{u' \in \tilde{S}_u} Z_{u'}. \quad (5.13)$$

(Wegen Beobachtung 2 sind die Mengen auf der rechten Seite nicht leer.) Weiter existiert eine zulässige Zerlegung von \tilde{T}_u , denn es ist $u \in \tilde{T}_v$, also $C(u) = 0$. Daraus folgt

$$w\tilde{T}_u \leq \sum_{z \in Z_u} U_z.$$

Linke und rechte Seite dieser Ungleichung können wir „aufblähen“ zu (man beachte (5.13))

$$w_u + \sum_{u' \in \tilde{S}_u} w\tilde{T}_{u'} \leq \sum_{u' \in \tilde{S}_u} \sum_{z \in Z_{u'}} U_z. \quad (5.14)$$

Andererseits existiert auch zu $u' \in \tilde{S}_u$ eine zulässige Zerlegung von $\tilde{T}_{u'}$, somit ist

$$w\tilde{T}_{u'} \geq \sum_{z \in Z_{u'}} L_z.$$

Wird diese Ungleichung für alle $u' \in \tilde{S}_u$ von (5.14) subtrahiert, erhalten wir

$$w_u \leq \sum_{u' \in \tilde{S}_u} \sum_{z \in Z_{u'}} U_z - L_z,$$

und die rechte Seite ist nach (5.13) gleich $\sum_{u \in Z_u} U_z - L_z$.

Nun hängt in unserer Problemstellung $U_z - L_z$ nicht von z ab, und ist somit gleich $U_v - L_v$. Wir erhalten die Behauptung $w_u \leq |Z|(U_v - L_v)$. \square

Da nach Beobachtung 4 ein TWSS auf $T(R)$ zu lösen ist, und nach Hilfssatz 5.21 gilt $\sum_{u \in R - \{v\}} w_u / (U - L) \leq |R||Z|$, ergibt sich mit Satz 5.15 ein Aufwand $O(|R|^2|R||Z|) = O(|R|^3|Z|)$ zur Bestimmung von $C(v)$.

5.4.2 Der Fall $v \notin Z$

Die Vorgehensweise im Fall $v \notin Z$ wird auf die bisherige Untersuchung zurückgeführt. Nach Beobachtung 1 ist ein zulässiger Teilbaum T' von \tilde{T}_v gesucht, der in v wurzelt. T' muß genau ein Zentrum enthalten, und es kommen hierfür nur solche Zentren $z \in \tilde{T}_v$ in Frage, für die $P[z, v]$ kein weiteres Zentrum enthält. Setzen wir daher

$$Z' := \{z \succ v : z \in Z, P(z, v) \cap Z = \emptyset\},$$

so muß T' genau einen der Pfade $P[z, v]$, $z \in Z'$, enthalten.

Das Problem der Bestimmung von $C(v)$ läßt sich also lösen, indem wir für jedes $z \in Z'$ den Baum $\tilde{T}_v/P[z, v)$, in dem nun $v \in Z$ ist, mit den oben hergeleiteten Methoden untersuchen. Wird für einen dieser Bäume ein zulässiger, in v wurzelnder Teilbaum gefunden, so ist $C(v) = 0$, andernfalls $C(v) = \infty$.

5.4.3 Zusammenfassende Darstellung des Verfahrens

Wir stellen das Verfahren in Pseudo-Code dar (Algorithmus 5.6). Da verschiedene Bäume eine Rolle spielen (T, \tilde{T}, \hat{T}), kennzeichnen wir für Pfade und Zentren-Mengen durch Akzente, auf welchen Baum sie sich beziehen ($\tilde{P}[u, v], \hat{Z}$, usw.). Im Verlauf des Algorithmus werden Kontraktionen ausgeführt, die den Baum verändern, deshalb wird mit einer Kopie \tilde{T} gearbeitet (Zeile 1). Die Bedingung in Zeile 3 prüft, ob es unterhalb von v Zentren gibt, nur dann kann $C(v) = 0$ sein. In den Zeilen 4 bis 8 werden die möglichen Zentren für den gesuchten zulässigen Teilbaum bestimmt. Durch Rückführung auf das TWSS wird in den Zeilen 9 bis 18 entschieden, ob es einen zulässigen Teilbaum gibt, der in v wurzelt. Sobald ein solcher gefunden ist (Zeile 14), wird $C(v) = 0$ gesetzt und der Teilbaum festgehalten (Zeile 15), wobei die vorgenommenen Kontraktionen rückgängig gemacht werden (Operator **expandiere**), um einen Teilbaum von T zu erhalten. In den Zeilen 21 bis 29 wird geprüft, ob Unzulässigkeit bereits erkennbar ist, andernfalls wird, wenn $C(v) = \infty$ ist, in Zeile 30 die Ecke v aus dem Baum herausdividiert.

Nach Abschluß von Algorithmus 5.6 kann mittels trivialer Rückwärtsrechnung wie sie in Algorithmus 4.1 angeführt wird, eine zulässige Zerlegung von T als Teilmenge von $\{T^*(v) : v \in \tilde{T}\}$ bestimmt werden.

Da der Aufwand zur Lösung eines TWSS in Zeile 14 wie oben bemerkt $O(|Z||R|^3)$ ist, und Zeile 14 innerhalb zwei verschachtelten „for all“ Schleifen steht, erhalten wir folgende Abschätzung für den Gesamtaufwand:

Satz 5.22

Der Aufwand zur Lösung des Problems $T/\text{Zent}, U_z - L_z = \text{const}/\cdot/\exists$ mit Algorithmus 5.6 ist $O(p^2n^4)$, $n = |V|$, $p = |Z|$.

5.5 Lösung von $\|\cdot\|_\infty$ -Problemen

Die Minimierung der maximalen Abweichung der Teilbaumgewichte von einem vorgegebenen Zielwert μ kann unmittelbar auf die Verfahren der vorangehenden Abschnitte zurückgeführt werden. Es sei zum Beispiel eine Instanz des Problems $T/\cdot/\cdot/\|\cdot\|_\infty$ vorgelegt, also ein eckengewichteter Baum $T = (V, E, w \rightarrow \mathbb{Z}_{\geq 0})$ und μ . Der optimale Zielfunktionswert ζ^* wird durch Binärsuche bestimmt: Ist für ein ζ das Problem

Algorithmus 5.6 Rechne vorwärts $T/Zent, U_z - L_z = const / \cdot / \exists$

Input: $T = (V, E, w \rightarrow \mathbb{Z}_{\geq 0}), v_0, Z, L_z, U_z \in \mathbb{Z}_{\geq 0}, U_z - L_z$ unabhängig von $z \in Z$

Output: „unzulässig“ oder \tilde{T} und zulässige Teilbäume $T^*(v) \triangleleft T, v \in \tilde{T}$

```

1:  $\tilde{T} := T; \tilde{Z} := Z; C(v) := \infty, v \in V$ 
2: for all  $v \in V$  von den Blättern zur Wurzel in  $T$  do
3:   if  $V(\tilde{T}_v) \cap \tilde{Z} \neq \emptyset$  then
4:     if  $v \in \tilde{Z}$  then
5:        $Z' := \{v\}$ 
6:     else
7:        $Z' := \{z \in \tilde{Z} : z \succ v, \tilde{P}(z, v) \cap \tilde{Z} = \emptyset\}$ 
8:     end if
9:     for all  $z \in Z'$  do
10:      if  $C(v) = \infty$  then
11:         $\dot{T} := \tilde{T} / \tilde{P}[z, v]; \dot{Z} := \tilde{Z} \cup \{v\}$  /*  $\dot{T} = \tilde{T}$ , falls  $z = v$  */
12:         $R := \{u \in V(\dot{T}) : u \succ v, \dot{P}[u, v] \cap \dot{Z} = \emptyset\}$ 
13:         $T(R) :=$  Teilbaum von  $\dot{T}$  mit Eckenmenge  $R$ 
14:        if  $\exists T' \triangleleft T(R), v \in T', L_z \leq wT' \leq U_z$  then
15:           $C(v) := 0; T^*(v) := \text{expandiere}(T')$ 
16:        end if
17:      end if
18:    end for
19:  end if
20:  if  $C(v) = \infty$  then
21:    if  $v = v_0$  then
22:      gib aus „unzulässig“; STOP
23:    else if  $v \in \tilde{Z}$  then
24:      if  $\text{pre}(v) \in \tilde{Z}$  then
25:        gib aus „unzulässig“; STOP
26:      else
27:         $\tilde{Z} := \tilde{Z} \cup \{\text{pre}(v)\}$ 
28:      end if
29:    end if
30:     $\tilde{T} := \tilde{T} / v$ 
31:  end if
32: end for

```

$T/L, U/\cdot/\exists$ mit $L := \mu - \zeta$, $U := \mu + \zeta$ lösbar, so ist $\zeta^* \leq \zeta$, andernfalls gilt $\zeta^* > \zeta$.

Da wir $0 \leq \mu \leq wT$ voraussetzen können (andernfalls ist die vorliegende Instanz von $T/\cdot/\cdot/\|\cdot\|_\infty$ trivial), kann der Suchbereich für ζ^* zunächst auf das Intervall $[0, wT]$ beschränkt werden. Wird für ζ jeweils die Mitte des verbliebenen Intervalls gewählt, halbiert sich mit jedem getesteten Problem $T/L, U/\cdot/\exists$ die Länge dieses Intervalls und ζ^* ist nach $O(\log wT)$ solcher Tests bestimmt. Somit folgt aus Satz 5.19 unmittelbar

Satz 5.23

Der Aufwand zur Lösung von $T/\cdot/\cdot/\|\cdot\|_\infty$ ist $O(n^5 \log wT)$, $n = |V|$.

Analog ergibt sich für das Problem $T/\cdot/p/\|\cdot\|_\infty$ eine polynomiale Komplexität aus Satz 5.20. Auch hier kommt ein Faktor $\log wT$ hinzu, denn wieder können wir annehmen, daß $0 \leq \mu \leq wT$ ist. (Für alle $\mu \leq 0$ hat $T/\cdot/p/\|\cdot\|_\infty$ dieselbe Optimallösung, genauso für alle $\mu \geq wT$.)

Satz 5.24

Der Aufwand zur Lösung von $T/\cdot/p/\|\cdot\|_\infty$ ist $O(p^3 n^4 \log wT)$, $n = |V|$.

Zuletzt wird Problem $T/Zent/\cdot/\|\cdot\|_\infty$ durch die folgenden Überlegungen auf das Lösen von $O(\log wT)$ Instanzen vom Typ $T/Zent, U_z - L_z = const/\cdot/\exists$ zurückgeführt. Für jedes Zentrum z ist ein Zielgewicht μ_z für den z enthaltenden Teilbaum gegeben. Die maximale Abweichung des Gewichts der Teilbäume von ihrem Zielgewicht ist zu minimieren. Sei ζ^* dieses Minimum. Wie oben erhalten wir ζ^* durch Binärsuche: Setze für $\zeta \geq 0$ $L_z := \mu_z - \zeta$ und $U_z := \mu_z + \zeta$. Dadurch wird eine Instanz von $T/Zent, U_z - L_z = const/\cdot/\exists$ definiert. Ist sie unzulässig, so muß $\zeta^* > \zeta$ sein, andernfalls ist $\zeta^* \leq \zeta$. Der folgende Hilfssatz zeigt, daß die Suche nach ζ^* a priori auf ein Intervall beschränkt werden kann, dessen Länge höchstens wT beträgt. Dabei wird berücksichtigt, daß auch negative Zielgewichte für die Teilbäume, oder solche, die größer sind als wT , vorgegeben sein können.¹⁰

Hilfssatz 5.25

Es seien $\zeta_1 := 0$, $\zeta_2 := \max\{-\mu_z : \mu_z \leq 0\}$ und $\zeta_3 := \max\{\mu_z - wT : \mu_z \geq wT\}$. (Dabei ist $\max \emptyset = 0$.) Dann gilt für ζ^* die Ungleichung

$$\max\{\zeta_1, \zeta_2, \zeta_3\} \leq \zeta^* \leq wT + \max\{\zeta_1, \zeta_2, \zeta_3\}. \quad (5.15)$$

Beweis. Zuerst zeigen wir die linke Ungleichung in (5.15). $\zeta^* \geq \zeta_1$ ist trivial. $\zeta^* \geq \zeta_2$ gilt, weil jeder Teilbaum nichtnegatives Gewicht hat (Voraussetzung $w \geq 0$). Weiter folgt $\zeta^* \geq \zeta_3$ aus der Tatsache, daß $wT' \leq wT$ gilt für jeden Teilbaum T' .

¹⁰Ob die Wahl solcher Zielgewichte sinnvoll ist, ist hier nicht zu diskutieren, jedenfalls soll sie nicht per Definition (des Problems $T/Zent/\cdot/\|\cdot\|_\infty$) ausgeschlossen werden.

Die rechte Ungleichung in (5.15) ergibt sich wie folgt. Jeder Teilbaum T' , für dessen Zentrum z gilt $\mu_z \leq 0$, hat $|wT' - \mu_z| \leq wT - \mu_z$. Genauso erfüllt jeder Teilbaum mit Zentrum z , für das $\mu_z \geq wT$ gilt, $|wT' - \mu_z| \leq \mu_z$. Alle anderen Teilbäume ($0 \leq \mu_z \leq wT$) haben $|wT' - \mu_z| \leq wT$. Somit läßt sich ζ^* , die maximale Abweichung (im Optimum), abschätzen durch

$$\begin{aligned} \zeta^* &\leq \max \{ \max \{ wT - \mu_z : \mu_z \leq 0 \}, \max \{ \mu_z : \mu_z \geq wT - \mu_z \}, wT \} \\ &= \max \{ wT + \zeta_2, wT + \zeta_3, wT + \zeta_1 \} \end{aligned}$$

□

Aus dem Hilfssatz und Satz 5.22 erhalten wir nun eine Abschätzung für den Aufwand zur Lösung eines $T/Zent/\cdot/\|\cdot\|_\infty$ Problems.

Satz 5.26

Der Aufwand zur Lösung von $T/Zent/\cdot/\|\cdot\|_\infty$ ist $O(p^2 n^4 \log wT)$, $p = |Z|$, $n = |V|$.

Kapitel 6

Ein Branch&Bound–Verfahren für

$$T/ \cdot /p/ \| \cdot \|_{\infty}$$

In diesem Kapitel wird ein Branch&Bound–Verfahren für das Baumzerlegungsproblem $T/ \cdot /p/ \| \cdot \|_{\infty}$ vorgestellt. Nach Satz 5.24 ist dieses Problem mit polynomialem Aufwand lösbar, aufbauend auf Algorithmus 5.5. Da auf der anderen Seite die Laufzeit eines Branch&Bound–Verfahrens schlecht a priori analysiert werden kann, ist die Frage nach der Motivation für dessen Entwicklung und Darstellung aufzuwerfen. Folgende Gründe sprechen dafür:

1. Das Branch&Bound–Verfahren wurde entwickelt, als die polynomiale Lösbarkeit von Problem $T/ \cdot /p/ \| \cdot \|_{\infty}$ noch nicht bekannt war.
2. Die Laufzeitabschätzung in Satz 5.24 ist von hohem Grad. Daher ist nicht sicher, ob sich das dahintersteckende Verfahren (dynamische Programmierung) in der Praxis als effizienter erweist, als das Branch&Bound–Verfahren. Für letzteres zeigen Tests, daß es auch große Probleme effizient lösen kann (siehe 6.2).
3. Das Branch&Bound–Verfahren ist Teil von GAMOR und wird insbesondere in 8.3 zur Berechnung von Gebietsaufteilungen verwendet.

6.1 Darstellung des Verfahrens

Die prinzipielle Architektur von Branch&Bound–Verfahren wird als bekannt vorausgesetzt.¹ Unser Verfahren für $T/ \cdot /p/ \| \cdot \|_{\infty}$ folgt diesem Aufbau. Wir beschreiben zunächst den Suchbaum und stellen dann die Berechnung unterer und oberer

¹Es sei zum Beispiel auf Ibaraki [55] verwiesen, als eine der vielen Darstellungen in der Literatur.

Schranken dar. Hierauf folgt die Beschreibung von Reduktionskriterien, welche die Verkleinerung des Suchbaumes zum Ziel haben.

Eine Präsentation des Verfahrens unter Berücksichtigung aller Details wäre an dieser Stelle zu umfangreich. Wir verweisen dafür auf Schröder [96]. Dort ist auch eine Diskussion einiger Design-Entscheidungen zu finden, die hier nicht wiederholt wird. Vielmehr bringen wir das Verfahren in der Zusammensetzung, die sich in umfangreichen Testrechnungen am effizientesten erwiesen hat.

Voraussetzungen. Unser Branch&Bound-Verfahren zur Lösung von $T/\cdot/p/\|\cdot\|_\infty$ macht zwei Voraussetzungen an die Probleminstanzen, die eine leichte Einschränkung gegenüber der Definition dieses Problems in 4.2 darstellen.

- Die Eckengewichte w_v , $v \in V$, des zu zerlegenden Baumes $T = (V, E)$ müssen ganzzahlig und positiv sein.
- Das Zielgewicht μ muß das mittlere Gewicht der Teilbäume sein, das heißt $\mu = wT/p$.

Hinsichtlich des Einsatzes des Verfahrens in Algorithmen für die Gebietsaufteilung sind diese Einschränkungen nicht wesentlich. Denn dabei entsprechen die Eckengewichte den (stets positiven) Größenattributen der KGE, welche (ggf. nach Skalierung) auf ganzzahlige Werte gerundet werden können. Ferner ist die Minimierung der Abweichung von der *mittleren Größe* der Bezirke das in der Praxis normalerweise angestrebte Ziel.

Für die Darstellung des Branch&Bound-Verfahrens ist es zweckmäßig, für den Baum T eine Wurzel $v_0 \in V$ zu wählen, und T als Wurzelbaum aufzufassen. Die Wahl der Wurzel ist beliebig und wird für das ganze Verfahren beibehalten. Wir erinnern an die in 4.1.1.1 eingeführte Notation, insbesondere an die durch die Wurzel gestiftete Ordnung \prec auf den Ecken.

Weiter ist es von Vorteil, Zerlegungen von T durch Eckenteilmengen zu beschreiben, welche die Wurzel enthalten. Ist $\pi = \{T'_1, \dots, T'_s\}$ eine Zerlegung von T , so sind die Wurzeln der Teilbäume durch $v_i = \min_{\prec} V(T'_i)$, $i = 1, \dots, s$ gegeben, dabei muß für genau ein i gelten $v_i = v_0$. Ist andererseits $V' \subset V$ eine Teilmenge der Ecken, die v_0 enthält, so wird dadurch in eindeutiger Weise eine Zerlegung $\pi(V')$ von T definiert: In dieser Zerlegung sind die Ecken in V' die Wurzeln der Teilbäume, das heißt, $\pi(V')$ wird gebildet, indem die Kanten $e(v)$, $v \in V' - \{v_0\}$, aus T entfernt werden. Jede p -elementige Teilmenge V' , die die Wurzel enthält, definiert eine zulässige Zerlegung für das Problem $T/\cdot/p/\|\cdot\|_\infty$. Ihr Zielfunktionswert ist

$$c\pi(V') = \max\{|wT'_i - \mu| : i = 1, \dots, p\}.$$

6.1.1 Der Suchbaum

6.1.1.1 Aufbau des Suchbaumes

Knoten des Suchbaumes. Im Suchbaum des Branch&Bound-Verfahrens wird das Problem, unter allen zulässigen Zerlegungen von T die beste zu finden, in Teilprobleme zerlegt. Diese sind den Knoten zugeordnet und es wird jeweils eine Teilmenge der Zerlegungen untersucht und bewertet.

Eine solche Teilmenge der Zerlegungen ist durch zwei (disjunkte) Teilmengen der Ecken charakterisiert, auf folgende Weise: $V^1 \subset V$ mit $v_0 \in V^1$ enthält Ecken, die zwingend als Wurzel eines Teilbaumes vorgeschrieben sind. $V^0 \subset V$ mit $V^0 \cap V^1 = \emptyset$ enthält Ecken, die als Wurzel eines Teilbaumes verboten sind. Die Teilmenge der Zerlegungen ist damit gegeben durch

$$\{\pi(V') : V' \supseteq V^1, V' \cap V^0 = \emptyset\}.$$

In jedem Knoten des Suchbaumes wird ein Teilproblem untersucht. Ein Knoten φ ist daher ein Paar $\varphi = (V^0, V^1)$ mit

$$V^0, V^1 \subset V, \quad V^0 \cap V^1 = \emptyset, \quad v_0 \in V^1.$$

Ferner ist

$$\mathcal{P}_p(T, \varphi) := \{\pi(V') : V' \supseteq V^1, V' \cap V^0 = \emptyset, |V'| = p\}$$

die Menge der zulässigen Zerlegungen unter den im Knoten φ untersuchten.

Struktur des Suchbaumes. Der Knoten $\varphi_0 = (\{v_0\}, \emptyset)$ bildet die Wurzel des Suchbaumes. Offenbar ist $\mathcal{P}_p(T, \varphi_0)$ die Menge aller Zerlegungen von T in p Teilbäume, und damit die Menge aller zulässigen Zerlegungen für das Problem $T/\cdot/p/\|\cdot\|_\infty$.

Ein Knoten ist ein Blatt des Suchbaumes, wenn $\mathcal{P}_p(T, \varphi)$ einelementig ist. Dies ist genau dann der Fall, wenn $|V^1| = p$ oder $|V^0| = |V| - p$ ist. Die entsprechende (zulässige) Zerlegung ist gegeben durch $\pi(V^1)$ beziehungsweise $\pi(V - V^0)$.

Ist ein Knoten kein Blatt, so ist die Menge

$$V^f = V^f(\varphi) = V - (V^0 \cup V^1)$$

der *freien* Ecken jedenfalls nicht leer. Um die Nachfolgerknoten von φ zu bilden, wird eine Ecke $v \in V^f$ ausgewählt; die Nachfolgerknoten sind dann

$$\varphi' = (V^0 \cup \{v\}, V^1) \quad \text{und} \quad \varphi'' = (V^0, V^1 \cup \{v\}).$$

Die Bildung der Nachfolgerknoten wird als *Verzweigen* des Knotens φ bezeichnet. Es ist klar, daß $\mathcal{P}_p(T, \varphi)$ durch $\mathcal{P}_p(T, \varphi')$ und $\mathcal{P}_p(T, \varphi'')$ in zwei disjunkte Teilmengen aufgeteilt wird.

Die Struktur des Suchbaumes ist nicht statisch festgelegt, sondern ergibt sich dynamisch. Zum Beispiel entscheidet die Auswahlregel für die Ecke $v \in V^f$ beim Verzweigen, welches die Nachfolgerknoten von φ sind.

6.1.1.2 Das Ausloten und das Reduzieren von Knoten

Ziel des Branch&Bound–Verfahrens ist die Bestimmung einer Zerlegung $\pi \in \mathcal{P}_p(T, \varphi_0)$, die $c\pi$ minimiert. Dazu wird dieses Problem durch Verzweigen in Teilprobleme aufgeteilt. Andererseits ist für die Effizienz entscheidend, den untersuchten Teil des Suchbaumes möglichst klein zu halten. Hierzu wird das Ausloten und das Reduzieren von Knoten eingesetzt.

Beide Techniken basieren auf der Kenntnis einer (möglichst guten) zulässigen Zerlegung π^* . Dies ist die beste bisher durch das Verfahren bestimmte Lösung.

Ausloten. Zum Ausloten eines Knotens φ wird gewöhnlich eine untere Schranke lb berechnet, also ein Wert, der

$$lb \leq \min c \mathcal{P}(T, \varphi)$$

erfüllt. φ ist ausgelotet, wenn $lb \geq c\pi^*$ gilt. Ein ausgeloteter Knoten muß nicht verzweigt werden.

In unserem Branch&Bound–Verfahren ist es jedoch geschickter, lb nicht explizit zu berechnen. Für die verwendete untere Schranke (siehe 6.1.4) wäre dies ungleich aufwendiger, als den Test

$$\text{„Ist } lb \geq \alpha\text{?“}$$

mit $\alpha = c\pi^*$ auszuführen. Bei positiver Antwort auf diesen Test ist der Knoten ausgelotet.

Reduzieren. Reduzieren eines Knotens φ ist der Übergang zu einem Nachkommen φ' , wobei

$$c\pi^* \leq \min c (\mathcal{P}_p(\varphi) - \mathcal{P}_p(\varphi'))$$

gilt. Die Mengen V^0 und/oder V^1 werden somit vergrößert; zu beachten ist, daß die dadurch von der weiteren Untersuchung ausgeschlossenen Zerlegungen nicht besser sind als π^* .

In unserem Verfahren werden zwei Kriterien zum Reduzieren verwendet, eines erlaubt das Hinzufügen einer freien Ecke zu V^0 , das andere zu V^1 . Sie können auch mehrfach hintereinander angewendet werden (siehe 6.1.3).

6.1.1.3 Heuristische obere Schranke

Ausloten und Reduzieren sind umso wirksamer, je besser die Zerlegung π^* ist. Neben diesen Operationen ist es daher wichtig, möglichst schnell gute Zerlegungen zu finden.

Erreicht der Suchvorgang ein Blatt des Suchbaumes, ist dadurch eine zulässige Zerlegung π von T bestimmt. Falls diese besser ist als π^* , wird die Aktualisierung $\pi^* := \pi$ vorgenommen.

Verläßt man sich allein auf diesen Mechanismus zur Gewinnung zulässiger Lösungen, ist es nahezu zwingend, den Suchbaum per Tiefensuche zu traversieren. Unser Verfahren sollte nicht auf diese eine Strategie festgelegt sein, deshalb wurde eine Heuristik zur Berechnung einer zulässigen Zerlegung π , ausgehend von einem Knoten φ , eingebaut (siehe 6.1.5). Dabei wird π so bestimmt, daß $\pi \in \mathcal{P}_p(T, \varphi)$ gilt.

6.1.1.4 Untersuchung eines Knotens

Die folgende Prozedur `untersuche_knoten` zeigt, wie die skizzierten Bausteine des Branch&Bound-Verfahrens zusammenspielen. In den folgenden Abschnitten werden sie detailliert besprochen.

Algorithmus 6.1 Untersuchen eines Knotens im Suchbaum

```

1: procedure untersuche_knoten( $\varphi$ ) begin
2:   if  $\varphi$  ist Blatt then
3:     aktualisiere ggf.  $\pi^*$ ;
4:   else
5:     reduziere  $\varphi$  soweit wie möglich
6:     if  $\varphi$  ist Blatt then
7:       aktualisiere ggf.  $\pi^*$ ;
8:     else if  $\varphi$  kann nicht ausgelotet werden then
9:       berechne heuristisch  $\pi \in \mathcal{P}_p(T, \varphi)$ 
10:      aktualisiere ggf.  $\pi^*$ ;
11:      verzweige  $\varphi$ 
12:     end if
13:   end if
14: end

```

Zu Zeile 6 ist anzumerken, daß ein Knoten durch Reduzieren zu einem Blatt werden kann. In Zeile 8 ist zu testen, ob die untere Schranke für φ größer oder gleich $c\pi^*$ ist. Ist das Resultat negativ, so kann φ nicht ausgelotet werden; die Bedingung in Zeile 8 ist dann erfüllt.

Ist es nötig, φ zu verzweigen, werden die beiden Nachfolgerknoten zunächst ei-

ner Liste der zu untersuchenden Knoten hinzugefügt. Aus dieser Liste wird dann in geeigneter Weise der nächste Knoten für `untersuche_knoten` ausgewählt. Details hierzu sind in Schröder [96] zu finden.

6.1.2 Einige Definitionen

Zur Darstellung der Verfahrensbestandteile Reduzieren, Ausloten und heuristische obere Schranke werden einige Definitionen und Bezeichnungen benötigt.

Der Wald F^φ . Zu einem Knoten $\varphi = (V^0, V^1)$ des Suchbaumes gehört in natürlicher Weise ein Wald F^φ , der aus dem Baum T folgendermaßen gewonnen wird:

- Zunächst werden alle Kanten $e(v)$, $v \in V^1 - \{v_0\}$, aus T entfernt, dadurch werden $|V^1|$ Bäume gebildet.
- Aus diesen Bäumen werden die Elemente von V^0 herausdividiert, wobei das Gewicht zusammenfallender Ecken addiert wird.

Der Wald F^φ spiegelt somit die im Knoten φ vorliegenden Entscheidungen des Branch&Bound-Verfahrens wieder. Zu jeder Ecke $v \in V^1$ gibt es eine Komponente in F^φ , deren Wurzel v ist. Wir bezeichnen diese Komponente mit T_v^φ .

Ist $v \in V^f$ eine freie Ecke, so gehört sie zu einer Komponente von F^φ . Den in v wurzelnden Teilbaum innerhalb dieser Komponente bezeichnen wir ebenfalls mit T_v^φ .

Damit sind für $v \in V^1 \cup V^f$ die (Teil-)Bäume T_v^φ definiert; die Anzahl der Ecken in einem solchen Baum sei n_v .² Offenbar gilt

$$\sum_{v \in V^1} n_v = |V^1| + |V^f|. \quad (6.1)$$

Die Anzahl zu entfernender Kanten. Für jede Zerlegung aus $\mathcal{P}_p(T, \varphi)$ gibt es eine ihr entsprechende Zerlegung von F^φ ; sie wird gebildet, indem einige Komponenten weiter aufgeteilt werden. Insgesamt sind

$$s^* := p - |V^1|$$

Kanten aus F^φ zu entfernen, um einen Wald zu erhalten, der einer zulässigen Zerlegung von T entspricht. Sei nun für $v \in V^1$ \bar{s}_v (beziehungsweise \underline{s}_v) die maximale (minimale) Anzahl an Kanten die hierbei aus T_v^φ entfernt werden kann (muß).

Beobachtung 1. Es gilt

$$\bar{s}_v = \min \{s^*, n_v - 1\} \quad \text{und} \quad \underline{s}_v = \max \{0, s^* - |V^f| + (n_v - 1)\}.$$

²Hier ist es nicht nötig, einen oberen Index φ zu schreiben, da im folgenden stets auf einen festen Knoten φ Bezug genommen wird.

Beweis. Die linke Formel ist offensichtlich. Für die rechte Gleichung beachte man die Tatsache, daß in den Bäumen T_u^φ , $u \in V^1 - \{v\}$, maximal $\sum_{u \in V^1 - \{v\}} n_u - 1$ Kanten entfernt werden können, zusammen mit (6.1). \square

Die Imbalance. Die Imbalance ist ein Maß, mit welchem die Auswirkung des Hinzufügens einer freien Ecke zu V^1 auf den Zielfunktionswert abgeschätzt wird.

Sei T_v^φ , $v \in V^1$, eine Komponente von F^φ und $u \in T_v^\varphi$, $u \neq v$. Es bestehe die Absicht, aus T_v^φ insgesamt s Kanten zu entfernen. Falls die Kante $e(u)$ darunter ist, beträgt die maximale Abweichung vom mittleren Gewicht μ mindestens

$$\text{imb}(u, s) = \min_{\underline{\zeta} \leq \zeta \leq \bar{\zeta}} \max \left\{ \left| \frac{wT_u^\varphi}{\zeta + 1} - \mu \right|, \left| \frac{wT_v^\varphi - wT_u^\varphi}{s - \zeta} - \mu \right| \right\}. \quad (6.2)$$

Die Variable ζ ist die Anzahl der Kanten, die aus T_u^φ entfernt wird, sie wird optimiert unter der Annahme, daß die beiden aus dem Entfernen von $e(u)$ resultierenden Teilbäume in Komponenten genau gleichen Gewichtes unterteilt werden können. Dabei muß ζ zwischen $\underline{\zeta}$ und $\bar{\zeta}$ liegen, wobei

$$\bar{\zeta} = \min \{s - 1, n_u - 1\} \quad \text{und} \quad \underline{\zeta} = \max \{0, s - 1 - (n_v - n_u - 1)\} \\ = \max \{0, s - n_v + n_u\}$$

gelten.

Den Wert $\text{imb}(u, s)$ bezeichnen wir als *Imbalance* von u bezüglich s . Das folgende Ergebnis zeigt, daß die Minimierung über ζ nicht explizit ausgeführt werden muß. Die Imbalance kann daher mit konstantem Zeitaufwand berechnet werden.

Satz 6.1

Sei $\sigma = (s + 1)wT_u^\varphi / wT_v^\varphi - 1$. Für das minimierende ζ in (6.2) gilt

$$\zeta = \begin{cases} \bar{\zeta} & \text{falls } \bar{\zeta} < \sigma, \\ \underline{\zeta} & \text{falls } \underline{\zeta} > \sigma, \\ \lfloor \sigma \rfloor \text{ oder } \lceil \sigma \rceil & \text{sonst.} \end{cases}$$

Der Beweis des Satzes ist in Schröder [96] zu finden.

6.1.3 Kriterien für das Reduzieren

In unserem Branch&Bound-Verfahren werden zwei Reduktionskriterien benutzt.

Mit dem ersten Kriterium kann gegebenenfalls V^0 erweitert werden. Es sei $\varphi = (V^0, V^1)$ der aktuelle Knoten und π^* die bisher beste zulässige Zerlegung. Sei weiter $v \in V^1$ mit $\bar{s}_v \geq 1$ und $u \in T_v^\varphi$, $u \neq v$. Falls die Ungleichung

$$c\pi^* \leq \min \{ \text{imb}(u, s) : \max\{1, \underline{s}_v\} \leq s \leq \bar{s}_v \} \quad (6.3)$$

erfüllt ist, kann die Kante $e(u)$ in einer besseren Zerlegung als π^* nicht entfernt sein. Denn sonst müßten in einer solchen Zerlegung aus T_v^φ s Kanten (für ein s zwischen $\max\{1, \underline{s}_v\}$ und \overline{s}_v) entfernt sein, und die maximale Abweichung von μ betrüge damit mindestens $\text{imb}(u, s)$. Es kann also $V^0 := V^0 \cup \{u\}$ gesetzt werden.

Das zweite Kriterium dient der Vergrößerung von V^1 . Ist $u \in V^f$ und

$$c\pi^* \leq w_u + w_{\text{pre}(u)} - \mu \quad (\text{im Baum } T_v^\varphi, v \in V^1, \text{ der } u \text{ enthält}) \quad (6.4)$$

so muß die Kante $e(u)$ in jeder besseren Zerlegung als π^* entfernt sein. Denn sonst enthielte die Zerlegung einen Teilbaum mit einem Gewicht von mindestens $\mu + c\pi^*$; die Menge V^1 kann daher um u erweitert werden.

In (6.4) beachte man, daß die Eckengewichte aufgrund des Herausdividierens von V^0 in F^φ im allgemeinen größer sind, als in T . Das Kriterium zieht also die im Knoten φ getroffenen Entscheidungen in Betracht.

Ablauf des Reduzierens. In jedem Knoten werden die freien Ecken im allgemeinen mehrfach durchlaufen, um Reduktionen aufgrund von (6.3) und (6.4) zu versuchen. Denn durch das Anwenden eines Kriteriums auf eine Ecke wird möglicherweise erreicht, daß danach auch eine andere Ecke (6.3) oder (6.4) erfüllt (was zuvor nicht der Fall war). Genauer kann das Vergrößern von V^0 die Gültigkeit von (6.3) oder (6.4) für andere Ecken nach sich ziehen. Das Erweitern von V^1 kann dagegen nur (6.3) für andere Ecken gültig werden lassen.

In unserem Branch&Bound-Verfahren wird zunächst (6.3) solange angewendet, wie dies möglich ist. Darauf wird (6.4) für alle (noch) freien Ecken geprüft. Ergeben sich dabei Erfolge, so wird zu (6.3) zurückgekehrt. Das Reduzieren endet, wenn für keine freie Ecke mehr (6.3) oder (6.4) erfüllt ist.

Aufwand für das Reduzieren. In unserer Implementierung ist der Aufwand für das Reduzieren eines Knotens $O(|V|^2p)$. (Siehe Schröder [96] für eine Analyse.)

6.1.4 Das Ausloten: Test einer unteren Schranke

Nach dem Reduzieren des aktuellen Knotens wird versucht, ihn mit Hilfe einer unteren Schranke auszuloten. Wir verwenden eine Schranke, die auf der Verwandtschaft der Probleme $T/\cdot/p/\text{maximin}$ und $T/\cdot/p/\text{minimax}$ mit dem zu lösenden $T/\cdot/p/\|\cdot\|_\infty$ Problem basiert. (Hierbei muß $\mu = wT/p$ gelten, was durch unsere Voraussetzungen (siehe Seite 116) gewährleistet ist.)

6.1.4.1 Ausloten des Wurzelknotens

Ist L^* der optimale Zielfunktionswert von $T/\cdot/p/\text{maximin}$ für T und c^* derjenige von $T/\cdot/p/\|\cdot\|_\infty$, so gilt

$$L^* \geq \mu - c^*,$$

denn die optimale Zerlegung für $T/\cdot/p/\|\cdot\|_\infty$ ist auch für $T/\cdot/p/\text{maximin}$ zulässig und hat für dieses Problem den Zielfunktionswert $\mu - c^*$.

Genauso erfüllt der optimale Zielfunktionswert U^* von $T/\cdot/p/\text{minimax}$ die Beziehung

$$U^* \leq \mu + c^*.$$

Aus beiden Ungleichungen folgt

$$c^* \geq \max\{U^* - \mu, \mu - L^*\} =: lb.$$

Der rechte Term definiert unsere untere Schranke im Fall des Wurzelknotens φ_0 .

Wesentlich ist nun die Beobachtung, daß die Berechnung von lb recht aufwendig ist, denn dazu müssen die Probleme $T/\cdot/p/\text{maximin}$ und $T/\cdot/p/\text{minimax}$ gelöst werden (vgl. die Erörterung nach Korollar 4.26 auf Seite 78). Andererseits kann der Test

$$\text{Ist } lb \geq \alpha? \tag{6.5}$$

für $\alpha \in \mathbb{R}$ in linearer Zeit ausgeführt werden, gestützt auf das Lösen der Probleme $T/[L, U]/p/\exists$ (siehe Korollar 4.25). Zum Ausloten wird jedoch nur dieser Test mit $\alpha = c\pi^*$ (für die bisher beste Zerlegung π^*) benötigt.

In dieser einfachen Form ist die Idee allerdings nur im Wurzelknoten φ_0 anwendbar. Tiefer im Suchbaum besitzt der Wald F^φ im allgemeinen mehrere Komponenten und es ist a priori nicht klar, wie die Anzahl s^* der noch zu bildenden Teilbäume auf diese Komponenten aufgeteilt werden soll. Daher ist zunächst unbekannt, wie die Anzahl der Teilbäume zu wählen ist, wenn für eine Komponente T_v^φ die Probleme vom Typ $T/[L, U]/p/\exists$ gelöst werden. Für das Ausloten vom Wurzelknoten verschiedener Knoten ist somit eine verfeinerte Betrachtung nötig. Aus ihr ergibt sich dann auch die genaue Vorgehensweise beim Ausloten des Wurzelknotens.

6.1.4.2 Ausloten beliebiger Knoten

Sei $\varphi = (V^0, V^1)$ der aktuelle Knoten des Suchbaumes und $v \in V^1$. Eine *partielle untere Schranke* $lb_v(q)$ ist eine untere Schranke für das Baumzerlegungsproblem $T_v^\varphi/\cdot/q/\|\cdot\|_\infty$, also die Zerlegung der Komponente T_v^φ von F^φ in q Teilbäume mit dem Ziel die maximale Abweichung von μ zu minimieren. (Hierbei ist $\mu = wT/p$ das „globale“ mittlere Gewicht.) Für alle $q \in \{1, \dots, n_v\}$ ist dieses Problem zulässig.

Ist nun für alle $v \in V^1$ und alle jeweils zulässigen q eine partielle untere Schranke bekannt, so läßt sich daraus eine untere Schranke lb für alle Zerlegungen aus $\mathcal{P}_p(T, \varphi)$ gewinnen:

$$lb = \min \left\{ \max_{v \in V^1} lb_v(q_v) : \underline{s}_v \leq q_v - 1 \leq \bar{s}_v \text{ für alle } v \in V^1 \text{ und } \sum_{v \in V^1} (q_v - 1) = s^* \right\} \quad (6.6)$$

Darin wird über alle Tupel $(q_v)_{v \in V^1}$ minimiert, die eine mögliche Aufteilung der noch abzuschneidenden s^* Teilbäume auf die Komponenten von F^φ charakterisieren. Für das minimierende Tupel ist die maximale partielle Schranke eine untere Schranke für den Knoten φ . (Für einen Beweis siehe Schröder [96, Proposition 2.3].)

Eine enumerierende Berechnung von (6.6) wäre unvertretbar aufwendig. Deshalb muß die partielle Schranke so definiert werden, daß (6.6) effizient berechnet werden kann. Aus der in 6.1.4.1 dargestellten Idee läßt sich eine solche Schranke gewinnen.

Für $v \in V^1$ und $q \in \{1, \dots, n_v\}$ sei $U_v^*(q)$ der optimale Zielfunktionswert des Problems $T_v^\varphi/\cdot/q/\text{maximin}$ und $L_v^*(q)$ der von $T_v^\varphi/\cdot/q/\text{minimax}$. Die Definition unserer partiellen unteren Schranke lautet

$$lb_v(q) = \max \{U_v^*(q) - \mu, \mu - L_v^*(q)\}. \quad (6.7)$$

Sie stimmt mit der oben eingeführten unteren Schranke für den Wurzelknoten φ_0 überein. Nun bezeichne für $\alpha \in \mathbb{R}_{\geq 0}$

$$Q_v(\alpha) = \{q : 1 \leq q \leq n_v, lb_v(q) < \alpha\}.$$

Wie oben erläutert, ist unsere Absicht die Verwendung von $\alpha = c\pi^*$. Die Grundlage für die effiziente Ausführung des Tests (6.5) ist das folgende Ergebnis.

Satz 6.2

Für $v \in V^1$ und $\alpha \in \mathbb{R}_{\geq 0}$ ist $Q_v(\alpha)$ ein (möglicherweise leeres) Intervall ganzer Zahlen.

Den Beweis und die Berechnung von $Q_v(\alpha)$ bringen wir nach dem entscheidenden Resultat:

Korollar 6.3

Es sei $Q'_v(\alpha) = Q_v(\alpha) \cap \{\underline{s}_v + 1, \dots, \bar{s}_v + 1\}$, $v \in V^1$. Für die Schranke lb gemäß (6.6) und (6.7) gilt: Es ist $lb \geq \alpha$ genau dann, wenn eine der Bedingungen

- i. $Q'_v(\alpha) = \emptyset$ für ein $v \in V^1$,
- ii. $\sum_{v \in V^1} (\min Q'_v(\alpha) - 1) > s^*$,

$$\text{iii. } \sum_{v \in V^1} (\max Q'_v(\alpha) - 1) < s^*$$

wahr ist.

Beweis. Die Bedingung $lb \geq \alpha$ ist äquivalent zur Aussage

$$\text{Es gibt kein Tupel } (q_v)_{v \in V^1} \text{ mit } \underline{s}_v \leq q_v - 1 \leq \bar{s}_v \text{ für alle } v \text{ und} \\ \sum_{v \in V^1} (q_v - 1) = s^* \text{ und } \max_{v \in V^1} lb_v(q_v) < \alpha \quad (6.8)$$

und damit auch zu

$$\text{Es gibt kein Tupel } (q_v)_{v \in V^1} \text{ mit } q_v \in Q'_v(\alpha) \text{ für alle } v \text{ und } \sum_{v \in V^1} (q_v - 1) = s^*. \quad (6.9)$$

Weil alle $Q'_v(\alpha)$ (möglicherweise leere) Intervalle ganzer Zahlen sind, ist (6.9) äquivalent zu i. bis iii. \square

Die Überprüfung der Bedingungen i. bis iii. (und damit das Ausloten von φ) ist trivial, wenn die Intervalle $Q_v(\alpha)$ berechnet sind. Letzteres kann nun auf das Lösen der Probleme $T_v^\varphi/L/\cdot/\max|\pi|$ und $T_v^\varphi/U/\cdot/\min|\pi|$ mit geeignet gewählter Gewichtschränke L beziehungsweise U zurückgeführt werden.

Beweis von Satz 6.2. Es ist $lb_v(q) < \alpha$ genau dann, wenn

$$U_v^*(q) < \mu + \alpha \quad \text{und} \quad L_v^*(q) > \mu - \alpha \quad (6.10)$$

gilt. Wegen der Ganzzahligkeit der Gewichte in T_v^φ sind auch $U_v^*(q)$ und $L_v^*(q)$ ganzzahlig. Damit ist (6.10) äquivalent zu

$$U_v^*(q) \leq \lceil \mu + \alpha - 1 \rceil \quad \text{und} \quad L_v^*(q) \geq \lfloor \mu - \alpha + 1 \rfloor \quad (6.11)$$

Ist die linke Bedingung für ein q erfüllt, dann gilt dies offenbar auch für alle $q' \in \{q+1, \dots, n_v\}$. Das minimale solche q findet man durch Lösen von $T_v^\varphi/U/\cdot/\min|\pi|$ mit $U = \lceil \mu + \alpha - 1 \rceil$. Es sei mit q_U^* bezeichnet.

Genauso findet man das maximale q , für das die rechte Bedingung in (6.11) gilt, durch Lösen von $T_v^\varphi/L/\cdot/\max|\pi|$ mit $L = \lfloor \mu - \alpha + 1 \rfloor$; es sei mit q_L^* bezeichnet.

Damit gilt

$$Q_v(\alpha) = \{1, \dots, q_L^*\} \cap \{q_U^*, \dots, n_v\}.$$

\square

Aufgrund von Satz 4.24 kann somit $Q_v(\alpha)$ mit Aufwand $O(|V(T_v^\varphi)|)$ bestimmt werden. Damit erhalten wir das zentrale Ergebnis über das Ausloten in unserem Branch&Bound-Verfahren.

Korollar 6.4

Der Test, ob ein Knoten φ des Suchbaumes basierend auf der Schranke lb gemäß (6.6) und (6.7) ausgelotet werden kann, ist mit Aufwand $O(|V|)$ durchführbar.

Algorithmus 6.2 Berechnung einer Zerlegung von T , ausgehend von F^φ

Input: Wald F^φ zum aktuellen Knoten $\varphi = (V^0, V^1)$ und $s_v, v \in V^1$.

Output: V' mit $\pi(V') \in \mathcal{P}_p(T, \varphi)$

- 1: $F' := F^\varphi; V' := V^1; s'_v := s_v$ für $v \in V^1$
 - 2: **while** $|V'| < p$ **do**
 - 3: wähle $v \in V'$ mit $s'_v > 0$
 - 4: wähle $u \in T'_v$ so, daß $\text{imb}'(u, s'_v)$ minimal ist; dabei sei ζ minimierend in (6.12)
 - 5: entferne die Kante $e(u)$ aus F'
 - 6: $s'_u := \zeta; s'_v := s'_v - \zeta - 1$
 - 7: $V' := V' \cup \{u\}$
 - 8: **end while**
-

6.1.5 Heuristik zur Berechnung einer oberen Schranke

Sei $\varphi = (V^0, V^1)$ der aktuelle Knoten. Wenn er nicht ausgelotet werden kann, wird eine Heuristik zur Berechnung einer Zerlegung $\pi \in \mathcal{P}_p(T, \varphi)$ eingesetzt. Sie beruht auf der Idee, zunächst $V' = V^1$ zu setzen, und dann s^* freie Ecken mit kleiner Imbalance zu V' hinzuzufügen. Für die Zerlegung $\pi = \pi(V')$ gilt dann $\pi \in \mathcal{P}_p(T, \varphi)$.

Dabei werden aus der Komponente $T_v^\varphi, v \in V^1$, insgesamt s_v Ecken gewählt, mit $\underline{s}_v \leq s_v \leq \bar{s}_v$ und $\sum_{v \in V^1} s_v = s^*$. Die s_v werden zuvor so bestimmt, daß sie einer Lösung von (6.6) entsprechen, wobei jedoch eine anders definierte partielle untere Schranke zugrunde gelegt wird.³

Es sei $u \in V^f$ und T_u^φ die Komponente von F^φ , die u enthält. Eine kleine Imbalance $\text{imb}(u, s_v)$ von u ist ein Indiz dafür, daß es günstig ist, die Kante $e(u)$ zu entfernen, wenn T_u^φ in $s_v + 1$ Teilbäume zerlegt wird; dadurch ist unsere Heuristik motiviert.

Im Lauf der Heuristik (Algorithmus 6.2) werden aus F^φ Kanten entfernt. Der resultierende Wald wird mit F' bezeichnet. Er umfaßt zu jedem $v \in V'$ eine in v wurzelnde Komponente T'_v . Für die weiteren Ecken $u \in F'$ ist T'_u der Teilbaum, der von u und allen seinen Nachfolgern in F' gebildet wird. Weiter wird durch s'_v die Anzahl der noch aus T'_v zu entfernenden Kanten angegeben, $v \in V'$. Die Imbalance wird nach der Formel

$$\text{imb}'(u, s) = \min_{\underline{\zeta} \leq \zeta \leq \bar{\zeta}} \max \left\{ \left| \frac{wT'_u}{\zeta + 1} - \frac{wT'_v}{s + 1} \right|, \left| \frac{wT'_v - wT'_u}{s - \zeta} - \frac{wT'_v}{s + 1} \right| \right\} \quad (6.12)$$

berechnet. Man beachte, daß im Unterschied zu (6.2) in der Heuristik mit dem „lokalen“ mittleren Gewicht $wT'_v/s + 1$ gearbeitet wird; dies hat sich in Testrechnungen

³Für die Details verweisen wir auf Schröder [96]. Die zugrundegelegte partielle Schranke wurde im vorangehenden Abschnitt aus Platzgründen nicht diskutiert.

als vorteilhafter erwiesen (siehe Schröder [96, Abschnitt 4.2]).

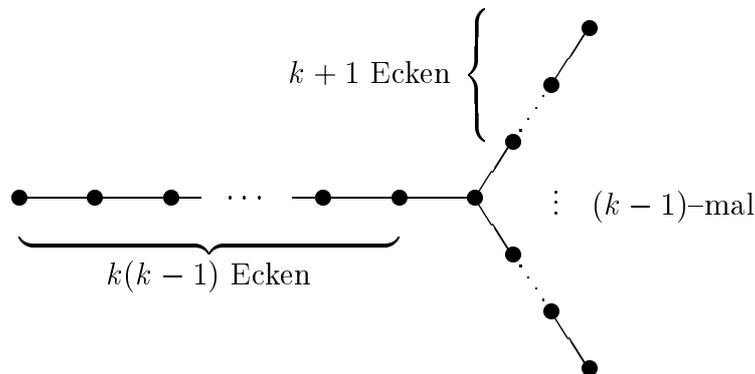
Wie der folgende Satz zeigt, muß die Frage nach einer Güte-Garantie für die Heuristik negativ beantwortet werden. Die zugrundeliegenden Instanzen von $T/\cdot/p/\|\cdot\|_\infty$ sind jedoch sehr speziell. Geht man von den Rechenergebnissen des nächsten Abschnitts aus, ist erkennbar, daß Algorithmus 6.2 im allgemeinen eine gute Lösung liefert.

Satz 6.5

Für $k \geq 2$ gibt es einen eckengewichteten Baum T^k und $p_k \in \mathbb{Z}_{>0}$ mit folgenden Eigenschaften:

- Algorithmus 6.2 berechnet (im Wurzelknoten φ_0 des Suchbaumes) eine Zerlegung mit Zielfunktionswert $k - 1$.
- Der optimale Zielfunktionswert der Instanz $T^k/\cdot/p_k/\|\cdot\|_\infty$ beträgt Eins.

Beweis. Wie folgende Abbildung zeigt, konstruieren wir T^k als Spinne mit einem langen Bein, bestehend aus $k(k - 1)$ Ecken, und $k - 1$ Beinen der Länge $k + 1$.



Das Gewicht aller Ecken der Beine ist Eins; der zentralen Ecke weisen wir ein Gewicht von $k + 1$ zu. Damit ist $wT^k = 2k^2$. Nun setzen wir $p_k := 2k$; dies ergibt ein mittleres Gewicht $\mu = wT^k/p_k = k$.

Man überzeugt sich leicht, daß Algorithmus 6.2 (unabhängig von der gewählten Wurzel von T^k) zunächst $2k - 2$ Teilbäume vom Gewicht μ abschneidet: $k - 1$ davon werden durch die jeweils äußeren k Ecken der kurzen Beine gebildet; weitere $k - 1$ entstehen durch Zerlegen des langen Beines. Der verbleibende Stern vom Gewicht $2k$ muß nun noch in zwei Teilbäume zerlegt werden. Dies ist nur möglich, indem eine Ecken abgeschnitten wird und für sich einen der beiden Teilbäume bildet. Damit beträgt die maximale Abweichung der Teilbaumgewichte vom Mittel $k - 1$.

Optimal dagegen ist es, das lange Bein in k Abschnitte der Länge $k-1$ zu zerlegen und die kurzen Beine direkt an der zentralen Ecke abzuschneiden. So entstehen $k + (k-1) + 1 = p_k$ Teilbäume mit Gewicht $k-1$ oder $k+1$. \square

6.2 Rechenergebnisse

Die in Schröder [96] präsentierten Rechenergebnisse für unser Branch&Bound-Verfahren dienen vor allem dem Ziel, die bestmögliche Konfiguration des Verfahrens zu finden; sie sollen hier nicht noch einmal aufgeführt werden. Ziel dieses Abschnitts ist vielmehr die Überprüfung der Frage, ob der Algorithmus effizient genug ist, um in Verfahren für die Gebietsaufteilung zum Einsatz zu kommen, insbesondere bei großen Problemen.

Die Rechenergebnisse dieses Abschnitts basieren daher auf der Anwendung des Branch&Bound-Verfahrens auf große, zufällig generierte Bäume mit Eigenschaften, die durch die Absicht der Anwendung in Teil III motiviert sind.

6.2.1 Testprobleme

6.2.1.1 Zufällige Bäume

Die zufällige Erzeugung von Bäumen orientiert sich an folgenden Überlegungen.

1. Es sollen mittlere und große Probleme gerechnet werden. Daher ist eine Eckenzahl zwischen 100 und 1000 sinnvoll.
2. Mit Blick auf die Anwendung wird davon ausgegangen, daß die Bäume Gerüste von ebenen Graphen sind. Es ist somit zu erwarten, daß die Eckengrade klein sind.

Der einfache Algorithmus 6.3 zur Generierung zufälliger Bäume orientiert sich an diesen Überlegungen.

In der äußeren Iteration (Zeilen 3 bis 12) wird zur Ecke v die Anzahl s ihrer Söhne zufällig bestimmt. Dabei wird folgende Wahrscheinlichkeitsverteilung verwendet:

s	0	1	2	3	4
P	0.4	0.2	0.25	0.1	0.05

Tabelle 6.1: Wahrscheinlichkeit für Anzahl der Söhne

Algorithmus 6.3 Erzeugen eines zufälligen Baumes mit kleinen Eckengraden.

Input: Eckenzahl n

Output: $T = (V, E)$ mit $|V| = n$

```

1:  $V := \{1, \dots, n\}$ 
2:  $v := 1; u := 1$ 
3: while  $u < n$  do
4:   wähle zufällig  $s \in \{0, \dots, 4\}$  mit Wahrscheinlichkeit gemäß Tabelle 6.1
5:   if  $v = u$  und  $s = 0$  then  $s := 1$  end if
6:   if  $s + u > n$  then  $s := n - u$  end if
7:   while  $s > 0$  do
8:      $u := u + 1; s := s - 1$ 
9:      $E := E \cup \{(v, u)\}$ 
10:  end while
11:   $v := v + 1$ 
12: end while

```

In der inneren Schleife (Zeilen 7 bis 10) werden die Kanten zwischen der Ecke v und ihren Söhnen erzeugt. Die Zeilen 5 und 6 dienen lediglich dem Abfangen von Randfällen.

6.2.1.2 Zufällige Eckengewichte

Zufällige Eckengewichte wurden mit zwei verschiedenen Verteilungen generiert.

1. Gleichverteilte ganzzahlige Gewichte aus $\{50, \dots, 1000\}$.
2. Logarithmisch-normalverteilte Gewichte. Dazu wurden mit der Methode nach Teichroew (Knuth [60, S. 112]) $N(0, 1)$ -verteilte Zufallszahlen x_1, \dots, x_n gezogen und diese wurden mittels

$$w_v := \lfloor \exp(x_v + 9) \rfloor$$

auf (angenähert) $LOG(9, 1)$ -verteilte Gewichte transformiert.

Mit dem Modell der zweiten Variante wurde versucht, Probleme zu simulieren, in denen die Eckengewichte den Bevölkerungszahlen in kleinsten Gebietseinheiten (KGE) entsprechen. Dies ist zum Beispiel für den Anwendungsbereich der Wahlkreisgestaltung eine plausible Annahme. Dem Modell liegt die Idee zugrunde, daß Bevölkerungszahlen aus (exponentiellen) Abnahme- oder Zunahme-Prozessen hervorgehen, wobei die Änderungsrate einer Normalverteilung genügt.⁴

⁴Inwiefern dieses Modell realitätsnah ist, kann hier nicht weiter untersucht werden. Letztlich müßte das seitens der Bevölkerungsgeographie entschieden werden. Die in Bär [8, S. 46] abgebil-

6.2.1.3 Menge der Testinstanzen

Erzeugung der gewichteten Bäume. Zunächst wurde mit Algorithmus 6.3 je ein Baum mit $|V| \in \{100, 250, 500, 1000, 1500\}$ generiert. Zu jedem dieser Bäume wurden zehnmal sowohl gleichverteilte, als auch logarithmisch-normalverteilte Gewichte erzeugt. Damit standen zu jeder Eckenzahl und jedem Gewichte-Typ zehn eckengewichtete Bäume zur Verfügung.

Wahl der Teilbaumzahl p . Zunächst war die Idee, für jeden eckengewichteten Baum zehn verschiedene Instanzen von $T/\cdot/p/\|\cdot\|_\infty$ zu lösen, mit Werten von p , die gleichmäßig im Intervall $\{1, \dots, |V|\}$ verteilt sind. Für die Bäume mit gleichverteilten Eckengewichten wurde dies auch so durchgeführt.

Im Fall logarithmisch-normalverteilter Gewichte ist es jedoch nicht sinnvoll, p deutlich größer als $|V|/10$ zu wählen. Denn sonst ist in der Regel das Gewicht einiger Ecken deutlich größer als das mittlere Gewicht μ der Teilbäume; somit ist jede Lösung optimal, die einen Teilbaum enthält, welcher nur aus der Ecke maximalen Gewichtes besteht. Daher sind Instanzen mit großem p (relativ zu $|V|$) sehr einfach lösbar.

Für logarithmisch-normalverteilte Gewichte wurden daher Instanzen mit fünf verschiedenen Werten p gelöst, die sich aus

$$p = \lfloor |V|i/90 \rfloor \quad \text{mit } i \in \{3, 5, 7, 9, 11\}$$

ergeben.

6.2.2 Testergebnisse

Die Testergebnisse können den Tabellen 6.2 und 6.3 entnommen werden. Jede Zeile enthält die aggregierten Ergebnisse für zehn Instanzen, in denen derselbe Baum der Größe $|V| = n$ in p Teilbäume zerlegt wurde; mit jeweils anderen Eckengewichten.

Angegeben ist die Größe des Suchbaumes (Anzahl der Knoten), gemittelt über zehn Instanzen, sowie die minimale und die maximale Größe für diese Instanzen. Ferner ist die minimale, die mittlere und die maximale Rechenzeit angegeben. Die Einheit sind CPU-Sekunden auf einem PC mit Pentium-Prozessor und 500MHz Taktfrequenz. Die letzte Spalte ist für beide Tabellen unterschiedlich und wird unten erläutert.

de Lorenzkurve der Bevölkerungsverteilung in Deutschland zeigt jedoch deutlich das Prinzip der Konzentration, das heißt, viele kleine und wenige sehr große Gemeinden. Ähnlich verhalten sich die logarithmisch-normalverteilten Zufallszahlen.

n	p	Knoten Suchbaum			CPU-Sekunden			$t_{avg}/(\tilde{p}n^2)$ $\times 10^6$
		min	avg	max	min	avg	max	
100	5	1	2.4	9	0.08	0.09	0.10	1.760
100	15	1	3.6	9	0.08	0.11	0.16	0.713
100	25	1	2.4	7	0.08	0.10	0.15	0.416
100	35	1	6.3	13	0.07	0.21	0.42	0.594
100	45	1	7.0	25	0.10	0.23	0.45	0.516
100	55	2	9.1	21	0.13	0.42	0.77	0.927
100	65	1	3.7	7	0.10	0.17	0.34	0.489
100	75	2	6.5	13	0.11	0.20	0.38	0.808
100	85	3	7.7	15	0.12	0.18	0.25	1.200
100	95	1	1.8	4	0.06	0.08	0.11	1.700
250	12	1	7.1	22	0.49	0.53	0.69	0.712
250	37	1	9.1	25	0.49	0.74	1.09	0.320
250	62	1	7.2	20	0.48	0.96	2.30	0.247
250	87	6	16.2	28	1.04	2.51	5.76	0.462
250	112	7	17.3	47	1.12	2.96	5.72	0.423
250	137	3	17.5	35	1.18	4.08	9.28	0.578
250	162	4	9.3	12	1.02	2.29	5.65	0.416
250	187	8	18.2	41	1.14	2.12	4.11	0.538
250	212	2	22.6	69	0.63	1.46	3.33	0.614
250	237	1	5.4	12	0.35	0.52	0.74	0.641
500	25	1	15.4	64	2.11	2.48	3.39	0.396
500	75	4	10.3	17	2.29	3.15	4.30	0.168
500	125	2	7.0	13	1.92	3.13	4.95	0.100
500	175	5	24.1	65	2.53	18.94	47.90	0.433
500	225	7	25.9	73	11.10	23.18	59.61	0.412
500	275	9	34.0	123	10.70	30.27	48.94	0.538
500	325	2	16.4	29	4.33	22.89	44.66	0.523
500	375	10	21.3	43	4.42	10.88	20.43	0.348
500	425	11	16.9	25	3.31	5.05	6.76	0.269
500	475	1	12.9	46	1.51	2.88	6.03	0.460
1000	50	1	16.1	35	10.89	13.07	16.22	0.261
1000	150	12	21.6	56	13.74	21.71	35.39	0.145
1000	250	4	12.6	36	10.34	37.92	160.58	0.152
1000	350	30	59.8	96	72.08	168.73	302.45	0.482
1000	450	25	62.0	122	43.64	333.82	606.30	0.742
1000	550	18	34.3	57	72.64	314.14	633.68	0.698
1000	650	7	18.5	43	29.08	104.38	225.51	0.298
1000	750	17	31.9	46	27.18	48.20	75.09	0.193
1000	850	21	31.8	48	23.62	34.88	47.86	0.233
1000	950	3	10.7	22	11.26	14.34	20.16	0.287

Tabelle 6.2: Testergebnisse für gleichverteilte Eckengewichte

n	p	Knoten Suchbaum			CPU-Sekunden			t_{avg}/n^2 $\times 10^5$
		min	avg	max	min	avg	max	
100	3	1	1.0	1	0.08	0.08	0.09	0.820
100	5	1	3.5	13	0.08	0.09	0.12	0.910
100	7	1	3.1	7	0.09	0.10	0.12	0.980
100	10	1	4.1	11	0.08	0.11	0.14	1.080
100	12	1	5.8	17	0.08	0.12	0.17	1.170
250	8	1	2.9	7	0.48	0.49	0.52	0.790
250	13	1	4.4	15	0.48	0.51	0.57	0.819
250	19	1	15.0	43	0.48	0.65	0.85	1.037
250	25	1	4.7	17	0.15	0.55	0.74	0.875
250	30	1	4.1	16	0.07	0.45	0.83	0.722
500	16	1	10.5	29	2.14	2.26	2.43	0.902
500	27	1	14.4	55	2.07	2.42	3.34	0.969
500	38	1	10.6	53	2.11	2.48	4.30	0.990
500	50	1	4.7	16	1.42	2.15	2.98	0.862
500	61	1	2.9	11	0.27	1.66	3.41	0.662
1000	33	1	18.6	52	8.84	10.11	13.03	1.011
1000	55	7	20.1	45	9.87	12.74	18.44	1.274
1000	77	1	7.6	28	8.29	9.58	12.09	0.958
1000	100	1	8.9	54	2.72	8.67	15.82	0.867
1000	122	1	4.1	20	0.64	7.61	31.75	0.761
1500	50	5	19.1	64	20.67	23.69	28.01	1.053
1500	83	1	13.1	34	19.00	27.29	48.16	1.213
1500	116	1	8.7	29	18.47	24.23	37.30	1.077
1500	150	1	7.4	31	11.92	22.03	47.99	0.979
1500	183	1	2.4	9	1.56	11.64	43.07	0.517

Tabelle 6.3: Testergebnisse für logarithmisch-normalverteilte Eckengewichte

6.2.2.1 Gleichverteilte Eckengewichte

Wie aus Tabelle 6.2 hervorgeht, nimmt die Suchbaumgröße nur sehr moderat mit der Problemgröße zu. Der größte Suchbaum aller gelösten 400 Instanzen hat 123 Knoten, dies zeigt, daß die Komponenten des vorgestellten Branch&Bound–Verfahrens sehr wirksam sind.⁵

Weiter ist zu erkennen, daß die Rechenzeit sowohl von der Eckenzahl n als auch von p abhängt. Insbesondere für Instanzen, in denen p um $n/2$ liegt, steigt sie stark an. Läßt man diese einmal außer acht, können alle Probleme in Rechenzeiten unter einer Minute gelöst werden. Damit erscheint das Verfahren geeignet, um in Algorithmen für die Gebietsaufteilung auch bei großen Problemen zum Einsatz zu kommen.

In der letzten Spalte von Tabelle 6.2 wird versucht, eine empirische Aufwandsabschätzung des Branch&Bound–Verfahrens für die gelösten Instanzen vorzunehmen. Numerisch wird die Hypothese geprüft, daß der Aufwand nicht stärker wächst als $O(\tilde{p}n^2)$, wobei $\tilde{p} = \min\{p, n - p\}$ ist. Dazu wird die mittlere Rechenzeit der zehn Instanzen jeder Spalte durch $\tilde{p}n^2$ dividiert und in einen Bereich um Eins normiert. Die Tatsache, daß die Werte der letzten Spalte mit zunehmender Problemgröße tendenziell abnehmen, läßt die Hypothese gerechtfertigt erscheinen.⁶

Dieses Ergebnis ist insbesondere bedeutsam für die am Eingang des Kapitels aufgeworfene Frage nach dem Sinn eines Suchbaum–Verfahrens für ein als polynomial erwiesenes Problem. Die Komplexität in Satz (5.23) ist $O(p^3n^4 \log wT)$ und damit um Größenordnungen schlechter als der empirische Aufwand des Branch&Bound–Verfahrens. Ferner ist die Abhängigkeit von der Größe der Gewichte ein Nachteil der dynamischen Programmierung gegenüber dem Branch&Bound. Beim gegenwärtigen Stand ist letzteres Verfahren für den praktischen Einsatz zur Lösung von $T/\cdot/p/\|\cdot\|_\infty$ Problemen wohl vorzuziehen; in GAMOR ist dem Rechnung getragen.

6.2.2.2 Logarithmisch–normalverteilte Eckengewichte

Tabelle 6.3 zeigt, daß auch im Falle logarithmisch–normalverteilter Eckengewichte das Branch&Bound–Verfahren effizient das $T/\cdot/p/\|\cdot\|_\infty$ Problem löst. Selbst bei Bäumen mit 1500 Ecken liegt die mittlere Rechenzeit unter einer halben Minute. Die generierten Suchbäume sind klein, maximal (bei 250 Instanzen) waren 64 Knoten nötig, um die optimale Lösung zu finden und ihre Optimalität nachzuweisen.

⁵Dabei kann auf keinen der Bestandteile Reduzieren, Ausloten und obere Schranke Heuristik verzichtet werden, wie in Schröder [96] demonstriert wurde.

⁶Nicht dargestellt ist der Umstand, daß sich für den Nenner $\tilde{p}n^{1.5}$ zunehmende Quotienten ergeben. Das empirische Wachstum des Rechenaufwandes wird daher einem Exponenten zwischen 1.5 und 2 folgen.

Da bei fester Eckenzahl mit zunehmendem p die Probleme eher einfacher zu lösen sind (vgl. die Diskussion in 6.2.1.3), wurde in der letzten Spalte von Tabelle 6.3 (im Unterschied zu Tabelle 6.2) die Hypothese geprüft, daß das empirische Wachstum der mittleren Rechenzeit $O(n^2)$ ist. Die Werte zeigen, daß diese Annahme plausibel ist. Auch dieses Resultat ist ein starkes Argument für die praktische Einsetzbarkeit des Branch&Bound-Verfahrens.

Teil III

Anwendung gleichmäßiger Baumzerlegung in Modellen für die Gebietsaufteilung

Kapitel 7

Modelle und Verfahren für ebene Probleme

Im dritten Teil unserer Arbeit werden bekannte Modelle und Verfahren für die Gebietsaufteilung analysiert und neue entwickelt. Ein Schwerpunkt stellt dabei die Anwendung von gleichmäßiger Baumzerlegung in solchen Verfahren dar. Auf diese Weise werden die Gedankenstränge der Teile I und II, die bisher weitgehend nebeneinander liefen, zusammengeführt.

Wie in 3.1.2 ausgeführt wurde, ist es sinnvoll, eine Unterscheidung von ebenen Modellen (in denen KGE und Zentren Punkten der Ebene entsprechen) und Netzwerk-Modellen (KGE und Zentren sind Ecken eines Graphen) vorzunehmen. In diesem Kapitel werden Modelle vom ersten Typ behandelt; Netzwerk-Modelle sind Gegenstand von Kapitel 8.

Am Anfang steht ein bereits 1965 von Hess et al. [50] vorgeschlagenes Modell für die Zuordnung von KGE zu Zentren, das im Laufe der Jahre immer wieder verwendet und weiterentwickelt wurde. Da es sich um ein lineares Programm mit sehr vielen binären Entscheidungsvariablen handelt, hat es sich durchgesetzt, eine lineare Relaxation zu lösen und durch Runden zu einer ganzzahligen Lösung zu kommen. Wir zeigen in 7.2.4 und 7.2.5, daß mit gleichmäßiger Baumzerlegung das Rundungsproblem im optimalen Sinn gelöst werden kann. Daraus ergibt sich auch eine einfache Abschätzung für die maximale Abweichung der Bezirksgrößen vom Mittel, die beim Runden in Kauf genommen werden muß (Satz 7.13).

Berechnungsbeispiele. An verschiedenen Stellen werden in diesem und den folgenden Kapiteln Rechenverfahren durch Berechnungsbeispiele mit GAMOR veranschaulicht. Dabei werden reale Karten (Frankreich, Deutschland) zugrundegelegt; ein

Erzeugen „zufälliger Karten“ wurde nicht für sinnvoll gehalten.¹

In diesen Beispielen sind die Größenattribute für die KGE proportional zu ihrer Fläche festgelegt. Wäre die Bevölkerungsverteilung völlig homogen, würden so Bezirke mit (nahezu) Bevölkerungsgleichheit generiert. Doch ist nicht die (unzutreffende) Homogenitäts–Annahme für uns entscheidend. Motivation für die Festlegung der KGE–Größenattribute proportional zur Fläche ist vielmehr die Tatsache, daß sich auf diese Weise die Balance der Bezirksgrößen optisch wiedergeben läßt; durch die Bildung von Bezirken nahezu gleicher Fläche.

Den Berechnungsbeispielen liegen somit keine Datensätze aus der Praxis zugrunde. Es wird jedoch deutlich, daß die Berechnung von Gebietsaufteilungen, die auf solchen Daten beruhen, mit GAMOR ohne weiteres möglich ist.

Zufällig erzeugte Testprobleme. Für einige Untersuchungen war die Bereitstellung einer ausreichend großen Anzahl von (zufälligen) Testproblemen erforderlich. Instanzen solcher Testprobleme wurden folgendermaßen erzeugt: Die KGE–Orte wurden zufällig gleichverteilt als Punkte im Quadrat bestimmt. Die Größenattribute w_v der KGE wurden, wie in 6.2.1.2 beschrieben, auf zwei Arten zufällig gezogen: gleichverteilt beziehungsweise logarithmisch–normalverteilt. Als Nachbarschaftsgraph der KGE wurde der sog. Gabriel–Graph der KGE–Orte gewählt; in diesem Graph sind zwei Orte adjazent, falls der Kreis, dessen Durchmesser die Verbindungsstrecke dieser Orte ist, keine weiteren Orte enthält (siehe z.B. Gordon [45, S. 19] für Eigenschaften des Gabriel–Graphen).

7.1 Ein klassisches Zuordnungsmodell für die Gebietsaufteilung

7.1.1 KGE und Zentren als Punkte der Ebene

Zunächst machen wir einige Vorüberlegungen, die mit der Repräsentation von KGE und Zentren als Punkte der Ebene einhergehen.

Sei V die Menge der KGE. Jede KGE $v \in V$ wird in einem Punkt mit den Koordinaten

$$(ost_v, nord_v) \in \mathbb{R}^2$$

lokalisiert angenommen. Dies kann zum Beispiel die Lage der bedeutendsten Kommune in der KGE sein.

¹Es wäre auch zu fragen, was ein geeignetes Modell solcher Karten sein könnte. Was sind wesentliche Kenngrößen einer Karte von KGE–Umrißdaten? Was wäre eine angemessene Wahrscheinlichkeitsverteilung für diese Größen?

Wenn jedoch keine Information über die geographische Struktur innerhalb der KGE vorliegt, kann ihr in folgender Weise ein punktförmiger Ort zugeschrieben werden. Dazu muß eine Umrißlinie der KGE in Form eines geschlossenen jordanischen Polygonzugs

$$K_v = ((o_1^v, n_1^v), \dots, (o_s^v, n_s^v), (o_{s+1}^v, n_{s+1}^v)) \quad \text{mit} \quad (o_{s+1}^v, n_{s+1}^v) = (o_1^v, n_1^v)$$

und

$$(o_l^v, n_l^v) \in \mathbb{R}^2 \quad l = 1, \dots, s$$

bekannt sein.² Eine plausible Wahl für den Ort von v ist der Schwerpunkt der von K_v umschlossenen Fläche. Er wird mittels der diskreten Integralformeln

$$ost_v = \frac{1}{3} \frac{\sum_{l=1}^s (o_l^v n_{l+1}^v - n_l^v o_{l+1}^v) (o_l^v + o_{l+1}^v)}{\sum_{l=1}^s (o_l^v n_{l+1}^v - n_l^v o_{l+1}^v)}$$

und

$$nord_v = \frac{1}{3} \frac{\sum_{l=1}^s (o_l^v n_{l+1}^v - n_l^v o_{l+1}^v) (n_l^v + n_{l+1}^v)}{\sum_{l=1}^s (o_l^v n_{l+1}^v - n_l^v o_{l+1}^v)}$$

berechnet. Die Möglichkeit einer solchen Schwerpunktberechnung ist in GAMOR realisiert, da hier KGE durch ihre Grenzlinie beschrieben sind (vgl. 3.3.2).

Die Zentren $I = \{1, \dots, p\}$ der (zu bildenden) Bezirke sind wie die KGE durch Punkte, mit Koordinaten

$$(Ost_i, Nord_i) \in \mathbb{R}^2, \quad i \in I,$$

charakterisiert.

Ausgehend von den Punkten lassen sich nun Entfernungen d_{iv} zwischen Zentren und KGE berechnen. Hierfür gibt es verschiedene Vorschläge in der Literatur. Wir besprechen kurz die wichtigsten.

Euklidische Distanzen. Die Entfernung zwischen Zentrum $i \in I$ und KGE $v \in V$ ist

$$d_{iv} = \left((Ost_i - ost_v)^2 + (Nord_i - nord_v)^2 \right)^{1/2}.$$

Dies entspricht der Luftlinienentfernung und stellt eine plausible Art der Entfernungsmessung dar (Cloonan [25]).

²Ein Polygonzug ist jordanisch, wenn sich die Verbindungsstrecken der Punkte des Polygonzugs nicht schneiden. (Außer in den gemeinsamen Endpunkten benachbarter Strecken.)

Quadrierte euklidische Distanzen. Hierbei ist die Formel zur Berechnung der Distanz

$$d_{iv} = (Ost_i - ost_v)^2 + (Nord_i - nord_v)^2.$$

Diese Art der Entfernungsmessung erscheint zunächst merkwürdig, da sie kaum als tatsächliche Entfernung interpretiert werden kann. Dennoch wird sie von vielen Autoren verwendet (Fleischmann und Paraschis [35], George et al. [42], Hess et al. [50], Hess und Samuels [51], Hojati [52], Marlin [68]). Nach Einschätzung des Verfassers gibt es dafür zwei Gründe.

1. In location–allocation Modellen ist der location Schritt besonders einfach umzusetzen, wenn mit quadriert euklidischen Distanzen gearbeitet wird (vgl. 9.1.1.1). Diese Begründung wird in der Literatur häufig zugunsten dieser Art der Distanzmessung vorgebracht.
2. Ebenso wichtig erscheint aber eine Beobachtung, die in 7.3.2 herausgearbeitet wird, und die auch im allocation Schritt für die Verwendung von quadriert euklidischen Distanzen spricht. In diesem Fall nämlich wird die Ebene in Gebiete aufgeteilt, die konvexen Polygonen entsprechen und damit eine Tendenz zur Bildung von Bezirken, die der intuitiven Vorstellung von kompakter Form entsprechen, geschaffen.

7.1.2 Das Zuordnungsmodell nach Hess et al.

Bereits 1965 wurde von Hess et al. [50] ein lineares Programm vom Typ der Zuordnungsmodelle zur Gewinnung kompakter und balancierter Bezirke aus einer Menge von KGE vorgeschlagen.

Die Zuordnung von KGE zu Zentren wird mit Hilfe der Variablen

$$x_{iv} = \begin{cases} 1 & \text{KGE } v \text{ wird Zentrum } i \text{ zugeordnet,} \\ 0 & \text{sonst} \end{cases}$$

beschrieben. Das Modell lautet^{3,4}

$$(ZP) \quad \min \sum_{i,v} d_{iv} w_v x_{iv} \quad (7.1a)$$

$$\text{unter} \quad \sum_{i \in I} x_{iv} = 1 \quad (v \in V) \quad (7.1b)$$

$$\mu(1 - \tau) \leq \sum_{v \in V} w_v x_{iv} \leq \mu(1 + \tau) \quad (i \in I) \quad (7.1c)$$

$$x_{iv} \in \{0, 1\} \quad (v \in V, i \in I) \quad (7.1d)$$

Jede KGE muß genau einem Zentrum zugeordnet werden, dies wird durch die Nebenbedingungen (7.1b) abgesichert.

Da w_v die Größe der KGE v ist (hinsichtlich des anwendungsspezifischen Attributs), ergibt sich die Größe des Zentrum i entsprechenden Bezirks als $\sum_{v \in V} w_v x_{iv}$. Das Ziel balancierter Bezirksgrößen wird mit den Nebenbedingungen (7.1c) modelliert. Jeder Bezirk muß eine Größe haben, die innerhalb einer maximal erlaubten relativen Abweichung $\tau > 0$ (der sogenannten *Toleranz*) von der mittleren Größe $\mu = \sum_{v \in V} w_v / p$ liegt.

Die Zielfunktion (7.1a) mißt die totale gewichtete Entfernung der KGE von den Zentren. Ihre Minimierung strebt danach, jede KGE einem möglichst nahegelegenen Zentrum zuzuordnen. Dadurch werden kompakte Bezirke gebildet, zumindest unter der Voraussetzung, daß die Zentren „sinnvoll“ im aufzuteilenden Gebiet verteilt sind.

Die Verwendung des 0–1–Programms ZP in praktischen Situationen wird durch die in aller Regel große Anzahl an Binärvariablen allerdings eingeschränkt. Bei einem Problem mit 500 KGE, die in 50 Bezirke zusammenzufassen sind, hat ZP bereits 25000 Binärvariablen; eine auch für moderne Solver erhebliche Größenordnung.⁵

³Für die Bezeichnungen der auftretenden Größen erinnern wir an 3.1.

⁴Das von Hess et al. [50, S. 1004] formulierte Modell ist umfangreicher als das hier dargestellte; es umfaßt noch Variablen zur Bestimmung der Lage der Zentren. Da wir diesen Aspekt gesondert in Kapitel 9 behandeln, beschränken wir uns hier auf den Zuordnungsteil des Modells.

⁵In GAMOR besteht die Möglichkeit, das Modell ZP (7.1) dem sehr leistungsfähigen Solver CPLEX, gegenwärtig in der Version 6.6, zur Bearbeitung zu übergeben. Nach Erfahrungen des Autors ist es durchaus möglich, damit Probleme von der im Text angesprochenen Größe in vertretbarer Zeit auf einem PC zu lösen.

Andererseits kommt es ebenfalls vor, daß CPLEX auch bei kleinen Problemen (z.B. 100 KGE) scheitert. Entweder wird keine zulässige Lösung gefunden, oder auch nach sehr langer Rechenzeit ist die Lücke zwischen unterer und oberer Schranke noch sehr groß; schließlich paßt der aufgebaute Suchbaum nicht mehr in den Speicher des Rechners. Dies tritt zum Beispiel bei ungünstiger Lage der Zentren oder bei (zu) klein gewählter Toleranz τ auf.

Problematisch aus Praxissicht ist somit, daß das Lösen von ZP mit einem Standard–Solver wie CPLEX kein robustes Verfahren darstellt. Die im weiteren Verlauf des Kapitels entwickelten Verfahren zur heuristischen Lösung von ZP haben diesen Nachteil nicht.

Hinzu kommt, daß eine maximal zulässige Abweichung τ vom Größenmittel vorgegeben werden muß. Dies kann schwierig sein, wenn sich ein solcher Wert nicht in natürlicher Weise aus der Anwendungssituation ergibt. Dann muß eine willkürliche Festsetzung von τ vorgenommen werden. Ideal wären perfekt balancierte Bezirke, die alle genau die Größe μ haben, doch ist dieses aufgrund der diskreten Natur des Zuordnungsproblems in der Regel nicht möglich. Auch bei Vorgabe eines zu kleinen τ kann ZP unlösbar sein; dabei ist a priori nicht klar, wo die Grenze liegt.

Aus diesen Gründen werden in der Literatur Modelle vorgeschlagen, die als Relaxationen von ZP betrachtet werden können:

1. Eine häufig gewählte Möglichkeit, zuerst bei Hess et al. [50] zu finden, ist die Relaxation der Ganzzahligkeitsbedingung (7.1d), gleichzeitig kann $\tau = 0$ gesetzt werden. Das resultierende lineare Programm ist ein effizient lösbares Transportproblem. Es erzeugt perfekt balancierte, kompakte Bezirke, wobei allerdings einige KGE nicht eindeutig einem Bezirk zugeordnet sind. Mit Rundungsverfahren wird dies nachträglich sicher gestellt. Diese Vorgehensweise ist Gegenstand von Abschnitt 7.2.
2. Eine anderer Weg, vorgeschlagen von Zoltners und Sinha [115], ist die Relaxation der Balance-Bedingungen (7.1c), indem sie mit Lagrange-Multiplikatoren in die Zielfunktion gebracht werden. Das resultierende 0–1-Programm ist trivial zu lösen und hat kompakte Bezirke als Ergebnis. Durch ein geeignetes iteratives Anpassen der Multiplikatoren, wobei viele Gebietsaufteilungen erzeugt werden, wird eine Aufteilung mit balancierten Bezirken bestimmt. Diesen Zugang zur heuristischen Lösung von ZP analysieren wir in 7.3.

7.2 KGE–Zentren–Zuordnung mittels Transportmodell und Splitauflösung

7.2.1 Das Transportmodell

Wir gehen aus vom ganzzahligen Programm ZP (7.1), setzen $\tau = 0$, und relaxieren die Ganzzahligkeitsbedingung. Damit erhalten wir das lineare Programm

$$(TP) \quad \min \sum_{i,v} d_{iv} w_v x_{iv} \quad (7.2a)$$

$$\text{unter} \quad \sum_{i \in I} x_{iv} = 1 \quad (v \in V) \quad (7.2b)$$

$$\sum_{v \in V} w_v x_{iv} = \mu \quad (i \in I) \quad (7.2c)$$

$$x_{iv} \geq 0 \quad (v \in V, i \in I) \quad (7.2d)$$

für die Zuordnung von KGE zu Zentren. Das Modell TP hat die Struktur des wohlbekannten Transportproblems. Zu seiner Lösung existieren effiziente Verfahren, Klassiker des Operations Research.⁶

Diese Eigenschaft von TP macht es möglich, auch sehr große Gebietsaufteilungsprobleme damit anzugehen. Zum Beispiel lösen Fleischmann und Paraschis [35] ein Problem mit 1400 KGE und 168 Zentren, bei George et al. [42] werden sogar über 35000 KGE zugeordnet; in beiden Studien wird TP bzw. ein sehr nahe verwandtes Modell eingesetzt.

Jedoch ist das Ergebnis von TP (siehe Abbildung 7.1) im allgemeinen keine zufriedenstellende Lösung des Gebietsaufteilungsproblems, denn nicht alle Variablen werden im Optimum ganzzahlig sein. Folglich gibt es KGE, die mehreren Zentren (anteilig) zugeordnet sind. Dies ist Anlaß zu folgender

Definition 7.1

Eine KGE $v \in V$, für die in der optimalen Lösung von TP mehrere Variablen x_{iv} , $i \in I$, positive Werte haben, heißt *Split*.

Aufgrund von Bedingung (7.2b) ist ein Split zwar vollständig zugeordnet, aber eben nicht eindeutig. Zum Beispiel ist er zu 40% einem Zentrum i und zu 60% einem Zentrum i' zugeordnet. Wir bezeichnen die Menge der Splits im folgenden mit V^S .

Um zu einer Gebietsaufteilung ohne Splits zu kommen, muß ein Verfahren auf die optimale Lösung von TP angewendet werden, das jeden Split *eindeutig* einem Zentrum zuordnet. Dabei kommen (nur) solche Zentren in Betracht, denen bereits

⁶Als neuere Quelle sei zum Beispiel Dantzig und Thapa [27] genannt.

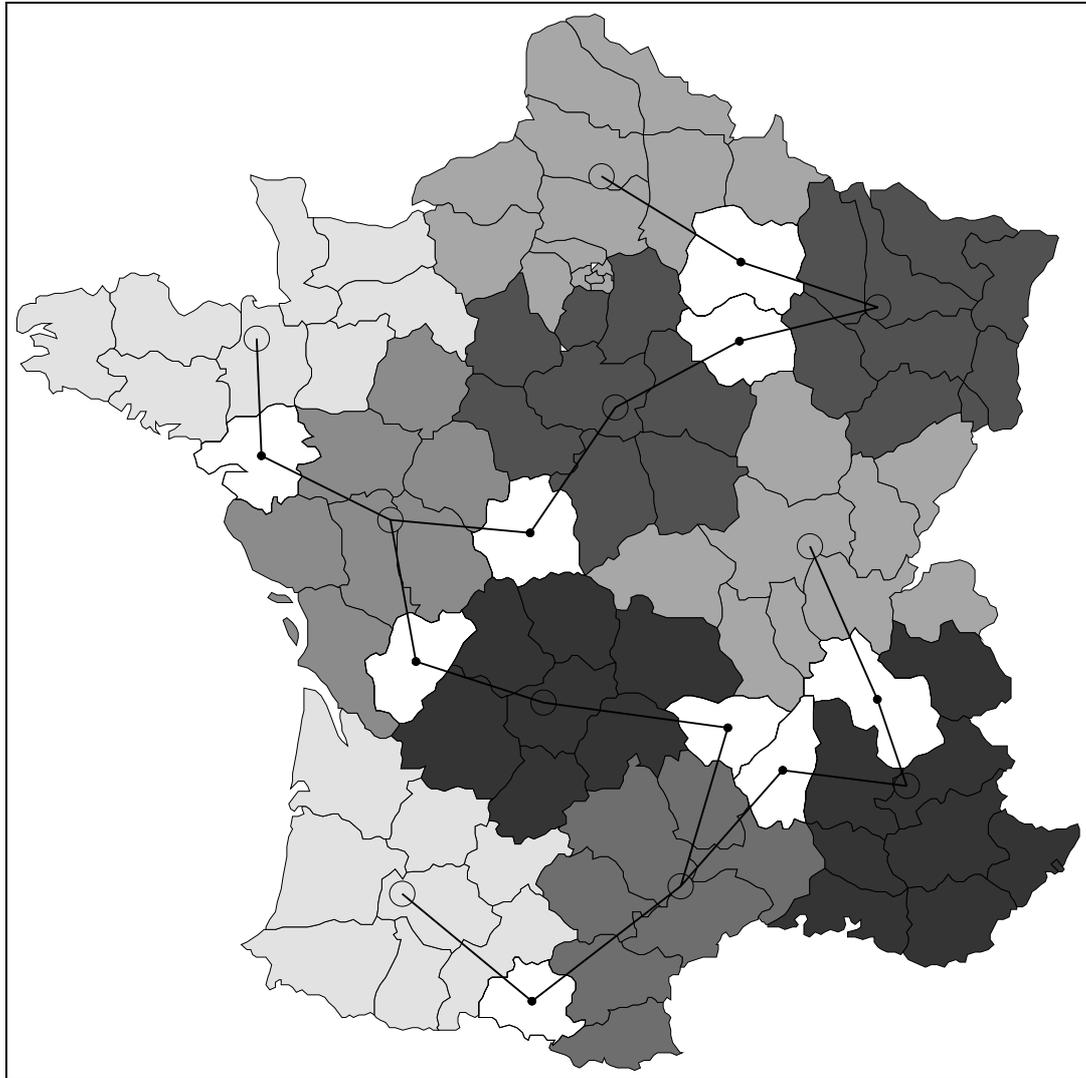


Abbildung 7.1: Lösung von TP (7.2). Die eingezeichneten Verbindungslinien entsprechen nichtganzzahligen Variablen. *Splits* (vgl. Def. 7.1) sind weiß dargestellt.

ein Teil des jeweiligen Splits zugeordnet ist. Auf diese Weise wird die durch das Lösen von TP gewonnene Information einbezogen. Algebraisch gesprochen, werden die nichtganzzahligen Variablen in der Optimallösung von TP auf ganzzahlige Werte gerundet.

7.2.2 Splitauflösung und Splitadjazenz

Den Prozeß der eindeutigen Zuordnung der Splits, der sich an das Optimieren von TP anschließt, nennen wir *Splitauflösung*. Er wird in diesem und den folgenden Abschnitten eingehend studiert. Hier stellen wir die Frage, welche Gegebenheiten bei der Splitauflösung zu berücksichtigen sind.

Zunächst interessiert, wieviele Splits überhaupt vorkommen können. Das folgende, in der Literatur oft angeführte Ergebnis zeigt, daß es stets weniger Splits als Bezirke gibt.

Proposition 7.2

In einer optimalen Basislösung von TP ist die Anzahl an Splits höchstens $p - 1$.

Zur Voraussetzung der Proposition ist folgendes zu sagen: Wenn die optimale Lösung von TP eindeutig ist, entspricht sie einer Ecke des durch die Nebenbedingungen (7.2b) bis (7.2d) beschriebenen Polyeders im $\mathbb{R}^{|V||I|}$ und ist somit Basislösung. Die üblichen Verfahren zur Lösung von TP, wie zum Beispiel der Simplex-Algorithmus, liefern jedoch auch im Falle eines mehrdeutigen Optimums eine Basislösung.

Eine Basislösung ist dadurch charakterisiert, daß die Spalten der Nebenbedingungsmatrix des linearen Programms TP zu Variablen mit positivem Wert linear unabhängig sind. Dies hat die folgende, wohlbekannte Interpretation: Der Subgraph des bipartiten Graphen

$$G = (I \cup V, I \times V), \quad (7.3)$$

dessen Kanten den Variablen mit positivem Wert (in einer Basislösung von TP) entsprechen, enthält keinen Zyklus.⁷ Um Proposition 7.2 zu beweisen, betrachten wir denjenigen Subgraphen von G , dessen Kanten den gebrochenen Variablen entsprechen.

Definition 7.3

Gegeben sei eine optimale Basislösung (x_{iv}) von TP. Der bipartite Graph $T^S = (U^S, E^S)$ mit $U^S = I \cup V^S$ und

$$E^S = \{(i, v) : 0 < x_{iv} < 1\}$$

heißt *Splitadjazenz*.

⁷Andernfalls könnte man die Spalten der Nebenbedingungsmatrix von TP, die den Zykluskanten entsprechen, durch Summierung mit alternierenden Vorzeichen zum Nullvektor kombinieren.

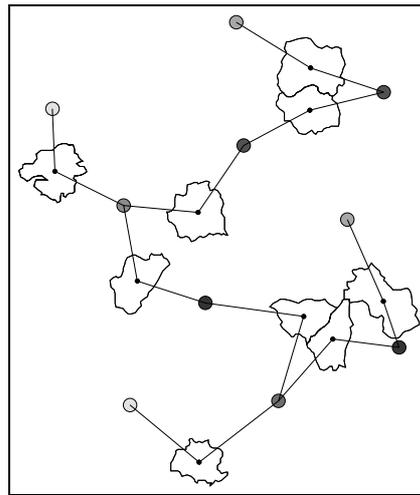


Abbildung 7.2: Splitadjazenz zum Beispiel aus Abbildung 7.1.

In Abbildung 7.2 wird diese Definition veranschaulicht.

Beweis von Proposition 7.2. Jede Kante von T^S inzidiert mit genau einem Split und jeder Split hat mindestens einen Grad von zwei in T^S . Daraus folgt $|E^S| \geq 2|V^S|$. Da T^S zyklenfrei ist, gilt $|E^S| \leq |I| + |V^S| - 1$. Somit ist $|V^S| \leq |I| - 1 = p - 1$. \square

Proposition 7.2 ist somit eine triviale Konsequenz aus der Struktur des linearen Programms TP. Ihr Nutzen wird deutlich, wenn die Splitauflösung durch einen menschlichen Entscheider „von Hand“ vorgenommen wird. Es ist gesichert, daß in der Regel nur ein kleiner Teil der KGE betroffen ist, die Aufgabe also überschaubar ist.

Auf der anderen Seite wurden für die Splitauflösung Heuristiken vorgeschlagen, die sich auf dem Rechner ausführen lassen.

7.2.3 Eine einfache Heuristik zur Splitauflösung

In der Lösung von TP ist die Größe jedes Bezirks exakt gleich der mittleren Größe μ , wenn die zugeordneten Anteile der Splits in der Bezirksgröße berücksichtigt werden. Deshalb ist eine unausweichliche Konsequenz jedes Verfahrens zur Splitauflösung, daß Abweichungen von der mittleren Größe in Kauf genommen werden müssen.

Auf der anderen Seite ist es aufgrund der Balance-Bedingung wünschenswert, daß diese Abweichungen klein sind. Die folgende Idee einer Heuristik zur Splitauflösung liegt daher auf der Hand:

Ordne jeden Split demjenigen Zentrum zu, zu welchem in der optimalen Lösung von TP der größte Anteil dieses Splits gehört.

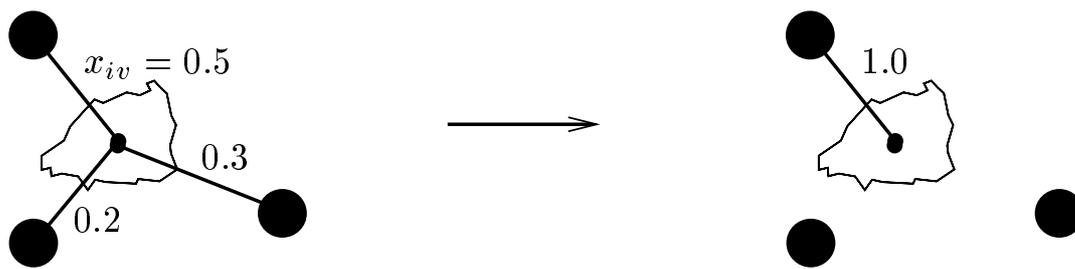


Abbildung 7.3: Heuristik AssignMAX zur Splitauflösung.

Abbildung 7.3 veranschaulicht diese Heuristik; in Pseudo-Code Darstellung lautet sie wie folgt.

Algorithmus 7.1 Heuristik AssignMAX zur Splitauflösung

Input: optimale Basislösung (x_{iv}) von TP

- 1: **for all** $v \in V^S$ **do**
 - 2: $i^* := \arg \max\{x_{iv} : i \in I\}$
 - 3: ordne KGE v Zentrum i^* zu
 - 4: **end for**
-

Der Vorschlag, Splitauflösung mittels AssignMAX vorzunehmen, wurde bereits 1971 von Hess und Samuels [51] gemacht. Auch in der kürzlich erschienenen Arbeit von George et al. [42] kommt diese Heuristik zum Einsatz.

Naheliegender ist die Frage, inwieweit mit AssignMAX das Ziel erreicht wird, die Abweichung der Bezirksgrößen vom Mittel μ klein zu halten. Im folgenden Beispiel 7.1 resultiert die Anwendung von AssignMAX in einer größeren Abweichung, als der minimal möglichen. Darüberhinaus zeigt das Beispiel, daß der Quotient zwischen diesen Abweichungen beliebig groß werden kann. Somit gilt der

Satz 7.4

Die Heuristik AssignMAX hat keine endliche Garantie für das Ziel der Minimierung der maximalen Abweichung der Bezirksgrößen von ihrem Mittel μ .

Beispiel 7.1

Der Input von AssignMAX ist eine optimale Basislösung des linearen Programms TP. Wir konstruieren direkt diese Lösung, ein dazu „passendes“ Programm TP läßt sich leicht angeben.

Eine Basislösung von TP ist charakterisiert durch die Spezifikation

- einer Splitadjazenz $T^S = (I \cup V^S, E^S)$,
- der Größe w_v , $v \in V^S$, der Splits, und
- der Größe W_i , $i \in I$, der Bezirke *ohne* ihre jeweiligen Anteile an den Splits.

Daraus ergibt sich die mittlere Größe der Bezirke zu

$$\mu = \frac{1}{|I|} \left(\sum_{v \in V^S} w_v + \sum_{i \in I} W_i \right)$$

und die Variablen x_{iv} , $(i, v) \in E^S$, haben eindeutig bestimmte Werte aufgrund der Zyklenfreiheit von T^S .

In unserem Beispiel wählen wir für T^S die Struktur eines d -Baumes (siehe 4.1.1.2). Dabei ist $d \geq 4$ beliebig. Die Eckenmenge sei gegeben durch

$$\begin{aligned} V^S &:= \{1, \dots, d\}, \\ I &:= \{0, 1^1, \dots, 1^{d-1}, \dots, \dots, d^1, \dots, d^{d-1}\}, \end{aligned}$$

die Kantenmenge durch

$$E^S := \{(v, 0) : v \in V^S\} \cup \{(v, v^j) : v \in V^S, j = 1, \dots, d-1\}.$$

(Für $d = 4$ ist so der auf Seite 53 rechts gezeigte d -Baum gegeben.) Die Größe von KGE und Bezirken setzen wir zu

$$w_v := d + 1, v \in V^S, \quad W_0 := 1, \quad W_{v^j} := 2d, v^j \in I - \{0\}.$$

Daraus ergibt sich eine mittlere Größe von $\mu = 2d + 1$. Die Bezirke zu den Ecken v^j in T^S sind um eins kleiner als μ , somit ist

$$x_{v^j, v} = \frac{1}{w_v} = \frac{1}{d + 1}.$$

Da v zu $d - 1$ solcher Ecken adjazent ist, gilt für die verbleibende Kante $(0, v)$

$$x_{0, v} = 1 - (d - 1) \frac{1}{d + 1} = \frac{2}{d + 1}.$$

AssignMAX ordnet daher jeden Split $v \in V^S$ dem Zentrum 0 zu. Der zugehörige Bezirk hat danach die Größe $1 + d(d + 1)$, die Abweichung vom Mittel ist

$$|1 + d(d + 1) - (2d + 1)| = d^2 - d.$$

Ordnet man dagegen alle Splits einem von 0 verschiedenen (adjazenten) Zentrum zu (einer Ecke also), erhält man einerseits Bezirke der Größe $3d + 1$ und andererseits solche der Größen $2d$ bzw. 1. Damit ist die maximale Abweichung vom Mittel gleich

$$\max \{|3d + 1 - (2d + 1)|, |2d - (2d + 1)|, |1 - (2d + 1)|\} = \max \{d, 1, 2d\} = 2d.$$

Für $d \geq 4$ ist dies kleiner, als die durch AssignMAX produzierte maximale Abweichung. Ferner kann der Quotient $(d^2 - d)/2d$ beliebig groß werden. \square

Die Beispiel 7.1 zugrundeliegende Optimalbasis von TP ist recht artifizuell und ähnliches wird kaum aus einem praktischen Gebietsaufteilungsproblem resultieren. Insbesondere ist das starke Anwachsen der Eckengrade in T^S in der Praxis nicht zu erwarten. Somit zeigt das Beispiel eher die theoretischen Grenzen der Heuristik AssignMAX auf. Bei kleinen Eckengraden ergibt sich aus dem folgenden Satz gegebenenfalls eine nützliche Abschätzung der maximalen Abweichung von μ , die aus der Anwendung von AssignMAX resultiert.

Satz 7.5 (worst case für AssignMAX)

Sei $\bar{\delta}_V = \max \{\text{grad}(v) : v \in V^S\}$, $\bar{\delta}_I = \max \{\text{grad}(i) : i \in I\}$ (Grade bezogen auf T^S) und $\bar{w} = \max \{w_v : v \in V^S\}$. Dann ist die mit AssignMAX erzielte maximale Abweichung der Bezirksgrößen von μ nicht größer als $(1 - 1/\bar{\delta}_V)\bar{\delta}_I\bar{w}$.

Beweis. Sei i ein Zentrum. Es gilt

$$\mu = \sum_{v \in V} w_v x_{iv} = \sum_{v \in V^S} w_v x_{iv} + \sum_{\substack{v \in V - V^S \\ x_{iv} = 1}} w_v.$$

AssignMAX ordne einen Split $v \in V^S$ dem Zentrum i zu. Dann ist

$$x_{iv} = \max \{x_{i'v} : i' \in I, x_{i'v} > 0\}.$$

Die Anzahl der positiven $x_{i'v}$, $i' \in I$, ist gleich $\text{grad}(v)$ in T^S . Damit gilt $x_{iv} \geq 1/\text{grad}(v)$. Durch das Zuordnen von v zu i kommt somit höchstens $w_v(1 - 1/\text{grad}(v))$ zur Bezirksgröße von i hinzu. Daher übersteigt die maximale Bezirksgröße μ um nicht mehr als $(1 - 1/\bar{\delta}_V)\bar{\delta}_I\bar{w}$.

Auf der anderen Seite ist x_{iv} auf jeden Fall nicht größer als $1/2$, wenn der Split v dem Zentrum i nicht zugeordnet wird. Somit ist die minimale Bezirksgröße höchstens $(1/2)\bar{\delta}_I\bar{w}$ kleiner als μ . Diese Abweichung ist nicht größer als $(1 - 1/\bar{\delta}_V)\bar{\delta}_I\bar{w}$, da $\bar{\delta}_V \geq 2$. \square

Bemerkung 7.6

Für das Beispiel 7.1 ergibt die Abschätzung in Satz 7.5 eine obere Schranke von $(1 - 1/d)d(d + 1) = d^2 - 1$. Die tatsächlich von AssignMAX produzierte Abweichung ist gleich $d^2 - d$. Dies zeigt, daß das Beispiel (nahezu) einen *worst case* für AssignMAX darstellt.

7.2.4 Optimale Splitauflösung durch Baumzerlegung

Die Heuristik AssignMAX des letzten Abschnitts ist ein Verfahren zum Auflösen der aus der Lösung von TP resultierenden Splits. Sie dient dem Ziel, die (unvermeidbaren)

Abweichungen der Bezirksgrößen von ihrem Mittelwert klein zu halten. Das von AssignMAX verwendete Kriterium stellt jedoch nur eine lokale Betrachtungsweise des Splitauflösungsproblems dar.

Um zum Ziel einer optimalen Splitauflösung zu kommen, ist es nötig, sich vor Augen zu halten, daß die Entscheidung, welchem adjazenten Bezirk ein bestimmter Split zugeordnet wird, durchaus Einfluß darauf hat, in welcher Weise andere Splits aufgelöst werden sollten. Es ist daher notwendig, diese gegenseitigen Abhängigkeiten in Betracht zu ziehen.

In 7.2.2 wurde mit der Splitadjazenz bereits die hierfür wesentliche Struktur definiert. Splits und adjazente Zentrum bilden zusammen einen bipartiten, zyklensfreien Graphen (siehe Definition 7.3). Unser Ziel ist es nun, Verfahren zur optimalen Splitauflösung anzugeben, die auf der Struktur der Splitadjazenz beruhen. Dabei wird das in Teil II entwickelte Material zur gleichmäßigen Baumzerlegung angewendet.

Die grundlegende Idee ist einfach. Jede Möglichkeit, die Splits (in einer beliebigen Weise) aufzulösen, entspricht eineindeutig einer Baumzerlegung der Splitadjazenz in Teilbäume, die jeweils genau ein Zentrum enthalten. Dabei gehen wir davon aus, daß die Splitadjazenz baumförmig ist, daß also neben der Zyklensfreiheit auch Zusammenhang gegeben ist. (Durch Einfügen geeigneter Kanten kann diese Voraussetzung stets gesichert werden.) Das Gewicht der Teilbäume in der Zerlegung entspricht der Größe der Bezirke nach der Splitauflösung. Die Zielsetzung, möglichst gleichgroße Bezirke zu erzielen, entspricht der Zielsetzung der gleichmäßigen Baumzerlegung im Sinne von Abschnitt 4.2.

Die optimale Lösung eines solchen Baumzerlegungsproblems berücksichtigt implizit alle Möglichkeiten der Zerlegung des Baumes. Daraus folgt: Wird Splitauflösung vorgenommen, indem die Splitadjazenz in einer optimalen Weise „gleichmäßig“ zerlegt wird, so werden dabei implizit *alle* Möglichkeiten der Splitauflösung berücksichtigt. Dies rechtfertigt die Sprechweise von einer *optimalen* Splitauflösung. Unter allen denkbaren Möglichkeiten, die Splits aufzulösen, wird diejenige gewählt, welche die größte Gleichmäßigkeit der Bezirksgrößen nach sich zieht. Hierbei ist jedoch klar, daß noch nicht präzisiert ist, was „größte Gleichmäßigkeit“ bedeutet — ebenso wie in 4.2 verschiedene Modelle der gleichmäßigen Baumzerlegung eingeführt wurden.

7.2.4.1 Gleichmäßige Baumzerlegung der Splitadjazenz

Wir gehen im folgenden davon aus, daß die Splitadjazenz $T^S = (U^S, E^S)$ (siehe Definition 7.3) ein Baum ist.⁸ In diesem sind durch $U^S = I \cup V^S$ zwei verschiedene

⁸Der Fall, daß T^S nicht zusammenhängend ist, kann durchaus eintreten. (Er ist genau dann gegeben, wenn die optimale Basislösung von TP degeneriert ist.) Jedoch stellt dies kein ernsthaftes Problem dar, denn es gibt zwei einfache Möglichkeiten, damit umzugehen. Die erste Möglichkeit ist,

„Typen“ von Ecken gegeben, die den Zentren beziehungsweise den Splits entsprechen (vgl. Abb. 7.2). Wir betrachten Zerlegungen von T^S in Teilbäume, die jeweils genau ein Zentrum enthalten und bezeichnen solche Zerlegungen als *zulässig*. Ist $\pi = \{T'_i \triangleleft T^S : i \in I\}$ eine zulässige Zerlegung, so enthält Teilbaum T'_i gerade das Zentrum $i \in I$. Außerdem enthält T'_i eine (eventuell leere) Teilmenge der dazu adjazenten Splits. Auf diese Weise ist jeder Split genau einem (adjazenten) Zentrum zugeordnet, mit anderen Worten ist durch π das Resultat einer Splitauflösung gegeben.

Andererseits ist klar, daß umgekehrt jedes solche Resultat durch eine geeignete Zerlegung π charakterisiert werden kann. Wir halten daher fest:

Proposition 7.7 (Splitauflösung durch Baumzerlegung)

Jedes Resultat einer Splitauflösung entspricht eineindeutig einer zulässigen Zerlegung von T^S .

Aus einer Splitauflösung resultiert die Größe der entstehenden Bezirke. Der durch Zentrum i gegebene Bezirk hat die Größe

$$\sum_{\substack{v \in V - V^S \\ x_{iv} = 1}} w_v \quad + \quad \text{Größe aller zugeordneten Splits.}$$

Der erste Summand hängt nicht von der Splitauflösung ab, es ist die Größe aller KGE, die bereits durch Lösen von TP dem Zentrum i eindeutig zugeordnet wurden. Wir geben daher jeder Ecke $i \in I$ von T^S das Gewicht

$$W_i := \sum_{\substack{v \in V - V^S \\ x_{iv} = 1}} w_v. \tag{7.4}$$

Sei weiter

$$W_v := w_v \tag{7.5}$$

das Gewicht der Split-Ecken $v \in V^S$ (was lediglich einer einheitlichen Notation mit großgeschriebenem W dient). Dadurch haben wir T^S zu einem eckengewichteten Baum gemacht und es ist nun klar, daß der folgende Zusammenhang besteht:

Proposition 7.8

Das Gewicht jedes Teilbaumes in einer zulässigen Zerlegung π von T^S ist die Größe des ihm entsprechenden Bezirkes nach der durch π repräsentierten Splitauflösung.

auf jede Zusammenhangskomponente (dies sind Bäume) getrennt ein Verfahren der gleichmäßigen Baumzerlegung anzuwenden, die zweite ist das Einfügen von Kanten (i, v) des Graphen (7.3) in den Wald T^S , für die $x_{iv} = 0$ gilt. Hierbei werden gerade so viele Kanten eingefügt, daß ein Baum entsteht, wobei Kanten mit kleinem Wert $d_{iv}w_v$ zu bevorzugen sind. (Zum Beispiel kann man die den Basisvariablen mit Wert 0 entsprechenden Kanten wählen.)

Das Ziel balancierter Bezirke bedeutet daher, daß eine zulässige Zerlegung von T^S zu bestimmen ist, in welcher die Gewichte WT'_i , $i \in I$, dem mittleren Gewicht μ möglichst nahe kommen.⁹ Daher können wir auf die Modelle zur gleichmäßigen Baumzerlegung aus Teil II zurückgreifen. Die folgende Tabelle gibt die wesentlichen Zusammenhänge wieder.

Zielsetzung für optimale Splitauflösung. Erziele gleichmäßige Bezirksgrößen durch Minimierung der...	Modell für die gleichmäßige Baumzerlegung der Splitadjazenz (siehe 4.2.2)
totalen Abweichung (von der mittleren Größe)	$T/2\text{-Zent}/\cdot/\ \cdot\ _1$
Standardabweichung	$T/2\text{-Zent}/\cdot/\ \cdot\ _2$
maximalen Abweichung (von der mittleren Größe)	$T/2\text{-Zent}/\cdot/\ \cdot\ _\infty$

Tabelle 7.1: Optimale Splitauflösung durch gleichmäßige Baumzerlegung

Auch andere Zielsetzungen sind denkbar, doch sind die wesentlichen hier aufgeführt. Insbesondere der Minimierung der maximalen Abweichung kommt für die Praxis große Bedeutung zu.

7.2.4.2 Komplexität optimaler Splitauflösung

Aufgrund des in Tabelle 7.1 dargestellten Zusammenhangs lassen sich nun in Teil II entwickelte Ergebnisse direkt übertragen. Dies betrifft insbesondere Komplexitätsergebnisse aus 4.3.

Korollar 7.9 (Komplexität der optimalen Splitauflösung)

Optimale Splitauflösung zur Minimierung der totalen Abweichung der Bezirksgrößen von ihrem Mittel ist \mathcal{NP} -schwer. Dasselbe gilt für die Minimierung der Varianz der Bezirksgrößen.

Das Korollar folgt im wesentlichen aus Satz 4.17. Dabei ist jedoch zu beachten, daß nicht einfach die Aussage übertragen werden kann. Der Grund ist, daß nicht alle Instanzen von $T/2\text{-Zent}/\cdot/\|\cdot\|_\rho$ der gleichmäßigen Zerlegung einer Splitadjazenz (in der die Eckengewichte nicht beliebig gewählt werden können), entsprechen. Jedoch überzeugt man sich leicht, daß die in der Transformation zum Beweis von Satz 4.17 verwendete Problem Instanz Eckengewichte aufweist, die eine Interpretation als Splitadjazenz erlauben.

⁹Man beachte die leicht ersichtliche Beziehung $\mu = \sum_{v \in V} w_v / p = WT^S / p$, $p = |I|$, das heißt, die mittlere Bezirksgröße stimmt mit dem mittleren Gewicht überein.

Korollar 7.10 (Komplexität der optimalen Splitauflösung)

Splitauflösung zur Minimierung der maximalen Abweichung der Bezirksgrößen von ihrem Mittel ist ein polynomial lösbares Problem.

Dieses Ergebnis folgt direkt aus Satz 4.28. Er wird durch die Verfahren aus Kapitel 5 bewiesen, wobei an dieser Stelle vor allem Algorithmus 5.6 bedeutsam ist, denn er behandelt Bäume mit Zentren. Seine Komplexität ist $O(p^2 n^4)$ (Satz 5.22), sie geht direkt in die Aussage von Satz 5.26 ein, der den Lösungsaufwand für $T/Zent/\cdot/\|\cdot\|_\rho$ durch $O(p^2 n^4 \log wT)$ abschätzt. Hierbei ist n die Anzahl aller Ecken und p die Anzahl der Zentren.

Die Splitadjazenz T^S hat höchstens $n = 2p - 1$ Ecken und p Zentren, somit resultiert eine Komplexität von $O(p^6 \log wV)$ der optimalen Splitauflösung zur Minimierung der maximalen Abweichung.

Zunächst ist die Garantie der Polynomialität ein wichtiges Resultat, aber die Abschätzung durch ein Polynom sechster Ordnung kann für praktische Fragestellungen nicht befriedigen. Doch ist hier Spielraum zur Verbesserung gegeben, denn dieses für ein gleichmäßiges Baumzerlegungsproblem mit Zentren hergeleitete Ergebnis läßt zwei Eigenschaften der Splitadjazenz außer acht.

1. Die Zentren sind in der Splitadjazenz nicht beliebig angeordnet. Durch die Tatsache, daß T^S Subgraph eines bipartiten Graphen ist, wechseln sich Ecken vom Zentren- und vom Split-Typ in T^S gewissermaßen ab.
2. Die Eckengrade in der Splitadjazenz sind in praktischen Problemstellungen klein, sie werden kaum fünf oder sechs übersteigen.

Im folgenden Abschnitt verfeinern wir daher die Ideen aus Teil II, um zu praktikablen Verfahren für die optimale Splitauflösung zu kommen.

7.2.5 Praktikable Verfahren zur optimalen Splitauflösung

Unser Ziel ist die Entwicklung von Verfahren, die die soeben angeführten Eigenschaften der Splitadjazenz ausnutzen. Einige Vorbemerkungen mögen diesen längeren Abschnitt motivieren.

1. Es werden Ideen des Teils II zur gleichmäßigen Baumzerlegung auf die spezielle Struktur des Splitadjazenz angepaßt. Dies bedingt einen etwas breiten Formalismus und gibt dem folgenden einen technischen Charakter. Auf der Ideenebene kommt jedoch nichts Neues hinzu.
2. Das Ergebnis wird ein sehr allgemeines Verfahren (Algorithmus 7.2) zur optimalen Splitauflösung sein, das sich für die in Tabelle 7.1 aufgeführten Ziele

einsetzen läßt; sich darüber hinaus jedoch auch für mehrkriterielle Fragestellungen eignet (siehe 7.2.5.4). In praktischen Fällen hat dieses Verfahren lineare Laufzeit (Bemerkung 7.12).

3. Wir erhalten eine schöne Abschätzung für die maximale Abweichung der Bezirksgrößen vom Mittel bei Verwendung des Verfahrens (Satz 7.13).
4. Speziell für die Minimierung der maximalen Abweichung, einem wichtigen Ziel in der Praxis, läßt sich durch Anwendung der Ideen aus 5.4 eine polynomiale Laufzeit beweisen; dies mit besserer Abschätzung als der nach Korollar 7.10 angegebenen (siehe 7.2.5.6).

7.2.5.1 Vorbemerkungen

Die Struktur der Splitadjazenz $T^S = (U^S, E^S)$ (siehe Definition 7.3) ist baumförmig; jede Ecke ist entweder Split oder Zentrum. Jede Kante verbindet einen Split mit einem Zentrum, und alle Endecken sind Zentren. Diese Eigenschaften von T^S geben Anlaß, den Algorithmus 4.1 für das allgemeine Baumzerlegungsproblem zu spezialisieren. Dabei wird vorausgesetzt, daß die zulässigen Teilbäume dadurch charakterisiert sind, daß sie genau eine Zentren–Ecke enthalten. Es ist also mit den Bezeichnungen von 4.1.2

$$\mathcal{T} = \{T' = (U', E') \triangleleft T^S : |U' \cap I| = 1\}.$$

Algorithmus 4.1 setzt voraus, daß eine Ecke als Wurzel des zu zerlegenden Baumes ausgezeichnet ist. Im Fall von T^S wählen wir eine Ecke $v_0 \in I$ als Wurzel, dies ist jedoch nur eine technische Vereinbarung.

Wir machen nun wie in 4.1.2 die Voraussetzung, daß die zulässigen Teilbäume mit „Kosten“ aus einem geordneten Monoid M (Definition 4.1) bewertet sind,

$$c : \mathcal{T} \rightarrow M.$$

Damit abstrahieren wir zunächst von der Zielsetzung einer gleichmäßigen Baumzerlegung der Splitadjazenz, wie sie im vorangehenden Abschnitt herausgearbeitet wurde. Der Nutzen wird sich später erweisen.

Die Kosten einer zulässigen Zerlegung $\pi = \{T'_i : i' \in I\}$ ergeben sich zu $c\pi = \sum_{i \in I} cT'_i$ (Addition in M), zu bestimmen ist eine Zerlegung minimaler Kosten. Wie in 4.1.3 wird dies durch dynamische Programmierung „von den Blättern zur Wurzel“ realisiert, hier berücksichtigen wir jedoch die spezielle Struktur von T^S .

Notation. Zu einer Ecke $u \in U^S$ sei T_u der Teilbaum von T^S , der in u wurzelt. Da wir in den Rekursionsgleichungen vier Fälle zu unterscheiden haben, schreiben wir

für die Zwischenoptima hier nicht $C(u)$ wie in (4.2), sondern verwenden die folgende Notation (siehe Abbildung 7.4):

Für eine Zentrum–Ecke $u \in I$ seien die Kosten einer optimalen Zerlegung von T_u mit $\alpha(u)$ bezeichnet, für eine Split–Ecke $u \in V^S$ dagegen mit $\beta(u)$. Sei $u \in I$. Die Kosten einer optimalen Zerlegung des Teilbaumes, der aus T_u und dem Vorgänger $\text{pre}(u)$ besteht, bezeichnen wir mit $\alpha'(u)$. Ist dagegen $u \in V^S$, so bezeichnen wir mit $\beta'(u)$ die Kosten einer optimalen Zerlegung des Waldes, der aus T_u durch Entfernen der Ecke u resultiert.

Da ein zulässiger Teilbaum $T' = (U', E')$ von T^S durch seine Ecken bestimmt ist, verwenden wir ferner die Notation $c(U')$ für cT' .

7.2.5.2 Rekursionsgleichungen für die optimale Zerlegung von T^S

Unser Ziel ist die Entwicklung von Rekursionsgleichungen für α, α' und β, β' . Alle Blätter von T^S sind Zentren–Ecken. Ist u ein solches Blatt, so ist offenbar

$$\alpha(u) = c(\{u\}) \quad \text{und} \quad \alpha'(u) = c(\{u, \text{pre}(u)\}). \quad (7.6)$$

Ist dagegen $u \in I$ kein Blatt, so setzen sich die Eckenmengen aller zulässigen, u umfassenden, Teilbäume zusammen aus u , einer Teilmenge der Söhne von u und gegebenenfalls der Ecke $\text{pre}(u)$. Ist ein Sohn $u' \in S_u$ in einem solchen Teilbaum nicht enthalten, so ist der in u' wurzelnde Baum $T_{u'}$ optimal zu zerlegen, durch $\beta(u')$ sind die entsprechenden Kosten gegeben. Ist u' jedoch im Teilbaum enthalten, so ist der Wald $T_{u'} - \{u'\}$ zu zerlegen, mit optimalen Kosten $\beta'(u')$. Die Rekursionsgleichung für $\alpha(u)$ lautet also

$$\alpha(u) = \min_{S' \subseteq S_u} c(\{u\} \cup S') + \sum_{u' \in S'} \beta'(u') + \sum_{u' \in S_u - S'} \beta(u'). \quad (7.7)$$

Die Gleichung für $\alpha'(u)$ ist nahezu dieselbe, jedoch werden per Definition die Teilbäume berücksichtigt, die $\text{pre}(u)$ enthalten:

$$\alpha'(u) = \min_{S' \subseteq S_u} c(\{u, \text{pre}(u)\} \cup S') + \sum_{u' \in S'} \beta'(u') + \sum_{u' \in S_u - S'} \beta(u'). \quad (7.8)$$

Die Zwischenoptima $\alpha(u)$ und $\alpha'(u)$ für eine Ecke $u \in I$ werden also berechnet aus den Zwischenoptima $\beta(u')$ und $\beta'(u')$ für alle Söhne von u .

Genauso berechnen sich die β, β' aus den α, α' . Sei dazu $u \in V^S$. In einer optimalen Zerlegung von T_u muß u genau einer Zentrum–Ecke $u' \in S_u$ zugeordnet sein. Für alle anderen Söhne u'' ist $T_{u''}$ optimal zu zerlegen, die Kosten hierfür sind $\alpha(u'')$. Weiter ist $T_{u'} \cup \{\text{pre}(u') = u\}$ optimal zu zerlegen, das heißt zu Kosten von $\alpha'(u')$.

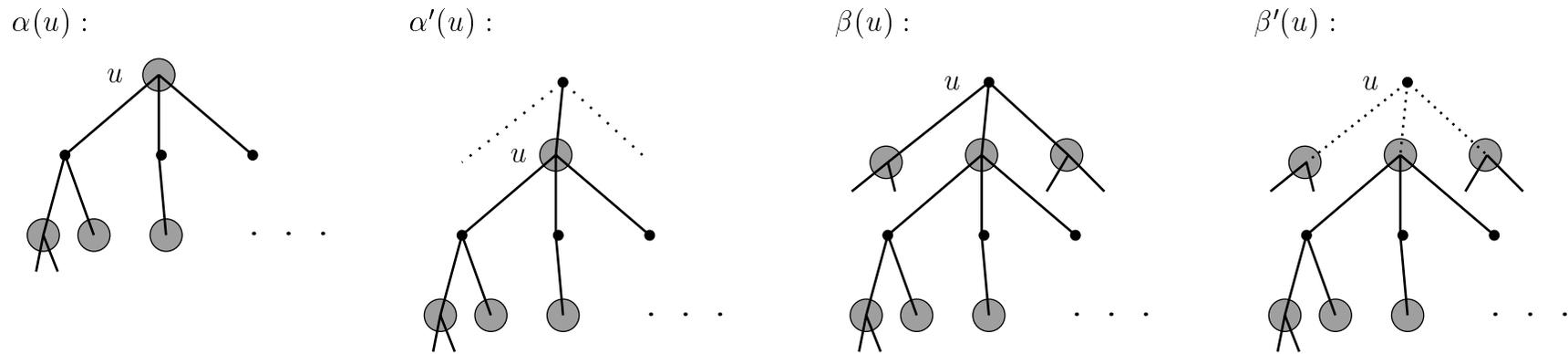


Abbildung 7.4: Bezeichnungen für Zwischenoptima in der dynamischen Programmierung für die Zerlegung von T^S .

Wir erhalten damit

$$\beta(u) = \min_{u' \in S_u} \alpha'(u') + \sum_{u'' \in S_u - \{u'\}} \alpha(u''). \quad (7.9)$$

Für die Gleichung zur Berechnung von $\beta'(u)$ beachte man, daß u darin keinem seiner Söhne, sondern dem Vater zugeordnet wird,

$$\beta'(u) = \sum_{u' \in S_u} \alpha(u'). \quad (7.10)$$

Die Gleichungen (7.6) bis (7.10) erlauben die Berechnung der Zwischenoptima in T^S von den Blättern aufsteigend, bis schließlich mit $\alpha(v_0)$ die Kosten einer optimalen Zerlegung π^* von T^S bestimmt sind. Durch Rückwärtsrechnung wird dann π^* selbst ermittelt. Algorithmus 7.2 faßt das Verfahren zusammen. Es ist eine Anpassung des einfachen Algorithmus 4.1 zur Baumzerlegung an die spezielle Struktur der Splitadjazenz.

Algorithmus 7.2 Zerlegung von T^S mit minimalen Kosten

Input: Splitadjazenz $T^S = (U^S, E^S) = (V^S \cup I, E^S)$

Output: Zerlegung π^* mit Kosten $c\pi^*$

- 1: wähle Wurzel $v_0 \in I$
 - 2: **for all** $u \in U^S$ von den Blättern zur Wurzel **do**
 - 3: **if** $u \in I$ **then**
 - 4: **if** u ist Blatt **then**
 - 5: Berechne $\alpha(u), \alpha'(u)$ gemäß (7.6)
 - 6: **else if** $u = v_0$ **then**
 - 7: Berechne $\alpha(u)$ gemäß (7.7)
 - 8: **else**
 - 9: Berechne $\alpha(u)$ gemäß (7.7)
 - 10: Berechne $\alpha'(u)$ gemäß (7.8)
 - 11: **end if**
 - 12: **else** /* $u \in V^S$ */
 - 13: Berechne $\beta(u)$ gemäß (7.9)
 - 14: Berechne $\beta'(u)$ gemäß (7.10)
 - 15: **end if**
 - 16: **end for**
 - 17: $c\pi^* := \alpha(v_0)$
 - 18: Bestimme π^* durch Rückwärtsrechnung
-

7.2.5.3 Aufwand der optimalen Zerlegung von T^S

Mit Algorithmus 7.2 läßt sich eine optimale Zerlegung π^* der Splitadjazenz in einem sehr allgemeinen Sinn bestimmen. Unten werden wir Beispiele für interessante Monoide und Kostenfunktionen c geben. Doch zunächst schätzen wir den Aufwand zur Bestimmung von π^* ab. Die maximalen Grade $\bar{\delta}_I$ und $\bar{\delta}_V$ seien wie in Satz 7.5 definiert.

Proposition 7.11

Der Aufwand zur Berechnung einer kostenminimalen Zerlegung der Splitadjazenz ist $O(\bar{\delta}_I 2^{\bar{\delta}_I} |I| + \bar{\delta}_V |V^S|)$.

Beweis. Der erste Summand schätzt den Aufwand für (7.7) und (7.8) ab. Dabei wird angenommen, daß der Aufwand zur Berechnung der Funktion c proportional zur Eckenzahl des Teilbaumes ist,¹⁰ und das Minimum durch Enumeration über alle Teilmengen $S' \subseteq S_u$ bestimmt wird. Wegen $|S_u| \leq \bar{\delta}_I - 1$ und dem Umstand, daß sowohl α und α' zu berechnen sind, ergibt sich ein Exponent von $\bar{\delta}_I$.

Der zweite Summand entspricht dem Aufwand zur Berechnung von (7.9) und (7.10). \square

Bemerkung 7.12

Ein Aufwand zur Berechnung einer optimalen Zerlegung von T^S mit einem exponentiellen Term erscheint zunächst wenig nützlich. Wir erinnern jedoch daran, daß in *praktischen* Problemstellungen die Splitadjazenz kleine Grade aufweist. Nehmen wir eine feste obere Schranke für den Maximalgrad an — zum Beispiel fünf oder sechs — so ergibt sich in Proposition 7.11 ein *linearer* Aufwand zur Bestimmung der optimalen Zerlegung.

Auch die Heuristik AssignMAX (Algorithmus 7.1) hat einen linearen Aufwand. Ihr Ziel ist eine balancierte Größe der Bezirke nach der Splitalfösung. Mit Algorithmus 7.2 halten wir nun ein viel mächtigeres Werkzeug in der Hand, um balancierte Bezirke zu bestimmen, ebenfalls bei linearem Aufwand (in praktischen Fällen).

7.2.5.4 Balancierte Bezirke mit optimaler Zerlegung von T^S

Durch geeignete Wahl des Monoids M und der Kostenfunktion c lassen sich mit Algorithmus 7.2 die in Tabelle 7.1 aufgeführten Operationalisierungen des Ziels balancierter Bezirksgrößen im optimalen Sinn lösen. Dazu wird jeweils das Baumzerlegungsproblem in der rechten Spalte optimal gelöst.

¹⁰Für den Aufwand der Operationen im Monoid M siehe Bemerkung 4.2.

Für die Minimierung der totalen Abweichung der Bezirksgrößen von der mittleren Größe μ wählen wir $M = (\mathbb{R}, +, \leq)$ und

$$c(U') = \left| \sum_{u \in U'} W_u - \mu \right| \quad (U' \subset U^S) \quad (7.11)$$

(Die Eckengewichte W_u werden in (7.4) und (7.5) definiert.)

Analog ist die Vorgehensweise zur Minimierung der Standardabweichung der Bezirksgrößen. Wir wählen $M = (\mathbb{R}, +, \leq)$ und

$$c(U') = \left| \sum_{u \in U'} W_u - \mu \right|^2 \quad (U' \subset U^S) \quad (7.12)$$

(Genauer wird so die Varianz minimiert, dies ist jedoch äquivalent zur Minimierung der Standardabweichung aufgrund der Monotonie der Quadratwurzelfunktion.)

Zur Minimierung der maximalen Abweichung von μ ist $M = (\mathbb{R}, \max, \leq)$ und c wie in (7.11) zu wählen. Dadurch erhalten wir einen praktikablen Algorithmus zur Minimierung der maximalen Abweichung. Er ist, wie oben ausgeführt, jedoch nur bei beschränktem Maximalgrad $\bar{\delta}_I$ polynomial. Nach Korollar 7.10 ist andererseits das Problem generell polynomial lösbar. In 7.2.5.6 werden wir zeigen, daß eine polynomiale Implementierung von Algorithmus 7.2 im Fall $M = (\mathbb{R}, \max, \leq)$ und Kosten c gemäß (7.11) auch bei unbeschränktem Maximalgrad möglich ist. Dies wird eine Bestätigung von Korollar 7.10 liefern, jedoch mit einer verbesserten Aufwandsabschätzung.

Zuvor jedoch soll die Mächtigkeit von Algorithmus 7.2 zur optimalen Splitauflösung, die aus seiner Allgemeinheit herrührt, anhand eines weiteren Beispiels seiner Anwendung demonstriert werden. Es wird durch folgende Überlegung motiviert.

Die Minimierung der maximalen Abweichung der Bezirksgrößen vom Mittel μ hat aus Praxissicht große Bedeutung. Es wird dabei jedoch nur der kleinste oder der größte Bezirk in der Zielfunktion berücksichtigt. Deshalb ist denkbar, daß selbst nach einer *optimalen* Splitauflösung unter diesem Kriterium eine verbesserte Balance möglich bleibt, indem Splits, die zu Bezirken mit einer näher am Mittelwert liegenden Größe gehören, anders zugeordnet werden, ohne daß dabei die *maximale* Abweichung vergrößert wird. Oder anders formuliert: Das Problem $T/2$ -Zent/ \cdot / $\|\cdot\|_\infty$ hat für T^S in der Regel keine eindeutige Optimallösung.¹¹

Aus diesem Grunde ist es sinnvoll, unter allen Optimallösungen von $T/2$ -Zent/ \cdot / $\|\cdot\|_\infty$ eine auszuwählen, die ein weiteres Zielkriterium, zum Beispiel die Minimierung der totalen Abweichung, optimiert. Damit wird der Praxisforderung nach möglichst

¹¹Die kleinstmögliche maximale Abweichung wird im allgemeinen aus einer „lokalen“ Gegebenheit herrühren (zum Beispiel einem sehr großen Split), für andere Bereiche der Splitadjazenz sind dann verschiedene Splitauflösungen möglich, die durch das Zielkriterium nicht bewertet werden.

kleiner maximaler Abweichung Rechnung getragen, zusätzlich jedoch wird der Spielraum ausgenutzt, der sich aus diesem Kriterium ergibt.

Im Kern ist die Idee also, ein Problem mit mehreren, nach Priorität geordneten Zielkriterien, zu lösen. Mit Algorithmus 7.2 ist dies ohne weiteres möglich — lediglich M und c sind geeignet zu definieren. Wir zeigen dies am diskutierten Beispiel, wobei die allgemeine Idee leicht zu sehen ist.

Optimale Splitauflösung bei zwei Zielkriterien. Wir definieren

$$M = (\mathbb{R} \times \mathbb{R}, \oplus, \leq_{\text{lex}}),$$

dies ist die Menge \mathbb{R}^2 mit lexikographischer Ordnung

$$(m_1, m_2) \leq_{\text{lex}} (m'_1, m'_2) \iff m_1 < m'_1 \text{ oder } (m_1 = m'_1 \wedge m_2 \leq m'_2)$$

und folgender „Addition“:

$$(m_1, m_2) \oplus (m'_1, m'_2) = (\max\{m_1, m'_1\}, m_2 + m'_2).$$

Man überzeugt sich leicht, daß dies ein geordnetes Monoid im Sinne der Definition 4.1 ist. Nun wird

$$c(U') = \left(\left| \sum_{u \in U'} W_u - \mu \right|, \left| \sum_{u \in U'} W_u - \mu \right| \right)$$

gesetzt. Dann berechnet Algorithmus 7.2 unter allen Splitaufösungen, die die maximale Abweichung der Bezirksgrößen vom Mittel μ minimieren, diejenige mit geringster totaler Abweichung.

Eine weitere nützliche Definition von c berücksichtigt ein Kompaktheitsmaß als nachgeordnetes Optimierungsziel. Die Eckenmenge U' eines zulässigen Teilbaumes besteht aus einer Zentrum-Ecke $u \in I$ und dazu adjazenten Split-Ecken U'' , das heißt $U' = \{u\} \cup U''$. Dies führt zu einem Beitrag von

$$\sum_{u'' \in U''} d_{uu''} w_{u''} = \sum_{u'' \in U''} d_{uu''} W_{u''}$$

im Kompaktheitsmaß des Modells ZP (7.1a) – (7.1d). Wird also

$$c(U') = c(\{u\} \cup U'') = \left(\left| \sum_{u \in U'} W_u - \mu \right|, \sum_{u'' \in U''} d_{uu''} W_{u''} \right)$$

gesetzt, so bestimmt Algorithmus 7.2 unter allen Splitaufösungen mit kleinster maximaler Abweichung der Bezirksgrößen von μ , diejenige, die zur am meisten kompakten Gebietsaufteilung führt.

7.2.5.5 Worst–case Abschätzung der maximalen Abweichung

In Satz 7.5 wurde die mit der Heuristik AssignMAX erzielbare maximale Abweichung der Bezirksgrößen von μ im schlechtesten Fall abgeschätzt. Wir zeigen nun, wie sich dies durch optimale Splitauflösung mit Algorithmus 7.2 verbessern läßt.

Satz 7.13

Sei $\bar{w} = \max \{w_v : v \in V^S\}$. Die durch Splitauflösung mit Algorithmus 7.2 (mit $M = (\mathbb{R}, \max, \leq)$ und c gemäß (7.11) gewählt) erzielte maximale Abweichung der Bezirksgrößen von μ ist nicht größer als \bar{w} .

Beweis. Sei $M = (\mathbb{R}, \max, \leq)$ und c gemäß (7.11) definiert. Dann gehen die Rekursionsgleichungen (7.6) bis (7.10) über in

$$\alpha(u) = |W_u - \mu| \quad \text{und} \quad \alpha'(u) = |W_u + W_{\text{pre}(u)} - \mu| \quad (7.13)$$

falls $u \in I$ Blatt ist, und für alle anderen $u \in I$ in (es sei $\max \beta'(S') := \max \{\beta'(u'') : u'' \in S'\}$ usw.)

$$\alpha(u) = \min_{S' \subseteq S_u} \max \left\{ |W_u + \sum_{u' \in S'} W_{u'} - \mu|, \max \beta'(S'), \max \beta(S_u - S') \right\} \quad (7.14)$$

$$\alpha'(u) = \min_{S' \subseteq S_u} \max \left\{ |W_u + W_{\text{pre}(u)} + \sum_{u' \in S'} W_{u'} - \mu|, \max \beta'(S'), \max \beta(S_u - S') \right\} \quad (7.15)$$

und für $u \in V^S$ erhalten wir

$$\beta(u) = \min_{u' \in S_u} \max \{ \alpha'(u'), \max \alpha(S_u - \{u'\}) \} \quad (7.16)$$

$$\beta'(u) = \max \alpha(S_u) \quad (7.17)$$

Wir zeigen die Behauptung des Satzes, indem durch Induktion von den Blättern zur Wurzel in T^S bewiesen wird, daß für alle Ecken $u \in U^S$ $\alpha(u), \alpha'(u) \leq \bar{w}$ beziehungsweise $\beta(u), \beta'(u) \leq \bar{w}$ gilt.

Für ein Blatt $u \in I$ gilt $W_u \leq \mu$ und $W_u + W_{\text{pre}(u)} \geq \mu$, denn es ist $W_u + W_{\text{pre}(u)} x_{u \text{pre}(u)} = \mu$ aufgrund von (7.2c), (7.4) und (7.5). Somit sind $\alpha(u)$ und $\alpha'(u)$ nicht größer als $W_{\text{pre}(u)} \leq \bar{w}$. Dies verankert die Induktion.

Für β und β' ist der Induktionsschritt aufgrund der Gleichungen (7.16) und (7.17) trivial.

Es verbleibt daher, den Induktionsschritt für (7.14) und (7.15) durchzuführen. Der mittlere und rechte Term in den Maximum–Ausdrücken dieser Gleichungen ist

aufgrund der Induktionsvoraussetzung nicht größer als \bar{w} , unabhängig von der Menge S' . Sei $u \in I$ kein Blatt. Aus (7.2c) folgt ähnlich wie oben

$$W_u \leq \mu \quad \text{und} \quad W_u + W_{\text{pre}(u)} + \sum_{u' \in S_u} W_{u'} \geq \mu.$$

Nun ist $W_{\text{pre}(u)} \leq \bar{w}$ und $W_{u'} \leq \bar{w}$ für $u' \in S'$ nach Definition von \bar{w} . Es gibt also eine Menge $S' \subseteq S_u$ mit

$$\mu - \bar{w} \leq W_u + \sum_{u' \in S'} W_{u'} \leq \mu + \bar{w}.$$

Dies zeigt, daß auch der erste Term in den geschweiften Klammern in (7.14) durch \bar{w} abgeschätzt werden kann. Somit ist $\alpha(u) \leq \bar{w}$.

Analog ist $\alpha'(u) \leq \bar{w}$, weil eine Menge S'' mit

$$\mu - \bar{w} \leq W_u + W_{\text{pre}(u)} + \sum_{u' \in S''} W_{u'} \leq \mu + \bar{w}$$

existiert. □

7.2.5.6 Minimierung der maximalen Abweichung mit polynomialem Aufwand

Um zu einer polynomialen Implementierung von Algorithmus 7.2 im Fall des Problems $T/2\text{-Zent}/\cdot/\|\cdot\|_\infty$ zu kommen, ist es ausreichend, die Optimierungsprobleme (7.7) und (7.8) in polynomialer Zeit zu lösen. Wir zeigen, wie dies auf Satz 5.4 (Komplexität des WEAK SUBSET SUM Problem) zurückgeführt werden kann. Dabei beschränken wir uns auf die Bestimmung von $\alpha(u)$ nach (7.7), für (7.8) ist die Vorgehensweise genau dieselbe.

Wie im Beweis von Satz 7.13 gezeigt, nimmt (7.7) im Fall der Problemstellung $T/2\text{-Zent}/\cdot/\|\cdot\|_\infty$ die Form

$$\alpha(u) = \min_{S' \subseteq S_u} \max \left\{ |W_u + \sum_{u' \in S'} W_{u'} - \mu|, \max \beta'(S'), \max \beta(S_u - S') \right\} \quad (7.18)$$

an. Dieses Minimierungsproblem wird in eine Folge von Entscheidungsproblem umgewandelt, indem der optimale Zielwert $\alpha(u)$ durch Binärsuche bestimmt wird. Sei dazu

$$h(S') = \max \left\{ |W_u + \sum_{u' \in S'} W_{u'} - \mu|, \max \beta'(S'), \max \beta(S_u - S') \right\} \quad (S' \subseteq S_u) \quad (7.19)$$

wobei wir hier und in folgendem ein festes $u \in I$ betrachten, das kein Blatt von T^S ist. Es gilt somit

$$\alpha(u) = \min_{S' \subseteq S_u} h(S').$$

Satz 7.13 liefert zunächst die Information $\alpha(u) \in [0, \bar{w}]$. Durch Lösen des folgenden Entscheidungsproblems können wir dieses Intervall halbieren. Sei $H := \bar{w}/2$.

$$\text{Existiert } S' \subseteq S_u \text{ mit } h(S') \leq H? \quad (7.20)$$

Im Fall einer positiven Antwort ist $\alpha(u) \in [0, \bar{w}/2]$, andernfalls ist $\alpha(u) \in (\bar{w}/2, \bar{w}]$. Die Suche wird fortgesetzt, indem H als Mitte des verbleibenden Intervalls gesetzt und (7.20) erneut gelöst wird. Gehen wir von ganzzahligen Größenattributen w_v der KGE aus, so sind die möglichen Werte von $\alpha(u)$ diskret, wobei zwischen zwei Ganzzahlen höchstens zwei dieser Werte liegen. (Man beachte, daß μ trotz ganzzahliger w_v nicht ganzzahlig sein muß.) Mit $O(\log \bar{w})$ Tests der Form (7.20) kann somit das Intervall für $\alpha(u)$ so verkleinert werden, daß es nur noch einen möglichen Wert enthält. Damit ist schließlich $\alpha(u)$ bestimmt.

Wir wenden uns der Frage zu, wie das Entscheidungsproblem (7.20) gelöst wird. Dazu sei H ein fester Wert. Drei einfache Beobachtungen, die sich unmittelbar aus (7.19) ergeben, dienen als Reduktionskriterien.

Beobachtung 1. Gibt es ein $u' \in S_u$ mit $\beta(u') > H$ und $\beta'(u') > H$, so ist die Antwort in (7.20) negativ.

Beobachtung 2. Ist $u' \in S_u$ mit $\beta(u') > H$ und $\beta'(u') \leq H$, so ist $u \in S'$ in jeder Lösung von (7.20).

Beobachtung 3. Ist $u' \in S_u$ mit $\beta(u') \leq H$ und $\beta'(u') > H$, so ist $u \notin S'$ in jeder Lösung von (7.20).

Sei nun

$$R = R(H) := \{u' \in S_u : \beta(u') \leq H \text{ und } \beta'(u') \leq H\}$$

die Menge der *relevanten* Söhne von u (bezüglich H). Sei weiter

$$W(H) := W_u + \sum (W_{u'} : u' \in S_u, \beta'(u') \leq H < \beta(u)).$$

Aufgrund unserer Beobachtungen und (7.19) ist Problem (7.20) äquivalent zu

$$\text{Existiert } R' \subseteq R \text{ mit } |W(H) + \sum_{u' \in R'} W_u' - \mu| \leq H? \quad (7.21)$$

Die Zurückführung dieser Frage auf das WEAK SUBSET SUM Problem ist nun offensichtlich. Wir setzen $L := \mu - W(H) - H$ und $U := \mu - W(H) + H$. Damit ist (7.21) äquivalent zu

$$\text{Existiert } R' \subseteq R \text{ mit } L \leq \sum_{u' \in R'} W_u' \leq U? \quad (7.22)$$

Nach Satz 5.4 ist der Aufwand zum Lösen des WSS (7.22) $O(|R|^2 \sum_{u' \in R} W_{u'} / (U-L))$. Durch Abschätzen von $W_{u'}$, $u' \in R$, weisen wir nach, daß dies polynomial ist.

Hilfssatz 7.14

Für $u' \in R(H)$ ist $W_{u'} < 2pH$ ($p = |I|$).

(Der Beweis folgt unten.) Aufgrund von Hilfssatz 7.14 ist der Aufwand für (7.22) $O(p|R(H)|^3)$. Das Minimum $\alpha(u)$ läßt sich daher in $O(p|S_u|^3 \log \bar{w})$ Rechenschritten bestimmen. Wir erhalten den folgenden Satz, der Proposition 7.11 im hier vorliegenden Fall verbessert; die exponentielle Abschätzung wird durch eine polynomiale ersetzt.

Satz 7.15

Der Aufwand zur Berechnung einer Zerlegung der Splitadjazenz, die die maximale Abweichung der Bezirksgrößen vom Mittel μ minimiert, ist $O(\bar{\delta}_I^3 |I|^2 \log \bar{w} + \bar{\delta}_V |V^S|)$ ($\bar{w} = \max\{w_v : v \in V^S\}$).

Hilfssatz 7.14 entspricht genau dem Hilfssatz 5.21 in 5.4. Die Argumentation des dortigen Beweises kann unverändert übernommen werden und muß lediglich formal an die hier vorliegende Situation angepaßt werden. Wir bringen den Beweis dennoch, um keine Lücke entstehen zu lassen.

Beweis von Hilfssatz 7.14. Sei $u' \in R = R(H)$. Das Gewicht des Teilbaumes $T_{u'}$ ist $WT_{u'}$, ferner bezeichnen wir mit $p_{u'} = |I \cap T_{u'}|$ die Anzahl der Zentrum-Ecken in $T_{u'}$. Aus $\beta(u') \leq H$ folgt $WT_{u'} \leq p_{u'}(\mu + H)$. Außerdem impliziert $\beta^l(u') \leq H$ für alle $u'' \in S_{u'}$ die Abschätzung $WT_{u''} \geq p_{u''}(\mu - H)$. Die Beziehungen

$$WT_{u'} = W_{u'} + \sum_{u'' \in S_{u'}} WT_{u''} \quad \text{und} \quad p_{u'} = \sum_{u'' \in S_{u'}} p_{u''}$$

sind trivial, für die rechte beachte man $u' \notin I$. Durch Subtraktion erhalten wir

$$W_{u'} \leq p_{u'}(\mu + H) - \sum_{u'' \in S_{u'}} p_{u''}(\mu - H) = p_{u'} 2H.$$

Wegen $p_{u'} < p$ folgt daraus die Behauptung. □

7.2.6 Partielle Splitauflösung bei beschränkter Abweichung

In den Abschnitten 7.2.4 und 7.2.5 wurde gezeigt, wie optimale Splitauflösung mit Hilfe von dynamischer Programmierung, basierend auf der Splitadjazenz, umgesetzt werden kann. Insbesondere die Minimierung der maximalen Abweichung der Bezirksgrößen vom Mittelwert μ wurde eingehend diskutiert.

In der Praxis sind jedoch Situationen gegeben, in denen eine bestimmte Abweichung vom Mittelwert nicht überschritten werden darf. Wenn diese kleiner ist, als die aus einer optimalen Splitauflösung resultierende Abweichung, ist es nicht möglich, mit der bisher vorgestellten Vorgehensweise — Lösung des linearen Programms TP (7.2) und vollständige Splitauflösung — zu einer zulässigen Gebietsaufteilung zu gelangen.

Als Ausweg in einer solchen Situation wird von Fleischmann und Paraschis [35] die Abschwächung von Zulässigkeit in einer anderen Richtung vorgeschlagen. Wenn darauf verzichtet wird, *alle* Splits aufzulösen, kann im allgemeinen eine kleinere maximale Abweichung erzielt werden, denn in der Lösung von TP sind die Bezirksgrößen perfekt balanciert; die maximale Abweichung beträgt Null.

Fleischmann und Paraschis [35] haben daher Heuristiken entwickelt, die möglichst viele Splits auflösen und die verbleibenden so auf adjazente Zentren verteilen, daß Bezirke entstehen, deren Größe die erlaubte Abweichung vom Mittel nicht überschreitet. Es sind somit Verfahren zur *partiellen* Splitauflösung bei beschränkter Abweichung. Das Ergebnis eines solchen Verfahrens ist eine Gebietsaufteilung mit einigen KGE, die anteilig mehreren (benachbarten) Gebieten zugeordnet sind. Durch (manuelle) Nachbearbeitung können diese KGE unterteilt und dadurch klare Gebietsabgrenzungen hergestellt werden.

Aus folgenden Gründen gehen wir in diesem Abschnitt auf Heuristiken zur partiellen Splitauflösung ein.

1. In diesem Zusammenhang wurde zuerst die Struktur der Splitadjazenz für das Lösen von Gebietsaufteilungsproblemen ausgenutzt. Darin kann die Wurzel für unsere Verfahren zur optimalen Splitauflösung gesehen werden.
2. Heuristiken zur partiellen Splitauflösung können in Praxissituationen hilfreich sein, wenn eine zulässige Abweichung vom Mittel der Bezirksgrößen nicht überschritten werden darf.
3. Fleischmann und Paraschis [35] geben zunächst eine einfache Heuristik an, die wir hier *sukzessive Toleranzvergrößerung* nennen. Sie wird dann jedoch als nicht wirksam genug verworfen, und die Autoren entwickeln die deutlich komplexere *local assignment* Heuristik. (Letztere werden wir aus Platzgründen nicht darstellen.)

Unseres Erachtens benutzen Fleischmann und Paraschis bei der sukzessiven Toleranzvergrößerung eine ungeeignete Zielfunktion und erhalten *deswegen* eine unbefriedigende Wirksamkeit. Wir schlagen daher eine andere Zielfunktion vor. Rechentests zeigen, daß damit die sukzessive Toleranzvergrößerung genauso erfolgreich im Auflösen von Splits ist, wie die *local assignment* Heuristik.

7.2.6.1 Sukzessive Toleranzervergrößerung

Wir geben zunächst die Heuristik sukzessiver Toleranzvergrößerung nach Fleischmann und Paraschis [35, S. 526] an. Auf eine verbale Beschreibung folgt die formale Darstellung.

Zunächst wird das lineare Programm TP (7.2) (S. 143) gelöst. Aufgrund der Nebenbedingung (7.2c) beträgt die erlaubte Toleranz Null. Nun werden alle ganzzahligen Variablen in der Lösung von TP fixiert und das Problem mit einer Toleranz $\tau' > 0$ für die restlichen Variablen erneut gelöst. Somit werden die Nebenbedingungen (7.2c) ersetzt durch

$$\mu(1 - \tau') \leq \sum_{v \in V} w_v x_{iv} \leq \mu(1 + \tau') \quad (i \in I) \quad (7.23)$$

Sind in der Lösung bezüglich dieser relaxierten Bedingungen weitere Variablen ganzzahlig, so werden auch diese fixiert. Daraufhin wird die Toleranz weiter vergrößert und das Problem mit den verbliebenen Variablen erneut gelöst.

Iterativ wird so versucht, möglichst viele Variablen auf ganzzahlige Werte zu fixieren. Abgebrochen wird, wenn τ' mit der maximal zulässigen Toleranz τ übereinstimmt. Die Erhöhung von τ' geschieht jedesmal um einen festen Betrag τ/s , wobei s die vorgegebene Anzahl der Iterationen ist.

Mit dem schrittweisen Erhöhen der Toleranz verbindet sich die Erwartung, im allgemeinen mehr Variablen fixieren zu können, als wenn sofort $\tau' = \tau$ gesetzt würde. Testrechnungen zeigen, daß dies tatsächlich meistens der Fall ist.

Das Fixieren aller Variablen x_{iv} , $i \in I$, zu einer KGE v bedeutet, daß v eindeutig zugewiesen ist. Daher ist das Ziel der Heuristik, möglichst viele der Splits in der Lösung von TP aufzulösen, ohne die zulässige Toleranz τ zu überschreiten.

Das von Fleischmann und Paraschis berichtete Problem der mangelnden Wirksamkeit dieser Heuristik liegt unseres Erachtens in der Zielfunktion (7.2a). Sie entspricht nicht der eigentlichen Absicht. Wenn durch Vergrößern von τ' die Nebenbedingungen gelockert werden, tendiert die Zielfunktion (7.2a) dazu, für jeden Split v die Variable x_{iv} für das *nächstgelegene* Zentrum i in der Splitadjazenz zu erhöhen.

Heuristisch sinnvoller ist es jedoch, die Variable zu demjenigen Zentrum zu erhöhen, das bereits den größten Anteil an der KGE v hat. Eine Zielfunktion die dieser Absicht entspricht, läßt sich leicht angeben. Mit ihr führt sukzessive Toleranzvergrößerung zu Ergebnissen, die denen der local assignment Heuristik vergleichbar sind (siehe 7.2.6.3). Da letztere deutlich komplizierter ist, präferiert der Verfasser die sukzessive Toleranzvergrößerung als Heuristik zur partiellen Splitauflösung.

Formale Darstellung. Die Heuristik der sukzessiven Toleranzvergrößerung ist Algorithmus 7.3. In jeder Iteration dieses Verfahrens wird ein lineares Programm gelöst,

mit jeweils veränderten Koeffizienten. Es hat die folgende Gestalt.

$$(\text{LP}(\tau', \zeta, l, u)) \quad \min \sum_{i,v} \zeta_{iv} w_v x_{iv} \quad (7.24a)$$

$$\text{unter} \quad \sum_{i \in I} x_{iv} = 1 \quad (v \in V) \quad (7.24b)$$

$$\mu(1 - \tau') \leq \sum_{v \in V} w_v x_{iv} \leq \mu(1 + \tau') \quad (i \in I) \quad (7.24c)$$

$$l_{iv} \leq x_{iv} \leq u_{iv} \quad (v \in V, i \in I) \quad (7.24d)$$

Variablenfixierungen werden in Algorithmus 7.3 durch das Gleichsetzen von unterer und oberer Schranke vorgenommen. In der ersten Iteration ist $\tau' = 0$. Aufgrund der Initialisierung von ζ , l und u stimmt das gelöste $\text{LP}(0, \zeta, l, u)$ mit TP (7.2) überein. In dessen Lösung haben die meisten Variablen ganzzahlige Werte und werden fixiert. Mit einer um $\Delta\tau$ vergrößerten Toleranz wird die nächste Iteration ausgeführt.

Algorithmus 7.3 Sukzessive Toleranzerhöhung zur Splitauflösung

Input: Anzahl Iterationen $s > 0$, maximale Toleranz τ

Output: Lösung (x_{iv}) /* Zuordnung von KGE zu Zentren */

- 1: $\zeta_{iv} := d_{iv} w_v$ für alle i, v
 - 2: $l_{iv} := 0$ und $u_{iv} := 1$ für alle i, v
 - 3: $\tau' := 0$; $\Delta\tau := \tau/s$
 - 4: **while** $\tau' \leq \tau$ **do**
 - 5: (x_{iv}) sei Optimallösung von $\text{LP}(\tau', \zeta, l, u)$ (7.24)
 - 6: **for all** $i \in I, v \in V$ **do** /* fixiere ganzzahlige Variablen */
 - 7: **if** $x_{iv} = 0$ **then** $u_{iv} := 0$ **end if**
 - 8: **if** $x_{iv} = 1$ **then** $l_{iv} := 1$ **end if**
 - 9: **end for**
 - 10: Aktualisiere ggf. Zielfunktion (ζ_{iv}) /* siehe Text */
 - 11: $\tau' := \tau' + \Delta\tau$
 - 12: **end while**
-

Ab der zweiten Iteration ist das Ziel die Auflösung möglichst vieler Splits der Optimallösung von TP. In der Version von Fleischmann und Paraschis [35] der Heuristik wird, wie oben ausgeführt, die Zielfunktion dabei nicht verändert. Damit entfällt das Aktualisieren in Zeile 10. Unser Ziel ist jedoch, die Splitauflösung günstig zu beeinflussen, indem die Zielfunktion geeignet gewählt wird.

Naheliegender ist es, die Vergrößerung des Wertes von Variablen knapp über Null teuer zu machen, dagegen die Vergrößerung von Variablen nahe an Eins billig. Wir verwenden deshalb das Ergebnis der k -ten Iteration um die Zielkoeffizienten für die

$(k + 1)$ -te Iteration festzulegen. Dazu setzen wir in Zeile 10

$$\zeta_{iv} := 1/x_{iv} \quad (7.25)$$

für alle nichtganzzahligen Variablen x_{iv} . (Da die anderen Variablen durch die Schranken fixiert sind, ist die Größe ihres Zielkoeffizients ohne Bedeutung.) Mit dieser Zielfunktion ist der Wert der Lösung der k -ten Iteration gerade die Anzahl der gebrochenen Variablen. Da in der $(k + 1)$ -ten Iteration die Nebenbedingungen (7.24c) weniger strikt ist, wird die neue Lösung einen kleineren Wert haben, dazu müssen in der Tendenz Variablen mit Wert nahe an Eins vergrößert und solche mit Wert nahe an Null verkleinert werden.

Um zu überprüfen, wie sich die vorgeschlagene Modifikation der Zielfunktion auf die Anzahl an verbleibenden Splits auswirkt, wurde eine große Zahl an Testproblemen mit beiden Varianten der Heuristik gerechnet. Darüber berichten wir in 7.2.6.3. Jedoch kann bereits aufgrund folgender Überlegung erwartet werden, daß mit der modifizierten Zielfunktion im allgemeinen mehr Splits aufgelöst werden können.

Man nehme an, daß mit $s = 1$ nur eine Iteration zur Splitalösung in Algorithmus 7.3 gemacht wird. Wie groß muß τ gewählt werden, damit *alle* Splits aufgelöst werden können?

Die ursprüngliche Zielfunktion ordnet einen Split dem nächstgelegenen Zentrum zu. Dies wird im allgemeinen eine größere maximale Abweichung von μ ergeben, als wenn jeder Split wie in der Heuristik AssignMAX (Algorithmus 7.1) zugeordnet wird. Letzteres wird durch die gemäß (7.25) modifizierte Zielfunktion erreicht. Damit nun alle Splits in nur einer Iteration aufgelöst werden können, muß τ mindestens so groß sein, wie diese maximale Abweichung. Für die modifizierte Zielfunktion können daher im allgemeinen bei kleinerem τ in einer Iteration alle Splits aufgelöst werden.

Aus diesem Grund ist zu erwarten, daß die Anzahl der verbleibenden Splits bei einer Zielfunktion gemäß (7.25) geringer ist.

7.2.6.2 Die local assignment / deassignment Heuristiken

Wie oben erwähnt, wird die local assignment Heuristik von Fleischmann und Paraschis [35] aus Platzgründen hier nicht dargestellt. Ihre Idee ist jedoch einfach zu erklären.

Ihr Ziel ist die Fixierung möglichst vieler Variablen von $LP(\tau', \zeta, l, u)$ (7.24) auf ganzzahlige Werte, ohne daß das verbleibende Problem unzulässig wird. Zunächst wird das Problem (wie in Algorithmus 7.3) mit $\tau' = 0$, $l = 0$ und $u = 1$ gelöst, dies ist das Transportproblem TP (7.2). Die Variablen mit ganzzahligem Wert werden fixiert (durch Setzen der Schranken l_{iv} bzw. u_{iv}). Dann wird $\tau' = \tau$ gesetzt, wodurch sich der zulässige Bereich von (7.24) vergrößert. Dieser Spielraum wird nun von der

local assignment Heuristik benutzt, um nichtganzzahlige Variablen x_{iv} auf Eins zu fixieren, und damit Splits vollständig einem Bezirken zuzuordnen.

Fleischmann und Paraschis zeigen, daß es möglich ist, aufgrund „lokaler“ Kriterien zu entscheiden, ob eine solche Fixierung das lineare Programm $LP(\tau, \zeta, l, u)$ zulässig beläßt (daher der Name der Heuristik). Diese Kriterien nutzen ganz entscheidend die Baum-Struktur der Splitadjazenz aus.

Eine ganz ähnliche Vorgehensweise setzt Hojati [52] zur partiellen Splitauflösung ein. Es liegt nahe, sie in Analogie zu Fleischmann und Paraschis [35] als local deassignment Heuristik zu bezeichnen, denn Hojati fixiert Variablen auf Null. Um zu prüfen, ob $LP(\tau, \zeta, l, u)$ dabei zulässig bleibt, verwendet Hojati einen LP Solver: Es wird probenhalber für eine Variable die Schranke $u_{iv} = 0$ gesetzt, mit dem Solver Zulässigkeit geprüft, und im negativen Fall u_{iv} auf Eins zurückgesetzt. Dann wird mit der nächsten Variablen in gleicher Weise verfahren.

Der Rückgriff auf einen LP Solver erscheint dabei zwar bequem, ist aber andererseits unnötig; es lassen sich ohne weiteres zur local assignment Heuristik analoge Kriterien angeben, wann die Fixierung einer Variablen auf Null das lineare Programm $LP(\tau, \zeta, l, u)$ zulässig beläßt. (In GAMOR ist die local deassignment Heuristik durch Einbeziehung solcher Kriterien implementiert.)

Als Motivation für seine Heuristik nennt Hojati [52] das unbefriedigende Verhalten der local assignment Heuristik auf den von ihm untersuchten Problemdaten. Grundsätzlich sind die beiden Heuristiken jedoch als äquivalent anzusehen, denn ob man durch assignment oder durch deassignment mehr Splits auflöst, hängt letztlich nur von der Reihenfolge ab, in welcher die Variablen fixiert werden. Die folgenden Rechentests sollten daher auch prüfen, ob die local deassignment im Mittel mehr Splits auflöst.

7.2.6.3 Rechenergebnisse

Die folgenden sechs Verfahrensvarianten zur partiellen Splitauflösung wurden auf etwa eintausend zufällig erzeugte Gebietsaufteilungsprobleme angewendet.¹²

-
- A local assignment Heuristik (Fleischmann und Paraschis [35])
 - B local deassignment Heuristik (Hojati [52])
 - C sukz. Toleranzerhöhung, $\zeta_{iv} = d_{iv}w_v$, $s = 5$ (Fleischmann u. Paraschis [35])
 - D sukz. Toleranzerhöhung, $\zeta_{iv} = 1/x_{iv}$, $s = 1$
 - E sukz. Toleranzerhöhung, $\zeta_{iv} = 1/x_{iv}$, $s = 5$
 - F sukz. Toleranzerhöhung, $\zeta_{iv} = 1/x_{iv}$, $s = 25$
-

¹²Zur Erzeugung dieser Testprobleme siehe S. 138.

Untenstehende Tabelle zeigt die aufsummierte Anzahl der verbliebenen Splits für jeweils eine Teilmenge dieser Probleme und eines der Verfahren A bis F. Zusammengefaßt wurden dabei Probleme mit gleicher Anzahl n an KGE. Die Anzahl p der Bezirke wurde variiert, auf jede Kombination von n und p entfielen zehn oder zwanzig Problemstellungen. Jedes dieser Einzelprobleme wurde mit vier verschiedenen Werten von τ gelöst.¹³ So ergeben sich zum Beispiel für $n = 50$ insgesamt $3 \times 20 \times 4 = 240$ Probleme.

n	p	#Prob	Anzahl verbliebener Splits						
			A	B	C	D	E	F	
50	5,7,10	240	995	1000	1194	1020	1007	1008	
100	10,13,20	240	1750	1748	2187	1784	1738	1722	
250	17,25,33,49	320	4404	4484	5720	4565	4392	4354	
500	25,33,49,64	160	2880	2940	3900	3026	2883	2871	
1000	49,64,100	120	3283	3294	4525	3442	3259	3206	
			1080	13312	13466	17526	13837	13279	13161

Aus den Ergebnissen lassen sich folgende Schlüsse ziehen.

1. Sukzessive Toleranzerhöhung ist bei Verwendung der Zielfunktion $\zeta_{iv} = d_{iv}w_v$ in LP(τ, ζ, l, u) zur Auflösung von Splits deutlich weniger wirksam als die local assignment Heuristik (Spalten A und C). Dies wurde bereits von Fleischmann und Paraschis [35] beobachtet.
2. Sukzessive Toleranzerhöhung mit der von uns vorgeschlagenen Zielfunktion (7.25) ist schon bei moderater Anzahl an Iterationen in Algorithmus 7.3 im Durchschnitt besser als die local assignment Heuristik (Spalten A und E). Nur eine Iteration durchzuführen, ist unzureichend (Spalte D); sehr viele Iterationen bewirken noch eine gewisse Verbesserung (Spalte F).
3. Die local assignment Heuristik schneidet im Durchschnitt besser ab, als die local deassignment Heuristik (Spalten A und B). Dies steht im Gegensatz zur Feststellung von Hojati [52], der allerdings nur einen Problem Datensatz betrachtet.¹⁴

¹³Die Wahl von τ orientierte sich dabei an der aus optimaler Splitalösung resultierenden maximalen Abweichung vom Mittel μ . Der Wert von τ betrug jeweils 5%,10%,20% und 40% dieser Abweichung.

¹⁴Hojati [52] gibt Problem Daten und Ergebnisse vollständig an. Daher war es möglich, die Berechnungen mit GAMOR nachzuvollziehen. Wir erhielten mit der local deassignment Heuristik sechs und mit der local assignment Heuristik fünf verbleibende Splits. Auch wenn das Ergebnis von der Reihenfolge abhängt, in der die Variablen fixiert werden, ist die Aussage von Hojati, mit der local assignment Heuristik hätte keiner der zehn Splits aufgelöst werden können, nicht ganz nachvollziehbar. Mit der local deassignment Heuristik verbleiben bei Hojati ebenfalls sechs Splits.

7.3 KGE-Zentren-Zuordnung mittels Lagrange-Relaxation und Subgradienten-Verfahren

Im vorangehenden Abschnitt 7.2 haben wir dargestellt, wie durch Lösen des Transportproblems (7.2) und nachfolgender Splitauflösung eine Zuordnung der KGE zu den Zentren berechnet werden kann. Diese Vorgehensweise kann als primales Verfahren interpretiert werden. Im folgenden gehen wir auf ein duales Verfahren ein, das von Zoltners und Sinha [115] für die Lösung von Gebietsaufteilungsproblemen vorgeschlagen wurde.

Zunächst stellen wir das Verfahren vor. Danach geben wir eine anschauliche Interpretation dafür und schließlich arbeiten wir die Beziehungen zur Vorgehensweise in Abschnitt 7.2 heraus. Dabei wird deutlich werden, daß letztere als methodisch überlegen angesehen werden muß.¹⁵

7.3.1 Das Verfahren von Zoltners und Sinha

Das von Zoltners und Sinha [115] vorgeschlagene Verfahren kann als Lagrange-Heuristik angesehen werden. Ausgehend von einer Lagrange-Relaxation¹⁶ des Zuordnungsmodells (7.1) wird mit dem Subgradienten-Verfahren der Lagrange-Dual approximativ gelöst. Dabei ist man aber in erster Linie nicht am Wert der Lagrangefunktion oder an den Lagrange-Multiplikatoren sondern an den Lösungen der Lagrange-Subprobleme interessiert. Dies sind — in gewissem Sinne — primale Lösungen, die KGE-Zentren-Zuordnungen charakterisieren. Unter den im Verlauf der Iterationen des Subgradienten-Verfahrens berechneten dieser Lösungen wählt man schließlich die beste hinsichtlich der vorgegebenen Zielsetzung, die in diesem Fall in der Balance der entsprechenden Gebietsaufteilung gegeben ist.

Es ist also festzuhalten, daß im Unterschied zur gängigen Verwendung von Lagrange-Relaxation und Subgradienten-Verfahren in der ganzzahligen Programmierung, hier das Ziel nicht in der Berechnung einer unteren Schranke für den optimalen Zielfunktionswert eines gemischtganzzahligen linearen Programms besteht.

Ausgangspunkt des Verfahrens ist das ganzzahlige Programm (7.1) mit $\tau = 0$,

¹⁵Diese Tatsache ist etwas überraschend, denn Zoltners und Sinha [115] zielten mit ihrem Ansatz auf eine Verbesserung der von Hess und Samuels [51] entwickelten Methode „Transportproblem und Splitauflösung“. Es wird deutlich, daß durch die optimale Splitauflösung mittels Baumzerlegung das eigentliche Kernproblem gelöst wird.

¹⁶Anwendungsorientierte Darstellungen der Verwendung von Lagrange-Relaxation in der ganzzahligen Programmierung findet man zum Beispiel in Ahuja et al. [4] und in Reeves [91]. Eine tiefergehende Einführung geben Parker und Rardin [86].

also

$$(ZP0) \quad \min \sum_{i,v} d_{iv} w_v x_{iv} \quad (7.26a)$$

$$\text{unter} \quad \sum_{i \in I} x_{iv} = 1 \quad (v \in V) \quad (7.26b)$$

$$\sum_{v \in V} w_v x_{iv} = \mu \quad (i \in I) \quad (7.26c)$$

$$x_{iv} \in \{0, 1\} \quad (v \in V, i \in I) \quad (7.26d)$$

Dieses Programm ist in der Regel unzulässig. Nun werden die Balance-Bedingungen (7.26c) relaxiert und mit den Lagrange-Multiplikatoren λ_i , $i \in I$, in die Zielfunktion gebracht; wir erhalten das Lagrange-Subproblem

$$C(\lambda) = \min \sum_{i,v} (d_{iv} - \lambda_i) w_v x_{iv} + \mu \sum_i \lambda_i \quad (7.27a)$$

$$\text{unter} \quad \sum_{i \in I} x_{iv} = 1 \quad (v \in V) \quad (7.27b)$$

$$x_{iv} \in \{0, 1\} \quad (v \in V, i \in I) \quad (7.27c)$$

Dieses Problem besitzt die triviale Lösung

$$C(\lambda) = \sum_v w_v \min_i \{d_{iv} - \lambda_i\} + \mu \sum_i \lambda_i.$$

Das heißt, zu $v \in V$ wird für ein i , das $d_{iv} - \lambda_i$ minimiert, $x_{iv} = 1$ gesetzt; alle anderen x_{jv} sind Null.

Beobachtung 1. Es ist der Fall denkbar, daß das minimierende Zentrum i zu einer KGE v nicht eindeutig ist. Wie soll eine solche Mehrdeutigkeit aufgelöst werden? Diese Frage wird von Zoltners und Sinha [115] nicht angesprochen. Sie scheint auf den ersten Blick auch nebensächlich zu sein. Naheliegend ist es zum Beispiel, eine zufällige Auswahl von i zu treffen, wenn es mehrere den Term $d_{iv} - \lambda_i$ minimierende Zentren zu einer KGE gibt. Die aufgeworfene Frage wird jedoch in Abschnitt 7.3.3 als ein Kernproblem erkannt werden.

Zur approximativen Lösung des Lagrange-Duals

$$(LD) \quad C^* = \max C(\lambda) \quad \text{unter} \quad \lambda \in \mathbb{R}^p \quad (7.28)$$

wird das Subgradienten-Verfahren verwendet. Hierzu geben Zoltners und Sinha [115] keine Details an (z.B. zur Wahl der Schrittweite, Abbruchkriterien, etc.). Aus einer

optimalen Lösung (x_{iv}) des Lagrange-Subproblems wird in jeder Iteration des Verfahrens der Subgradient (η_i) mit

$$\eta_i = \mu - \sum_v w_v x_{iv}$$

gebildet, und die Lagrange-Multiplikatoren werden gemäß

$$\lambda_i := \lambda_i + \kappa \eta_i \quad (7.29)$$

aktualisiert. Dann wird das Lagrange-Subproblem erneut gelöst. Die Schrittweite κ wird im Verlauf des Verfahrens zunehmend verkleinert und tendiert gegen Null.

Diskussion. Durch die Lösung des Lagrange-Subproblems ist eine (von λ abhängige) Zuordnung von KGE zu Zentren gegeben. Dabei wird jede KGE ihrem nächstgelegenen Zentrum zugeordnet, wobei die modifizierten Distanzen $d_{iv} - \lambda_i$ zugrundegelegt werden. Anschaulich wird durch die Aktualisierungsvorschrift (7.29) der Multiplikator für Bezirke, die kleiner sind als μ , vergrößert, für zu große Bezirke dagegen verkleinert. Dadurch werden die modifizierten Distanzen entsprechend verändert, so daß in der nächsten Iteration zum Beispiel das Zentrum eines vorher zu kleinen Bezirkes „attraktiver“ für die KGE wird. Das Subgradienten-Verfahren tendiert somit dazu, den Lagrange-Multiplikatoren Werte zu geben, die eine balancierte Gebietsaufteilung als Lösung des Lagrange-Subproblems hervorbringen.

Da in jeder Iteration eine KGE-Zentren-Zuordnung berechnet wird, kann nun die beste hinsichtlich der gewählten Balance-Messung (maximale Abweichung von μ , totale Abweichung, ...) als Ergebnis des Verfahrens festgehalten werden.

Der Vorteil der Vorgehensweise nach Zoltners und Sinha [115] gegenüber der Bestimmung einer Gebietsaufteilung mittels Transportproblem und Splitauflösung, scheint somit vor allem darin zu bestehen, daß hier viele Gebietsaufteilungen berechnet werden, unter denen dann die beste gewählt werden kann, während im zweiten Fall nur eine Gebietsaufteilung bestimmt wird. Da die Tendenz des Subgradienten-Verfahrens zu balancierten Bezirken geht, wird man erwarten, daß so möglicherweise eine besser balancierte Gebietsaufteilung bestimmt werden kann, als mit den Methoden des Abschnitts 7.2. Wir werden jedoch zeigen, daß dies nur sehr selten vorkommen kann (siehe 7.3.3).

7.3.2 Anschauliche Interpretation des Verfahrens

Um eine anschauliche Interpretation des Verfahrens von Zoltners und Sinha [115] zu geben, erinnern wir daran, daß in diesem Kapitel KGE und Zentren durch Punkte der Ebene repräsentiert sind (siehe 7.1.1).

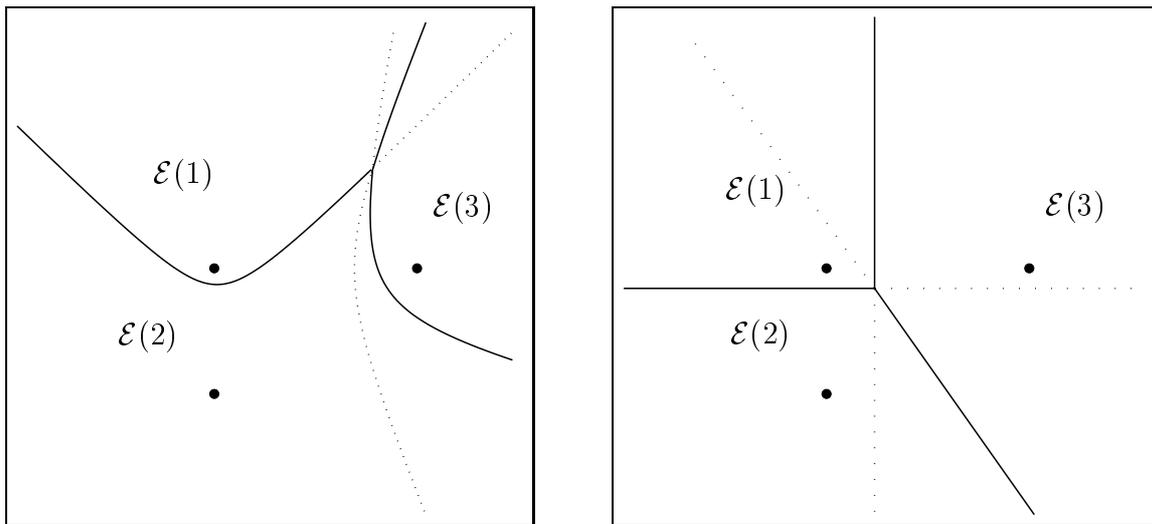


Abbildung 7.5: Einzugsbereiche für drei Zentren bei euklidischen und quadriert-euklidischen Distanzen.

Wie oben erläutert, wird eine KGE-Zentren-Zuordnung gebildet, indem jede KGE dem (genauer: einem) nächstgelegenen Zentrum bezüglich der modifizierten Distanzen $d_{iv} - \lambda_i$ zugeordnet wird. Betrachten wir dieses Vorgehen aus Sicht der Zentren, so werden Zentrum i alle KGE zugeordnet, die innerhalb seines *Einzugsbereiches*

$$\mathcal{E}(i) = \{x \in \mathbb{R}^2 : d_i(x) - \lambda_i \leq d_j(x) - \lambda_j \forall j \neq i\} \quad (7.30)$$

liegen. Dabei ist $d_i(x)$ die Entfernung von x nach $(Ost_i, Nord_i)$ bezüglich der zugrundegelegten Distanzmessung (siehe 7.1.1). KGE, die auf dem Rand von $\mathcal{E}(i)$ liegen (für die keine echte Ungleichung in (7.30) besteht), können auch einem anderen Zentrum zugeordnet werden.

Es ist klar, daß das Innere und die Ränder der Einzugsbereiche aller Zentren eine Partition der Ebene bilden. Sie kann als eine Gebietsaufteilung der Ebene interpretiert werden (wobei Ränder nicht eindeutig zugewiesen sind).

Man beachte nun, daß diese Gebietsaufteilung nur durch die Lage der Zentren und die Lagrange-Multiplikatoren λ_i , $i \in I$, bestimmt ist. Das Subgradienten-Verfahren verändert die Multiplikatoren, und verschiebt so die Grenzen der Einzugsbereiche. Dabei ist die Tendenz, die Ebene so aufzuteilen, daß die Gesamtgröße der KGE in allen Einzugsbereichen etwa gleich ist.

Wir möchten diese Betrachtung noch ein wenig vertiefen. Die geometrische Gestalt einer solchen Gebietsaufteilung der Ebene wird wesentlich durch die Art der Distanzmessung, die Definition von $d_i(x)$ also, bestimmt (siehe Abbildung 7.5).

Haben zwei Zentren i und j angrenzende Einzugsbereiche, so ist die gemeinsame

Grenze durch die Gleichung

$$d_i(x) - \lambda_i = d_j(x) - \lambda_j \quad (7.31)$$

gegeben. Für die euklidische Distanz ist dies die Gleichung eines Hyperbelastes, denn gerade die Punkte x gleicher (euklidischer) Entfernungsdifferenz von $(Ost_i, Nord_i)$ und $(Ost_j, Nord_j)$ erfüllen (7.31).¹⁷ Die Grenzen der Einzugsbereiche bestehen daher aus Abschnitten von Hyperbelästen.

Wird dagegen die quadrierte euklidische Distanz zugrundegelegt, so sind die sich aus Gleichung 7.31 ergebenden Grenzen Geradenabschnitte, also Strecken. Die folgende Eigenschaft der Aufteilung der Ebene bei dieser Art der Distanzmessung zeigt, warum sie gerade so attraktiv ist.

Proposition 7.16

Für quadrierte euklidische Distanzen sind die Einzugsbereiche $\mathcal{E}(i)$ (eventuell entartete) konvexe Polygone.

Beweis. $\mathcal{E}(i)$ ist die Menge aller Punkte $x = (x_1, x_2)$, die das Ungleichungssystem

$$d_i(x) - \lambda_i \leq d_j(x) - \lambda_j \quad \text{für alle } j \in I - \{i\}$$

erfüllen. Für quadrierte euklidische Distanzen $d_i(x)$ und $d_j(x)$ fallen die Quadrate von x_1 und x_2 heraus, so daß wir lineare Ungleichungen in x_1 und x_2 erhalten. Die dadurch definierte Teilmenge des \mathbb{R}^2 ist ein konvexes Polygon, das leer oder eine Strecke sein kann. \square

7.3.3 Beziehung zur Vorgehensweise in 7.2

Sowohl das Verfahren von Zoltners und Sinha [115] als auch die Vorgehensweise in 7.2 haben als Ausgangspunkt das ganzzahlige Programm ZP0 (7.26). Aufgrund von (7.26c) hat es im allgemeinen keine zulässigen Lösungen. Während in 7.2 nun der Weg beschritten wurde, die lineare Relaxation TP (7.2) zu lösen und dann durch Splitauflösung (also Runden der gebrochenen Variablen) eine ganzzahlige Lösung zu bestimmen, die möglichst balanciert ist, lösen Zoltners und Sinha [115] approximativ den Lagrange-Dual LD (7.28), der sich durch Relaxation von (7.26c) ergibt. Unter den Lösungen der Lagrange-Subprobleme (diese sind ganzzahlig) wird dabei die am besten balancierte gewählt.

¹⁷Speziell für $\lambda_i = \lambda_j$ erhalten wir die Mittelsenkrechte der beiden Zentrenorte. Ist die Differenz zwischen λ_i und λ_j betragsmäßig größer als die Entfernung der Zentrenorte, so ist der Hyperbelast nicht definiert, andernfalls schneidet er die Verbindungsstrecke der Zentrenorte senkrecht.

Damit haben beide Vorgehensweisen einen gemeinsamen Ausgangspunkt und ein gemeinsames Ziel. Es stellt sich daher die Frage nach einem Vergleich. Wie wird das Ziel möglichst balancierter Bezirke ohne gesplittete KGE besser erreicht? Im folgenden soll gezeigt werden, daß die Vorteile bei der Methode „TP mit optimaler Splitauflösung“ nach 7.2 liegen.

Die Basis des Vergleichs beider Verfahren ist der folgende Satz, der für Leser, die mit Lagrange-Relaxation vertraut sind, keine Überraschung darstellt.

Satz 7.17

Die optimalen dualen Variablen (im Sinne der linearen Programmierung) λ_i^* , $i \in I$, zu Nebenbedingung (7.2c) in TP lösen den Lagrange-Dual LD.

Beweis. Es sei $X = \{(x_{iv}) \in \{0, 1\}^{p \times n} : \sum_i x_{iv} = 1 \text{ für alle } v\}$. (Nach (7.27) wird $C(\lambda)$ für gegebene Multiplikatoren $\lambda = (\lambda_i)$ durch Minimierung über X berechnet.) Aus einem zentralen Satz über den Lagrange-Dual (siehe Nemhauser und Wolsey [79, Th. 6.2, S. 327]) folgt, daß LD äquivalent zum Problem

$$\min \sum_{i,v} d_{iv} w_v x_{iv} \tag{7.32a}$$

$$\text{unter} \quad \sum_{v \in V} w_v x_{iv} = \mu \quad (i \in I) \tag{7.32b}$$

$$(x_{iv}) \in \text{conv}(X) \tag{7.32c}$$

ist. Dies ist insbesondere für endliches X (was hier gegeben ist) ein lineares Programm, und dessen optimale duale Variablen zu (7.32b) lösen LD.

Die konvexe Hülle $\text{conv}(X)$ für X ist hier einfach zu beschreiben, es ist¹⁸

$$\text{conv}(X) = \{(x_{iv}) \in \mathbb{R}^{p \times n} : \sum_i x_{iv} = 1 \text{ für alle } v, x_{iv} \geq 0\}.$$

Damit ist (7.32) gerade das Programm TP (7.2) und es folgt die Behauptung. \square

Satz 7.17 wirft die Frage auf, warum Zoltners und Sinha [115] den Lagrange-Dual mit dem Subgradienten-Verfahren approximativ lösen, wo doch das Transport-Problem TP effizient gelöst werden kann. Wäre es nicht besser, direkt mit den optimalen Lagrange-Multiplikatoren zu arbeiten? Die folgende Beobachtung zeigt auf, welche Problematik dabei besteht.

Beobachtung 2. Für optimale duale Variablen λ_i^* , $i \in I$, zu Nebenbedingung (7.2c) in TP gilt: Für jede gesplittete KGE $v \in V^S$ in der Lösung von TP wird das Minimum von

$$d_{iv} - \lambda_i^*, \quad i \in I,$$

¹⁸Dies folgt zum Beispiel aus der totalen Unimodularität der Koeffizientenmatrix der X beschreibenden Gleichungen.

für alle Zentren i angenommen, die zu v in der Splitadjazenz benachbart sind. Insbesondere ist das minimierende i nicht eindeutig bestimmt.

Beweis. Die Behauptung folgt aus dem Satz vom komplementären Schlupf. Zunächst besteht die duale Nebenbedingung

$$\lambda_i + w_v \sigma_v \leq d_{iv}. \quad (7.33)$$

(σ_v seien die dualen Variablen zu den Nebenbedingungen (7.2b) von TP.) Daraus ergibt sich im Optimum (für jede KGE v) $d_{iv} - \lambda_i^* \geq w_v \sigma_v^*$ für alle i .

Nun ist für Basisvariablen x_{iv} in der optimalen Lösung von TP (dazu gehören alle Variablen mit Wert echt zwischen 0 und 1, denen die Kanten der Splitadjazenz entsprechen) die Ungleichung (7.33) mit Gleichheit erfüllt. Dies zeigt die Behauptung. \square

Beobachtung 2 macht deutlich, daß das von Zoltners und Sinha [115] vernachlässigte Problem der Mehrdeutigkeit bei der Lösung des Lagrange-Subproblems (siehe Beobachtung 1 in 7.3.1) gerade dann zu Tage tritt, wenn dabei mit den optimalen Lagrange-Multiplikatoren gearbeitet wird. Die an sich naheliegende Regel, Mehrdeutigkeiten per Zufallsentscheidung aufzulösen, entspricht damit einer zufälligen Splitauflösung.

Anders formuliert heißt das: Die Problematik, die auftretenden Mehrdeutigkeiten im Lagrange-Subproblem bei optimalen Multiplikatoren zu behandeln, ist äquivalent zur Problemstellung der Splitauflösung, wie sie in Abschnitt 7.2.2 formuliert wurde. Bei Verwendung von λ_i^* steht man somit vor der Frage der optimalen Splitauflösung in ihrer ganzen Komplexität.

Das damit aufgeworfene Problem ist jedoch für Zoltners und Sinha [115] tatsächlich vernachlässigbar, da sie *nicht* mit optimalen Lagrange-Multiplikatoren arbeiten. Vielmehr werden diese durch das Subgradienten-Verfahren nur approximiert. Folglich werden Mehrdeutigkeiten in aller Regel nicht auftreten, und es ist nicht nötig, sich Gedanken über deren systematische Behandlung zu machen. Es kann davon ausgegangen werden, daß in jeder Iteration die λ_i eindeutig eine Gebietsaufteilung als Lösung des Lagrange-Subproblems determinieren.

Haben wir mit der eingehenden Untersuchung des Splitauflösungsproblems in 7.2 also völlig Unnötiges geleistet? Ist der von Zoltners und Sinha [115] eingeschlagene Weg der einfachere und elegantere?

Die Antwort darauf ergibt sich, wenn man das Verhalten des Subgradienten-Verfahrens genauer in den Blick nimmt. Bis auf „Einschwingvorgänge“ in den ersten Iterationen schwanken die Multiplikatoren — darauf zielt das Verfahren! — nahe um ihren optimalen Wert. Durch eine kleine Abweichung werden zwar die Mehrdeu-

tigkeiten im Lagrange-Subproblem vermieden, andererseits wird die dabei erzeugte Gebietsaufteilung in aller Regel unter denen zu finden sein, die sich durch eine Splitauflösung aus der Lösung von TP gewinnen lassen. Bezeichnen wir die Menge der durch Splitauflösung erreichbaren Gebietsaufteilungen mit \mathcal{G}^S , so läßt sich die Suche nach balancierten Gebietsaufteilungen per Subgradienten-Verfahren dahingehend interpretieren, daß (ganz überwiegend) Gebietsaufteilungen aus \mathcal{G}^S erzeugt und hinsichtlich des Balance-Ziels bewertet werden.

Optimale Splitauflösung ist die Bestimmung der besten Gebietsaufteilung in \mathcal{G}^S hinsichtlich dieses Ziels und wurde in 7.2 durch Baumzerlegung erreicht. Das Subgradienten-Verfahren dagegen leistet vor allem ein Durchsuchen der Menge \mathcal{G}^S , in der Hoffnung, durch ausreichend lange Suche eine gut balancierte Gebietsaufteilung zu finden.

Damit kann optimale Splitauflösung als der direkte Weg zum Ziel angesehen werden, während die Vorgehensweise von Zoltners und Sinha [115] mehr einer etwas unsystematischen Suche entspricht.

Betrachtet man dies unter Effizienzgesichtspunkten, wird der Unterschied noch deutlicher. Während die optimale Splitauflösung aufgrund der in Praxis-Problemen kleinen Grade in der Splitadjazenz auch für große Probleme in Sekundenbruchteilen vorgenommen werden kann, müssen für das Subgradienten-Verfahren Dutzende bis Hunderte Iterationen ausgeführt werden. Bei großen Problemen erfordert dies durchaus Rechenzeiten im Minutenbereich.¹⁹ Zwar wird dabei häufig die Gebietsaufteilung gefunden, die aus optimaler Splitauflösung hervorgeht, doch hat man auch hierfür keine Gewähr.

Bemerkung 7.18

Der Leser mag die voranstehenden Ausführungen möglicherweise etwas unpräzise finden. Die stets auftauchende „im allgemeinen“-Einschränkung hat jedoch einen einfachen Grund. Ein Satz mit der Aussage

Die Vorgehensweise „Transportproblem und Splitauflösung“ nach 7.2 bestimmt nie eine schlechter balancierte Gebietsaufteilung, als sie durch das Subgradienten-Verfahren erzielt werden kann.

ist nicht korrekt. Es sind durchaus Beispiele konstruierbar, in denen es außerhalb der Menge \mathcal{G}^S eine besser balancierte Gebietsaufteilung gibt, als alle darin liegenden, und es lassen sich Lagrange-Multiplikatoren angeben, für die diese Gebietsaufteilung Lösung des Lagrange-Subproblems ist. Neben der Tatsache, daß solche Beispiele eben konstruiert sind, ist es aber vor allem nicht wahrscheinlich, daß das

¹⁹Solche Rechenzeiten wurden beim Experimentieren mit diesen Verfahren in GAMOR gemessen.

Subgradienten-Verfahren in einem solchen Fall gerade die „richtigen“ Multiplikatoren trifft. In mehreren hundert gerechneten Testproblemen tauchte kein einziger Fall auf, der eine Ausnahme zum angeführten „Satz“ darstellt.

Kapitel 8

Modelle und Verfahren für Probleme auf Netzwerken

Wie in Kapitel 2 deutlich wurde, ist der geographische Zusammenhang der gebildeten Bezirke eine wesentliche Forderung in Problemen der Gebietsaufteilung. Ein Bezirk, der aus mehreren, nicht zusammenhängenden, Teilen besteht ist — je nach Anwendung — unerwünscht oder sogar unzulässig. Die Modelle und Verfahren des vorangehenden Kapitels können den Zusammenhang der gebildeten Bezirke nicht garantieren, da sie keine Information über Nachbarschaft von KGE einbeziehen. In diesem Kapitel werden Modelle vorgestellt, in denen die KGE Ecken eines Graphen sind und die KGE jedes gebildeten Bezirks einen zusammenhängenden Subgraphen induzieren.

8.1 Topologische Ausgangssituation

8.1.1 KGE als Ecken eines Graphen

Werden die KGE als Ecken eines ungerichteten Graphen $G = (V, E)$ aufgefaßt, so wird dadurch eine Relation, gegeben durch die Kantenmenge E , zwischen den KGE zum Ausdruck gebracht. Die naheliegendste Interpretation einer solchen Relation ist die Nachbar-Beziehung: Eine Kante $e = (u, v) \in E$ modelliert die Eigenschaft, daß die KGE u und v benachbart sind.¹

Die Vorstellung von Nachbarschaft muß dabei zunächst abstrakt gesehen werden. Zum Beispiel könnte sie zum Ausdruck bringen, daß u und v als Flächen ein Stück gemeinsame Grenze haben. Jedoch kann eine solche, auf einem rein geographi-

¹Wir verwenden die gerichtete Schreibweise für Kanten als Ecken-Paar, fassen sie hier jedoch als ungerichtet auf. Folglich wird nicht zwischen den Kanten (u, v) und (v, u) unterschieden.

schen Kriterium beruhende Nachbarschaftsdefinition für eine Anwendungssituation ungeeignet sein, in der das Bereisen der Bezirke im Vordergrund steht. Wenn die gemeinsame Grenze zweier KGE ein Flußlauf ohne Brücke ist, wird man von ihrer Nachbarschaft wenig Nutzen haben. Benachbartsein sollte dann durch das Vorhandensein einer geeigneten Verkehrsverbindung zwischen den KGE definiert sein.

Diese Überlegung zeigt, daß der Graph G unter Berücksichtigung der zugrundeliegenden Anwendung gewonnen werden muß. Dies kann bei einer großen Menge an KGE einen nicht unerheblichen Aufwand in der Datenaufbereitung bedeuten.²

In den meisten Fällen wird G zumindest im groben eine geographische Nachbarschaft der KGE widerspiegeln. Dies hat eine aus modelltheoretischer Sicht interessante Eigenschaft zur Konsequenz: Der Graph G (wir bezeichnen ihn im folgenden als *Nachbarschaftsgraph*) ist im allgemeinen planar. Deswegen ist die Annahme eines *dünnen* Graphen G mit $|E| = O(|V|)$ gerechtfertigt, mit einem kleinen Quotienten aus Kantenzahl und Eckenzahl. Für die Laufzeitabschätzung von Algorithmen kann aus dieser Annahme oft ein Vorteil gezogen werden.

Zusätzlich kann davon ausgegangen werden, daß eine Einbettung von G in die Ebene bekannt ist. In 7.1.1 wurde der Ort einer KGE als Punkt in der Ebene \mathbb{R}^2 angenommen. Werden die Ecken des Nachbarschaftsgraphen nun durch diese Punkte und die Kanten (zwischen benachbarten KGE) durch ihre Verbindungsstrecken repräsentiert, so erhält man eine im allgemeinen überkreuzungsfreie Einbettung des Graphen in die Ebene.

Bezirkszentren. Auch in Modellen mit Netzwerk-Topologie spielen Zentren von Bezirken eine wichtige Rolle. Im Gegensatz zu ebenen Modellen, in denen die Zentrenorte nicht mit dem Ort von KGE übereinstimmen müssen (siehe 7.1.1), ist es bei Netzwerk-Modellen sinnvoll, die Zentren als Teilmenge der KGE zu beschreiben. Sie sind dann durch eine p -elementige Teilmenge $Z \subset V$ gegeben; jedes Zentrum $i \in Z$ hat einen zugeordneten Bezirk $B(i)$. Dabei wird man die Vorstellung haben, daß das Zentrum selbst Teil dieses Bezirkes ist (also $i \in B(i)$), insbesondere, wenn das Zentrum in der Anwendung zum Beispiel dem Wohnort eines Vertreters entspricht.

²Für das auf Seite 205 gezeigte Beispiel der deutschen Kreise und kreisfreien Städte mit 544 KGE wäre eine manuelle Erstellung von G zu aufwendig gewesen. Da für die KGE aber Umrißdaten in Form von Polygonzügen vorlagen, wurde durch einen geeigneten Algorithmus festgestellt, welche KGE ein Stück gemeinsamer Grenze besitzen und auf diese Weise ein Graph bestimmt, der nur noch eine geringfügige manuelle Nachbearbeitung erforderte. Der Graph hat 1320 Kanten.

8.1.2 Distanzmessung über kürzeste Wege

Zur Bestimmung der Entfernung zweier KGE stehen zunächst die in 7.1.1 vorgestellten Distanzmaße zur Verfügung, die von Koordinatenpaaren $(ost_v, nord_v)$ für die KGE ausgehen. Da mit dem Graphen G nun jedoch mehr Information bereitsteht, ist es sinnvoll, ein Distanzmaß einzubeziehen, das auf kürzesten Wegen in G basiert. Wir bezeichnen es als *KW-Distanz*.

Grundlage dafür ist eine nichtnegative Bewertung der Kanten von G

$$d : E \longrightarrow \mathbb{R}_{\geq 0}.$$

Sie kann zum Beispiel durch die euklidische Entfernung der Koordinaten benachbarter KGE gewonnen werden, oder auch durch eine Abschätzung der Reisezeit zwischen solchen KGE.

Ausgehend von d wird die Distanzmatrix

$$D = (D_{uv})$$

bestimmt, deren Einträge für Paare u, v von KGE die Länge des kürzesten Weges zwischen u und v in G angeben.

Dabei ist allgemein ein (*gerichteter*) *Weg* in einem (gerichteten) Graphen $G = (V, E)$ eine Folge e_1, \dots, e_k von Kanten $e_i = (u_i, v_i) \in E$, mit $v_i = u_{i+1}$ für $i = 1, \dots, k-1$, wobei u_1, \dots, u_k paarweise verschieden sind.³ Ein Weg von u nach v ist gegeben, wenn $u_1 = u$ und $v_k = v$ ist. Die Länge eines Weges ist $\sum_{i=1}^k d(e_i)$, und für einen kürzesten Weg von u nach v schreiben wir $KW[u, v]$. Wir setzen voraus, daß die Dreiecksungleichung erfüllt, das heißt für $(u, v) \in E$ ist $KW[u, v] = (u, v)$.

Die Distanzmatrix D ist definiert durch

$$D_{uv} = \sum (d(e) : e \in KW[u, v]).$$

Aufgrund der Dreiecksungleichung ist $D_{uv} = d(u, v)$ für $(u, v) \in E$. Wir können also D als Fortsetzung von d auf alle Paare von Ecken ansehen.

Ist G wie in 8.1.1 ein ungerichteter Graph, so können in der Definition des Weges die Kanten „in beiden Richtungen“ benutzt werden. Die Länge d einer Kante hängt dabei nicht von der Richtung ab und die Matrix D ist folglich symmetrisch.

³Erklärung: Um kürzeste Wege sowohl im ungerichteten wie im gerichteten Fall einzuführen, wird hier „überladene“ Notation verwendet. In gerichteten Graphen sollen kürzeste Wege gerichtet sein. Der hier vornehmlich interessierende Nachbarschaftsgraph der KGE wird jedoch wie in 8.1.1 als ungerichtet aufgefaßt.

Berechnung der Distanzmatrix. Zur Berechnung kürzester Wege gibt es prominente Algorithmen,⁴ am bekanntesten dürfte der Dijkstra–Algorithmus sein. Damit werden die kürzesten Wege von einer festen Ecke zu allen anderen Ecken bestimmt, mithin eine Zeile der Matrix D . In der Implementierung mit Binär–Heap ist der Aufwand dafür $O(|E| \log |V|)$. Zur Berechnung von D sind also $O(|V||E| \log |V|)$ Schritte nötig.

Aufgrund der in 8.1.1 diskutierten Voraussetzung $|E| = O(|V|)$ kann der Aufwand zur Berechnung von D mit $O(|V|^2 \log |V|)$ angenommen werden. Im Fall eines planaren Graphen G kann man mit spezialisierten Verfahren den logarithmischen Faktor loswerden (Frederickson [37]). Jedoch ist der Dijkstra–Algorithmus einfach zu implementieren und $|V|$ wird wenige Tausend selten übersteigen, so daß der Einsatz eines solchen Spezialverfahrens in der Praxis wohl nicht nötig ist.

8.1.3 Modellierung der Zusammenhangsbedingung

Die entscheidende Motivation für die Untersuchung von Gebietsaufteilungsmodellen mit Netzwerk–Topologie ist die Forderung nach geographisch zusammenhängenden Bezirken. Sie wird modelliert durch die Bedingung, daß die KGE jedes Bezirks $B \subset V$ einen zusammenhängenden Subgraphen von G induzieren. Diese graphentheoretisch elegante Formulierung ist bei der Umsetzung in Optimierungsmodellen jedoch ziemlich unhandlich. Im folgenden diskutieren wir einige Möglichkeiten dafür.

8.1.3.1 Schnittformulierung des Zusammenhangs

Eine gängige Umsetzung der Zusammenhangsbedingung basiert auf Schnitten im Graphen. Für den Fall der Modellierung zusammenhängender Bezirke, läßt sie sich folgendermaßen realisieren. Ist $U \subset V$, $U \neq \emptyset$, so muß eine zusammenhängende Teilmenge B von V , die nicht ganz in U liegt, mindestens eine Ecke aus der Nachbarschaft $\dot{N}(U)$ enthalten. Dabei ist die Nachbarschaft definiert durch

$$\dot{N}(U) = \{v \in V - U : \text{es gibt eine Kante } (v, u) \in E \text{ oder } (u, v) \in E \text{ mit } u \in U\}.$$

Hier wird also der Schnitt zwischen U und seinem Komplement zugrundegelegt.

Eine Modellierung des Zusammenhangs durch Nebenbedingungen in den Variablen

$$x_{iv} = \begin{cases} 1 & \text{KGE } v \text{ gehört zum } i\text{-ten Bezirk,} \\ 0 & \text{sonst,} \end{cases}$$

⁴Eine umfassende Darstellung ist zum Beispiel in Ahuja et al. [4] zu finden.

lautet damit

$$\sum_{v \in \dot{N}(U)} x_{iv} \geq \left(-|U| + \sum_{u \in U} x_{iu} \right) + \frac{1}{|V|} \left(-|U| + \sum_{u \in V} x_{iu} \right). \quad (8.1)$$

Dabei gibt es für jedes Paar aus einer Teilmenge $U \subset V$, $U \neq \emptyset$ und einem $i \in I = \{1, \dots, p\}$ eine Nebenbedingung. Die rechte Seite ist genau dann positiv, wenn der i -te Bezirk alle Ecken aus U und insgesamt mehr als $|U|$ Ecken enthält. Dann aber muß auch die linke Seite positiv (und ganzzahlig) sein. Offenbar reicht es aus, (8.1) für zusammenhängende U zu formulieren.⁵

Ein gravierender Nachteil dieser Zusammenhang-Modellierung ist die exponentielle Anzahl an Nebenbedingungen, die sie erfordert. Nur in Verfahren, die mit einer impliziten Problemformulierung arbeiten, läßt sie sich einsetzen. Außerdem ist nicht klar, wie gut die algorithmischen Eigenschaften dieser Nebenbedingungen sind, mit anderen Worten, wie gut sie die zulässigen Lösungen bei Relaxation der Ganzzahligkeitsbedingung für die x_{iv} charakterisieren.

8.1.3.2 Flußformulierung des Zusammenhangs

Eine andere gängige Modellierung von Zusammenhang, die das Problem der exponentiellen Anzahl von Nebenbedingungen umgeht, basiert auf Mehrgüter-Flüssen. Da wir den Ansatz nicht weiterverfolgen, beschränken wir uns auf eine Skizzierung.

Der Einfachheit halber gehen wir von einer gegebenen Menge $Z \subset V$, $|Z| = p$, an Zentren aus. Ein Bezirk $B(i) \subset V$ mit Zentrum $i \in Z$ ist zusammenhängend, wenn für jede KGE $v \in B(i)$ in G ein Weg nach i existiert, der ganz in $B(i)$ verläuft. Dieser Weg wird nun durch den Fluß eines Gutes von v nach i charakterisiert.

Der Fluß findet in einem Netzwerk $\tilde{G} = (V, \tilde{E})$ mit Eckenmenge V statt, in welchem jedes Paar in G adjazenter Ecken durch zwei entgegengesetzt gerichtete Kanten verbunden ist. Die Zugehörigkeit von v zu $B(i)$ wird durch die Binärvariable x_{iv} beschrieben. Zusätzlich gibt es zu jeder Kante e von \tilde{G} eine (ebenfalls binäre) Flußvariable x_{iv}^e . Zwei Gruppen von Nebenbedingungen sind nun nötig. Erstens die Inkompressibilitätsbedingung für den Fluß des zu v und i gehörenden Gutes an der Ecke u ,

$$\sum_{u': (u, u') \in \tilde{E}} x_{iv}^{(u, u')} - \sum_{u': (u', u) \in \tilde{E}} x_{iv}^{(u', u)} = \begin{cases} 0 & \text{für } u \neq i, v, \\ +x_{iv} & \text{für } u = v, \\ -x_{iv} & \text{für } u = i. \end{cases}$$

⁵Drexel und Haase [31] verwenden eine einfachere Nebenbedingung als (8.1); sie machen dabei jedoch die Voraussetzung, daß jeder Bezirk sein Zentrum enthält.

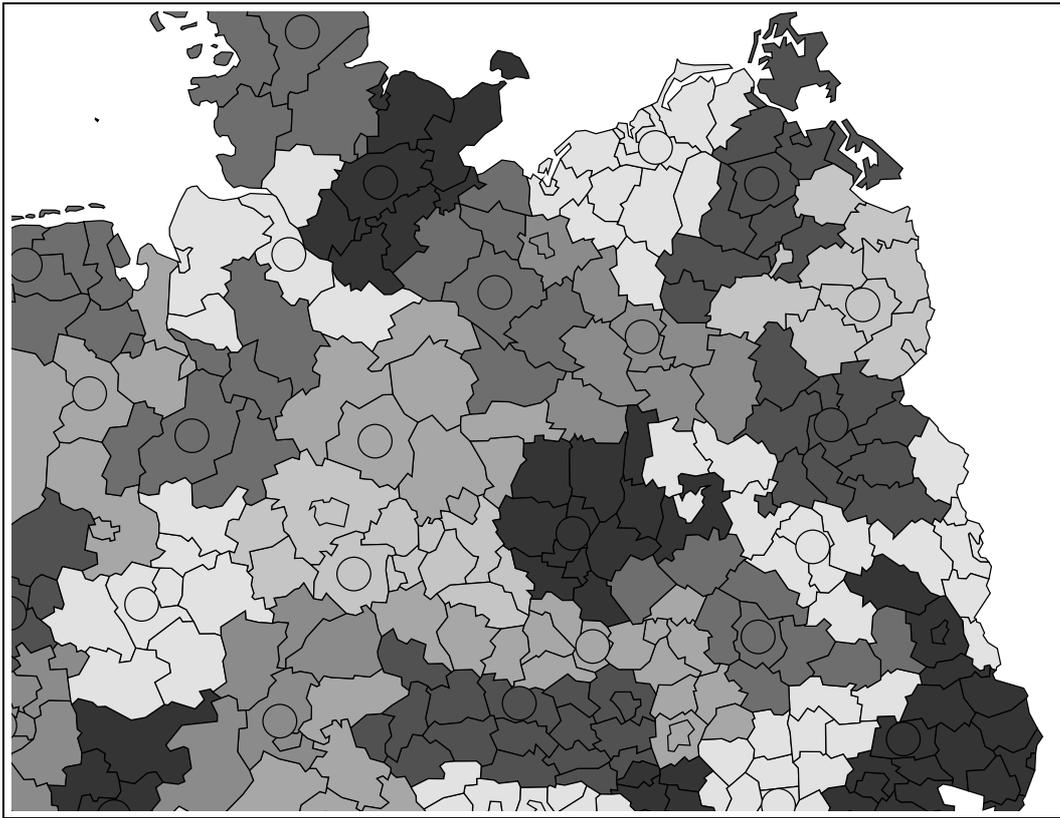


Abbildung 8.1: Zusammenhängende Bezirke; zum Teil entsprechen sie nicht dem Kompaktheitsziel.

und zweitens eine Bedingung, die dafür sorgt, daß die Ecke u zu $B(i)$ gehört, wenn durch sie das Gut einer anderen Ecke v nach i fließt,

$$\sum_{u': (u', u) \in \tilde{E}} x_{iv}^{(u', u)} \leq x_{iu}.$$

Gegenüber der Formulierung in 8.1.3.1 kommt dieses Modell mit einer polynomialen Anzahl von Nebenbedingungen aus. Jedoch ist die Variablenzahl von der Ordnung $\Theta(|E||V|p)$, und die Anzahl an Nebenbedingungen ist $\Theta(|V|^2p)$. Auch für Probleme mit dünnem Graphen liegt damit eine etwa kubisch wachsende Anzahl Binärvariablen und Nebenbedingungen vor, für große Probleme ($|V| \approx 1000$) ist dies nicht handhabbar.

8.1.3.3 Verstärkte Zusammenhangsformulierungen

Neben den algorithmischen Problemen, die die bisher vorgestellten Modellierungen von Zusammenhang aufwerfen, ist auch die Frage zu stellen, ob sie den Erfordernissen

der Praxis gerecht werden. Denn Zusammenhang der Bezirke allein reicht in der Regel zur Charakterisierung guter Gebietsaufteilungen nicht aus. So sind in Abbildung 8.1 zwar alle Bezirke geographisch zusammenhängend, doch zum Beispiel ist rechts im Bild ein mäanderförmiger Bezirk zu erkennen, der im Hinblick auf das Kompaktheitsziel vermutlich nicht akzeptabel wäre. In der Praxis sind beide Kriterien wichtig, Zusammenhang und Kompaktheit. Zusammenhang der Bezirke als Teilmengen der Ecken von G im graphentheoretischen Sinn ist daher in aller Regel nicht ausreichend.

Mehrfach wurde deshalb in der Literatur vorgeschlagen, die Zusammenhangsforderung zu verstärken (Mehrotra et al. [71], Zoltners und Sinha [115]). Wir stellen diese Konzepte im folgenden vor.

KW–Zusammenhang. Hierbei wird von gegebenen Bezirkszentren $Z \subset V$, $|Z| = p$ ausgegangen. Zu jedem $i \in Z$ wird ein Baum $T(i) = (V, E(i))$ definiert. $E(i)$ ist die Vereinigung der kürzesten Wege aller Ecken $v \in V$ nach i , das heißt,

$$E(i) = \bigcup_{v \in V - \{i\}} KW[v, i]. \quad (8.2)$$

Daß $T(i)$ Baumstruktur hat, ist bei Eindeutigkeit der kürzesten Wege offensichtlich, andernfalls können geeignete Wege gewählt werden. Der Baum $T(i)$ wird nun als Wurzelbaum mit Wurzel i aufgefaßt.

Ein Bezirk $B(i)$ zu i ist nur dann zulässig, wenn $B(i)$ einen Teilbaum von $T(i)$ induziert, der i enthält. Neben dem graphentheoretischen Zusammenhang ist hier ein Kompaktheitsaspekt einbezogen, denn mit einer KGE v müssen auch alle KGE, die auf dem kürzesten Weg $KW[v, i]$ zum Zentrum i liegen, Teil des Bezirkes sein. Im folgenden bezeichnen wir solche Bezirke als *KW–zusammenhängend*.

Der Vorteil dieser verstärkten Formulierung ist ein zweifacher. Erstens entspricht sie mehr der Vorstellung guter Bezirke und zweitens, und das ist wesentlich, ist sie sehr einfach zu modellieren. Dazu reichen die Nebenbedingungen

$$x_{iv} \leq x_{i, \text{pre}_i(v)} \quad (i \in Z, v \in V - \{i\}) \quad (8.3)$$

aus, wobei $\text{pre}_i(v)$ den Vorgänger von v in $T(i)$ angibt.

Präzedenz–Zusammenhang. Auf der anderen Seite wurde das Konzept KW–zusammenhängender Bezirke auch als zu einschränkend empfunden (Zoltners und Sinha [115]). Eine Abschwächung, die immer noch stärker als bloßer Zusammenhang ist, wird erzielt, wenn an die Stelle des Wurzelbaumes $T(i)$ eine Präzedenz mit Eckenmenge V und Wurzel i tritt.⁶ Die Kantenmenge dieser Präzedenz wird aus einer

⁶Eine Präzedenz ist ein kreisfreier gerichteter Graph. Besitzt sie eine Wurzel, ist diese von allen Ecken aus durch einen gerichteten Weg erreichbar.

Teilmenge von E gebildet, indem deren Elementen eine Richtung „hin zur Wurzel“ aufgeprägt wird. Im Unterschied zum Wurzelbaum ist die Vorgängerbeziehung in einer Präzedenz nicht eindeutig. Daher wird (8.3) ersetzt durch

$$x_{iv} \leq \sum_{u \in \text{Pre}_i(v)} x_{iu} \quad (i \in Z, v \in V - \{i\}) \quad (8.4)$$

wobei $\text{Pre}_i(v)$ die Menge der (direkten) Vorgänger von v in der zur Wurzel i gehörenden Präzedenz bezeichnet. Mit v muß also mindestens ein Vorgänger von v zum Bezirk $B(i)$ gehören.

Eine solche Präzedenz verwenden Mehrotra et al. [71]. In ihrem Modell sind alle Kanten von G mit Eins bewertet ($d(e) = 1$ für alle $e \in E$). Aus (8.2) ergibt sich daher kein Baum (da die kürzesten Wege vielfach nicht eindeutig sind), sondern, durch Aufprägen der Richtung hin zu i auf die Kanten, eine Präzedenz. Mehrotra et al. verwenden in ihrem Modell die Nebenbedingungen (8.4).

Gerüst–Zusammenhang. Die Konzepte von KW– und Präzedenz–Zusammenhang sind bezüglich gegebenen Zentren $Z \subset V$ definiert. Wenn solche Zentren nicht ohne weiteres verfügbar sind, ist es nützlich, eine Zusammenhangsformulierung zu haben, die ohne sie auskommt, andererseits aber algorithmisch einfacher zu handhaben ist, als die Formulierungen von 8.1.3.1 und 8.1.3.2. Die Idee hierzu ist recht naheliegend und ermöglicht — im Rahmen unserer Arbeit besonders interessant — die Anwendung gleichmäßiger Baumzerlegung. Dieselbe Idee wird von Maravalle und Simeone [66] im Kontext des *regional clustering* (siehe 2.3.3) verwendet.

Man wähle im Graphen G in geeigneter Weise ein Gerüst $T = (V, E')$, mit $E' \subset E$. Ein Bezirk, dessen zugehörige KGE einen Teilbaum von T induzieren, ist *T–zusammenhängend*. Unmittelbar einsichtig ist, daß er einem zusammenhängenden Subgraphen von G entspricht. Andererseits sind nicht alle zusammenhängenden Subgraphen auch *T–zusammenhängend*.

Daraus lassen sich nun zwei wesentliche Vorteile ziehen. Erstens kann durch die Wahl von T die Gestalt zulässiger Bezirke beeinflußt werden, so daß Gebietsaufteilungen wie in Abbildung 8.1 vermieden werden. Zweitens können nun Verfahren der Baumzerlegung angewendet werden, um ausgehend von T eine Gebietsaufteilung zu bestimmen. Da das Zerlegen eines Baumes in Teilbäume algorithmisch viel besser zu handhaben ist, als das Zerlegen eines zusammenhängenden Graphen in zusammenhängende Subgraphen (siehe 4.3 und 4.4), eröffnen sich Perspektiven, auch sehr große Gebietsaufteilungsprobleme zu lösen, in denen die Zusammenhangsforderung nicht vernachlässigt werden darf.

Vorschläge für die Auswahl eines Gerüsts T machen wir in 8.3.2. Zuvor untersuchen wir die Auswirkung der Einbeziehung von KW–Distanzen in die Modelle von

Kapitel 7.

8.2 Diskussion der Übertragbarkeit von Verfahren für ebene Probleme

Die in 7.2 und 7.3 vorgestellten Modelle und Verfahren machen nicht explizit die Voraussetzung, daß die einbezogenen Distanzen d_{iv} aus einem ebenen Modell abgeleitet werden. Deshalb erscheint es naheliegend, diese Verfahren auch einzusetzen, wenn mit einem Distanzmaß gearbeitet wird, das aus kürzesten Wegen gewonnen wird (siehe 8.1.2). Bei genauerem Hinsehen zeigt sich jedoch, daß die Art der Distanzmessung das Verhalten der Verfahren stark beeinflussen kann. Dies wird im folgenden diskutiert.

8.2.1 Transportmodell bei KW-Distanz

Das Modell TP (7.2) minimiert eine Zielfunktion, in welche die Entfernung d_{iv} zwischen Zentren i und KGE v eingeht. Durch die Wahl von Zentren $Z \subset V$ und Berechnung der Distanzmatrix D des Graphen G ist es möglich, durch Setzen von

$$d_{iv} := D_{iv} \tag{8.5}$$

diese Entfernung als KW-Distanz zu messen.

Durch Lösen von TP kann dann eine Gebietsaufteilung berechnet werden. Zunächst wird man vielleicht vermuten, daß das Ergebnis sich nur wenig von demjenigen unterscheidet, das man bei quadriert-euklidischen Distanzen erhalten würde.

Beispiel 8.1

Die Abbildungen 8.2 und 8.3 zeigen, daß diese Vermutung unzutreffend ist. Die extrem ungünstige Lage der Zentren in diesem Beispiel wurde gewählt, um den Unterschied deutlich herauszubringen. Für quadriert-euklidische Distanzen ist die Lösung von TP eine Gebietsaufteilung, in welcher die Bezirke eine einigermaßen akzeptable Gestalt haben. Bei Verwendung von KW-Distanzen hingegen bestehen die gebildeten Bezirke aus verstreut durcheinander liegenden Bruchstücken, so daß von einer Gebietsaufteilung eigentlich nicht gesprochen werden kann.

Neben dem Eindruck, daß man aufgrund von Beispiel 8.1 das Modell TP nicht mit auf KW-Distanz basierender Zielfunktion lösen sollte, stellt sich die Frage nach der Ursache für das aufgezeigte Phänomen. Letztere ist in den additiven Beziehungen zwischen den Zielkoeffizienten D_{iv} zu suchen, die sich aus ihrer Gewinnung als Länge kürzester Wege ergibt. Sie bewirken, daß das duale Programm zu TP in hohem

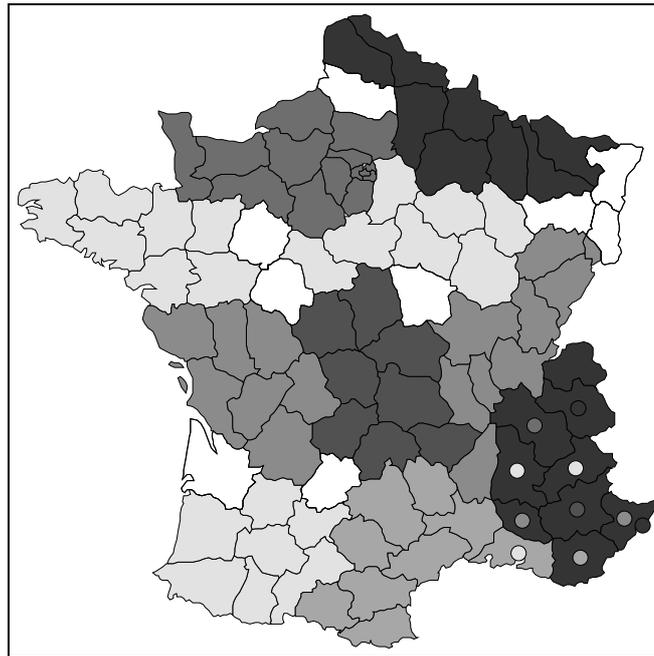


Abbildung 8.2: Lösung von TP für ein Problem mit ungünstigen Zentren bei quadriert-euklidischen Distanzen. (Weiße KGE sind gesplittet.)

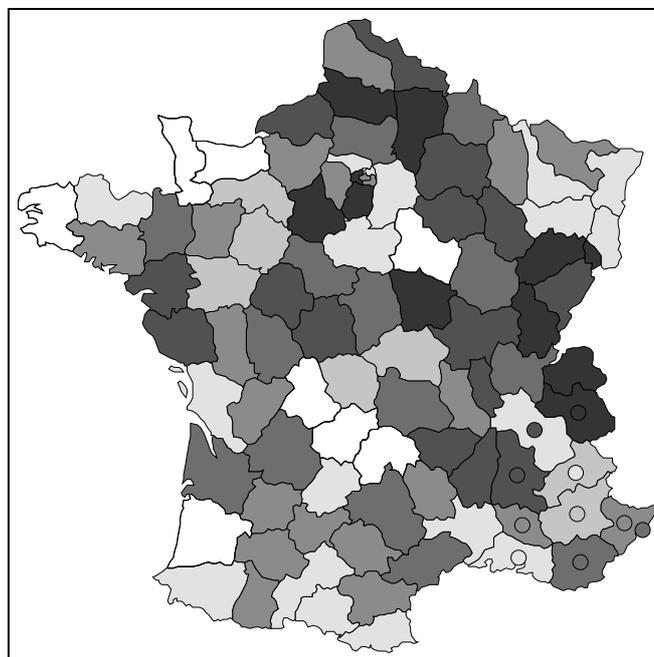


Abbildung 8.3: Lösung des Problems von Abb. 8.2, jedoch KW-Distanzen. Die „Bezirke“ bestehen aus verstreut liegenden Bruchstücken.

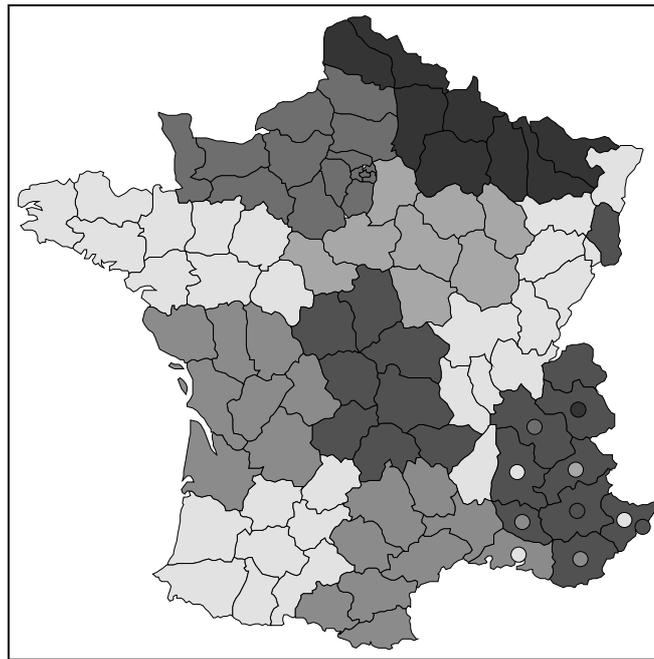


Abbildung 8.4: Lösung mit Subgradienten–Verfahren für ein Problem mit ungünstigen Zentren bei quadriert–euklidischen Distanzen.

Maße degeneriert ist. Für TP bedeutet dies, daß es im allgemeinen viele optimale Basislösungen gibt. Wird dieses Transportproblem mit einem Standard–Löser wie CPLEX optimiert, so besteht kein Einfluß auf die Auswahl der zurückgegebenen Lösung.

Aus diesem ergibt sich aber auch ein Hinweis, wie man dem Problem zu Leibe rücken kann. Duale Degeneriertheit bedeutet eine starke Abhängigkeit zwischen den Spalten des primalen Problems. TP hat zuviele Variablen, wenn mit KW–Distanzen gearbeitet wird. In Abschnitt 8.3 wird mit einem Min–Kosten–Flußproblem, das in der Regel viel weniger Variablen hat, diese Schwierigkeit behoben.

8.2.2 Lagrange–Relaxation bei KW–Distanz

In 7.3 wurde die Berechnung einer Gebietsaufteilung über das Subgradienten–Verfahren basierend auf Lagrange–Relaxation vorgestellt. Der Verwendung von KW–Distanzen in der Zielfunktion (7.26a) beziehungsweise (7.27a) gemäß (8.5) steht zunächst nichts im Wege. Doch wie im Fall des Transportmodells kann diese Entscheidung starke Auswirkungen auf die Gestalt der Lösung haben.

Beispiel 8.2

Die Abbildungen 8.4 und 8.5 zeigen die Lösungen für Beispiel 8.1, die vom Sub-

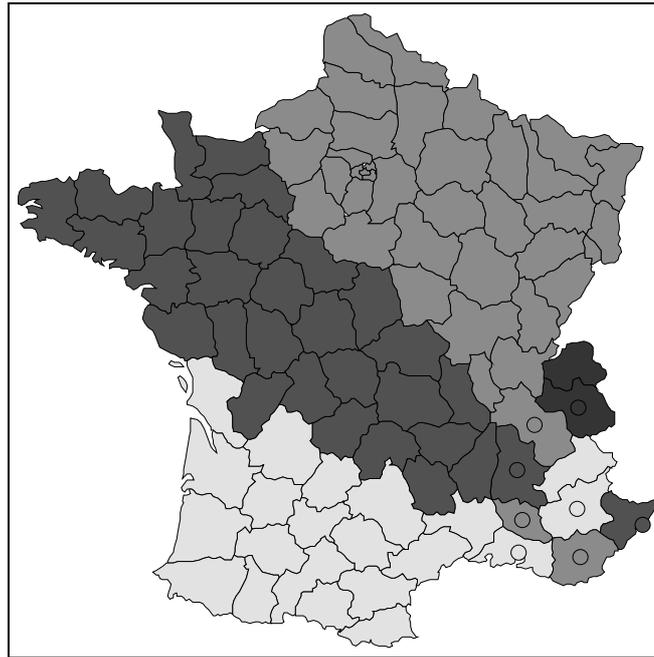


Abbildung 8.5: Lösung des Problems von Abb. 8.4, jedoch KW-Distanzen. Die KGE sind im wesentlichen auf drei Bezirke verteilt. Es sind nur acht Zentren zu sehen, da zwei Bezirke leer sind.

gradienten-Verfahren geliefert werden; sowohl für quadriert-euklidische als auch für KW-Distanzen. Die erste Gebietsaufteilung (Abb. 8.4) enthält einen unzusammenhängenden Bezirk. Da hier der Nachbarschaftsgraph G in die Berechnung nicht einging, ist dies nicht weiter überraschend. Die Notwendigkeit von Modellen, die G einbeziehen, wird jedoch erneut deutlich.

Die auf der Basis von KW-Distanz berechnete Gebietsaufteilung (Abb. 8.5) ist wie die in Abbildung 8.3 unbrauchbar. Drei der zehn zu bildenden Bezirke enthalten nahezu alle KGE und zwei Bezirke sind leer.

Die Ursache für das in Beispiel 8.2 beobachtete Verhalten wird in folgendem Satz deutlich.

Satz 8.1

Es werde das Lagrange-Subproblem (7.27) mit KW-Distanz gemäß (8.5) gelöst. Falls für alle KGE $v \in V$ das minimierende Zentrum $i \in Z$ eindeutig bestimmt ist, sind die gebildeten Bezirke leer oder KW-zusammenhängend.

Beweis. Die KGE $v \in V$ sei dem Zentrum i zugeordnet. Sei u die auf v folgende Ecke in $KW[v, i]$. Es genügt zu zeigen, daß auch u dem Zentrum i zugeordnet wird.

Nach Voraussetzung ist

$$D_{iv} - \lambda_i < D_{jv} - \lambda_j \quad \text{für } j \neq i.$$

Ferner ist $D_{iv} = D_{iu} + d(u, v)$ und $D_{jv} \leq D_{ju} + d(u, v)$. Daraus folgt

$$D_{iu} + d(u, v) - \lambda_i < D_{ju} + d(u, v) - \lambda_j \quad \text{für } j \neq i,$$

und dies zeigt die Behauptung. □

Satz 8.1 geht auf Zoltners und Sinha [115] zurück. Dort wird er ohne die Eindeutigkeitsvoraussetzung „bewiesen“, diese ist jedoch unverzichtbar. Andernfalls muß durch eine geeignete Vorgehensweise bei der Auflösung von Mehrdeutigkeiten für den Zusammenhang der gebildeten Bezirke gesorgt werden.

Die extreme Lage der Zentren in Beispiel 8.2 erlaubt nicht die Bildung von „vernünftigen“ KW-zusammenhängenden Bezirken; dies erklärt die Gebietsaufteilung in Abbildung 8.5. Wenn mit KW-Distanz gearbeitet wird, kann nur bei einer angemessenen Lage der Zentren mit dem Subgradienten-Verfahren eine brauchbare Gebietsaufteilung erzeugt werden. Dann ist auch Satz 8.1 nützlich, da er den Zusammenhang der gebildeten Bezirke gewährleistet. Da das Subgradienten-Verfahren die optimalen Lagrange-Multiplikatoren nur approximativ berechnet, ist die Eindeutigkeitsvoraussetzung des Satzes 8.1 im allgemeinen erfüllt.

8.2.3 Zusammenfassung

In den vorangehenden Abschnitten wurde das Verhalten der Verfahren aus 7.2 und 7.3 bei Zugrundelegung von Netzwerk-Topologie diskutiert. Diese Verfahren setzen Zentren voraus, die in diesem Fall durch eine p -elementige Teilmenge der KGE gegeben sind. Die in die Zielfunktion eingehenden Distanzen d_{iv} werden als KW-Distanz realisiert.

Wenn die Lage der Zentren nicht so ist, daß die Bezirke „um sie herum“ gebildet werden können, stellen sich die Verfahren als unbrauchbar heraus.

Um Gebietsaufteilungsprobleme mit Netzwerk-Topologie zu bearbeiten, sind deshalb Methoden nötig, die entweder unabhängig von (der Lage von) Zentren sind, oder die Bestimmung guter Zentren ermöglichen. Der ersten Möglichkeit wird im folgenden Abschnitt nachgegangen, die zweite ist Gegenstand von Kapitel 9.

8.3 Gebietsaufteilung durch gleichmäßige Baumzerlegung

Modelle für die Gebietsaufteilung, in denen die Zerlegung des Graphen G in zusammenhängende Subgraphen in voller Allgemeinheit formuliert ist, sind algorithmisch problematisch (siehe 8.1.3). Deshalb wird hier der in 8.1.3.3 eingeführte Begriff des T -Zusammenhangs zugrundegelegt, um Verfahren für die Gebietsaufteilung zu gewinnen, in denen der Zusammenhang der gebildeten Bezirke gewährleistet ist.

8.3.1 Gerüst-zusammenhängende balancierte Bezirke

Es sei $T = (V, E')$ ein Gerüst von G . (Wie T gewählt werden kann, ist Gegenstand des nächsten Abschnitts.) Jede Baumzerlegung $\pi = \{T'_1, \dots, T'_p\}$ von T induziert eine Gebietsaufteilung, in welcher die den Ecken eines Teilbaumes T'_i entsprechenden KGE jeweils einen Bezirk bilden, also $B_i = V(T'_i)$. Jeder dieser Bezirke ist T -zusammenhängend. Somit lassen sich über Baumzerlegung von T Gebietsaufteilungen gewinnen, in denen alle Bezirke zusammenhängend sind. Damit wird ein aus Anwendungssicht wesentliches Kriterium erfüllt.

Um zusätzlich das Ziel einer balancierten Größe der Bezirke zu erreichen, ist eine gleichmäßige Baumzerlegung von T vorzunehmen. Die Eckengewichte $w_v, v \in V$, sind dabei die Größenattribute der KGE.

Wird zum Beispiel das Problem $T/\cdot/p/\|\cdot\|_\infty$ gelöst, so ist das Ergebnis eine Gebietsaufteilung, die die maximale Abweichung von der mittleren Größe μ minimiert; dies unter allen Aufteilungen in T -zusammenhängende Bezirke.

In Teil II wurden Verfahren entwickelt, mit denen sich die maximale Abweichung von μ auch für die Zerlegung großer Bäume mit vertretbarem Aufwand (im Sinne des Optimums) minimieren läßt. Damit können nun große Gebietsaufteilungsprobleme ($|V| \geq 500$) mit Zusammenhangsforderung effizient bearbeitet werden.

Die Zielsetzung der Minimierung der totalen Abweichung von μ oder der Varianz der Bezirkgrößen hingegen führt auf \mathcal{NP} -vollständige Baumzerlegungsprobleme für T (siehe Satz 4.17). Da wir hierfür keine Algorithmen entwickelt haben, können wir in dieser Arbeit solche Balance-Modelle in nicht mit dem Konzept des T -Zusammenhangs kombinieren.⁷

Bezirkszentren. Die Berücksichtigung von Bezirkszentren $Z \subset V, |Z| = p$, ist bei diesem Zugang ohne weiteres möglich. Ist zu jedem Zentrum ein Bezirk zu bilden,

⁷In 7.2.5 wurden derartige Balance-Modelle bei ebenen Problemen behandelt.

der dieses Zentrum enthält, so wird dies durch Lösen des Problems $T/\text{Zent}/\cdot/\|\cdot\|_\infty$ erreicht. Nach Satz 5.26 ist das mit polynomialem Aufwand möglich.

8.3.2 Auswahl des Gerüstes

Durch gleichmäßige Baumzerlegung eines Gerüstes T des Graphen G werden zwei wichtige Ziele der Gebietsaufteilung berücksichtigt, Balance und Zusammenhang. In der Regel kann auf die Kompaktheit der Bezirke aber nicht verzichtet werden (siehe Abbildung 8.1). Die Idee ist daher, die Freiheit bei der Auswahl von T zu nutzen, um Distanzinformation zu berücksichtigen. Damit soll erreicht werden, daß die gleichmäßige Zerlegung von T Bezirke generiert, die Kompaktheitsanforderungen genügen.

8.3.2.1 Verwendung des Minimalgerüstes

Eine naheliegende Idee zur Auswahl des Gerüstes T ist die Verwendung des (genauer: eines) Minimalgerüstes, des mit der Kantenbewertung d ausgestatteten Graphen G (siehe 8.1.2).⁸ Mit den bekannten Algorithmen von Kruskal oder Prim läßt sich das Minimalgerüst effizient berechnen. Unter der Voraussetzung, daß G ein dünner Graph mit $|E| = O(|V|)$ ist, beträgt der Aufwand dafür $O(|V| \log |V|)$ (Ahuja et al. [4, Seite 534]).

Beispiel 8.3

Abbildung 8.6 zeigt zwei Gebietsaufteilungen, die durch gleichmäßige Baumzerlegung des Minimalgerüstes gewonnen sind. Während sich für $p = 4$ eine Aufteilung ergibt, in der die Bezirke einigermaßen kompakt und gut balanciert sind (maximale Abweichung 1.61%), ist in der Lösung für $p = 6$ ein sehr deformierter Bezirk enthalten und die maximale Abweichung ist mit etwa 22% unbefriedigend.

Beispiel 8.3 macht deutlich, daß das Minimalgerüst im allgemeinen nicht geeignet ist, wenn durch gleichmäßige Baumzerlegung eine Gebietsaufteilung bestimmt werden soll, die neben Balance und Zusammenhang der Bezirke auch dem Kompaktheitsziel gerecht wird.

8.3.2.2 Verwendung der Optimalbasis eines Flußproblems

Eine Erklärung für Beispiel 8.3 ist die Tatsache, daß das Minimalgerüst keine Information über die Zahl der zu bildenden Bezirke enthält. Ferner kann nicht erwartet

⁸Ein *Minimalgerüst* $T = (V, E')$ hat unter allen Gerüsten von G eine minimale Summe $\sum_{e \in E'} d(e)$ der Kantenbewertungen. Üblich ist auch die Bezeichnung *minimaler spannender Baum*.

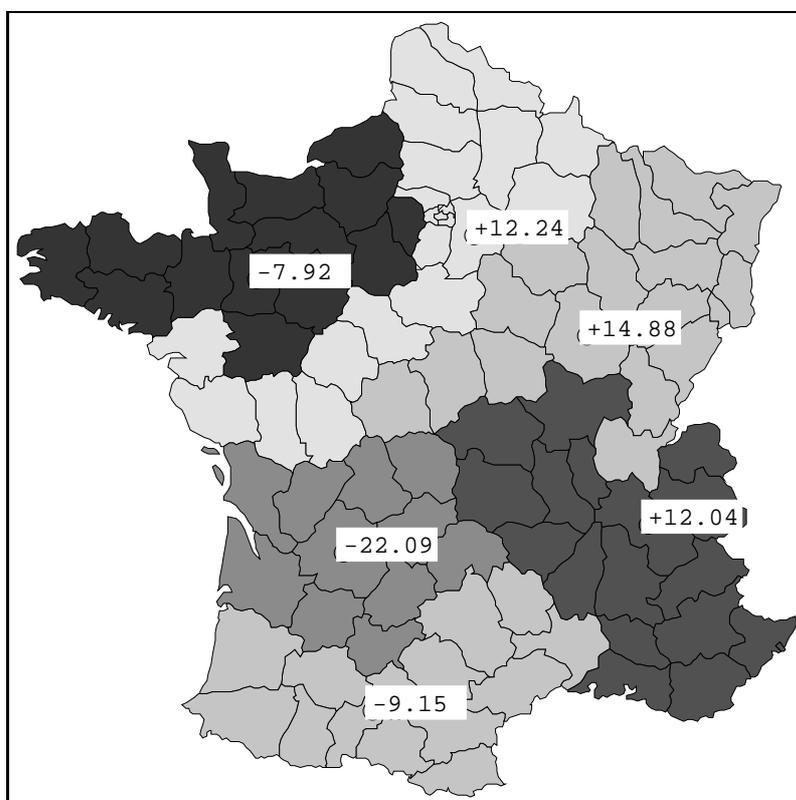
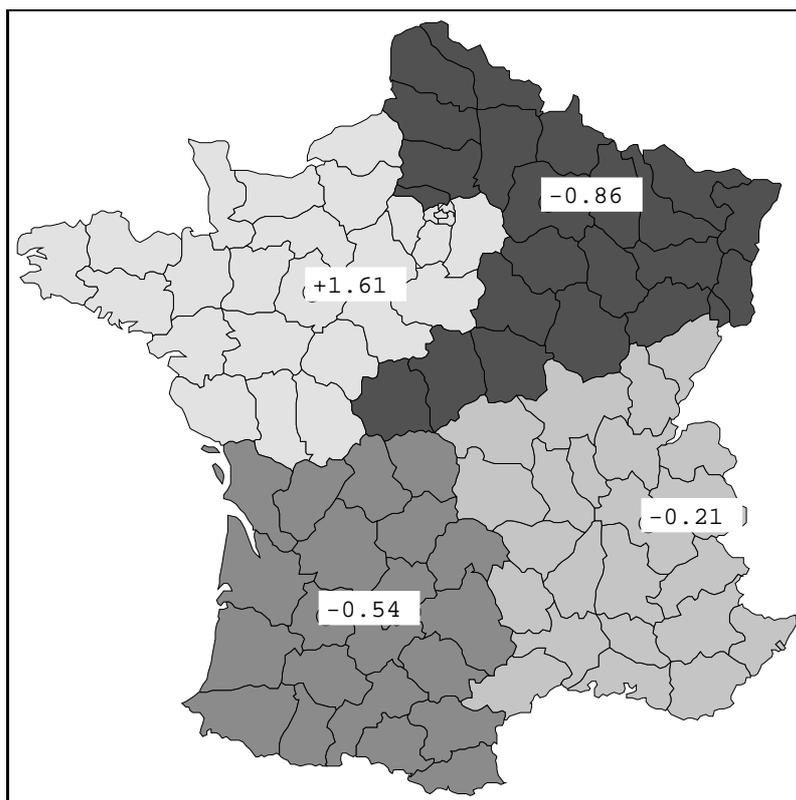


Abbildung 8.6: Gebietsaufteilungen durch Lösen von $T/\cdot/p/\|\cdot\|_\infty$ mit $p = 4$ beziehungsweise $p = 6$ für das Minimalgerüst des Nachbarschaftgraphen G .

werden kann, daß ein einzelnes Gerüst für jedes p zu einer brauchbaren Aufteilung führt.

Zur Gewinnung eines Gerüsts, das sowohl Distanzaspekte als auch die Anzahl der zu bildenden Bezirke einbezieht, schlagen wir deswegen die Verwendung der Optimalbasis eines geeigneten Minimalkosten-Flußproblems vor. Die effiziente Lösbarkeit solcher Probleme und die Tatsache, daß ihre Basen (im Sinn der linearen Programmierung) Wald-Struktur haben, machen diesen Ansatz geeignet.

Bei der Definition des Flußproblems gehen wir von einer p -elementigen Teilmenge K der Ecken von G aus. Die Elemente von K bilden die Senken des Problems. Sie können auch als „Stützpunkte“ für die Berechnung des Gerüsts angesehen werden.⁹

Das Netzwerk \tilde{G} . Sei $\tilde{G} = (V, \tilde{E})$ das (bereits in 8.1.3.2 eingeführte) Netzwerk, in dem jedes Paar in G adjazenter Ecken u und v durch zwei entgegengesetzt gerichtete Kanten verbunden ist. Die (Einheits-)Flußkosten für diese beiden Kanten sind gleich und entsprechen der durch die Kantenbewertung d von G gegebenen Distanz von u und v . Wir schreiben daher $d(u, v)$ für die Flußkosten auf der Kante (u, v) in \tilde{G} .

Das Minimalkosten-Flußproblem MF ist gegeben durch

$$(MF) \quad \min \sum_{(u,v) \in \tilde{E}} d(u, v) y_{uv} \quad (8.6a)$$

$$\text{unter} \quad \sum_{u: (v,u) \in \tilde{E}} y_{vu} - \sum_{u: (u,v) \in \tilde{E}} y_{uv} = \begin{cases} w_v & v \in V - K \\ w_v - \mu & v \in K \end{cases} \quad (8.6b)$$

$$y_{uv} \geq 0 \quad (u, v) \in \tilde{E} \quad (8.6c)$$

Jede KGE $v \in V - K$ ist eine Quelle und bringt einen Fluß der Stärke w_v hervor; in jede Senke $v \in K$ fließt ein Fluß von $w_v - \mu$. Der Fluß in die Senken kann als Netto-Zufluß angesehen werden, der sich aus einem Brutto-Zufluß von μ abzüglich w_v ergibt.

Das folgende Ergebnis ist wohlbekannt und liefert ein Gerüst T für die gleichmäßige Baumzerlegung.

Proposition 8.2

Ist \tilde{G} zusammenhängend, so ist das lineare Programm MF lösbar, und die den Basisvariablen (einer jeden Basis) entsprechenden Kanten bilden ein Gerüst von \tilde{G} .

⁹Die Senken sind ein den Bezirkszentren ähnliches Konzept. Wir treffen dennoch eine begriffliche Unterscheidung, da die Senken keine unmittelbare Bedeutung für die zu bildenden Bezirke haben; sie dienen lediglich der Berechnung des Gerüsts.

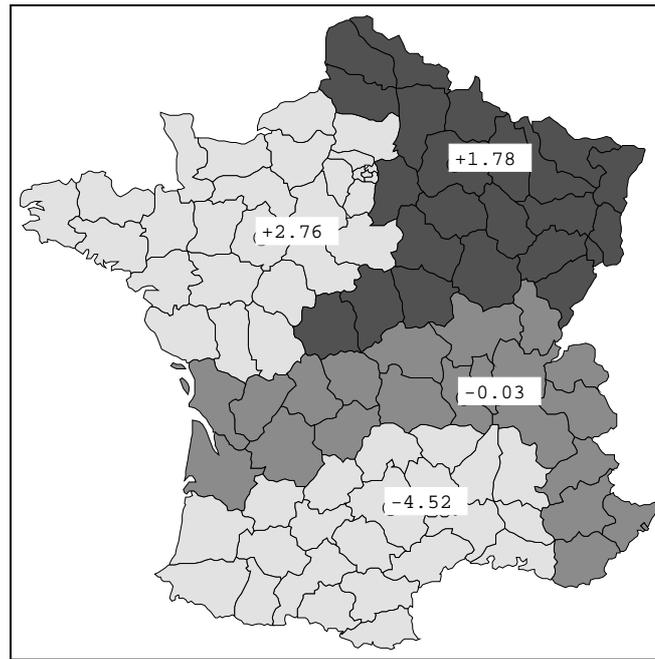


Abbildung 8.7: Gebietsaufteilung durch Lösen von $T/\cdot/p/\|\cdot\|_\infty$ wobei T aus der Optimalbasis von MF mit zufälligen Senken gewonnen wurde.

Ist die Optimalbasis von MF nicht degeneriert, so bilden die Kanten mit positivem Fluß in der Optimallösung ein Gerüst von \tilde{G} . Andernfalls muß die Basisinformation aus dem verwendeten LP-Löser gewonnen werden.

Aus dem zur Optimalbasis von MF gehörenden Gerüst von \tilde{G} wird nun direkt ein Gerüst $T = (V, E')$ von G gewonnen. Dabei sind u und v genau dann adjazent in T , wenn zwischen ihnen eine Basiskante verläuft.

Es ist klar, daß die Wahl der Senken entscheidenden Einfluß auf die Gestalt von T und damit auf die durch Baumzerlegung gewonnene Gebietsaufteilung hat. Bei einer zufälligen Auswahl der Menge K zum Beispiel ist nicht zu erwarten, daß kompakte Bezirke erzielt werden.

Beispiel 8.4

Abbildung 8.7 zeigt eine Gebietsaufteilung, die durch gleichmäßige Baumzerlegung des aus der Optimalbasis von MF gebildeten Gerüsts T errechnet wurde. Dabei wurden die vier Senken zufällig gewählt. Auch hier treten wie in Beispiel 8.3 deformierte Bezirke auf.

Andererseits ist die gute Balance (maximale Abweichung 4.52%) erwartungsgemäß, denn in die Senken von MF tritt ein Brutto-Fluß von μ ein; hieraus resultiert im allgemeinen eine gute Zerlegbarkeit von T in Teilbäume mit einem Gewicht von ungefähr μ .

Beispiel 8.4 macht die Notwendigkeit eines geeigneten Verfahrens zur Bestimmung der Senken deutlich. Ein solches Verfahren wird im folgenden Abschnitt entwickelt. Darauf aufbauend können dann durch Lösen von MF und Baumzerlegung Gebietsaufteilungen berechnet werden, in denen auch die Kompaktheit der Bezirke befriedigt.

8.3.3 Berechnung von Senken für das Flußproblem MF

Für die Berechnung der Senken für das Flußproblem MF (8.6) schlagen wir vor, von einem Gerüst \tilde{T} von \tilde{G} (zum Beispiel dem Minimalgerüst) auszugehen und K optimal (im Sinne der Zielfunktion von MF) zu wählen, unter der Voraussetzung, daß nur auf den Kanten von \tilde{T} ein Fluß stattfinden darf. (Die Richtung der Kanten wird dabei nicht berücksichtigt.) Das Flußproblem MF wird damit in gewissem Sinne umgekehrt, indem zuerst ein Gerüst (also eine Basis) gewählt wird, und dann optimale Senken dazu bestimmt werden. Wird anschließend MF mit diesen Senken gelöst, erhält man als Optimalbasis ein im allgemeinen von \tilde{T} verschiedenes Gerüst.

Somit ergibt sich in natürlicher Weise ein alternierendes Verfahren zur Bestimmung der Senken. Da der Zielfunktionswert von MF sich von Schritt zu Schritt höchstens verkleinern kann, konvergiert er gegen ein (lokales) Minimum, das bei einer (bis auf pathologische Ausnahmen) eindeutig bestimmten Menge K angenommen wird. Erfahrungsgemäß sind nur wenige Iterationen auszuführen, bis diese Menge K gefunden ist.

Der algorithmische Vorteil der Bestimmung von optimalen Senken zu einem Gerüst \tilde{T} von \tilde{G} liegt in der effizienten Lösbarkeit dieses Problems, bedingt durch die Baumstruktur. Wir nutzen also auch an dieser Stelle die Tatsache, daß Optimierungsprobleme auf Graphen oft polynomial lösbar sind, wenn der Graph baumförmig ist.

Wie werden optimale Senken zu einem Gerüst bestimmt? Wir entwickeln im folgenden zwei Verfahren. Das erste ist eine Heuristik mit linearer Laufzeit. Gelingt es, mit ihr eine zulässige Menge an Senken zu bestimmen, ist diese auch optimal. Für den Fall des Scheiterns der Heuristik lösen wir das Problem mit dynamischer Programmierung, der Aufwand dafür ist $O(p^2|V|)$. Erfahrungsgemäß ist die Heuristik jedoch in der Mehrzahl der Fälle erfolgreich.

8.3.3.1 Eine Rundungsheuristik zur Bestimmung optimaler Senken

Es sei $\tilde{T} = (V, \tilde{E}')$ ein Gerüst von \tilde{G} , mit den Eckengewichten w_v , $v \in V$. Wir wählen eine beliebige Ecke $v_0 \in V$ als Wurzel und fassen \tilde{T} als Wurzelbaum auf. Es sei an die in 4.1.1.1 eingeführte Notation erinnert.

Für $v \in V$ sei die Variable s_v die Anzahl der Senken in \tilde{T}_v . Zunächst ist $s_{v_0} = p$ vorgegeben. Da in jede Senke ein Brutto-Fluß von μ hineinfließt, ist für $v \neq v_0$ der

Algorithmus 8.1 Rundungsheuristik für optimale Senken bezüglich \tilde{T}

```

1:  $s_{v_0} := p$ 
2: for all  $v \in V - \{v_0\}$  do
3:    $s_v := \lfloor w\tilde{T}_v/\mu + 0.5 \rfloor$ 
4: end for
5: if  $s_v, v \in V$ , erfüllen (8.8) then
6:    $K := \{v \in V : s_v - \sum_{u \in \tilde{S}_v} s_u = 1\}$ 
7: else
8:   gib aus „gescheitert“
9: end if

```

Fluß auf der Kante zwischen v und $\text{pre}(v)$ betragsmäßig gleich

$$|w\tilde{T}_v - s_v\mu|.$$

Unser Ziel ist daher die Berechnung von

$$\min \sum_{v \in V - \{v_0\}} d(v, \text{pre}(v)) |w\tilde{T}_v - s_v\mu|. \quad (8.7)$$

Dabei sind die Nebenbedingungen

$$s_v - \sum_{u \in \tilde{S}_v} s_u \in \{0, 1\} \quad (v \in V) \quad (8.8a)$$

$$s_v \in \mathbb{Z}_{\geq 0} \quad (v \in V - \{v_0\}) \quad (8.8b)$$

$$s_{v_0} = p \quad (8.8c)$$

einzuhalten. (Die erste Bedingung sichert, daß jede Ecke höchstens einmal Senke sein kann.)

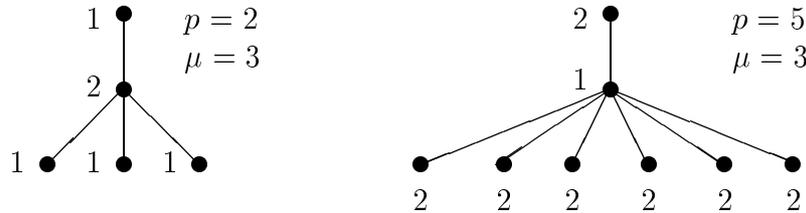
Zur Motivation der Heuristik dividieren wir die Zielfunktion (8.7) durch μ . Ohne Berücksichtigung der Nebenbedingungen ist dann klar, daß der optimale Wert für s_v sich durch Runden von $w\tilde{T}_v/\mu$ ergibt, weil dies den Term $|w\tilde{T}_v/\mu - s_v|$ minimiert.

Die Heuristik (Algorithmus 8.1) besteht deshalb einfach darin, s_v für $v \in V - \{v_0\}$ in dieser Weise zu berechnen, und dann zu prüfen, ob die Nebenbedingungen (8.8) erfüllt sind. Dies betrifft vor allem (8.8a); wenn diese Bedingung für die s_v besteht, so ist die Menge K der optimalen Senken durch diejenigen Ecken gegeben, für die die linke Seite in (8.8a) den Wert eins hat. Es ist klar, daß dies genau p Ecken sind, wenn (8.8) erfüllt ist.

Falls die durch Runden berechneten s_v die Nebenbedingungen verletzen, scheitert die Heuristik. Das folgende Beispiel zeigt, daß dies vorkommen kann.

Beispiel 8.5

Bei diesen beiden Bäumen sei die höchste Ecke Wurzel, die Gewichte w_v und die Anzahl der Senken p seien wie angegeben. Für μ ergibt sich in beiden Fällen der Wert drei.



Für den linken Baum berechnet die Heuristik $s_v = 0$ für die Blätter und $s_v = 2$ für die zentrale Ecke. Damit ist für diese Ecke (8.8a) verletzt.

Beim rechten Baum ist für die Blätter $s_v = 1$ und für die zentrale Ecke $s_v = 4$, daher erhält man in (8.8a) sogar einen negativen Wert.

Das folgende Ergebnis ist offensichtlich.

Proposition 8.3

Der Aufwand von Algorithmus 8.1 ist $O(|V|)$.

8.3.3.2 Berechnung optimaler Senken mit dynamischer Programmierung

Für den Fall des Scheiterns der Rundungsheuristik aus Algorithmus 8.1 wird das durch (8.7) und (8.8) gegebene Optimierungsproblem mit dynamischer Programmierung gelöst. Wie in 8.3.3.1 wählen wir eine beliebige Wurzel $v_0 \in V$ für \tilde{T} .

Sei dazu $C(v, s)$ der optimale Zielfunktionswert für den Teil-, „Baum“, der aus \tilde{T}_v und der Kante $(v, \text{pre}(v))$ besteht,¹⁰ wenn dieser s Senken enthält. Gesucht ist $C(v_0, p)$.

Um Rekursionsgleichungen für $C(v, s)$ zu entwickeln, ist die Definition eines weiteren Zwischenziels nötig. Ist $\tilde{S} = \{u_1, \dots, u_{k_v}\}$ die Menge der Söhne von v , so sei $\tilde{C}(v, i, s)$ der optimale Zielfunktionswert für den Teilbaum, der aus v , den Kanten $(u_1, v), \dots, (u_i, v)$ und den $\tilde{T}_{u_1}, \dots, \tilde{T}_{u_i}$ gebildet wird, wenn dieser Teilbaum s Senken enthält. (Für $i = 0$ besteht der Teilbaum nur aus der Ecke v .)

Die Rekursionsgleichungen für die Vorwärtsrechnung von den Blättern zur Wurzel

¹⁰Ist v die Wurzel, so kann gedanklich eine solche Kante hinzugefügt werden.

lauten wie folgt:

$$\tilde{C}(v, 0, s) = \begin{cases} 0 & s \in \{0, 1\}, \\ \infty & \text{sonst,} \end{cases} \quad (8.9)$$

$$\tilde{C}(v, i, s) = \min \{C(u_i, s') + \tilde{C}(v, i-1, s-s') : 0 \leq s' \leq s\} \\ \text{für } 1 \leq i \leq k_v, v \text{ kein Blatt,} \quad (8.10)$$

und

$$C(v, s) = \tilde{C}(v, k_v, s) + \begin{cases} d(v, \text{pre}(v))|w\tilde{T}_v - s\mu| & 0 \leq s \leq |V(\tilde{T}_v)|, \\ \infty & \text{sonst.} \end{cases} \quad (8.11)$$

Proposition 8.4

Der Aufwand für die Vorwärtsrechnung zur Bestimmung optimaler Senken für \tilde{T} ist $O(p^2|V|)$.

Beweis. Der Aufwand wird dominiert durch die Berechnung von $O(2|V|p)$ Zwischenzielen der Form $\tilde{C}(v, i, s)$ in (8.10). \square

Die optimalen s_v werden dann durch Rückwärtsrechnung ermittelt. Darin werden die Söhne von v in umgekehrter Reihenfolge u_{k_v}, \dots, u_1 betrachtet und jeweils s_{u_i} aus dem (während der Vorwärtsrechnung festgehaltenen) in (8.10) minimierenden s' bestimmt. Der Aufwand für die Rückwärtsrechnung ist $O(|V|)$. Damit gilt das folgende Ergebnis.

Satz 8.5

Mit dynamischer Programmierung gemäß (8.9) bis (8.11) können optimale Senken für \tilde{T} mit Aufwand $O(p^2|V|)$ bestimmt werden.

8.3.4 Zusammenfassende Darstellung und Beispiele

8.3.4.1 Zusammenfassung

Zusammengefaßt ist unsere Vorgehensweise zur Bestimmung einer Gebietsaufteilung in T -zusammenhängende, balancierte und kompakte Bezirke in Algorithmus 8.2.

8.3.4.2 Praktische Umsetzung

Die praktische Umsetzung von Algorithmus 8.2 in GAMOR wurde folgendermaßen vorgenommen. In Zeile 2 wird eine Implementierung des Verfahrens von Prim benutzt. Problem MF (Zeile 10) wird mit dem Solver CPLEX gelöst. In Zeile 13 wird das Branch&Bound Verfahren aus Kapitel 6 benutzt.

Algorithmus 8.2 Berechnung einer Gebietsaufteilung bei Netzwerktopologie

Input: gewichteter Graph $G = (V, E, w, d)$ und $p \in \{1, \dots, |V|\}$ **Output:** Bezirke $B_1, \dots, B_p \subset V$, zusammenhängend in G , die eine Partition von V bilden

- 1: Bilde Netzwerk \tilde{G} ausgehend von G (siehe Seite 197)
 - 2: Berechne Minimalgerüst \tilde{T} von \tilde{G}
 - 3: weiter := True; $K := \emptyset$
 - 4: **while** weiter **do**
 - 5: Wende Rundungsheuristik (Algorithmus 8.1) zur Berechnung einer Senkenmenge K auf \tilde{T} an
 - 6: **if** Heuristik gescheitert **then**
 - 7: berechne K mit dynamischer Programmierung (siehe 8.3.3.2)
 - 8: **end if**
 - 9: **if** K unverändert **then** weiter := False **end if**
 - 10: Löse Minimalkosten-Flußproblem MF (8.6) auf \tilde{G} ; \tilde{T} sei Optimalbasis
 - 11: **end while**
 - 12: $T = (V, E', w)$ sei Gerüst von G , das \tilde{T} entspricht
 - 13: Löse Baumzerlegungsproblem $T / \cdot / p / \|\cdot\|_\infty$ für T
 - 14: $B_i :=$ Eckenmenge des i -ten Teilbaumes in der Zerlegung, $i \in I$
-

8.3.4.3 Beispiele**Beispiel 8.6**

Abbildung 8.8 zeigt zwei Gebietsaufteilungen, die mit Algorithmus 8.2 berechnet wurden. Ein Vergleich mit Abbildung 8.6 macht deutlich, daß hier das zur Baumzerlegung verwendete Gerüst sehr viel besser das Ziel balancierter und kompakter Bezirke unterstützt. Algorithmus 8.2 erscheint geeignet für das Design praxistauglicher Gebietsaufteilungen.

Hierbei ist vor allem der Gesichtspunkt der Effizienz hervorzuheben. Die Gesamt-rechenzeit betrug für jede der Aufteilungen in Abbildung 8.8 etwa 0.15 Sekunden. Die While-Schleife in den Zeilen 4 bis 11 wurde jeweils dreimal durchlaufen. Nur für $p = 6$ war dabei (in der ersten Iteration) der Rückgriff auf dynamische Programmierung zur Bestimmung der Senken erforderlich. Das Problem $T / \cdot / p / \|\cdot\|_\infty$ konnte in beiden Fällen im Wurzelknoten des Suchbaumes gelöst werden, die Rechenzeit dafür betrug ungefähr 0.8 Sekunden.

Beispiel 8.7

Um die Effizienz von Algorithmus 8.2 für große Probleme zu testen, wurde eine Gebietsaufteilung mit 25 Bezirken auf der Basis von 544 KGE (die deutschen Kreise

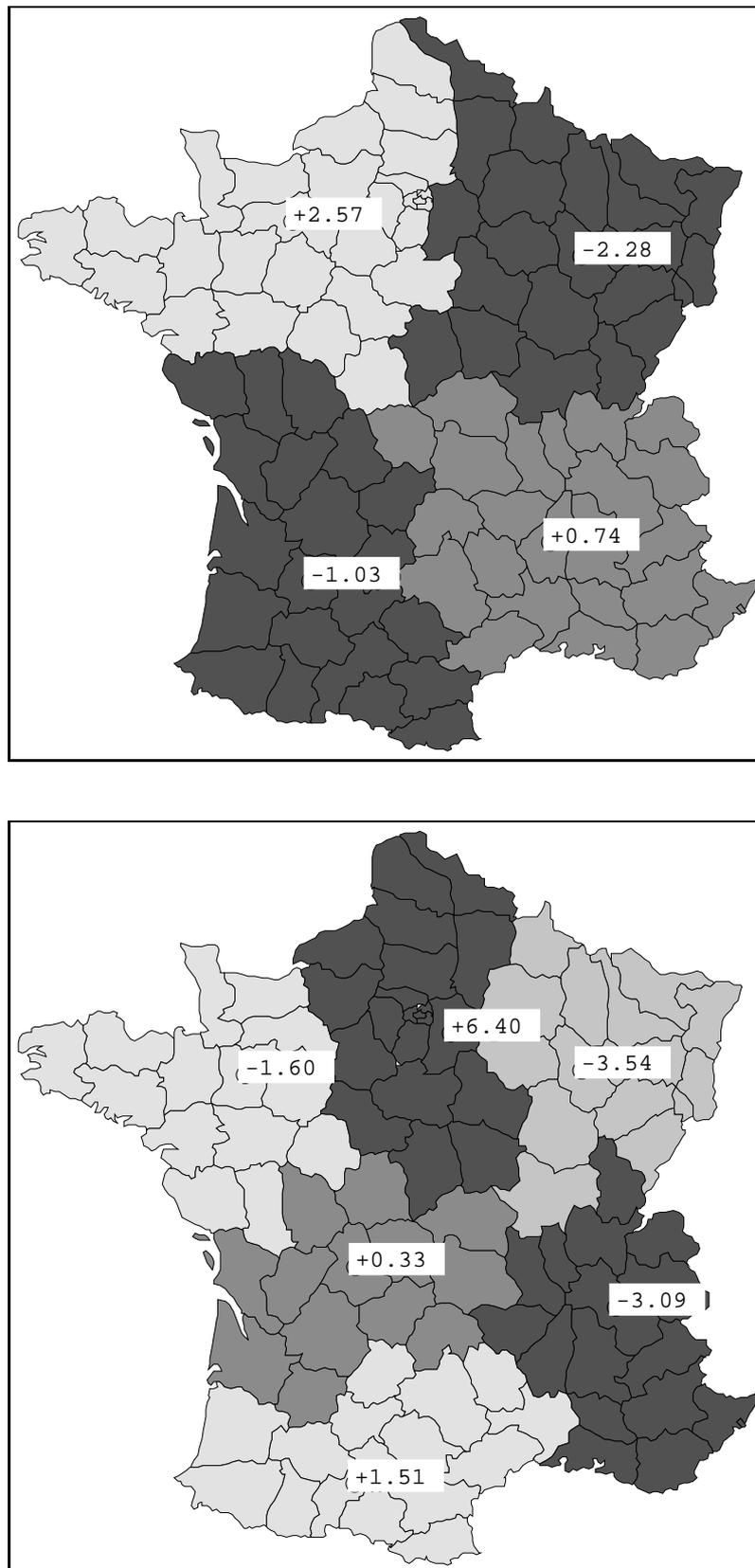


Abbildung 8.8: Gebietsaufteilungen mit $p = 4$ beziehungsweise $p = 6$, berechnet mit Algorithmus 8.2

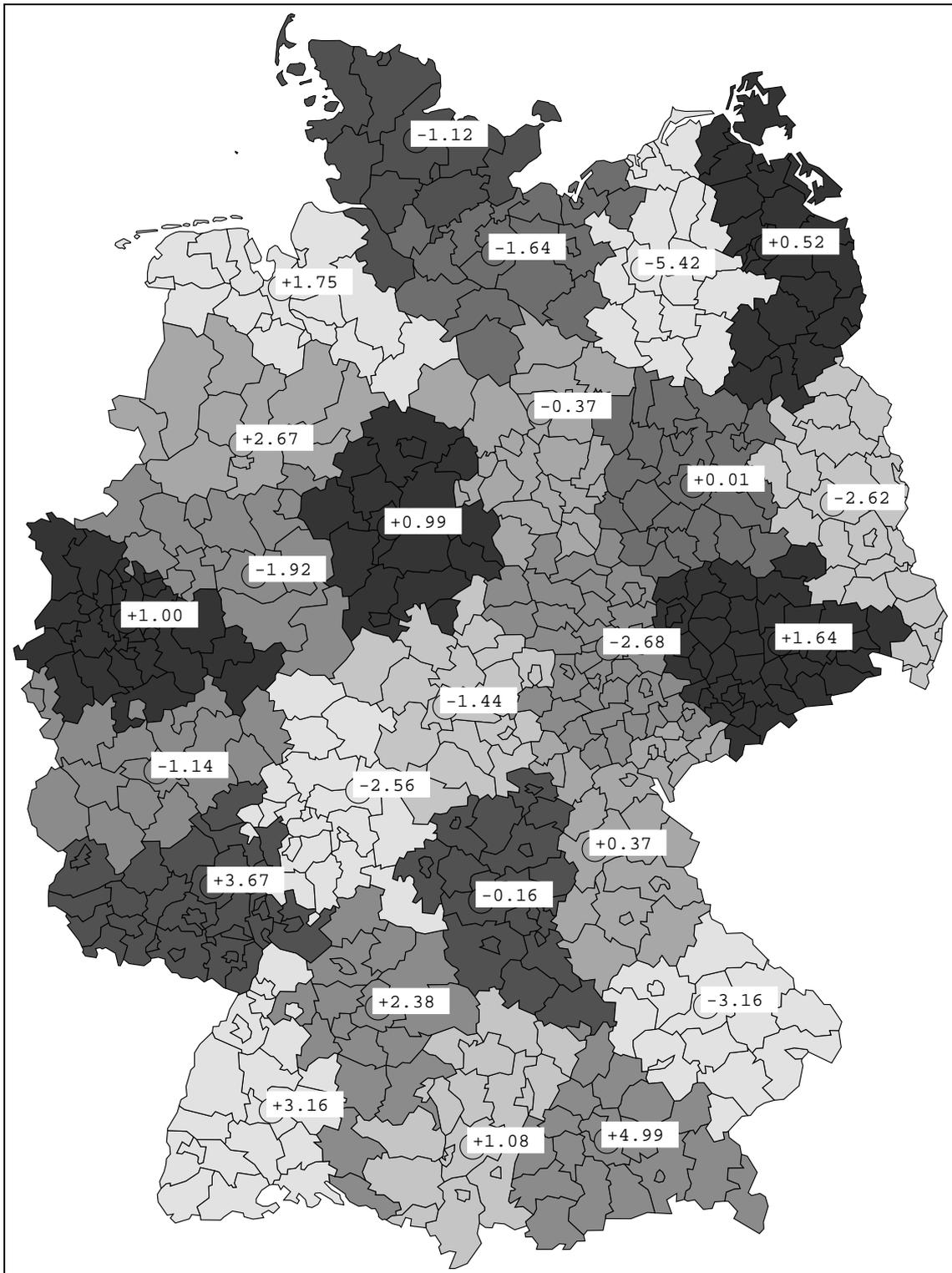


Abbildung 8.9: Gebietsaufteilung mit $|V| = 544$ und $p = 25$ berechnet mit Algorithmus 8.2

und kreisfreien Städte) berechnet (Abb. 8.9).

Die Gesamtrechenzeit dafür betrug etwa fünf Sekunden, wovon 4.3 Sekunden auf das Branch&Bound in Zeile 13 entfielen. Es mußten 23 Knoten des Suchbaumes abgearbeitet werden, bis die optimale Baumzerlegung bestimmt war. Die While-Schleife in den Zeilen 4 bis 11 mußte fünf mal durchlaufen werden, wobei im ersten Durchlauf dynamische Programmierung zur Bestimmung der Senken nötig war.

Aus Beispiel 8.7 kann der Schluß gezogen werden, daß mit Algorithmus 8.2 auch Gebietsaufteilungen mit vielen KGE bei Netzwerk-Topologie und Zusammenhangsbedingung sehr effizient berechnet werden können.

8.3.5 Vorteile gegenüber herkömmlichen Verfahren

Der in GAMOR realisierte Algorithmus 8.2 aus 8.3.4 verbindet verschiedene Eigenschaften in einer Weise, die ihn gegenüber herkömmlichen Verfahren vorteilhaft erscheinen läßt.

1. Es werden keine im voraus gegebenen Bezirkszentren benötigt.
2. Es werden stets zusammenhängende Bezirke erzeugt.
3. Das Verfahren ist sehr effizient (siehe Beispiel 8.7).

Auch das Verfahren von Zoltners und Sinha [115] bildet in aller Regel zusammenhängende Bezirke, wenn es mit KW-Distanzen benutzt wird (Satz 8.1).¹¹ Jedoch ist der Algorithmus bei diesem Distanz-Typ auf gut gewählte Bezirkszentren angewiesen (Beispiel 8.2). Außerdem benötigt das Subgradienten-Verfahren eine deutlich längere Rechenzeit, als Algorithmus 8.2.¹² Selbst wenn gute Bezirkszentren bekannt sind, ist daher die hier entwickelte Vorgehensweise im Vorteil gegenüber dem Verfahren von Zoltners und Sinha [115].

Die Berechnung einer Gebietsaufteilung durch Lösen des Transportproblems und anschließende Splitauflösung (siehe 7.2) erzeugt sehr häufig unzusammenhängende Bezirke, auch wenn durch Verwendung von KW-Distanzen der Nachbarschaftsgraph der KGE indirekt einbezogen, und von gut verteilten Bezirkszentren ausgegangen wird.¹³ Daher ist dieser Ansatz im Falle einer strikten Zusammenhangsforderung nicht geeignet.

¹¹Unter einigen hundert zufällig erzeugten Testproblemen wurde nur in einem Fall eine Gebietsaufteilung mit unzusammenhängenden Bezirken berechnet.

¹²Für die erwähnte Menge an Testproblemen war die Rechenzeit des Subgradientenverfahrens (bei guten gegebenen Bezirkszentren!) in GAMOR im Durchschnitt drei bis vier mal länger, als die von Algorithmus 8.2.

¹³Für die Testprobleme traten in der Mehrzahl der Gebietsaufteilungen unzusammenhängende Bezirke auf.

Kapitel 9

Verfahren zur Zentrenbestimmung (simultane Standortplanung)

Ein Teil der Modelle für die Gebietsaufteilung in den Kapiteln 7 und 8 geht von a priori gegebenen Zentren der Gebiete aus. Aus den Beispielen in 8.2 wird erkennbar, daß die Wahl der Zentren entscheidenden Einfluß auf die berechnete Gebietsaufteilung hat. Deshalb sind Verfahren notwendig, mit denen sich geeignete Zentren bestimmen lassen.

Auch aus Anwendungssicht ist dies bedeutsam. Da die Kompaktheit der Gebietsaufteilung unter Bezugnahme auf Zentren gemessen wird (siehe zum Beispiel (7.1a), S. 141), empfehlen sich die Zentrenorte als Standorte von Vertretern oder Organisationseinheiten, denen die Betreuung der Gebiete unterliegt; insbesondere beim Bereisen der Gebiete ist dies der Fall. Die Bestimmung günstiger Zentren entspricht somit der Planung geeigneter Vertreterstandorte.

Methodisch gehören die im folgenden vorgestellten Verfahren zur Disziplin der Standortplanung im Operations Research. Dieses sehr umfangreiche Forschungsfeld können wir jedoch nur streifen. Verschiedentlich verweisen wir deshalb auf weiterführende Literatur.

Das Kapitel ist wie folgt gegliedert. In Abschnitt 9.1 wird dargestellt, wie zu gegebenen Bezirken optimale Zentren bestimmt werden können. Dies wird in 9.2 benutzt, um alternierend Zentren und Gebietsaufteilungen zu berechnen, und so schließlich zu einem (lokalen) Optimum zu kommen. Jedoch ist dieses location allocation Prinzip sensibel gegenüber der Wahl der Startzentren. Deshalb wurde von Hojati [52] eine Lagrange-Heuristik zur Bestimmung (global) optimaler Zentren vorgeschlagen; wir gehen darauf in 9.3 ein. Da die Heuristik nach Erfahrung des Autors aber nicht zufriedenstellend arbeitet, schlagen wir Erweiterungen vor. Rechentests zeigen, daß damit ein wirkungsvolles Verfahren zur Bestimmung guter Zentren gegeben ist.

9.1 Optimale Zentren bei gegebener Gebietsaufteilung

Beim location–allocation Prinzip werden alternierend Gebietsaufteilungen und Zentren bestimmt. Daher stellen wir zunächst die Frage, wie sich für eine gegebene Gebietsaufteilung optimale Bezirkszentren berechnen lassen. Sie führt direkt auf das Median–Problem der Standortplanung, das (angepaßt an unseren Kontext) folgendermaßen definiert ist.

Definition 9.1 (Median einer Menge von KGE)

Seien

- $B \subseteq V$ eine Teilmenge der KGE,
- $W_v \geq 0$, $v \in B$, Gewichte der KGE,
- \mathcal{Z} eine Menge zulässiger Zentren,
- $d : \mathcal{Z} \times V \longrightarrow \mathbb{R}_{\geq 0}$.

Ein $z \in \mathcal{Z}$, das

$$\sum_{v \in B} W_v d(z, v) \tag{9.1}$$

minimiert, heißt *Median* von B .

Die Funktion d dient zur Messung der Distanz zwischen KGE und möglichen Zentren. Die Gewichte der KGE werden zum Beispiel durch $W_v := w_v$ festgelegt. Für die Wahl von \mathcal{Z} und d sind im Kontext unserer Arbeit folgende Möglichkeiten von Interesse.

1. In Modellen mit ebener Topologie ist $\mathcal{Z} = \mathbb{R}^2$ und d ist die euklidische oder die quadriert–euklidische Distanz (vgl. S. 139).
2. In Modellen mit Netzwerk–Topologie ist $\mathcal{Z} = B$ und d ist zum Beispiel die KW–Distanz (vgl. 8.1.2).

Der Median z eines Bezirks B ist der günstigste Standort für die Betreuung dieses Bezirks, wenn der Aufwand durch die Funktion (9.1) gemessen wird. Indem für jeden Bezirk einer Gebietsaufteilung ein Median als Zentrum festgelegt wird, bestimmt man eine Menge \mathcal{Z} optimaler Zentren.

Die Berechnung des Medians wird durch Verfahren aus der Standortplanung ermöglicht. Für die unter 1. und 2. aufgeführten Beispiele für \mathcal{Z} und d führen wir im folgenden geeignete Verfahren an.

9.1.1 Berechnung des Medians bei ebenen Problemen

In ebenen Problemen der Standortplanung ist jeder Punkt der Ebene \mathbb{R}^2 als Standort zulässig.¹ Eine Zusammenstellung von Verfahren zur ebenen Standortplanung gibt Hamacher [46]. Hier benötigen wir nur die einfachsten.

9.1.1.1 Quadriert-euklidische Distanzen

Wie in 7.1.1 seien $(ost_v, nord_v) \in \mathbb{R}^2$, $v \in V$ die Orte der KGE. Ferner sei $(Ost_z, Nord_z)$ das zu bestimmende Zentrum. Für

$$d(z, v) = (Ost_z - ost_v)^2 + (Nord_z - nord_v)^2$$

wird der Median mit einer einfachen geschlossenen Formel berechnet:

$$Ost_z = \frac{\sum_{v \in B} W_v ost_v}{\sum_{v \in B} W_v} \quad Nord_z = \frac{\sum_{v \in B} W_v nord_v}{\sum_{v \in B} W_v} \quad (9.2)$$

Zum Beweis wird die Funktion (9.1) nach $(Ost_z, Nord_z)$ differenziert.

Der Aufwand zur Berechnung der optimalen Zentren bei gegebener Gebietsaufteilung mit Bezirken B_i , $i = 1, \dots, p$, ist für quadriert-euklidische Distanzen somit $O(\sum_i |B_i|) = O(|V|)$.

9.1.1.2 Euklidische Distanzen

Ist die Distanzfunktion durch

$$d(z, v) = ((Ost_z - ost_v)^2 + (Nord_z - nord_v)^2)^{1/2}$$

gegeben,² so läßt sich der Median nicht durch eine geschlossene Formel berechnen. Die Iterationsvorschrift des Weiszfeld-Algorithmus

$$(Ost_z^{k+1}, Nord_z^{k+1}) = \varphi(Ost_z^k, Nord_z^k),$$

wobei $\varphi : \mathbb{R}^2 \rightarrow \mathbb{R}^2$ durch

$$\varphi(\zeta) = \sum_{v \in B} \frac{\gamma_v(\zeta)}{\Gamma(\zeta)} (ost_v, nord_v)$$

mit

$$\gamma_v(\zeta) = \gamma_v(\zeta_1, \zeta_2) = \frac{W_v}{((\zeta_1 - ost_v)^2 + (\zeta_2 - nord_v)^2)^{1/2}} \quad \text{und} \quad \Gamma(\zeta) = \sum_{u \in B} \gamma_u(\zeta)$$

¹Häufig wird auch nur eine Teilmenge \mathcal{Z} der Ebene zugrundegelegt, zum Beispiel wenn es verbotene Bereiche für den Median gibt.

²In diesem Fall wird der Median auch als Steiner-Weber Standort bezeichnet.

gegeben ist, liefert jedoch im allgemeinen eine Folge von Punkten, die gegen den Median konvergiert. Dazu darf dieser allerdings nicht mit dem Ort ($ost_v, nord_v$) einer KGE übereinstimmen. Ein Kriterium zur Überprüfung dieses Falls und weitere Informationen zum Weiszfeld-Algorithmus findet man in Hamacher [46, Abschnitt 2.1.5].

9.1.2 Berechnung des Medians bei Netzwerk-Topologie

In Modellen mit Netzwerk-Topologie ist es sinnvoll, als Zentren nur die Knoten des Netzwerkes zuzulassen (vgl. S. 182). Darüberhinaus sollte das Zentrum eines Bezirkes in diesem enthalten sein. Die Berechnung des Medians ist somit trivial, da die Funktion (9.1) über die endliche Menge $\mathcal{Z} = B$ minimiert wird.

Der Aufwand zur Bestimmung des Medians ist in diesem Fall $O(|B|^2)$, unter der Voraussetzung, daß die Distanz $d(z, v)$ in konstanter Zeit berechnet werden kann. Werden KW-Distanzen $d(z, v) = D_{zv}$ zugrundegelegt und ist die Distanzmatrix D (siehe S. 183) bekannt, so ist diese Voraussetzung erfüllt.

Zur Bestimmung der Mediane aller Bezirke, also der Menge optimaler Zentren, ergibt sich damit ein Aufwand von $O(\sum_{i=1}^p |B_i|^2) = O(|V|^2)$.

9.2 Bestimmung von Zentren mit dem location-allocation Prinzip

9.2.1 Das GEOLINE-Modell

Das location-allocation Prinzip ist die alternierende Bestimmung von Zentren und die Zuordnung von KGE zu diesen Zentren. Begonnen wird mit der Festlegung einer ersten Zentrenmenge Z^0 , den *Startzentren*. Diesen werden die KGE zugeordnet, zum Beispiel durch Lösen des Transportproblems TP (7.2). Dadurch wird eine Gebietsaufteilung definiert (die Splits enthalten kann). Für die Bezirke dieser Gebietsaufteilung werden Mediane, d.h. optimale Zentren, mit den Methoden des vorangehenden Abschnitts bestimmt. Die so erhaltene Zentrenmenge Z^1 erlaubt das erneute Berechnen einer Gebietsaufteilung. In dieser Weise wird zwischen den Schritten location und allocation abgewechselt, bis ein Abbruchkriterium erfüllt ist.

Dieses Prinzip wurde von Hess et al. [50] für die Gebietsaufteilung zunächst im Zusammenhang der Einteilung von Wahlkreisen eingeführt, und etablierte sich dann als GEOLINE-Modell für die Verkaufsgebietsgestaltung (Hess und Samuels [51]). Später verwendeten es in modifizierter Form zum Beispiel auch Fleischmann und Paraschis [35].

Algorithmus 9.1 GEOLINE Verfahren (location–allocation)

- 1: Wähle Startzentren Z^0 .
 - 2: $k := 0$
 - 3: **repeat**
 - 4: $k := k + 1$
 - 5: Berechne Distanzen d_{iv} , $i \in Z^{k-1}$, $v \in V$.
 - 6: (x_{iv}) sei optimale Lösung von TP (7.2).
 - 7: Runde gegebenenfalls (x_{iv}) auf ganzzahlige Werte.
 - 8: Z^k sei Menge der Mediane (mit $W_v = w_v x_{iv}$ in (9.2))
für die Bezirke.
 - 9: **until** $Z^k = Z^{k-1}$ oder $k = \text{max_iteration}$
-

Das GEOLINE–Verfahren in der Form von Hess und Samuels [51] arbeitet mit quadriert–euklidischen Distanzen (Zeile 5) und Splitauflösung mittels AssignMAX (Zeile 7; siehe 7.2.3). Fleischmann und Paraschis [35] führen das Runden nur in der letzten Iteration durch und verwenden dabei die in 7.2.6 vorgestellte partielle Splitauflösung.

Algorithmus 9.1 abstrahiert von solchen Festlegungen. Alternative Distanz–Modelle (euklidisch, kürzeste Wege) oder Rundungsverfahren (optimale Splitauflösung unter verschiedenen Kriterien) lassen sich integrieren, indem die Zeilen 5 und 8 beziehungsweise 7 entsprechend ausgeführt werden. In GAMOR ist das GEOLINE–Verfahren in dieser modularen Form integriert, das gewünschte Distanzmodell und das Rundungsverfahren lassen sich über Kommandoparameter auswählen.

9.2.2 Konvergenz des location–allocation Prinzips

In praktischen Fällen konvergiert Algorithmus 9.1 recht schnell, es gilt $Z^k = Z^{k-1}$ zumeist für ein $k \leq 10$. Hess und Samuels [51] merken jedoch an, daß Konvergenz nicht gesichert ist, wenn (x_{iv}) in jeder Iteration gerundet wird — es kann vorkommen, daß der Algorithmus Zentrenmengen zyklisch durchläuft. Deshalb ist es sinnvoll, eine Maximalzahl `max_iteration` an Iterationen vorzugeben (Zeile 9).

Wird auf das Runden in Zeile 7 verzichtet, so läßt sich Konvergenz einfach nachweisen. Wesentliche Faktoren sind dabei, daß die Anzahl der Basislösungen von TP endlich ist und daß die Zielfunktion $\sum_{i,v} d_{iv} w_v x_{iv}$ von Iteration zu Iteration abnimmt. Um ein zyklisches Verhalten auszuschließen, sind neben dem Verzicht auf das Runden zwei Voraussetzungen nötig.

1. Die Basen von TP werden (zum Beispiel lexikographisch) geordnet, im Falle einer Mehrdeutigkeit der optimalen Basislösung wird in Zeile 6 von Algorithmus

9.1 die bezüglich dieser Ordnung kleinste gewählt.

2. Für den Fall einer nichteindeutigen optimalen Zentrenmenge in Zeile 8 wird ebenfalls nach einer festen Regel Z^k ausgewählt.

Unter Beachtung dieser Voraussetzungen bestimmen sich Optimallösungen von TP und Zentrenmengen Z^k gegenseitig in eindeutiger Weise. Zusammen mit dem Abnehmen der Zielfunktion von Iteration zu Iteration schließt dies ein Kreisen aus. Konvergenz, also Terminierung mit $Z^k = Z^{k-1}$ in Algorithmus 9.1, ergibt sich dann aus dem Endlichkeits-Argument.

9.2.3 Abhängigkeit von den Startzentren

Die durch das location-allocation Prinzip schließlich bestimmte Zentrenmenge ist von den Startzentren Z^0 abhängig und wird in der Regel kein globales Optimum darstellen. Hess und Samuels [51] empfehlen daher ein mehrmaliges Durchlaufen mit jeweils unterschiedlichen Startzentren. Jedoch stellt die Wahl der Startzentren ein gewisses Problem dar.

Die Bestimmung möglichst guter Zentren und die simultane Berechnung einer Gütegarantie dafür ist Gegenstand des nächsten Abschnitts. Die Menge \mathcal{Z} der zulässigen Zentren ist darin durch die Menge der KGE gegeben. Durch diese Einschränkung gegenüber ebenen Modellen läßt sich die Fragestellung in Gestalt eines warehouse location Problems formulieren.

9.3 Eine Lagrange-Heuristik zur Bestimmung optimaler Zentren

Hojati [52] bestimmt die Zentren der Bezirke mit einer Lagrange-Heuristik, die auf einer Formulierung des Problems als kapazitiertes warehouse location Problem basiert. Die Heuristik ist eine vereinfachte Version der von Beasley [11] vorgestellten Vorgehensweise zur Lösung verschiedener grundlegender Standortprobleme.

In der von Hojati dargestellten Form ist das Verfahren nach Erfahrungen des Autors jedoch nicht besonders erfolgreich in der Berechnung von guten Bezirkszentren. Daher schlagen wir im folgenden Modifikationen vor, die zu besseren Ergebnissen führen. Sie seien kurz skizziert, bevor wir eine detaillierte Darstellung bringen.

1. Hojati [52] dualisiert in Anlehnung an Beasley [11] Nachfrage- und Kapazitäts-Nebenbedingungen. Wir dualisieren dagegen nur die Nachfrage-Bedingungen. Das Lagrange-Subproblem ist trotzdem trivial lösbar, und es ist nur die halbe Anzahl Lagrange-Multiplikatoren im Subgradienten-Verfahren zu optimieren.

2. Hojati verwendet eine sehr einfache primale Heuristik, in der nur die besten gefundenen Lagrange-Multiplikatoren berücksichtigt werden. Dies ist im allgemeinen ungenügend. Wir schlagen vor, Algorithmus 9.1 als primale Heuristik zu verwenden. Dies ist recht aufwendig, da viele Transportprobleme gelöst werden müssen. Es können jedoch sehr gute Lösungen damit bestimmt werden; dies rechtfertigt unserer Ansicht nach den Aufwand.

9.3.1 Optimale Zentren: Ein kapazitiertes warehouse location Problem

9.3.1.1 Formulierung des Modells

Hojati [52] formuliert das Problem optimaler Zentren in folgender Weise als kapazitiertes warehouse location Modell.

$$(WLP) \quad \min \sum_{i,v} d_{iv} w_v x_{iv} \quad (9.3a)$$

$$\text{unter} \quad \sum_{i \in V} x_{iv} = 1 \quad (v \in V) \quad (9.3b)$$

$$\sum_{v \in V} w_v x_{iv} = \mu z_i \quad (i \in V) \quad (9.3c)$$

$$\sum_{i \in V} z_i = p \quad (9.3d)$$

$$0 \leq x_{iv} \leq 1 \quad (i, v \in V) \quad (9.3e)$$

$$z_i \in \{0, 1\} \quad (i \in V) \quad (9.3f)$$

Es kann als Erweiterung von TP (7.2) aufgefaßt werden, wobei hier die binären Variablen z_i die Entscheidung modellieren, ob KGE $i \in V$ ein Zentrum ist ($z_i = 1$) oder nicht ($z_i = 0$). WLP ist somit ein gemischtganzzahliges Programm. Man beachte, daß hier der Index i nicht aus der Menge $I = \{1, \dots, p\}$ stammt, sondern über alle KGE variiert. Die Menge \mathcal{Z} möglicher Zentren stimmt also mit der Menge V der KGE überein.

Gegenüber der üblichen Definition des warehouse location Problems³ ist anzumerken, daß hier die Standortvariablen z_i nicht in der Zielfunktion auftauchen, das Modell also keine Fixkosten einbezieht. Dient die Suche nach guten Bezirkszentren tatsächlich einer Standortentscheidung für die Betreuung der Bezirke, können mit der Standorteinrichtung und -unterhaltung verbundene Fixkosten jedoch problemlos in das Modell integriert werden.

³Siehe zum Beispiel Domschke und Drexl [30, S. 53].

Außerdem kommt gegenüber der üblichen Definition die Nebenbedingung (9.3d) hinzu; sie legt fest, daß gerade so viele Zentren zu bestimmen sind, wie die Gebietsaufteilung Bezirke enthalten soll. (9.3d) ist in der Formulierung von WLP entbehrlich; bei der vorzunehmenden Lagrange–Relaxation ist diese Bedingung jedoch von Nutzen.

9.3.1.2 Lösungsverfahren für WLP

Das Programm WLP unter Rückgriff auf einen Solver für gemischtganzzahlige Programme wie CPLEX zu lösen, ist nur für kleine Probleminstanzen möglich. Obwohl dieser Löser durch Schnittebenen die lineare Relaxation von WLP erheblich verschärft, treten schon bei Instanzen mit $|V| = 100$ KGE und $p = 20$ Schwierigkeiten auf, WLP optimal zu lösen. Dies wurde bei entsprechenden Experimenten mit GAMOR beobachtet. Probleme der Größe $|V| = 50$ bereiten CPLEX (in der Version 6.6 des Lölers) dagegen keine Schwierigkeiten. Da wir an Verfahren für größere Probleme interessiert sind, bieten sich Heuristiken an.

Lagrange–Heuristiken haben den Vorteil, daß man neben einer Lösung des Problems eine Gütegarantie für diese Lösung erhält. Im Zusammenhang mit dem kapazitierten warehouse location Problem werden solche Heuristiken als besonders erfolgreich empfohlen.⁴ Deshalb gehen wir im folgenden auch diesen Weg, der für die Bestimmung von Zentren in der Gebietsaufteilung bereits von Hojati [52] beschritten wurde. Die Unterschiede unseres Ansatzes sind oben (S. 212) erwähnt.

9.3.2 Eine Lagrange–Heuristik zur Lösung von WLP

9.3.2.1 Lagrange–Relaxation von WLP

Hojati [52] dualisiert die Nachfrage–Nebenbedingungen (9.3b) und die Kapazitäts–Nebenbedingungen (9.3c), um zu einer Lagrange–Relaxation von WLP zu kommen. Hier dagegen werden nur die Nachfrage–Bedingungen dualisiert. Nach Cornuejols et al. [26, Th. 1] stimmen die erzielbaren unteren Schranken (das Optimum des jeweiligen Lagrange–Duals) für beide Vorgehensweisen überein.

Im zweiten Fall müssen jedoch nur die halbe Anzahl Lagrange–Multiplikatoren optimiert werden, was ein besseres Konvergenzverhalten des Subgradienten–Verfahrens erwarten läßt. Die Lösung des Lagrange–Subproblems ist in beiden Fällen trivial durch Sortierung auszuführen.

⁴Cornuejols et al. [26, S. 283] schreiben in Bezug auf das kapazitierte WLP: „We recommend the use of Lagrangian heuristics rather than the usual greedy or interchange–type heuristics.“

Das Lagrange-Subproblem für Multiplikatoren $\sigma_v \in \mathbb{R}$ (zu (9.3b)) lautet

$$C(\sigma) = \min \sum_{i,v} (d_{iv}w_v - \sigma_v)x_{iv} + \sum_v \sigma_v \quad (9.4)$$

unter (9.3c) — (9.3f)

Wird zunächst für jedes $i \in V$ das Rucksackproblem mit stetigen Variablen

$$\begin{aligned} \gamma_i(\sigma) = \min & \sum_v (d_{iv}w_v - \sigma_v)x_{iv} \\ \text{unter} & \sum_v w_v x_{iv} = \mu \\ & 0 \leq x_{iv} \leq 1 \quad (v \in V) \end{aligned}$$

gelöst (im wesentlichen durch Sortieren nach zunehmendem Quotienten $(d_{iv}w_v - \sigma_v)/w_v$), so ist auch (9.4) ein Rucksackproblem:

$$C(\sigma) = \min \sum_i \gamma_i(\sigma)z_i \quad (9.5a)$$

$$\text{unter} \quad \sum_i z_i = p \quad (9.5b)$$

$$z_i \in \{0, 1\} \quad (i \in V) \quad (9.5c)$$

Aufgrund seiner simplen Nebenbedingung wird es gelöst, indem die z_i zu den p größten Zielfunktionskoeffizienten $\gamma_i(\sigma)$ zu Eins und die verbleibenden zu Null gesetzt werden. Um (9.3c) zu erfüllen, wird zum Schluß $x_{iv} = 0$ gesetzt, wenn $z_i = 0$ ist (für alle $i, v \in V$).

9.3.2.2 Darstellung der Heuristik

Unsere Heuristik für WLP (Algorithmus 9.2) kombiniert das Subgradienten-Verfahren zur Berechnung einer unteren Schranke mit dem location-allocation Prinzip zur Gewinnung möglichst guter zulässiger Lösungen von WLP. Dabei werden die Startzentren in Algorithmus 9.1 in jeder Iteration entsprechend der optimalen Lösung des Lagrange-Subproblems gewählt. Auf diese Weise geht duale Information in die primale Heuristik ein.

Algorithmus 9.2 Lagrange-Heuristik für WLP

1: $\sigma_v := 0, v \in V; best_lb := -\infty; best_ub := +\infty; \kappa := 1.0$

2: **repeat**

3: Berechne Optimallösung von (9.4) wie in 9.3.2.1 beschrieben:

$(z_i)_{i \in V}$ und $(x_{iv})_{i,v \in V}$ mit Zielfunktionswert $C(\sigma)$

- 4: $best_lb := \max\{C(\sigma), best_lb\}$
 - 5: $Z^0 := \{i \in V : z_i = 1\}$
 - 6: Führe Algorithmus 9.1 mit Startzentren Z_0 aus (ohne Runden in Zeile 7); dies liefert eine zulässige Lösung von WLP mit Zielfunktionswert ub .
 - 7: $best_ub := \min\{ub, best_ub\}$
 - 8: Berechne Subgradient: $\eta_v := 1 - \sum_i x_{iv}, v \in V$
 - 9: Aktualisiere Lagrange-Multiplikatoren:

$$\sigma_v := \sigma_v + \kappa((best_ub - C(\sigma)) / \sum_{v' \in V} \eta_{v'}^2) \eta_v, v \in V$$
 - 10: Aktualisiere κ /* siehe Text */
 - 11: **until** Abbruchkriterium erfüllt
-

9.3.2.3 Anmerkungen zur Implementierung

Wahl der Schrittweite. Der Parameter κ bestimmt die Schrittweite, mit der die Lagrange-Multiplikatoren in Richtung des Subgradienten verändert werden (Zeile 9). Es ist gängige Praxis, κ im Laufe des Subgradienten-Verfahrens in geometrischer Weise zu verkleinern, also

$$\kappa := r\kappa,$$

wobei $r \in (0, 1)$ fest gewählt ist. Diese Verkleinerung wird immer dann vorgenommen (Zeile 10), wenn seit der letzten Verkleinerung in N aufeinanderfolgenden Iterationen keine Verbesserung von $best_lb$ erreicht wurde. Hojati [52] verwendet $r = 0.5$ und $N = 30$. In unserer Implementierung können die Parameter r und N frei gewählt werden.

In Reeves [91, S. 270] wird empfohlen, den Schrittweiten-Parameter mit $\kappa = 2$ zu initialisieren. Für unsere Heuristik haben wir die Erfahrung gemacht, daß dies ein zu großer Wert ist, empirisch ist die Initialisierung mit dem Wert 1.0 besser (Zeile 1).⁵

Abbruchkriterien. In folgenden Situationen wird Algorithmus 9.2 beendet.

1. Die relative Abweichung $(best_ub - best_lb)/best_ub$ ist kleiner als eine vorgegebene Toleranz, zum Beispiel 0.005.

Damit hat man ein Optimalitätszertifikat für die beste in Zeile 6 bestimmte Lösung von WLP in der Hand. Aufgrund einer häufig gegebenen *Dualitätslücke* (positive Abweichung zwischen optimalem Zielfunktionswert von WLP und optimalem Wert des Lagrange-Duals) wird dieses Kriterium allerdings selten erreicht.

⁵Mit einem Startwert $\kappa = 2$ wurde beobachtet, daß die Lagrange-Funktion $C(\sigma)$ in den ersten Iterationen stark abnimmt; erst nachdem κ kleinere Werte angenommen hat, tritt das gewünschte Anwachsen der Lagrange-Funktion ein.

2. Die quadrierte Norm $\sum_{v' \in V} \eta_{v'}^2$ des Subgradienten ist (nahe bei) Null. In diesem Fall sind die Lagrange-Multiplikatoren optimal.
3. Der Schrittweiten-Parameter κ fällt unter einen kleinen Wert, zum Beispiel 0.001.

Dies ist das häufigste Abbruchkriterium. Es ist sinnvoll, da bei sehr kleinem κ nur noch marginale Veränderungen der Multiplikatoren und der Lagrange-Funktion $C(\sigma)$ stattfinden.

Reduktionskriterien. Aufgrund dualer Information ist es häufig möglich, Reduktionen vorzunehmen, also Variablen zu fixieren. Für WLP ist dies vor allem für die ganzzahligen Variablen z_i interessant. Wir haben ein Kriterium implementiert, das es erlaubt, z_i auf Null zu fixieren.

Ist für ein $i \in V$ in der optimalen Lösung des Rucksackproblems (9.5) $z_i = 0$ und gilt

$$C(\sigma) + \gamma_i(\sigma) - \gamma_{i'}(\sigma) \geq \text{best_ub}$$

für ein i' , das unter allen j mit $z_j = 1$ den Wert $\gamma_j(\sigma)$ maximiert, so kann z_i auf Null fixiert werden. Der Grund ist, daß die linke Seite der Ungleichung eine untere Schranke für den optimalen Zielfunktionswert ist, wenn $z_i = 1$ gilt.

Das Fixieren von $z_i = 0$ verkleinert das Lagrange-Subproblem ($\gamma_i(\sigma)$ muß nicht mehr berechnet werden) und vereinfacht die primale Heuristik (i muß nicht mehr als mögliches Zentrum berücksichtigt werden).

Lösen von Transportproblemen. Im Verlauf von Algorithmus 9.2 sind in Zeile 6 viele Transport-Probleme zu lösen. Dies kann bei großen Problemen sehr zeitaufwendig sein. Mit zwei Maßnahmen haben wir versucht, den Aufwand zu begrenzen.

1. Der Parameter `max_iteration` in Algorithmus 9.1 wird nicht zu groß gewählt, zum Beispiel mit dem Wert 10.
2. Jede Zentren-Menge, für die das Transportproblem gelöst wurde, wird in einer Hash-Tabelle festgehalten. Algorithmus 9.1 wird abgebrochen, wenn ein Z^k generiert wird, das in der Tabelle gespeichert ist. Weitere Iterationen würden nur bereits vorgenommene Berechnungen wiederholen und sind damit überflüssig. Auf diese Weise wird erreicht, daß kein Transportproblem mehrfach gelöst wird.

9.3.3 Rechenergebnisse

Die Anwendung der Lagrange-Heuristik 9.2 auf eine größere Zahl von Testproblemen soll Informationen über den durchschnittlichen Rechenzeitaufwand und die Qualität

der berechneten Bezirkszentren (im Sinne der Zielfunktion von WLP) liefern. Letzteres ist durch Vergleich von $best_ub$ und $best_lb$ am Ende der Heuristik möglich.

Eine große Abweichung zwischen diesen Werten kann durch ein unbefriedigendes Verhalten der primalen Heuristik verursacht sein, oder auch durch eine große Dualitätslücke für die gewählte Lagrange-Relaxation von WLP. (Wir machen die Annahme, daß das Subgradienten-Verfahren den Lagrange-Dual annähernd optimal löst.) Da in vielen Fällen tatsächlich eine große Abweichung gegeben ist, haben wir für kleine Probleme ($|V| \leq 100$) die Optimallösung von WLP mit dem Solver CPLEX berechnet. Die Ergebnisse zeigen, daß die primale Heuristik sehr gute Lösungen liefert, und tatsächlich in vielen Fällen eine große Dualitätslücke vorliegt.

9.3.3.1 Ergebnisse der Lagrange-Heuristik

Tabelle 9.1 zeigt die Ergebnisse von Algorithmus 9.2.⁶ Die ersten beiden Spalten enthalten die Dimension der Probleme; $n = |V|$ ist die Anzahl der KGE und p die Anzahl der Bezirke. Für jede dieser Größenklassen wurden mehrere Probleme gerechnet, ihre Anzahl steht in der vierten Spalte. Dabei wurde jedes der Probleme zweimal behandelt. Im ersten Durchlauf wurde die Entfernungsmessung zwischen den Orten der KGE durch quadriert-euklidische Distanz vorgenommen, im zweiten Fall durch KW-Distanz. Die Zeilen von Tabelle 9.1 gehören somit paarweise zusammen, die jedem Zeilenpaar zugrundeliegenden Testprobleme unterscheiden sich nur durch die Koeffizienten d_{iv} in der Zielfunktion (9.3a). Die dritte Spalte der Tabelle kennzeichnet das jeweilige Distanzmodell.

Die Qualität der erzielten Lösungen ist in den Spalten fünf und sechs aufgeführt. Für jedes Testproblem wurde die relative Abweichung der am Ende erzielten besten unteren Schranke $best_lb$ von der besten gefundenen Lösung (mit Zielfunktionswert $best_ub$) ermittelt. Der Durchschnitt und das Maximum der prozentualen Abweichung über alle Testprobleme einer Zeile sind angegeben. Spalte „#opt“ gibt die Anzahl der Probleme an, für welche die relative Abweichung unter 0.5% lag; dies wurde als Optimalitätszertifikat angesehen.

Die letzten beiden Spalten der Tabelle geben die Rechenzeit für obere Schranke (Algorithmus 9.1) und untere Schranke (Subgradienten-Verfahren) in Algorithmus 9.2 gemittelt über die Testprobleme jeder Zeile an. Die Einträge sind CPU-Sekunden auf einem PC mit 800 MHz Pentium-Prozessor.

Aus den Werten in Tabelle 9.1 lassen sich folgende Schlüsse ziehen.

1. Nur in wenigen Fällen gelingt es der Heuristik, die Optimalität der besten bestimmten Zentrenmenge zu garantieren. In der Regel beträgt die relative

⁶Es wurde mit $r = 0.5$ und $N = 20$ gerechnet. Zur Erzeugung der Testprobleme siehe S. 138.

n	p	d	Anz.	$(ub - lb)/ub$		#opt	t_{ub}	t_{lb}
				avg%	max%			
50	5	qe	20	4.55	19.61	4	1.24	3.38
50	5	kw	20	2.98	8.60	5	1.20	3.20
50	7	qe	20	7.15	17.75		3.10	4.89
50	7	kw	20	4.26	14.29	2	2.40	4.13
50	10	qe	20	11.80	24.32		4.01	4.69
50	10	kw	20	6.52	12.00		3.59	4.73
100	10	qe	20	5.06	13.05		52.33	21.68
100	10	kw	20	2.73	6.24	1	39.70	20.81
100	13	qe	20	8.95	19.30		66.98	22.28
100	13	kw	20	4.16	10.30		52.98	23.59
100	20	qe	20	13.68	28.07		72.62	19.77
100	20	kw	20	6.65	14.41		65.90	21.18
250	17	qe	10	4.25	6.47		1011.03	154.27
250	17	kw	10	2.27	3.84		850.76	149.25
250	25	qe	10	5.57	10.13		1014.33	167.01
250	25	kw	10	3.39	9.24		875.16	150.75
250	33	qe	10	11.78	20.48		804.56	123.93
250	33	kw	10	4.98	9.46		876.82	140.16
250	49	qe	10	16.51	37.96		663.69	103.45
250	49	kw	10	10.10	20.45		763.49	132.15
500	25	qe	2	4.74	5.34		6669.99	645.30
500	25	kw	2	2.25	2.60		4722.77	611.40
500	33	qe	2	7.98	10.02		5474.15	525.62
500	33	kw	2	3.08	3.66		5532.46	636.30
500	49	qe	2	9.63	10.90		5080.45	471.23
500	49	kw	2	5.48	7.21		3546.48	635.58
500	64	qe	2	11.46	12.16		5875.23	552.58
500	64	kw	2	5.13	5.27		5088.85	562.99

Tabelle 9.1: Rechenergebnisse für Algorithmus 9.2

Abweichung der Schranken mehrere Prozent, sogar Abweichungen über 20 Prozent kommen häufig vor. Wie oben ausgeführt, kann daraus kein Schluß über die Qualität der besten gefundenen Lösung gezogen werden. Jedoch ist der Informationswert einer „Gütegarantie“ von über 20% eher gering.

2. Die Abweichungen nehmen mit abnehmendem Quotienten n/p zu. Je weniger KGE im Durchschnitt auf einen Bezirk entfallen, desto mehr nimmt die durch die Heuristik gelieferte Information ab.
3. Bei KW-Distanzen ist die Lückengröße regelmäßig kleiner als bei quadriert-euklidischen Distanzen. Dabei ist in etwa ein Faktor von zwei für die Zeilenpaare in Tabelle 9.1 zu beobachten.⁷
4. Die Rechenzeit wächst mit zunehmender Problemgröße stark an; vor allem betrifft dies den Aufwand für die obere Schranke. Der Parameter `max_iteration` in Algorithmus 9.1 wurde auf 10 gesetzt. Somit wurden sehr viele Transportprobleme gelöst, was den hohen Rechenzeitbedarf erklärt.

9.3.3.2 Qualität der Schranken

Tabelle 9.2 vergleicht für Probleme mit $n = 50$ und $n = 100$ die von Algorithmus 9.2 bestimmten Schranken mit dem optimalen Zielfunktionswert der jeweiligen Probleme, der mit dem Solver CPLEX (Version 6.6) bestimmt wurde. Alle Probleme der Dimension $n = 50$ konnten problemlos in Rechenzeiten von unter fünf Minuten optimal gelöst werden (vorletzte und letzte Spalte). Für $n = 100$ konnte nur ein Teil der Probleme gelöst werden, da der maximal zur Verfügung gestellte Speicher von 200 MByte für den Branch&Bound-Suchbaum in einigen Fällen nicht ausreichte. Spalte vier gibt die Anzahl der von CPLEX optimal gelösten Probleme an (von jeweils 20).

Die fünfte Spalte enthält die Anzahl der Probleme, bei denen die von Algorithmus 9.2 gefundene beste Lösung optimal war. Die durchschnittliche und maximale prozentuale Abweichung der oberen und unteren Schranke vom optimalen Zielfunktionswert ist in den folgenden Spalten angegeben. Die Rechenzeiten für die Schranken entsprechen denen in Tabelle 9.1.

Die Daten lassen folgende Schlußfolgerungen zu:

1. Die obere Schranke liefert in den meisten Fällen die Optimallösung. Für alle (von CPLEX optimal gelösten) Probleme liefert sie eine Lösung, die weniger als ein Prozent vom Optimum abweicht.

⁷Dieser multiplikative Zusammenhang läßt sich wohl dadurch erklären, daß die Zielfunktionskoeffizienten in (9.3a) bei quadriert-euklidischen Distanzen ungefähr dem Quadrat derjenigen bei KW-Distanzen gleichkommen.

n	p	d	Anzahl		ob. Schranke		unt. Schranke		t_{ub}	t_{lb}	t_{CPLEX}	
			alle	opt	avg%	max%	avg%	max%			avg	max
50	5	qe	20	19	0.00	0.02	4.55	19.59	1.24	3.38	10.02	57.85
50	5	kw	20	20	0.00	0.00	2.98	8.60	1.20	3.20	8.77	39.71
50	7	qe	20	20	0.00	0.00	7.15	17.75	3.10	4.89	24.50	241.58
50	7	kw	20	19	0.05	0.93	4.21	14.29	2.40	4.13	8.92	67.17
50	10	qe	20	18	0.01	0.15	11.80	24.32	4.01	4.69	38.47	273.69
50	10	kw	20	19	0.00	0.02	6.52	12.00	3.59	4.73	9.02	45.62
100	10	qe	17	16	0.01	0.14	4.15	9.85	50.73	21.61	1286.74	5282.01
100	10	kw	20	19	0.01	0.13	2.72	6.24	39.70	20.81	832.58	5014.48
100	13	qe	13	11	0.01	0.11	6.77	12.20	68.01	23.17	1320.11	3704.90
100	13	kw	18	15	0.03	0.20	4.06	10.15	51.21	23.49	573.69	2507.77
100	20	qe	5	3	0.03	0.13	10.10	17.98	76.62	19.98	1598.27	3348.53
100	20	kw	15	10	0.13	0.86	5.88	10.16	59.19	21.21	874.35	2390.64

Tabelle 9.2: Rechenergebnisse für Algorithmus 9.2 — mit CPLEX optimal gelöste Probleme.

2. Die beste untere Schranke weicht regelmäßig um einige Prozent vom Optimum ab, manchmal über 20%. Unter der Voraussetzung, daß das Subgradienten-Verfahren den Lagrange-Dual annähernd optimal löst, läßt dies auf eine im allgemeinen große Dualitätslücke schließen. Die großen Abweichungen zwischen den Schranken in Tabelle 9.1 lassen sich somit wohl durch diese Lücke erklären.
3. Die Schwierigkeit des Problems WLP zeigt sich an dem großen Zeit- und Speicheraufwand, den CPLEX schon bei 100 KGE benötigt. Unter den Problemen mit $n = 100$ und $p = 20$ konnten bei quadriert-euklidischer Distanzmessung nur 25 Prozent optimal gelöst werden.⁸

9.3.3.3 Iterationenzahl bei Berechnung der oberen Schranke

Der Aufwand zur Berechnung der oberen Schranke in Algorithmus 9.2 wächst nach Tabelle 9.1 mit zunehmender Problemgröße stark an. Die gewählte Einstellung von `max_iteration=10` für die primale Heuristik 9.1 könnte zu groß sein. Versuche mit einem kleineren Wert sollten prüfen, ob auch dadurch Rechenzeit eingespart werden kann. Tabelle 9.3 stellt die Ergebnisse dar.

Für jeden der Werte `max_iteration=1,2,5` sind Spalten entsprechend denjenigen aus Tabelle 9.2, welche die obere Schranke betreffen, aufgeführt. Es ergeben sich die folgenden Beobachtungen.

1. Wird lediglich eine Iteration von Algorithmus 9.1 durchgeführt, so erhält man eine nur mangelhafte Heuristik zur Lösung von WLP. Die aus der Lösung des Lagrange-Subproblems resultierende Zentrenmenge Z^0 (Zeile 5 von Algorithmus 9.2) stellt in der Regel keine gute Lösung von WLP dar. Der Zielfunktionswert der besten im Verlauf von Algorithmus 9.2 bestimmten solchen Zentrenmenge kann mehr als 50 Prozent über der Optimallösung liegen.⁹
2. Werden dagegen zwei Iterationen durchgeführt, führt dies zwar zu einer ungefähren Verdoppelung der Rechenzeit, doch die Abweichung von der Optimallösung liegt nun unter fünf Prozent; im Durchschnitt unter einem Prozent.

⁸Versuche, auch ein Problem mit $n = 250$ mit CPLEX zu lösen, waren auf ganzer Linie erfolglos. Nach mehreren Stunden Rechenzeit war die Speichergrenze von 200 MByte erreicht, die Lücke zwischen oberer und unterer Schranke des Branch&Bound lag trotzdem noch bei 18 Prozent. Zum Vergleich: Dasselbe Problem wurde von der Lagrange-Heuristik in weniger als zehn Prozent der Rechenzeit gelöst; mit einer Abweichung der Schranken von 0.76%.

⁹Auf Seite 213 wurde ausgeführt, daß die von Hojati [52] verwendete Heuristik ungenügend ist; hier wird der Grund ersichtlich. Hojatis Heuristik wählt als Lösung von WLP die Zentrenmenge Z^0 zu den Lagrange-Multiplikatoren, die die beste untere Schranke ergeben. Obwohl in den Spalten zu `max_iteration=1` in Tabelle 9.3 die Lagrange-Multiplikatoren *aller* Iterationen Berücksichtigung finden, sind die Ergebnisse unbefriedigend.

n	p	d	Anz	max_iteration = 1				max_iteration = 2				max_iteration = 5			
				opt	avg%	max%	t_{ub}	opt	avg%	max%	t_{ub}	opt	avg%	max%	t_{ub}
50	5	qe	20	9	2.83	16.53	0.44	15	0.16	2.37	0.83	19	0.00	0.02	1.22
50	5	kw	20	11	1.28	6.77	0.41	12	0.54	4.15	0.85	20	0.00	0.00	1.18
50	7	qe	20	1	6.58	30.80	1.02	12	0.09	0.70	2.10	20	0.00	0.00	3.07
50	7	kw	20	3	2.44	8.96	0.73	13	0.27	2.13	1.49	19	0.05	0.93	2.41
50	10	qe	20	1	11.99	33.81	1.33	10	0.37	2.95	2.53	18	0.01	0.15	4.01
50	10	kw	20	5	2.40	8.76	1.09	12	0.49	2.19	2.13	19	0.00	0.02	3.53
100	10	qe	17	0	11.96	42.90	12.07	7	0.41	1.42	26.45	16	0.01	0.13	47.77
100	10	kw	20	2	1.70	6.04	10.12	6	0.27	1.08	21.39	19	0.01	0.11	36.67
100	13	qe	13	1	16.72	56.62	14.61	2	0.83	2.85	32.41	11	0.06	0.41	66.23
100	13	kw	18	0	2.58	10.34	12.08	3	0.38	2.80	24.05	14	0.06	0.58	46.68
100	20	qe	5	0	20.05	52.77	14.96	2	1.07	3.49	35.71	4	0.10	0.48	71.84
100	20	kw	15	0	3.78	8.51	15.59	1	0.95	2.91	32.10	10	0.14	0.93	55.38

Tabelle 9.3: Einfluß des Parameters max_iteration auf die Lösungsgüte für die Probleme aus Tab. 9.2.

3. Bei fünf Iterationen beträgt die maximale Abweichung vom Optimum weniger als ein Prozent, bei einer nochmaligen Verdoppelung des Aufwandes. Die Ergebnisse für fünf Iterationen unterscheiden sich sowohl hinsichtlich Qualität der Lösung als auch Rechenzeit nur wenig von denen in Tabelle 9.2, wo mit maximal zehn Iterationen gerechnet wurde. Dies zeigt einen Konvergenzeffekt. Durch die auf Seite 217 beschriebene Technik wird ein wiederholtes Lösen gleicher Transportprobleme vermieden und die primale Heuristik kann oft nach wenigen Iterationen abgebrochen werden, wenn sie zu keiner neuen Schranke führt.

Zusammenfassung und Ausblick

Beiträge der Arbeit

Hauptziel der vorliegenden Arbeit war die Untersuchung von OR-Modellen zur Aufteilung geographischer Gebiete; vor allem ging es dabei um die Entwicklung von Algorithmen, die auf gleichmäßiger Baumzerlegung basieren. Motivation dieses Ansatzes ist die Tatsache, daß Optimierungsprobleme auf Bäumen in der Regel einfacher zu lösen sind, als solche auf Graphen mit Zyklen. Im folgenden fassen wir die wesentlichen Beiträge der Arbeit zusammen.

1. Zunächst wurde in Kapitel 2 die Anwendungsseite umfassend dargestellt. Für den Praktiker, der sich vor ein Gebietsaufteilungsproblem gestellt sieht, ergibt sich daraus eine Referenz, in die er sein Problem einordnen kann. Es wird deutlich, welche (anwendungsübergreifenden) Kriterien bei Gebietsaufteilungsproblemen im allgemeinen relevant sind.
2. Gleichmäßige Baumzerlegungen wurden in Teil II untersucht. Es wurde ein Klassifikationsschema für Optimierungsprobleme diesen Typs eingeführt (Abschnitt 4.2). Damit wird die Literatur zum Thema in einen einheitlichen Rahmen gebracht. Ferner wurden Komplexitätsergebnisse diskutiert und ergänzt (Abschnitt 4.3). Eigenentwicklungen wurden für Probleme mit Zielfunktion vom Typ $\|\cdot\|_\infty$ (Minimierung der maximalen Abweichung vom mittleren Gewicht der Teilbäume) geleistet:
 - Es wurde (unseres Wissens zum ersten Mal) nachgewiesen, daß die Probleme $T/\cdot/\cdot/\|\cdot\|_\infty$, $T/\cdot/p/\|\cdot\|_\infty$ und $T/Zent/\cdot/\|\cdot\|_\infty$ polynomial lösbar sind (Kapitel 5).
 - Da die dafür entwickelten polynomialen Algorithmen eine hohe Laufzeit-Komplexität haben, wurde für das aus Anwendungssicht besonders interessante Problem $T/\cdot/p/\|\cdot\|_\infty$ ein effizientes Branch&Bound-Verfahren vorgestellt (Kapitel 6).

3. Durch Anwendung gleichmäßiger Baumzerlegung wurden in Teil III neue Verfahren für location–allocation Modelle der Gebietsaufteilung erarbeitet.

- Das Runden einer durch das Transportproblem generierten Lösung auf ganzzahlige Werte (die sog. Splitauflösung) kann mit gleichmäßiger Baumzerlegung in einem optimalen Sinn unter vielfältigen Zielsetzungen vorgenommen werden; dies in der Regel sehr effizient. (Abschnitte 7.2.4 und 7.2.5)
- Die Bestimmung von Bezirken, die der Zusammenhangsforderung genügen, kann durch gleichmäßige Baumzerlegung auch für große Probleme (über 500 KGE) mit Rechenzeiten im Sekundenbereich realisiert werden. Ein wesentlicher Vorteil gegenüber dem Subgradienten–Verfahren von Zoltners und Sinha [115] ist, daß dabei keine Bezirkszentren vorgegeben werden müssen. (Abschnitt 8.3)

4. Die Bestimmung von Bezirkszentren / Vertreterstandorten (der location Schritt) wurde in Kapitel 9 behandelt. Beitrag unserer Arbeit ist dabei die Weiterentwicklung der von Hojati [52] vorgeschlagenen Lagrange–Heuristik zu einem Verfahren, das sehr gute Lösungen liefert (empirisch nicht schlechter als ein Prozent vom Optimum).

Fragestellungen für weitere Forschung

Im folgenden zeigen wir Aspekte auf, wie zukünftige Forschung die Ergebnisse der vorliegenden Arbeit weiterführen kann.

Erweiterung auf Modelle mit mehreren Aktivitätsmaßen. Gebietsaufteilungen mittels Anwendung gleichmäßiger Baumzerlegung in location–allocation Modellen zu berechnen, ist der Grundgedanke unserer Arbeit. Jedoch kann diese Idee allgemeiner verstanden werden, als es ihre konkrete Ausgestaltung in den Kapiteln 7 und 8 nahelegt. Darin wurden Bäume als Basisstruktur von Transport– oder Netzwerkflußproblem gewonnen. Diese Möglichkeit geht verloren, wenn allgemeinere Modelle zur Zuordnung von KGE zu Zentren zum Einsatz kommen.

Besonders wäre hierbei an Modelle zu denken, die mehr als ein Größenattribut der KGE einbeziehen.¹⁰ Um dennoch Nutzen aus der kombinatorisch einfachen Struktur von Baumzerlegungen zu ziehen, wären Heuristiken zu entwickeln, die aus der Basisinformation allgemeinerer Modelle einen Baum (oder Bäume) extrahieren. Wie

¹⁰Solche Modelle sind insbesondere für die Einteilung von Verkaufsgebieten von Bedeutung, wenn mehr als ein Aktivitätsmaß balanciert werden muß (Zoltners und Sinha [115]).

in 7.2.5.4 beispielhaft gezeigt wurde, kann das Zerlegen eines Baumes (in diesem Fall der Splitadjazenz) auch bezüglich mehrerer Kriterien unter Umständen effizient durchzuführen sein.

Verbesserte polynomiale Algorithmen zur gleichmäßigen Baumzerlegung.

Die in Kapitel 5 entwickelten Algorithmen für gleichmäßige Baumzerlegung unter $\|\cdot\|_\infty$ -Zielsetzung haben eine polynomiale Laufzeit, jedoch sind die Abschätzungen von hohem Grad. Hier wäre zu fragen, ob sich durch eine Verfeinerung der Analyse ein niedrigerer Grad beweisen läßt, oder es wären Algorithmen mit besserem Laufzeitverhalten zu entwickeln. Ein weiterer „Schönheitsfehler“ unserer Algorithmen ist die Abhängigkeit ihrer Laufzeit von der Größe der Eckengewichte. Willkommen wären Verfahren, die im strengen Sinn polynomial sind.¹¹

Zentrenbestimmung unter Einbeziehung von KW-Zusammenhang. Das in Kapitel 9 von Hojati [52] übernommene Modell WLP (9.3) kann um das Konzept des KW-Zusammenhangs erweitert werden, indem Ungleichungen der Form

$$x_{iv} \leq x_{i, \text{pre}_i(v)}$$

(vgl. 8.1.3.3) hinzugefügt werden. Die in 9.3.2.1 vorgenommene Lagrange-Relaxation führt dann immer noch zu einem einfachen Subproblem, das mit dynamischer Programmierung gelöst werden kann. Mit einer geeigneten primalen Heuristik könnte darauf aufbauend eine Lagrange-Heuristik für die Bestimmung guter Bezirkszentren konstruiert werden, welche die Zusammenhangsforderung für die Bezirke einbezieht.

¹¹Versuche, die streng polynomiale sog. *shifting* Methode vom Spezialfall kettenförmiger Bäume (der in Liverani et al. [63] behandelt ist) auf den allgemeinen Fall des Problems $T/\cdot/p/\|\cdot\|_\infty$ zu übertragen, waren bisher nicht erfolgreich. (Dies wurde dem Verfasser auch von den Autoren der zitierten Arbeit mitgeteilt.)

Glossar graphentheoretischer Begriffe

In der vorliegenden Arbeit wird nur von Grundbegriffen aus der Graphentheorie Gebrauch gemacht; wir setzen voraus, daß der Leser damit vertraut ist.¹² Da die Graphentheorie jedoch keine einheitliche und allgemein verwendete Benennung ihrer Konzepte hervorgebracht hat, soll die folgende Liste helfen, im Zweifelsfall zu klären, was die in unserer Arbeit benutzten Begriffe bedeuten.

Die Anordnung der Aufstellung folgt dem Alphabet. Gegebenenfalls verweisen wir auf Stellen im Text, die eine Definition oder ausführlichere Erläuterung bieten.

adjazent sind zwei \rightarrow Ecken, wenn sie mit einer gemeinsamen \rightarrow Kante \rightarrow inzidieren.

Baum Ein \rightarrow zusammenhängender Graph $G = (V, E)$ mit $|E| = |V| - 1$. Siehe S. 50.

Baumzerlegung Siehe S. 53.

bewerteter Graph \rightarrow Ecken und / oder \rightarrow Kanten sind reelle Werte zugeordnet, sie werden als Gewichte bzw. Längen bezeichnet. Oft sind sie auf nichtnegative oder positive Werte beschränkt.

bipartiter Graph Die Eckenmenge V von $G = (V, E)$ läßt sich aufteilen in zwei Teilmengen V', V'' , die beide einen \rightarrow Subgraphen mit leerer Kantenmenge \rightarrow induzieren. Mit anderen Worten: jede \rightarrow Kante in G \rightarrow inzidiert mit einer \rightarrow Ecke aus V' und mit einer aus V'' .

Blatt In einem \rightarrow Wurzelbaum eine \rightarrow Ecke ohne \rightarrow Söhne.

d -Baum Ein spezieller Typ von \rightarrow Baum, siehe S. 4.1.1.2.

¹²Lehrbücher zur Graphentheorie sind (unter vielen anderen) Berge [15], Jungnickel [58] oder Volkmann [108]. Darin wird weitaus mehr Material behandelt, als in unserer Arbeit eine Rolle spielt.

dünnere Graph Ein \rightarrow Graph aus einer Familie von Graphen, für welche die Kantenzahl nicht stärker anwächst als die Eckenzahl, d.h. $|E| = O(|V|)$. Zum Beispiel bilden die \rightarrow planaren Graphen eine solche Familie.

ebener Graph Synonym zu \rightarrow planarer Graph.

Ecke Element der Eckenmenge V eines \rightarrow Graphen $G = (V, E)$.

Einbettung, eingebetteter Graph Ein \rightarrow Graph, dessen \rightarrow Ecken durch Punkte und dessen \rightarrow Kanten durch diese Punkte verbindende Jordankurven in der Ebene repräsentiert sind.

Von Interesse sind vor allem *überkreuzungsfreie* Einbettungen, das heißt: (i) die Jordankurven verschiedener Kanten schneiden sich nicht in Punkten, die keine Ecke repräsentieren und (ii) die Jordankurve keiner Kante enthält den Punkt zu einer mit dieser Kante nicht \rightarrow inzidenten Ecke.

Endecke eines \rightarrow Baumes ist jede \rightarrow Ecke vom \rightarrow Grad 1 in diesem Baum.

eulerscher Graph Ein \rightarrow Graph, in welchem alle \rightarrow Ecken einen geraden \rightarrow Grad haben.

gerichteter Graph Ein \rightarrow Graph, in dem die Kantenrichtung ausdrücklich beachtet wird.

gerichteter Weg Ein \rightarrow Weg, in welchem jeweils Anfang und Ende benachbarter Kanten mit einer gemeinsamen Ecke inzidieren.

Gittergraph Ein spezieller Typ von \rightarrow Graph, siehe 4.4.

Grad einer \rightarrow Ecke ist die Anzahl der mit ihr \rightarrow inzidenten \rightarrow Kanten.

Graph Definiert durch Eckenmenge $V \neq \emptyset$ und Kantenmenge E . Für unsere Zwecke ist es auch reichend, E als Teilmenge von $V \times V - \{(v, v) : v \in V\}$ zu definieren. V und E sind endlich. (Wir betrachten nur einfache endliche Graphen.)

Per Definition sind die \rightarrow Kanten gerichtet, die Richtung spielt jedoch oft keine Rolle und wird dann „vergessen“ (\rightarrow gerichteter Graph, \rightarrow ungerichteter Graph).

induzierter Subgraph Eine nichtleere Teilmenge V' der Eckenmenge des \rightarrow Graphen $G = (V, E)$ *induziert* den \rightarrow Subgraph $G' = (V', E')$ mit $E' = \{(u, v) \in E : u, v \in V'\}$.

inzident sind eine \rightarrow Ecke v und eine \rightarrow Kante e , wenn v Anfang oder Ende von e ist.

Kante Element der Kantenmenge eines \rightarrow Graphen. Die Kante (u, v) hat den Anfang u und das Ende v . In \rightarrow ungerichteten Graphen wird die Richtung einer Kante vergessen; folglich bezeichnet hier (u, v) und (v, u) dieselbe Kante.

Kette Ein spezieller Typ von \rightarrow Baum, siehe S. 4.1.1.2.

Kontraktion einer \rightarrow Kante ist das Entfernen der Kante aus dem Graphen bei gleichzeitiger Identifikation der mit ihr \rightarrow inzidenten \rightarrow Ecken. Wir verwenden Kontraktion nur in \rightarrow Wurzelbäumen (siehe 4.1.1.1).

In einem mit Eckengewichten \rightarrow bewerteten Graph werden beim Kontrahieren die Gewichte zusammenfallender Ecken addiert.

Kreis Ein \rightarrow gerichteter Weg, der \rightarrow Zyklus ist.

Nachbarschaftsgraph (In dieser Arbeit) ein \rightarrow Graph, der das Benachbartsein kleinster Gebietseinheiten (KGE) beschreibt. Siehe 8.1.1.

Nachfolger einer \rightarrow Ecke v in einem \rightarrow Wurzelbaum sind alle Ecken, die bezüglich der Ordnung größer sind. Anschaulich sind es die Ecken „unterhalb“ von v , ohne v selbst. Siehe 4.1.1.1.

Netzwerk Synonym zu \rightarrow gerichteter Graph. Die Bezeichnung wird verwendet, wenn auf den Kanten des Graphen ein Fluß „stattfindet“.

Pfad Folge paarweise verschiedener \rightarrow Ecken, wobei benachbarte adjazent sind. (Wir verwenden den Begriff nur für Bäume, hier sind Pfade eindeutig durch die Angabe der ersten und der letzten Ecke bestimmt. Siehe 4.1.1.1.)

planarer Graph Ein \rightarrow Graph, der sich überkreuzungsfrei in die Ebene \rightarrow einbetten läßt.

Prädedenz Ein \rightarrow gerichteter Graph ohne \rightarrow Kreise.

Quotient Ein \rightarrow Graph, der durch \rightarrow Kontraktion einer oder mehrerer \rightarrow Kanten eines Graphen gebildet wird. Siehe 4.1.1.1.

Raube Ein spezieller Typ von \rightarrow Baum, siehe S. 4.1.1.2.

Sohn In einem \rightarrow Wurzelbaum ein \rightarrow Nachfolger einer Ecke v , der zu v \rightarrow adjazent ist. Siehe 4.1.1.1.

Spinne Ein spezieller Typ von \rightarrow Baum, siehe S. 4.1.1.2.

Stern Ein spezieller Typ von \rightarrow Baum, siehe S. 4.1.1.2.

Subgraph des \rightarrow Graphen $G = (V, E)$ ist jeder Graph $G' = (V', E')$ mit $V' \subseteq V$ und $E' \subseteq E$.

Teilbaum Ein \rightarrow zusammenhängender \rightarrow Subgraph eines \rightarrow Baumes. Siehe 4.1.1.1.

ungerichteter Graph Ein \rightarrow Graph, in welchem die Richtung der \rightarrow Kanten ausdrücklich vergessen wird. (Damit einher geht die Identifikation der Kanten (u, v) und (v, u) .)

Vorgänger

In einem \rightarrow Wurzelbaum ist der Vorgänger einer von der \rightarrow Wurzel verschiedenen \rightarrow Ecke diejenige Ecke, unter deren \rightarrow Söhnen v ist. Siehe 4.1.1.1.

In einer \rightarrow Präzedenz sind die Vorgänger einer Ecke v alle Ecken, die Ende einer Kante mit Anfang v sind.

Wald Ein \rightarrow Graph, der aus einem \rightarrow Baum durch Entfernen von \rightarrow Kanten hervorgeht.

Weg Eine Folge von \rightarrow Kanten, wobei benachbarte mit einer gemeinsamen \rightarrow Ecke \rightarrow inzidieren und nicht benachbarte Kanten nicht mit einer gemeinsamen Ecke inzidieren. Erlaubt ist jedoch, daß die erste und die letzte Kante mit einer gemeinsamen Ecke inzidieren (\rightarrow Zyklus), mit dieser darf dann aber keine andere Kante des Weges inzidieren.

Siehe auch \rightarrow gerichteter Weg.

Wurzel

In einem \rightarrow Wurzelbaum eine besonders ausgezeichnete \rightarrow Ecke.

In einer \rightarrow Präzedenz eine Ecke, welche von allen anderen Ecken aus durch einen \rightarrow gerichteten Weg erreichbar ist. (Eine Präzedenz kann höchstens eine Wurzel besitzen.)

Wurzelbaum Ein \rightarrow Baum, in welchem eine \rightarrow Ecke als *Wurzel* ausgezeichnet ist. Dies stiftet eine Ordnung auf den Ecken, in der die Wurzel das minimale Element ist. Siehe 4.1.1.1.

wurzeln Ein \rightarrow Teilbaum eines \rightarrow Wurzelbaumes, der von einer \rightarrow Ecke v und allen ihren \rightarrow Nachfolgern \rightarrow induziert wird, *wurzelt* in v . Siehe 4.1.1.1.

zusammenhängender Graph Ein \rightarrow Graph, in welchem zu jedem Paar von \rightarrow Ecken ein \rightarrow Weg existiert, dessen erste \rightarrow Kante mit der einen Ecke und dessen letzte Kante mit der anderen Ecke \rightarrow inzidiert.

Zyklus Ein \rightarrow Weg, dessen erste und letzte Kante mit einer gemeinsamen \rightarrow Ecke \rightarrow inzidieren. Hat der Weg nur zwei Kanten, müssen diese in zwei gemeinsamen Ecken inzidieren.

Literaturverzeichnis

- [1] El Houssaine Aghezzaf, Thomas L. Magnanti, and Laurence A. Wolsey. Optimizing constrained subtrees of trees. *Math. Program.*, 71A:113–126, 1995.
- [2] El Houssaine Aghezzaf and Laurence A. Wolsey. Modelling piecewise linear concave costs in a tree partitioning problem. *Discrete Appl. Math.*, 50:101–109, 1994.
- [3] Niv Ahituv and Oded Berman. *Operations Management of Distributed Service Networks*. Applications of Modern Technology in Business. Plenum Press, New York, 1988.
- [4] Ravindra K. Ahuja, Thomas L. Magnanti, and James B. Orlin. *Network Flows*. Prentice Hall, Englewood Cliffs, NJ, 1993.
- [5] Micah Altman. Traditional districting principles. *Social Science History*, 22:159–200, 1998.
- [6] F. Andria, P. Chiancone, and S. Piraino. Die Anwendung der Dynamischen Optimierung bei der Sozial–Sanitären Bezirkseinteilung. *Zeitschr. Oper. Res.*, 23:B33–B43, 1979.
- [7] Joanna R. Baker, Edward R. Clayton, and Laurence J. Moore. Redesign of primary response areas for county ambulance services. *Europ. J. Opl. Res.*, 41:23–32, 1989.
- [8] Jürgen Bär. *Bevölkerungsgeographie*. Ulmer, Stuttgart, 1997.
- [9] I. Barany, J. Edmonds, and L.A. Wolsey. Packing and covering a tree by subtrees. *Combinatorica*, 6:221–233, 1986.
- [10] Cynthia Barnhart, Ellis L. Johnson, George L. Nemhauser, Martin W. P. Savelsbergh, and Pamela H. Vance. Branch–and–prize: Column generation for solving huge integer programs. *Operations Research*, 46:316–329, 1998.

- [11] J. E. Beasley. Lagrangean heuristics for location problems. *Europ. J. Opl. Res.*, 65:383–399, 1993.
- [12] Ronald Becker, Isabella Lari, Mario Lucertini, and Bruno Simeone. Max–min partitioning of grid graphs into connected components. *Networks*, 32:115–125, 1998.
- [13] Ronald I. Becker, Stephen R. Schach, and Yehoshua Perl. A shifting algorithm for min–max tree partitioning. *J. ACM*, 29:58–67, 1982.
- [14] Richard Bellman. *Dynamic programming*. Princeton University Press, 1957.
- [15] Claude Berge. *Graphs*. Elsevier, Amsterdam, second rev. edition, 1985.
- [16] J. Blazewicz, K. Ecker, G. Schmidt, and J. Weglarz. *Scheduling in Computer and Manufacturing Systems*. Springer, Berlin, 1993.
- [17] Lawrence D. Bodin. A districting experiment with a clustering algorithm. *Annals of the New York Academy of Sciences*, pages 209–214, 1973.
- [18] Jean-Marie Bourjolly, Gilbert Laporte, and Jean-Marc Rousseau. Decoupage electoral automatisé: application à l’île de Montréal. *INFOR*, 19:113–124, 1981.
- [19] E. Andrew Boyd. Polyhedral results for the precedence-constrained knapsack problem. *Discrete Appl. Math.*, 41:185–201, 1993.
- [20] Bundeswahlgesetz (BWG). In: Rechtsgrundlagen für die Wahl zum 14. Deutschen Bundestag, Informationen des Bundeswahlleiters, März 1998.
- [21] Dreizehntes Gesetz zur Änderung des Bundeswahlgesetzes vom 15. November 1996. In: Rechtsgrundlagen für die Wahl zum 14. Deutschen Bundestag, Informationen des Bundeswahlleiters, März 1998.
- [22] Geon Cho and Dong X. Shaw. A depth-first dynamic programming algorithm for the tree knapsack problem. *INFORMS J. Comput.*, 9:431–438, 1997.
- [23] Geon Cho and Dong X. Shaw. Models and implementation techniques for local access telecommunication network design. In G. Yu, editor, *Industrial Applications of Combinatorial Optimization*, pages 316–344. Kluwer, Dordrecht, 1998.
- [24] Carmen Cirincione, Thomas A. Darling, and Timothy G. O’Rourke. Assessing South Carolina’s 1990s congressional districting. *Political Geography*, 19:189–211, 2000.

- [25] James B. Cloonan. A note on the compactness of sales territories. *Management Science*, 19(Part I):469, 1972.
- [26] G. Cornuejols, R. Sridharan, and J. M. Thizy. A comparison of heuristics and relaxations for the capacitated plant location problem. *Europ. J. Opl. Res.*, 50:280–297, 1991.
- [27] George B. Dantzig and Mukund N. Thapa. *Linear Programming 1: Introduction*. Springer, Berlin, 1997.
- [28] Richard F. Deckro. Multiple objective districting: A general heuristic approach using multiple criteria. *Opl. Res. Quart.*, 28:953–961, 1977.
- [29] Pietro Grilli di Cortona, Cecilia Manzi, Alline Pennisi, Federica Ricca, and Bruno Simeone. *Evaluation and Optimization of Electoral Systems*. SIAM monographs on discrete mathematics and applications. SIAM, Philadelphia, 1998.
- [30] Wolfgang Domschke and Andreas Drexl. *Logistik: Standorte*. Oldenbourg, München, 1996.
- [31] Andreas Drexl and Knut Haase. Fast approximation methods for sales force deployment. *Management Science*, 45:1307–1323, 1999.
- [32] Chris Easingwood. A heuristic approach to selecting sales regions and territories. *Opl. Res. Quart.*, 24:527–534, 1973.
- [33] Jaques A. Ferland and Gilles Gu enette. Decision support system for the school districting problem. *Operations Research*, 38:15–21, 1990.
- [34] Carlo Filippi and Giorgio Romanin-Jacur. Optimal allocation of two fixed service units acting as $M/G/1$ queues. *Transp. Sci.*, 30:60–74, 1996.
- [35] Bernhard Fleischmann and Jannis N. Paraschis. Solving a large scale districting problem: a case report. *Comput. & Opns. Res.*, 15:521–533, 1988.
- [36] Streit um Zuschnitt der Wahlkreise. Frankfurter Allgemeine Zeitung, Seite 2, 10.11.2000.
- [37] Greg N. Frederickson. Fast algorithms for shortest paths in planar graphs, with applications. *SIAM J. Comput.*, 16:1004–1022, 1987.

- [38] Greg N. Frederickson. Optimal algorithms for tree partitioning. In Alok Agarwal, editor, *Discrete algorithms. Proceedings of the 2nd annual ACM–SIAM symposium*, pages 168–177, New York, 1991. ACM.
- [39] Max-Planck-Institut für ausländisches öffentliches Recht und Völkerrecht. Wahlkreiseinteilung in westlichen europäischen Demokratien, den USA und Kanada. *Zeitschrift für ausländisches öffentliches Recht und Völkerrecht*, 57:633–676, 1997.
- [40] Michael R. Garey and David S. Johnson. *Computers and Intractability. A Guide to the Theory of NP-completeness*. Freeman, New York, 1979.
- [41] R. S. Garfinkel and G. L. Nemhauser. Optimal political districting by implicit enumeration techniques. *Management Science*, 16:B495–B508, 1970.
- [42] John A. George, Bruce W. Lamar, and Chris A. Wallace. Political district determination using large-scale network optimization. *Socio-Economic Planning Sciences*, 31:11–28, 1997.
- [43] Gesetz zur Neueinteilung der Wahlkreise für die Wahl zum Deutschen Bundestag (Wahlkreisneueinteilungsgesetz — WKNeuG). Bundesgesetzblatt Jahrgang 1998 Teil I Nr. 42, Juli 1998.
- [44] Thomas A. Glaze and Charles B. Weinberg. A sales territory alignment program and account planning system. In R. Bagozzi, editor, *Sales management: New Developments from Behavioral and Decision Model Research*, pages 325–343, Cambridge, MA, 1979. Marketing Science Institute.
- [45] A. D. Gordon. A survey of constrained classification. *Comput. Stat. Data Anal.*, 21:17–29, 1996.
- [46] Horst W. Hamacher. *Mathematische Lösungsverfahren für planare Standortprobleme*. Vieweg, Wiesbaden, 1995.
- [47] Saïd Hanafi, Arnaud Freville, and Polo Vaca. Municipal solid waste collection: An effective data structure for solving the sectorization problem with local search methods. *INFOR*, 37:236–254, 1999.
- [48] Friedrich Hanssmann. *Quantitative Betriebswirtschaftslehre*, page 175ff. Oldenbourg, München, zweite Auflage, 1985.
- [49] Robert E. Helbig, Patrick K. Orr, and Robert R. Roediger. Political redistricting by computer. *Comm. ACM*, 15:735–741, 1972.

- [50] S. W. Hess, J. B. Weaver, H. J. Siegfeldt, J. N. Whelan, and P. A. Zitlau. Nonpartisan political redistricting by computer. *Management Science*, 13:998–1008, 1965.
- [51] Sidney W. Hess and Stuart A. Samuels. Experiences with a sales districting model: criteria and implementation. *Management Science*, 18(Part II):P-41 – P-54, 1971.
- [52] Mehran Hojati. Optimal political districting. *Comput. & Opns. Res.*, 23:1147–1161, 1996.
- [53] David L. Horn and Charles R. Hampton. Practical application of district compactness. *Political Geography*, 12:103–120, 1993.
- [54] R. S. Howick and M. Pidd. Sales force deployment models. *Europ. J. Opl. Res.*, 48:295–310, 1990.
- [55] Toshihide Ibaraki. *Enumerative approaches to combinatorial optimization. Parts 1,2*, volume 10–11 of *Annals of Operations Research, Book Edition*. J.C. Baltzer, Basel, 1988.
- [56] David S. Johnson. The NP-completeness column: An ongoing guide. III. *J. Algorithms*, 3:182–195, 1982.
- [57] D.S. Johnson and K.A. Niemi. On knapsacks, partitions, and a new dynamic programming technique for trees. *Math. Oper. Res.*, 8:1–14, 1983.
- [58] Dieter Jungnickel. *Graphs, Networks and Algorithms*. Springer, Berlin, 1999.
- [59] Hans Kellerer, Ulrich Pferschy, and Maria Grazia Speranza. An efficient approximation scheme for the subset–sum problem. In *Proc. ISAAC'97*, volume 1350 of *Lecture Notes in Computer Science*, pages 394–403, Berlin, 1997. Springer.
- [60] Donald E. Knuth. *Seminumerical Algorithms*, volume 2 of *The Art of Computer Programming*. Addison–Wesley, 1969.
- [61] Sukhamay Kundu and Jayadev Misra. A linear tree partitioning algorithm. *SIAM J. Comput.*, 6:151–154, 1977.
- [62] R.L.M.J. van de Leensel, C.P.M. van Hoesel, and J.J. van de Klundert. Lifting valid inequalities for the precedence constrained knapsack problem. *Math. Program. Ser. A*, 86:161–185, 1999.

- [63] Marco Liverani, Aurora Morgana, Bruno Simeone, and Giovanni Storchi. Path equipartition in the Chebyshev norm. *Europ. J. Opl. Res.*, 123:428–436, 2000.
- [64] Mario Lucertini, Yehoshua Perl, and Bruno Simeone. Most uniform path partitioning and its use in image processing. *Discrete Appl. Math.*, 42:227–256, 1993.
- [65] Thomas L. Magnanti and Laurence A. Wolsey. Optimal trees. In *Network Models*, volume 7 of *Handbooks in Operations Research and Management Science*, pages 503–615. Elsevier, Amsterdam, 1995.
- [66] Maurizio Maravalle and Bruno Simeone. A spanning tree heuristic for regional clustering. *Commun. Statist. — Theory Meth.*, 24:625–639, 1995.
- [67] Maurizio Maravalle, Bruno Simeone, and Rosella Naldini. Clustering on trees. *Comput. Stat. Data Anal.*, 24:217–234, 1997.
- [68] Paul G. Marlin. Application of the transportation model to a large-scale “districting” problem. *Comput. & Opns. Res.*, 8:83–96, 1981.
- [69] Silvano Martello, David Pisinger, and Paolo Toth. New trends in exact algorithms for the 0 – 1 knapsack problem. *Europ. J. Opl. Res.*, 123:325–332, 2000.
- [70] Silvano Martello and Paolo Toth. *Knapsack problems*. Wiley, Chichester, 1990.
- [71] Anuj Mehrotra, Ellis L. Johnson, and George L. Nemhauser. An optimization based heuristic for political districting. *Management Science*, 44:1100–1114, 1998.
- [72] Giovanni Messe. Partitions of chains with minimum imbalance. *Cahiers du Centre d’Etudes de Recherche Operationelle*, 27:235–245, 1985.
- [73] Bertrand Meyer. *Eiffel: the language*. Prentice Hall, New York, 1992.
- [74] R. Minciardi, P. P. Puliafito, and R. Zoppoli. A districting procedure for social organizations. *Europ. J. Opl. Res.*, 8:47–57, 1981.
- [75] F. Murtagh. A survey of algorithms for contiguity–constrained clustering and related problems. *The Computer Journal*, 28:82–88, 1985.
- [76] Fionn D. Murtagh. Contiguity–constrained hierarchical clustering. *DIMACS Series in Discrete Mathematics and Computer Science*, 19:143–151, 1995.

- [77] Luc Muyldermans, Dirk Cattrysse, Dirk Van Oudheusden, and Tsippy Lotan. Districting for salt spreading operations. Technical Report WP99-12, Centrum voor Industrieel Beleid, Univ. Leuven, 1999. to appear in *Europ. J. Opl. Res.*
- [78] Luc Muyldermans, Dirk Cattrysse, Dirk Van Oudheusden, and Tsippy Lotan. A districting procedure for winter gritting. In *Proceedings 10th Mini EURO Conference, HCP 99 (Human Centered Processes)*, pages 61–66, 1999.
- [79] George L. Nemhauser and Laurence A. Wolsey. *Integer and Combinatorial Optimization*. Wiley, 1988.
- [80] Richard G. Niemi, Bernard Grofman, Carl Carlucci, and Thomas Hofeller. Measuring compactness and the role of a compactness standard in a test for partisan and racial gerrymandering. *The Journal of Politics*, 52:1155–1179, 1990.
- [81] Bjoern Nygreen. European assembly constituencies for Wales — comparing of methods for solving a political districting problem. *Math. Program.*, 42:159–169, 1988.
- [82] W. Ogryczak and J. Malczewski. Health care districts planning by multiobjective analysis with the mpsx/370 package. *Arch. Auto. Telemekh.*, 32:367–379, 1987.
- [83] Dirk Van Oudheusden, Dirk Cattrysse, and Tsippy Lotan. On the importance of districting and its potential impact on routing. In Hilde Meersman et al., editor, *World transport research: Proceedings 8th WCTR Conference, Antwerp, July 1998*, pages 521–531, Amsterdam, 1999. World Conference on Transport Reserach Society (WCTRS), Pergamon.
- [84] Pier Carlo Palermo, Corrado De Giorgi, and Giorgio Tagliabue. An interactive approach to the facility location districting problem. In *Adv. Oper. Res.*, pages 341–346, 1977.
- [85] Kyungchul Park and Sungsoo Park. Lifting cover inequalities for the precedence-constrained knapsack problem. *Discrete Appl. Math.*, 72:219–241, 1997.
- [86] R. Gary Parker and Ronald L. Rardin. *Discrete Optimization*. Academic Press, Boston, 1988.
- [87] Yehoshua Perl and Stephen R. Schach. Max–min tree partitioning. *J. ACM*, 28:5–15, 1981.

- [88] Yehoshua Perl and Uzi Vishkin. Efficient implementation of a shifting algorithm. *Discrete Appl. Math.*, 12:71–80, 1985.
- [89] Michael Pinedo. *Scheduling. Theory, Algorithms and Systems*. Prentice–Hall, London, 1995.
- [90] David Pisinger. Linear time algorithms for knapsack problems with bounded weights. *J. Algorithms*, 33:1–14, 1999.
- [91] Colin R. Reeves, editor. *Modern Heuristic Techniques for Combinatorial Problems*. McGraw–Hill, London, 1995.
- [92] Federica Ricca and Bruno Simeone. Political districting: Traps, criteria, algorithms and tradeoffs. *Ricerca Operativa AIRO*, 27:81–119, 1997.
- [93] David Ronen. Sales territory alignment for sparse accounts. *OMEGA The Int. J. of Mgmt. Sci.*, 11:501–505, 1983.
- [94] N. Samphaiboon and T. Yamada. Heuristic and exact algorithms for the precedence–constrained knapsack problem. *Journal of Optimization Theory and Applications*, 105:659–676, 2000.
- [95] Wolfgang Schreiber. Die Neueinteilung der Wahlkreise für die Wahl zum 14. Deutschen Bundestag — verfassungswidrig? *Zeitschrift für Rechtspolitik*, 30:105–110, 1997.
- [96] Michael Schröder. Ein exaktes Verfahren für gleichmäßige Baumzerlegungen. ANDOR Arbeitspapier, Dezember 1998.
- [97] M. Segal and D. B. Weinberger. Turfing. *Operations Research*, 25:367–386, 1977.
- [98] Roy J. Shanker, Ronald E. Turner, and Andris A. Zoltners. Sales territory design: an integrated approach. *Management Science*, 22:309–320, 1975.
- [99] Dong X. Shaw. A note on the tree partitioning problem. working paper School of Industrial Engineering, Purdue University, West Lafayette, 1996.
- [100] D.X. Shaw. A unified limited column generation approach for facility location problems on trees. *Ann. Oper. Res.*, 87:363–382, 1999.
- [101] David Simchi-Levi and Oded Berman. Optimal locations and districts of two traveling salesmen on a tree. *Networks*, 20:803–815, 1990.

- [102] Caterina De Simone, Mario Lucertini, Stefano Pallotino, and Bruno Simeone. Fair dissections of spiders, worms and caterpillars. *Networks*, 20:323–344, 1990.
- [103] Bernd Skiera. *Verkaufsgebietseinteilung zur Maximierung des Deckungsbeitrags*. Gabler, Wiesbaden, 1996.
- [104] Bernd Skiera. Wieviel Deckungsbeitrag verschenkt man durch eine gleichartige Einteilung der Verkaufsgebiete? *Zeitschrift für Betriebswirtschaftliche Forschung*, 49:723–746, 1997.
- [105] Bernd Skiera and Söhnke Albers. COSTA: Ein Entscheidungs–Unterstützungs–System zur deckungsbeitragsmaximalen Einteilung von Verkaufsgebieten. *Zeitschrift für Betriebswirtschaft*, 64:1261–1283, 1994.
- [106] R. G. Smith, L. R. Foulds, and E. G. Read. A political redistricting problem. In *Proceedings: Operational Research Society of New Zealand*, pages 37–52, 1975.
- [107] Arthur J. Swersey. The deployment of police, fire and emergency medical units. In S.M. Pollock et al., editor, *Operations Research and the Public Sector*, volume 6 of *Handbooks in Operations Research and Management Science*, pages 151–200. North–Holland, Amsterdam, 1994.
- [108] Lutz Volkmann. *Fundamente der Graphentheorie*. Springer, Wien, 1996.
- [109] Bericht der Wahlkreiskommission für die 14. Wahlperiode des Deutschen Bundestages. In: Bundesanzeiger Nr. 80a, Jg. 52, 2000.
- [110] Ergänzender Bericht der Wahlkreiskommission für die 14. Wahlperiode des Deutschen Bundestages. Bundestags–Drucksache 14/4031, 2000.
- [111] Larry Wall, Tom Christiansen, and Randal L. Schwartz. *Programming Perl*. O’Reilly, second edition, 1996.
- [112] Frank Wartenberg. *Entscheidungsunterstützung im persönlichen Verkauf*. Lang, Frankfurt, 1997.
- [113] H. P. Young. Measuring the compactness of legislative districts. *Legislative Studies Quarterly*, 13:105–115, 1988.
- [114] Andris A. Zoltners. A unified approach to sales territory alignment. In R. Bagozzi, editor, *Sales management: New Developments from Behavioral and Decision Model Research*, pages 360–376, Cambridge, MA, 1979. Marketing Science Institute.

- [115] Andris A. Zoltners and Prabhakant Sinha. Sales territory alignment: A review and model. *Management Science*, 29:1237–1256, 1983.