Gültige Ungleichungen für das Pfadweite- Problem

von

Tom Rihm

Masterarbeit in Mathematik

vorgelegt der

Fakultät für Mathematik, Informatik und Naturwissenschaften der Rheinisch- Westfälischen Technischen Hochschule Aachen

im Januar 2012

angefertigt am Lehrstuhl II für Mathematik

Prof. Dr. Ir. Arie M. C. A. Koster

RWTH AACHEN UNIVERSITY

LEHRSTUHL II FÜR MATHEMATIK

Abstract

In dieser Arbeit wird ein ganzzahliges lineares Programm aufgestellt, um die Pfadweite von Graphen zu bestimmen. Durch zusätzliche gültige Ungleichungen (oft Facetten- definierende Ungleichungen) wird versucht, möglichst viele nicht- ganzzahlige Lösungen zu verbieten, denn beim Lösen von ganzzahligen linearen Programmen spielt die Suche nach einer Beschreibung (vollständig oder teilweise) der ganzzahligen Hülle durch lineare Ungleichungen eine bedeutende Rolle.

Inhaltsverzeichnis

Abstract I

Abbildungsverzeichnis V

Tabellenverzeichnis VI

1 Einleitung 1

2 Grundlegende Definitionen und Bezeichnungen 3
- 2.1 Allgemeine Definitionen und Eigenschaften . . . 3
- 2.2 Grundlagen der ganzzahligen linearen Optimierung . . . 3
  - 2.2.1 Polyedertheorie . . . 3
  - 2.2.2 Lineare Optimierung . . . 5
  - 2.2.3 Das Schnittebenenverfahren . . . 6
  - 2.2.4 Die Chvátal- Gomory Prozedur . . . 9
- 2.3 Notationen aus der Graphentheorie . . . 11

3 Baum- und Pfadweite 13
- 3.1 Definition . . . 13
- 3.2 Eigenschaften . . . 15
- 3.3 Das Räuber- und- Gendarm- Spiel . . . 16
- 3.4 Untere und obere Schranken . . . 18

4 Das erste Modell 23
- 4.1 Anforderungen ans Modell . . . 23
- 4.2 Modellierung . . . 24
- 4.3 Ganzzahliges Lineares Programm . . . 27
  - 4.3.1 Herleitung des Ganzzahligen Linearen Programms . . . 27
  - 4.3.2 Dimension des Polyeders . . . 29

5 Gültige Ungleichungen erster Ordnung 31
- 5.1 Klasse A . . . 31
- 5.2 Klasse B . . . 39
- 5.3 Eine weitere Klasse gültiger Ungleichungen . . . 45
- 5.4 Beispiele . . . 47

6 Gültige Ungleichungen zweiter Ordnung **49**
6.1 Liften von Ungleichungen erster Ordnung zu Ungleichungen zweiter Ordnung . . . 49
6.2 Verallgemeinerung . . . 50
6.3 Eine weitere gültige Ungleichung zweiter Ordnung . . . 55
6.4 Beispiele . . . 56

7 Gültige Ungleichungen mehrfacher Ordnung **59**
7.1 Erste Ungleichung mehrfacher Ordnung . . . 59
7.2 Neue Notation . . . 61
7.3 Die Ungleichungen mehrfacher Ordnung . . . 63
7.4 Spezielle Ungleichungen . . . 68
7.5 Beispiele . . . 72

8 Das zweite Modell **75**
8.1 Die Variablen . . . 75
8.2 Das ganzzahlige lineare Programm . . . 75

9 Rechenstudie **77**
9.1 Die getesteten Graphen . . . 77
9.1.1 Die Mycielski- Graphen . . . 77
9.1.2 Die Queen- Graphen . . . 77
9.2 Ergebnisse für das erste Modell . . . 79
9.2.1 Myciel 3 . . . 79
9.2.2 Myciel 4 . . . 82
9.2.3 Myciel 5 . . . 84
9.2.4 Queen5_5 . . . 85
9.2.5 Queen6_6 . . . 87
9.2.6 Queen7_7 . . . 88
9.3 Ergebnisse für das zweite Modell . . . 90
9.4 Diskussion der Ergebnisse . . . 91

10 Schlusswort **93**

Literaturverzeichnis **VII**

Eidesstattliche Erklärung **XI**

Abbildungsverzeichnis

1 Tetraeder 5
2 Beispiel einer Schnittebene 8
3 Beispiel eines Baumes: Dieser Graph ist zusammenhängend und die Folge (d, b, a, c, g) ist ein Pfad. 12
4 Beispiel eines vollständigen Graphen: der K_5. Die Knoten a, b, c bilden eine Clique. 12
5 Der Graph G_0 14
6 Eine Baumzerlegung minimaler Weite von G_0. Es gilt: $tw(G_0) = 2$. . . 14
7 Eine Pfadzerlegung mit minimaler Weite zu dem Graphen G_0 aus Abb. 5. Es gilt: $pw(G_0) = 3$. 15
8 Das Räuber- und- Gendarm- Spiel mit Sehen 17
9 Beispiele möglicher Pfadzerlegungen für den Graphen G_1 23
10 Der Graph G_1 34
11 Der Graph G_2 47
12 Der Graph G_3 49
13 Der Graph G_4 52
14 Der Graph G_4 56
15 Der Graph G_1 62
16 Der Graph G_5 64
17 Der Graph G_5 72
18 Der Mycielski- Graph M_2 78
19 Der Mycielski- Graph M_3 78
20 Der Queen- Graph $Q_{3,3}$ 78
21 Der Queen- Graph $Q_{5,5}$ 78
22 Rechenzeit des GLP mit den Nebenbedingungen (1)-(6) für M_3 81
23 Rechenzeit des GLP mit den Nebenbedingungen (1)-(6),$(7.14)_{V,2}$ für M_3 81
24 Rechenzeit des GLP mit den Nebenbedingungen (1)-(6) für M_4 83
25 Rechenzeit des GLP mit den Nebenbedingungen (1)-(6),$(7.14)_{V,6}$ für M_4 83
26 Rechenzeit des GLP mit den Nebenbedingungen (1)-(6) für M_5 85
27 Rechenzeit des GLP mit den Nebenbedingungen (1)-(6) für $Q_{5,5}$ 85
28 Rechenzeit des GLP mit den Nebenbedingungen (1)-(6),$(7.14)_{V,2}$ für $Q_{5,5}$ 85
29 Rechenzeit des GLP mit den Nebenbedingungen (1)-(6) für $Q_{6,6}$ 88
30 Rechenzeit des GLP mit den Nebenbedingungen (1)-(6),$(7.14)_{V,4}$ für $Q_{6,6}$ 88
31 Rechenzeit des GLP mit den Nebenbedingungen (1)-(6) für $Q_{7,7}$ 88

Tabellenverzeichnis

1 Erläuterungen zur verwendeten Notation 79
2 Ganzzahlige lineare Programme für Myciel3 79
3 Lineare Relaxierung für Myciel3 80
4 Lineare Relaxierung für Myciel4 82
5 Ganzzahlige lineare Programme für Myciel4 83
6 Lineare Relaxierung für Myciel5 84
7 Ganzzahliges lineares Programm für Myciel5 84
8 Lineare Relaxierung für Queen5_5 86
9 Ganzzahlige lineare Programme für Queen5_5 86
10 Lineare Relaxierung für Queen6_6 87
11 Ganzzahlige lineare Programme für Queen6_6 87
12 Lineare Relaxierung für Queen7_7 89
13 Ganzzahlige lineare Programme für Queen7_7 89
14 Zweites Modell: ganzzahlige lineare Programme für Myciel3 90
15 Zweites Modell: ganzzahlige lineare Programme für Queen5_5 90

§ 1 Einleitung

Viele NP-schwere Probleme, wie z.B. das *Stabile- Mengen- Problem*, das *Cliquenproblem*, das *Knotenüberdeckungsproblem*, das *Färbungsproblem* oder das *Hamiltonkreisproblem* können effizient, meistens sogar in Linearzeit gelöst werden, falls der zugrunde liegende Graph ein Baum oder ein Pfad ist. Oft reicht auch eine spezielle Struktur (beschränkte Baum- oder Pfadweite) aus.

Die Begriffe der *Baumweite* und *Pfadweite* (Englisch treewidth bzw. pathwidth) wurden 1986 von N. Robertson und P. D. Seymour [31] eingeführt und beschreiben die *Ähnlichkeit* des Graphen mit einem Baum bzw. Pfad. Die Definition und wichtige Sätze sind in Kapitel 3 zu finden.

Auch in völlig anderen Bereichen finden Baum- und Pfadweite ihre Anwendungen. Zum Beispiel existieren Algorithmen, die NP- schwere Probleme aus der Numerik (Cholesky- Zerlegung) oder der Evolutionstheorie in polynomieller Zeit lösen, falls der Graph beschränkte Baum- oder Pfadweite hat [4]. Solche Algorithmen benötigen als Eingabe eine Baum- oder Pfadzerlegung mit einer möglichst geringen Weite, denn deren Laufzeit ist exponentiell in der Weite.

Das Bestimmen der Baum- und Pfadweite eines Graphen ist also auch für viele andere Probleme sehr wichtig. Es existieren bereits einige Algorithmen dafür. Allerdings sind sie in der Praxis kaum anwendbar. Desweiteren gibt es erst wenige Lösungsverfahren, die auf den Werkzeugen der linearen Programmierung basieren. Ziel dieser Arbeit ist es daher, die Pfadweite eines gegeben Graphen mit Hilfe eines ganzzahligen linearen Programms zu bestimmen und die dazugehörige Pfadzerlegung zu konstruieren.

Im Gegensatz zu linearen Programmen ist das Problem eine optimale Lösung eines ganzzahligen linearen Programms zu finden NP- schwer, da die ganzzahlige Hülle nicht bekannt ist. Aus diesem Grund ist die Suche nach einer Beschreibung der ganzzahligen Hülle durch lineare Ungleichungen beim Lösen des Problems von großer Bedeutung.

In Kapitel 4 wird das Pfadweite- Problem modelliert und das dazugehörige lineare Programm aufgestellt. Dann wird die ganzzahlige Hülle des zugehörigen Polyeders analysiert. Durch weitere gültige Ungleichungen wird versucht möglichst viele nicht- ganzzahlige Lösungen zu verbieten. Diese gültigen Ungleichungen kann man in zwei Klassen einteilen: die *Ungleichungen erster Ordnung* und die *Ungleichungen mehrfacher Ordnung*.

Kapitel 5 befasst sich mit den Ungleichungen erster Ordnung. Einige dieser Ungleichungen definieren Facetten, wie wir beweisen werden, andere jedoch nicht. Diese können wir zu Ungleichungen mehrfacher Ordnung liften. Kapitel 6 zeigt, wie Ungleichungen zweiter Ordnung von den schon bekannten Ungleichungen erster Ordnung aus dem vorigen Kapitel abgeleitet werden. Dieses Prinzip versuchen wir dann in Kapitel 7 zu verallgemeinern.

In Kapitel 9 werden das Modell und die hergeleiteten gültigen Ungleichungen an zwei Graphklassen getestet, und die Laufzeit wird mit einem zweiten Modell (Kapitel 8) verglichen. Zum Schluss werden die Ergebnisse zusammengefasst und analysiert. Kritikpunkte und mögliche Erweiterungen werden angesprochen.

§2 Grundlegende Definitionen und Bezeichnungen

Dieses Kapitel gibt einen Überblick über die für diese Arbeit relevanten Definitionen, Bezeichnungen und Notationen aus der ganzzahligen linearen Optimierung und der Graphentheorie und macht den Leser so mit der verwendeten Notation vertraut.

2.1 Allgemeine Definitionen und Eigenschaften

Seien A und B zwei beliebige Mengen. Wir schreiben $A \cup B$ für die Vereinigungsmenge und $A \cap B$ für die Schnittmenge. Falls A eine Teilmenge von B ist, so notieren wir $A \subseteq B$, bzw. $A \subset B$, falls A eine strikte Teilmenge ist. Die mengentheoretische Differenz ist

$$A \backslash B := \{x \mid x \in A \text{ und } x \notin B\}.$$

Falls $\mathbb{K}$ eine der Mengen $\mathbb{Z}$, $\mathbb{Q}$, $\mathbb{R}$ ist, dann ist $\mathbb{K}_{\geq 0} := \{x \in \mathbb{K} : \ x \geq 0\}$ und für die natürlichen Zahlen mit Null verwenden wir das Symbol $\mathbb{N}_0 := \mathbb{N} \cup \{0\}$. Die Ab- und Aufrundungsfunktionen für $a \in \mathbb{R}$ sind folgendermaßen definiert:

$$\lfloor a \rfloor := \max_{k \in \mathbb{Z}, k \leq a} k \quad \text{und} \quad \lceil a \rceil := \min_{k \in \mathbb{Z}, k \geq a} k.$$

2.2 Grundlagen der ganzzahligen linearen Optimierung

Die im Folgenden eingeführten Definitionen stammen größtenteils aus der Vorlesung Ganzzahlige Lineare Optimierung von Prof. Dr. Ir. Arie M. C. A. Koster aus dem Sommersemester 2009 [22] an der RWTH Aachen.

2.2.1 Polyedertheorie

Ein *Polyeder* P ist eine Teilmenge von $\mathbb{R}^n$, die durch endlich viele lineare Ungleichungen beschrieben wird:

$$P := \{x \in \mathbb{R}^n : \ Ax \leq b\}, \quad \text{wobei } A \in \mathbb{R}^{m \times n}, \ b \in \mathbb{R}^m.$$

Ein Polyeder, das beschränkt ist, heißt auch *Polytop*. Für eine Teilmenge $X = \{x_1, \ldots, x_k\} \subseteq \mathbb{R}^n$ bezeichnet man mit

$$conv(X) := \left\{ \sum_{i=1}^{k} \lambda_i \cdot x_i \;\middle|\; \lambda_i \in \mathbb{R}_{\geq 0} \text{ für alle } i \in \{1, \ldots, k\} \text{ und } \sum_{i=1}^{k} \lambda_i = 1 \right\}$$

die konvexe Hülle von X. Dies ist die kleinste konvexe Menge, die X enthält.

Eine Menge $X \subseteq \mathbb{Z}^n \times \mathbb{R}^p$ wird durch ein Polyeder $P \subseteq \mathbb{R}^{n+p}$ beschrieben, falls $X = P \cap (\mathbb{Z}^n \times \mathbb{R}^p)$. Dann heißt P eine *Formulierung* für X. Eine Formulierung P_1 ist besser als P_2, falls $P_1 \subset P_2$. Es existiert keine bessere Formulierung für X als die konvexe Hülle $conv(X)$.

Die Vektoren $x_1, \ldots, x_k \in \mathbb{R}^n$ heißen *affin unabhängig*, wenn aus

$$\sum_{i=1}^{k} \lambda_i \cdot x_i = 0 \text{ und } \sum_{i=1}^{k} \lambda_i = 0$$

stets folgt, dass $\lambda_1 = \lambda_2 = \ldots = \lambda_k = 0$ ist.

(2.1) Lemma
Folgende Aussagen sind äquivalent:

- Die Vektoren $x_1, \ldots, x_k \in \mathbb{R}^n$ sind affin unabhängig.
- Die Vektoren $(x_2 - x_1), \ldots, (x_k - x_1) \in \mathbb{R}^n$ sind linear unabhängig. ◇

Die *Dimension* $\dim(P)$ eines Polyeders $P \subseteq \mathbb{R}^n$ ist eins weniger als die maximale Anzahl affin unabhängiger Vektoren in P.

Eine Ungleichung $\pi \cdot x \leq \pi_0$ mit $\pi \in \mathbb{R}^n$ und $\pi_0 \in \mathbb{R}$ heißt *gültige Ungleichung* für X, falls $\pi \cdot y \leq \pi_0$ für alle $y \in X$ gilt.

Seien $\pi \cdot x \leq \pi_0$ und $\mu \cdot x \leq \mu_0$ zwei gültige Ungleichungen. Die Ungleichung $\pi \cdot x \leq \pi_0$ dominiert die Ungleichung $\mu \cdot x \leq \mu_0$, wenn ein $\lambda > 0$ existiert, so dass $\lambda \cdot \mu \leq \pi$ und $\pi_0 \leq \lambda \cdot \mu_0$ und $(\pi, \pi_0) \neq (\lambda \cdot \mu, \lambda \cdot \mu_0)$ gilt.

Sei $P \subseteq \mathbb{R}^n$ ein Polyeder. Eine Menge $S \subseteq P$ ist eine *Seitenfläche* von P, wenn eine gültige Ungleichung $\pi \cdot x \leq \pi_0$ existiert, für die gilt:

$$S = \{x \in P : \ \pi \cdot x = \pi_0\}$$

S heißt *nicht- triviale* bzw. *echte* Seitenfläche, falls $S \neq \emptyset$ bzw. $S \neq P$. Eine Menge F ist eine *Facette* von P, genau dann wenn F eine nicht- triviale echte Seitenfläche von P ist und keine strikte Teilmenge einer anderen echten Seitenfläche ist. Die Dimension einer Facette ist genau eins weniger als die Dimension des Polyeders:

$$\dim(F) = \dim(P) - 1.$$

(2.2) Beispiel
Die Seitenflächen eines dreidimensionalen Tetraeders (siehe Abbildung 1) sind:

- die leere Menge,
- die 4 Ecken,
- die 6 Kanten,
- die 4 dreieckigen zweidimensionalen Flächen
- und das ganze Tetraeder.

Die Ecken und die Kanten sind in den Flächen des Tetraeders strikt enthalten. Also sind nur die dreieckigen Flächen Facetten des Tetraeders. ◇

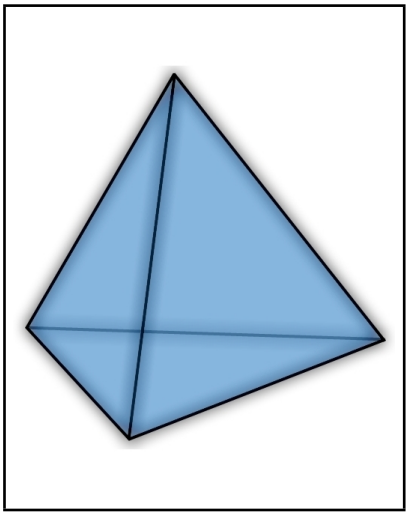

Abbildung 1: Tetraeder

2.2.2 Lineare Optimierung

Mit Hilfe der linearen Optimierung können wir viele Probleme aus der Realität (Tourenplanung, Frequenzzuweisung, u.s.w.) modellieren und dank guter Lösungsverfahren (Branch- and- Bound, Simplex, Schnittebenenverfahren, u.s.w.) effizient lösen. Das Schnittebenenverfahren wird später in Kapitel 2.2.3 noch genauer betrachtet.

Von einem linearen Optimierungsproblem sprechen wir, falls wir eine lineare Zielfunktion unter linearen Nebenbedingungen minimieren bzw. maximieren möchten. Seien $A \in \mathbb{R}^{m \times n}$, $b \in \mathbb{R}^m$ und $c \in \mathbb{R}^n$. Dann ist

$$\begin{aligned} \min \quad & c^T \cdot x & \\ \text{s.t.} \quad & A \cdot x & \leq b \\ & x & \geq 0 \end{aligned}$$

ein lineares Programm (LP). Ein ganzzahliges lineares Programm (GLP) hat die gleiche Form, nur dass zusätzlich die Ganzzahligkeit der Variablen gefordert wird.

Ein Punkt $x \in \mathbb{R}^n$, der alle Nebenbedingungen erfüllt, heißt *zulässige Lösung* des LP. Eine zulässige Lösung x^* ist *optimale Lösung*, falls kein zulässiges x mit kleinerem Zielfunktionswert existiert. Die Menge der zulässigen Lösungen eines LP

$$P_{LP} := \{x \in \mathbb{R}^n \mid A \cdot x \leq b,\ x \geq 0\}$$

ist ein konvexes Polyeder.

Eine optimale Lösung, falls überhaupt eine optimale Lösung existiert, ist stets eine Ecke des Polyeders P_{LP}. Diese kann durch das Simplex- Verfahren, welches 1951 von G. B. Dantzig [13] entwickelt wurde, gefunden werden. Um die optimale Lösung eines linearen Programms zu finden, können wir uns also auf die Ecken des Polyeders beschränken.

Als lineare Relaxierung eines GLP bezeichnen wir das lineare Programm, das wir durch Weglassen der Ganzzahligkeitsbedingung erhalten. Wir notieren P_{LP} für die Menge der zulässigen Lösungen der linearen Relaxierung und

$$P_{GLP} := conv(P_{LP} \cap \mathbb{Z}^n)$$

für die ganzzahlige Hülle. Das ist die konvexe Hülle der zulässigen ganzzahligen Lösungen des ganzzahligen linearen Programms. P_{GLP} ist wieder ein Polyeder. Die benötigte Anzahl von Ungleichungen, um die ganzzahlige Hülle vollständig zu beschreiben, kann allerdings exponentiell groß sein, und die einzelnen Ungleichungen sind oft nicht bekannt. Würde es uns gelingen die ganzzahlige Hülle vollständig zu beschreiben, würden wir mit dem Simplexalgorithmus eine optimale ganzzahlige Lösung erhalten. Es gilt:

- Eine optimale Lösung der linearen Relaxierung ist stets eine duale Schranke für das Originalproblem.
- Ist die optimale Lösung der linearen Relaxierung ganzzahlig, so ist sie auch optimal für das Originalproblem.
- Falls die lineare Relaxierung keine zulässige Lösung hat, so hat auch das Originalproblem keine zulässige Lösung.

Diese lineare Relaxierung spielt eine tragende Rolle beim nun folgenden Schnittebenenverfahren.

2.2.3 Das Schnittebenenverfahren

Bei den Lösungsverfahren unterscheidet man zwischen exakten Verfahren und heuristischen Verfahren. Die exakten Verfahren sind nach [15] unterteilbar in:

- Entscheidungsbaumverfahren: z.B. das Branch- and- Bound Verfahren von A. H. Land, A. G. Doig [23] und von R. J. Dakin [12]
- Schnittebenenverfahren

- Kombinationen aus den ersten beiden, wie zum Beispiel das Branch- and- Cut Verfahren von M. Padberg und G. Rinaldi [26].

Bei größeren Graphen stößt das Branch- and- Bound Verfahren schnell an seine Grenzen.

Das Schnittebenenverfahren wurde von G. Dantzig, R. Fulkerson und S. Johnson im Jahr 1954 [14] eingeführt, um das Problem des Handlungsreisenden (TSP: Traveling Salesman Problem) zu lösen. Beim TSP werden die Kurzzyklusungleichungen (subtour elimination constraints) benötigt. Die Anzahl dieser Ungleichungen beträgt $2^n - 2 \cdot (n-1)$ und wächst exponentiell mit der Anzahl der Städte. Eine vollständige Beschreibung des Polytops durch Ungleichungen ist daher sehr aufwendig und bei größeren Graphen nicht durchführbar. Die Idee des Schnittebenenverfahrens ist es daher, solche großen Mengen an Ungleichungen nicht auf einmal ins LP zu laden, sondern nur die, die gebraucht werden und erst dann, wenn sie wirklich gebraucht werden.

Das Schnittebenenverfahren: Gegeben sei ein GLP.

1. Löse die LP-Relaxierung des Problems. Sei x^* die optimale Lösung.
2. a Falls $x^* \in \mathbb{Z}^n$: STOP. Dann ist x^* eine optimale Lösung des ganzzahligen Problems.

   b Sonst: Finde eine Ungleichung, die gültig ist für das GLP und von x^* verletzt wird, oder einen Beweis, dass so eine nicht existiert. Füge diese Ungleichung zur LP- Relaxierung hinzu und gehe zu (1).

Die in 2.b zu lösende Aufgabe heißt *Separationsproblem*.

Zuerst wird eine optimale Lösung der LP-Relaxierung berechnet. Dann werden weitere Ungleichungen, sogenannte Schnittebenen (engl. cutting planes) dem LP hinzugefügt. Diese Ungleichungen werden von allen zulässigen Punkten des GLP erfüllt, nicht aber von der aktuellen LP- Lösung. Die besten Schnittebenen, die man so finden kann, sind die Facetten des GLP- Polyeders. Beim erneuten Lösen kommt daher eine neue Lösung heraus. Dieses Verfahren wird solange durchgeführt, bis eine ganzzahlige Lösung und somit optimale Lösung des GLP gefunden wird, oder ein Beweis erbracht wird, dass eine solche Lösung nicht existiert.

Ein gutes Beispiel zeigt Abbildung 2. Hier sind die Lösungsmenge P_{LP} und die optimale Lösung x^*_{LP} der linearen Relaxierung, sowie die ganzzahlige Hülle P_{GLP} und die optimale Lösung x^*_{GLP} des ganzzahligen linearen Programms abgebildet. Die

Schnittebene schneidet x^*_{LP} vom Polyeder ab, ohne dabei eine zulässige ganzzahlige Lösung mit abzuschneiden.

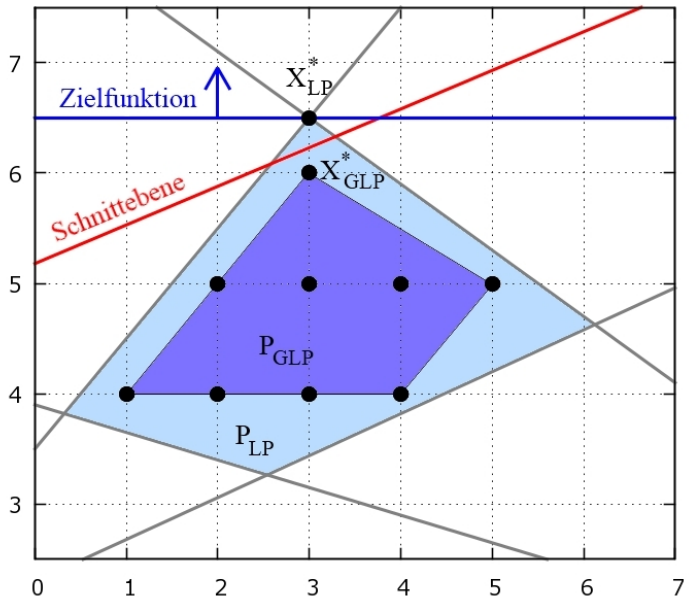

Abbildung 2: Beispiel einer Schnittebene

(2.3) Satz (Grötschel, Lovasz, Schrijver [17])
Das Optimierungsproblem ist in polynomieller Zeit lösbar, genau dann, wenn das zugehörige Separationsproblem in polynomieller Zeit lösbar ist. ◇

Allerdings ist eine vollständige Beschreibung der ganzzahligen Hülle nur für wenige Probleme bekannt, d.h. oft sind nicht alle Facetten des Problems bekannt, und am Ende des Schnittebenenverfahrens ist die optimale LP- Lösung noch immer fraktional. Deswegen wird in der Praxis auch häufig eine Kombination vom Branch- and-Bound und dem Schnittebenenverfahren angewendet: das sogenannte Branch- and-Cut Verfahren [26] von M. Padberg und G. Rinaldi. Hier werden die einzelnen Teilprobleme mit dem Schnittebenenverfahren gelöst. Ist die Lösung noch immer fraktional und werden keine Ungleichungen mehr gefunden, so teilt man das Problem wie im Branch- and- Bound Verfahren auf (Branching).

Bei den Schnittebenen unterscheidet man zwischen *problemspezifischen Ungleichungen* und *allgemeinen Ungleichungen*.

Die *problemspezifischen Ungleichungen* gelten nur für ein bestimmtes Optimierungsproblem, denn sie basieren auf bestimmten kombinatorischen Strukturen. Hier kann man oft beweisen, dass sie Facetten definieren und so in der Praxis sehr effizient sind. Zum Beispiel die ungeraden Kreis- Ungleichungen beim Stabilen- Mengen-Problem, die Kurzzyklusungleichungen beim TSP oder die ungeraden Mengen- Ungleichungen beim Matching- Problem. Solche problemspezifischen Ungleichungen werden im weiteren Verlauf dieser Arbeit für das Pfadweite Problem bestimmt.

Der Vorteil von *allgemeinen Ungleichungen* ist, dass sie völlig unabhängig von der Struktur des Problems für eine Vielzahl von sehr unterschiedlichen Optimierungsproblemen gelten. Viele dieser Ungleichungen haben aber nicht die Facetten- Eigenschaft und daher konvergiert das Schnittebenenverfahren zu langsam. Eine Methode solche Ungleichungen zu bestimmen, ist die nun folgende Chvátal- Gomory Prozedur.

2.2.4 Die Chvátal- Gomory Prozedur

Das Bestimmen der ganzzahligen Hülle konvexer Polyeder ist von großer Bedeutung, denn falls die ganzzahlige Hülle bekannt ist, kann das Lösen von ganzzahligen linearen Optimierungsproblemen auf lineare Programme reduziert werden.

Eine der bekanntesten Methoden zur Erzeugung gültiger Ungleichungen ist die Chvátal- Gomory Prozedur [16]:

Gegeben sei ein Polyeder $P = \{x \in \mathbb{R}^n : \ Ax \leq b, x \geq 0\}$, wobei A eine $m \times n$ Matrix mit Spalten $a_1, a_2, \ldots, a_n$ ist. Weiter sei $X = P \cap \mathbb{Z}^n$.

Die Bestimmung einer gültigen Ungleichung erfolgt mit der C-G- Prozedur in drei Schritten:

Chvátal- Gomory Prozedur:

- Schritt 1: Wähle $u \in \mathbb{R}^m_{\geq 0}$. Die Ungleichung

$$\sum_{j=1}^{n} \left((u^T a_j) \cdot x_j \right) \leq u^T b$$

ist gültig für P, da $u \geq 0$ und $\sum_{j=1}^{n} \ a_j x_j \leq b$.

- Schritt 2: Die Ungleichung

$$\sum_{j=1}^{n} \left(\left\lfloor u^T a_j \right\rfloor \cdot x_j \right) \leq u^T b$$

ist gültig für P, da $x \geq 0$.

- Schritt 3: Die Ungleichung

$$\sum_{j=1}^{n} \left(\left\lfloor u^T a_j \right\rfloor \cdot x_j \right) \leq \left\lfloor u^T b \right\rfloor$$

heißt Chvátal- Gomory- Schnitt und ist gültig für X, da $x \in \mathbb{Z}^n$.

Die Chvátal- Gomory Prozedur nutzt die Ganzzahligkeitsbedingung der Variablen explizit aus.

Wir definieren

$$P^{(0)} := P_{LP},$$

$$P^{(1)} := P_{LP} \cap \bigcap_{u \geq 0} \left\{ x \mid \left\lfloor u^T \cdot A \right\rfloor \cdot x \leq \left\lfloor u^T \cdot b \right\rfloor \right\}$$

$$P^{(k+1)} := \left(P^{(k)} \right)^{(1)}.$$

Wir nennen $P^{(k)}$ den k- ten *Chvátal-Abschluss* von P_{LP}.

(2.4) Satz (V. Chvátal [11] und A. Schrijver [33])
Sei P ein Polyeder. Dann kann jede gültige Ungleichung für $X = P \cap \mathbb{Z}^n$ durch endlich viele Wiederholungen der C-G- Prozedur erzeugt werden. D.h. es existiert ein $t \in \mathbb{N}_0$ mit

$$P_{GLP} = P^{(t)} \subseteq P^{(t-1)} \subseteq \ldots \subseteq P^{(1)} \subseteq P^{(0)} = P_{LP}. \qquad \diamond$$

Der *Chvátal- Rank* von P ist definiert als die kleinste ganze Zahl $t \in \mathbb{N}_0$, für die $P^{(t)} = P_{GLP}$ gilt.

2.3 Notationen aus der Graphentheorie

In dieser Ausarbeitung bezeichnet $G = (V, E)$ einen endlichen und einfachen (d.h. ungerichtet und ohne Mehrfachkanten und Schleifen) Graphen, mit $n := |V|$ Knoten und $m := |E|$ Kanten. Da die Kanten nicht gerichtet sind, entspricht eine Kante einer 2- elementigen Teilmenge von V. Für eine Teilmenge $I \subseteq V$ der Knoten, ist

$$G[I] = \Big(I, \big\{\{v, w\} \in E \;\big|\; v, w \in I\big\}\Big)$$

der von I induzierte Teilgraph von G, d.h. $G[I]$ besteht aus allen Knoten aus I und allen Kanten, die in G zwischen ihnen verlaufen. Wir kürzen $G[V \backslash I]$ oft mit $G \backslash I$ ab.

Zwei Knoten $v, w \in V$ heißen *adjazent* oder *benachbart*, falls $\{v, w\} \in E$, also wenn sie durch eine Kante verbunden sind. Ein Knoten $v \in V$ und eine Kante $e \in E$ heißen *inzident* genau dann, wenn $v \in e$. Die *Nachbarschaft* eines Knotens ist die Menge aller seiner Nachbarn, also

$$N(v) = \{w \in V | \{v, w\} \in E\}.$$

Die abgeschlossene Nachbarschaft ist $N[v] = N(v) \cup \{v\}$.

Der *Grad* eines Knotens $v \in V$ ist die Anzahl seiner Nachbarn und wir kürzen ihn mit $d_G(v)$ ab, bzw. $d(v)$, falls klar ist, um welchen Graphen es sich handelt. Der *Minimalgrad* $\delta(G)$ und der *Maximalgrad* $\Delta(G)$ definieren wir wie folgt:

$$\delta(G) := \min_{v \in V} d(v) \qquad \text{und} \qquad \Delta(G) := \max_{v \in V} d(v).$$

Ein *Weg* der Länge k ist eine Folge von Knoten $(v_1, v_2, \ldots, v_k)$ mit $\{v_i, v_{i+1}\} \in E$ für alle $i = 1, \ldots, k-1$. Sind zusätzlich alle Knoten paarweise verschieden, so spricht man von einem *Pfad*. Enthält der Pfad alle Knoten von G, so nennt man ihn auch einen *Hamiltonpfad*. Ein *Kreis* ist ein geschlossener Weg, d.h. $v_1 = v_k$, mit mindestens 3 (verschiedenen) Knoten.

Ein Graph ist genau dann *zusammenhängend*, wenn zwischen je zwei Knoten mindestens ein Weg existiert. Eine *Zusammenhangskomponente* des Graphen ist eine maximale zusammenhängende Teilmenge seiner Knoten.

Ein *Baum* ist ein kreisfreier zusammenhängender Graph. Ein *Unterbaum* ist ein Teilgraph eines Baumes, der selbst wieder ein Baum ist.

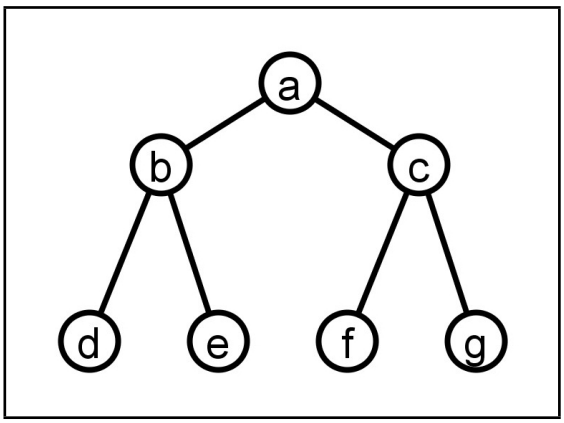

Abbildung 3: Beispiel eines Baumes: Dieser Graph ist zusammenhängend und die Folge (d, b, a, c, g) ist ein Pfad.

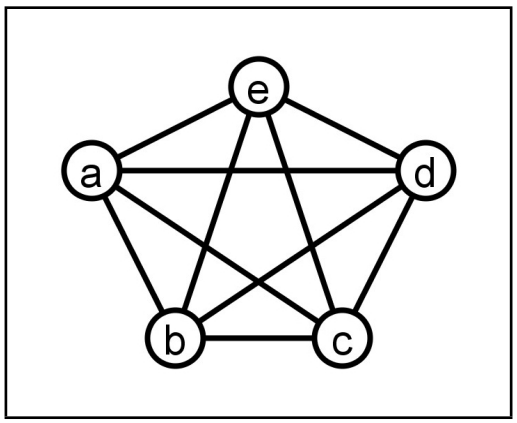

Abbildung 4: Beispiel eines vollständigen Graphen: der K_5. Die Knoten a, b, c bilden eine Clique.

Ein Graph heißt *vollständig*, falls je zwei Knoten durch eine Kante miteinander verbunden sind. Eine *Clique* ist eine Teilmenge der Knoten $Q \subseteq V$, wobei $G[Q]$ vollständig ist. Die *Cliquenzahl* $\omega(G)$ ist die Anzahl der Knoten einer größten Clique von G.

§3 Baum- und Pfadweite

Dieses Kapitel gibt eine Einführung in die Theorie der Baum- und Pfadweite. Dazu gehören zum einen die formale Definition, zum anderen aber auch wichtige Sätze und ein Überblick über existierende Algorithmen und Approximationen. Eine spielerische Anwendung wie das *Räuber- und- Gendarm- Spiel* darf natürlich auch nicht fehlen.

3.1 Definition

Der Begriff der *Baumweite* (Englisch treewidth) wurde von N. Robertson und P. D. Seymour [31] eingeführt und besagt, wie groß die *Ähnlichkeit* des Graphen mit einem Baum ist, d.h. wie viele Knoten wir maximal zu einem Knoten verschmelzen müssen, damit der Graph wie ein Baum aussieht.

(3.1) Definition (Baumweite)
Sei $G = (V, E)$ ein einfacher ungerichteter Graph. Eine *Baumzerlegung* von G ist ein Paar (X, T), wobei $T = (I, F)$ ein Baum ist und $X = \{X_i | i \in I\}$ eine Familie von Untermengen von V (diese Untermengen werden als Taschen bezeichnet, englisch: bags), so dass

1. $\bigcup_{i \in I} X_i = V$, d.h. jeder Knoten aus V ist in mindestens einer Tasche.
2. Für alle Kanten $\{v, w\} \in E$ existiert ein Index $i \in I$ mit $v, w \in X_i$, d.h. die beiden Endknoten jeder Kante sind zusammen in mindestens einer Tasche.
3. Für alle Indizes $i, j, k \in I$ gilt: Falls j auf dem Pfad in T von i zu k liegt, dann ist $X_i \cap X_k \subseteq X_j$, d.h. für jeden Knoten $v \in V$ bilden die Mengen $I_v = \{i \in I | v \in X_i\}$ einen zusammenhängenden Unterbaum von T.

Die *Weite* der Baumzerlegung (X, T) von G ist die Kardinalität der größten Tasche minus 1, also die Zahl $max_{i \in I} |X_i| - 1$. Die *Baumweite* von G ist die kleinste Weite aller möglichen Baumzerlegungen und wird $tw(G)$ abgekürzt. ◇

Die Subtraktion von 1 dient der Normierung; so haben Bäume Baumweite 1. Je größer die Baumweite ist, umso weniger ähnelt der Graph einem Baum. Ein Graph hat genau dann Baumweite 0, wenn er keine Kanten enthält. Jeder Graph hat die

triviale Baumzerlegung (X, T), wobei $X = \{V\}$ und T der Baum, bestehend aus einem einzigen Knoten, ist.

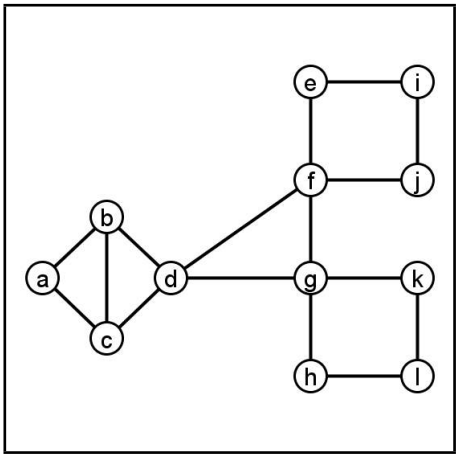

Abbildung 5: Der Graph G_0

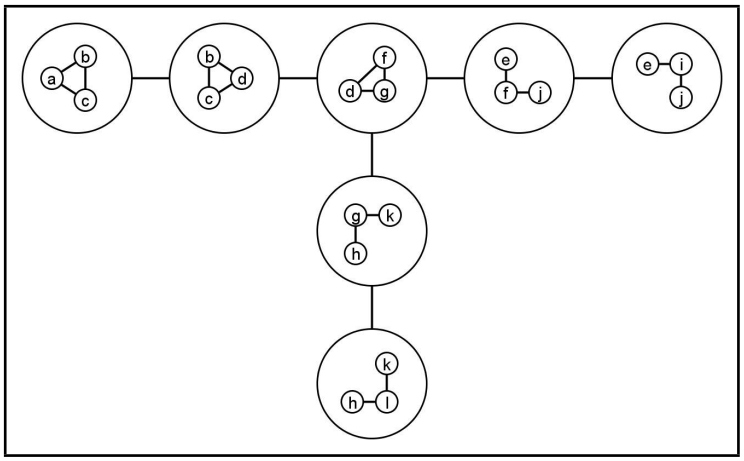

Abbildung 6: Eine Baumzerlegung minimaler Weite von G_0. Es gilt: $tw(G_0) = 2$.

Falls T in Definition (3.1) ein Pfad ist, dann sprechen wir von *Pfadzerlegung* bzw. *Pfadweite*: auch diese Definition stammt von N. Robertson und P. D. Seymour [31]:

(3.2) Definition (Pfadweite)
Sei $G = (V, E)$ ein einfacher ungerichteter Graph. Eine *Pfadzerlegung* von G ist eine Folge von Knotenteilmengen $(X_1, X_2, \ldots, X_r)$, so dass

1. $\bigcup_{1 \leq i \leq r} X_i = V$,
2. Für jede Kante $\{v, w\} \in E$, existiert ein Index $i \in \{1, \ldots, r\}$ mit $v, w \in X_i$,
3. Für alle Indizes $i, j, k \in \{1, \ldots, r\}$ gilt: falls $i \leq j \leq k$, dann ist $X_i \cap X_k \subseteq X_j$.

Die *Weite* der Pfadzerlegung $(X_1, X_2, \ldots, X_r)$ von G ist die Zahl $max_{1 \leq i \leq r} |X_i| - 1$. Die *Pfadweite* von G ist die kleinste Weite aller möglichen Pfadzerlegungen und wird $pw(G)$ abgekürzt. ◇

Ein Pfad hat Pfadweite 1. Je größer die Pfadweite, umso weniger ähnelt der Graph einem Pfad.

Es existieren viele äquivalente oder ähnliche Formulierungen zur Baumweite. So zeigte zum Beispiel D. J. Rose [32] die Äquivalenz zwischen der Klasse der partiellen

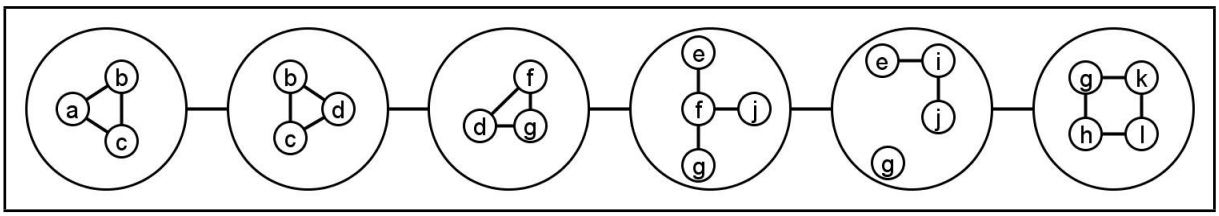

Abbildung 7: Eine Pfadzerlegung mit minimaler Weite zu dem Graphen G_0 aus Abb. 5. Es gilt: $pw(G_0) = 3$.

k- Bäume und der Klasse der Graphen mit einer Baumweite von höchstens k. Eine Übersicht über verwandte Begriffe ist in H. L. Bodlaender [6] zu finden.

3.2 Eigenschaften

(3.3) Satz (Siehe [18] Satz 10.19 und 10.20)

1. Sei $G = (V, E)$ ein Graph und $G' = (V', E')$ ein Teilgraph von G. Dann gilt: $tw(G') \leq tw(G)$ und $pw(G') \leq pw(G)$.
2. Die Baumweite eines Graphen $G = (V, E)$ ist gleich dem Maximum der Baumweiten der Zusammenhangskomponenten von G.
3. Die Pfadweite eines Graphen $G = (V, E)$ ist gleich dem Maximum der Pfadweiten der Zusammenhangskomponenten von G. ◇

Beweis

1. Entfernt man alle Knoten aus $V \backslash V'$ aus der Baumzerlegung (bzw. Pfadzerlegung) von G, so erhält man eine Baumzerlegung (bzw. Pfadzerlegung) für G'.
2. Sei $G = (V, E)$ ein Graph mit k Zusammenhangskomponenten $G_1, \ldots, G_k$. Wegen Teil 1 gilt:
$$tw(G) \geq \max_{i=1,\ldots,k} tw(G_i).$$
Verbindet man die k Baumzerlegungen zu einem Baum, so entsteht eine Baumzerlegungen für G. Somit gilt auch:
$$tw(G) \leq \max_{i=1,\ldots,k} tw(G_i).$$
Insgesamt folgt also die Behauptung.

3. Sei $G = (V, E)$ ein Graph mit k Zusammenhangskomponenten $G_1, \ldots, G_k$. Wegen Teil 1 gilt:
$$pw(G) \geq \max_{i=1,\ldots,k} pw(G_i).$$
Verkettet man die k Pfadzerlegungen zu einem Pfad, so entsteht eine Pfadzerlegung für G. Somit gilt auch:
$$pw(G) \leq \max_{i=1,\ldots,k} pw(G_i).$$
Insgesamt folgt also die Behauptung. □

Also können wir uns im Folgenden auf einfache und zusammenhängende Graphen beschränken.

(3.4) Satz (Arnborg, Corneil, Proskurowski[3]. Bodlaender [5])
a) Das Problem, die Baum- bzw. Pfadweite eines Graphen zu bestimmen, ist NP-vollständig.

b) Für alle $k \in \mathbb{N}$ kann man in Linearzeit feststellen, ob ein Graph höchstens Baum- bzw. Pfadweite k hat und falls ja, diese auch bestimmen. ◇

3.3 Das Räuber- und- Gendarm- Spiel

Als Einführung in die Theorie der Baum- und Pfadweite dient in der Literatur oft eine spielerische Herangehensweise: *das Räuber- und- Gendarm- Spiel* von A. LaPaugh [24] bzw. T. D. Parsons [27]. Wie der Name es andeutet, wird ein Räuber von mehreren Gendarmen auf einem Spielfeld gejagt. Als Spielfeld dient ein Graph $G = (V, E)$. Der Räuber und die Gendarmen bewegen sich von Knoten zu Knoten mit zwei Unterschieden:

- Der Räuber kann sich zu jedem Zeitpunkt beliebig schnell von einem Knoten zu einem anderen bewegen, solange ein Pfad ohne Gendarm zwischen den beiden Knoten existiert.
- Die Gendarmen bewegen sich auch von Knoten zu Knoten, allerdings müssen diese Knoten nicht miteinander verbunden sein, und der Gendarm benötigt eine bestimmte Zeit dafür. Anschaulich betrachtet fliegen die Gendarmen mit dem Helikopter von Knoten zu Knoten. Während des Flugs sind weder Anfangs- noch Endknoten besetzt, und der Räuber könnte den so entstandenen Weg nutzen.

Ziel des Spiels ist es, den Räuber mit so wenigen Gendarmen wie möglich einzufangen, wobei der Räuber nur auf Knoten verhaftet werden kann. Dabei gibt es zwei Varianten:

- **Mit Sehen:** (Siehe dazu P. D. Seymour und R. Thomas [34] und L. M. Kirousis und C. H. Papadimitriou[19]) Die Gendarmen wissen (dank guter Überwachungskameras) immer wo, d.h. in welchem Knoten des Graphen der Räuber sich gerade aufhält und können ihre Strategie danach ausrichten. Dabei gelingt es nicht den Räuber zu schnappen, wenn einfach ein Gendarm zu dem Knoten fliegt, wo der Räuber sich gerade aufhält. Denn der Räuber ist schneller und kann in der Zwischenzeit zu einem anderen Knoten fliehen. Vorausgesetzt natürlich es existiert ein Fluchtweg, der nicht von einem Gendarm zugesetzt ist. Die minimale Anzahl an Gendarmen die benötigt werden, um den Räuber einzufangen, ist genau $tw(G) + 1$.
- **Ohne Sehen bzw. unsichtbar:** (Siehe dazu A. LaPaugh [24]) Die Gendarmen haben keine Information darüber, wo der Räuber sich gerade aufhält. Sie müssen den Graphen von einer Seite zur anderen durchsuchen, um den Räuber zu finden. Dabei müssen sie aufpassen, dass der Räuber zu keinem Zeitpunkt in den bereits durchsuchten Teilgraph fliehen kann. Die minimale Anzahl an Gendarmen die benötigt werden, um den Räuber einzufangen, ist genau $pw(G) + 1$.

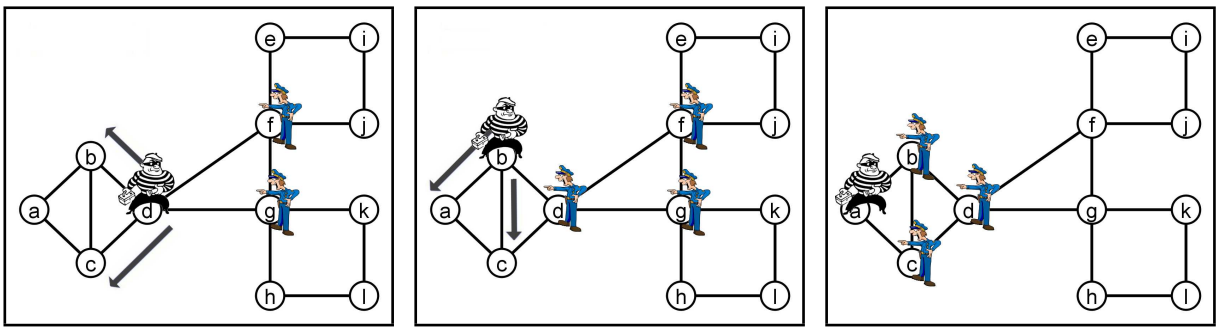

Abbildung 8: Das Räuber- und- Gendarm- Spiel mit Sehen

(3.5) Beispiel
Als Beispiel betrachten wir die beide Varianten des Spiels auf dem Graph aus Abbildung 5:

- **Mit Sehen:** In Abbildung 8 sind die drei wichtigsten Phasen dargestellt, um den Räuber zu schnappen: Der Räuber startet auf Knoten d. In diesem Fall setzen wir zwei Gendarmen auf die Knoten f und g. Somit kann der Räuber,

falls wir einen zusätzlichen Gendarm auf den Knoten d setzen, nur noch auf die Knoten a,b und c flüchten. Wir mussten einen zusätzlichen Gendarm hinzuziehen, denn hätte einer der beiden Gendarmen aus f oder g seinen Knoten verlassen, um zu Knoten d zu fliegen, hätte der Räuber in der Zwischenzeit zu dem leer gewordenen Knoten fliehen können. Es gelingt den Gendarmen schließlich zu dritt den Räuber zu fangen. Denn im dritten Bild aus Abb. 8 muss nur noch der Gendarm aus Knoten d zu a gehen, um den Räuber zu verhaften. Der Räuber kann nicht fliehen, denn alle seine Fluchtwege sind zugestellt. Somit ist die Baumweite des angegebenen Graphen 2.

- **Ohne Sehen:** Auf demselben Graphen bräuchten wir 4 Gendarmen, um den Räuber beim Spiel ohne Sehen einzufangen. Denn hier können die Gendarmen nicht einfach in der Mitte des Graphen anfangen und dann schauen, wo der Räuber ist. Sie müssen auf einer Seite anfangen und dann den Graphen Knoten für Knoten so absuchen, dass der Räuber nie mehr auf bereits abgesuchte Knoten flüchten kann. Also ist die Pfadweite des Graphen 3. ◇

Mit Hilfe des Beispiels wurde der Zusammenhang zur Baum- bzw. Pfadzerlegung deutlich:

- **Mit Sehen:** In einer optimalen Strategie besetzen wir in jedem Schritt genau die Knoten des Graphen, die auch in der entsprechenden Tasche einer Baumzerlegung minimaler Weite sind, mit einem Gendarm. Dann arbeiten wir uns zu einer Blatttasche des Baumes der Zerlegung vor, je nachdem wohin der Räuber flüchtet.
- **Ohne Sehen:** In einer Pfadzerlegung minimaler Weite entsprechen die Knoten aus Tasche i genau den Knoten, auf denen, in einer optimalen Strategie, im i-ten Schritt ein Gendarm steht.

3.4 Untere und obere Schranken

Dieses Kapitel gibt einen kleinen Überblick über bekannte untere und obere Schranken für die Baum- und Pfadweite. Solche Abschätzungen sind oft sehr hilfreich bei der Bestimmung der Baum- bzw. Pfadweite eines Graphen. Zuerst sind ein paar untere Schranken für die Baumweite aufgelistet. Da jede Pfadzerlegung auch eine Baumzerlegung ist, gelten diese unteren Schranken auch für die Pfadweite:

$$tw(G) \leq pw(G) \leq |V| - 1.$$

Man kann leicht erkennen, dass jede Clique in mindestens einer Tasche sein muss.

(3.6) Satz (H. L. Bodlaender, R. H. Möhring [9])
Gegeben sei ein Graph $G = (V, E)$, eine Clique $Q \subseteq V$ und eine Baumzerlegung $(\{X_i | \ i \in I\}, \ T = (I, F))$. Dann existiert ein $i \in I$ mit $Q \subseteq X_i$. ◇

Beweis
Dieser Satz kann sehr schnell mit der *Helly- Eigenschaft* bewiesen werden. Siehe dazu H. L. Bodlaender und R. H. Möhring [9]. Es geht aber auch anders:
Alternativer Beweis: Falls $|Q| \leq 2$ ist, ist die Behauptung wegen Bedingung 1 und 2 der Baumzerlegung erfüllt. Sei nun $|Q| \geq 3$:
Angenommen es existiert kein $i \in I$ mit $Q \subseteq X_i$. Dann existieren 3 paarweise verschiedene Knoten $w_1, w_2, w_3 \in Q$, so dass $\{w_1, w_2\}$, $\{w_1, w_3\}$, $\{w_2, w_3\}$ in 3 verschiedenen Taschen X_a, X_b, X_c liegen. Solche 3 Knoten existieren, denn sonst gäbe es eine Tasche X_i mit $Q \subseteq X_i$. Wegen Bedingung 3 der Baumzerlegung existiert ein Pfad zwischen X_a und X_b, zwischen X_a und X_c und zwischen X_b und X_c. Dies ist allerdings ein Widerspruch zur Kreisfreiheit von T. □

Dieser Satz gilt auch für die Pfadweite.

(3.7) Satz (S. Ramachandramurthi [28]-[30])
Für jeden Graphen $G = (V, E)$ gilt:

$$tw(G) \geq \gamma_R(G) \geq \delta_2(G) \geq \delta(G),$$

wobei $\delta_2(G) := \min\{d(v) | \ v \in V \text{ und } \exists w \in V \backslash \{v\} : \ d(w) \leq d(v)\}$ und

$$\gamma_R(G) := \begin{cases} |V| - 1 & \text{, falls } G \text{ eine Clique ist} \\ \min_{v,w \in V, \ \{v,w\} \notin E} \max\{d(v), d(w)\} & \text{, sonst.} \end{cases}$$ ◇

Beweis
Gegeben sei ein Graph $G = (V, E)$ und eine Baumzerlegung

$$(\{X_i | \ i \in I\}, \ T = (I, F))$$

der Weite $tw(G)$, so dass $|I|$ minimal ist. Der Beweis erfolgt in drei Schritten wie in [8]:

- **Behauptung 1:** Es existiert ein Knoten $v \in V$ mit $d(v) \leq tw(G)$.

Beweis
Falls die Baumzerlegung nur eine Tasche hat, ist die Behauptung trivial. Angenommen es gibt mindestens 2 Taschen: Sei $s \in I$ ein Blatt von T. Da die Baumzerlegung minimal ist, gibt es einen Knoten $v \in X_s$, so dass $v \notin X_t$ für alle $t \in I \backslash \{s\}$. Sonst wäre wegen der dritten Bedingung der Baumzerlegung jeder Knoten aus X_s auch in der Vorgängertasche, und die Tasche s wäre überflüssig. Ein Widerspruch zu Minimalität. Außerdem müssen wegen der zweiten Bedingung der Baumzerlegung auch alle Nachbarn von v in der Tasche s sein. Und somit ist $d(v) \leq tw(G)$. Insbesondere ist $\delta(G) \leq tw(G)$. □

- **Behauptung 2:** Falls der Graph G nicht vollständig ist, so enthält er mindestens 2 nicht benachbarte Knoten v, v', so dass $d(v), d(v') \leq tw(G)$.

  Beweis
  Wähle das Blatt $s \in I$ und den Knoten $v \in X_s$ so wie in Behauptung 1. Wähle dann ein weiteres Blatt $s' \in I \backslash \{s\}$. Der Baum T hat mindestens 2 Blätter, da G nicht vollständig ist. Da die Baumzerlegung minimal ist, gibt es einen Knoten $v' \in X_{s'}$, so dass $v' \notin X_w$ für alle $w \in I \backslash \{s'\}$. Die Knoten v und v' sind nicht benachbart, denn sonst wären sie zusammen in mindestens einer Tasche der Baumzerlegung. Da v' nur in einer Tasche ist, gilt auch: $d(v') \leq tw(G)$. Insgesamt folgt also die Behauptung. Insbesondere ist $tw(G) \geq \delta_2(G)$. □

- **Behauptung 3:** $tw(G) \geq \gamma_R(G)$.

  Beweis
  Falls der Graph vollständig ist, gilt: $tw(G) = |V| - 1 = \gamma_R(G)$. Jetzt können wir annehmen, dass der Graph nicht vollständig ist. Dann existieren wegen Behauptung 2, zwei nicht benachbarte Knoten $v, w \in V$ mit Grad kleiner gleich $tw(G)$. Also $\gamma_R(G) \leq tw(G)$. □

Der Rest, also $\gamma_R(G) \geq \delta_2(G) \geq \delta(G)$ folgt aus der Definition. □

(3.8) Satz (L. Sunil Chandran und C. R. Subramanian [35])
Für jeden Graphen $G = (V, E)$ gilt:

$$tw(G) \geq \max\left(\left\lceil \frac{d^*}{2} \right\rceil, \frac{1}{4 \cdot e \cdot (g+1)} \cdot (d^* - 1)^{\lfloor (g-1)/2 \rfloor} - 2\right).$$

wobei d^* der Durchschnittsgrad und g die Taillenweite vom Graphen ist. e ist der natürliche Logarithmus. ◇

Die Taillenweite g ist die Länge eines kürzesten, nicht trivialen Kreises des Graphen.

(3.9) Satz (Rose (1974) in [32])
Für jeden Graphen $G = (V, E)$ mit $tw(G) \leq k$ gilt:

$$|E| \leq k \cdot |V| - \frac{1}{2} \cdot k(k+1).$$

◇

Beweis
Sei $G = (V, E)$ ein Graph mit $tw(G) \leq k$. Der Beweis erfolgt mit einer Induktion über $|V| = n$ wie in Lemma 10.16 von [18].

- Falls $n \leq k+1$, so hat G höchstens so viele Kanten wie ein vollständiger Graph mit $k+1$ Knoten, d.h.

$$|E| \leq \frac{1}{2} \cdot k \cdot (k+1) = k \cdot (k+1) - \frac{1}{2} \cdot k \cdot (k+1) = k \cdot n - \frac{1}{2} \cdot k \cdot (k+1).$$

- Sei jetzt $n > k+1$. Nach Satz (3.7) existiert ein Knoten $v \in V$ mit $d(v) \leq k$. Sei $G' := G \backslash \{v\} = (V', E')$. Dann gilt erstens: $tw(G') \leq k$ und zweitens gilt:

$$\begin{aligned} |E| &\leq |E'| + k \\ &\overset{\text{I.V.}}{\leq} k \cdot (n-1) - \frac{1}{2} \cdot k \cdot (k+1) + k \\ &= k \cdot n - \frac{1}{2} \cdot k \cdot (k+1). \end{aligned}$$

Und damit ist der Beweis erbracht. □

Jetzt wird noch eine obere Schranke für die Pfadweite vorgestellt. Sie ist also auch gültig für die Baumweite.

(3.10) Satz (J. Kneis, D. Mölle, S. Richter und P. Rossmanith (2008) in [20])
Sei $G = (V, E)$ ein Graph. Dann gilt: $pw(G) \leq \frac{m}{5.769} + \mathcal{O}(\log(n))$. ◇

Für weitere untere und obere Schranken siehe H. L. Bodlaender und A. M. C. A. Koster [8] und [7].

§4 Das erste Modell

Ein mathematisches Modell ist nichts anderes als die Beschreibung eines Systems. Dabei können es mehrere mögliche Modelle für ein System geben, wobei die Qualität des Modells in erster Linie von unseren Anforderungen abhängt.

4.1 Anforderungen ans Modell

Abbildung 9 zeigt, dass es schon für einen Graphen mit nur 4 Knoten sehr viele sehr unterschiedliche Pfadzerlegungen gibt. Die drei Pfadzerlegungen haben zwar alle Weite 2, sie unterscheiden sich aber z.B. an der Anzahl der Taschen und an der Anzahl der Knoten in der ersten Tasche.

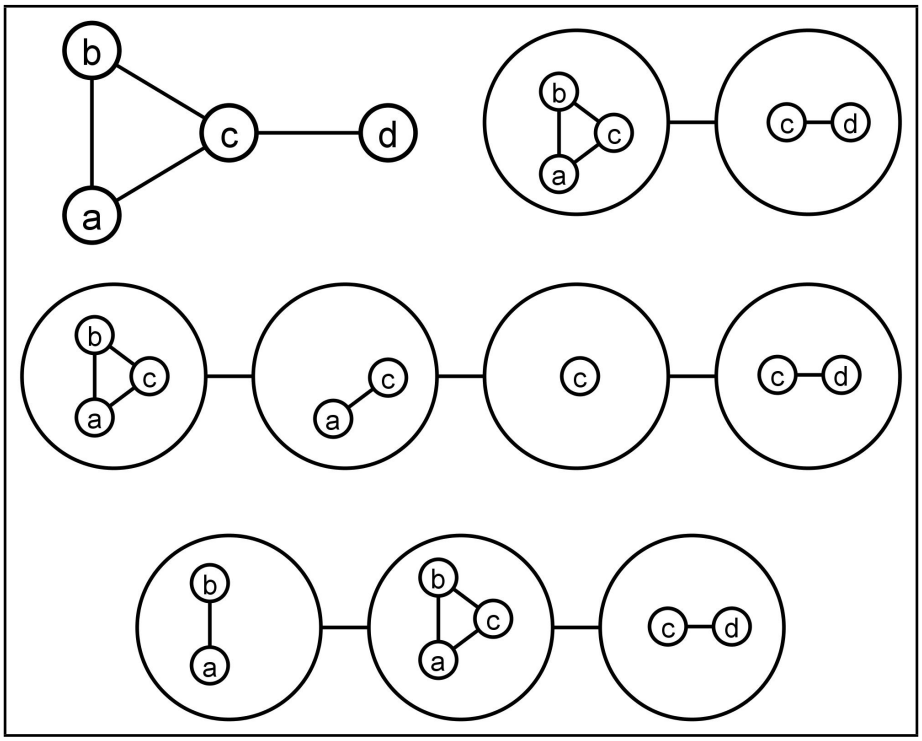

Abbildung 9: Beispiele möglicher Pfadzerlegungen für den Graphen G_1

Wir versuchen deswegen das Problem so zu modellieren, dass möglichst wenige Zerlegungen in Frage kommen, ohne dabei alle Pfadzerlegungen der gleichen Weite zu verbieten. Hilfreich dafür ist es, für einen gegebenen Graphen von vornherein

- die Anzahl der Taschen

- die Anzahl der Knoten in der ersten Tasche
- und die Anzahl der Knoten, die in jedem Schritt, d.h. von einer Tasche zur nächsten, hinzu bzw. heraus kommen

festzusetzen.

Ausserdem sollte das Modell möglichst wenige Variablen haben und in der Lage sein, schon bekannte untere und obere Schranken zu verwenden.

4.2 Modellierung

Die Richtigkeit des ersten Modells beruht auf dem folgenden Satz:

(4.1) Satz
Sei $G = (V, E)$ ein Graph mit Pfadweite k. Dann existiert eine Pfadzerlegung $(X_1, \ldots, X_r)$ der Weite k, so dass jede Tasche genau ein Knoten enthält, der noch nicht in der Vorgängertasche war. D.h. $\left|X_{i+1} \backslash X_i\right| = 1$ für alle $i \in \{1, \ldots, r-1\}$. $\diamond$

Beweis
Gegeben sei eine Pfadzerlegung der Weite k mit $\left|X_{i+1} \backslash X_i\right| \neq 1$ für mindestens ein $i \in \{1, \ldots, r-1\}$. Um daraus die geforderte Pfadzerlegung der gleichen Weite zu konstruieren, unterscheiden wir zwei Fälle:

1. Fall: $\left|X_{i+1} \backslash X_i\right| = 0$: Lösche X_{i+1} und verbinde X_i mit X_{i+2} (falls existent) und die Bedingung ist erfüllt.
2. Fall: $\left|X_{i+1} \backslash X_i\right| = h \geq 2$: Sei $X_{i+1} \backslash X_i = \{w_1, \ldots, w_h\}$. Füge einen Pfad von $h-1$ neue Taschen

$$\begin{aligned} X_{i_1} &= X_{i+1} \backslash \{w_2,\ w_3,\ \ldots,\ w_h\}, \\ X_{i_2} &= X_{i+1} \backslash \{w_3,\ \ldots,\ w_h\}, \\ &\vdots \\ X_{i_{h-2}} &= X_{i+1} \backslash \{w_{h-1},\ w_h\} \\ X_{i_{h-1}} &= X_{i+1} \backslash \{w_h\} \end{aligned}$$

   zwischen X_i und X_{i+1} ein.

Wir erhalten in beiden Fällen eine Pfadzerlegung der gleichen Weite. Wir können diesen Schritt so oft wiederholen, bis wir die gewünschte Pfadzerlegung erhalten.$\square$

Dieser Satz macht einem sofort klar, dass der Minimalgrad eine untere Schranke für die Pfadweite darstellt, denn der Knoten der in die letzte Tasche neu hinzu kommt, ist nur in einer Tasche. Also müssen auch alle seine Nachbarn in dieser Tasche sein. Und somit gilt: $pw(G) \geq \delta(G)$.

Gegeben sei nun ein beliebiger einfacher Graph G mit Knotenmenge V und Kantenmenge E. Jetzt können wir die Größe der ersten Tasche festlegen. Ideal hierfür ist die beste bekannte untere Schranke der Pfadweite +1 für den aktuellen Graphen zu wählen. Oft sind jedoch keine unteren Schranken bekannt. Deswegen wurde die am leichtesten zu bestimmende untere Schranke gewählt: der Minimalgrad $\delta(G)$. Ist eine bessere untere Schranke LB bekannt, kann man $\delta(G)$ im Folgenden einfach durch LB ersetzen.

Die grundlegende Idee des ersten Modells sieht dann folgendermaßen aus: Wir starten mit einer Tasche mit $\delta(G) + 1$ Knoten. Von einer Tasche zur nächsten kommt laut Satz (4.1) immer genau ein neuer Knoten hinzu, und beliebig viele können heraus genommen werden. Also brauchen wir genau $n - \delta(G)$ Taschen. Dafür sei die Menge aller Taschen gegeben durch

$$I := \{1, \ \ldots, \ n - \delta(G)\}.$$

Diese Modellierung ist sinnvoll, da wir so wenigstens für die ersten Taschen genau wissen, wie viele Knoten sich darin befinden. Diese Information ist hilfreich, um später zusätzliche gültige Ungleichungen zu bestimmen.

Wir verzichten ganz bewusst darauf, in jedem Schritt eine feste Anzahl an Knoten zu entfernen, da wir sonst keinen Einfluss darauf haben, wie viele Knoten in die erste Tasche kommen. Außerdem bleibt das Modell einfacher und hat weniger Ungleichungen.

Um das lineare Programm zu formulieren, benötigen wir zuerst die Variablen.

Wir benötigen zwei Variablen- Typen: Der eine gibt uns die Information in welche Tasche ein Knoten v zum ersten Mal hineinkommt, der andere besagt ab welcher Tasche der Knoten nicht mehr gebraucht wird, d.h. vor welcher Tasche er wieder herausgenommen wird:

$$k_{i,v} = \begin{cases} 1 & \text{, falls } i \text{ der kleinste Index der Tasche ist mit } v \in X_i \\ 0 & \text{, sonst} \end{cases}$$

$$y_{i,v} = \begin{cases} 1 & \text{, falls der Knoten } v \text{ in keiner Tasche mit Index größer oder gleich } i \text{ ist} \\ 0 & \text{, sonst} \end{cases}$$

Der Vorteil dieser Wahl der Variablen ist, dass so Bedingung 3 für die Pfadzerlegung automatisch erfüllt ist.

Ziel ist es, die maximale Anzahl der Knoten, die zusammen in einer Tasche sind, zu minimieren. Dazu benötigen wir eine weitere Variable:

$$c \in \mathbb{N}_0$$

die uns die größte Zahl an Knoten, die zusammen in einer Tasche sind, gibt. Dieses c gilt es zu minimieren, und $c - 1$ gibt uns dann die Weite der gefundenen Zerlegung.

(4.2) Beispiel
Als Beispiel betrachten wir den Graphen G_1 aus Abbildung 9. Von den drei Pfadzerlegungen genügt nur die untere Pfadzerlegung unserem Modell: Der Graph hat vier Knoten und der Minimalgrad ist 1. Also benötigen wir $4 - 1 = 3$ Taschen und es kommen zwei Knoten in die erste Tasche. Hier sind das die Knoten a und b. Also gilt:

$$k_{1,a} = k_{1,b} = 1, \quad k_{1,c} = k_{1,d} = 0 \quad \text{und} \quad y_{1,a} = y_{1,b} = y_{1,c} = y_{1,d} = 0.$$

Beide Knoten a und b müssen auch in der zweiten Tasche sein, da sie noch nicht zusammen mit Knoten c in einer Tasche waren. Zur zweiten Tasche kommt genau ein weiterer Knoten hinzu, nämlich c und wir erhalten

$$k_{2,a} = k_{2,b} = k_{2,d} = 0\,, \quad k_{2,c} = 1 \quad \text{und} \quad y_{2,a} = y_{2,b} = y_{2,c} = y_{2,d} = 0.$$

Die Knoten a und b müssen nicht mehr in die dritte Tasche, da sie mit allen ihren Nachbarn in der zweiten Tasche waren. In die dritte Tasche kommt der letzte noch verbliebene Knoten d neu hinzu:

$$k_{3,a} = k_{3,b} = k_{3,c} = 0 \quad , \quad k_{3,d} = 1 \quad \text{und} \quad y_{3,a} = y_{3,b} = 1\,, \quad y_{3,c} = y_{3,d} = 0.$$

Die Weite der Zerlegung ist 2. Dies entspricht auch der Pfadweite des angegebenen Graphen. ◇

Man kann leicht erkennen, dass $y_{1,v} = 0$ für alle Knoten $v \in V$, da ein Knoten zuerst in einer Tasche drin gewesen sein muss, bevor er endgültig in keiner anderen Tasche mehr ist. Diesen Gedanken können wir noch etwas weiterführen: Wegen Bedingung 2 aus der Definition der Pfadzerlegung muss auch jeder Nachbar von v zusammen mit v in mindestens einer Tasche sein. Da in der ersten Tasche aber nur $\delta(G) + 1$ Knoten hineinpassen und ab der zweiten Tasche genau ein weiterer Knoten hinzukommt, gilt sogar:

(4.3) Lemma
Für alle Knoten $v \in V$ und alle $i \in I$ mit $i \leq 1 + d(v) - \delta(G)$ gilt:

$$y_{i,v} = 0. \qquad \diamond$$

Da wir im Folgenden öfters ein $i \in I$ mit $i \geq 2 + d(v) - \delta(G)$ betrachten, definieren wir:

$$I_v := \{2 + d(v) - \delta(G), \ldots, n - \delta(G)\}.$$

4.3 Ganzzahliges Lineares Programm

4.3.1 Herleitung des Ganzzahligen Linearen Programms

Jetzt stellen wir das ganzzahlige lineare Programm auf: Laut Bedingung 1 der Pfadzerlegung muss jeder Knoten in mindestens einer Tasche sein:

$$\sum_{i \in I} k_{i,v} = 1 \qquad \forall v \in V.$$

In die erste Tasche sollen genau $\delta(G) + 1$ Knoten hineinkommen, also

$$\sum_{v \in V} k_{1,v} = \delta(G) + 1.$$

Ab der zweiten Tasche soll dann in jede Tasche genau ein neuer Knoten hinzukommen. Dies kann durch folgende Gleichung gefordert werden:

$$\sum_{v \in V} k_{i,v} = 1 \qquad \forall i \in I \backslash \{1\}.$$

Da wir $n - \delta(G)$ Taschen zur Verfügung haben, um die n Knoten zu verteilen und in die erste Tasche $\delta(G) + 1$ Knoten kommen und in alle anderen Taschen jeweils ein neuer Knoten hinzukommt, ist Bedingung 1 der Pfadzerlegung erfüllt. D.h. jeder Knoten ist mindestens in einer Tasche.

Bedingung 2 der Pfadzerlegung fordert, dass die beiden Endpunkte jeder Kante zusammen in mindestens einer Tasche vorkommen. Dazu genügt die folgende Ungleichung:

$$y_{i,v} \leq 1 - k_{i,w} \qquad \forall v \in V,\ \forall w \in N(v),\ \forall i \in I_v.$$

So wie wir die Variablen gewählt haben, ist die dritte Bedingung der Pfadzerlegung automatisch erfüllt. Es muss noch gelten: wenn ein Knoten in einer Tasche nicht mehr gebraucht wird, so wird er auch für alle folgenden Taschen nicht mehr gebraucht:

$$y_{i,v} \leq y_{i+1,v} \qquad \forall v \in V,\ \forall i \in I_v \backslash \{n - \delta(G)\}.$$

Und ein Knoten, der schon rausgenommen wurde, darf in keine spätere Tasche mehr rein:

$$y_{i,v} \leq 1 - k_{i,v} \qquad \forall v \in V,\ \forall i \in I_v.$$

Um die Pfadweite zu erhalten, minimieren wir die maximale Anzahl an Knoten, die zusammen in einer Tasche sind. Wir starten mit $\delta(G) + 1$ Knoten in Tasche 1. Dann kommt für jede weitere Tasche ein weiterer Knoten hinzu. Somit haben wir bis Tasche i genau $\delta(G) + i$ Knoten, die schon in mindestens einer Tasche waren. Um die Anzahl der Knoten in Tasche i zu bestimmen, brauchen wir nur noch alle Knoten abzuziehen, die wir bis Tasche i schon wieder rausgenommen haben, und wir erhalten:

$$\delta(G) + i - \sum_{\substack{v \in V \text{mit} \\ i \in I_v}} y_{i,v} \leq c \qquad \forall i \in I.$$

Das ganzzahlige lineare Programm (GLP) lautet nun:

$$\begin{aligned}
&\min\ c - 1 \\
&\text{s.d.:} \sum_{i \in I} k_{i,v} = 1 && \forall v \in V && (1)\\
&\sum_{v \in V} k_{i,v} = 1 && \forall i \in I \backslash \{1\} && (2)\\
&\sum_{v \in V} k_{1,v} = \delta(G) + 1 && && (3)\\
&y_{i,v} \leq y_{i+1,v} && \forall v \in V,\ \forall i \in I_v \backslash \{n - \delta(G)\} && (4)\\
&y_{i,v} \leq 1 - k_{i,w} && \forall v \in V,\ \forall w \in N[v],\ \forall i \in I_v && (5)\\
&\delta(G) + i - \sum_{\substack{v \in V \text{mit} \\ i \in I_v}} y_{i,v} \leq c && \forall i \in I && (6)\\
&k_{i,v}, y_{i,v} \in \{0,1\} && \forall v \in V,\ \forall i \in I && (7)
\end{aligned}$$

Es sei noch bemerkt, dass Bedingung (3) überflüssig ist, da sie durch (1) und (2) automatisch erfüllt sein muss.

Das Polyeder P_{GLP} bezeichnet die ganzzahlige Hülle der zulässigen Lösungen der Ungleichungen (1) bis (7). Um weitere gültige Ungleichungen oder sogar facettendefinierende Ungleichungen für P_{GLP} zu finden, betrachten wir die LP- Relaxierung. Diese entsteht durch Weglassen der Ganzzahligkeitsbedingung (7). Dann bestimmen wir die optimale Lösung der LP- Relaxierung x^* für einen beliebigen Graphen und bestimmen eine gültige Ungleichung für P_{GLP}, die von x^* verletzt wird.

4.3.2 Dimension des Polyeders

(4.4) Satz
Die Dimension des Polyeders ist:

$$\dim\big(P_{GLP}\big) = 2 \cdot n^2 - n \cdot \big(\delta(G) + 3\big) - 2 \cdot m + 2 + \delta(G). \qquad \diamond$$

Beweis
Wir haben $2 \cdot n \cdot (n - \delta(G)) + 1$ Variablen. Allerdings haben einerseits wegen Lemma (4.3)

$$\sum_{v \in V} 1 + d(v) - \delta(G) = n + 2 \cdot m - \delta(G) \cdot n$$

Variablen den festen Wert 0. Und andererseits enthält das ganzzahlige lineare Programm $n + (n - \delta(G) - 1)$ Gleichungen $\Big($(1) und (2)$\Big)$. Gleichung (3) folgt aus (1) und (2). Somit ist die Dimension des Polyeders P_{GLP}:

$$\begin{aligned}\dim\big(P_{GLP}\big) &= 2 \cdot n \cdot (n - \delta(G)) + 1 - (n + 2 \cdot m - \delta(G) \cdot n) - (n + n - \delta(G) - 1) \\ &= 2 \cdot n^2 - n \cdot (\delta(G) + 3) - 2 \cdot m + 2 + \delta(G) \qquad \square\end{aligned}$$

§5 Gültige Ungleichungen erster Ordnung

Es sei ein beliebiger, einfacher und zusammenhängender Graph $G = (V, E)$ mit Minimalgrad $\delta(G)$ und eine Pfadzerlegung des Graphen gemäß den Voraussetzungen unseres Modells gegeben. Die Werte der Variablen $k_{i,v}$, $y_{i,v}$ und c seien im Folgenden entsprechend gewählt.

5.1 Klasse A

Mit Hilfe von Ungleichung (5) können wir neue und stärkere Ungleichungen erzeugen: Wenn $y_{i,v} = 1$, also wenn der Knoten $v \in V$ weder in Tasche X_i ($i \in I_v$) noch in allen nachfolgenden Taschen vorkommt, dann müssen v sowie alle Nachbarn von v schon in einer Tasche mit Index kleiner als i sein. Auf der anderen Seite kann wegen (2) nur ein Knoten neu in eine Tasche kommen. Also folgt die Gültigkeit von:

(5.1) Gültige Ungleichung

$$y_{i,v} \leq 1 - k_{i,v} - \sum_{w \in N(v)} k_{i,w} \qquad \forall v \in V,\ \forall i \in I_v. \qquad \diamond$$

Beweis
Die Gültigkeit von Ungleichung (5.1) kann man mit der Chvátal- Gomory Prozedur beweisen. Für jeden Knoten $v \in V$ und für alle $i \in I$ gilt:

$$\begin{aligned} y_{i,v} + k_{i,w} &\overset{(5)}{\leq} 1 && \forall w \in N[v] \\ d(v) \cdot \sum_{w \in N[v]} k_{i,w} &\overset{(2)}{\leq} d(v) \end{aligned}$$

$$\Big(d(v)+1\Big) \cdot y_{i,v} + \Big(d(v)+1\Big) \cdot \sum_{w \in N[v]} k_{i,w} \leq \Big(2 \cdot d(v) + 1\Big)$$

Nachdem wir jetzt beide Seiten durch $\Big(d(v)+1\Big)$ teilen und die Koeffizienten der linken Seite abrunden,

$$\left\lfloor \frac{d(v)+1}{d(v)+1} \right\rfloor \cdot y_{i,v} + \left\lfloor \frac{d(v)+1}{d(v)+1} \right\rfloor \cdot \sum_{w \in N[v]} k_{i,w} \leq \frac{2 \cdot d(v) + 1}{d(v)+1}$$

können wir die Koeffizienten der rechten Seite auch abrunden, da die linke Seite ganzzahlig sein muss. Wir erhalten also

$$y_{i,v} + \sum_{w \in N[v]} k_{i,w} \leq \left\lfloor \frac{2 \cdot d(v) + 1}{d(v) + 1} \right\rfloor = 1$$

und somit ist die Gültigkeit der Ungleichung bewiesen. Da wir die Chvátal- Gomory Prozedur nur einmal anwenden mussten, liegt die Ungleichung im ersten Chvátal-Abschluss. □

(5.2) Satz
Ungleichung (5.1) definiert eine Facette für alle Knoten $v \in V$ mit $1 \leq d(v) \leq n-2$ und alle Indizes $i \in I$ mit $i \geq 2 + d(v) - \delta(G)$, so dass für alle $w \in V \backslash N[v]$ mit $2 + d(w) - \delta(G) > i$ gilt:

$$|N[v] \cap N[w]| \geq 1.$$

◇

Beweis
Laut Satz (4.4) ist die Dimension der ganzzahligen Hülle:

$$\dim\,(P_{GLP}) = 2 \cdot n^2 - \delta(G) \cdot n - 3 \cdot n - 2 \cdot m + 2 + \delta(G).$$

Wir suchen also $2 \cdot n^2 - \delta(G) \cdot n - 3 \cdot n - 2 \cdot m + 2 + \delta(G)$ affin unabhängige Punkte, die die Ungleichung

$$y_{i,v} \leq \sum_{w \in V \backslash N[v]} k_{i,w} \tag{8}$$

für feste i, v gemäß den Voraussetzungen des Satzes mit Gleichheit erfüllen.

(a) Setze $c = n$ und $y_{j,x} = 0$ für alle $x \in V$ und alle $j \in I_x$. Da $y_{i,v} = 0$, muss auch $k_{i,w} = 0$ für alle $w \in V \backslash N[v]$ gelten.

- Setze $k_{i,v} = 1$. Es müssen noch $n-1$ Knoten auf $n - \delta(G) - 1$ Taschen verteilt werden. Dafür gibt es

  $$(n-1) \cdot (n - \delta(G) - 1) - (n-1) - (n - \delta(G) - 2) + 1$$

  affin unabhängige Punkte[1].

[1] Das erhält man, indem man das dazugehörige LP aufstellt und die Dimension bestimmt.

- Setze $k_{i,u} = 1$ für ein $u \in N(v)$. Dann gibt es $n - \delta(G) - 1$ verschiedene $j \in I \backslash \{i\}$ um $k_{j,v} = 1$ zu setzen. Wir können also weitere

$$n - \delta(G) - 1$$

  affin unabhängige Punkte konstruieren.

- Falls $d(v) \geq 2$, dann können wir für jedes $x \in N[v] \backslash \{v, u\}$ einen weiteren Punkt mit $k_{i,x} = 1$ konstruieren, also weitere

$$d(v) - 1$$

  affin unabhängige Punkte. Falls $d(v) = 1$, ist diese Zahl sowieso gleich 0.

(b) Wir konstruieren für alle $w \in V$ und alle $j \in I_w$ jeweils einen Punkt mit $y_{j,w} = 1$, $y_{j-1,w} = 0$, $c = n$, $y_{n-\delta(G),x} = 0$ (falls möglich) für alle $x \neq w$, der die Ungleichung (8) mit Gleichheit erfüllt:

- Falls $2 + d(w) - \delta(G) \leq i$ und $|N[v] \backslash N[w]| \geq 1$, dann setzen wir alle Knoten aus $N[w]$ in die ersten $1 + d(w) - \delta(G)$ Taschen, $y_{j,w} = 1$ und $k_{i,u} = 1$ für ein $u \in N[v] \backslash N[w]$. Da $y_{n-\delta(G),x} = 0$ für alle $x \neq w$, ist insbesondere $y_{i,v} = 0$. Somit ist Ungleichung (8) mit Gleichheit erfüllt.

- Falls $2 + d(w) - \delta(G) \leq i$, $|N[v] \backslash N[w]| = 0$ und $v \neq w$ oder falls $v = w$ und $j \leq i$, dann setzen wir alle Knoten aus $N[w]$ in die ersten $1 + d(w) - \delta(G)$ Taschen und $y_{j,w} = 1$, $y_{i,v} = 1$. Somit ist Ungleichung (8) mit Gleichheit erfüllt.

- Falls $v = w$ und $j > i$, dann setzen wir alle Knoten aus $N[v]$ in die ersten i Taschen, so dass $k_{i,u} = 1$ für ein $u \in N[v]$. Dann ist $y_{i,v} = 0$, $y_{j,v} = 1$ und in Ungleichung (8) gilt Gleichheit.

- Falls $2 + d(w) - \delta(G) > i$ und $|N[v] \cap N[w]| \geq 1$, dann setzen wir alle Knoten aus $N[w]$ in die ersten $1 + d(w) - \delta(G)$ Taschen, wobei $k_{i,u} = 1$ für ein $u \in N[v] \cap N[w]$ und $y_{j,w} = 1$, $y_{i,v} = 0$. Somit ist Ungleichung (8) mit Gleichheit erfüllt.

- Falls $2 + d(w) - \delta(G) > i$ und $|N[v] \cap N[w]| = 0$, dann existiert kein Punkt mit $y_{2+d(w)-\delta(G),w} = 1$, der Ungleichung (8) mit Gleichheit erfüllt. Somit definiert die Ungleichung in diesem Fall auch keine Facette.

Dies können wir nacheinander für alle $w \in V$ und alle $j \geq 2 + d(w) - \delta(G)$ machen und erhalten

$$\sum_{w \in V} \Big(n - \delta(G) - \Big(2 + d(w) - \delta(G) \Big) + 1 \Big) = \sum_{w \in V} \Big(n - d(w) - 1 \Big) = n^2 - 2 \cdot m - n$$

affin unabhängige Punkte.

(c) Setze $c = n$, $y_{i,v} = 1$, $y_{i-1,v} = 0$ und $y_{j,w} = 0$ für alle $w \neq v$ und alle $j \in I_w$. Wir haben $n - d(v) - 1$ Möglichkeiten, um $k_{i,w} = 1$ für ein $w \in V \backslash N[v]$ zu setzen. Eine dieser Möglichkeiten wurde schon in (b) betrachtet. Also bleiben noch

$$n - d(v) - 2$$

affin unabhängige Punkte.

(d) In allen Punkten davor wurde $c = n$ gesetzt. Den letzten affin unabhängigen Punkt konstruieren wir, indem wir einen Punkt aus (b) oder (c) nehmen und $c = n - 1$ setzen.

Insgesamt erhalten wir

$$\begin{aligned}
&(n-1)\cdot(n-\delta(G)-1)-(n-1)-(n-\delta(G)-2)+1\\
+&(n-\delta(G)-1)+(d(v)-1)+(n^2-2\cdot m-n)+(n-d(v)-2)+1\\
=&2\cdot n^2-\delta(G)\cdot n-3\cdot n-2\cdot m+2+\delta(G)
\end{aligned}$$

affin unabhängige Punkte und damit ist der Satz bewiesen. □

(5.3) Beispiel
Wir betrachten den Graphen G_1 aus Abbildung 10. Laut Satz (4.4) ist die Dimension der zugehörigen ganzzahligen Hülle:

$$\begin{aligned}
\dim\,(P_{GLP}(G_1)) &= 2n^2 - n\big(\delta(G)+3\big) - 2m + 2 + \delta(G)\\
&= 2\cdot 4^2 - 4\cdot(1+3) - 2\cdot 4 + 2 + 1 = 11
\end{aligned}$$

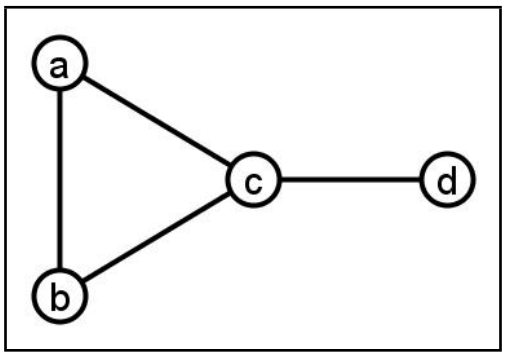

Abbildung 10: Der Graph G_1

Aus (5.1) folgt die Gültigkeit der Ungleichungen

$$\begin{aligned} y_{2,d} &\leq 1 - k_{2,c} - k_{2,d} && = k_{2,a} + k_{2,b} \\ y_{3,a} &\leq 1 - k_{3,a} - k_{3,b} - k_{3,c} && = k_{3,d} \\ y_{3,b} &\leq 1 - k_{3,a} - k_{3,b} - k_{3,c} && = k_{3,d} \\ y_{3,d} &\leq 1 - k_{3,c} - k_{3,d} && = k_{3,a} + k_{3,b} \end{aligned}$$

Diese Ungleichungen definieren laut Satz (5.2) Facetten von $P_{GLP}(G_1)$. ◇

In den Fällen, wo Ungleichung (5.1) nicht Facetten- definierend ist, können wir sie liften: Siehe dazu Beispiel (6.2).

Sei ein beliebiger Knoten $v \in V$ gegeben. Wir wissen bereits, dass $y_{i,v} = 0$ für alle $i \leq 1 + d(v) - \delta(G)$ ist. Wir versuchen jetzt für $i = 2 + d(v) - \delta(G)$ gültige Ungleichungen herzuleiten. Danach erhöhen wir das i schrittweise um 1 und finden so weitere gültige Ungleichungen.

(5.4) Gültige Ungleichung

$$y_{i,v} \leq \sum_{j=i}^{n-\delta(G)} k_{j,w} \qquad \forall v \in V \text{ mit } d(v) \leq n-2$$
$$\forall w \in V \backslash N[v] \text{ und } i = 2 + d(v) - \delta(G) \in I. \quad \diamond$$

Beweis
Sei $i = 2 + d(v) - \delta(G)$. Wir unterscheiden zwei Fälle:

- Falls $y_{i,v} = 0$, ist die Ungleichung gültig, denn die rechte Seite kann nie negativ werden.
- Falls $y_{i,v} = 1$, dann sind v und alle seine Nachbarn schon in den ersten $i - 1 = 1 + d(v) - \delta(G)$ Taschen, wo genau $d(v) + 1$ verschiedene Knoten reinpassen. Somit darf kein anderer Knoten, also kein $w \in V \backslash N[v]$, in den ersten $1 + d(v) - \delta(G)$ Taschen sein. Also $\sum_{j=i}^{n-\delta(G)} k_{j,w} = 1$ für alle $w \in V \backslash N[v]$.

D.h. die Ungleichung ist gültig. □

(5.5) Satz
Ungleichung (5.4) definiert eine Facette für alle Knoten $v \in V$ mit $1 \leq d(v) \leq n-2$, $i = 2 + d(v) - \delta(G) \in I$ und alle Knoten $w \in V \backslash N[v]$, so dass kein $x \in V \backslash \{v, w\}$ existiert mit

$$w \notin N[x] \quad , \quad i \leq 2 + d(x) - \delta(G) \quad \text{und} \quad N[v] \not\subseteq N[x]. \quad \diamond$$

Beweis
Wir brauchen $2 \cdot n^2 - \delta(G) \cdot n - 3 \cdot n - 2 \cdot m + 2 + \delta(G)$ affin unabhängige Punkte, die die Ungleichung

$$y_{i,v} \leq \sum_{j=i}^{n-\delta(G)} k_{j,w} \tag{9}$$

für entsprechende i, v, w mit Gleichheit erfüllen.

Falls $d(v) > \delta(G)$:

(a) Setze $c = n$ und $y_{h,x} = 0$ für alle $x \in V$ und alle $h \in I_x$. Da $y_{i,v} = 0$, muss auch $k_{j,w} = 0$ für alle $j \in \{i, \ldots, n - \delta(G)\}$ gelten.

- Setze $k_{i-1,w} = 1$. Es müssen noch $n-1$ Knoten auf $n - \delta(G) - 1$ Taschen verteilt werden. Dafür gibt es

  $$(n-1) \cdot (n - \delta(G) - 1) - (n-1) - (n - \delta(G) - 2) + 1$$

  affin unabhängige Punkte.

- Setze $k_{i-2,w} = 1$. Dann gibt es $n-1$ verschiedene $x \in V \backslash \{w\}$ um $k_{i,x} = 1$ zu setzen. Dies geht nur, falls $i \geq 3$, also falls $d(v) > \delta(G)$. Wir können also weitere

  $$n - 1$$

  affin unabhängige Punkte konstruieren.

- Konstruiere Punkte, für die nacheinander $k_{j,w} = 1$ gilt, für $j \in \{1, \ldots, i-3\}$. Wir können also weitere

  $$d(v) - \delta(G) - 1$$

  affin unabhängige Punkte konstruieren.

(b) Wähle ein $x \in V$ und ein $h \in I_x$ und setze $c = n$, $y_{h,x} = 1$, $y_{h-1,x} = 0$ und $y_{n-\delta(G),u} = 0$ für alle $u \in V \backslash \{x\}$. Dann unterscheiden wir vier Fälle:

- Falls $w \in N[x]$, setze alle Knoten aus $N[x]$ in die ersten $1 + d(x) - \delta(G)$ Taschen, wobei $k_{1,w} = 1$ und $y_{i,v} = 0$. Somit wird Ungleichung (9) mit Gleichheit erfüllt.

- Falls $w \notin N[x]$ und $2 + d(x) - \delta(G) \leq i - 1$, setze alle Knoten aus $N[x]$ in die ersten $1 + d(x) - \delta(G)$ Taschen und $k_{i-1,w} = 1$. In Ungleichung (9) gilt dann Gleichheit.

- Falls $w \notin N[x]$, $2 + d(x) - \delta(G) \geq i$ und $N[v] \subseteq N[x]$, setze alle Knoten aus $N[v]$ in die ersten $i-1$ Taschen und alle Knoten aus $N[x]$ in die ersten $h-1$ Taschen und $k_{n-\delta(G),w} = 1$, so dass $y_{i,v} = 1$ und $y_{h,x} = 1$ gelten kann. Ungleichung (9) ist mit Gleichheit erfüllt.
- Der Fall $w \notin N[x]$, $2 + d(x) - \delta(G) \geq i$ und $N[v] \not\subseteq N[x]$ ist laut den Voraussetzungen des Satzes ausgeschlossen, denn dann gibt es keinen Punkt mit $y_{h,x} = 1$ für $h = 2 + d(x) - \delta(G)$ der die Ungleichung (9) mit Gleichheit erfüllt.

So können wir nacheinander für alle $x \in V$ und alle $h \geq 2 + d(x) - \delta(G)$

$$\sum_{x \in V} \Big(n - d(x) - 1 \Big) = n^2 - 2 \cdot m - n$$

affin unabhängige Punkte konstruieren.

(c) Setze $c = n$, $y_{i,v} = 1$ und $y_{h,x} = 0$ für alle $x \neq v$ und alle $h \in I_x$. Es gibt $n - \delta(G) - i + 1 = n - d(v) - 1$ verschiedene $j \in \{i, \ldots, n - \delta(G)\}$ um $k_{j,w} = 1$ zu setzen. Ein solcher Punkt wurde schon in (b) konstruiert. Es bleiben

$$n - d(v) - 2$$

affin unabhängige Punkte.

(d) In allen Punkten davor wurde $c = n$ gesetzt. Den letzten affin unabhängigen Punkt konstruieren wir, indem wir einen Punkt aus (b) oder (c) nehmen und $c = n - 1$ setzen.

Insgesamt erhalten wir

$$\begin{aligned} &(n-1) \cdot (n - \delta(G) - 1) - (n-1) - (n - \delta(G) - 2) + 1 \\ &+(n-1) + (d(v) - \delta(G) - 1) + (n^2 - 2 \cdot m - n) + (n - d(v) - 2) + 1 \\ &=2 \cdot n^2 - \delta(G) \cdot n - 3 \cdot n - 2 \cdot m + 2 + \delta(G) \end{aligned}$$

affin unabhängige Punkte und damit ist der Satz für $d(v) > \delta(G)$ bewiesen.

Sei jetzt $d(v) = \delta(G)$. Dann ist $i = 2$:

- Setze $c = n$ und $y_{h,x} = 0$ für alle $x \in V$ und alle $h \in I_x$. Da $y_{2,v} = 0$, muss auch $k_{1,w} = 1$ gelten. Es müssen noch $n-1$ Knoten auf $n - \delta(G)$ Taschen verteilt werden, wobei in die erste Tasche noch genau $\delta(G)$ Knoten kommen. Dafür gibt es
$$(n-1) \cdot (n - \delta(G)) - (n-1) - (n - \delta(G) - 1) + 1$$
affin unabhängige Punkte.

- Die Abschnitte (b), (c), (d) für $d(v) > \delta(G)$ gelten auch für $d(v) = \delta(G)$. Somit erhalten wir weitere

$$(n^2 - 2 \cdot m - n) + (n - d(v) - 2) + 1$$

affin unabhängige Punkte.

Insgesamt erhalten wir

$$\begin{aligned} &(n-1) \cdot (n - \delta(G)) - (n-1) - (n - \delta(G) - 1) + 1 \\ &+ (n^2 - 2 \cdot m - n) + (n - d(v) - 2) + 1 \\ \overset{d(v)=\delta(G)}{=} &2 \cdot n^2 - \delta(G) \cdot n - 3 \cdot n - 2 \cdot m + 2 + \delta(G) \end{aligned}$$

affin unabhängige Punkte und damit ist der Satz auch für $d(v) = \delta(G)$ bewiesen. □

Falls wir i in Ungleichung (5.4) um eins erhöhen, also $i = 3 + d(v) - \delta(G)$, dann könnte sein, dass $k_{h,w} = 1$ für ein $h \in \{1, \ldots, i-1\}$ und $w \in V \backslash N[v]$. In diesem Fall wäre $\sum_{j=i}^{n-\delta(G)} k_{j,w} = 0$, obwohl $y_{i,v} = 1$ sein könnte. Aus genau diesem Grund addieren wir noch die Summe für einen zweiten Knoten aus $V \backslash N[v]$ zur rechten Seite hinzu:

$$y_{i,v} \leq \sum_{j=i}^{n-\delta(G)} \sum_{w \in S} k_{j,w} \qquad \begin{array}{l} \forall v \in V,\ \forall S \subseteq V \backslash N[v] \text{ mit } |S| = 2, \\ \forall i = 3 + d(v) - \delta(G) \end{array}$$

Für $i = 4 + d(v) - \delta(G)$ muss analog $|S| = 3$ sein, und so weiter. Auf diesem Prinzip basiert die folgende gültige Ungleichung, die die gültigen Ungleichungen (5.1) und (5.4) verallgemeinert:

(5.6) Gültige Ungleichung (Klasse A)
Für alle Knoten $v \in V$ und für alle Taschenindizes $i \in I$ mit $i \geq 2 + d(v) - \delta(G)$ gilt:

$$y_{i,v} \leq \sum_{j \in J} \sum_{w \in S} k_{j,w} \qquad \begin{array}{l} \forall\ S \subseteq V \backslash N[v],\ \forall\ J \subseteq \{i, \ldots, n - \delta(G)\} \\ \text{mit } |S|,\ |J| \geq 1 \text{ und } |S| + |J| = n - d(v). \end{array} \quad \diamond$$

Beweis
Falls $y_{i,v} = 0$, ist die Ungleichung trivialerweise erfüllt.

Sei jetzt $y_{i,v} = 1$. Dann sind in den ersten $i - 1$ Taschen genau

$$h := \Big(i - 1 + \delta(G)\Big) - \Big(d(v) + 1\Big) = i - 2 + \delta(G) - d(v)$$

verschiedene Knoten aus $V \backslash N[v]$.

- Falls $J = \{i, \ldots, n - \delta(G)\}$ muss $|S| = h + 1$ gelten, damit sichergestellt ist, dass für mindestens einen Knoten $w \in S$ und ein $j \in J$, $k_{j,w} = 1$ gilt. In diesem Fall ist

$$|S| + |J| = \Big(i - 2 + \delta(G) - d(v) + 1\Big) + \Big(n - \delta(G) - i + 1\Big) = n - d(v)$$

  und die Ungleichung ist gültig. Wäre $|S| \leq h$, dann könnten alle Knoten aus S schon in den ersten $i - 1$ Taschen sein.

- Falls $J = \{i, \ldots, n - \delta(G)\} \backslash \{p\}$ für ein $p \in \{i, \ldots, n - \delta(G)\}$, dann muss analog $|S| = h + 2 = i + \delta(G) - d(v)$ gelten, damit die Ungleichung gültig bleibt. D.h. falls J ein Element weniger hat, muss $|S|$ ein Element mehr haben.

- u.s.w.

- Falls $|J| = 1$, muss

$$|S| = h + 1 + n - \delta(G) - i = n - d(v) - 1$$

  gelten, damit die Ungleichung gültig bleibt.

In allen Fällen gilt also $|S| + |J| = n - d(v)$ und somit ist die Gültigkeit bewiesen. □

(5.7) Bemerkung
Für $|J| = 1$ reduziert sich Ungleichung (5.6) zu (5.1) und für $|S| = 1$ reduziert sie sich zu (5.4). Bei Graphen über 20 Knoten erzeugt Ungleichung (5.6) schon viel zu viele Ungleichungen. Deswegen werden in der Rechenstudie bei solchen Graphen nur die wichtigsten Fälle betrachtet: die Ungleichungen (5.1) und (5.4). ◇

5.2 Klasse B

Seien nun $i \in I$ und $v \in V$ beliebig gewählt. Falls $y_{i,v} = 1$, dann darf der Knoten $w \in N[v]$ weder in der i- ten Tasche noch in allen darauffolgenden Taschen sein. Also

(5.8) Gültige Ungleichung

$$y_{i,v} \leq 1 - \sum_{j=i}^{n-\delta(G)} k_{j,w} \quad \forall v \in V,\ \forall w \in N[v],\ \forall i \in I_v.$$

◇

Beweis
Die Gültigkeit dieser Ungleichung können wir auch wieder mit der Chvátal- Gomory Prozedur beweisen. Sei $v \in V,\ w \in N[v]$ und $i \in I_v$ gegeben. Dann gilt:

$$\begin{aligned}
y_{i,v} &\overset{(5)}{\leq} 1 - k_{i,w} \\
y_{i,v} &\overset{(4)}{\leq} y_{i+1,v} \overset{(5)}{\leq} 1 - k_{i+1,w} \\
&\vdots \\
y_{i,v} &\overset{(4)}{\leq} y_{n-\delta(G),v} \overset{(5)}{\leq} 1 - k_{n-\delta(G),w} \\
0 &\overset{(1)}{\leq} \Big(n - \delta(G) - i\Big) - \Big(n - \delta(G) - i\Big) \cdot \sum_{j=i}^{n-\delta(G)} k_{j,w}
\end{aligned}$$

$$\Big(n - \delta(G) - i + 1\Big) \cdot y_{i,v} \leq \Big(2 \cdot (n - \delta(G) - i) + 1\Big) - \Big(n - \delta(G) - i + 1\Big) \cdot \sum_{j=i}^{n-\delta(G)} k_{j,w}$$

$$\Longrightarrow \left\lfloor \frac{\Big(n - \delta(G) - i + 1\Big)}{\Big(n - \delta(G) - i + 1\Big)} \right\rfloor \cdot y_{i,v} \leq \left\lfloor \frac{\Big(2 \cdot (n - \delta(G) - i) + 1\Big)}{\Big(n - \delta(G) - i + 1\Big)} \right\rfloor - \left\lfloor \frac{\Big(n - \delta(G) - i + 1\Big)}{\Big(n - \delta(G) - i + 1\Big)} \right\rfloor \cdot \sum_{j=i}^{n-\delta(G)} k_{j,w}$$

Diese Ungleichung ist ein CG- Schnitt und im ersten Chvátal- Abschluss enthalten.□

(5.9) Satz
Ungleichung (5.8) definiert eine Facette für alle Knoten $v \in V$ mit $0 \leq d(v) \leq n-2$, alle Indizes $i \in I_v$ mit

$$2 + d(v) - \delta(G) \leq i \leq n - \delta(G) - 1$$

und alle Knoten $w \in N[v]$, so dass kein $x \in N[w] \backslash \{v\}$ existiert mit

$$d(x) + 2 - \delta(G) \leq i \quad \text{und} \quad |N[x] \cup N[v]| \geq i + \delta(G). \qquad \diamond$$

Beweis
Wir suchen $2 \cdot n^2 - \delta(G) \cdot n - 3 \cdot n - 2 \cdot m + 2 + \delta(G)$ affin unabhängige Punkte, die die Ungleichung

$$y_{i,v} \leq \sum_{j=1}^{i-1} k_{j,w} \tag{10}$$

für entsprechende i, v, w mit Gleichheit erfüllen.

(a) Setze $c = n$ und $y_{h,x} = 0$ für alle $x \in V$ und alle $h \in I_x$. Da $y_{i,v} = 0$, muss auch $k_{j,w} = 0$ für alle $j \in \{1, \ldots, i-1\}$ gelten.

- Setze $k_{i,w} = 1$. Es müssen noch $n-1$ Knoten auf $n - \delta(G) - 1$ Taschen verteilt werden. Dafür gibt es

$$(n-1) \cdot (n - \delta(G) - 1) - (n-1) - (n - \delta(G) - 2) + 1$$

affin unabhängige Punkte.

- Setze $k_{i+1,w} = 1$. Dann gibt es $n-1$ verschiedene $x \in V \backslash \{w\}$ um $k_{i,x} = 1$ zu setzen. Dies geht nur falls $i \leq n - \delta(G) - 1$. Wir können also weitere

$$n-1$$

affin unabhängige Punkte konstruieren.

- Falls $i \leq n - \delta(G) - 2$, konstruiere Punkte, für die nacheinander gilt:

$$k_{i+2,w} = 1, k_{i+3,w} = 1, \ldots, k_{n-\delta(G),w} = 1.$$

Wir können also weitere

$$n - \delta(G) - i - 1$$

affin unabhängige Punkte konstruieren. Dies gilt auch für $i = n - \delta(G) - 1$, denn dann ist $n - \delta(G) - i - 1 = 0$.

(b) Wähle ein $x \in V$ und ein $j \in I_x$ und setze $c = n$, $y_{j,x} = 1$, $y_{j-1,x} = 0$ und $y_{n-\delta(G),u} = 0$ für alle $u \in V \backslash \{x\}$ (falls möglich). Dann unterscheiden wir drei Fälle:

- Falls $x \notin N[w]$ setze alle Knoten aus $N[x]$ in die ersten $1 + d(x) - \delta(G)$ Taschen und $k_{n-\delta(G),w} = 1$. Somit wird Ungleichung (10) mit Gleichheit erfüllt.
- Falls $x \in N[w]$, $i \geq 2 + d(x) - \delta(G)$ und $i \geq |N[v] \cup N[x]| + 1 - \delta(G)$ setze $k_{1,w} = 1$ und die übrigen Variablen so, dass $y_{j,x} = 1$, $y_{j-1,x} = 0$ und $y_{i,v} = 1$, $y_{i-1,v} = 0$ gilt. Somit gilt Gleichheit in (10).
- Falls $x \in N[w]$, $i \leq 1 + d(x) - \delta(G)$ konstruieren wir einen Punkt mit $k_{i,w} = 1$ und die übrigen Variablen so, dass $y_{j,x} = 1$, $y_{j-1,x} = 0$ und $y_{i,v} = 0$ gilt. Auch jetzt gilt Gleichheit in (10).

Dies können wir nacheinander für alle $x \in V$ und alle $j \geq 2 + d(x) - \delta(G)$ machen und erhalten

$$\sum_{x \in V} \Big(n - d(x) - 1 \Big) = n^2 - 2 \cdot m - n$$

affin unabhängige Punkte.

(c) Setze $c = n$, $y_{i,v} = 1$, $y_{i-1,v} = 0$ und $y_{h,x} = 0$ für alle $x \in V \backslash \{v\}$ und alle $h \in I_x$. Wir haben $i - 1$ Möglichkeiten, um $k_{j,w} = 1$ für ein $j \in \{1, \ldots, i-1\}$ zu setzen. Eine dieser Möglichkeiten wurde schon in (b) betrachtet. Also bleiben noch $i - 2$ affin unabhängige Punkte.

(d) In allen Punkten davor wurde $c = n$ gesetzt. Den letzten affin unabhängigen Punkt konstruieren wir, indem wir einen Punkt aus (b) oder (c) nehmen und $c = n - 1$ setzen.

Insgesamt erhalten wir

$$\begin{aligned} &(n-1) \cdot (n - \delta(G) - 1) - (n-1) - (n - \delta(G) - 2) + 1 \\ &+(n-1) + (n - \delta(G) - i - 1) + (n^2 - 2 \cdot m - n) + (i-2) + 1 \\ &=2 \cdot n^2 - \delta(G) \cdot n - 3 \cdot n - 2 \cdot m + 2 + \delta(G) \end{aligned}$$

affin unabhängige Punkte und damit ist der Satz bewiesen. □

In den Fällen wo Ungleichung (5.8) keine Facette definiert, können wir sie liften. Siehe dazu Kapitel 6.1.

Sei ein beliebiger Knoten $v \in V$ gegeben, $i = 2 + d(v) - \delta(G) \in I$ und $y_{i,v} = 1$. Dann dürfen nur Knoten aus $N[v]$ in den ersten $i - 1$ Taschen sein:

(5.10) Gültige Ungleichung

$$y_{i,v} \leq \sum_{w \in N[v]} k_{j,w} \qquad \begin{aligned} &\forall v \in V \text{ mit } d(v) \leq n-2, \\ &\forall i, j \in I, \text{ mit } i = 2 + d(v) - \delta(G) \text{ und } 1 \leq j < i. \end{aligned} \qquad \diamond$$

Beweis
Sei $i = 2 + d(v) - \delta(G)$. Wir unterscheiden zwei Fälle:

- Falls $y_{i,v} = 0$, ist die Ungleichung gültig, denn die rechte Seite kann nie negativ werden.

- Falls $y_{i,v} = 1$, dann müssen v und alle seine Nachbarn ($d(v)+1$ Knoten) schon in den ersten $i-1 = 1+d(v)-\delta(G)$ Taschen sein. Da passen nach dem gewählten Modell genau $i-1+\delta(G) = d(v)+1$ Knoten rein. Somit darf kein anderer Knoten, also kein $w \in V\backslash N[v]$, in den ersten $1+d(v)-\delta(G)$ Taschen sein. Also $\sum_{w\in N[v]} k_{j,w} = 1$ für alle $j < i$.

D.h. die Ungleichung ist gültig. □

(5.11) Satz
Ungleichung (5.10) definiert eine Facette für alle Knoten $v \in V$ mit $1 \leq d(v) \leq n-3$, $i = 2+d(v)-\delta(G)$ und alle Indizes $j \in I$ mit $1 < j < i$ so dass

- kein $x \in V$ existiert, mit $N[x] \subseteq N[v]$, $1+d(x)-\delta(G) \geq j$ und $i \leq |N[v] \cup N[x]| - \delta(G)$.
- kein $x \in V\backslash\{v\}$ existiert, mit $j \geq 2+d(x)-\delta(G)$, $|V\backslash(N[v]\cup N[x])| = 0$ und $i \leq |N[v]\cup N[x]| - \delta(G)$. ◇

Beweis
Wir brauchen $2\cdot n^2 - \delta(G)\cdot n - 3\cdot n - 2\cdot m + 2 + \delta(G)$ affin unabhängige Punkte, die die Ungleichung

$$y_{i,v} \leq \sum_{w\in N[v]} k_{j,w} \tag{11}$$

für gegebene v, i, j gemäß den Voraussetzungen des Satzes mit Gleichheit erfüllen.

(a) Setze $c = n$ und $y_{h,x} = 0$ für alle $x \in V$ und alle $h \in I_x$. Da $y_{i,v} = 0$, muss auch $k_{j,w} = 0$ für alle $w \in N[v]$ gelten.

- Setze $k_{j,x_1} = 1$ für ein $x_1 \in V\backslash N[v]$. Es müssen noch $n-1$ Knoten auf $n-\delta(G)-1$ Taschen verteilt werden. Dafür gibt es

$$(n-1)\cdot(n-\delta(G)-1) - (n-1) - (n-\delta(G)-2) + 1$$

affin unabhängige Punkte.

- Setze $k_{j,x_2} = 1$ für ein $x_2 \in V\backslash(N[v]\cup\{x_1\}$. Es gibt $n-\delta(G)-1$ verschiedene $h \in \{1,\ldots,n-\delta(G)\}\backslash\{j\}$, um $k_{h,x_1} = 1$ zu setzen. Wir können also weitere

$$n-\delta(G)-1$$

affin unabhängige Punkte konstruieren.

- Konstruiere für jedes $x_3 \in V\backslash\big(N[v] \cup \{x_1, x_2\}\big)$ einen Punkt mit $k_{j,x_3} = 1$ und wir erhalten weitere
$$n - d(v) - 3$$
affin unabhängige Punkte. Falls $d(v) = n - 3$ ist diese Zahl sowieso gleich 0.

(b) Wir konstruieren für alle $x \in V$ und alle $h \in I_x$ einen Punkt mit $y_{h,x} = 1$, $y_{h-1,x} = 0$, $c = n$, $y_{n-\delta(G),z} = 0$ (falls möglich) für alle $z \in V\backslash\{x\}$ und der die Ungleichung (11) mit Gleichheit erfüllt.

- Falls $j \leq 1 + d(x) - \delta(G)$ und $|N[x]\backslash N[v]| \geq 1$, setze alle Knoten aus $N[x]$ in die ersten $1 + d(x) - \delta(G)$ Taschen, $k_{j,u} = 1$ für ein $u \in N[x]\backslash N[v]$ und $y_{h,x} = 1$, $y_{i,v} = 0$. Ungleichung (11) wird dann mit Gleichheit erfüllt.
- Falls $j \leq 1 + d(x) - \delta(G)$ und $|N[x]\backslash N[v]| = 0$, dann ist laut Voraussetzung $i \geq |N[x] \cup N[v]| - \delta(G) + 1$. Wir setzen alle Knoten u aus $N[x]$ in die ersten $1 + d(x) - \delta(G)$ Taschen (d.h. $k_{j,w} = 1$ für ein $w \in N[x] \subseteq N[v]$), alle Knoten aus $N[v]$ in die ersten $i - 1$ Taschen und $y_{h,x} = 1$, $y_{i,v} = 1$. Somit ist Ungleichung (11) mit Gleichheit erfüllt.
- Falls $j \geq 2 + d(x) - \delta(G)$ und $|V\backslash(N[v] \cup N[x])| \geq 1$, setzen wir alle Knoten aus $N[x]$ in die ersten $1 + d(x) - \delta(G)$ Taschen, $k_{j,u} = 1$ für ein $u \in V\backslash(N[x] \cup N[v])$ und $y_{h,x} = 1$, $y_{i,v} = 0$. Somit ist Ungleichung (11) wieder mit Gleichheit erfüllt.
- Falls $j \geq 2 + d(x) - \delta(G)$ und $|V\backslash(N[v] \cup N[x])| = 0$, dann ist laut Voraussetzung $i \geq |N[x] \cup N[v]| - \delta(G) + 1 = n - \delta(G) + 1$. So ein i existiert aber nicht. Somit brauchen wir diesen Fall auch nicht weiter betrachten.

Dies können wir nacheinander für alle $x \in V$ und alle $h \geq 2 + d(x) - \delta(G)$ machen und erhalten
$$\sum_{x \in V} \Big(n - \delta(G) - (2 + d(x) - \delta(G)) + 1 \Big) = \sum_{x \in V} \Big(n - d(x) - 1 \Big) = n^2 - 2 \cdot m - n$$
affin unabhängige Punkte.

(c) Setze $c = n$, $y_{i,v} = 1$, $y_{i-1,v} = 0$ und $y_{h,x} = 0$ für alle $x \in V\backslash\{v\}$ und alle $h \in I_x$. Wir haben $d(v) + 1$ Möglichkeiten um $k_{j,w} = 1$ für ein $w \in N[v]$ zu setzen. Eine dieser Möglichkeiten wurde schon in (b) berücksichtigt. Also bleiben noch
$$d(v)$$
affin unabhängige Punkte.

(d) In allen Punkten davor wurde $c = n$ gesetzt. Den letzten affin unabhängigen Punkt konstruieren wir, indem wir einen Punkt aus (b) oder (c) nehmen und $c = n - 1$ setzen.

Insgesamt erhalten wir

$$\begin{aligned}&(n-1)\cdot(n-\delta(G)-1)-(n-1)-(n-\delta(G)-2)+1\\ &+(n-\delta(G)-1)+(n-d(v)-3)+(n^2-2\cdot m-n)+d(v)+1\\ =&2\cdot n^2-\delta(G)\cdot n-3\cdot n-2\cdot m+2+\delta(G)\end{aligned}$$

affin unabhängige Punkte und damit ist der Satz bewiesen. □

Analog zu Ungleichung (5.6) können wir die Ungleichungen (5.8) und (5.10) verallgemeinern. Dabei müssen wir im Gegensatz zu (5.6) zwei Fälle unterscheiden: Nämlich $1 \in J$ und $1 \notin J$.

(5.12) Gültige Ungleichung (Klasse B)
Für alle Knoten $v \in V$ und für alle Taschen $i \in I$ mit $i \geq 2 + d(v) - \delta(G)$ gilt:

$$y_{i,v} \leq \sum_{j\in J}\sum_{w\in S} k_{j,w} \qquad \forall\, S \subseteq N[v],\ \forall\, J \subseteq \{1,\ldots,i-1\} \text{ mit } |S|,\ |J| \geq 1$$
$$\text{und } |S|+|J| = \begin{cases} i & \text{, falls } 1 \in J \\ i+\delta(G) & \text{, falls } 1 \notin J. \end{cases}$$ ◇

Beweis
Der Beweis erfolgt analog zum Beweis von Ungleichung (5.6) mit dem Unterschied, dass wir unterscheiden müssen ob $1 \in J$ oder $1 \notin J$. Da die erste Tasche $\delta(G)$ Knoten mehr enthält als alle anderen, muss $|S| + |J|$ um genau $\delta(G)$ größer sein falls $1 \notin J$ ist, als wenn $1 \in J$. □

5.3 Eine weitere Klasse gültiger Ungleichungen

Ungleichung (5.6) entstand aus (5.1) und (5.4). Ungleichung (5.12) entstand aus (5.8) und (5.10). Wir können aber auch (5.4) und (5.10) kombinieren. Das sieht dann folgendermaßen aus: Falls $i = 3 + d(v) - \delta(G)$ und $y_{i,v} = 1$, dann darf höchstens ein Knoten aus $V \backslash N[v]$ in den ersten $i - 1 = 2 + d(v) - \delta(G)$ Taschen sein. Und falls das der Fall ist, so müssen alle übrigen Knoten aus $V \backslash N[v]$ in den restlichen Taschen (von i bis $n - \delta(G)$) sein. Insgesamt also:

(5.13) Gültige Ungleichung

$$y_{i,v} \leq \sum_{w \in (N[v] \cup \{x\})} k_{j,w} + \sum_{h=i}^{n-\delta(G)} k_{h,x} \qquad \forall v \in V,\ \forall x \in V \backslash N[v],\ \forall i,j \in I, \text{ mit } i = 3 + d(v) - \delta(G) \text{ und } 1 \leq j < i. \quad \diamond$$

Beweis
Falls $y_{i,v} = 0$, ist die Ungleichung trivialerweise erfüllt.

Sei jetzt $y_{i,v} = 1$. Angenommen $\sum_{w \in N[v]} k_{j,w} + \sum_{h=i}^{n-\delta(G)} k_{h,x} = 0$. Wir zeigen, dass in diesem Fall $k_{j,x} = 1$ gilt. Da $y_{i,v} = 1$, müssen alle $d(v) + 1$ Knoten aus $N[v]$ eine Anfangstasche mit Index kleiner als i haben. Da $i = 3 + d(v) - \delta(G)$, existiert genau ein Knoten aus $V \backslash N[v]$ mit Anfangstaschenindex kleiner als i. Da einerseits $\sum_{w \in N[v]} k_{j,w} = 0$ und andererseits $\sum_{h=i}^{n-\delta(G)} k_{h,x} = 0$ ist, ist dieses $x \in V \backslash N[v]$ gerade dieser Knoten mit $k_{j,x} = 1$. Somit ist die rechte Seite immer größer gleich 1, falls $y_{i,v} = 1$. □

Um dieses Kapitel abzuschließen, kommt noch eine weitere Ungleichung hinzu, die die Anzahl der Knoten in der ersten Tasche auf eine etwas andere Art ausnutzt und somit Ungleichung (5.10) für den Fall $j = 1$ verschärft.

(5.14) Gültige Ungleichung

$$(\delta(G) + 1 - j) \cdot y_{i,v} \leq \sum_{w \in N[v]} k_{1,w} \qquad \forall v \in V,\ \forall j \in \{0, \ldots, \delta(G)\},\ \forall i \in I, \text{ mit } i = j + 2 + d(v) - \delta(G). \quad \diamond$$

Beweis
Falls $y_{i,v} = 0$, ist die Ungleichung trivialerweise gültig. Sei jetzt $y_{i,v} = 1$:

- Falls $j = 0$ ist $i = 2 + d(v) - \delta(G)$, und nur Knoten aus $N[v]$ können in den ersten $i - 1$ Taschen sein.
- Falls $j = 1$, ist genau ein Knoten aus $V \backslash N[v]$ in den ersten $i - 1$ Taschen.
- u.s.w.

Somit ist die Anzahl der Knoten aus $V\backslash N[v]$, die in den ersten $i-1$ Taschen sind und wo gleichzeitig $y_{i,v} = 1$ gilt, gegeben durch j. Insbesondere können maximal j Knoten aus $V\backslash N[v]$ in der ersten Tasche sein. D.h. es sind mindestens $\delta(G) + 1 - j$ Knoten aus $N[v]$ in der ersten Tasche und somit ist die Ungleichung gültig. □

5.4 Beispiele

Wir wenden die eben hergeleiteten Ungleichungen an dem Graphen G_2 aus Abbildung 11 an:

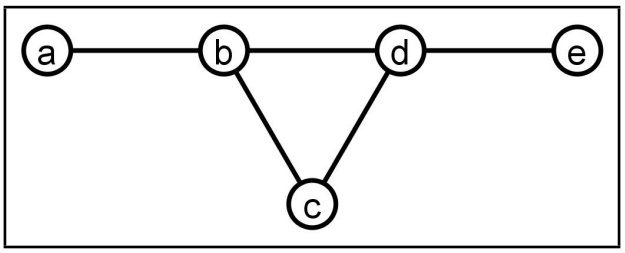

Abbildung 11: Der Graph G_2

- Ungleichung (5.6) für den Knoten a und den Index $i = 2$: Wir wählen ein $S \subseteq V\backslash N[a] = \{c, d, e\}$ und ein $J \subseteq \{2, 3, 4\}$ mit $|S| + |J| = n - d(a) = 5 - 1 = 4$.
  - Sei $S = \{d\}$ und $J = \{2, 3, 4\}$. Dann gilt:

    $$y_{2,a} \leq \sum_{j \in J} \sum_{w \in S} k_{j,w} = k_{2,d} + k_{3,d} + k_{4,d}.$$

    Diese Ungleichung hätten wir auch mit (5.4) herleiten können und ist wegen Satz (5.5) Facetten- definierend.
  - Sei $S = \{c, e\}$ und $J = \{2, 4\}$. Dann gilt:

    $$y_{2,a} \leq \sum_{j \in J} \sum_{w \in S} k_{j,w} = k_{2,c} + k_{2,e} + k_{4,c} + k_{4,e}.$$

    Die letzte Ungleichung definiert auch eine Facette[2] des Polyeders.

[2]Facetten können mit Hilfe der Software PORTA (**PO**lyhedron **R**epresentation **T**ransformation **A**lgorithm) von T. Christof und A. Löbel [10] bestimmt werden.

- Ungleichung (5.12) für den Knoten a und den Index $i = 2$: Wir wählen ein $S \subseteq N[a] = \{a, b\}$ und ein $J \subseteq \{1\}$ mit $|S| + |J| = i = 2$. Sei $S = \{a\}$ und $J = \{1\}$. Dann gilt:

$$y_{2,a} \leq \sum_{j \in J} \sum_{w \in S} k_{j,w} = k_{1,a}.$$

  Diese Ungleichung definiert laut Satz (5.9) eine Facette.

- Ungleichung (5.13) für den Knoten e: Dann ist $i = 3 + d(e) - d(G) = 3$. Wir wählen $j = 2$ und ein $x \in V \backslash N[e] = \{a, b, c\}$. Sei $x = b$. Dann gilt:

$$y_{3,e} \leq \sum_{w \in (N[e] \cup \{b\})} k_{2,w} + \sum_{h=3}^{4} k_{h,b} = k_{2,d} + k_{2,e} + k_{2,b} + k_{3,b} + k_{4,b}.$$

  Auch diese Ungleichung ist Facetten- definierend.

§6 Gültige Ungleichungen zweiter Ordnung

Alle Ungleichungen, die in Kapitel 5 hergeleitet wurden, enthalten genau eine y- Variable. Wir nennen sie deswegen Ungleichungen erster Ordnung. Diese berücksichtigen allerdings den Einfluss einer zweiten y- Variable nicht. Im Folgenden studieren wir diese Verbindung genauer und leiten auf diesem Weg starke gültige Ungleichungen mit zwei y- Variablen her: die *gültigen Ungleichungen zweiter Ordnung*.

6.1 Liften von Ungleichungen erster Ordnung zu Ungleichungen zweiter Ordnung

Beim Liften von Ungleichungen werden Variablen, die nicht in der Ungleichung sind, zur Ungleichung hinzugefügt, um sie so zu verschärfen. Dabei erhalten die Koeffizenten der Variablen einen Wert ungleich Null, so dass die Ungleichung gerade noch gültig bleibt.

Dazu betrachten wir die Ungleichungen erster Ordnung, die nicht die Facetten- Eigenschaft haben, genauer an zwei Beispielen:

(6.1) Beispiel

Wir wenden Ungleichung (5.8) auf den Graphen G_3 aus Abbildung 12 an und erhalten unter anderem:

$$y_{3,a} \leq k_{1,b} + k_{2,b} \quad (12)$$
$$y_{3,d} \leq k_{1,e} + k_{2,e} \quad (13)$$
$$y_{3,a} \leq k_{1,c} + k_{2,c} \quad (14)$$
$$y_{3,d} \leq k_{1,c} + k_{2,c} \quad (15)$$

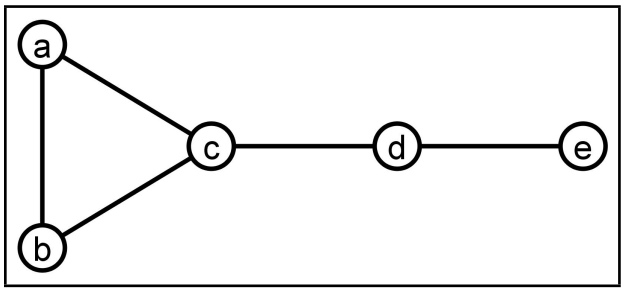

Abbildung 12: Der Graph G_3

Die ersten beiden definieren laut Satz (5.9) Facetten, die anderen beiden jedoch nicht. Wir wissen, dass $y_{3,a} + y_{3,d} \leq 1$, da in den ersten beiden Taschen nur $\delta(G) + 2 = 3$ verschiedene Knoten hineinpassen und $|N[a] \cup N[d]| = 5 > 3$. Also können wir Ungleichungen (14) und (15) liften und erhalten:

$$y_{3,a} + y_{3,d} \leq k_{1,c} + k_{2,c} \quad (16)$$

Das Liften ist nicht eindeutig, denn wir hätten Ungleichung (15) auch zu

$$y_{3,b} + y_{3,d} \leq k_{1,c} + k_{2,c} \tag{17}$$

liften können. Die beiden Ungleichungen (16) und (17) definieren Facetten des Polyeders $P_{GLP}(G_3)$. ◇

Das zweite Beispiel zeigt den Fall, wo Ungleichung (5.1) keine Facette definiert:

(6.2) Beispiel
Laut Ungleichung (5.1) gilt für den Graphen G_3 aus Abbildung 12 unter anderem:

$$y_{2,e} \leq k_{2,a} + k_{2,b} + k_{2,c}$$
$$y_{3,e} \leq k_{3,a} + k_{3,b} + k_{3,c}$$

Laut Satz (5.2) ist nur die zweite Ungleichung Facetten- definierend. Weiter gilt laut Ungleichung (5.10):

$$y_{3,a} \leq k_{2,a} + k_{2,b} + k_{2,c}.$$

Und da $|N[a] \cup N[e]| = 5 \geq 3 = \delta(G) + 2$ ist $y_{2,e} + y_{3,a} \leq 1$ und somit gilt:

$$y_{2,e} + y_{3,a} \leq k_{2,a} + k_{2,b} + k_{2,c}.$$

Diese letzte Ungleichung definiert eine Facette. ◇

6.2 Verallgemeinerung

Das eben vorgestellte Prinzip wird jetzt verallgemeinert. Dafür definieren wir:

(6.3) Definition
Für zwei Knoten $v, w \in V$ definieren wir:

$$\sigma_0(v) := 1 + d(v) - \delta(G)$$
$$\sigma_1(v, w) := |N[v] \cup N[w]| - \delta(G).$$ ◇

Die Bedeutung der Definition ist klar:

$$\begin{aligned} y_{i,v} &= 0 \quad \text{für alle } v \in V \text{ und } i = 1, \ldots, \sigma_0(v) \\ y_{i,v} + y_{j,w} &\leq 1 \quad \text{für alle } v, w \in V,\ v \neq w,\ i \in I_v,\ j \in I_w \text{ mit } i, j \leq \sigma_1(v,w), \end{aligned}$$

d.h. $\sigma_1(v,w)$ ist der größtmögliche Index, so dass $y_{\sigma_1(v,w),v} + y_{\sigma_1(v,w),w} \leq 1$ gilt.

Diese Definition ist Grundlage des nun folgenden Satzes:

(6.4) Gültige Ungleichung
Für zwei beliebige Knoten $v, w \in V$ mit $v \neq w$ und $i_v, i_w \in I$ mit $i_v, i_w \leq \sigma_1(v,w)$ gilt:

$$y_{i_v,v} + y_{i_w,w} \leq \underbrace{\sum_{j \in J_v \setminus J_w} \sum_{x \in S_v} k_{j,x} + \sum_{j \in J_w \setminus J_v} \sum_{x \in S_w} k_{j,x} + \sum_{j \in J_v \cap J_w} \sum_{x \in S_v \cup S_w} k_{j,x}}_{=: g(J_v, S_v, J_w, S_w)}$$

wobei J_v, S_v und J_w, S_w beliebig gemäß (5.6) und/oder (5.12) gewählt werden können. ◇

Beweis
Laut Ungleichung (5.6) und/oder (5.12) gilt:

$$\begin{aligned} y_{i_v,v} &\leq \sum_{j \in J_v} \sum_{x \in S_v} k_{j,x} \leq g(J_v, S_v, J_w, S_w) \\ y_{i_w,w} &\leq \sum_{j \in J_w} \sum_{x \in S_w} k_{j,x} \leq g(J_v, S_v, J_w, S_w) \end{aligned}$$

Da $i_v, i_w \leq \sigma_1(v,w)$ gilt $y_{i_v,v} + y_{i_w,w} \leq 1$ und somit auch

$$y_{i_v,v} + y_{i_w,w} \leq g(J_v, S_v, J_w, S_w). \qquad \square$$

Der letzte Satz ist nur eine Verbesserung von (5.6) oder (5.12), falls $|J_v \cap J_w| \geq 1$ und $|S_v \cap S_w| \geq 1$.

(6.5) Beispiel
Wir wenden Satz (6.4) für die Knoten a und d des Graphen G_4 aus Abbildung 13 an:

$$\sigma_1(a,d) = |N[a] \cup N[d]| - \delta(G) = 4 - 1 = 3$$

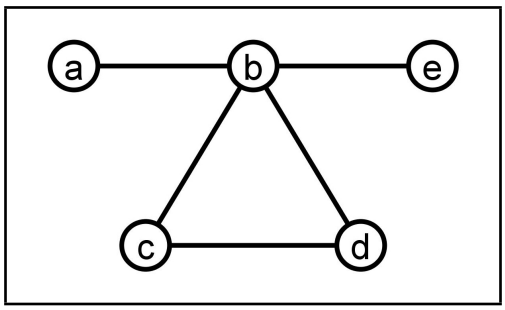

Abbildung 13: Der Graph G_4

- Wir wählen $i_a = 2$, $i_d = 3$ und J_a, S_a gemäß Ungleichung (5.6):

  $$J_a = \{2\} \subseteq \{2,3,4\} \qquad \text{und} \quad S_a = \{c,d,e\} \subseteq \{c,d,e\} = V \backslash N[a].$$

  Wir wählen J_d, S_d gemäß Ungleichung (5.12):

  $$J_d = \{1,2\} \subseteq \{1,2\} \qquad \text{und} \quad S_d = \{d\} \subseteq \{b,c,d\} = N[d].$$

  Somit folgt aus Satz (6.4) die Gültigkeit von:

  $$y_{2,a} + y_{3,d} \leq g(J_a, S_a, J_d, S_d) = k_{1,d} + k_{2,c} + k_{2,d} + k_{2,e}.$$

  Diese letzte Ungleichung definiert eine Facette.

- Wir wählen $i_a = 3$, $i_d = 3$ und J_a, S_a und J_d, S_d gemäß Ungleichung (5.6):

  $$\begin{aligned} J_a &= \{3\} \subseteq \{3,4\} && \text{und} \quad S_a = \{c,d,e\} \subseteq \{c,d,e\} = V \backslash N[a], \\ J_d &= \{3,4\} \subseteq \{3,4\} && \text{und} \quad S_d = \{e\} \subseteq \{a,e\} = V \backslash N[d]. \end{aligned}$$

  Somit folgt aus Satz (6.4) die Gültigkeit von:

  $$y_{3,a} + y_{3,d} \leq g(J_a, S_a, J_d, S_d) = k_{3,c} + k_{3,d} + k_{3,e} + k_{4,e}.$$

  Diese letzte Ungleichung definiert auch eine Facette. ◇

Falls wir jetzt $i_w := \sigma_1(v,w) + 1$ setzen, dann könnte $y_{i_v,v} + y_{i_w,w} = 2$ sein und Satz (6.4) wäre nicht anwendbar. Ist dies der Fall, so muss w oder ein Nachbar von w in die i_v-te Tasche reinkommen:

(6.6) Gültige Ungleichungen
Es seien zwei beliebige verschiedene Knoten $v, w \in V$ mit $|N[w] \backslash N[v]| \geq 1$ und $i_w = \sigma_1(v,w) + 1 \leq n - \delta(G)$ gegeben.

(a) Dann gilt für $i_v = \sigma_0(v) + 1 \in I$, für alle $u \in (N[w] \backslash N[v])$ und für alle S_v, J_v und S_w, J_w gemäß (5.6) und/oder (5.12):

$$y_{i_v,v} + y_{i_w,w} \leq g(J_v, S_v, J_w, S_w) \; + \sum_{j=i_v}^{\sigma_1(v,w)} k_{j,u}$$

(b) Dann gilt für $i_v \in \{\sigma_0(v)+1, \ldots, \sigma_1(v,w)\}$, für alle $h \in \{i_v, \ldots, \sigma_1(v,w)\}$ und für alle S_v, J_v und S_w, J_w gemäß (5.6) und/oder (5.12):

$$y_{i_v,v} + y_{i_w,w} \leq g(J_v, S_v, J_w, S_w) \; + \sum_{u \in (N[w] \backslash N[v])} k_{h,u} \qquad \diamond$$

Beweis
Laut Ungleichung (5.6) und/oder (5.12) gilt:

$$y_{i_v,v} \leq \sum_{j \in J_v} \sum_{x \in S_v} k_{j,x} \leq g(J_v, S_v, J_w, S_w)$$
$$y_{i_w,w} \leq \sum_{j \in J_w} \sum_{x \in S_w} k_{j,x} \leq g(J_v, S_v, J_w, S_w)$$

Falls $y_{i_v,v} + y_{i_w,w} \leq 1$ gelten (a) und (b). Es sei jetzt

$$y_{i_v,v} = y_{i_w,w} = 1.$$

(a) Dann müssen alle Knoten aus $N[w] \backslash N[v]$ ihre Anfangstasche zwischen $i_v = \sigma_0(v) + 1$ und $\sigma_1(v,w)$ haben. Also

$$\sum_{j=\sigma_0(v)+1}^{\sigma_1(v,w)} k_{j,u} = 1.$$

Daraus folgt die Gültigkeit vom ersten Teil des Satzes.

(b) In den ersten $\sigma_1(v,w)$ Taschen sind nur Knoten aus $N[v] \cup N[w]$, wobei die Knoten aus $N[v]$ alle schon in den ersten $i_v - 1$ Taschen sind. Also dürfen zwischen i_v und $\sigma_1(v,w)$ nur Knoten aus $N[w] \backslash N[v]$ neu in die Taschen kommen. Somit muss

$$\sum_{u \in (N[w] \backslash N[v])} k_{h,u} = 1$$

gelten. Also folgt auch der zweite Teil des Satzes. □

Das Prinzip kristallisiert sich heraus: Falls wir eine Ungleichung mit zwei y- Variablen auf der linken Seite haben, brauchen wir auf der rechten Seite zwei Komponenten: Die erste wird gleich 1, sobald eine der y- Variablen den Wert 1 annimmt. Die zweite Komponente ist 0 und wird nur dann sicher gleich 1, wenn beide y- Variablen gleichzeitig den Wert 1 haben.

Wir versuchen jetzt mit Hilfe des eben beschriebenen Prinzips die Ungleichungen erster Ordnung, wie (5.1), (5.4) und (5.10), in Ungleichungen zweiter Ordnung überzuführen:

(6.7) Gültige Ungleichungen
Es seien zwei beliebige Knoten $v, w \in V$ mit $v \neq w$ und $\sigma_1(v,w) + 1 \leq n - \delta(G)$ gegeben.

(a) Für alle $i_v, i_w, j \in I$ mit $j \geq \max(i_v, i_w) \geq \sigma_1(v,w) + 1$ gilt:

$$y_{i_v,v} + y_{i_w,w} \leq g(J_v, S_v, J_w, S_w) \; + \sum_{u \in V \setminus (N[v] \cup N[w])} k_{j,u}$$

(b) Für alle $i_v, i_w \in I$ mit $\max(i_v, i_w) = \sigma_1(v,w) + 1$ und für alle $u \in V \setminus (N[v] \cup N[w])$ gilt:

$$y_{i_v,v} + y_{i_w,w} \leq g(J_v, S_v, J_w, S_w) \; + \sum_{j=\max(i_v,i_w)}^{n-\delta(G)} k_{j,u}$$

(c) Für alle $i_v, i_w, j \in I$ mit $\max(i_v, i_w) = \sigma_1(v,w) + 1$ und $j \leq \max(i_v, i_w) - 1$ gilt:

$$y_{i_v,v} + y_{i_w,w} \leq g(J_v, S_v, J_w, S_w) \; + \sum_{u \in (N[v] \cup N[w])} k_{j,u}$$

wobei S_v, J_v und S_w, J_w jeweils gemäß (5.6) und/oder (5.12) beliebig gewählt sind. ◇

Beweis
Laut Ungleichung (5.6) und/oder (5.12) gilt:

$$y_{i_v,v} \leq \sum_{j \in J_v} \sum_{x \in S_v} k_{j,x} \leq g(J_v, S_v, J_w, S_w)$$
$$y_{i_w,w} \leq \sum_{j \in J_w} \sum_{x \in S_w} k_{j,x} \leq g(J_v, S_v, J_w, S_w)$$

Falls $y_{i_v,v} + y_{i_w,w} \leq 1$, gelten also die drei Teile des Satzes. Es sei jetzt

$$y_{i_v,v} = y_{i_w,w} = 1.$$

Wir müssen zeigen, dass dann die zweite Komponente auf der rechten Seite in (a), (b) und (c) gleich eins ist.

(a) Da $y_{i_v,v} + y_{i_w,w} = 2$, dürfen nur Knoten $u \in V\backslash(N[w] \cup N[v])$ ihre Anfangstasche zwischen $\max(i_v, i_w)$ und $n - \delta(G)$ haben. Also gilt:

$$\sum_{u \in V\backslash(N[v] \cup N[w])} k_{j,u} = 1 \qquad \text{für alle } j \in \{\max(i_v, i_w), \ldots, n - \delta(G)\}.$$

(b) Jeder Knoten $u \in V\backslash(N[w] \cup N[v])$ darf nur in den Taschen mit Index größer gleich $\max(i_v, i_w)$ sein:

$$\sum_{j=\max(i_v,i_w)}^{n-\delta(G)} k_{j,u} = 1.$$

(c) In allen Taschen mit Index kleiner als $\max(i_v, i_w)$ sind nur Knoten aus $N[v] \cup N[w]$:

$$\sum_{u \in (N[v] \cup N[w])} k_{j,u} = 1 \qquad \text{für alle } j \in \{1, \ldots, \max(i_v, i_w) - 1\}.$$

Und damit ist der Satz bewiesen. □

6.3 Eine weitere gültige Ungleichung zweiter Ordnung

(6.8) Gültige Ungleichung
Für alle Knoten $v, w \in V$, $v \neq w$ mit $|N[v] \cup N[w]| \leq n - 2$ und für alle Indizes $i_v, i_w \in I$ mit $i_v \geq 2 + d(v) - \delta(G)$ und $i_w \geq 2 + d(w) - \delta(G)$ gilt:

$$y_{i_v,v} + y_{i_w,w} \leq \sum_{j \in J} \sum_{u \in S} k_{j,u}$$

$S = V\backslash(N[v] \cup N[w])$,
$\forall\, J \subseteq \{\max(i_v, i_w), \ldots, n - \delta(G)\}$ mit $|S|,\ |J| \geq 2$
und $|S| + |J| \geq n - \min(d(v), d(w))$ ◇

Beweis
Laut Satz (5.6) gilt:

$$y_{i_v,v} \leq \sum_{j\in J} \sum_{u\in S} k_{j,u} \text{ und } y_{i_w,w} \leq \sum_{j\in J} \sum_{u\in S} k_{j,u}.$$

Somit ist die Ungleichung gültig für $y_{i_v,v} + y_{i_w,w} \leq 1$.

Sei jetzt $y_{i_v,v} = y_{i_w,w} = 1$. Dann sind in allen Taschen mit Indizes größer oder gleich $\max(i_v, i_w)$ nur Knoten aus $S = V\backslash(N[v] \cup N[w])$. Und da $|J| \geq 2$ ist die rechte Seite auch größer oder gleich 2, und somit ist die Gültigkeit bewiesen. □

6.4 Beispiele

Wir betrachten den Graphen G_4 aus Abbildung 14:

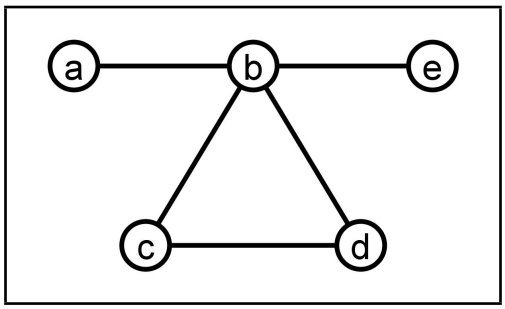

Abbildung 14: Der Graph G_4

- Wir wenden Ungleichung (6.6) für die Knoten a und e an. Dann ist $i_a = 2$ und $i_e = \sigma_1(a,e) + 1 = 3$. Wir wählen $S_a = \{c\} \subseteq \{c,d,e\} = V\backslash N[a]$ und $J_a = \{2,3,4\} \subseteq \{2,3,4\}$ gemäß Satz (5.6) und $S_e = \{a,c,d\} \subseteq \{a,c,d\} = V\backslash N[e]$ und $J_e = \{4\} \subseteq \{3,4\}$ gemäß Satz (5.6). Dann gilt:

$$y_{2,a} \overset{(5.6)}{\leq} \sum_{j\in J_a} \sum_{u\in S_a} k_{j,u} = k_{2,c} + k_{3,c} + k_{4,c}$$

$$y_{3,e} \overset{(5.6)}{\leq} \sum_{j\in J_e} \sum_{u\in S_e} k_{j,u} = k_{4,a} + k_{4,c} + k_{4,d}$$

$$g(J_a, S_a, J_e, S_e) = k_{2,c} + k_{3,c} + k_{4,c} + k_{4,a} + k_{4,d}$$

und schließlich gemäß (6.6)(a) bzw. (6.6)(b) für $e \in N[e]\backslash N[a]$:

$$y_{2,a} + y_{3,e} \leq g(J_v, S_v, J_w, S_w) \; + \; k_{2,e} \; = \; k_{2,c} + k_{3,c} + k_{4,c} + k_{4,a} + k_{4,d} + k_{2,e}.$$

Dies entspricht sogar einer Facetten- definierenden Ungleichung.

- Wir wenden Ungleichung (6.7) für die Knoten a und e mit $i_a = 3$ und $i_e = 3$ an. Dann gilt $\max(i_a, i_e) = \sigma_1(a, e) + 1 = 3$ und die Teile (a),(b),(c) des Satzes sind anwendbar. Wir wählen $S_a = \{b\} \subseteq \{a, b\} = N[a]$ und $J_a = \{1, 2\} \subseteq \{1, 2\}$ gemäß Satz (5.12) und $S_e = \{b\} \subseteq \{b, e\} = N[e]$ und $J_e = \{1, 2\} \subseteq \{1, 2\}$ gemäß Satz (5.12). Dann gilt:

$$y_{3,a} \overset{(5.12)}{\leq} \sum_{j \in J_a} \sum_{u \in S_a} k_{j,u} = k_{1,b} + k_{2,b}$$

$$y_{3,e} \overset{(5.12)}{\leq} \sum_{j \in J_e} \sum_{u \in S_e} k_{j,u} = k_{1,b} + k_{2,b}$$

$$g(J_a, S_a, J_e, S_e) = k_{1,b} + k_{2,b}$$

  - Dann gilt laut Ungleichung (6.7)(a) für $j = 3 \geq \max(i_a, i_e)$:

$$y_{3,a} + y_{3,e} \leq g(J_a, S_a, J_e, S_e) + \sum_{u \in V \setminus (N[a] \cup N[e])} k_{3,u} = k_{1,b} + k_{2,b} + k_{3,c} + k_{3,d}$$

  - für den Knoten $c \in V \setminus (N[a] \cup N[e])$ gemäß Ungleichung (6.7)(b):

$$y_{3,a} + y_{3,e} \leq g(J_a, S_a, J_e, S_e) \; + \sum_{j=3}^{4} k_{j,c} \; = \; k_{1,b} + k_{2,b} + k_{3,c} + k_{4,c}$$

  - und schließlich für $j = 2 \leq \max(i_a, i_e) - 1$ gemäß Ungleichung (6.7)(c):

$$y_{3,a} + y_{3,e} \leq g(J_a, S_a, J_e, S_e) + \sum_{u \in (N[a] \cup N[e])} k_{2,u} = k_{1,b} + k_{2,b} + k_{2,a} + k_{2,b} + k_{2,e}$$

  Diese drei Ungleichungen definieren keine Facetten, denn

$$y_{3,d} \leq k_{1,b} + k_{2,b}$$

  und da $|N[a] \cup N[d]| - \delta(G) + 1 \geq 3$ können $y_{3,a}$ und $y_{3,d}$ nicht gleichzeitig 1 sein. Analog für $y_{3,d}$ und $y_{3,e}$. Daraus folgt:

$$y_{3,a} + y_{3,d} \leq k_{1,b} + k_{2,b} \quad \text{und} \quad y_{3,e} + y_{3,d} \leq k_{1,b} + k_{2,b}.$$

  Und somit gilt auch:

$$y_{3,a} + y_{3,e} + y_{3,d} \leq k_{1,b} + k_{2,b} + k_{3,c} + k_{3,d}$$
$$y_{3,a} + y_{3,e} + y_{3,d} \leq k_{1,b} + k_{2,b} + k_{3,c} + k_{4,c}$$
$$y_{3,a} + y_{3,e} + y_{3,d} \leq k_{1,b} + k_{2,b} + k_{2,a} + k_{2,b} + k_{2,e}$$

  Die drei letzten Ungleichungen sind Facetten- definierend. Solche Ungleichungen können später mit Satz (7.9) gefunden werden.

§7 Gültige Ungleichungen mehrfacher Ordnung

Wir versuchen die Ungleichungen zweiter Ordnung zu verallgemeinern, doch zuerst benötigen wir noch eine weitere gültige Ungleichung:

7.1 Erste Ungleichung mehrfacher Ordnung

(7.1) Gültige Ungleichung

$$\sum_{i=2+d(v)-\delta(G)}^{2+d(v)} y_{i,v} \leq \sum_{w \in N[v]} k_{1,w} \qquad \forall v \in V. \qquad \diamond$$

Beweis
Sei $h := \sum_{i=2+d(v)-\delta(G)}^{2+d(v)} y_{i,v}$. Dann sind maximal $\delta(G) + 1 - h \geq 0$ verschiedene Knoten aus $V \backslash N[v]$ in den ersten $1 + d(v) - \delta(G)$ Taschen. D.h. es können maximal $\delta(G) + 1 - h$ Knoten aus $V \backslash N[v]$ in der ersten Tasche sein, bzw. es sind mindestens h Knoten aus $N[v]$ in der ersten Tasche. Somit gilt:

$$\sum_{i=2+d(v)-\delta(G)}^{2+d(v)} y_{i,v} = h \leq \sum_{w \in N[v]} k_{1,w}. \qquad \square$$

(7.2) Gültige Ungleichung
Ungleichung (7.1) definiert eine Facette für alle Knoten $v \in V$ mit

$$d(v) \leq n - \delta(G) - 3. \qquad \diamond$$

Beweis
Wir benötigen $2 \cdot n^2 - \delta(G) \cdot n - 3 \cdot n - 2 \cdot m + 2 + \delta(G)$ affin unabhängige Punkte, die die Ungleichung

$$\sum_{i=2+d(v)-\delta(G)}^{2+d(v)} y_{i,v} \leq \sum_{w \in N[v]} k_{1,w} \tag{18}$$

für ein festes $v \in V$ mit Gleichheit erfüllen.

(a) Setze $c = n$ und $y_{j,x} = 0$ für alle $x \in V$ und alle $j \in I_x$. Da $\sum_{i=2+d(v)-\delta(G)}^{2+d(v)} y_{i,v} = 0$, muss auch $k_{1,w} = 0$ für alle $w \in N[v]$ gelten. Dafür gibt es

$$n \cdot (n - \delta(G)) - (d(v) + 1) - (n - \delta(G) - 1) - n + 1$$

affin unabhängige Punkte.

(b) Wir konstruieren für alle $x \in V$ und alle $j \in I_x$ jeweils einen Punkt mit $y_{j,x} = 1$, $y_{j-1,x} = 0$, $c = n$, $y_{n-\delta(G),u} = 0$ für alle $u \neq x$ (falls möglich) und der die Ungleichung (18) mit Gleichheit erfüllt:

- Falls $|N[x] \backslash N[v]| \geq \delta(G) + 1$, dann kommen $\delta(G) + 1$ Knoten aus $N[x] \backslash N[v]$ in die erste Tasche und die übrigen Knoten werden so verteilt, dass $y_{j,x} = 1$ gelten kann. Ungleichung (18) ist dann mit Gleichheit erfüllt.
- Falls $|N[x] \backslash N[v]| = h$ für ein $h \in \{0, \ldots, \delta(G)\}$. Dann ist $|N[v] \cup N[x]| = 1 + d(v) + h$ und wir setzen die h Knoten aus $N[x] \backslash N[v]$ in die erste Tasche. Die Knoten aus $N[x] \cap N[v]$ kommen in die ersten $1 + d(x) - \delta(G)$ Taschen. Somit sind in den ersten $1 + d(x) - \delta(G)$ Taschen nur Knoten aus $N[x]$ und in der ersten Tasche sind genau $\delta(G) + 1 - h$ Knoten aus $N[v]$. Die Knoten aus $N[v] \backslash N[x]$ setzen wir in die ersten $1 + d(v) - \delta(G) + h$ Taschen und $y_{1+d(v)-\delta(G)+h,v} = 0$, $y_{2+d(v)-\delta(G)+h,v} = 1$ und $y_{j,x} = 1$. Somit ist Ungleichung (18) mit Gleichheit erfüllt.

Dies können wir nacheinander für alle $x \in V$ und alle $j \geq 2 + d(x) - \delta(G)$ machen und erhalten

$$\sum_{x \in V} \Big(n - \delta(G) - (2 + d(x) - \delta(G)) + 1 \Big) = \sum_{x \in V} \Big(n - d(x) - 1 \Big) = n^2 - 2 \cdot m - n$$

affin unabhängige Punkte.

(c) Setze $c = n$, $y_{2+d(v),v} = 1$, $y_{1+d(v),v} = 0$ und $y_{j,x} = 0$ für alle $x \in V \backslash \{v\}$ und alle $j \in I_x$. Dann muss $\sum_{w \in N[v]} k_{1,w} = 1$ sein. Wir haben also $d(v) + 1$ Möglichkeiten, um $k_{1,w} = 1$ für ein $w \in N[v]$ zu setzen. Eine dieser Möglichkeiten ist affin abhängig zu einem Punkt aus (b). Also bleiben noch

$$d(v)$$

neue affin unabhängige Punkte.

(d) In allen Punkten davor wurde $c = n$ gesetzt. Den letzten affin unabhängigen Punkt konstruieren wir, indem wir einen Punkt aus (b) oder (c) nehmen und $c = n - 1$ setzen.

Insgesamt erhalten wir

$$\begin{aligned} & n \cdot (n - \delta(G)) - (d(v) + 1) - (n - \delta(G) - 1) - n + 1 + (n^2 - 2 \cdot m - n) + d(v) + 1 \\ =& 2 \cdot n^2 - \delta(G) \cdot n - 3 \cdot n - 2 \cdot m + 2 + \delta(G) \end{aligned}$$

affin unabhängige Punkte und damit ist der Satz bewiesen. □

7.2 Neue Notation

Mit Hilfe einer neuen Notation können die Ungleichungen, die in den letzten Kapiteln hergeleitet wurden, verallgemeinert werden:

(7.3) Bezeichnungen
Die Funktion

$$\begin{aligned} f : 2^{I \times V} &\longrightarrow \mathbb{N}^{I \times V} \times \mathbb{N}^{I \times V} \times \ldots \times \mathbb{N}^{I \times V} \\ T &\longmapsto f(T) \end{aligned}$$

ordnet für eine Teilmenge $T \subseteq (I \times V)$, der Summe $\sum_{(i,v) \in T} y_{i,v}$ die Menge aller bisher bekannten gültigen Ungleichungen zu. ◇

D.h. für eine Teilmenge $T \subseteq (I \times V)$ und ein $\rho \in f(T)$ gilt:

$$\sum_{(i,v) \in T} y_{i,v} \leq \sum_{(i,v) \in \rho} k_{i,v}.$$

Die Elemente ρ aus $f(T)$ sind *Multimengen*. Der Unterschied von Multimengen gegenüber den *gewöhnlichen Mengen* besteht darin, dass die Elemente der Multimenge mehrfach vorkommen können. Deswegen bezeichnet $\rho(i, v)$ die Anzahl, wie oft das Element (i, v) in der Multimenge ρ vorkommt.

Wir definieren ein paar Operationen auf Multimengen:

$$(\rho_1 + \rho_2)(i, v) = \rho_1(i, v) + \rho_2(i, v).$$

Die Vereinigung zweier Multimengen ist die kleinste Multimenge, die die beiden enthält:

$$\begin{aligned} (\rho_1 \cup \rho_2)(i, v) &= \max\big(\rho_1(i, v), \rho_2(i, v)\big) \\ (\rho_1 \cap \rho_2)(i, v) &= \min\big(\rho_1(i, v), \rho_2(i, v)\big). \end{aligned}$$

Es folgt ein kleines Beispiel, um den Leser mit der neuen Notation vertraut zu machen:

(7.4) Beispiel

Wir betrachten den Graphen G_1 aus Abbildung 15. Für $y_{2,d}$ seien folgende gültige Ungleichungen bekannt:

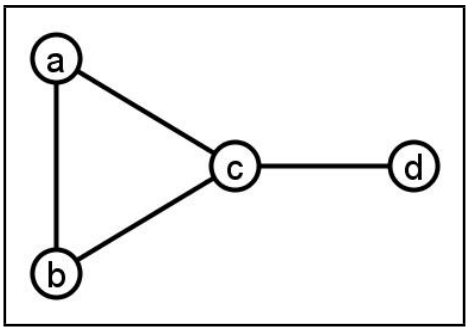

Abbildung 15: Der Graph G_1

$$y_{2,d} \overset{(5.1)}{\leq} k_{2,a} + k_{2,b} \tag{19}$$

$$y_{2,d} \overset{(5.8)}{\leq} k_{1,d} \tag{20}$$

$$y_{2,d} \overset{(5.4)}{\leq} k_{2,b} + k_{3,b} \tag{21}$$

$$y_{2,d} \overset{(5.4)}{\leq} k_{2,a} + k_{3,a} \tag{22}$$

Somit gilt für $T = \{(2,d)\}$:

$$f(T) = \Big\{\{(2,a),(2,b)\},\{(1,d)\},\{(2,b),(3,b)\},\{(2,a),(3,a)\}\Big\}$$

Für $\rho = \{(2,a),(2,b)\} \in f(T)$ gilt: $\rho(2,a) = \rho(2,b) = 1$ und $\rho(i,v) = 0$ für alle $(i,v) \in (I \times V)\backslash\{(2,a),(2,b)\}$. Wir erhalten

$$y_{2,d} = \sum_{(i,v)\in T} y_{i,v} \leq \sum_{(i,v)\in\rho} k_{i,v} = k_{2,a} + k_{2,b}$$

und dies entspricht Ungleichung (19). ◇

(7.5) Lemma

Für jede Teilmenge $T = T_1 \cup T_2 \subseteq (I \times V)$ gilt:

$$\rho_1 \in f(T_1),\ \rho_2 \in f(T_2) \Longrightarrow (\rho_1 + \rho_2) \in f(T).$$ ◇

Beweis

Der Beweis folgt aus der Definition:

$$\sum_{(i,v)\in T} y_{i,v} \leq \sum_{(i,v)\in T_1} y_{i,v} + \sum_{(i,v)\in T_2} y_{i,v} \overset{(7.3)}{\leq} \sum_{(i,v)\in\rho_1} k_{i,v} + \sum_{(i,v)\in\rho_2} k_{i,v}$$ □

7.3 Die Ungleichungen mehrfacher Ordnung

(7.6) Definition (Bedingungen)
Bedingung A
Eine Teilmenge $\emptyset \neq S = \{(i_1, w_1), \ldots, (i_h, w_h)\} \subseteq (I \times V)$ mit $i_1 \leq i_2 \leq \ldots \leq i_h$ erfüllt Bedingung A genau dann, wenn

$$i_g \geq |\bigcup_{j=1}^{g} N[w_j]| + 1 - \delta(G) \quad \text{für alle } g \in \{1, \ldots, h\}.$$

Bedingung B
Eine Teilmenge $S = \{(i_1, w_1), \ldots, (i_h, w_h)\} \subseteq (I \times V)$ mit $i_1 \leq i_2 \leq \ldots \leq i_h$ erfüllt Bedingung B genau dann, wenn S Bedingung A erfüllt und zusätzlich gilt:

$$i_h = |\bigcup_{j=1}^{h} N[w_j]| + 1 - \delta(G).$$

◇

Bedingung A bedeutet nur, dass alle durch S gegebenen y- Variablen gleichzeitig den Wert 1 annehmen können. Bedingung B fordert, dass zusätzlich zu A keine Knoten aus $V \setminus \left(\bigcup_{j=1}^{n} N[w_j]\right)$ in den ersten $i_h - 1$ Taschen sind, falls alle y- Variablen den Wert 1 annehmen.

(7.7) Gültige Ungleichung
Sei eine Menge $T = \{(i_1, w_1), \ldots, (i_h, w_h)\} \subseteq (I \times V)$ mit $h \geq 2$ gegeben, die die Bedingung A nicht erfüllt und sei

$$\Phi(T) := \left\{ \emptyset \neq S \subset T \;\middle|\; S \text{ erfüllt Bedingung A} \right\}.$$

Weiter sei für jedes $S \in \Phi(T)$ ein $\rho_S \in f(S)$ gegeben. Dann gilt:

$$\sum_{(i,v) \in T} y_{i,v} \leq \sum_{(i,v) \in \bigcup_{S \in \Phi(T)} \rho_S} k_{i,v}.$$

◇

Beweis
Für alle $S \in \Phi(T)$ gilt nach Konstruktion:

$$\sum_{(i,v) \in S} y_{i,v} \leq \sum_{(i,v) \in \rho_S} k_{i,v}.$$

Sei $T_0 := \{(i,v) \in T \mid y_{i,v} = 0\}$ und $T_1 := \{(i,v) \in T \mid y_{i,v} = 1\}$. Dann ist $T_1 \in \Phi(T)$, da $T_1 \subset T$ und T_1 erfüllt Bedingung A. Also gilt:

$$\sum_{(i,v)\in T} y_{i,v} = \underbrace{\sum_{(i,v)\in T_0} y_{i,v}}_{=0} + \sum_{(i,v)\in T_1} y_{i,v} \leq \sum_{(i,v)\in \rho_{(T_1)}} k_{i,v} \leq \sum_{(i,v)\in \bigcup_{S\in\Phi(T)} \rho_S} k_{i,v}. \qquad \square$$

(7.8) Beispiel
Wir betrachten den Graphen G_5 aus Abbildung 16 und wenden Ungleichung (7.7) für $T = \{(2,a),(2,b),(3,d)\}$ an: Da $|N[1] \cup N[2] \cup N[4]| = 4 > 3$ erfüllt T Bedingung A und somit ist der Satz anwendbar. Weiter ist:

$$\Phi(T) = \left\{ \underbrace{\{(2,a)\}}_{=t_1}, \underbrace{\{(2,b)\}}_{=t_2}, \underbrace{\{(3,d)\}}_{=t_3}, \underbrace{\{(2,a),(3,d)\}}_{=t_4}, \underbrace{\{(2,b),(3,d)\}}_{=t_5} \right\}.$$

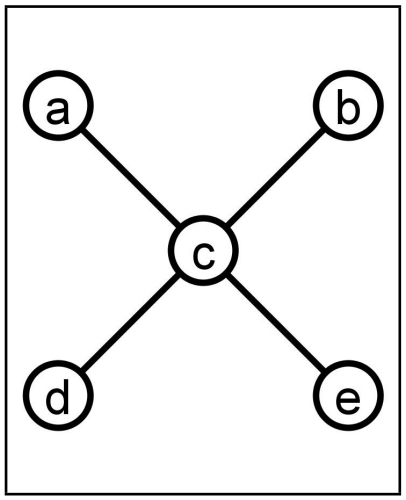

Abbildung 16: Der Graph G_5

Dann gelten folgende Ungleichungen erster Ordnung:

$$\begin{aligned} y_{2,a} &\overset{(5.6)}{\leq} k_{2,e} + k_{3,e} + k_{4,e} \\ y_{2,b} &\overset{(5.6)}{\leq} k_{2,e} + k_{3,e} + k_{4,e} \\ y_{3,d} &\overset{(5.6)}{\leq} k_{3,e} + k_{4,e} + k_{3,b} + k_{4,b} \end{aligned}$$

Mit Hilfe von Ungleichung (6.6) können wir folgende Ungleichungen zweiter Ordnung erzeugen:

$$\begin{aligned} y_{2,a} + y_{3,d} &\leq k_{2,e} + k_{3,e} + k_{4,e} + k_{3,b} + k_{4,b} + k_{2,d} \\ y_{2,b} + y_{3,d} &\leq k_{2,e} + k_{3,e} + k_{4,e} + k_{3,b} + k_{4,b} + k_{2,d} \end{aligned}$$

Somit ist:

$$\begin{aligned} \rho_{t_1} &= \Big\{(2,e),(3,e),(4,e)\Big\} \\ \rho_{t_2} &= \Big\{(2,e),(3,e),(4,e)\Big\} \\ \rho_{t_3} &= \Big\{(3,e),(4,e),(3,b),(4,b)\Big\} \\ \rho_{t_4} &= \Big\{(2,e),(3,e),(4,e),(3,b),(4,b),(2,d)\Big\} \\ \rho_{t_5} &= \Big\{(2,e),(3,e),(4,e),(3,b),(4,b),(2,d)\Big\} \end{aligned}$$

Und schließlich erhalten wir eine gültige Ungleichung für T:

$$y_{2,a} + y_{2,b} + y_{3,d} = \sum_{(i,v)\in T} y_{i,v} \leq \sum_{(i,v)\in\bigcup_{S\in\Phi(T)}\rho_S} k_{i,v} \leq k_{2,e} + k_{3,e} + k_{4,e} + k_{3,b} + k_{4,b} + k_{2,d}.$$

Diese Ungleichung definiert sogar eine Facette. ◇

Analog zu Ungleichung (6.7), können wir auch Ungleichungen mehrfacher Ordnung konstruieren:

(7.9) Gültige Ungleichungen
Sei die Menge $T = \{(i_1, w_1), \ldots, (i_h, w_h)\} \subseteq (I \times V)$ mit $h \geq 2$ gegeben, die die Bedingung A erfüllt. Weiter sei für jedes $j \in \{1, \ldots, h\}$ ein $\rho_j \in f(T\backslash\{(i_j, w_j)\})$ gegeben.

(a) Dann gilt für alle $g \in \{\max(i_1, \ldots, i_h), \ldots, n - \delta(G)\}$:

$$\sum_{(i,v)\in T} y_{i,v} \leq \sum_{(i,v)\in\bigcup_{j=1}^{h}\rho_j} k_{i,v} + \sum_{u\in V\backslash\left(\bigcup_{j=1}^{h} N[w_j]\right)} k_{g,u}.$$

Falls T zusätzlich Bedingung B erfüllt:

(b) Dann gilt für alle $u \in V\backslash\left(\bigcup_{j=1}^{h} N[w_j]\right)$:

$$\sum_{(i,v)\in T} y_{i,v} \leq \sum_{(i,v)\in\bigcup_{j=1}^{h}\rho_j} k_{i,v} + \sum_{g=\max(i_1,\ldots,i_h)}^{n-\delta(G)} k_{g,u}.$$

(c) Und für alle $g \in \{1, \ldots, \max(i_1, \ldots, i_h) - 1\}$ gilt:

$$\sum_{(i,v)\in T} y_{i,v} \leq \sum_{(i,v)\in\bigcup_{j=1}^{h}\rho_j} k_{i,v} + \sum_{u\in\left(\bigcup_{\substack{j=1\\ \text{mit } i_j>g}}^{h} N[w_j]\right)} k_{g,u}.$$

◇

Beweis
Falls ein $\alpha \in \{1, \ldots, h\}$ existiert mit $y_{i_\alpha, w_\alpha} = 0$, dann sind (a), (b) und (c) gültig, denn in dem Fall ist:

$$\sum_{(i,v)\in T} y_{i,v} = \sum_{(i,v)\in T\backslash\{(i_\alpha,w_\alpha)\}} y_{i,v} \leq \sum_{(i,v)\in\rho_\alpha} k_{i,v} \leq \sum_{(i,v)\in\bigcup_{j=1}^{h}\rho_j} k_{i,v}.$$

Sei jetzt $y_{i_\alpha, w_\alpha} = 1$ für alle $\alpha \in \{1, \ldots, h\}$:

(a) Dann sind in allen Taschen mit Index größer oder gleich $\max(i_1,\ldots,i_h)$ nur Knoten aus $V\setminus\left(\bigcup_{j=1}^{h} N[w_j]\right)$. Also gilt:

$$\sum_{u\in V\setminus\left(\bigcup_{j=1}^{h} N[w_j]\right)} k_{g,u} = 1 \qquad \text{für alle } g\in\{\max(i_1,\ldots,i_h),\ldots,n-\delta(G)\}.$$

Somit gilt für alle $\alpha\in\{1,\ldots,h\}$ und für alle $g\in\{\max(i_1,\ldots,i_h),\ldots,n-\delta(G)\}$:

$$\begin{aligned}\sum_{(i,v)\in T} y_{i,v} &= \sum_{(i,v)\in T\setminus\{(i_\alpha,w_\alpha)\}} y_{i,v} + y_{i_\alpha,w_\alpha}\\ &\le \sum_{(i,v)\in\rho_\alpha} k_{i,v} + 1 = \sum_{(i,v)\in\rho_\alpha} k_{i,v} + \sum_{u\in V\setminus\left(\bigcup_{j=1}^{h} N[w_j]\right)} k_{g,u}\\ &\le \sum_{(i,v)\in\bigcup_{j=1}^{h}\rho_j} k_{i,v} + \sum_{u\in V\setminus\left(\bigcup_{j=1}^{h} N[w_j]\right)} k_{g,u}.\end{aligned}$$

(b) Dann sind alle Knoten aus $V\setminus\left(\bigcup_{j=1}^{h} N[w_j]\right)$ nur in Taschen mit Index größer oder gleich $\max(i_1,\ldots,i_h)$. Also gilt:

$$\sum_{g=\max(i_1,\ldots,i_h)}^{n-\delta(G)} k_{g,u} = 1 \qquad \text{für alle } u\in V\setminus\left(\bigcup_{j=1}^{h} N[w_j]\right).$$

Somit gilt für alle $\alpha\in\{1,\ldots,h\}$ und für alle $u\in V\setminus\left(\bigcup_{j=1}^{h} N[w_j]\right)$:

$$\sum_{(i,v)\in T} y_{i,v} \le \sum_{(i,v)\in\rho_\alpha} k_{i,v} + 1 \le \sum_{(i,v)\in\bigcup_{j=1}^{h}\rho_j} k_{i,v} + \sum_{g=\max(i_1,\ldots,i_h)}^{n-\delta(G)} k_{g,u}.$$

(c) Dann sind nur Knoten aus $\left(\bigcup_{j=1}^{h} N[w_j]\right)$ in Taschen mit Index kleiner als $\max(i_1,\ldots,i_h)$ und die Knoten aus $N[w_j]$ sind nur in den ersten i_j-1 Taschen, für alle $j\in\{1,\ldots,h\}$. Also gilt:

$$\sum_{u\in\left(\bigcup_{\substack{j=1\\ \text{mit } i_j>g}}^{h} N[w_j]\right)} k_{g,u} = 1 \qquad \text{für alle } g\in\{1,\ldots,\max(i_1,\ldots,i_h)-1\}.$$

Somit gilt für alle $\alpha\in\{1,\ldots,h\}$ und für alle $g\in\{1,\ldots,\max(i_1,\ldots,i_h)-1\}$:

$$\sum_{(i,v)\in T} y_{i,v} \le \sum_{(i,v)\in\rho_\alpha} k_{i,v} + 1 \le \sum_{(i,v)\in\bigcup_{j=1}^{h}\rho_j} k_{i,v} + \sum_{u\in\left(\bigcup_{\substack{j=1\\ \text{mit } i_j>g}}^{h} N[w_j]\right)} k_{g,u}. \qquad \square$$

Die folgenden Ungleichungen sind Verallgemeinerungen von (6.6):

(7.10) Gültige Ungleichungen
Sei die Menge $T = \{(i_1, w_1), \ldots, (i_h, w_h)\} \subseteq (I \times V)$ mit $h \geq 1$ gegeben, die die Bedingung B erfüllt. Weiter sei ein Knoten $w_0 \in V \backslash \{w_1, \ldots, w_h\}$ mit

$$\max(i_1, \ldots, i_h) < \left| \bigcup_{j=0}^{h} N[w_j] \right| - \delta(G) + 1 =: i_0 \in I$$

und für jedes $j \in \{0, \ldots, h\}$ ein $\rho_j \in f((T \cup (i_0, w_0)) \backslash \{(i_j, w_j)\})$ gegeben.

(a) Dann gilt für alle $u \in N[w_0] \backslash \left(\bigcup_{j=1}^{h} N[w_j] \right)$:

$$y_{i_0,w_0} + \sum_{(i,v)\in T} y_{i,v} \leq \sum_{(i,v)\in \bigcup_{j=0}^{h} \rho_j} k_{i,v} + \sum_{g=\max(i_1,\ldots,i_h)}^{i_0-1} k_{g,u}.$$

(b) Für alle $g \in \{\max(i_1, \ldots, i_h), \ldots, i_0 - 1\}$ gilt:

$$y_{i_0,w_0} + \sum_{(i,v)\in T} y_{i,v} \leq \sum_{(i,v)\in \bigcup_{j=0}^{h} \rho_j} k_{i,v} + \sum_{u\in N[w_0]\backslash \left(\bigcup_{j=1}^{h} N[w_j] \right)} k_{g,u}.$$

◇

Beweis
Falls ein $\alpha \in \{0, \ldots, h\}$ existiert mit $y_{i_\alpha, w_\alpha} = 0$, dann gelten (a) und (b), denn in dem Fall ist:

$$y_{i_0,w_0} + \sum_{(i,v)\in T} y_{i,v} = \sum_{(i,v)\in (T\cup\{i_0,w_0\})\backslash\{(i_\alpha,w_\alpha)\}} y_{i,v} \leq \sum_{(i,v)\in\rho_\alpha} k_{i,v} \leq \sum_{(i,v)\in\bigcup_{j=0}^{h}\rho_j} k_{i,v}.$$

Sei jetzt $y_{i_\alpha, w_\alpha} = 1$ für alle $\alpha \in \{0, \ldots, h\}$: Dann bleibt, so wie in Satz (7.9), nur noch zu zeigen, dass die zweite Komponente in Teil (a) und (b) jeweils gleich 1 ist: Da Bedingung B erfüllt ist, sind in den ersten $\max(i_1, \ldots, i_h) - 1$ Taschen nur Knoten aus $\bigcup_{j=1}^{h} N[w_j]$. Da $y_{i_0,w_0} = 1$, müssen alle Knoten aus $N[w_0]$ in den ersten $i_0 - 1$ Taschen sein. Somit gilt einerseits für alle $u \in N[w_0] \backslash \left(\bigcup_{j=1}^{h} N[w_j] \right)$:

$$\sum_{g=\max(i_1,\ldots,i_h)}^{i_0-1} k_{g,u} = 1$$

und andererseits für alle $g \in \{\max(i_1, \ldots, i_h), \ldots, i_0 - 1\}$:

$$\sum_{u \in N[w_0] \setminus \left(\bigcup_{j=1}^{h} N[w_j]\right)} k_{g,u} = 1.$$

□

7.4 Spezielle Ungleichungen

In diesem Kapitel werden einige Ungleichungen mehrfacher Ordnung beschrieben, die alle mit Ungleichung (7.7) hergeleitet werden können. Es lohnt sich aber sie explizit zu betrachten.

(7.11) Definition
Sei eine Teilmenge der Knoten $\emptyset \neq S \subseteq V$, ein Knoten $v \in V$ und eine Zahl $j \in \{1, ..., |S| - 1\}$ gegeben. Dann bezeichnet die Zahl

$$\sigma_j(S, v) := \min_{\substack{w_1,\ldots,w_j \in S \setminus \{v\} \\ w_1,\ldots,w_j \text{ verschieden}}} \left| N[v] \cup \bigcup_{i=1}^{j} N[w_i] \right| - \delta(G)$$

den größtmöglichen Index einer Tasche, wo v und maximal $j-1$ von v verschiedene Knoten aus S für y den Wert 1 annehmen können. ◇

$\sigma_1(S, v)$ entspricht dann genau dem schon Bekannten aus den vorigen Kapiteln. Es gilt:

$$\sum_{v \in S} y_{\sigma_j(S,v),v} \leq j \qquad \forall S \subseteq V,\ \forall j \in \{1, \ldots, |S| - 1\}.$$

(7.12) Gültige Ungleichung

$$\sum_{w \in N[v]} y_{i,w} \leq \sum_{j=1}^{i-1} k_{j,v} \quad \forall v \in V,\ \forall i \in I \text{ mit } i \leq \min_{w_1, w_2 \in N[v]} |N[w_1] \cup N[w_2]| - \delta(G).$$

◇

Beweis
Aus Ungleichung (5.8) folgt:

$$y_{i,w} \leq \sum_{j=1}^{i-1} k_{j,v} \qquad \forall w \in N[v].$$

Da $i \leq \min_{w_1,w_2 \in N[v]} |N[w_1] \cup N[w_2]| - \delta(G)$ ist $\sum_{w \in N[v]} y_{i,w} \leq 1$. Somit gilt:

$$\sum_{w \in N[v]} y_{i,w} \leq \sum_{j=1}^{i-1} k_{j,v}.$$

□

(7.13) Satz
Ungleichung (7.12) definiert eine Facette für alle Knoten $v \in V$ und alle $i \in I$ mit $i \leq \min_{w_1,w_2 \in N[v]} |N[w_1] \cup N[w_2]| - \delta(G)$ und $i \leq n - \delta(G) - 1$. ◇

Beweis
Wir benötigen $2 \cdot n^2 - \delta(G) \cdot n - 3 \cdot n - 2 \cdot m + 2 + \delta(G)$ affin unabhängige Punkte, die die Ungleichung

$$\sum_{w \in N[v]} y_{i,w} \leq \sum_{j=1}^{i-1} k_{j,v} \tag{23}$$

für feste v, i gemäß den Voraussetzungen des Satzes mit Gleichheit erfüllen.

(a) Setze $c = n$ und $y_{h,x} = 0$ für alle $x \in V$ und alle $h \in I_x$. Da $\sum_{w \in N[v]} y_{i,w} = 0$, muss auch $k_{j,v} = 0$ für alle $j \in \{1, \ldots, i-1\}$ gelten.

- Setze $k_{i,v} = 1$. Es müssen noch $n - 1$ Knoten auf $n - \delta(G) - 1$ Taschen verteilt werden. Dafür gibt es

  $$(n-1) \cdot (n - \delta(G) - 1) - (n-1) - (n - \delta(G) - 2) + 1$$

  affin unabhängige Punkte.

- Setze $k_{i+1,v} = 1$. Dann gibt es $n - 1$ verschiedene $x \in V \backslash \{v\}$, um $k_{i,x} = 1$ zu setzen. Wir können also weitere

  $$n - 1$$

  affin unabhängige Punkte konstruieren.

- Falls $i \leq n - \delta(G) - 2$, dann können wir für jedes $h \in \{i+2, \ldots, n - \delta(G)\}$ einen weiteren Punkt mit $k_{h,v} = 1$ konstruieren, also weitere

  $$n - \delta(G) - i - 1$$

  affin unabhängige Punkte. Das gilt auch für $i = n - \delta(G) - 1$, denn dann ist diese Zahl gleich Null.

(b) Wir konstruieren für alle $x \in V$ und alle $h \in I$ mit $h \geq 2 + d(x) - \delta(G)$ jeweils einen Punkt mit $y_{h,x} = 1,\ y_{h-1,x} = 0,\ c = n$, und $y_{n-\delta(G),u} = 0$ für alle $u \neq x$ und der (23) mit Gleichheit erfüllt.

- Falls $x \notin N[v]$, setzen wir alle Knoten aus $N[x]$ in die ersten $1 + d(x) - \delta(G)$ Taschen, $y_{h,x} = 1$ und $k_{n-\delta(G),v} = 1$. Linke und rechte Seite der Ungleichung sind dann gleich 0.
- Falls $x \in N[v]$, unterscheiden wir zwei Fälle:
  - Falls $h \geq i + 1$: setze $k_{i,v} = 1$ und alle anderen Knoten aus $N[x]$ in die ersten $h - 1$ Taschen, $y_{i,v} = 0$, $y_{i,x} = 0$, $y_{h-1,x} = 0$, $y_{h,x} = 1$. Linke und rechte Seite der Ungleichung sind dann gleich 0.
  - Falls $h \leq i$: setze $k_{1,v} = 1$ und alle anderen Knoten aus $N[x]$ in die ersten $h - 1$ Taschen, $y_{h,x} = y_{i,x} = 1$. Linke und rechte Seite der Ungleichung sind dann gleich 1.

In beiden Fällen ist Ungleichung (23) mit Gleichheit erfüllt. Dies können wir nacheinander für alle $x \in V$ und alle $h \geq 2 + d(x) - \delta(G)$ machen und erhalten

$$\sum_{x \in V} \Big(n - \delta(G) - (2 + d(x) - \delta(G)) + 1 \Big) = \sum_{x \in V} \Big(n - d(x) - 1 \Big) = n^2 - 2 \cdot m - n$$

affin unabhängige Punkte.

(c) Setze $c = n,\ y_{i,v} = 1,\ y_{i-1,v} = 0$ und $y_{h,x} = 0$ für alle $x \in V \backslash \{v\}$ und alle $h \in I_x$. Wir können für alle $j \in \{1, \ldots, i-1\}$ einen Punkt konstruieren mit $k_{j,v} = 1$. Einer dieser Punkte ist affin abhängig zu einem Punkt aus (b). Also bleiben noch

$$i - 2$$

affin unabhängige Punkte.

(d) In allen Punkten davor wurde $c = n$ gesetzt. Den letzten affin unabhängigen Punkt konstruieren wir, indem wir einen Punkt aus (b) oder (c) nehmen und $c = n - 1$ setzen.

Insgesamt erhalten wir

$$\begin{aligned} &(n-1) \cdot (n - \delta(G) - 1) - (n-1) - (n - \delta(G) - 2) + 1 \\ &+ (n-1) + (n - \delta(G) - i - 1) + (n^2 - 2 \cdot m - n) + (i - 2) + 1 \\ =&2 \cdot n^2 - \delta(G) \cdot n - 3 \cdot n - 2 \cdot m + 2 + \delta(G) \end{aligned}$$

affin unabhängige Punkte und damit ist der Satz bewiesen. □

(7.14) Gültige Ungleichungen

$$\sum_{v\in S} y_{\sigma_h(S,v),v} \leq h \cdot \sum_{i=1}^{p-1} k_{i,w} \qquad \forall S \subseteq V \text{ mit } |S| \geq 2,\ \forall w \in \bigcap_{v\in S} N[v],$$
$$\forall h \in \{1,\ldots,|S|-1\},\ p = \max_{v\in S} \sigma_h(S,v)$$

$$\sum_{v\in S} y_{\sigma_h(S,v),v} \leq h \cdot \sum_{w\in V\setminus\bigcap_{v\in S} N[v]} k_{i,w} \qquad \forall S \subseteq V \text{ mit } |S| \geq 2,\ \forall h \in \{1,\ldots,|S|-1\},$$
$$\forall i \in \{\max_{v\in S} \sigma_h(S,v),\ldots,n-\delta(G)\} \qquad \diamond$$

Beweis
Laut Definition (7.11) gilt:

$$\sum_{v\in S} y_{\sigma_h(S,v),v} \leq h.$$

Und falls $\sum_{i=1}^{p-1} k_{i,w} = 0$ dann ist nach Ungleichung (5.8)

$$\sum_{v\in S} y_{\sigma_h(S,v),v} = 0.$$

□

Ungleichungen (7.14) sind Erweiterungen von (7.12). Der Aufwand, um für alle $S \subseteq V$ neue Ungleichungen zu konstruieren ist bei größeren Graphen allerdings zu groß. Aus diesem Grund bieten sich die folgenden Ungleichungen an:

(7.15) Gültige Ungleichungen
Für alle Knoten $v \in V$, für alle Zahlen $h \in \{1,\ldots,d(v)\}$ und für alle Indizes

$$i \in \left\{ \min_{w\in N[v]} \sigma_h(N[v],w),\ldots,\max_{w\in N[v]} \sigma_h(N[v],w) \right\}$$

gilt:

$$\sum_{w\in N[v]} y_{\min(\sigma_h(N[v],w),i),w} \leq h \cdot \sum_{j=1}^{i-1} k_{j,v}$$
$$\sum_{w\in N[v]} y_{\min(\sigma_h(N[v],w),i),w} \leq h - h \cdot \sum_{w\in\bigcap_{u\in N[v]} N[u]} k_{j,w} \qquad \forall j \in \{i,\ldots,n-\delta(G)\}. \qquad \diamond$$

Beweis
Der Beweis erfolgt analog zum Beweis von Ungleichung (7.14). □

7.5 Beispiele

Als Beispiel betrachten wir den Graphen G_5 aus Abbildung 17. Um eine gültige Ungleichung für

$$y_{2,a} + y_{2,e} + y_{3,b} + y_{3,d} + y_{3,e}$$

herzuleiten, müssen wir alle vorgestellten Ungleichungen kombinieren.

Also sei $T = \{\underbrace{(2,a)}_{=t_1}, \underbrace{(2,e)}_{=t_2}, \underbrace{(3,b)}_{=t_3}, \underbrace{(3,d)}_{=t_4}, \underbrace{(3,e)}_{=t_5}\}$. Dann wenden wir Ungleichung (7.7) an. Es gilt:

$$\begin{aligned}\Phi(T) = T \cup \Big\{ & \underbrace{\{(2,a),(3,b)\}}_{=t_6}, \underbrace{\{(2,a),(3,d)\}}_{=t_7}, \underbrace{\{(2,a),(3,e)\}}_{=t_8}, \underbrace{\{(2,e),(3,b)\}}_{=t_9}, \\ & \underbrace{\{(2,e),(3,d)\}}_{=t_{10}}, \underbrace{\{(2,e),(3,e)\}}_{=t_{11}}, \underbrace{\{(3,b),(3,d)\}}_{=t_{12}}, \underbrace{\{(3,b),(3,e)\}}_{=t_{13}}, \\ & \underbrace{\{(3,d),(3,e)\}}_{=t_{14}}, \underbrace{\{(2,e),(3,b),(3,e)\}}_{=t_{15}}, \underbrace{\{(2,e),(3,d),(3,e)\}}_{=t_{16}} \Big\}.\end{aligned}$$

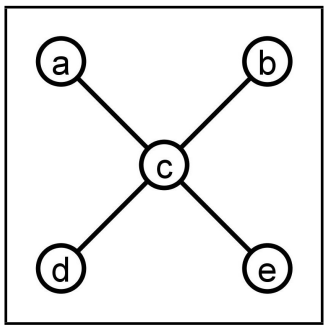

Abbildung 17: Der Graph G_5

Laut Ungleichung (5.8) gilt:

$$y_{2,a},\ y_{2,e} \leq k_{1,c} \quad \text{und} \quad y_{3,b}\ y_{3,d}\ y_{3,e} \leq k_{1,c} + k_{2,c}.$$

Somit gilt auch nach Lemma (7.5):

$$\begin{aligned} y_{2,a} + y_{3,b} &\leq k_{1,c} + k_{1,c} + k_{2,c}, \\ y_{2,a} + y_{3,d} &\leq k_{1,c} + k_{1,c} + k_{2,c}, \\ &\text{u.s.w.} \end{aligned}$$

Also

$$\begin{aligned}
\rho_{t_1} = \rho_{t_2} &= \Big\{(1,c)\Big\} \in f(t_i), && i \in \{1,2\},\\
\rho_{t_3} = \rho_{t_4} = \rho_{t_5} &= \Big\{(1,c),(2,c)\Big\} \in f(t_i), && i \in \{3,4,5\},\\
\rho_{t_6} = \rho_{t_7} = \rho_{t_8} = \rho_{t_9} = \rho_{t_{10}} = \rho_{t_{11}} &= \Big\{(1,c),(1,c),(2,c)\Big\} \in f(t_i), && i \in \{6,\ldots,11\}.
\end{aligned}$$

Nach Ungleichung (7.9) gilt:

$$\begin{aligned}
\rho_{t_{12}} &= \Big\{(1,c),(2,c),(4,a),(4,e)\Big\} \in f(t_{12}),\\
\rho_{t_{13}} &= \Big\{(1,c),(2,c),(4,a),(4,e)\Big\} \in f(t_{13}),\\
\rho_{t_{14}} &= \Big\{(1,c),(2,c),(4,a),(4,e)\Big\} \in f(t_{14}).
\end{aligned}$$

Und nach Ungleichung (7.1) und Lemma (7.5) gilt:

$$\rho_{t_{15}} = \rho_{t_{16}} = \Big\{(1,c),(1,e),(1,c),(2,c)\Big\}.$$

Und schließlich erhalten wir eine gültige Ungleichung für T:

$$y_{2,a} + y_{2,e} + y_{3,b} + y_{3,d} + y_{3,e} \leq \sum_{(i,v)\in\bigcup_{j=1}^{16}\rho_{t_j}} k_{i,v} = 2\cdot k_{1,c} + k_{1,e} + k_{2,c} + k_{4,a} + k_{4,e}.$$

Diese Ungleichung definiert sogar eine Facette.

§8 Das zweite Modell

8.1 Die Variablen

Beim zweiten Modell werden wir ausnutzen, dass die Knoten jeder Clique zusammen in mindestens einer Tasche sind. Die Menge der Taschen bleibt $I := \{1, \ldots, n - \delta(G)\}$, aber die Anzahl der Knoten, die von einer Tasche zur nächsten neu hereinkommt, kann beliebig sein. Wir definieren:

$$K := \text{Menge aller Cliquen } q \text{ mit } |q| \geq 3$$
$$Q := V \cup E \cup K.$$

Dann brauchen wir neue Variablen:

$$d_{i,q} = \begin{cases} 1 & \text{, falls Clique } q \in Q \text{ in Tasche } X_i \text{ ist} \\ 0 & \text{, sonst} \end{cases}$$
$$c \in \mathbb{N}_0.$$

8.2 Das ganzzahlige lineare Programm

Die Knoten jeder Clique sind zusammen in mindestens einer Tasche:

$$\sum_{i \in I} d_{i,q} \geq 1 \qquad \forall q \in Q$$

Seien q_1, q_2 zwei Cliquen mit $q_1 \subset q_2$. Falls dann q_2 in der Tasche X_i ist, so ist natürlich auch q_1 in dieser Tasche. Also

$$d_{i,q_2} \geq d_{i,q_1} \qquad \forall i \in I,\ \forall q_1, q_2 \in Q \text{ mit } q_2 \subset q_1.$$

Alle Taschen, die eine bestimmte Clique $q \in Q$ enthalten, sind zusammenhängend:

$$d_{i,q} + d_{k,q} \leq d_{j,q} + 1 \qquad \forall i, j, k \in I \text{ mit } i < j < k,\ \forall q \in Q$$

Wir minimieren die maximale Anzahl an Knoten, die zusammen in einer Tasche sind:

$$\sum_{v \in V} d_{i,v} \leq c \qquad \forall i \in I.$$

Das ganzzahlige lineare Programm **(V1)** sieht dann folgendermaßen aus:

$$\min\ c - 1$$

$$\text{s.d.:} \sum_{v \in V} d_{1,v} = \delta(G) + 1 \tag{24}$$

$$\sum_{i \in I} d_{i,q} \geq 1 \qquad \forall q \in Q \tag{25}$$

$$d_{i,q_1} \geq d_{i,q_2} \qquad \forall i \in I,\ \forall q_1, q_2 \in Q \text{ mit } q_1 \subset q_2 \tag{26}$$

$$d_{i,q} + d_{k,q} \leq d_{j,q} + 1 \qquad \forall i,j,k \in I \text{ mit } i < j < k,\ \forall q \in Q \tag{27}$$

$$\sum_{v \in V} d_{i,v} \leq c \qquad \forall i \in I \tag{28}$$

Bei der Wahl von Q ist das Modell flexibel. Weitere Varianten:

- **Variante (V2):** Es reicht aus, Q als Vereinigung der Knoten und Kanten zu definieren, also
$$Q := V \cup E.$$
So verzichtet man aber auf den wahrscheinlich größten Vorteil des Modells, nämlich, dass große Cliquen *automatisch* in eine Tasche platziert werden.

- **Variante (V3):** Eine andere Möglichkeit besteht darin, eine Teilmenge der Kanten und Cliquen zu definieren: Sei
$$S := \{s_1, s_2, \ldots, s_h\} \subseteq E \cup K,$$
so dass für alle $e \in E$ ein $i \in \{1, \ldots, h\}$ existiert, mit $e \subseteq s_i$.
Dann ist $Q := V \cup S$. So müssen nicht alle Kanten einzeln in Q sein, sondern es reicht aus, dass für jede Kante $\{v, w\}$ eine Clique q in Q ist mit $\{v, w\} \subseteq q$.

Eine Erweiterung **(E)** des Modells anhand der Variablen:

$$x_i = \begin{cases} 1 & \text{, falls } |X_i| > 0 \\ 0 & \text{, sonst} \end{cases} \qquad i \in I$$

und den zusätzlichen Nebenbedingungen:

$$x_{i+1} \leq x_i \qquad \forall i \in I \backslash \{n\}$$

$$x_i \leq \sum_{v \in V} d_{i,v} \qquad \forall i \in I$$

$$d_{i,v} \leq x_i \qquad \forall i \in I,\ \forall v \in V$$

$$\sum_{i \in I} x_i \leq n - c + 1 \text{ und } x_1 = 1$$

ist möglich.

§9 Rechenstudie

Mit Hilfe der Modellierungssprache *ZIMPL* [21] können wir das Pfadweite- Problem als (ganzzahliges) lineares Programm formulieren und mit der Lösungssoftware *SCIP* [1] lösen. ZIMPL und SCIP sind frei erhaltbar. Die im Folgenden vorgestellte Rechenstudie wurde auf einem Computer mit zwei 2,93 GHz Prozessoren und 4 GB Ram durchgeführt. Um vergleichbare Daten zu erhalten, wurde die Standarteinstellung von SCIP version 2.0.2 beibehalten.

9.1 Die getesteten Graphen

9.1.1 Die Mycielski- Graphen

Die *Mycielski- Graphen* [2] bilden eine Folge von dreiecksfreien Graphen $M_1, M_2, M_3, \ldots$, wobei der Graph $M_{n+1} = (V, E)$ wie folgt aus dem Graphen $M_n = (V', E')$ hervorgeht:

$$V = V' \cup V'' \cup \{z\},$$

wobei $V'' = \{y_1, \ldots, y_n\}$ eine Kopie von $V' = \{x_1, \ldots, x_n\}$ ist und z ein weiterer Knoten ist. Die Kanten werden auf drei verschiedene Arten erzeugt:

1. Alle Kanten aus M_n bleiben bestehen.
2. y_j wird mit allen Nachbarn von x_j verbunden ($1 \leq j \leq n$).
3. z wird mit allen Knoten aus V'' verbunden.

$$\text{d.h. } E = E' \cup \{(x_i, y_j) | (x_i, x_j) \in E'\} \cup \{(y_j, z) | 1 \leq j \leq n\}.$$

Als Ausgangsgraph dient der Graph M_1, der zwei Knoten x_1, x_2 hat, die durch eine Kante verbunden sind.

9.1.2 Die Queen- Graphen

Die *Queen- Graphen* (Damen- Graphen) $Q_{n,n} = (V, E)$ werden für ein $n \in \mathbb{N}$ wie folgt konstruiert (siehe [25]):

- Jeder Knoten $v \in V$ repräsentiert ein Feld des $n \times n$- Schachbretts.

- Zwei Knoten $v, w \in V$ sind genau dann benachbart, wenn die zugehörigen Felder auf dem Schachbrett in derselben Zeile, Spalte, oder Diagonale liegen.

Die Kanten entsprechen also den Zugmöglichkeiten einer Dame beim Schachspiel.

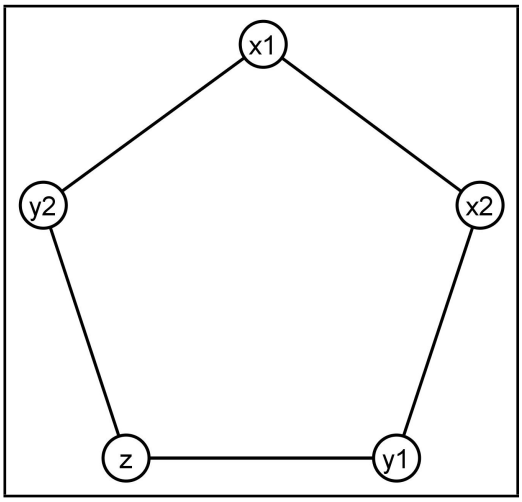

Abbildung 18: Der Mycielski- Graph M_2

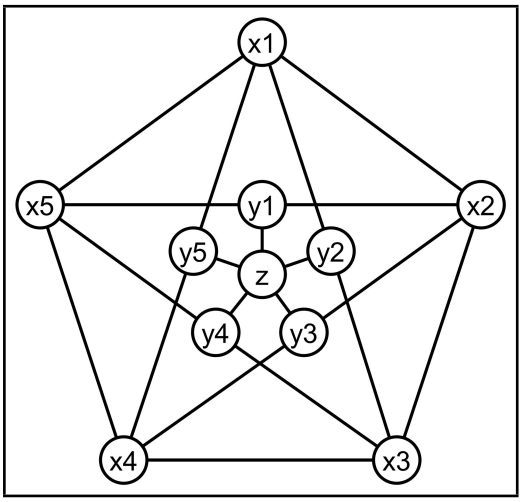

Abbildung 19: Der Mycielski- Graph M_3

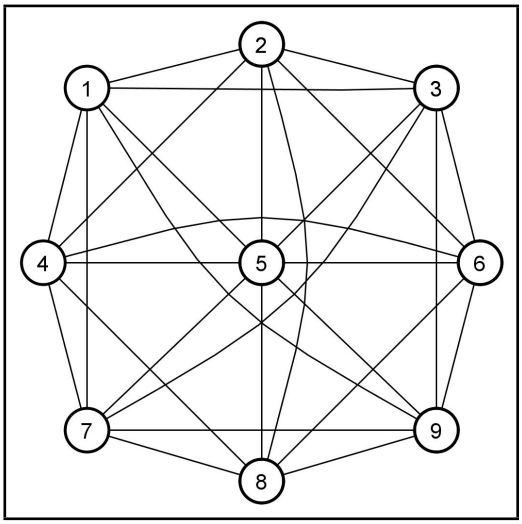

Abbildung 20: Der Queen- Graph $Q_{3,3}$

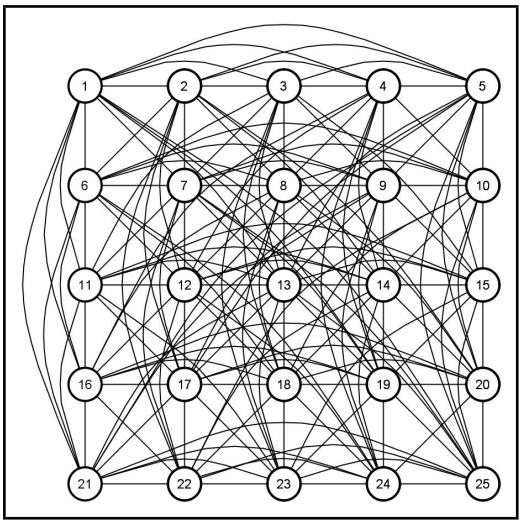

Abbildung 21: Der Queen- Graph $Q_{5,5}$

9.2 Ergebnisse für das erste Modell

Die Ungleichungen, die in den vorigen Kapiteln hergeleitet wurden, werden auf den eben vorgestellten Graphen getestet. Dabei wird um abzukürzen folgende Notation verwendet:

Notation	Beschreibung
Ord.1*	Ungleichungen (5.1), (5.4), (5.8), (5.10), (5.13) und (5.14).
Ord.1	Ungleichungen (5.6), (5.12), (5.13) und (5.14).
(6.6)*	Ungleichung (6.6) nur für $g(J_v, S_v, J_w, S_w) = \sum_{j=1}^{\sigma_1(v,w)} k_{j,x}$ für alle $x \in N[v] \cap N[w]$.
(6.7)*	Ungleichung (6.7) nur für $g(J_v, S_v, J_w, S_w) = \sum_{j=1}^{\sigma_1(v,w)} k_{j,x}$ für alle $x \in N[v] \cap N[w]$.
(6.8)*	Ungleichung (6.8) nur für $i = j$
$(7.14)_j$	Ungleichung (7.14) für $h \leq j$
$(7.14)_{V,j}$	Ungleichung (7.14) nur für $S = V$ und $h \leq j$

Tabelle 1: Erläuterungen zur verwendeten Notation

9.2.1 Myciel 3

Der Graph M_3 (siehe Abbildung 19) hat 11 Knoten, 20 Kanten und der Minimalgrad ist 3. Zuerst wird die Pfadweite mit Hilfe eines ganzzahligen linearen Programms bestimmt. Siehe dazu Tabelle 2. Die Pfadweite ist dementsprechend 5.

Neben-bedingungen	Anzahl Variablen	Anzahl Nebenb.	Lösung des GLP	Zeit in Sek.
(1)-(6)	159	406	5,0000	4,50
(1)-(6),(5.1)	159	476	5,0000	3,99
(1)-(6),(5.4)	159	476	5,0000	6,41
(1)-(6),(5.8)	159	726	5,0000	5,11
(1)-(6),(5.10)	159	413	5,0000	4,46
(1)-(6),(7.1)	159	417	5,0000	5,38
(1)-(6),$(7.14)_{V,2}$	159	413	5,0000	0,71

Tabelle 2: Ganzzahlige lineare Programme für Myciel3

Wir testen die hergeleiteten Ungleichungen, indem wir die lineare Relaxierung betrachten. Die lineare Relaxierung für die Standard- Ungleichung ergibt den Wert 3. Dieser Wert entspricht dem Minimalgrad; das liegt daran, dass unser Modell in die erste Tasche der Pfadzerlegung immer $\delta(G)+1$ Knoten hinein legt. Da der Graph M_3 nur 11 Knoten hat, können wir die Ungleichungen (5.6) und (5.12) zusammen ins LP laden. Im Folgenden werden wir auch größere Graphen betrachten, so dass wir in (5.6) bzw. in (5.12) nicht für alle $S \subseteq V \backslash N[v]$ und alle $J \subseteq \{i, ..., n-\delta(G)\}$ neue Ungleichungen erzeugen können. Deswegen beschränken wir uns nur auf die wichtigsten Fälle: (5.1),(5.4),(5.8) und (5.10). Wie aus Tabelle 3 hervorgeht, ist die Lösung der LP- Relaxierung für den angegebenen Graphen in beiden Fällen gleich 3.

Nebenbedingungen (1)-(6)+	Anzahl Nebenb.	Lösung der LP Relaxierung	Zeit in Sek.
Ord.1*	1027	3,0000	0,08
Ord.1	27088	3,0000	5,99
(6.4)	5436	3,2000	0,48
$(6.6)^*$	2026	3,0000	0,16
$(6.7)^*$	3466	3,1034	0,25
$(6.8)^*$	4596	3,0000	0,22
Ord.1*,(6.4),$(6.6)^*$, $(6.7)^*$, $(6.8)^*$	14927	3,2667	3,32
Ord.1,(6.4),$(6.6)^*$, $(6.7)^*$, $(6.8)^*$	40988	3,2667	27,83
Ord.1*,(6.4),$(6.6)^*$, $(6.7)^*$, $(6.8)^*$, (7.1)	14938	3,2667	4,84
$(7.14)_1$	9564	4,0000	0,16
$(7.14)_2$	15865	5,0000	0,28
$(7.14)_{V,1}$	410	4,0000	0,02
$(7.14)_{V,2}$	413	5,0000	0,01
$(7.15)_1$	667	3,8000	0,01
$(7.15)_2$	1012	3,8571	0,11
$(7.15)_3$	1385	3,8571	0,14
$(7.15)_4$	1595	3,8571	0,18
$(7.15)_5$	1630	3,8571	0,11
+O1*,(6.4),(6.6),(6.7),(6.8), (7.1),$(7.15)_5$	16162	3,8800	6,63

Tabelle 3: Lineare Relaxierung für Myciel3

Durch das Hinzufügen aller Ungleichungen erster Ordnung gelingt es also nicht, eine bessere untere Schranke zu erhalten wie der Minimalgrad. Der Grund dafür ist, dass der Minimalgrad zu hoch ist, um die Wirkung zu zeigen. Die wenigen

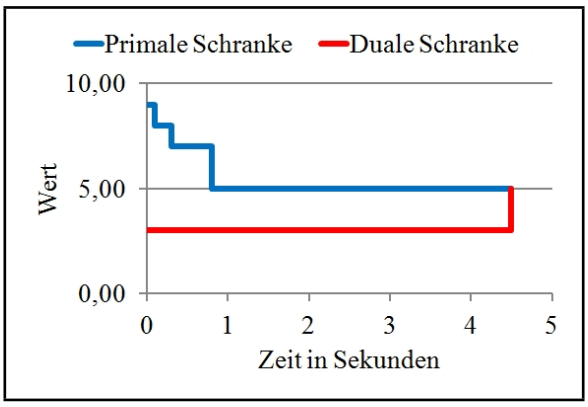

Abbildung 22: Rechenzeit des GLP mit den Nebenbedingungen (1)-(6) für M_3

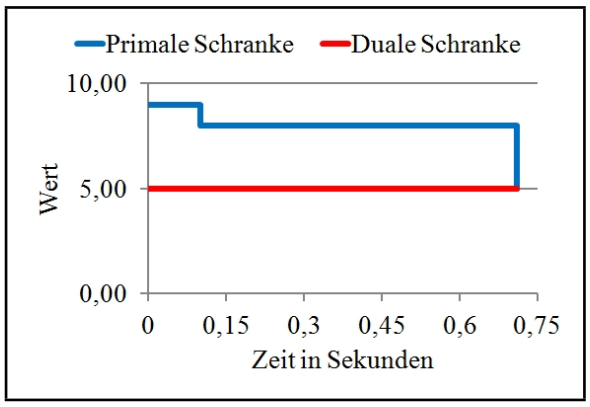

Abbildung 23: Rechenzeit des GLP mit den Nebenbedingungen (1)-(6),$(7.14)_{V,2}$ für M_3

ausgewählten Ungleichungen zweiter Ordnung erhöhen den Wert auf 3,2667.

Mit Hilfe von Ungleichng (7.14) erhalten wir bereits eine Lösung mit Wert 5. Dabei ist es nicht nötig für jedes $S \subseteq V$ eine Ungleichung zu erzeugen. Es reicht nur $S = V$ zu wählen. Durch das Hinzufügen von nur 7 Ungleichungen zu den Standard- Ungeichungen (1)-(6) erhalten wir bei diesem Graphen bereits eine (nicht ganzzahlige) Lösung mit dem selben Wert wie die Pfadweite; nämlich 5. D.h. wir erhalten eine untere Schranke mit Wert 5 in 0,01 Sekunden. Das ganzzahlige Programm, das nur die Standard- Ungeichungen (1)-(6) enthält, erreicht diese Schranke erst nach über 4 Sekunden wie aus Abbildung 22 ersichtlich ist. Ungleichung (7.14) mit $S = V$ hilft uns also schnell eine gute untere Schranke zu finden.

(9.1) Bemerkung
Da der Graph Myciel3 M_3 keinen Knoten enthält, der zu allen anderen Knoten adjazent ist, reduziert sich Ungleichung $(7.14)_{V,j}$ zu

$$\sum_{v \in S} y_{\sigma_h(V,v),v} \leq h \qquad \forall h \in \{1, \ldots, j\}.$$

◇

9.2.2 Myciel 4

Der Graph M_4 hat 23 Knoten, 71 Kanten und der Minimalgrad ist 4.

Nebenbedingungen (1)-(6)+	Anzahl Nebenb.	Lösung der LP Relaxierung	Zeit in Sek.
—	2928	4,0000	0,42
Ord.1*	7417	4,3051	8,19
(6.4)	124717	5,4628	1573,32
(6.6)*	26023	4,0000	60,59
(6.7)*	97198	4,4993	547,44
Ord.1*,(6.4),(6.6),(6.7)	246571	out of memory	—
Ord.1*, (7.1)	7440	4,3051	11,29
$(7.14)_{V,1}$	2937	5,0000	0,21
$(7.14)_{V,2}$	2945	6,0000	0,27
$(7.14)_{V,3}$	2952	7,0000	0,34
$(7.14)_{V,4}$	2958	7,0000	0,26
$(7.14)_{V,5}$	2964	9,0000	0,26
$(7.14)_{V,6}$	2969	10,0000	0,23
$(7.15)_1$	5810	5,0000	1,04
$(7.15)_2$	9161	5,8571	3,53
$(7.15)_3$	12820	6,3044	8,37
$(7.15)_4$	16701	6,3044	15,96
$(7.15)_5$	19749	6,4126	24,36
$(7.15)_6$	22031	6,4126	28,47
$(7.15)_7$	23334	6,4126	36,16
+O1*,(6.4),(7.1),$(7.15)_5$	146050	6,6150	4338,92

Tabelle 4: Lineare Relaxierung für Myciel4

Dank der linearen Relaxierung wissen wir jetzt, dass 10 eine duale Schranke (untere Schranke) für die Pfadweite von M_4 ist. Die ganzzahligen linearen Programme zeigen, dass 10 sogar der Pfadweite entspricht. Siehe dazu Tabelle 5.

Neben-bedingungen	Anzahl Variablen	Anzahl Nebenb.	Lösung des GLP	Zeit in Sek.
(1)-(6)	802	2928	10,0000	2297,31
(1)-(6),$(7.14)_{V,6}$	802	2969	10,0000	36,14

Tabelle 5: Ganzzahlige lineare Programme für Myciel4

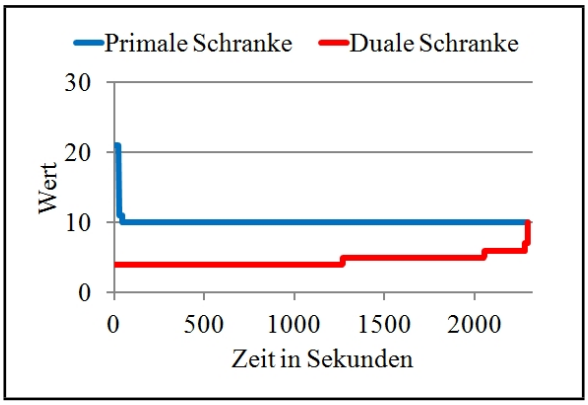

Abbildung 24: Rechenzeit des GLP mit den Nebenbedingungen (1)-(6) für M_4

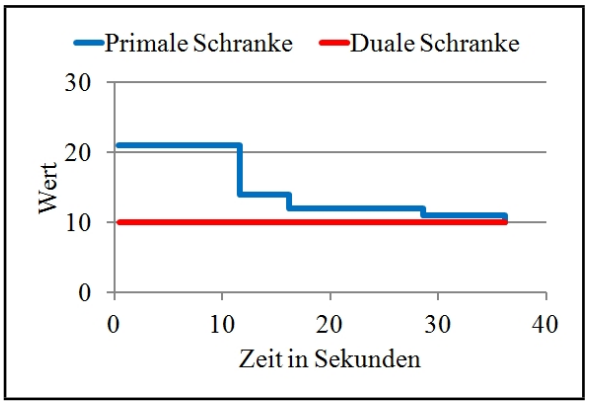

Abbildung 25: Rechenzeit des GLP mit den Nebenbedingungen (1)-(6),$(7.14)_{V,6}$ für M_4

9.2.3 Myciel 5

Der Graph M_5 hat 47 Knoten, 236 Kanten und der Minimalgrad ist 5. Die lineare Relaxierung (Tabelle 6) ergibt einen Wert von 10,1040. Da die Pfadweite ganzzahlig ist, erhalten wir eine untere Schranke von 11.

Nebenbedingungen (1)-(6)+	Anzahl Nebenb.	Lösung der LP Relaxierung	Zeit in Sek.
—	19530	5,0000	12,86
Ord.1*	50491	6,1706	940,77
Ord.1*, (7.1)	50538	6,1706	900,79
$(7.14)_{V,1}$	19550	6,0000	15,54
$(7.14)_{V,2}$	19569	7,0000	19,07
$(7.14)_{V,3}$	19587	8,0000	14,39
$(7.14)_{V,4}$	19604	8,0000	14,80
$(7.15)_1$	47455	6,0000	59,36
$(7.15)_2$	78530	6,9091	216,35
$(7.15)_3$	111583	7,8497	939,27
$(7.15)_4$	146305	8,2810	1293,04
$(7.15)_5$	182365	9,2983	4475,19
+Ord.1*, (7.1),$(7.14)_{V,4}$, $(7.15)_5$	213447	10,1040	13760,80

Tabelle 6: Lineare Relaxierung für Myciel5

Nach 1000 min hat das ganzzahlige Programm eine Pfadzerlegung der Weite 20 gefunden. Somit bleibt ein Abstand von 81,82% zur dualen Schranke aus der linearen Relaxierung.

Neben-bedingungen	Anzahl Variablen	Anzahl Nebenb.	Duale Schranke	Gap in %	Primale Schranke	Zeit in Minuten
(1)-(6)	3665	19530	5	300,00%	20	1000

Tabelle 7: Ganzzahliges lineares Programm für Myciel5

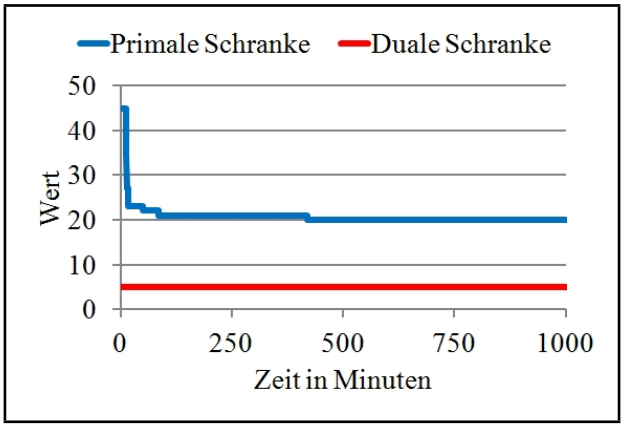

Abbildung 26: Rechenzeit des GLP mit den Nebenbedingungen (1)-(6) für M_5

9.2.4 Queen5_5

Der Graph $Q_{5,5}$ hat 25 Knoten, 160 Kanten und der Minimalgrad ist 12. Wir betrachten zuerst die lineare Relaxierung (Tabelle 8). Diese ergibt eine duale Schranke von 17. Dann wird die Pfadweite mit Hilfe eines ganzzahligen linearen Programms bestimmt. Siehe dazu Tabelle 9. Die Pfadweite ist 18.

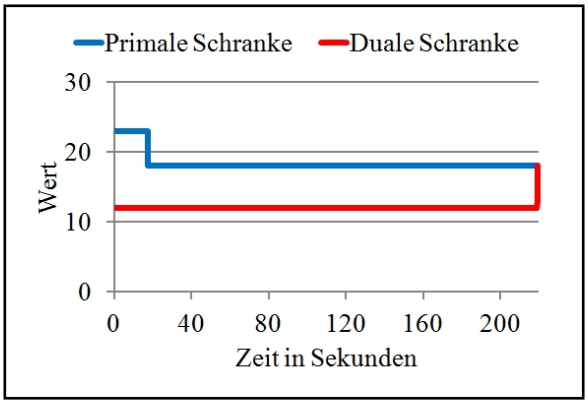

Abbildung 27: Rechenzeit des GLP mit den Nebenbedingungen (1)-(6) für $Q_{5,5}$

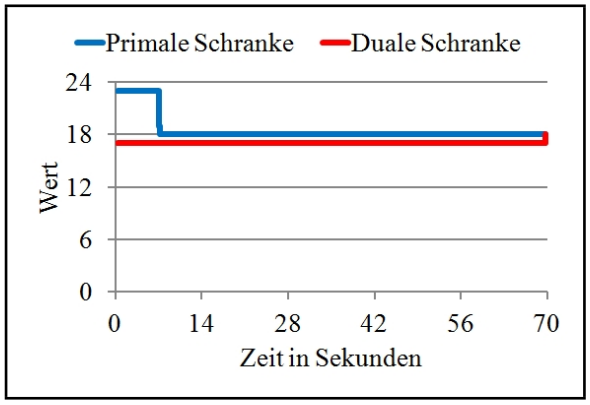

Abbildung 28: Rechenzeit des GLP mit den Nebenbedingungen (1)-(6),$(7.14)_{V,2}$ für $Q_{5,5}$

Nebenbedingungen (1)-(6)+	Anzahl Nebenb.	Lösung der LP Relaxierung	Zeit in Sek.
—	4138	12,0000	0,18
Ord.1*	9302	12,0000	3,19
(6.4)	195730	12,0000	885,36
(6.6)*	134450	12,0000	333,08
(6.7)*	130290	12,0000	93,32
Ord.1*, (7.1)	9327	12,0000	3,25
$(7.14)_{V,1}$	4145	16,0000	0,07
$(7.14)_{V,2}$	4150	17,0000	0,12
$(7.14)_{V,3}$	4154	17,0000	0,09
$(7.14)_{V,4}$	4157	17,0000	0,09
$(7.14)_{V,5}$	4160	17,0000	0,07
$(7.15)_1$	5713	15,6923	0,97
$(7.15)_2$	7613	16,5385	3,49
$(7.15)_3$	9690	16,5385	5,29
$(7.15)_4$	11827	16,5385	4,04
$(7.15)_5$	14024	16,5385	5,49
$(7.15)_6$	16224	16,5385	18,31
+Ord.1*,(7.1),$(7.14)_{V,4}$, $(7.15)_4$	17035	17,0000	4,34

Tabelle 8: Lineare Relaxierung für Queen5_5

Neben-bedingungen	Anzahl Variablen	Anzahl Nebenb.	Lösung des GLP	Zeit in Sek.
(1)-(6)	606	4138	18,0000	219,18
(1)-(6),$(7.14)_{V,2}$	606	4150	18,0000	69,70

Tabelle 9: Ganzzahlige lineare Programme für Queen5_5

9.2.5 Queen6_6

Der Graph $Q_{6,6}$ hat 36 Knoten, 290 Kanten und der Minimalgrad ist 15. Die lineare Relaxierung (Tabelle 10) ergibt einen Wert von 25. Diese Lösung ist allerdings noch nicht ganzzahlig. Somit ist 25 eine untere Schranke für die Pfadweite.

Nebenbedingungen (1)-(6)+	Anzahl Nebenb.	Lösung der LP Relaxierung	Zeit in Sek.
Ord.1*	27202	15,0000	41,78
Ord.1*, (7.1)	27238	15,0000	53,80
$(7.14)_{V,1}$	12303	21,0000	1,55
$(7.14)_{V,2}$	12313	24,0000	0,40
$(7.14)_{V,3}$	12321	24,0000	0,56
$(7.14)_{V,4}$	12328	25,0000	0,25
$(7.15)_1$	17630	20,6875	8,63
$(7.15)_2$	24506	22,7500	70,31
$(7.15)_3$	31926	22,7500	194,07
$(7.15)_4$	39702	22,7500	232,26
$(7.15)_5$	47658	22,7500	207,68
+Ord.1*,(7.1),$(7.15)_4$	54650	22,7500	679,78

Tabelle 10: Lineare Relaxierung für Queen6_6

Mit dem ganzzahligen Programm 11 erhalten wir eine ganzzahlige Lösung von 25.

Neben-bedingungen	Anzahl Variablen	Anzahl Nebenb.	Lösung des GLP	Zeit in Sek.
(1)-(6)	1437	12290	25,0000	7197,92
(1)-(6),$(7.14)_{V,4}$	1437	12328	25,0000	172,07

Tabelle 11: Ganzzahlige lineare Programme für Queen6_6

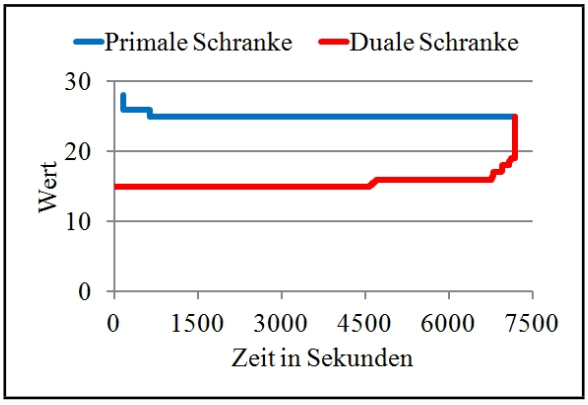

Abbildung 29: Rechenzeit des GLP mit den Nebenbedingungen (1)-(6) für $Q_{6,6}$

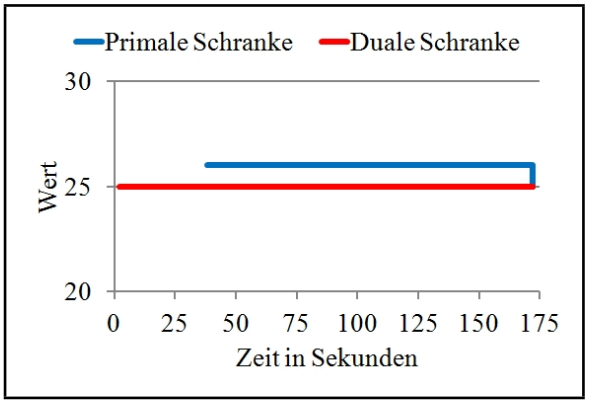

Abbildung 30: Rechenzeit des GLP mit den Nebenbedingungen (1)-(6),$(7.14)_{V,4}$ für $Q_{6,6}$

9.2.6 Queen7_7

Der Graph $Q_{7,7}$ hat 49 Knoten, 476 Kanten und der Minimalgrad ist 18. Mit Hilfe der linearen Relaxierung erhalten wir eine untere Schranke von 33 für die Pfadweite. Siehe dazu Tabelle 12.

Die beste gefundene ganzzahlige Lösung des ganzzahligen linearen Programms (Tabelle 12) ist 35. Somit bleibt nur noch ein Abstand von 6,06% zwischen der gefundenen ganzzahligen Lösung und der dualen Schranke.

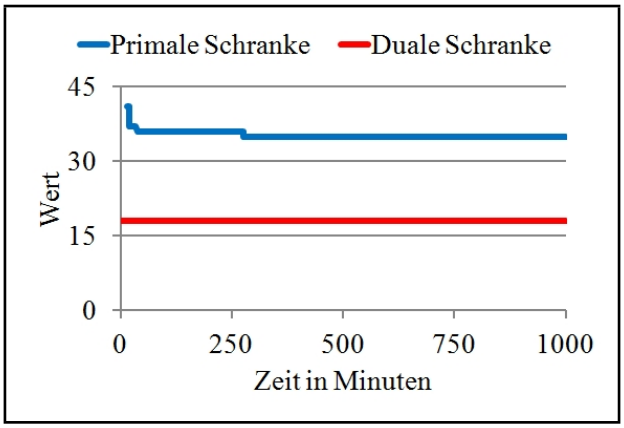

Abbildung 31: Rechenzeit des GLP mit den Nebenbedingungen (1)-(6) für $Q_{7,7}$

Nebenbedingungen (1)-(6)+	Anzahl Nebenb.	Lösung der LP Relaxierung	Zeit in Sek.
—	29934	18,0000	15,86
+Ord.1*,(7.1)	65528	18,0000	312,79
$(7.14)_{V,1}$	29951	26,0000	5,59
$(7.14)_{V,2}$	29964	30,0000	3,00
$(7.14)_{V,3}$	29975	30,0000	2,65
$(7.14)_{V,4}$	29985	33,0000	2,34
$(7.15)_1$	46005	25,5790	217,62
$(7.15)_2$	65514	28,8421	939,55
$(7.15)_3$	86620	29,0778	2253,68
$(7.15)_4$	108631	29,9887	2833,33
+Ord.1*,(7.1),$(7.14)_{V,4}$, $(7.15)_4$	144276	33,0000	4512,29

Tabelle 12: Lineare Relaxierung für Queen7_7

Neben-bedingungen	Anzahl Variablen	Anzahl Nebenb.	Duale Schranke	Gap in %	Primale Schranke	Zeit in Minuten
(1)-(6)	2920	29934	18,0000	94,44%	35,0000	1000
(1)-(6),$(7.14)_{V,4}$	2920	29985	33,0000	6,06%	35,0000	1000

Tabelle 13: Ganzzahlige lineare Programme für Queen7_7

9.3 Ergebnisse für das zweite Modell

Wir wenden das zweite Modell zuerst auf den Graphen Myciel3 an. Dieser Graph ist allerdings dreiecksfrei und enthält somit keine Cliquen mit 3 oder mehr Knoten. Die Varianten (V1) und (V2) sind somit identisch.

Neben-bedingungen	Anzahl Variablen	Anzahl Nebenb.	Lösung des GLP	Zeit in Sek.
(V1)	249	2096	5,0000	2379,81
(V1) mit (E)	257	2201	5,0000	479,10

Tabelle 14: Zweites Modell: ganzzahlige lineare Programme für Myciel3

Verglichen zum ganzzahligen linearen Programm des ersten Modells (4,5s) hat das zweite Modell 100 mal so lange gebraucht. Das überascht auch nicht, da das zweite Modell in diesem Fall $249 - 159 = 90$ Variablen und $2201 - 406 = 1795$ Nebenbedingungen mehr hat.

Wenden wir jetzt das zweite Modell auf den Graphen Queen5_5 an: Da jeder Knoten bei diesem Graphen ein Feld des 5×5- Schachbretts repräsentiert, bilden die Knoten jeder Zeile eine Clique. Analoges gilt für die Spalten und Diagonalen. Bei Variante (V3) reicht es also aus, dass Q nur aus der Vereinigung von V und der Menge der Cliquen, die aus den Zeilen, Spalten oder Diagonalen gebildet werden, besteht. Mit 651 Variablen hat das ganzzahlige lineare Programm nur unwesentlich mehr Variablen als das ganzzahlige Programm des ersten Modells. Die Lösungen der ganzzahligen linearen Programme sind in Tabelle 15 zu finden.

Neben-bedingungen	Anzahl Variablen	Anzahl Nebenb.	Duale Schranke	Gap in %	Primale Schranke	Zeit in Minuten
(V2) mit (E)	2419	57621	12,0000	50%	18,0000	1000
(V3) mit (E)	651	15677	12,0000	50%	18,0000	1000

Tabelle 15: Zweites Modell: ganzzahlige lineare Programme für Queen5_5

Beide ganzzahligen Programme haben nach 1000 Minuten noch einen Unterschied von 50% zwischen der primalen und der dualen Schranke und sind somit deutlich langsamer als die ganzzahligen Programme des ersten Modells.

9.4 Diskussion der Ergebnisse

Das ganzzahlige Programm des ersten Modells ist deutlich schneller als das Programm des zweiten Modells. Das liegt einerseits an der Wahl der Variablen und andererseits an der Anzahl der Nebenbedingungen:

- Die Beziehungen zwischen den einzelnen Variablen sind im ersten Modell klarer bestimmt. Gilt z. B. im ersten Modell $k_{i,v} = 1$ für ein $v \in V$ und ein $i \in I_v$, so ist automatisch $k_{i,w} = 0$ für alle $w \in V \backslash \{v\}$ und $k_{j,v} = 0$ für alle $j \in I_v \backslash \{i\}$. Beim zweiten Modell sagt $d_{i,v}$ noch nichts über $d_{j,v}$ oder $d_{i,w}$ aus.
- Die große Anzahl an Nebenbedingungen des zweiten Modells liegt vor allem an Ungleichung (27). Denn von den 57621 Nebenbedingungen des ganzzahligen Programms für den Graphen $Q_{5,5}$, stammen allein 52910 Nebenbedingungen aus Ungleichung (27). Ohne diese Ungleichung würden nur noch 4711 Nebenbedingungen bleiben, was eher in der Größenordnung des ersten Modells liegt.

Aus diesen Gründen wurden nur zusätzliche gültige Ungleichungen für das erste Modell bestimmt.

Mit dem ganzzahligen Programm des ersten Modells, d.h. mit den Ungleichungen (1) bis (6), konnte die Pfadweite für die Graphen M_3, M_4, $Q_{5,5}$ und $Q_{6,6}$ exakt bestimmt werden. Bei den Graphen M_5 und $Q_{7,7}$ bleibt ein Abstand zwischen der primalen und der dualen Schranke.

In dieser Arbeit wurde eine Vielzahl gültiger Ungleichungen hergeleitet. Diese Menge an Ungleichungen ist allerdings zu groß, um die Ungleichungen alle zusammen ins LP zu laden. In der Rechenstudie wurde deswegen das lineare Programm immer nur für ausgewählte Ungleichungen gelöst. Dabei hat sich herausgestellt, dass die Ungleichungen mit den meisten y- Variablen auf der linken Seite die besten dualen Schranken liefern. Z.B. haben wir mit Ungleichung $(7.14)_{V,j}$ in allen Beispielen schnell eine sehr gute untere Schranke erhalten.

Dazu muss erwähnt werden, dass die Laufzeit von *Zimpl*, um die Ungleichungen $(7.14)_{V,j}$ zu erzeugen, exponentiell zu j steigt. Der Aufwand um die

$$\sigma_j(V,v) := \min_{\substack{w_1,\ldots,w_j \in V\backslash\{v\} \\ w_1,\ldots,w_j \text{ verschieden}}} \left| N[v] \cup \bigcup_{i=1}^{j} N[w_i] \right| - \delta(G) \qquad , v \in V$$

für größere Graphen zu berechnen, ist für große j zu hoch. Deswegen wurden nur Ungleichungen mit kleinem j hergeleitet.

In der Rechenstudie wurden nur die Ungleichungen erster und zweiter Ordnung und die Ungleichungen (7.14) und (7.15) betrachtet, denn der Aufwand, um alle Ungleichung z. B. aus (7.7) herzuleiten, ist zu groß. Trotzdem hat die lineare Relaxierung für die Graphen M_3, M_4 und $Q_{6,6}$ zu einer Lösung mit gleichem Wert wie die Pfadweite geführt.

Es sei noch bemerkt, dass der Graph G_4 aus Abbildung 13 mit nur 5 Knoten bereits 1894 Ungleichungen hat, die Facetten definieren.

§ 10 Schlusswort

In dieser Arbeit wurde das Pfadweite- Problem auf zwei verschiedene Arten modelliert. Dabei stellten sich die Vorzüge des ersten Modells klar heraus, weshalb nur gültige Ungleichungen für das erste Modell bestimmt wurden.

Im Folgenden wurden also viele gültige Ungleichungen für das erste Modell hergeleitet, denn beim Lösen von ganzzahligen linearen Programmen sind Kenntnisse über die ganzzahlige Hülle sehr wichtig. Dabei wurde zwischen Ungleichungen *erster Ordnung*, Ungleichungen *zweiter Ordnung* und Ungleichungen *mehrfacher Ordnung* unterschieden.

Es wurde bewiesen, dass die Ungleichungen erster Ordnung oft Facetten definieren. Desweiteren konnte mit Hilfe der Ungleichungen erster Ordnung die Ungleichungen zweifacher und mehrfacher Ordnung auf verschiedene Arten konstruiert werden. Dass viele dieser Ungleichungen sehr stark sind, zeigen die Lösungen der linearen Relaxierung. Für zwei Klassen der Ungleichungen mehrfacher Ordnung wurde zudem bewiesen, unter welchen Bedingungen sie Facetten definieren.

Eine Erweiterung von Ungleichung (7.7) ist zudem denkbar: statt einer gewöhnlichen Menge könnte T als Multimenge gewählt werden. Dies ist ohne große Änderungen machbar.

Die Pfadweite von kleineren Graphen, wie M_3, M_4, $Q_{5,5}$ und $Q_{6,6}$, also Graphen die bis zu 36 Knoten enthalten, konnte mit dem ganzzahligen linearen Programm schnell und exakt berechnet werden. Bei den größeren Graphen M_5 und $Q_{7,7}$ konnten, mit den zusätzlich hergeleiteten gültigen Ungleichungen, gute untere Schranken bestimmt werden.

Bei den größeren Graphen ist die Menge der Ungleichungen zu groß, um die Ungleichungen alle zusammen ins lineare Programm zu laden. Wenn es gelingt die stärksten dieser Ungleichungen zu separieren, so kann man unter Anwendung des Schnittebenverfahrens auch die Pfadweite größerer Graphen bestimmen.

Literatur

[1] T. Achterberg: *SCIP: Solving constraint integer programs,* Mathematical Programming Computation, Vol. 1, No. 1, (2009), pp. 1-41, http://scip.zib.de/

[2] M. Aigner, G. M. Ziegler: *Das Buch der Beweise,* Springer- Verlag, 3. Auflage, (2010), p. 299.

[3] S. Arnborg, D. G. Corneil, und A. Proskurowski: *Complexity of finding embeddings in a k-tree,* SIAM J. Algebraic Discrete Methods, Vol. 8, No. 2, (1987), pp. 277-284.

[4] H.L Bodlaender: *A tourist guide through treewidth,* Acta Cybernetica, Vol. 11, (1993), pp. 1-21.

[5] H. L. Bodlaender: *A linear-time algorithm for finding tree-decompositions of small treewidth,* SIAM J. Computing, Vol. 25, No. 6, (1996), pp. 1305-1317.

[6] H. L. Bodlaender: *A partial k-arboretum of graphs with bounded treewidth,* Theoretical Computer Science, Vol. 209, No. 1-2, (1998), pp. 1-45.

[7] H. L. Bodlaender, A. M. C. A. Koster: *Treewidth computations I. Upper bounds,* Information and Computation, Vol. 208, No. 3, (2010), pp. 259-275.

[8] H. L. Bodlaender, A. M. C. A. Koster: *Treewidth computations II. Lower bounds,* Information and Computation, Vol. 209, No. 7, (2011), pp. 1103-1119.

[9] H. L. Bodlaender, R. H. Möhring: *The Pathwidth and Treewidth of Cographs,* SIAM Journal on Discrete Mathematics, Vol. 6, No. 2, (1993), pp. 181-188.

[10] T. Christof, A. Löbel: *PORTA - **PO**lyhedron **R**epresentation **T**ransformation **A**lgorithm,* http://typo.zib.de/opt-long_projects/Software/Porta/

[11] V. Chvátal: *Edmonds polytopes and a hierarchy of combinatorial problems,* Discrete Mathematics, Vol. 4, No. 4, (1973), pp. 305-337.

[12] R. J. Dakin: *A tree- search algorithm for mixed integer programming problems,* The Computer Journal, Vol. 8, No. 3, (1965), pp. 250-255.

[13] G. B. Dantzig: *Maximization of a linear function of variables subject to linear inequalities,* Activity Analysis of Production and Allocation, Cowles Commission Monograph No. 13, John Wiley & Sons Inc., (1951), pp. 339-347.

[14] G. B. Dantzig, R. Fulkerson und S. Johnson: *Solution of a large-scale traveling-salesman problem,* Operations Research, Vol. 2, No. 4 (1954), pp. 393-410.

[15] W. Domschke, A. Drexl: *Einführung in Operations Research,* Springer- Verlag, 6. Auflage (2004), pp. 120-156.

[16] R. E. Gomory: *Outline of an algorithm for integer solutions to linear programs,* Bulletin of the American Mathematical Society, Vol. 64, No. 5, (1958), pp. 275-278.

[17] M. Grötschel, L. Lovász, A. Schrijver: *The ellipsoid method and its consequences in combinatorial optimization,* Combinatorica, Vol. 1, No. 2, (1981), pp. 169-197.

[18] F. Gurski, I. Rothe, J. Rothe und E. Wanke: *Exakte Algorithmen für schwere Graphenprobleme,* eXamen.press, Springer- Verlag (2010), pp. 259-284.

[19] L. M. Kirousis, C. H. Papadimitriou: *Searching and pebbling,* Journal Theoretical Computer Science, Vol. 47, No. 2, (1986), pp. 205-218.

[20] J. Kneis, D. Mölle, S. Richter und P. Rossmanith: *A bound on the pathwidth of sparse graphs with applications to exact algorithms,* SIAM Journal on Discrete Mathematics, Vol. 23, No 1, (2008), pp. 407-427.

[21] T. Koch: *Rapid Mathematical Prototyping,* Technische Universität Berlin, (2004), http://zimpl.zib.de/

[22] A. M. C. A. Koster: *Vorlesung zur Ganzzahligen Linearen Optimierung,* RWTH Aachen, (Sommersemester 2009).

[23] A. H. Land, A. G. Doig: *An automatic method of solving discrete programming problems,* Econometrica, Vol. 28, No. 3, (1960), pp. 497-520.

[24] A. LaPaugh: *Recontamination does not help search a graph,* Tech. Report 335, Dept. of Elec. Eng. and Comp. Sci., Princeton Univ. (1982).

[25] S. Neuhaus: *Eigenschaften kleinster dominierender Mengen und Dominanzzahlen von Damengraphen,* Dissertation zur Erlangung des Doktorgrades, Verlag Dr. Hut, München, (2009), p. 5.

[26] M. Padberg, G. Rinaldi: *Optimization of a 532-city symmetric traveling salesman problem by branch and cut ,* Operations Research Letters, Vol. 6, Nr. 1 (März 1987), pp. 1-7.

[27] T. D. Parsons: *Pursuit- Evasion in a Graph,* In: Theory and Application of Graphs, Y. Alavi and D. R. Lick, Springer-Verlag, Lecture Notes in Mathematics, Vol. 642, (1978), pp. 426-441.

[28] S. Ramachandramurthi: *Algorithms for VLSI Layout Based on Graph Width Metrics,* PhD thesis, Computer Science Department, University of Tennessee, Knoxville, Tennessee, USA, (1994).

[29] S. Ramachandramurthi: *A lower bound for treewidth and its consequences,* In E. W. Mayr, G. Schmidt, and G. Tinhofer, editors, Proceedings of the 20th International Workshop on Graph- Theoretic Concepts in Computer Science, WG'94, Springer Verlag, Lecture Notes in Computer Science, Vol. 903, (1995), pp. 14-25.

[30] S. Ramachandramurthi: *The structure and number of obstructions to treewidth,* SIAM Journal on Discrete Mathematics, Vol. 10, No. 1, (1997), pp. 146-157.

[31] N. Robertson, P. D. Seymour: *Graph minors. II. Algorithmic aspects of tree-width,* Journal of Algorithms, Vol. 7, No. 3, (1986), pp. 309-322.

[32] D. J. Rose: *On simple characterization of k-trees,* Discrete Mathematics, Vol. 7, No. 3-4, (1974), pp. 317-322.

[33] A. Schrijver: *On cutting planes,* Annals of Discrete Mathematics, Vol. 9, (1980) pp. 291-296.

[34] P. D. Seymour, R. Thomas: *Graph searching, and a min- max- theorem for tree- width,* Journal of Combinatorial Theory, Series B, Vol. 58, No. 1, (1993), pp. 22-33.

[35] L. Sunil Chandran, C. R. Subramanian: *Girth and treewidth,* Journal of Combinatorial Theory, Series B, Vol. 93 , No. 1, (2005), pp. 23-32.

Eidesstattliche Erklärung

Ich versichere hiermit, dass ich die vorliegende Arbeit selbständig und ohne Benutzung anderer als der angegebenen Hilfsmittel angefertigt habe. Alle Stellen, die wörtlich oder sinngemäß aus veröffentlichten und nicht veröffentlichten Schriften anderer entnommen sind, sind als solche kenntlich gemacht. Die Arbeit ist in gleicher oder ähnlicher Form noch nicht als Prüfungsarbeit eingereicht worden.

Aachen, den 21. Januar 2012 .....

(Tom Rihm)