

Primale Separierung für Max-Cut und gradbeschränkte minimale Spannbäume

Diplomarbeit

von

Markus Behle

angefertigt im
Institut für Informatik der
UNIVERSITÄT ZU KÖLN

unter Anleitung von

Prof. Dr. Michael Jünger

Köln, Februar 2003

Inhaltsverzeichnis

1	Einleitung	5
1.1	Vorwort	5
1.2	Begriffe aus der Graphentheorie	7
1.3	Hilfsmittel aus der Polyedertheorie	8
2	Primale Separierung	11
2.1	Unterschied zwischen primaler und Standard-Separierung . . .	11
2.2	0/1 Separierung	14
2.3	Entwicklung der primalen Separierung	15
2.4	Ein primaler Schnittebenenalgorithmus	16
2.5	Resultate zur primalen Separierung	24
2.5.1	Stabile Menge Problem	24
2.5.2	Rucksack Problem	26
2.5.3	Handlungsreisendenproblem	26
3	Max-Cut	31
3.1	Das Max-Cut Problem	31
3.2	Anwendung der Hilfsmittel aus der Polyedertheorie	32
3.3	Die Umschalt-Operation	33
3.4	Anwendung der Umschaltoperation	37
3.5	Primale Separierung für Ungleichungsklassen	37
3.5.1	Dreiecksungleichungen	38
3.5.2	Kreisungleichungen	39
3.5.3	Clique-Ungleichungen	42
3.5.4	Hypermetrische Ungleichungen	46
4	Gradbeschränkte minimale Spannbäume	53
4.1	Die Problemstellung	53
4.2	Resultate aus der Polyedertheorie	55
4.3	Weitere gültige Ungleichungen	55
4.4	Primale Separierung für Ungleichungsklassen	56
4.4.1	Triviale Separierungen	56
4.4.2	Zusammenhangsungleichungen	56
4.4.3	Spezielle Kreis-Eliminierungsungleichungen	58
4.4.4	Kreis-Eliminierungsungleichungen	60
4.5	Konstruktion eines gradbeschränkten Spannbaumes	64

5	Resultate	65
5.1	Theoretische Ergebnisse	65
5.2	Rechenresultate	67
6	Zusammenfassung & Ausblick	73
	Abbildungsverzeichnis	75
	Tabellenverzeichnis	77
	Literaturverzeichnis	79
	Danksagung	83
	Eidesstattliche Versicherung	85

Kapitel 1

Einleitung

1.1 Vorwort

Im vergangenen Jahr riefen Letchford & Lodi [26] dazu auf, die primale Separierung erneut zu untersuchen. Diesem Aufruf soll in dieser Diplomarbeit gefolgt werden. Wir betrachten dazu das Max-Cut und das gradbeschränkte minimale Spannbaum Problem.

Seit einigen Jahren werden Schnittebenenverfahren oder Branch & Cut zur Lösung von ganzzahligen und gemischtganzzahligen linearen Programmen sowie zur Lösung kombinatorischer Optimierungsprobleme eingesetzt. Ein Schnittebenenalgorithmus arbeitet mit einer Relaxierung des zu lösenden Problems. Diese wird mit Hilfe des dualen Simplex-Algorithmus optimal gelöst. Dadurch kann bei einem Maximierungs- bzw. Minimierungsproblem eine obere bzw. untere Schranke für die Optimallösung angegeben werden. Im Laufe des Verfahrens wird die Relaxierung iterativ durch Hinzunahme weiterer Schnittebenen verbessert, bis deren Lösung für das Gesamtproblem zulässig ist. In dieser Arbeit werden wir alle Verfahren, die mit dem dualen Simplex-Algorithmus arbeiten, Standard-Verfahren nennen.

Beim primalen Ansatz arbeitet man immer mit einer für das Gesamtproblem zulässigen Lösung. Diese wird nach Hinzufügen weiterer Schnittebenen durch den Einsatz des primalen Simplex-Algorithmus verbessert, um Optimalität zu erreichen. Der Vorteil der primalen Methode ist also, dass man selbst bei Abbruch des Algorithmus eine für das gesamte Problem zulässige Lösung hat. Eine Schranke zur Abschätzung der momentan besten bekannten Lösung erhält man nicht. Sie kann jedoch leicht durch Lösung der zum Zeitpunkt des Abbruchs des Algorithmus bestehenden Relaxierung berechnet werden. Padberg & Hong [34] haben als Erste Anfang der 80er Jahre einen Schnittebenenalgorithmus für das Handlungsreisendenproblem angegeben, der als primaler Schnittebenenalgorithmus verstanden werden kann.

In der Veröffentlichung von Letchford & Lodi wurde darauf hingewiesen, dass fast alle Lösungsverfahren, die in der modernen Literatur zu finden sind, die Standard-Variante benutzen. Das Problem, hier passende Schnittebenen zu finden, die man zu der Relaxierung hinzufügt, nennt man Standard-Separierungsproblem. Benutzt man einen primalen Schnittebenenalgorithmus, so muss man, um hinzuzufügende Schnittebenen zu finden, ein so genanntes pri-

males Separierungsproblem lösen. Padberg & Grötschel [33] zeigten, dass man vom Komplexitätstheoretischen Standpunkt aus das Standard-Separierungsproblem nie schneller lösen kann als das entsprechende primale Separierungsproblem. Letchford & Lodi beschrieben in einigen Fällen sogar Verfahren, die praktisch schneller primal separieren als die entsprechenden Standard-Verfahren. Aus diesem Grund riefen sie dazu auf, primale Schnittebenenverfahren und damit die primale Separierung nochmals zu untersuchen.

Die Erläuterungen der Begriffe, die zu solch einer Untersuchung notwendig sind, finden sich in den Abschnitten des 1. Kapitels. Im Kapitel 2 werden die primale und die Standard-Separierung definiert und einige bekannte Komplexitätstheoretische Resultate präsentiert. Die Anwendung der primalen Separierung wird im Rahmen eines primalen Schnittebenenalgorithmus verdeutlicht.

Das Ising-Spin-Glas Modell wird in der statistischen Physik oft verwendet. Die Berechnung der Grundzustände solcher Modelle kann durch Lösen eines Max-Cut Problems durchgeführt werden. Auch bei der Entwicklung von elektronischen Halbleiterchips treten Probleme auf, die sich auf Max-Cut Probleme zurückführen lassen. Aus diesem Grund haben wir in Kapitel 3 die primale Separierung für das Max-Cut Problem genauer untersucht. Hier stellte sich heraus, dass im Vergleich zur Standard-Separierung keine Verbesserung bzgl. der Laufzeiten für die einzelnen Separierungen erzielt werden konnten.

Günther Raidl stellte uns auf einem gemeinsamen Treffen das gradbeschränkte minimale Spannbaum Problem vor, welches häufig bei der Planung von Computernetzwerken auftritt. Er hatte bereits einen Standard Branch & Cut Algorithmus für dieses Problem implementiert. Bei unserer Untersuchung dieses Problems, die sich in Kapitel 4 findet, konnten wir für einige Ungleichungsklassen primale Separierungen herleiten, die schneller sind als die bekannten Standard-Separierungen. Dies motivierte uns, einen primalen Schnittebenenalgorithmus für das gradbeschränkte minimale Spannbaum Problem zu implementieren. Unseres Wissens nach ist dies der erste primale Schnittebenenalgorithmus für dieses Problem.

Im 5. Kapitel werden die gewonnenen theoretischen Erkenntnisse zusammengefasst. Um unseren Algorithmus mit den aus der Literatur bekannten Verfahren vergleichen zu können, berechneten wir mehrere Instanzklassen, die von anderen Autoren veröffentlicht wurden. Die dabei erzielten Rechenresultate stellen wir vor. Für einige Instanzklassen zeigt sich, dass die Qualität der Lösungen unseres Algorithmus mit denen der besten bisher bekannten Verfahren vergleichbar und teilweise sogar besser ist.

Eine Zusammenfassung dieser Diplomarbeit und ein Ausblick auf die Entwicklung von primalen Branching Regeln wird im abschließenden Kapitel 6 gegeben.

1.2 Begriffe aus der Graphentheorie

Sowohl das Max-Cut als auch das gradbeschränkte minimale Spannbaum Problem sind mittels Graphen definiert. Die Graphentheorie stellt einige Begriffe zur Untersuchung dieser Probleme bereit, die hier kurz zusammengefasst werden.

Gegeben sei ein endlicher, ungerichteter Graph $G = (V, E)$ mit Knotenmenge V und Kantenmenge E . Diesen assoziiert man mit dem reellen Vektorraum \mathbb{R}^m , bei dem die Komponenten der Vektoren durch die Elemente von E indiziert werden. Die Anzahl der Knoten wird mit $n := |V|$ und die Anzahl der Kanten mit $m := |E|$ abgekürzt. Typischerweise nimmt man an, dass die Knotenmenge $V = \{1, 2, \dots, n\}$ ist, und gibt dann die Kanten ij mit $i, j \in V$ als Paare dieser Knoten an. Da der Graph ungerichtet ist, gilt dabei $ij = ji$. Existieren alle $\binom{n}{2}$ möglichen Kanten ij mit $i \neq j$, so bezeichnet man den Graphen G als *vollständig*. Den vollständigen Graphen mit n Knoten nennt man K_n . Eine Teilmenge der Knotenmenge V , die einen vollständigen Subgraph von G bildet, nennt man *Clique*.

Die zu einer gegebenen Teilmenge der Kanten $A \subseteq E$ inzidenten Knoten bilden die Menge

$$V(A) := \{i \in V \mid ij \in A\}.$$

Für jede Teilmenge $S \subseteq V$ der Knotenmenge fasst man die Kanten des von S induzierten Subgraphen zusammen als

$$E(S) := \{ij \in E \mid i \in S, j \in S\}.$$

Im Gegensatz dazu nennt man die Menge der Kanten mit einem Ende in S und dem anderen in $V \setminus S$ den durch S induzierten *Schnitt* (im Englischen „cut“). Man schreibt dies als

$$\delta(S) := \{ij \in E \mid i \in S, j \notin S\}.$$

Besteht die Menge S nur aus einem Knoten i , so schreibt man verkürzt $\delta(i)$ statt $\delta(\{i\})$. Die Anzahl $|\delta(i)|$ der Schnittkanten eines Knotens i bezeichnet man als *Knotengrad* von i .

Die Kantenfolge $(v_1, v_2), (v_2, v_3), \dots, (v_{r-1}, v_r)$ bildet einen *Weg* von v_1 nach v_r , wenn alle Knotenpaare v_k, v_l mit $k \neq l$ und $k, l \in \{1, \dots, r\}$ paarweise verschieden sind. Existiert zusätzlich zu diesem Weg eine Kante von v_r nach v_1 , und nimmt man diese zur Kantenfolge hinzu, so erhält man einen *Kreis* $(v_1, v_2), \dots, (v_{r-1}, v_r), (v_r, v_1)$.

Ein Graph G heißt *zusammenhängend*, wenn für alle Knotenpaare i, j mit $i \neq j$ und $i, j \in \{1, \dots, n\}$ ein Weg von i nach j existiert. Ist ein Graph G nicht zusammenhängend, so zerfällt er in mehrere zusammenhängende Subgraphen, die man *Zusammenhangskomponenten* nennt. Die Anzahl der Zusammenhangskomponenten eines Graphen $G = (V, E)$ bezeichnet man mit $s(V, E)$. Man nennt einen Graphen $G = (V, E)$ *2-fach zusammenhängend*, wenn G zusammenhängend ist und jeder der aus G durch Entfernung eines beliebigen Knotens i mit den dazugehörigen adjazenten Kanten gebildeten Graphen $G' = (V \setminus \{i\}, E \setminus \delta(i))$ zusammenhängend bleibt.

Man nennt jeden zusammenhängenden und kreisfreien Subgraphen T eines Graphen G einen *Baum*. Ist G zusammenhängend und definiert T einen Baum in G , der alle n Knoten enthält, so handelt es sich bei T um einen *Spannbaum*.

Für einen Schnitt $\delta(S)$, $S \subseteq V$ definiert man den *charakteristischen Vektor* $\chi^{\delta(S)} \in \mathbb{R}^m$ wie folgt:

$$\chi_e^{\delta(S)} := \begin{cases} 1 & \text{falls } e \in \delta(S) \\ 0 & \text{sonst} \end{cases}$$

Diesen Vektor nennt man auch *Inzidenzvektor* des Schnittes $\delta(S)$. Durch obige Definition erhält man eine Bijektion zwischen den Schnitten von G und deren $\{0,1\}$ Inzidenzvektoren im \mathbb{R}^m . Ebenso definiert man für jede andere Teilmenge $T \subseteq E$ den dazugehörigen charakteristischen Vektor:

$$\chi_e^T := \begin{cases} 1 & \text{falls } e \in T \\ 0 & \text{sonst} \end{cases}$$

Lässt man für T nur Spann bäume zu, so ergibt sich analog zu oben eine Bijektion zwischen den Spann bäumen von G und deren $\{0,1\}$ Inzidenzvektoren im \mathbb{R}^m .

Gegeben sei nun ein endlicher, ungerichteter Graph $G = (V, E)$, auf dessen Kanten eine Kostenfunktion $c : E \rightarrow \mathbb{R}$ definiert sei. Analog kann für die Knoten eine Gewichtsfunktion $w : V \rightarrow \mathbb{R}$ vorliegen. In beiden Fällen spricht man von einem *gewichteten* Graphen, den man oft als Paar (G, c) oder (G, w) angibt. Die Kosten einer Kante bezeichnet man mit c_e oder c_{ij} , die eines Knotens mit w_i . Mit den Kosten einer Teilmenge $T \subseteq E$ meint man die Summe der Kantenkosten der Elemente von T , d.h.

$$c(T) := \sum_{e \in T} c_e$$

Die Kosten eines Schnittes ergeben sich somit als

$$c(\delta(S)) := \sum_{e \in \delta(S)} c_e = \sum_{i \in S, j \notin S} c_{ij}.$$

Analog ist mit dem Gewicht einer Teilmenge $S \subseteq V$ die Summe der Knotengewichte der Elemente von S gemeint, also

$$w(S) := \sum_{i \in S} w_i.$$

1.3 Hilfsmittel aus der Polyedertheorie

Sowohl bei Branch & Cut als auch bei Schnittebenenverfahren benutzt man Mittel der polyedrischen Kombinatorik, weswegen hier nun einige Begriffe aus der Polyedertheorie erläutert werden.

Falls $X \subseteq \mathbb{R}^m$ so bezeichnet man mit

$$\text{cone}(X) := \left\{ \sum_{x \in X} \lambda_x x \mid \lambda_x \in \mathbb{R}^{\geq 0} \text{ für alle } x \in X \right\}$$

die *konische Hülle* und mit

$$\text{conv}(X) := \left\{ \sum_{x \in X} \lambda_x x \mid \lambda_x \in \mathbb{R}^{\geq 0} \text{ für alle } x \in X \text{ und } \sum_{x \in X} \lambda_x = 1 \right\}$$

die *konvexe Hülle* von X .

Sei \mathcal{I} eine nichtleere endliche Klasse von Teilmengen der Kantenmenge E von G . Die konvexe Hülle der Inzidenzvektoren aller $T \in \mathcal{I}$ ist ein *Polytop*:

$$P^{\mathcal{I}}(G) := \text{conv}\{\chi^T \mid T \in \mathcal{I}\}$$

Analog wird ein *Konus* definiert:

$$C^{\mathcal{I}}(G) := \text{cone}\{\chi^T \mid T \in \mathcal{I}\}$$

Offensichtlich gilt $P^{\mathcal{I}}(G) \subseteq C^{\mathcal{I}}(G)$. Da die Ecken von $P^{\mathcal{I}}(G)$ genau die Inzidenzvektoren χ^T sind und laut Definition $\chi^T \in \{0, 1\}^m$ gilt, spricht man auch von einem *0/1 Polytop*. Mit Hilfe solch eines Polytops und eines Vektors $c \in \mathbb{R}^m$ lässt sich ein zugehöriges Programm formulieren:

$$\begin{aligned} \max \quad & c^T x \\ \text{s.t.} \quad & x \in P^{\mathcal{I}} \quad (\text{P}) \end{aligned}$$

Zu jeder Optimallösung dieses Programms gibt es eine Ecke χ^T des Polytops, die den gleichen Zielfunktionswert wie die Optimallösung hat. Um Techniken der linearen Programmierung zur Lösung dieses Programms anwenden zu können, muss man das Polytop $P^{\mathcal{I}}$ als Lösungsmenge eines endlichen Ungleichungssystems darstellen:

$$P^{\mathcal{I}} = \{x \in \mathbb{R}^m \mid Ax \leq b\}$$

Diese zweite Darstellung existiert nach Theoremen von Minkowski und Weyl. Um den Zusammenhang zwischen Polytopen und linearen Gleichungssystemen herzustellen, verwendet man folgende Definitionen:

Für $a \in \mathbb{R}^m, a_0 \in \mathbb{R}$ heißt $H = \{x \in \mathbb{R}^m \mid a^T x \leq a_0\}$ *Halbraum*. Ein *Polyeder* ist der Durchschnitt endlich vieler Halbräume:

$$P = \{x \in \mathbb{R}^m \mid Ax \leq b\} \text{ mit } A \in \mathbb{R}^{k \times m}, b \in \mathbb{R}^k$$

Wenn eine Konstante R existiert, so dass $\max_{1 \leq i \leq m} |x_i| \leq R$ für alle $x = (x_i)_{i=1}^m \in P$, dann ist das Polyeder beschränkt und damit ein Polytop.

Somit lässt sich (P) auch als *lineares Programm* schreiben

$$\begin{aligned} \max \quad & c^T x \\ \text{s.t.} \quad & Ax \leq b \quad (\text{LP}) \end{aligned}$$

Es gibt einen auf der Fourier-Motzkin Elimination basierenden endlichen Algorithmus, der das Polytop $P^{\mathcal{I}}$ von der einen in die andere Darstellungsform überführt. Allerdings ist dieser wegen seiner nichtpolynomiellen Laufzeit nur für kleine Probleminstanzen praktikabel. Wenn man zu jeder beliebigen Instanz eines \mathcal{NP} -schwierigen Problems alle linearen Ungleichungen, die das assoziierte Polytop definieren, in polynomieller Zeit aufzählen könnte, gäbe es

mit Hilfe der polynomiellen Ellipsoid-Methode ein Verfahren, um das Problem in polynomieller Zeit zu lösen. Dies würde bedeuten, dass $\mathcal{P} = \mathcal{NP}$ ist. Zum heutigen Zeitpunkt vermutet man allerdings, dass $\mathcal{P} \neq \mathcal{NP}$ gilt. Also versucht man, das Problem zu relaxieren, d.h., man benutzt nur ein Teilsystem von Ungleichungen:

$$\begin{aligned} \max \quad & c^T x \\ \text{s.t.} \quad & x \in P \\ & P \supseteq P^{\mathcal{I}} \quad (\text{LP}_{\text{relax}}) \end{aligned}$$

Für die Auswahl der Ungleichungen, die bei der Relaxierung benutzt werden, sind folgende Definitionen hilfreich:

Eine Ungleichung $a^T x \leq a_0$ ist *gültig* bzgl. einer Teilmenge $X \subseteq \mathbb{R}^m$ gdw.

$$X \subseteq \{x \in \mathbb{R}^m \mid a^T x \leq a_0\}.$$

Sei P ein Polyeder. $F \subseteq P$ heißt *Seitenfläche* von P , falls es eine für P gültige Ungleichung $a^T x \leq a_0$ gibt, so dass

$$F = \{x \in P \mid a^T x = a_0\}.$$

Falls $F \neq P$, so ist F eine *echte* Seitenfläche. Die *Dimension* eines Polyeders ist

$$\dim(P) := \text{affrang}(P) - 1,$$

wobei $\text{affrang}(P)$ die maximale Anzahl affin unabhängiger Punkte in P ist. Es gilt allgemein:

$$0 \in P \Rightarrow \dim(P) = \text{rang}(P),$$

wobei $\text{rang}(P)$ die maximale Anzahl linear unabhängiger Punkte in P ist. Eine Seitenfläche F der $\dim(F) := \text{affrang}(F) - 1 = 0$, also $F = \{x\}$, heißt *Ecke* von P . Ist F eine nicht leere, echte Seitenfläche, die $\dim(P)$ viele affin unabhängige Punkte enthält, so spricht man von einer *Facette*. Lässt man bei der Beschreibung eines Polytops eine Facette weg, dann ändert sich das Polytop. Somit sind Facetten die Ungleichungen, die bevorzugt zur Relaxierung verwendet werden.

Bei Schnittebenenverfahren und Branch & Cut hat man zu jedem Zeitpunkt ein Teilsystem von für $P^{\mathcal{I}}$ gültige Ungleichungen. Dieses soll in gewissen Schritten der obigen Verfahren durch weitere geeignete Ungleichungen ergänzt werden. Durch Lösen des so genannten Separierungsproblems kann man solche Ungleichungen finden. Da die Geschwindigkeit obiger Verfahren unter anderem von der Geschwindigkeit der zugrunde liegenden Separierung abhängt, soll diese hier nun weiter untersucht werden.

Kapitel 2

Primale Separierung

Zu Anfang dieses Kapitels werden in Abschnitt 2.1 die primale und die Standard-Separierung definiert und verglichen. Anschließend werden die Definitionen in 2.2 auf 0/1 Polytope übertragen und die dazugehörigen komplexitätstheoretischen Resultate präsentiert. Abschnitt 2.3 gibt einen kurzen Überblick über die Entwicklung von Schnittebenenverfahren in den letzten 45 Jahren mit besonderem Schwerpunkt auf primale Verfahren und die primale Separierung. Der zentrale Abschnitt 2.4 dieses Kapitels stellt unseren primalen Schnittebenenalgorithmus vor. Abschließend werden in 2.5 einige Resultate zur primalen Separierung für Probleme zusammengefasst, die in Letchford & Lodi [27] betrachtet werden, nämlich das Stabile Menge Problem, das Rucksack Problem und das Handlungsreisendenproblem.

2.1 Unterschied zwischen primaler und Standard-Separierung

Bei Schnittebenenverfahren oder auch Branch & Cut tritt in den Standard-Varianten die Situation auf, dass ein Polytop P und ein Punkt $x^* \in \mathbb{R}^m$ gegeben sind. Es soll eine Schnittebene gefunden werden, die für P gültig ist und den Punkt x^* abschneidet. Abbildung 2.1 zeigt solch eine Schnittebene. Diese wird dem aktuellen Ungleichungssystem hinzugefügt, um im nächsten

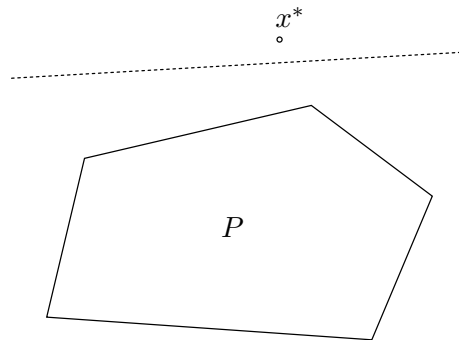


Abbildung 2.1: Die gestrichelte Linie zeigt eine Standard-Schnittebene, die für das Polytop P gültig ist und den Punkt x^* abschneidet.

Durchlauf einen von x^* verschiedenen Punkt zu finden. Das Problem, solch eine Schnittebene zu finden, nennt man

Standard-Separierungsproblem: Gegeben sei $x^* \in \mathbb{R}^m$. Finde eine Ungleichung, die für P gültig ist und durch x^* verletzt wird, oder beweise, dass solch eine nicht existiert.

Grötschel, Lovász & Schrijver [20], Karp & Papadimitriou [23] sowie Padberg & Rao [35] zeigten unter geeigneten Voraussetzungen, dass ein Optimierungsproblem genau dann in polynomieller Zeit gelöst werden kann, wenn das Separierungsproblem für das gesamte Polytop P , das die konvexe Hülle aller zulässigen Lösungen beschreibt, in polynomieller Zeit gelöst werden kann. Ist das zu betrachtende Optimierungsproblem \mathcal{NP} -schwierig, so ist damit auch das zugehörige Separierungsproblem \mathcal{NP} -schwierig. In diesem Fall betrachtet man nicht das ganze Polytop P sondern nur eine Klasse von gültigen Ungleichungen, die eine partielle Beschreibung des Polytops P darstellen. Für viele der bekannten kombinatorischen Optimierungsprobleme kennt man solche Ungleichungsklassen. Das Separierungsproblem lautet dann wie folgt:

Standard-Separierungsproblem für eine Klasse \mathcal{F} von Ungleichungen: Gegeben sei $x^* \in \mathbb{R}^m$. Finde ein Mitglied von \mathcal{F} , das durch x^* verletzt wird, oder beweise, dass solch eines nicht existiert.

Die Schwierigkeit dieses Separierungsproblems hängt hierbei von der Klasse der Ungleichungen und dem Optimierungsproblem ab. Für einige Klassen gibt es polynomielle Separierungsverfahren, wohingegen man von anderen weiß, dass ihre Separierung \mathcal{NP} -schwierig ist.

Bei der primalen Variante hat man zusätzlich zu einem Vektor $x^* \in \mathbb{R}^m$ noch einen weiteren Vektor $\bar{x} \in \mathbb{R}^m$ gegeben. Dieser repräsentiert die momentan beste zulässige Lösung und ist aufgrund der Arbeitsweise obiger Verfahren ein Eckpunkt des Polytops P . Fügt man nun dem Gleichungssystem eine Schnittebene im Sinne der oben genannten Standard-Separierung hinzu, so wäre der neu berechnete Punkt $\hat{x} = \varepsilon x^* + (1 - \varepsilon)\bar{x}$ mit $0 < \varepsilon < 1$ nur eine konvexe Kombination von x^* und \bar{x} . Geht man davon aus, dass P ein 0/1 Polytop ist, dann wäre \hat{x} immer noch fraktional und man würde weder einen Fortschritt in Hinblick auf das Auffinden einer besseren Lösung noch auf den Beweis der dualen Zulässigkeit und damit Optimalität von \bar{x} erreichen. Um dies zu verhindern, verlangt man zusätzlich von der zu findenden Ungleichung, dass \bar{x} sie mit Gleichheit erfüllt. Anzumerken ist, dass diese intuitive Überlegung auch auf gemischte 0/1 Probleme übertragbar ist. Sie gilt allerdings nicht mehr für allgemeine ganzzahlige Probleme, denn eine konvexe Kombination von x^* und \bar{x} kann selbst dann ganzzahlig sein, wenn x^* es nicht ist.

Also gilt es nun, eine Schnittebene zu finden, auf der \bar{x} liegt und die x^* abschneidet. Solch eine primale Schnittebene ist in Abbildung 2.2 dargestellt. Das dazugehörige Problem heißt

Primales Separierungsproblem: Gegeben seien $x^* \in \mathbb{R}^m$ und \bar{x} , welcher ein Eckpunkt von P ist. Finde eine Ungleichung, die für P gültig ist, durch x^* verletzt wird und durch \bar{x} mit Gleichheit erfüllt wird, oder beweise, dass solch eine nicht existiert.

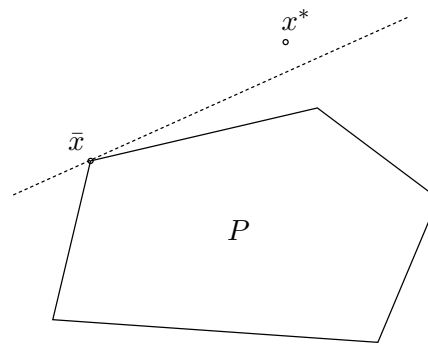


Abbildung 2.2: Eine für das Polytop P gültige primale Schnitt Ebene ist durch die gestrichelte Linie dargestellt. Sie hat die Eigenschaften, dass sie x^* abschneidet und \bar{x} auf ihr liegt.

Analog zu oben lautet die Definition für eine Klasse von für P gültige Ungleichungen:

Primales Separierungsproblem für eine Klasse \mathcal{F} von Ungleichungen: Gegeben seien $x^* \in \mathbb{R}^m$ und \bar{x} , welcher ein Eckpunkt von P ist. Finde ein Mitglied von \mathcal{F} , das durch x^* verletzt wird und durch \bar{x} mit Gleichheit erfüllt wird, oder beweise, dass solch eines nicht existiert.

Nach einer Idee von Padberg & Grötschel [33] kann die primale Separierung auf die Standard-Separierung reduziert werden kann. Man zeigt, dass eine Ungleichung das primale Separierungsproblem genau dann löst, wenn sie das Standard-Separierungsproblem für den Punkt $\varepsilon x^* + (1 - \varepsilon)\bar{x}$ löst, wobei ε eine kleine positive Zahl ist, deren Kodierungslänge polynomiell in der Kodierungslänge der Ungleichungen ist, die P definieren.

Diese Reduktion gilt sowohl für eine Klasse von Polytopen als auch für eine Klasse von Ungleichungen. Vom Komplexitätstheoretischen Standpunkt aus betrachtet bedeutet es, dass die primale Separierung nicht schwieriger als die Standard-Separierung ist. Bis jetzt ist allerdings nicht bekannt, ob die Rückrichtung gilt, d.h., ob beide polynomiell äquivalent sind. Es besteht also noch die Möglichkeit, dass die primale Separierung einfacher ist als die Standard-Separierung.

Das Separierungsproblem für eine Klasse \mathcal{F} von Ungleichungen besteht bei genauerer Betrachtung aus zwei Aufgaben:

1. Es soll ein Mitglied von \mathcal{F} gefunden werden, das den gegebenen Bedingungen entspricht.
2. Falls solch ein Mitglied nicht existiert, soll dafür der Beweis erbracht werden.

Wenn ein Separierungsalgorithmus kein verletztes Mitglied findet und zusätzlich beweist, dass es keines mehr geben kann, so spricht man von *exakter* Separierung. Für manche Ungleichungsklassen kennt man jedoch keinen polynomiellen Separierungsalgorithmus oder weiß, dass das Separierungsproblem \mathcal{NP} -schwierig ist. In diesen Fällen sucht man mittels Heuristiken nach verletzten Ungleichungen. Terminieren solche Verfahren, ohne ein verletztes Mitglied

zu finden, kann nicht garantiert werden, dass es wirklich kein verletztes Mitglied mehr gibt. Die Separierung ist also nicht exakt.

2.2 0/1 Separierung

Bei kombinatorischen Optimierungsproblemen besteht die Lösung aus der Auswahl von Elementen aus einer endlichen diskreten Menge. Hierzu zählen auch einige Optimierungsprobleme aus der Graphentheorie, wo es um die Auswahl von Knoten oder Kanten geht, wie z.B. bei Max-Cut oder gradbeschränkten minimalen Spannbäumen. Bei dem zu solch einem Problem gehörigen Polytop wissen wir, dass es ein 0/1 Polytop ist. Der Punkt \bar{x} ist ein Eckpunkt dieses Polytops, woraus folgt, dass $\bar{x} \in \{0, 1\}^m$. Man spricht hier von einem *0/1 Punkt*. Für diese speziellen Polytope definiert man analog zu oben folgende Separierungsprobleme:

Standard 0/1 Separierungsproblem: Gegeben sei $x^* \in \mathbb{R}^m$. Finde eine Ungleichung, die für P gültig ist und durch x^* verletzt wird, oder beweise, dass solch eine nicht existiert.

Standard 0/1 Separierungsproblem für eine Klasse \mathcal{F} von Ungleichungen: Gegeben sei $x^* \in \mathbb{R}^m$. Finde ein Mitglied von \mathcal{F} , das durch x^* verletzt wird, oder beweise, dass solch eines nicht existiert.

Primales 0/1 Separierungsproblem: Gegeben seien $x^* \in \mathbb{R}^m$ und \bar{x} , welcher ein 0/1 Eckpunkt von P ist. Finde eine Ungleichung, die für P gültig ist, durch x^* verletzt wird und durch \bar{x} mit Gleichheit erfüllt wird, oder beweise, dass solch eine nicht existiert.

Primales 0/1 Separierungsproblem für eine Klasse \mathcal{F} von Ungleichungen: Gegeben seien $x^* \in \mathbb{R}^m$ und \bar{x} , welcher ein 0/1 Eckpunkt von P ist. Finde ein Mitglied von \mathcal{F} , das durch x^* verletzt wird und durch \bar{x} mit Gleichheit erfüllt wird, oder beweise, dass solch eines nicht existiert.

Eisenbrand, Rinaldi & Ventura [13] lieferten den fehlenden Reduzierbarkeitsbeweis in einer Kette von Beweisen, welche zeigt, dass die primale 0/1 Separierung polynomiell äquivalent zur Standard 0/1 Separierung ist. Sie zeigten, dass das 0/1 Augmentierungsproblem auf die primale 0/1 Separierung reduziert werden kann. Beim 0/1 Augmentierungsproblem ist gefordert, zu einem gegebenen 0/1 Punkt eine bessere Lösung zu finden oder zu beweisen, dass dieser Punkt schon optimal ist. Wie schon gesehen, kann die primale 0/1 Separierung auf die Standard 0/1 Separierung reduziert werden. Ebenfalls wurde schon bemerkt, dass die Standard-Separierung und das Optimierungsproblem polynomiell äquivalent sind, was auch für die 0/1 Varianten gilt. Grötschel & Lovász [19] sowie Schulz, Weismantel & Ziegler [41] bewiesen, dass das 0/1 Optimierungsproblem und das 0/1 Augmentierungsproblem polynomiell äquivalent sind, womit sich die Kette der Reduktionsbeweise schließt. Abbildung 2.3 zeigt die zeitliche Entwicklung dieser Beweiskette.

Als wichtiges Resultat daraus lässt sich festhalten, dass die primale 0/1 Separierung genau dann \mathcal{NP} -schwierig ist, wenn das zugehörige Optimie-

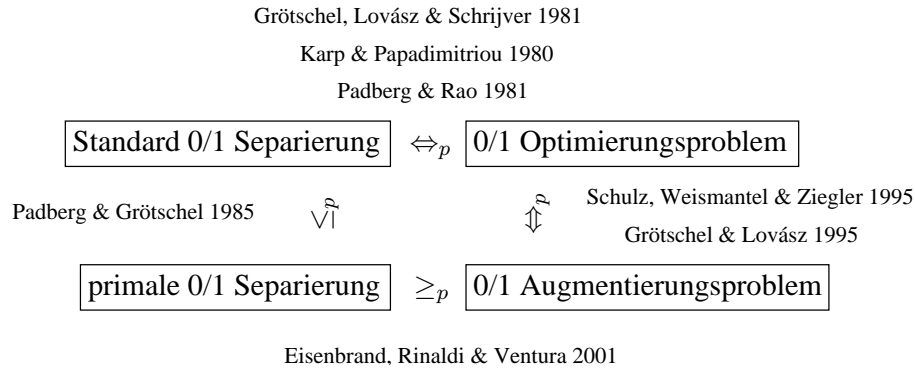


Abbildung 2.3: Entwicklung der Beweiskette zur polynomiellen Äquivalenz der primalen 0/1 Separierung und der Standard 0/1 Separierung.

rungsproblem \mathcal{NP} -schwierig ist. Für Probleme wie z.B. das Max-Cut Problem oder gradbeschränkte minimale Spannbäume steht damit aber immer noch die Möglichkeit offen, dass die primale Separierung für eine Klasse von Ungleichungen konzeptuell einfacher und dadurch praktisch schneller durchführbar ist als die Standard-Variante. Dies wird in den Kapiteln 3 und 4 untersucht, wobei verkürzt nur von Separierung statt von 0/1 Separierung gesprochen wird.

2.3 Entwicklung der primalen Separierung

Vergleicht man die Anzahl der Veröffentlichungen zur Standard-Separierung mit denen, die es zur primalen Separierung gibt, so erhält man den Eindruck, dass sich fast alle Arbeiten innerhalb dieses Bereiches auf die Standard-Separierung konzentrieren. Die primale Variante wurde bis auf einige Ausnahmen außer Acht gelassen. Dies hat wahrscheinlich eine historische Begründung.

Etwa zehn Jahre nach der Entwicklung des Simplex-Algorithmus durch Dantzig, genauer gesagt im Jahre 1958, leitete Gomory zum ersten Mal eine Schnittebene her, um sie in einem Schnittebenenalgorithmus zu verwenden. Er benutzte sie im dualen Kontext, was vermutlich dazu geführt hat, dass die Entwicklung und Verbesserung von Schnittebenenalgorithmen fast ausschließlich in Richtung der dualen Variante vorangetrieben wurde. Gomory leitete seine Schnitte aus den Zeilen des Simplex-Tableaus her. Sie waren allgemeiner Natur und damit nicht problemspezifisch. Anfang der 60er Jahre wurde Gomory's Methode von Ben-Israel & Charnes benutzt, um analog einen primalen Schnittebenenalgorithmus zu erstellen, der 1965 von Young verbessert wurde. Er war der erste primale Schnittebenenalgorithmus, der nur für ganzzahlige lineare Programme geschrieben wurde. Im Jahre 1968 wurde dieser Algorithmus durch Young selbst [46] vereinfacht. Der Algorithmus neigt typischerweise dazu, lange Abfolgen degenerierter Pivots zu erzeugen, bei denen, obwohl neue Schnitte hinzugefügt werden, die derzeit zulässige und ganzzahlige Lösung dieselbe bleibt. Glover [18] und Arnold & Bellmore [1, 2, 3] versuchten, dieses Problem zu beheben, und modifizierten den Algorithmus. Der letzte Verbesserungsvorschlag kam 1997 von Sharma & Sharma [42]. Die resultierenden Algorithmen können dennoch nicht mit dualen fraktionalen Schnittebenen-

gorithmen und erst recht nicht mit Branch & Cut konkurrieren.

Den Ansatz, den Padberg & Hong [34] 1980 zur Lösung des Handlungsreisendenproblems verfolgten, unterschied sich in einigen Punkten von Young's Ansatz. Sie benutzten nur ein Teilsystem von Ungleichungen und fügten nur Schnitte hinzu, die Facetten definieren. Zusätzlich erlaubten sie, dass mehrere Schnitte bei einer Iteration hinzugefügt wurden. Um diese Ungleichungen zu finden, entwickelten sie Identifizierungsalgorithmen, welche in der modernen Literatur Separierungsalgorithmen genannt werden. Diese Identifizierungsalgorithmen suchten nach Ungleichungen, die nicht nur den fraktionalem Vektor x^* abschneiden, sondern zusätzlich durch die aktuelle Lösung \bar{x} mit Gleichheit erfüllt werden. Es handelte sich also um die ersten primalen Separierungsalgorithmen. Fünf Jahre später bewiesen Padberg & Grötschel [33], dass diese zusätzliche Bedingung keine Erschwernis ist, indem sie die primale Separierung auf die Standard-Separierung reduzierten.

Der primale Ansatz von Padberg & Hong hatte bis auf sehr wenige Ausnahmen keine Auswirkung auf die Forschung in diesem Bereich. Barahona & Titan [5] entwickelten 1991 einen Algorithmus zur Lösung des Max-Cut Problems, der zu jedem Zeitpunkt eine zulässige Lösung hat und deswegen „in diesem Sinne als ein primaler Algorithmus gesehen werden kann“. Eine weitere Ausnahme ist die Veröffentlichung von De Simone & Rinaldi [9] von 1994. Hier wurde zur Lösung des Max-Cut Problems auf vollständigen Graphen die Zielfunktion direkt am Anfang bzgl. einer Startlösung umgeschaltet (Begriffe und Resultate zur Umschalt-Operation siehe Abschnitt 3.3). Anschließend leiteten sie ein neues Kriterium zur Überprüfung der Optimalität einer gefundenen Lösung her. Da sie nach dem Umschalten standardmäßig auf dem Schnitt-Konus separierten, betrieben sie indirekt primale Separierung, wenn auch nur bzgl. eines festen Vektors \bar{x} .

Erst im vorletzten Jahr begann das Interesse an der primalen Separierung wieder zu wachsen. Letchford & Lodi [26, 27] riefen dazu auf, die primalen Schnittebenenalgorithmus erneut zu untersuchen, und Eisenbrand, Rinaldi & Ventura [13] lieferten das fehlende Teilstück zum Beweis, dass die 0/1 Varianten der primalen und Standard-Separierung polynomiell äquivalent sind.

2.4 Ein primaler Schnittebenenalgorithmus

Um die Anwendung der primalen Separierung zu verdeutlichen, wird nun ein primaler Schnittebenenalgorithmus für 0/1 Probleme vorgestellt, der eine überarbeitete Version des Algorithmus von Letchford & Lodi [26] ist. Den ersten Algorithmus dieser Art stellten Padberg & Hong [34] vor.

Für 0/1 Probleme ist, wie schon in Abschnitt 2.1 gesehen, intuitiv klar, warum man bei der primalen Separierung zusätzlich fordert, dass die aktuelle Lösung auf einer gefundenen Schnittebene liegen soll. Da auch das Max-Cut Problem und gradbeschränkte minimale Spannbäume zu den 0/1 Problemen gehören, reicht die Beschränkung des Algorithmus auf diese Art von Problemen in unserem Kontext aus.

Das zu lösende 0/1 Problem muss als lineares Programm formuliert vorliegen. Im Folgenden wird davon ausgegangen, dass dabei das zugrunde liegende

Ungleichungssystem zur expliziten Speicherung zu groß ist. Dies ist in der Praxis oft der Fall, weil bei vielen Problemen eine vollständige Beschreibung des Polytop P nicht bekannt ist oder sie eine exponentielle Größe hat. In unserem Zusammenhang stellt also $Ax \leq b$ immer nur eine Relaxierung des Problems dar, die durch die Hinzunahme von Schnittebenen bei jeder Iteration verbessert wird.

Unser Algorithmus startet mit einer für das Gesamtproblem zulässigen Lösung, welche \bar{x} genannt wird. Diese sollte vorzugsweise „gut“ sein. Das Ziel ist es, die Lösung \bar{x} zu verbessern, indem sie durch eine andere Ecke des Polytops P ersetzt wird. Dies geschieht so lange, bis keine Verbesserung mehr möglich ist und die Optimalität bewiesen werden kann. Das zu \bar{x} passende Simplex-Tableau lässt sich leicht konstruieren: alle x_i , die bei der Startlösung auf 1 gesetzt sind, bilden die Basis \bar{B} . Die restlichen Variablen werden in die Nichtbasis \bar{N} eingefügt. Setze $A_{\bar{B}} = I$ und alle Einträge von b auf 1, wobei I die Einheitsmatrix ist. Deren Dimension und die Dimension von b sind gleich der Anzahl der auf 1 gesetzten Einträge von \bar{x} . Alle Einträge in $A_{\bar{N}}$ sind gleich 0. Ist zum Beispiel $\bar{x} = (0, 1, 1, 0, 0)^T$ eine Lösung, dann setzt man $\bar{B} = \{2, 3\}$, $\bar{N} = \{1, 4, 5\}$, $A = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \end{pmatrix}$ und $b = \begin{pmatrix} 1 \\ 1 \end{pmatrix}$. Anschließend wird A mit Hilfe von Schlupfvariablen $s_k \geq 0$, die auch zur Nichtbasis \bar{N} hinzugefügt werden, von

$$\begin{aligned} \max \quad & c^T x \\ \text{s.t.} \quad & Ax \leq b \\ & 0 \leq x_i \leq 1 \quad \text{für } i = 1, \dots, m \end{aligned}$$

nach

$$\begin{aligned} \max \quad & c^T x \\ \text{s.t.} \quad & Ax + Is = b \\ & 0 \leq x_i \leq 1 \quad \text{für } i = 1, \dots, m \\ & s \geq 0 \end{aligned}$$

transformiert. Die Dimension von I und damit auch die Anzahl der Einträge von s wird so gewählt, dass sie zur Anzahl der Zeilen von A passt. Das so konstruierte Tableau beinhaltet einige der Nebenbedingungen $0 \leq x_i \leq 1$, die meistens als Schranken an das jeweilige Programm übergeben werden, welches das lineare Programm löst. Im Folgenden wird immer nur von A gesprochen, womit allerdings nicht das ursprüngliche A sondern die Erweiterung mit den Spalten für die Schlupfvariablen gemeint ist.

Ist das zu \bar{x} gehörige Tableau dual zulässig, so ist \bar{x} optimal, und der Algorithmus stoppt. Falls dies nicht der Fall ist, wird ein primaler Simplex-Schritt durchgeführt, der einen neuen Lösungsvektor $A_{\bar{B}}^{-1}b$ liefert. Setze nun $x^* \in \mathbb{R}^m$ wie folgt:

$$x_j^* = \begin{cases} (A_{\bar{B}}^{-1}b)_i & \text{falls die } i\text{-te Zeile von } A \text{ zur Basisvariablen } x_j \text{ gehört} \\ 0 & \text{falls } x_j \text{ nicht in der Basis ist} \end{cases}$$

Falls x^* binär ist, so wird getestet, ob x^* für das gesamte Problem zulässig ist. Dieser Test wird von einer externen Routine gemacht. Beim Max-Cut Problem

wird z.B. getestet, ob x^* einen Schnitt darstellt. Wird dieser Test erfüllt, so wurde eine Lösung gefunden, die von nun an das neue \bar{x} darstellt. Hierbei kann es sich um die gleiche oder eine bessere Lösung handeln. Auch wenn die gleiche Lösung gefunden wurde, ist es wichtig, die neue Basis zu speichern, da sich diese im Vergleich zur alten Basis verändert hat.

Ist x^* nicht binär oder nicht für das gesamte Problem zulässig, dann wird eine primale Schnittebene generiert, die x^* abschneidet. Diese wird zum Tableau, welches zu \bar{x} gehört, hinzugefügt. Von diesem Tableau ausgehend wird wieder ein primaler Simplex-Schritt durchgeführt, der wiederum ein anderes x^* liefert usw. Der Algorithmus terminiert, wenn bewiesen werden konnte, dass das zu \bar{x} gehörige Tableau dual zulässig ist.

Im Gegensatz zur Standard-Separierung kann der Fall auftreten, dass das gegenwärtig gefundene binäre x^* nicht für das Gesamtproblem zulässig ist, aber keine primale Schnittebene gefunden werden kann, die es abschneidet. In diesem Fall muss einmal die Standard-Separierung aufgerufen werden, um dieses x^* mittels einer Standard-Schnittebene abzuschneiden. Wenn für einen unzulässigen fraktionalen Punkt x^* keine primale Schnittebene gefunden werden kann, dann sind die Voraussetzungen für einen gemischtganzzahligen Gomory-Schnitt gegeben. Dieser fungiert in diesem Zusammenhang immer als primale Schnittebene, was wir gegen Ende dieses Abschnittes beweisen werden.

Die eben beschriebenen Fälle können anhand eines Beispiels verdeutlicht werden. Gegeben sei dazu das folgende lineare 0/1 Programm:

$$\begin{aligned} \max \quad & x_1 + x_2 + x_3 \\ \text{s.t.} \quad & x_1 + x_2 \leq 1 \\ & x_1 + x_3 \leq 1 \\ & x_2 + x_3 \leq 1 \\ & x_i \in \{0, 1\} \quad \text{für } i = 1, 2, 3 \end{aligned}$$

Angenommen, nur die ersten beiden Ungleichungen sind in der momentanen Relaxierung aufgenommen, und die gegenwärtig beste gefundene Lösung sei $\bar{x} = (1, 0, 0)^T$. Ein zugehöriges Simplex-Tableau ist

$$\begin{array}{c|ccc} & x_3 & s_1 & s_2 \\ \hline & 1 & -1 & 0 \\ x_1 & 1 & 0 & 1 \\ x_2 & 0 & -1 & -1 \end{array}$$

x_1 verlässt nun die Basis, und x_3 wird stattdessen in die Basis aufgenommen.

$$\begin{array}{c|ccc} & x_1 & s_1 & s_2 \\ \hline & 2 & -1 & -1 \\ x_3 & 1 & 0 & 1 \\ x_2 & 1 & 1 & 0 \end{array}$$

Der neu gefundene Vektor ist demnach $x^* = (0, 1, 1)^T$. Er ist nicht für das Gesamtproblem zulässig, denn er verletzt die dritte Ungleichung. Ein primaler Separierungsalgorithmus findet diese Ungleichung jedoch nicht, da \bar{x} sie nicht mit Gleichheit erfüllt. Also wird die Standard-Separierung aufgerufen, die die Ungleichung $x_2 + x_3 \leq 1$ findet. Die Schlupfvariable s_3 wird dieser Ungleichung

hinzugefügt und in die Basis aufgenommen. Durch die Nichtbasisvariablen lässt sich die Schnittebene dann darstellen als $s_3 = -1 + 2x_1 + s_1 + s_2$, so dass das neue Tableau wie folgt aussieht:

$$\begin{array}{c|ccc} & x_1 & s_1 & s_2 \\ \hline & 2 & -1 & -1 & -1 \\ x_3 & 1 & 1 & 0 & 1 \\ x_2 & 1 & 1 & 1 & 0 \\ s_3 & -1 & -2 & -1 & -1 \end{array}$$

Da dieses Tableau nicht mehr primal zulässig ist, wird eine neue zulässige Basis mit Basislösung \bar{x} berechnet.

$$\begin{array}{c|ccc} & x_3 & s_1 & s_2 \\ \hline & 1 & 1 & -1 & 0 \\ x_1 & 1 & 1 & 0 & 1 \\ x_2 & 0 & -1 & 1 & -1 \\ s_3 & 1 & 2 & -1 & 1 \end{array}$$

Nach einem anschließenden primalen Simplex-Schritt sieht das Tableau wie folgt aus:

$$\begin{array}{c|ccc} & s_3 & s_1 & s_2 \\ \hline & \frac{3}{2} & -\frac{1}{2} & -\frac{1}{2} & -\frac{1}{2} \\ x_1 & \frac{1}{2} & -\frac{1}{2} & \frac{1}{2} & \frac{1}{2} \\ x_2 & \frac{1}{2} & \frac{1}{2} & \frac{1}{2} & -\frac{1}{2} \\ x_3 & \frac{1}{2} & \frac{1}{2} & -\frac{1}{2} & \frac{1}{2} \end{array}$$

Daraus resultiert der fraktionale Vektor $x^* = (\frac{1}{2}, \frac{1}{2}, \frac{1}{2})^T$. Da die primale Separierung keine primale Schnittebene finden kann, wird nun ein gemischtganzzahliger Gomory-Schnitt generiert. Unabhängig davon, aus welcher Zeile des Tableaus dieser gewonnen wird, lautet er $-\frac{1}{2}s_1 - \frac{1}{2}s_2 - \frac{1}{2}s_3 \leq -\frac{1}{2}$. Drückt man ihn mit Hilfe der x -Variablen aus, so erhält man $x_1 + x_2 + x_3 \leq 1$. Diese Schnittebene wird von $\bar{x} = (1, 0, 0)^T$ mit Gleichheit erfüllt und schneidet den fraktionalen Vektor $x^* = (\frac{1}{2}, \frac{1}{2}, \frac{1}{2})^T$ ab. Somit löst sie das primale Separierungsproblem.

Man geht nun wieder von dem zu \bar{x} gehörenden Tableau aus. Nachdem die Schlupfvariable s_4 zum gemischtganzzahligen Gomory-Schnitt hinzugefügt und in die Basis aufgenommen wurde, kann man sie durch die Nichtbasisvariablen ausdrücken als $s_4 = -x_3 + s_1$. Das Tableau ist damit

$$\begin{array}{c|ccc} & x_3 & s_1 & s_2 \\ \hline & 1 & 1 & -1 & 0 \\ x_1 & 1 & 1 & 0 & 1 \\ x_2 & 0 & -1 & 1 & -1 \\ s_3 & 1 & 2 & -1 & 1 \\ s_4 & 0 & 1 & -1 & 0 \end{array}$$

Nach einem primalen Simplex-Schritt, der s_4 aus der Basis nimmt und x_3 hinzufügt, gelangt man zu

		s_4	s_1	s_2
	1	-1	0	0
x_1	1	-1	1	1
x_2	0	1	0	-1
s_3	1	-2	1	1
x_3	0	1	-1	0

Das Tableau ist primal und dual zulässig, womit die Optimalität des für das Gesamtproblem zulässigen Vektors $(1, 0, 0)^T$ bewiesen ist.

Die zu dem Vektor \bar{x} gehörige Basis und Nichtbasis nennen wir im Folgenden \bar{B} bzw. \bar{N} . Mit B und N werden die Basis bzw. Nichtbasis des Tableaus bezeichnet, auf dem die Simplex-Schritte durchgeführt werden. Der durch das Tableau gegebene Lösungsvektor sei x^* . Unser primaler Schnittebenenalgorithmus lässt sich nun wie folgt skizzieren:

- Schritt 1: Finde einen guten Startvektor \bar{x} und konstruiere dazu das entsprechende Tableau mit Matrix A , Vektor b , Basis \bar{B} und Nichtbasis \bar{N} . Setze $x^* = \bar{x}$, $B = \bar{B}$ und $N = \bar{N}$.
- Schritt 2: Falls das zu x^* gehörige Tableau dual zulässig ist, stoppe mit „Optimale Lösung gefunden“.
- Schritt 3: Führe einen primalen Simplex-Schritt durch. Sei x^* der neu erhaltene Vektor.
- Schritt 4: Falls x^* binär und für das Gesamtproblem zulässig ist, speichere $\bar{x} = x^*$, $\bar{B} = B$ und $\bar{N} = N$ und gehe zu Schritt 2.
- Schritt 5: Rufe die primale Separierung für bekannte Ungleichungsklassen auf. Falls verletzte Ungleichungen gefunden werden, setze $B = \bar{B}$ und $N = \bar{N}$, füge eine oder mehrere der gefundenen Schnittebenen dem Tableau hinzu und gehe zu Schritt 3.
- Schritt 6: Falls x^* binär aber nicht für das Gesamtproblem zulässig ist, finde eine Ungleichung, die durch x^* verletzt wird, aber nicht von \bar{x} mit Gleichheit erfüllt wird, und füge diese dem Tableau hinzu. Berechne eine zulässige Basis \bar{B} und \bar{N} mit Basislösung \bar{x} , setze $B = \bar{B}$ und $N = \bar{N}$ und gehe zu Schritt 3.
- Schritt 7: Generiere anhand des gegenwärtigen Tableaus einen oder mehrere gemischtganzzahlige Gomory-Schnitte. Setze $B = \bar{B}$ und $N = \bar{N}$, füge eine oder mehrere der gefundenen Schnittebenen dem Tableau hinzu und gehe zu Schritt 3.

Das in Schritt 1 konstruierte Tableau, welches zu dem Startvektor \bar{x} gehört, ist primal zulässig. Dies ist notwendig, um einen primalen Simplex-Schritt durchführen zu können. Dass diese Voraussetzung auch noch nach der Hinzunahme von weiteren Ungleichungen in Schritt 5 und 7 gilt, sichert unsere

Beobachtung 2.4.1: Sei $Ax = b$ mit der Basis B so, dass das zugehörige Simplex-Tableau primal zulässig ist. Sei \bar{x} der Lösungsvektor. Durch Hinzunahme von Ungleichungen $Fx \leq f$ mit $F \in \mathbb{R}^{k \times m}$ und $f \in \mathbb{R}^k$, die von \bar{x} mit Gleichheit erfüllt werden, bleibt die primale Zulässigkeit erhalten.

Beweis: Die hinzuzufügenden Ungleichungen werden mit den Schlupfvariablen $s = (s_1, \dots, s_k)^T$ mit $s_i \geq 0$ für $i = 1, \dots, k$ erweitert. Anschließend fügt man die Schlupfvariablen s in die Basis ein. Somit kann man sie ausdrücken als $s = f - Fx$. Damit die primale Zulässigkeit erhalten bleibt, muss $f - Fx \geq 0$ für den aktuellen Lösungsvektor gelten. Dieser erfüllt sogar $F\bar{x} = f$. Also gilt $s = 0$ und das Tableau ist weiterhin primal zulässig. \square

In Schritt 6 verliert das Tableau durch die Hinzunahme einer Ungleichung, die von \bar{x} nicht mit Gleichheit erfüllt wird, die primale Zulässigkeit. Da man im anschließenden Schritt 3 wieder vom Punkt \bar{x} aus einen primalen Simplex-Schritt durchführt, muss man eine zulässige Basis berechnen, deren Basislösung \bar{x} lautet. Dies erreichen wir folgendermaßen:

Nachdem dem Tableau die Ungleichung hinzugefügt wurde, ändert man die Zielfunktionskoeffizienten. Für ein Minimierungsproblem setzt man

$$c_i = \begin{cases} 1 & \text{falls } \bar{x}_i = 0 \\ -1 & \text{falls } \bar{x}_i = 1 \end{cases} \quad \forall i \in \{1, \dots, m\}$$

und für ein Maximierungsproblem analog

$$c_i = \begin{cases} -1 & \text{falls } \bar{x}_i = 0 \\ 1 & \text{falls } \bar{x}_i = 1 \end{cases} \quad \forall i \in \{1, \dots, m\}$$

Mittels des primalen Simplex-Algorithmus berechnet man nun das Optimum, welches im Punkt \bar{x} angenommen wird. Somit erhält man eine zulässige Basis mit Basislösung \bar{x} . Anschließend werden die Zielfunktionskoeffizienten wieder auf ihre ursprünglichen Werte zurückgesetzt. Die primale Zulässigkeit des Tableaus bleibt dabei erhalten.

Um in Schritt 7 einen gemischtganzzahligen Gomory-Schnitt erzeugen zu können, muss der zu diesem Zeitpunkt aktuelle Lösungsvektor fraktional sein. Dies ist nach Schritt 6 der Fall, denn wenn x^* vor Schritt 6 schon fraktional war, dann bleibt x^* der aktuelle Lösungsvektor. War x^* allerdings binär, so wird Schritt 7 nicht erreicht. Dass der gemischtganzzahlige Gomory-Schnitt in dem vorgestellten Algorithmus eine primale Schnittebene liefert, besagt unser

Satz 2.4.2: Unter den Voraussetzungen, die zu Beginn des Schrittes 7 gelten, erzeugt der gemischtganzzahlige Gomory-Schnitt eine Schnittebene, die von \bar{x} mit Gleichheit erfüllt wird.

Beweis: Zu Beginn des Schrittes 7 ist x^* eine fraktionale Ecke des Polytops, das die momentane Relaxierung darstellt. Man betrachte das unbeschränkte, m -dimensionale Polyeder, das durch die Ungleichungen gegeben ist, die x^* mit Gleichheit erfüllt. x^* ist die einzige Ecke dieses Polyeders. Von x^* ausgehend gibt es genau m Strahlen, die die Kanten des Polyeders sind. Man wählt nun eine Variable von x^* , die fraktional ist, hier x_k genannt. Von x^* aus bewegt man sich so lange entlang einer der Kanten des Polyeders, bis x_k entweder 0 oder 1 wird. Dies ergibt einen Punkt. Entsprechend verfährt man mit den restlichen Kanten und erhält m Punkte, die eine Hyperebene definieren. Man

bezeichnet diese als einfache disjunktive Schnittebene (im Englischen „simple disjunctive cut“).

Sei das zu x^* gehörende Tableau gebildet aus A , b , der Basis B und der Nichtbasis N . Setze:

$$\begin{aligned}\bar{a}_{ij} &:= (A_B^{-1}A_N)_{ij} \\ f_{ij} &:= \bar{a}_{ij} - \lfloor \bar{a}_{ij} \rfloor \\ \bar{b}_i &:= A_B^{-1}b \\ f_{i0} &:= \bar{b}_i - \lfloor \bar{b}_i \rfloor\end{aligned}$$

Sei die i -te Zeile des Tableaus diejenige Zeile, die zu x_k gehört. Sie liefert die Darstellung von x_k mit Hilfe der Nichtbasisvariablen als

$$x_k = \bar{b}_i - \sum_{j \in N} \bar{a}_{ij} y_j$$

wobei y_j jeweils eine Variable von x oder s ist. Dann lautet die aus der i -ten Zeile gewonnene einfache disjunktive Schnittebene

$$- \sum_{\substack{j \in N \\ \bar{a}_{ij} \geq 0}} \bar{a}_{ij} y_j + \frac{f_{i0}}{1 - f_{i0}} \sum_{\substack{j \in N \\ \bar{a}_{ij} < 0}} \bar{a}_{ij} y_j \leq -f_{i0} \quad (2.1)$$

Aufgrund der Arbeitsweise des Algorithmus weiß man, dass x^* durch einen primalen Simplex-Schritt aus \bar{x} entstanden ist. Somit verbindet einer der m Strahlen \bar{x} und x^* . Also ist \bar{x} einer der m Punkte, die die Hyperebene definieren, womit \bar{x} die Ungleichung der einfachen disjunktiven Schnittebene mit Gleichheit erfüllt.

Weiterhin ist bekannt, dass die einfache disjunktive Schnittebene verschärft werden kann. Sei dazu N_Z die Indexmenge der Nichtbasisvariablen, die ganzzahlig sein müssen, also die Variablen von x . Analog sei $N_Q := N \setminus N_Z$, was somit die Indexmenge der Schlupfvariablen darstellt, die momentan nicht in der Basis sind. Die Verschärfung der einfachen disjunktiven Schnittebene lautet dann

$$- \sum_{\substack{j \in N_Z \\ f_{ij} \leq f_{i0}}} f_{ij} y_j - \frac{f_{i0}}{1 - f_{i0}} \sum_{\substack{j \in N_Z \\ f_{ij} > f_{i0}}} (1 - f_{ij}) y_j - \sum_{\substack{j \in N_Q \\ \bar{a}_{ij} \geq 0}} \bar{a}_{ij} y_j + \frac{f_{i0}}{1 - f_{i0}} \sum_{\substack{j \in N_Q \\ \bar{a}_{ij} < 0}} \bar{a}_{ij} y_j \leq -f_{i0} \quad (2.2)$$

Dies ist der gemischtganzzahlige Gomory-Schnitt. Es bleibt zu zeigen, dass dieser auch von \bar{x} mit Gleichheit erfüllt wird. Wir beweisen dies durch folgende Überlegungen:

Man kann die einfache disjunktive Schnittebene (2.1) darstellen als

$$- \sum_{\substack{j \in N_Z \\ \bar{a}_{ij} \geq 0}} \bar{a}_{ij} y_j + \frac{f_{i0}}{1 - f_{i0}} \sum_{\substack{j \in N_Z \\ \bar{a}_{ij} < 0}} \bar{a}_{ij} y_j - \sum_{\substack{j \in N_Q \\ \bar{a}_{ij} \geq 0}} \bar{a}_{ij} y_j + \frac{f_{i0}}{1 - f_{i0}} \sum_{\substack{j \in N_Q \\ \bar{a}_{ij} < 0}} \bar{a}_{ij} y_j \leq -f_{i0} \quad (2.3)$$

Die Schlupfvariablen s_i lassen sich durch die Variablen von x ausdrücken als

$$s_i = b_i - \sum_{j=1}^m a_{ij} x_j$$

Setzt man diese in (2.3) und (2.2) ein, so ergeben sich Darstellungen der beiden Schnittebenen, die nur auf den Variablen von x basieren. Ein Vergleich dieser beiden Darstellungen zeigt, dass sich nur die Koeffizienten für diejenigen x_i unterscheiden, die in der Nichtbasis N_Z sind.

Da x^* aus \bar{x} durch eine Simplex-Iteration entstanden ist, weiß man, dass zwischen der zu \bar{x} gehörigen Basis \bar{B} sowie Nichtbasis \bar{N} und der zu x^* gehörigen Basis B sowie Nichtbasis N ein Basiswechsel stattgefunden hat. Wurde dabei keine Variable $x_i \in \bar{B}$ zu N_Z hinzugenommen, so sind alle Variablen aus N_Z auch in \bar{N}_Z enthalten. Da $\bar{x}_i = 0$ für alle $i \in \bar{N}_Z$ ist und (2.3) von \bar{x} mit Gleichheit erfüllt wird, trifft dies damit auch für den gemischtganzzahligen Gomory-Schnitt (2.2) zu. Wenn ein $x_i \in \bar{B}$ mit $\bar{x}_i = 0$ in die Nichtbasis N_Z gewechselt ist, wird (2.2) ebenfalls von \bar{x} mit Gleichheit erfüllt. Den übrig gebliebenen Fall, dass ein $x_i \in \bar{B}$ mit $\bar{x}_i = 1$ in die Nichtbasis N_Z aufgenommen wurde, untersuchen wir im Folgenden genauer.

Das zu \bar{x} gehörende Tableau sehe wie folgt aus:

$$\begin{array}{c|ccc}
 \dots & \dots & \dots & \dots \\
 \hline
 0 & \dots & \tilde{a}_{1j} \leq 0 & \dots \\
 1 & \dots & \tilde{a}_{2j} > 0 & \dots \\
 x_i & 1 & \dots & \tilde{a}_{3j} > 0 \dots \\
 & 1 & \dots & \tilde{a}_{4j} \leq 0 \dots
 \end{array}$$

In diesem Tableau sind alle möglichen Fälle enthalten. Es kann keine Variable x_l mit $\bar{x}_l = 0$ geben, für die das Element in der j -ten Spalte $\tilde{a}_{.j} > 0$ ist, da ansonsten x_l und nicht x_i die Basis verlassen hätte. Außerdem gilt

$$\begin{aligned}
 \forall i \neq 3 \text{ mit } \tilde{a}_{ij} > 0 : \frac{1}{\tilde{a}_{3j}} &\leq \frac{1}{\tilde{a}_{ij}} \\
 \Rightarrow \forall i \neq 3 : \tilde{a}_{3j} &\geq \tilde{a}_{ij}
 \end{aligned}$$

Nach einem primalen Simplex-Schritt erhält man das zu dem fraktionalen Vektor x^* gehörige Tableau

$$\begin{array}{c|ccc}
 & \dots & \dots & x_1 \\
 \hline
 (1) & -\frac{\tilde{a}_{1j}}{\tilde{a}_{3j}} & \dots & -\frac{\tilde{a}_{1j}}{\tilde{a}_{3j}} \dots \\
 (2) & 1 - \frac{\tilde{a}_{2j}}{\tilde{a}_{3j}} & \dots & -\frac{\tilde{a}_{2j}}{\tilde{a}_{3j}} \dots \\
 (3) & \frac{1}{\tilde{a}_{3j}} & \dots & \frac{1}{\tilde{a}_{3j}} \dots \\
 (4) & 1 - \frac{\tilde{a}_{4j}}{\tilde{a}_{3j}} & \dots & -\frac{\tilde{a}_{4j}}{\tilde{a}_{3j}} \dots
 \end{array}$$

Soll nun aus Zeile (1) ein gemischtganzzahliger Gomory-Schnitt abgeleitet werden, so muss diese Zeile zu einer fraktionalen Variable x_k gehören. Damit folgt $0 < -\frac{\tilde{a}_{1j}}{\tilde{a}_{3j}} < 1$. Der Koeffizient von x_i in der einfachen disjunktiven Schnittebene (2.3) ist demnach $-\bar{a}_{1j} = \frac{\tilde{a}_{1j}}{\tilde{a}_{3j}}$. Da $f_{1j} = f_{10}$ gilt, lautet der Koeffizient von x_i im gemischtganzzahligen Gomory-Schnitt (2.2) ebenfalls $-f_{1j} = \frac{\tilde{a}_{1j}}{\tilde{a}_{3j}}$.

Wird aus Zeile (2) ein gemischtganzzahliger Gomory-Schnitt abgeleitet, dann muss $0 < 1 - \frac{\tilde{a}_{2j}}{\tilde{a}_{3j}} < 1$ gelten. Dies ist äquivalent zu $-1 < -\frac{\tilde{a}_{2j}}{\tilde{a}_{3j}} < 0$. Es ist

$f_{20} = 1 - \frac{\tilde{a}_{2j}}{\tilde{a}_{3j}}$. Somit lautet der Koeffizient von x_i in (2.3) $\frac{f_{20}}{1-f_{20}}\bar{a}_{2j} = \frac{\tilde{a}_{2j}}{\tilde{a}_{3j}} - 1$. Weiter gilt $-1 < -\frac{\tilde{a}_{2j}}{\tilde{a}_{3j}} < 0 \Rightarrow \lfloor -\frac{\tilde{a}_{2j}}{\tilde{a}_{3j}} \rfloor = -1$. Also ist $f_{2j} = -\frac{\tilde{a}_{2j}}{\tilde{a}_{3j}} + 1 = f_{20}$ und somit hat x_i in (2.2) den gleichen Koeffizienten $-f_{2j} = \frac{\tilde{a}_{2j}}{\tilde{a}_{3j}} - 1$.

Die Ableitung des gemischtganzzahligen Gomory-Schnittes aus Zeile (3) setzt $0 < \frac{1}{\tilde{a}_{3j}} < 1$ voraus. Der Koeffizient von x_i in der einfachen disjunktiven Schnittebene ist $-\bar{a}_{3j} = -\frac{1}{\tilde{a}_{3j}}$. Wegen $f_{3j} = f_{30} = \frac{1}{\tilde{a}_{3j}}$ hat x_i in (2.2) auch den Koeffizienten $-f_{3j} = -\frac{1}{\tilde{a}_{3j}}$.

In allen drei Fällen hat also x_i in der einfachen disjunktiven Schnittebene (2.3) den gleichen Koeffizienten wie im gemischtganzzahligen Gomory-Schnitt (2.2), weswegen dieser ebenfalls von \bar{x} mit Gleichheit erfüllt wird.

Aus Zeile (4) kann kein gemischtganzzahliger Gomory-Schnitt abgeleitet werden, denn es gilt $\tilde{a}_{3j} > 0$ und $\tilde{a}_{4j} \leq 0 \Rightarrow \frac{\tilde{a}_{4j}}{\tilde{a}_{3j}} \leq 0 \Rightarrow 1 - \frac{\tilde{a}_{4j}}{\tilde{a}_{3j}} \geq 1$.

□

Abschließend wollen wir noch eine weitere Möglichkeit des Einsatzes eines primalen Schnittebenenverfahrens erwähnen. Wie bereits gesehen, hat man selbst bei Abbruch des Algorithmus eine für das gesamte Problem zulässige Lösung. Dies kann man ausnutzen, um die von einer Heuristik gelieferte Lösung nachträglich zu verbessern. Als die von dem primalen Schnittebenenalgorithmus benötigte Startlösung setzt man die Lösung, die die Heuristik gefunden hat. Wir begrenzen nun zeitlich die Anwendung des primalen Schnittebenenalgorithmus. Nach Abbruch haben wir wieder eine für das gesamte Problem zulässige Lösung, welche zumindest nicht schlechter als die heuristische Lösung ist. Eher ist anzunehmen, dass sie besser wenn nicht sogar optimal ist.

2.5 Resultate zur primalen Separierung

Bevor mit der Untersuchung des Max-Cut Problems bzw. der gradbeschränkten minimalen Spannbäume begonnen wird, soll ein Ausblick auf andere bekannte Probleme gegeben werden, für die die primale Separierung bereits untersucht wurde. Die Resultate, die Letchford & Lodi [27] bzgl. des Stabile Menge Problems, des Rucksack Problems und des Handlungsreisendenproblems erzielt haben, werden zusammengefasst. Von allen drei Problemen ist bekannt, dass sie \mathcal{NP} -schwierige 0/1 Probleme sind. Aufgrund der Erkenntnisse des Abschnittes 2.2 wird deswegen nur die Separierung einiger bekannter Ungleichungsklassen betrachtet.

2.5.1 Stabile Menge Problem

Das gewichtete Stabile Menge Problem lautet wie folgt: Gegeben sei ein gewichteter ungerichteter Graph (G, w) mit Knotenmenge V und Kantenmenge E . Finde eine Knotenteilmenge $S \subseteq V$, in der keine zwei Knoten adjazent sind und deren Gewicht so groß wie möglich ist. Dieses Problem kann folgendermaßen als ganzzahliges lineares Programm geschrieben werden:

$$\begin{aligned} \max \quad & w^T x \\ \text{s.t.} \quad & x_i + x_j \leq 1 \quad \forall ij \in E \\ & x_i \in \{0, 1\} \quad \forall i \in V \end{aligned}$$

Eine wichtige Klasse von facettendefinierenden Ungleichungen sind die *Clique-Ungleichungen*:

$$\sum_{i \in C} x_i \leq 1 \quad \text{mit } C \subseteq V \text{ ist eine Clique von } G$$

Das Standard-Separierungsproblem lautet demnach, eine Clique von G zu finden, deren x^* -Gewicht höher als 1 ist, oder zu beweisen, dass es solch eine nicht gibt. Diese Fragestellung beinhaltet das bekannte Maximum-Clique Problem, dessen Optimierungsvariante \mathcal{NP} -schwierig ist. Also ist die Standard-Separierung von Clique-Ungleichungen \mathcal{NP} -schwierig. Beim primalen Separierungsproblem muss die zu findende Clique zusätzlich zu der Forderung, dass ihr x^* -Gewicht höher als 1 ist, genau einen Knoten i enthalten, für den $\bar{x}_i = 1$ gilt. Dies reicht nicht aus, um die Komplexität des Problems zu verringern, denn das Standard-Separierungsproblem kann wie folgt auf das primale Separierungsproblem reduziert werden:

Der Graph G wird um einen künstlichen Knoten u erweitert, und die Ungleichungen $x_i + x_u \leq 1 \quad \forall i \in V$ werden dem Ungleichungssystem hinzugefügt. Definiert man $x_i = 0 \quad \forall i \in V$ und $x_u = 1$, so erhält man einen Eckpunkt des neuen Polytops, der *Anker* genannt wird. Eine Ungleichung $\alpha x \leq \gamma$ ist genau dann für das Stabile Menge Polytop gültig und wird von x^* verletzt, wenn die Ungleichung $\sum_{i \in V} \alpha_i x_i + \gamma x_u \leq \gamma$ für das neue Polytop gültig ist, von $(x^*, 0)$ verletzt und von dem Anker mit Gleichheit erfüllt wird. Da jede Clique von G durch Hinzunahme des künstlichen Knotens u zu einer Clique im erweiterten Graphen gemacht werden kann, lässt sich dies auf die Clique-Ungleichungen übertragen. Somit ist die primale Separierung der Clique-Ungleichungen ebenfalls \mathcal{NP} -schwierig.

Eine weitere bekannte Ungleichungsklasse für das Stabile Menge Polytop sind die *ungeraden Kreisungleichungen*:

$$\sum_{i \in C} x_i \leq \frac{|C| - 1}{2} \quad \text{mit } C \subseteq V \text{ induziert einen ungeraden Kreis von } G$$

Das Standard-Separierungsproblem für ungerade Kreisungleichungen kann in polynomieller Zeit gelöst werden, indem man $|V|$ Kürzeste Wege Probleme in einem entsprechend definierten Hilfsgraphen löst. Damit eine ungerade Kreisungleichung von \bar{x} mit Gleichheit erfüllt wird, müssen genau $\frac{|C|-1}{2}$ Knoten des Kreises in \bar{x} den Wert 1 annehmen. Da aber ein Kreis aus mindestens 3 Knoten besteht, gilt $\frac{|C|-1}{2} \geq 1$. Somit muss eine ungerade Kreisungleichung, die die zusätzliche Bedingung der primalen Separierung erfüllt, mindestens einen der Knoten enthalten, die in \bar{x} den Wert 1 haben. Aufgrund dieser Tatsache kann die Anzahl der zu berechnenden Kürzeste Wege Probleme von $|V|$ auf $\sum_{i \in V} \bar{x}_i$ reduziert werden. Vom komplexitätstheoretischen Standpunkt aus betrachtet ist dadurch das primale Separierungsproblem für ungerade Kreisungleichungen nicht leichter als die Standard-Variante. Wenn der Graph aber z.B. dicht ist, und die momentan beste Lösung \bar{x} eine kleine stabile Menge angibt, hat man praktisch gesehen eine Verbesserung.

2.5.2 Rucksack Problem

Beim Rucksack Problem ist eine endliche Menge $\{x_1, \dots, x_n\}$ von Gegenständen gegeben. Diesen Gegenständen sind Werte $p_i > 0$ und Größen $a_i > 0$ für $i = 1, \dots, n$ zugeordnet. Der Rucksack kann maximal die Größe $a_0 > 0$ fassen. Es gilt nun, eine Auswahl von Gegenständen zu treffen, die in den Rucksack passen und gleichzeitig den größtmöglichen Wert haben. Als lineares Programm formuliert lautet dies

$$\begin{aligned} \max \quad & p^T x \\ \text{s.t.} \quad & a^T x \leq a_0 \\ & x_i \in \{0, 1\} \quad \text{für } i = 1, \dots, n \end{aligned}$$

Eine Menge $C \subseteq \{1, \dots, n\}$, für die $\sum_{i \in C} a_i > a_0$ gilt, nennt man *Überdeckung*. Damit lassen sich die bekannten *Überdeckungsungleichungen* beschreiben

$$\sum_{i \in C} x_i \leq |C| - 1 \quad \text{mit } C \text{ ist eine Überdeckung}$$

Bekanntermaßen ist das Standard-Separierungsproblem für Überdeckungsungleichungen \mathcal{NP} -schwierig. Dies trifft auch für die primale Separierung zu, denn die Standard-Separierung kann mit dem gleichen Verfahren wie in 2.5.1 auf die primale Separierung reduziert werden.

Man nimmt einen künstlichen Gegenstand x_{n+1} mit Größe $p_{n+1} = M$ zur Menge der Gegenstände hinzu, wobei $M = \lceil \sum_{i=1}^n a_i \rceil$. Das Fassungsvermögen des Rucksackes wird ebenfalls um M erweitert, so dass die neue Ungleichung für das erweiterte Rucksack Problem $a^T x + Mx_{n+1} \leq a_0 + M$ lautet. Der Anker, der diesmal durch $x_i = 1$ für $i = 1, \dots, n$ und $x_{n+1} = 0$ definiert wird, ist wieder ein Eckpunkt des neuen zugehörigen Polytops. $C \subseteq \{1, \dots, n\}$ ist nun genau dann eine Überdeckung des Rucksack Problems, wenn $C \cup \{n+1\}$ eine Überdeckung des erweiterten Rucksack Problems ist. Weiterhin verletzt x^* genau dann eine Überdeckungsungleichung des Rucksack Problems, wenn der Punkt $(x^*, 1)$ eine Überdeckungsungleichung des erweiterten Rucksack Problems verletzt, die durch den Anker mit Gleichheit erfüllt wird.

Die Überdeckungsungleichungen können auf mehrere Arten *angehoben* (im Englischen „lifted“) werden, um Facetten zu definieren. Anheben bedeutet hier, geeignete Koeffizienten $b_j \geq 0$ für die Variablen auf der linken Seite der Ungleichung zu berechnen, die nicht in der Überdeckung enthalten sind. Die angehobene Überdeckungsungleichung zu einer Überdeckung C lautet demnach

$$\sum_{i \in C} x_i + \sum_{j \notin C} b_j x_j \leq |C| - 1$$

Die heuristischen Verfahren, die zur Standard-Separierung der angehobenen Überdeckungsungleichungen bekannt sind, können auf die primale Separierung solcher Ungleichungen übertragen werden.

2.5.3 Handlungsreisendenproblem

Zu den bekanntesten Problemen der kombinatorischen Optimierung zählt wohl das Handlungsreisendenproblem. Gegeben sind n Orte, die durch m mit Kosten belegten Strecken verbunden sind. Es soll eine Tour gefunden werden, die

jeden Ort genau einmal besucht, bei dem Startort endet und deren Kosten so gering wie möglich sind. Je nachdem, ob die Strecken in beide Richtungen genutzt werden können oder nur auf eine Richtung beschränkt sind, beschreibt man das Problem mit einem ungerichteten bzw. gerichteten Graphen. Man spricht dann von dem symmetrischen resp. asymmetrischen Handlungsreisendenproblem.

Symmetrisches Handlungsreisendenproblem

Zu einem gegebenen ungerichteten gewichteten Graphen (G, c) lautet die Formulierung des Problems als lineares Programm

$$\begin{aligned} \min \quad & c^T x \\ \text{s.t.} \quad & (\chi^{\delta(i)})^T x = 2 \quad \forall i \in V \end{aligned} \quad (2.4)$$

$$(\chi^{\delta(S)})^T x \geq 2 \quad \forall S \subset V: 2 \leq |S| < |V| \quad (2.5)$$

$$x_e \in \{0, 1\} \quad \forall e \in E$$

Die Gleichungen (2.4), die man unter der Bezeichnung *Gradgleichungen* kennt, beschreiben die affine Hülle des zugehörigen Polytops. Die Ungleichungen (2.5) sind die *facettendefinierenden Kurzzyklusbedingungen*. Für diese ist bekannt, dass das zugehörige Standard-Separierungsproblem wie folgt gelöst werden kann:

Anhand des abzuschneidenden x^* definiert man den Hilfsgraphen $G^* = (V, E^*)$, wobei $E^* := \{e \in E \mid x_e^* > 0\}$. Es gilt nun, eine Knotenteilmenge $S \subset V$ zu finden, die $(\chi^{\delta(S)})^T x^*$ minimiert, was auf die Berechnung eines minimalen gewichteten Schnittes hinausläuft. Der momentan schnellste Minimum-Schnitt-Algorithmus ist der von Nagamochi, Ono & Ibaraki, welcher in $O(|V||E^*| + |V|^2 \log |V|)$ läuft.

Sei nun \bar{x} der Inzidenzvektor einer Tour in G . Das primale Separierungsproblem für die Kurzzyklusbedingungen sucht nach einer Knotenteilmenge $S \subset V$, die $(\chi^{\delta(S)})^T \bar{x}$ minimiert und $(\chi^{\delta(S)})^T \bar{x} = 2$ erfüllt. Obwohl es exponentiell viele Kurzzyklusbedingungen gibt, erfüllen nur polynomiell viele, nämlich $|V| \frac{|V|-3}{2}$ die zusätzliche Bedingung der Gleichheit für \bar{x} . Dies lässt vermuten, dass das primale Separierungsproblem für diese Ungleichungsklasse leichter zu lösen ist, was auch tatsächlich zutrifft. Es gibt einen $O(|V||E^*|)$ Algorithmus zur Lösung der primalen Separierung für Kurzzyklusbedingungen, der durch einen „Divide et Impera“ Ansatz sogar auf $O(|E^*| \log |V|)$ gebracht werden kann.

Eine weitere bekannte Klasse von facettendefinierenden Ungleichungen sind die „*Kamm-Ungleichungen*“, die folgende Form haben:

$$(\chi^{E(H)})^T x + \sum_{i=1}^k (\chi^{E(T_i)})^T x \leq |H| + \sum_{i=1}^k (|T_i| - 1) - \frac{k+1}{2}$$

Dabei nennt man H den *Griff* und die T_i für $i = 1, \dots, k$ die *Zinken* des Kammes. Für sie muss gelten:

- 1.) $\emptyset \neq H \subset V$
- 2.) $T_i \subset V$ für $i = 1, \dots, k$
- 3.) $|T_i \cap H| \geq 1$ für $i = 1, \dots, k$
- 4.) $|T_i \setminus H| \geq 1$ für $i = 1, \dots, k$
- 5.) $T_i \cap T_j = \emptyset$ für $1 \leq i < j \leq k$
- 6.) $k \geq 3$ und k ungerade

Verlangt man zusätzlich, dass bei Bedingung 3.) und 4.) Gleichheit herrschen muss, so erhält man die *2-Matching Ungleichungen*, die mit Hilfe der Gradgleichungen (2.4) geschrieben werden können als

$$(\chi^{\delta(H) \setminus F})^T x + \sum_{e \in F} (1 - x_e) \geq 1,$$

wobei $\emptyset \neq H \subset V$ und F eine Menge von Kanten ist, für die $|F|$ ungerade und $F \subset \delta(H)$ gilt. Die Standard-Separierung dieser 2-Matching Ungleichungen kann auf einem entsprechend definierten Hilfsgraphen durch das Berechnen von $O(|E^*|)$ maximalen Flüssen in polynomieller Zeit gelöst werden.

Damit eine Tour \bar{x} , die bei der primalen Separierung gegeben ist, die 2-Matching Ungleichungen mit Gleichheit erfüllt, muss entweder

- (a) $(\chi^{\delta(H) \setminus F})^T \bar{x} = 1$ und $\bar{x}_e = 1$ für alle $e \in F$ oder
- (b) $\bar{x}_e = 0$ für alle $e \in \delta(H) \setminus F$ und $(\chi^F)^T \bar{x} = |F| - 1$ gelten.

Aus diesem Sachverhalt kann ein Algorithmus konstruiert werden, der ebenfalls $O(|E^*|)$ maximale Flüsse berechnet, um das primale Separierungsproblem zu lösen. Allerdings arbeitet er direkt auf dem Graphen G^* , wodurch er praktisch schneller ist.

Wenn man bei der Definition der Kamm-Ungleichungen von der 3.) oder der 4.) Bedingung verlangt, dass sie mit Gleichheit erfüllt sein muss, erhält man eine Klasse von Ungleichungen, die zwischen den Kamm- und den 2-Matching-Ungleichungen liegt: die so genannten *einfachen Kamm-Ungleichungen*. Letchford & Lodi beschreiben einen Algorithmus, der sowohl die primale als auch die Standard-Separierung dieser Ungleichungsklasse in polynomieller Zeit löst.

Asymmetrisches Handlungsreisendenproblem

Gegeben sei nun ein gerichteter Graph $G = (V, A)$, auf dessen Kanten eine Kostenfunktion c definiert ist. Für eine Knotenteilmenge $S \subset V$ bezeichnet man mit $\delta^+(S)$ die Schnittkanten, die von S ausgehen. Mit $\delta^-(S)$ sind analog die Schnittkanten gemeint, die in S enden. Damit lässt sich das Handlungsreisendenproblem als lineares Programm schreiben:

$$\begin{aligned} \min \quad & c^T x \\ \text{s.t.} \quad & (\chi^{\delta^+(i)})^T x = 1 \quad \forall i \in V \end{aligned} \tag{2.6}$$

$$(\chi^{\delta^-(i)})^T x = 1 \quad \forall i \in V \tag{2.7}$$

$$(\chi^{\delta^+(S)})^T x \geq 1 \quad \forall S \subset V: 2 \leq |S| < |V| \tag{2.8}$$

$$x_a \in \{0, 1\} \quad \forall a \in A$$

Die affine Hülle des zugehörigen Polytops wird auch hier von den Gleichungen (2.6) und (2.7) beschrieben, die man *eingehende* bzw. *ausgehende Gradgleichungen* nennt. Die Ungleichungen (2.8) sind äquivalent zu den Kurzzyklusbedingungen (2.5) und definieren ebenfalls Facetten.

Zwei Kanten $(i \rightarrow j)$ und $(p \rightarrow q)$ stehen zueinander in *Konflikt*, wenn sie beide nicht gleichzeitig in einer Tour auftreten können. Also stehen zwei Kanten genau dann zueinander in Konflikt, wenn genau eine der folgenden Bedingungen gilt:

- (a) $i = p$ und $j \neq q$
- (b) $i \neq p$ und $j = q$
- (c) $i = p$ und $j = q$

Ein *geschlossener alternierender Pfad* ist eine Folge a_1, \dots, a_k von Kanten, bei denen a_i mit a_{i+1} für $i = 1, \dots, k-1$ und a_1 mit a_k in Konflikt sind. Falls k ungerade ist, können höchstens $\lfloor \frac{k}{2} \rfloor$ der zugehörigen Variablen den Wert 1 annehmen. Diese Tatsache führt zur Definition der *schwachen ungeraden geschlossener-alternierender-Pfad Ungleichungen*.

Zur Standard-Separierung dieser Ungleichungsklasse definiert man einen Konfliktgraphen, auf dem man $|A^*|$ kürzeste Wege Probleme lösen muss, wobei analog zu oben $A^* := \{a \in A \mid x_a^* > 0\}$. Wird dazu der Algorithmus von Dijkstra benutzt, so erhält man eine Laufzeit von $O(|V||A^*|^2)$, die im schlimmsten Fall $O(|V|^5)$ sein kann. Die primale Separierung benötigt allerdings nur die Berechnung von $|V|$ kürzesten Wegen, wodurch sich die Laufzeit als $O(|V|^2|A^*|)$ ergibt. Da dies im schlimmsten Fall nur zu $O(|V|^4)$ führen kann, lässt sich die primale Separierung schneller lösen als die Standard-Separierung.

Kapitel 3

Max-Cut

Die Definition des Max-Cut Problems, für das in diesem Kapitel die primale Separierung untersucht wird, findet sich im einleitenden Abschnitt 3.1. Anschließend werden in 3.2 Begriffe aus der Polyedertheorie auf das Problem übertragen. Ein wichtiges Hilfsmittel für die primale Separierung ist die Umschalt-Operation, die zusammen mit einigen bekannten Resultaten in Abschnitt 3.3 vorgestellt wird. Die spezielle Anwendung dieser Operation für die primale Separierung zeigt Abschnitt 3.4. Schließlich wird in 3.5 die primale Separierung für die Dreiecksungleichungen und die Kreisungleichungen mit Hilfe der Umschalt-Operation hergeleitet. Für die Clique-Ungleichungen und die hypermetrischen Ungleichungen entwickeln wir ebenfalls in Abschnitt 3.5 Heuristiken zur primalen Separierung.

3.1 Das Max-Cut Problem

In vielen Bereichen des alltäglichen Lebens findet man Computer und andere elektronische Geräte, die die vielfältigsten Aufgaben übernehmen. Das Herzstück solcher Geräte bilden Halbleiterchips. Bei deren Entwicklung treten Probleme auf, die sich auf ein in der Informatik sehr bekanntes Problem transformieren lassen, nämlich auf das gewichtete Max-Cut Problem. Eine weitere Anwendung des Max-Cut Problems findet sich in der statistischen Physik, wo man unter anderem versucht, den Grundzustand von Ising-Spin-Glas Modellen zu berechnen. Auch hier ist eine Transformation auf das Max-Cut Problem möglich. Dadurch wird eine Untersuchung dieses Problems besonders motiviert. Die Problemstellung ist die folgende:

Gegeben sei ein gewichteter ungerichteter Graph (G, c) mit Knotenmenge V und Kantenmenge E . Finde eine Knotenteilmenge $S \subseteq V$, so dass die Kosten des durch S induzierten Schnittes $\delta(S)$ so groß wie möglich werden.

Abbildung 3.1 zeigt eine Beispiel-Instanz. Man versucht also, den maximalen Schnitt

$$\max_{S \subseteq V} c(\delta(S))$$

zu finden. Dieses Problem ist bekanntlich \mathcal{NP} -schwierig. Hat jede Kante die gleichen positiven Kosten, z.B. wenn $c(e) = 1$ für alle $e \in E$, dann spricht man

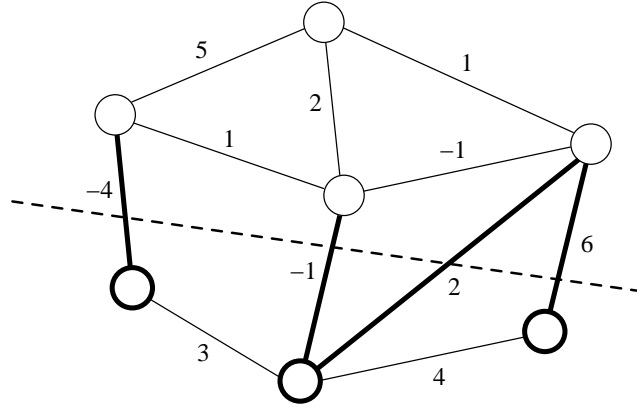


Abbildung 3.1: Eine Beispiel-Instanz des Max-Cut Problems. Die fett gedruckte Knotenmenge S induziert die fett gedruckten Schnittkanten $\delta(S)$. Die gestrichelte Linie veranschaulicht, warum man von einem Schnitt spricht. Dessen Kosten betragen hier $c(\delta(S)) = 3$.

nur von dem Max-Cut Problem. Auch im Folgenden wird nur von Max-Cut gesprochen, wobei jedoch immer die gewichtete Version gemeint ist.

3.2 Anwendung der Hilfsmittel aus der Polyedertheorie

Die im Abschnitt 1.3 allgemein definierten Begriffe lassen sich auf das Max-Cut Problem folgendermaßen übertragen. Sei \mathcal{I} nun die Menge aller Schnitte $\delta(S) \subseteq E$ des Graphen $G = (V, E)$. Dann ist das *Schnitt-Polytop* $P^{cut}(G)$ von G die konvexe Hülle der Inzidenzvektoren aller Schnitte von G :

$$P^{cut}(G) := \text{conv}\{\chi^{\delta(S)} \mid S \subseteq V\}$$

Analog wird der *Schnitt-Konus* definiert:

$$C^{cut}(G) := \text{cone}\{\chi^{\delta(S)} \mid S \subseteq V\}$$

Die Ecken von $P^{cut}(G)$ sind genau die Inzidenzvektoren $\chi^{\delta(S)}$ von Schnitten S . Somit ist das Schnitt-Polytop ein 0/1 Polytop. Da die leere Menge auch ein Schnitt ist und damit $\chi^\emptyset = 0 \in P^{cut}$, gilt $\dim(P^{cut}) = \text{rang}(P^{cut})$. Zusätzlich ist bekannt, dass das Schnitt-Polytop volldimensional ist, d.h. $\dim(P^{cut}) = m$. Somit gilt: $\dim(P^{cut}) = \text{affrang}(P^{cut}) - 1 = \text{rang}(P^{cut}) = m$.

Das Max-Cut Problem lässt sich als lineares Programm

$$\max\{c^T x \mid x \in P^{cut}(G)\}$$

schreiben. Da Max-Cut \mathcal{NP} -schwierig ist, kann, wie bereits gesehen, nicht erwartet werden, dass eine vollständige Darstellung der Form $\{x \in \mathbb{R}^m \mid Ax \leq b\}$ für $P^{cut}(G)$ in polynomieller Zeit gefunden werden kann. Deswegen beschränkt man sich bei der Separierung auf diejenigen Klassen von Ungleichungen, die man polynomiell oder mittels guter Heuristiken separieren kann.

3.3 Die Umschalt-Operation

Im Abschnitt 2.1 wurde der Unterschied zwischen primaler und Standard-Separierung dargestellt, welcher nur darin besteht, dass bei der primalen Separierung die zu findende Ungleichung zusätzlich zu allen anderen Bedingungen noch durch den gegebenen Vektor $\bar{x} \in \mathbb{R}^m$ mit Gleichheit erfüllt werden muss. Mit Hilfe der Umschalt-Operation, die sich bei Barahona & Mahjoub [4] finden lässt, kann man unter Berücksichtigung dieser Bedingung die primale Separierung auf die Standard-Separierung reduzieren. Dabei wird sich herausstellen, dass nur ein Spezialfall der Standard-Separierung auftritt, und somit die primale Separierung praktisch einfacher ist als die Standard-Separierung. Um diese Reduktion im nächsten Abschnitt durchführen zu können, werden nun Begriffe und Resultate, die im Zusammenhang mit der Umschalt-Operation stehen, präsentiert.

Für einen Graphen $G = (V, E)$ sei \mathcal{A} die Familie der Teilmengen von E , die einen Schnitt von G darstellen, also $\mathcal{A} := \{\delta(S) \mid S \subseteq V\}$. Unter der *symmetrischen Differenz* zweier Mengen versteht man allgemein

$$A \Delta B = (A \cup B) \setminus (A \cap B).$$

Für $A, B \in \mathcal{A}$ gilt, dass $A \Delta B \in \mathcal{A}$, denn die symmetrische Differenz zweier Schnitte ist wieder ein Schnitt:

Lemma 3.3.1: Seien $S, T \in V$. Dann gilt $\delta(S) \Delta \delta(T) = \delta(S \Delta T)$

Wir beweisen dies folgendermaßen:

Beweis: Da der Graph G ungerichtet ist, gilt aus Symmetriegründen für jede Kante $ij = ji$. Weiter ist

$$\begin{aligned} \delta(S) \cup \delta(T) &= \{ij \mid i \in S, j \notin S\} \cup \{ij \mid i \in T, j \notin T\} \\ &= \{ij \mid (i \in S \wedge j \notin S) \vee (i \in T \wedge j \notin T)\} \\ &= \{ij \mid [(i \in S \dot{\vee} i \in T) \wedge (j \in S \cap T \vee j \notin S \cup T)] \\ &\quad \vee [(i \in S \cap T) \wedge (j \notin S \cup T)] \\ &\quad \vee [i \in S \wedge i \notin T \wedge j \notin S \wedge j \in T]\} \end{aligned}$$

Außerdem

$$\begin{aligned} \delta(S) \cap \delta(T) &= \{ij \mid [(i \in S \cap T) \wedge (j \notin S \cup T)] \\ &\quad \vee [i \in S \wedge i \notin T \wedge j \notin S \wedge j \in T]\} \end{aligned}$$

Daraus folgt

$$\begin{aligned} \delta(S) \Delta \delta(T) &= (\delta(S) \cup \delta(T)) \setminus (\delta(S) \cap \delta(T)) \\ &= \{ij \mid (i \in S \dot{\vee} i \in T) \wedge (j \in S \cap T \vee j \notin S \cup T)\} \\ &= \{ij \mid i \in ((S \cup T) \setminus (S \cap T)) \wedge j \notin ((S \cup T) \setminus (S \cap T))\} \\ &= \delta((S \cup T) \setminus (S \cap T)) \\ &= \delta(S \Delta T) \end{aligned}$$

□

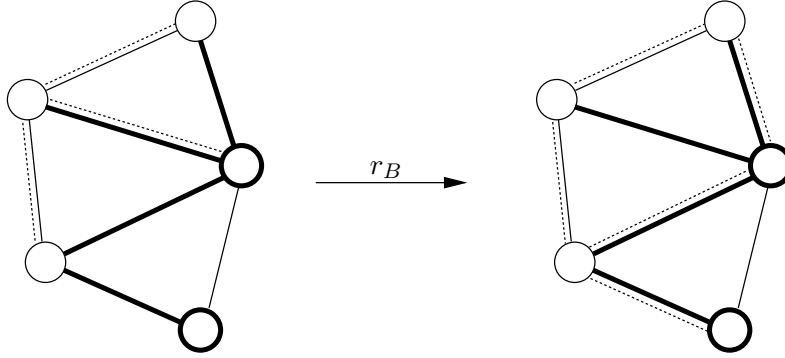


Abbildung 3.2: Veranschaulichung der Umschalt-Operation. Die gepunkteten Kanten sind diejenigen Kanten, für die $x_e = 1$ gilt. Die restlichen Kanten haben den x_e -Wert 0. Umgeschaltet wird bzgl. des Schnittes B , der durch die fett gedruckten Kanten dargestellt ist.

Die Menge \mathcal{A} ist somit abgeschlossen unter der symmetrischen Differenz. Daraus folgt insbesondere, dass $\emptyset \in \mathcal{A}$, denn $A \Delta A = \emptyset$.

Sei $x \in \mathbb{R}^m$ ein Vektor und $B \in \mathcal{A}$. Dann sei $x^B \in \mathbb{R}^m$ definiert durch

$$x_e^B := \begin{cases} -x_e & \text{falls } e \in B \\ x_e & \text{falls } e \in E \setminus B. \end{cases}$$

Somit kann eine Karte $r_B : \mathbb{R}^m \rightarrow \mathbb{R}^m$ mit $r_B(x) := x^B + \chi^B$ für $x \in \mathbb{R}^m$ erzeugt werden, wobei χ^B der charakteristische Vektor von B ist. Daraus folgt, dass

$$(r_B(x))_e := \begin{cases} 1 - x_e & \text{falls } e \in B \\ x_e & \text{falls } e \in E \setminus B. \end{cases}$$

Die so definierte Karte nennt man Umschaltkarte (im Englischen „switching mapping“). Offensichtlich ist sie eine affine Bijektion des \mathbb{R}^m . Außerdem gilt $r_B(r_B(x)) = x$ und damit $r_B^{-1} = r_B$. In Abbildung 3.2 wird die Anwendung dieser Operation an einem Beispiel gezeigt.

Besonders wichtig ist das Verhalten von Ungleichungen unter der Umschalt-Operation, welches im folgenden Satz, der in ähnlicher Form bei Deza & Laurent [10] (s.d. Corollary 26.3.5) zu finden ist, zusammengefasst wird:

Satz 3.3.2: Gegeben seien $a \in \mathbb{R}^m$, $a_0 \in \mathbb{R}$ und $B \in \mathcal{A}$. Dann gilt

$$\begin{aligned} a^T x \leq a_0 \text{ ist gültig für } P^{\text{cut}}(G) \\ \Leftrightarrow (a^B)^T x \leq a_0 - a^T \chi^B \text{ ist gültig für } r_B(P^{\text{cut}}(G)) \end{aligned}$$

Dieses Resultat wollen wir nun beweisen.

Beweis: Es gilt für alle Kanten $e \in E$

$$a_e(\chi^{C \Delta B} - \chi^B)_e = \begin{cases} -a_e & \text{falls } e \in C, e \in B \\ a_e & \text{falls } e \in C, e \notin B \\ 0 & \text{falls } e \notin C, e \in B \\ 0 & \text{falls } e \notin C, e \notin B \end{cases}$$

$$= \begin{cases} -a_e \chi_e^C & \text{falls } e \in B \\ a_e \chi_e^C & \text{falls } e \in E \setminus B \end{cases} = a_e^B \chi_e^C$$

Also

$$a^T(\chi^{C\Delta B} - \chi^B) = (a^B)^T \chi^C \quad (3.1)$$

Weiter ist

$$\begin{aligned} (r_B(\chi^C))_e &= \begin{cases} 1 - \chi_e^C & \text{falls } e \in B \\ \chi_e^C & \text{falls } e \in E \setminus B \end{cases} \\ &= \begin{cases} 1 & \text{falls } (e \in C \wedge e \notin B) \vee (e \notin C \wedge e \in B) \\ 0 & \text{falls } (e \in C \cup B) \vee (e \notin C \cap B) \end{cases} \\ &= \begin{cases} 1 & \text{falls } e \in C\Delta B \\ 0 & \text{falls } e \notin C\Delta B \end{cases} = \chi_e^{C\Delta B} \end{aligned}$$

Also

$$r_B(\chi^C) = \chi^{C\Delta B} \quad (3.2)$$

Sei o.B.d.A. $x = \chi^C \in r_B(P^{cut}(G))$. Somit

$$\begin{aligned} &(a^B)^T \chi^C \leq a_0 - a^T \chi^B \\ \stackrel{(3.1)}{\Leftrightarrow} &a^T(\chi^{C\Delta B} - \chi^B) \leq a_0 - a^T \chi^B \\ \Leftrightarrow &a^T \chi^{C\Delta B} \leq a_0 \\ \stackrel{(3.2)}{\Leftrightarrow} &a^T r_B(\chi^C) \leq a_0 \end{aligned}$$

wobei $r_B(\chi^C) \in r_B(r_B(P^{cut}(G))) = P^{cut}(G)$. Also ist $a^T x \leq a_0$ gültig für $P^{cut}(G)$ gdw. $(a^B)^T x \leq a_0 - a^T \chi^B$ gültig ist für $r_B(P^{cut}(G))$. \square

Die Ungleichung $(a^B)^T x \leq a_0 - a^T \chi^B$ wurde durch Umschalten von $a^T x \leq a_0$ bzgl. des Schnittes B gewonnen. Man nennt den Schnitt B *Wurzel* der Ungleichung $a^T x \leq a_0$, falls $a^T \chi^B = a_0$. Ist $a_0 = 0$ und B eine Wurzel, so lautet die umgeschaltete Ungleichung $(a^B)^T x \leq 0$. Dies bedeutet, dass die Homogenität einer Ungleichung durch das Umschalten bzgl. einer Wurzel erhalten bleibt.

Indem man die Menge aller $P^{cut}(G)$ definierenden Ungleichungen bzgl. des Schnittes B umschaltet, erhält man die Menge der $r_B(P^{cut}(G))$ definierenden Ungleichungen. Wir spezialisieren ein Korollar aus Deza & Laurent [10] (s.d. Corollary 26.3.4) und erhalten damit sogar

Satz 3.3.3: Sei $B \in \mathcal{A}$. Dann ist $r_B(P^{cut}(G)) = P^{cut}(G)$.

Dieses Ergebnis beweisen wir nun.

Beweis: Sei $P^{cut}(G) = \text{conv}\{\chi^{A_i} \mid A_i \in \mathcal{A}, i \in 1, \dots, k\}$, wobei k die Anzahl aller Schnitte von G ist und $A_i \neq A_j$ für $i \neq j$.

$$\begin{aligned} \Rightarrow r_B(P^{cut}(G)) &= \text{conv}\{r_B(\chi^{A_i}) \mid A_i \in \mathcal{A}, i \in 1, \dots, k\} \\ &\stackrel{(3.2)}{=} \text{conv}\{\chi^{A_i \Delta B} \mid A_i \in \mathcal{A}, i \in 1, \dots, k\} \end{aligned}$$

Es gilt:

$$\{\chi^{A_1}, \dots, \chi^{A_k}\} \text{ ist affin unabhängig}$$

$$\Leftrightarrow \{\chi^{A_1 \Delta B}, \dots, \chi^{A_k \Delta B}\} \text{ ist affin unabhängig.} \quad (3.3)$$

zu „ \Rightarrow “:

Seien $\chi^{A_1}, \dots, \chi^{A_l}$ affin unabhängig.

Annahme: $\chi^{A_1 \triangle B}, \dots, \chi^{A_l \triangle B}$ sind nicht affin unabhängig.

Dann existieren $\lambda_i \in \mathbb{R}, i \in 1, \dots, l$ für die gilt: $\sum_{i=1}^l \lambda_i \chi^{A_i \triangle B} = 0$, $\sum_{i=1}^l \lambda_i = 0$ und $\exists i \in 1, \dots, l : \lambda_i \neq 0$. Somit ergibt sich:

$$\sum_{i=1}^l \lambda_i \chi_e^{A_i} = 0, e \notin B \quad \text{und} \quad \sum_{i=1}^l \lambda_i (1 - \chi_e^{A_i}) = 0, e \in B$$

Für $e \in B$ gilt

$$\sum_{i=1}^l \lambda_i (1 - \chi_e^{A_i}) = 0 \Leftrightarrow \sum_{i=1}^l \lambda_i \chi_e^{A_i} = \sum_{i=1}^l \lambda_i \Leftrightarrow \sum_{i=1}^l \lambda_i \chi_e^{A_i} = 0$$

Insgesamt folgt damit $\sum_{i=1}^l \lambda_i \chi^{A_i} = 0$, $\sum_{i=1}^l \lambda_i = 0$ und $\exists i \in 1, \dots, l : \lambda_i \neq 0$. Das ist aber ein Widerspruch zur affinen Unabhängigkeit von $\chi^{A_1}, \dots, \chi^{A_l}$.

zu „ \Leftarrow “:

Laut „ \Rightarrow “ gilt: wenn $\{\chi^{A_1 \triangle B}, \dots, \chi^{A_l \triangle B}\}$ affin unabhängig ist, dann ist auch $\{\chi^{(A_1 \triangle B) \triangle B}, \dots, \chi^{(A_l \triangle B) \triangle B}\}$ affin unabhängig. Zudem ist

$$\chi^{A_i} = r_B(r_B(\chi^{A_i})) \stackrel{(3.2)}{=} r_B(\chi^{A_i \triangle B}) \stackrel{(3.2)}{=} \chi^{(A_i \triangle B) \triangle B}$$

Also ist (3.3) bewiesen.

Dies bedeutet nun, dass $\dim(P^{cut}(G)) = \dim(r_B(P^{cut}(G)))$.

Da \mathcal{A} unter der symmetrischen Differenz abgeschlossen ist, gilt $A_i \triangle B \in \mathcal{A}$ für $A_i, B \in \mathcal{A}$, also $\{A_i \triangle B \mid A_i \in \mathcal{A}, i \in 1, \dots, k\} \subseteq \mathcal{A}$.

Sei nun $C \in \mathcal{A}$. Wie im Beweis von (3.3) gesehen, gilt $\chi^C = \chi^{(C \triangle B) \triangle B}$, und damit $C = (C \triangle B) \triangle B$. Also folgt $\mathcal{A} \subseteq \{A_i \triangle B \mid A_i \in \mathcal{A}, i \in 1, \dots, k\}$.

Das bedeutet $\mathcal{A} = \{A_i \triangle B \mid A_i \in \mathcal{A}, i \in 1, \dots, k\}$, wodurch die Behauptung $r_B(P^{cut}(G)) = P^{cut}(G)$ bewiesen ist. \square

Somit ist die Klasse der für $P^{cut}(G)$ gültigen Ungleichungen unter Umschalten bzgl. des Schnittes B abgeschlossen. Darüber hinaus liefert (3.3) den Beweis für

Korollar 3.3.4:

$$\begin{aligned} a^T x \leq a_0 & \text{ beschreibt eine Facette von } P^{cut}(G) \\ \Leftrightarrow (a^B)^T x \leq a_0 - a^T \chi^B & \text{ beschreibt eine Facette von } P^{cut}(G) \end{aligned}$$

\square

Sei F die Seitenfläche von $P^{cut}(G)$, die durch $a^T x \leq a_0$ beschrieben wird, und $r_B(F) := \{r_B(x) \mid x \in F\}$. Die Wurzeln von $r_B(F)$ sind die Mitglieder $C \in \mathcal{A}$, für die $\chi^C \in r_B(F)$. Aus (3.2) geht hervor, dass diese Wurzeln genau die Vektoren $\chi^{A \triangle B}$ für $A \in \mathcal{A}, \chi^A \in F$ sind, und (3.3) zeigt, dass F und

$r_B(F)$ die gleiche Dimension haben. Somit stellt die Umschaltkarte eine 1-1 Korrespondenz zwischen den Seitenflächen des Polytops $P^{cut}(G)$ dar.

Hält man nun $A \in \mathcal{A}$ fest, so wird klar, dass die vollständige Menge aller Facetten von $P^{cut}(G)$ aus der Menge aller Facetten gewonnen werden kann, die den gegebenen Eckpunkt χ^A enthalten. Da insbesondere $\emptyset \in \mathcal{A}$ und nur Ungleichungen des Typs $a^T x \leq 0$ den Vektor $\chi^\emptyset = 0$ als Wurzel enthalten, kann man also die gesamte Flächenstruktur des Schnitt-Polytops $P^{cut}(G)$ aus der Flächenstruktur des Schnitt-Konus $C^{cut}(G)$ ableiten. Zusammengefasst ergibt sich als Gesamtergebnis (siehe auch Corollary 26.3.8 in [10]):

Satz 3.3.5: Sei

$$C^{cut}(G) = \{x \in \mathbb{R}^m \mid a_i^T x \leq 0 \text{ für } i = 1, \dots, k\}$$

Dann ist

$$P^{cut}(G) = \{x \in \mathbb{R}^m \mid (a_i^B)^T x \leq -a_i^T \chi^B \text{ für } i = 1, \dots, k \text{ und } B \in \mathcal{A}\}$$

□

3.4 Anwendung der Umschaltoperation

Wir wollen nun die oben vorgestellte Umschaltoperation mit den dazugehörigen Resultaten auf das primale Separierungsproblem anwenden. Unsere Vorgehensweise ist die folgende:

Bei der primalen Separierung hat man einen Vektor $x^* \in \mathbb{R}^m$ und einen Eckpunkt \bar{x} von $P^{cut}(G)$ gegeben, der zu einem Schnitt B von G korrespondiert. Somit gilt $\bar{x} = \chi^B$. Wir lösen nun das Standard-Separierungsproblem für den Schnitt-Konus $C^{cut}(G)$ und den Punkt $r_B(x^*)$ und nehmen an, dass wir eine verletzte Ungleichung $a^T x \leq 0$ erhalten. Diese schalten wir bzgl. des Schnittes B um. Die daraus resultierende Ungleichung $(a^B)^T x \leq -a^T \chi^B$ ist dann für das Schnitt-Polytop $P^{cut}(G)$ gültig, wird von x^* verletzt und laut Konstruktion von \bar{x} mit Gleichheit erfüllt.

Somit ist die primale Separierung für das Schnitt-Polytop äquivalent zur Standard-Separierung für den Schnitt-Konus. Bei der Standard-Separierung für den Schnitt-Konus müssen im Gegensatz zum Schnitt-Polytop nur die homogenen Ungleichungen betrachtet werden.

3.5 Primale Separierung für Ungleichungsklassen

Wie schon gesehen, lösen wir das primale Separierungsproblem für $P^{cut}(G)$, indem wir die Lösungen der Standard-Separierung für $C^{cut}(G)$ umschalten. Wird nur über eine Klasse \mathcal{F} von Ungleichungen separiert, so ist somit sicherzustellen, dass eine Ungleichung von \mathcal{F} nach Umschalten immer noch zu \mathcal{F} gehört. Die Klasse \mathcal{F} muss also abgeschlossen sein unter der Umschalt-Operation, d.h.

$$(a^T x \leq a_0) \in \mathcal{F} \Rightarrow ((a^B)^T x \leq a_0 - a^T \chi^B) \in \mathcal{F}$$

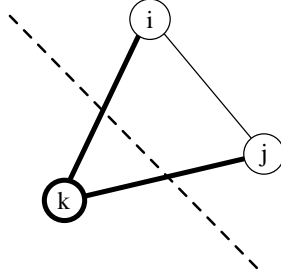


Abbildung 3.3: Eine Beispielsituation für die Anwendung der Dreiecksungleichungen.

Wegen den in Abschnitt 3.3 gezeigten Eigenschaften der Umschaltkarte folgt: Ist die Standard-Separierung für \mathcal{F} auf $C^{cut}(G)$ exakt, und ist \mathcal{F} unter Umschalten abgeschlossen, dann ist die primale Separierung für \mathcal{F} auf $P^{cut}(G)$ exakt.

Von den für das Max-Cut Problem bekannten Ungleichungsklassen werden nun einige von uns bzgl. der primalen Separierung untersucht.

3.5.1 Dreiecksungleichungen

Für jedes Tripel $i, j, k \in V$, für das gilt, dass alle Kanten $ij, ik, jk \in E$ vorhanden sind, gibt es vier Dreiecksungleichungen:

$$\begin{aligned} x_{ij} - x_{ik} - x_{jk} &\leq 0 \\ -x_{ij} + x_{ik} - x_{jk} &\leq 0 \\ -x_{ij} - x_{ik} + x_{jk} &\leq 0 \\ x_{ij} + x_{ik} + x_{jk} &\leq 2 \end{aligned}$$

Sie besagen, dass in einem Dreieck genau zwei oder keine Schnittkanten liegen. Eine Beispielsituation stellt Abbildung 3.3 dar. Es ist bekannt, dass sie Facetten von $P^{cut}(G)$ definieren. Darüber hinaus gilt unser

Satz 3.5.1: Dreiecksungleichungen sind unter Umschalten abgeschlossen.

Unser Beweis basiert auf der direkten Anwendung der Umschalt-Operation.

Beweis: O.B.d.A. ist es ausreichend, den Graphen $K_3 = (\{i, j, k\}, \{ij, ik, jk\})$ zu betrachten. Die 2. und 3. Ungleichung unterscheiden sich von der 1. nur durch Umm Nummerierung der Kanten. Also reicht es, das Verhalten von

$$\begin{aligned} x_{ij} - x_{ik} - x_{jk} &\leq 0 \\ x_{ij} + x_{ik} + x_{jk} &\leq 2 \end{aligned}$$

unter Umschalten bzgl. des Schnittes $\delta(S)$, $S \subseteq \{i, j, k\}$ zu untersuchen. Wenn $S = \emptyset$ oder $S = \{i, j, k\}$, dann ist $\chi^{\delta(S)} = 0$ und $a^{\delta(S)} = a$. Also verändert das Umschalten die Ungleichungen nicht. Ist $S = \{i\}$, so wird $x_{ij} - x_{ik} - x_{jk} \leq 0$ zu

$$-((1, -1, -1)^{\delta(i)})x \leq -(1, -1, -1)\chi^{\delta(i)} \Leftrightarrow -x_{ij} + x_{ik} - x_{jk} \leq 0$$

und für $S = \{j\}$ analog zu $-x_{ij} - x_{ik} + x_{jk} \leq 0$. Bei $S = \{k\}$ lautet die umgeschaltete Ungleichung

$$-((1, -1, -1)^{\delta(k)})x \leq -(1, -1, -1)\chi^{\delta(k)} \Leftrightarrow x_{ij} + x_{ik} + x_{jk} \leq 2.$$

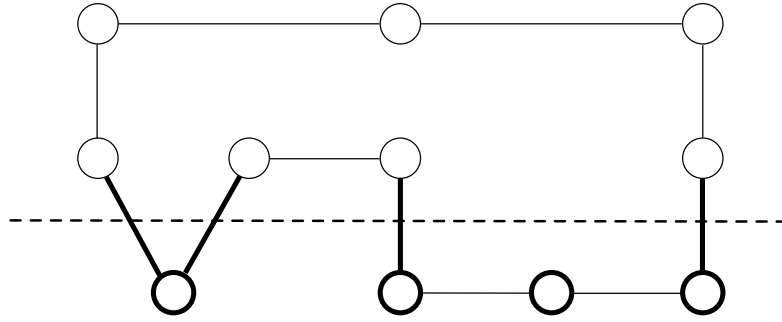


Abbildung 3.4: Ein Kreis kann nur eine gerade Anzahl von Schnittkanten enthalten.

Die Ungleichung $x_{ij} + x_{ik} + x_{jk} \leq 2$ wird bei $S = \{i\}$ zu

$$(1, 1, 1)^{\delta(i)} x \leq 2 - (1, 1, 1) \chi^{\delta(i)} \Leftrightarrow -x_{ij} - x_{ik} + x_{jk} \leq 0.$$

Unter $S = \{j\}$ bzw. $S = \{k\}$ verhält sie sich in analoger Weise. Enthält S zwei Elemente, so gilt $\delta(S) = \delta(\{i, j, k\} \setminus S)$. Damit ist die Behauptung bewiesen. \square

In einem vollständigen Graphen gibt es insgesamt $4 \binom{n}{3} = \frac{2}{3}(n^3 - 3n^2 + 2n)$ Dreiecksungleichungen, von denen $3 \binom{n}{3} = \frac{1}{2}(n^3 - 3n^2 + 2n)$ homogen sind. Löst man das jeweilige Separierungsproblem naiv durch Aufzählen aller Ungleichungen, so hat man sowohl im primalen als auch im Standard-Fall $O(n^3)$ viele Gleichungen. Beim primalen Problem müssen nur die homogenen und damit $\frac{1}{4}$ weniger Ungleichungen überprüft werden. Da sich für beide Fälle die gesamte Klasse der Dreiecksungleichungen in polynomieller Zeit aufzählen lässt, sind das primale und das Standard-Separierungsproblem exakt lösbar.

3.5.2 Kreisungleichungen

Sei C ein Kreis im Graphen G . Dieser kann nur eine gerade Anzahl von Schnittkanten enthalten, was auch anhand der Abbildung 3.4 ersichtlich ist. Die Kreisungleichungen bzgl. dieses Kreises spiegeln diesen Sachverhalt wider. Sie lauten folgendermaßen:

$$\sum_{e \in F} x_e - \sum_{e \in C \setminus F} x_e \leq |F| - 1 \text{ mit } F \subseteq C, |F| \text{ ungerade}$$

Sie sind facettendefinierend für $P^{cut}(G)$. Gilt $|C| = 3$, so erhält man die vier Dreiecksungleichungen. Somit umfasst die Klasse der Kreisungleichungen die Klasse der Dreiecksungleichungen. Barahona & Mahjoub [4] haben für die Standard-Separierung der Kreisungleichungen folgenden polynomiellen Algorithmus vorgestellt:

Sei der Punkt x^* gegeben. Nach Einsetzen dieses Punktes in die Kreisungleichung und Umformung erhält man

$$\sum_{e \in F} (1 - x_e^*) + \sum_{e \in C \setminus F} x_e^* \geq 1 \text{ mit } F \subseteq C, |F| \text{ ungerade}$$

Diese Ungleichung wird von x^* verletzt, wenn $\sum_{e \in F} (1 - x_e^*) + \sum_{e \in C \setminus F} x_e^* < 1$ ist. Ist dies der Fall, so liegt ein gewichteter Kreis C mit Gesamtkosten < 1 vor, bei dem eine ungerade Anzahl vieler Kanten die Kosten $1 - x_e^*$ und die restlichen Kanten die Kosten x_e^* haben.

Man sucht also nach einem minimal gewichteten Kreis, dessen Kantenkosten in dieser Weise gesetzt sind. Dazu wird ein Hilfsgraph G' erzeugt, der für jeden Knoten i von G zwei neue Knoten i' und i'' enthält. Für jede Kante ij in G werden in G' die Kanten $i'j'$ und $i''j''$ mit den Kosten x_{ij}^* und die Kanten $i'j''$ und $i''j'$ mit den Kosten $1 - x_{ij}^*$ gesetzt. Für jeden Knoten i von G wird nun in G' der kürzeste Weg von i' nach i'' berechnet. Von diesen n Wegen wird der Kürzeste ausgewählt. Er korrespondiert zu dem in G gesuchten Kreis C wie folgt: Ist in G' die Kante $i'j''$ oder $i''j'$ benutzt worden, so gehört in G die Kante ij zur Menge F und erhält die Kosten $1 - x_{ij}^*$. Die so erhaltene Menge F hat eine ungerade Kardinalität, weil in G' auf dem Weg von i' nach i'' ungerade viele Kanten des Typs $i'j''$ bzw. $i''j'$ benutzt werden müssen. Die restlichen Kanten des Typs $i'j'$ werden in G als Kanten ij zur Menge $C \setminus F$ hinzugefügt und erhalten die Kosten x_{ij}^* . Sind nun die Kosten des so in G gefundenen Kreises < 1 , so verletzt x^* die zugehörige Kreisungleichung.

Um mit Hilfe dieser Idee einen Algorithmus für die primale Separierung zu entwickeln, benötigen wir das Resultat von

Satz 3.5.2: Kreisungleichungen sind unter Umschalten abgeschlossen.

Da dieser Satz in unserem Kontext wichtig ist, wollen wir ihn beweisen.

Beweis: Für einen Graphen G sei ein Schnitt B , ein Kreis C und eine ungerade Menge $F \subseteq C$ gegeben. Die dadurch induzierte Kreisungleichung lautet

$$a^T x \leq |F| - 1$$

wobei

$$a_e = \begin{cases} 1 & \text{falls } e \in F \\ -1 & \text{falls } e \in C \setminus F \\ 0 & \text{falls } e \notin C \end{cases}$$

Die umgeschaltete Gleichung lautet dann

$$(a^B)^T x \leq |F| - a^T \chi^B - 1 \quad (3.4)$$

wobei nun

$$a_e^B = \begin{cases} 1 & \text{falls } e \in F \text{ und } e \notin B \\ -1 & \text{falls } e \in C \setminus F \text{ und } e \notin B \\ 0 & \text{falls } e \notin C \\ 1 & \text{falls } e \in C \setminus F \text{ und } e \in B \\ -1 & \text{falls } e \in F \text{ und } e \in B \end{cases}$$

$$= \begin{cases} 1 & \text{falls } e \in [(C \setminus F) \cap B] \cup [F \setminus B] \\ -1 & \text{falls } e \in [(C \setminus F) \setminus B] \cup [F \cap B] \\ 0 & \text{falls } e \notin C \end{cases}$$

Setze also $F' := [(C \setminus F) \cap B] \cup [F \setminus B]$. Dann gilt

$$\begin{aligned}
C \setminus F' &= [C \setminus [(C \setminus F) \cap B]] \cap [C \setminus [F \setminus B]] \\
&= [(C \setminus (C \setminus F)) \cup (C \setminus B)] \cap [(C \setminus F) \cup (C \cap B)] \\
&= [(F \cup (C \setminus B)) \cap (C \setminus F)] \cup [(F \cup (C \setminus B)) \cap (C \cap B)] \\
&= [(C \setminus B) \cap (C \setminus F)] \cup [F \cap B] \\
&= [(C \setminus F) \setminus B] \cup [F \cap B]
\end{aligned}$$

Damit ist klar, dass die linke Seite von (3.4) der zu F' gehörigen Kreisungleichung entspricht.

F' wird aus F gebildet, indem diejenigen Kanten, die in F und B enthalten sind, herausgenommen werden und diejenigen Kanten des Kreises, die nicht in F aber in B sind, hinzugenommen werden. Es gilt weiter

$$-a^T \chi^B = - \sum_{e \in F \cap B} 1 + \sum_{e \in (C \setminus F) \cap B} 1$$

Daraus folgt

$$|F'| = |F| - a^T \chi^B,$$

wodurch gezeigt wurde, dass die rechte Seite von (3.4) ebenfalls der zu F' gehörigen Kreisungleichung entspricht.

Da B ein Schnitt ist, gilt für den Kreis C , dass $|C \cap B|$ gerade ist. Wegen

$$|C \cap B| = |F \cap B| + |(C \setminus F) \cap B|$$

folgt für $|F \cap B|$ und $|(C \setminus F) \cap B|$, dass sie beide entweder gerade oder ungerade sind. Dadurch ist $-a^T \chi^B$ gerade, und es gilt: $|F|$ ungerade $\Rightarrow |F'|$ ungerade. \square

Die primale Separierung von Kreisungleichungen bzgl. eines Schnittes B , dem zugehörigen Punkt $\bar{x} = \chi^B$ und dem evtl. abzuschneidenden Punkt x^* lässt sich also auf die Standard-Separierung von homogenen Kreisungleichungen bzgl. des Punktes $r_B(x^*)$ reduzieren. Barahona & Titan [5] benutzten als Erste diesen Sachverhalt. Wir lösen das primale Separierungsproblem wie folgt:

Ist eine Kreisungleichung homogen, so gilt $|F| = 1$, und damit ist $F = \{f\}$. Die zu einem Kreis C gehörenden homogenen Kreisungleichungen lauten somit

$$x_f - \sum_{e \in C \setminus \{f\}} x_e \leq 0$$

Wir formen diese um und setzen $r_B(x^*)$ ein. Es ergibt sich

$$\sum_{e \in C \setminus \{f\}} (r_B(x^*))_e \geq (r_B(x^*))_f$$

In dem Graphen $G = (V, E)$ setzen wir also die Kantenkosten auf $r_B(x^*)$ und berechnen für alle Kanten $ij \in E$ den kürzesten Weg von i nach j . Gibt es eine Kante ij , deren Kosten größer sind als die Kosten des kürzesten Weges von i nach j , so haben wir eine von $r_B(x^*)$ verletzte homogene Kreisungleichung gefunden.

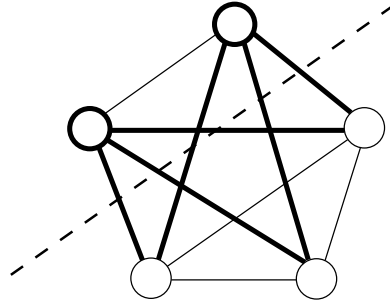


Abbildung 3.5: Im vollständigen Graphen K_5 dürfen höchstens $2 \cdot 3 = 6$ Kanten im Schnitt liegen.

Da für alle Kanten $0 \leq (x^*)_e \leq 1$ gilt, sind auch die Kantenkosten $r_B(x^*) \geq 0$. Deswegen kann die Berechnung der kürzesten Wege für alle Knotenpaare ebenfalls durch n Kürzeste Wege Berechnungen erfolgen. Sowohl die primale als auch die Standard-Separierung sind somit exakt und können beide mit derselben polynomiellen Laufzeit durchgeführt werden.

3.5.3 Clique-Ungleichungen

Gegeben sei eine Clique C der Kardinalität p . Die einen Schnitt $\delta(S)$ induzierende Knotenmenge S teilt dann C in zwei Teilmengen $C \setminus S$ und $C \cap S$ auf. Die Anzahl der innerhalb dieser Clique verlaufenden Schnittkanten beträgt somit $|C \setminus S| |C \cap S|$. Sie wird maximal, wenn sich die Kardinalitäten der beiden Teilmengen um höchstens 1 unterscheiden. Somit sind innerhalb einer Clique höchstens $\lfloor \frac{p}{2} \rfloor \lceil \frac{p}{2} \rceil$ Schnittkanten möglich, was durch die für $P^{cut}(G)$ gültige Clique-Ungleichung

$$\sum_{e \in E(C)} x_e \leq \lfloor \frac{p}{2} \rfloor \lceil \frac{p}{2} \rceil$$

widergespiegelt wird. Eine Clique der Kardinalität 5 mit dem dazugehörigen Schnitt veranschaulicht Abbildung 3.5. Bekanntlich definiert eine Clique-Ungleichungen genau dann eine Facette von $P^{cut}(G)$, wenn p ungerade ist. Deswegen wird im Weiteren davon ausgegangen, dass p ungerade ist. Sei also $p = 2k + 1$ mit $k \in \mathbb{Z}^+$. Dann gilt $\frac{p}{2} = k + \frac{1}{2}$. Durch ab- bzw. aufrunden von $\frac{p}{2}$ erhält man also k bzw. $k + 1$. Die Clique-Ungleichung lautet dann

$$\sum_{e \in E(C)} x_e \leq k(k + 1) \tag{3.5}$$

Wir betrachten nun das Verhalten der Clique-Ungleichung unter der Umschalt-Operation. Sei $a^T x \leq a_0$ eine Clique-Ungleichung auf dem Graphen G . Dann ist

$$a_e = \begin{cases} 1 & \text{falls } e \in E(C) \\ 0 & \text{falls } e \notin E(C) \end{cases}$$

und $a_0 = k(k+1)$. Nach Umschalten dieser Ungleichung bzgl. eines Schnittes B erhalten wir $(a^B)^T x \leq k(k+1) - a^T \chi^B$, wobei

$$a_e^B = \begin{cases} 1 & \text{falls } e \in E(C) \setminus B \\ -1 & \text{falls } e \in E(C) \cap B \\ 0 & \text{falls } e \notin E(C) \end{cases}$$

Erfüllt $\bar{x} = \chi^B$ die Ungleichung (3.5) mit Gleichheit, dann gilt

$$a^T \chi^B = \sum_{e \in E(C)} \bar{x}_e = k(k+1)$$

und die umgeschaltete Clique-Ungleichung ist homogen. Ist $E(C) \cap B \neq \emptyset$, dann haben einige Kanten in der umgeschalteten Ungleichung ein negatives Vorzeichen. Dies ist aber bei einer Clique-Ungleichung nicht möglich, weswegen die Clique-Ungleichungen unter Umschalten nicht abgeschlossen sind.

Da wir dennoch die Umschalt-Operation zur primalen Separierung benutzen wollen, müssen wir Ungleichungen finden, die umgeschaltet Clique-Ungleichungen ergeben. Welche Ungleichungen diese Eigenschaft haben, besagt unser

Satz 3.5.3: Sei C eine ungerade Clique der Kardinalität $2k+1$ mit $k \in \mathbb{Z}^+$. Sei $S \subseteq V$ mit $B = \delta(S)$ gegeben. Schaltet man

$$\sum_{e \in E(C) \setminus B} x_e \leq \sum_{e \in E(C) \cap B} x_e \text{ mit } ||C \setminus S| - |C \cap S|| = 1 \quad (3.6)$$

bzgl. B um, so erhält man die zu C gehörige Clique-Ungleichung (3.5).

Die Nebenbedingung $||C \setminus S| - |C \cap S|| = 1$ bedeutet, dass man die Clique C bzgl. S so partitionieren können muss, dass die Menge der Knoten, die in S sind, genau einen Knoten weniger bzw. mehr enthält als die Menge der Knoten, die nicht in S sind. Es folgt unser

Beweis: Nach Umformen der umzuschaltenden Ungleichung (3.6) ergibt sich

$$\sum_{e \in E(C) \setminus B} x_e - \sum_{e \in E(C) \cap B} x_e \leq 0$$

Dies entspricht $a^T x \leq a_0$ mit

$$a_e := \begin{cases} 1 & \text{falls } e \in E(C) \setminus B \\ -1 & \text{falls } e \in E(C) \cap B \\ 0 & \text{falls } e \notin E(C) \end{cases}$$

und $a_0 = 0$. Nach Umschalten bzgl. B erhalten wir $(a^B)^T x \leq -a^T \chi^B$ mit

$$a_e^B = \begin{cases} 1 & \text{falls } e \in E(C) \setminus B \wedge e \notin B \\ -1 & \text{falls } e \in E(C) \cap B \wedge e \notin B \\ -1 & \text{falls } e \in E(C) \setminus B \wedge e \in B \\ 1 & \text{falls } e \in E(C) \cap B \wedge e \in B \\ 0 & \text{falls } e \notin E(C) \end{cases} = \begin{cases} 1 & \text{falls } e \in E(C) \\ 0 & \text{falls } e \notin E(C) \end{cases}$$

Somit stimmt die linke Seite der umgeschalteten Ungleichung mit der von (3.5) überein. Nun zur rechten Seite:

Es gilt $||C \setminus S| - |C \cap S|| = 1$ und $|C \setminus S| + |C \cap S| = |C| = 2k + 1$. Daraus folgt entweder $|C \setminus S| = k + 1$ und $|C \cap S| = k$ oder $|C \setminus S| = k$ und $|C \cap S| = k + 1$. Weiter erhält man unter Berücksichtigung der Definition des Schnittes $B = \delta(S)$

$$-a^T \chi^B = |E(C) \cap B| = |C \cap S| |C \setminus S| = k(k + 1)$$

Also liefert das Umschalten von (3.6) bzgl. B die zu C gehörige Clique-Ungleichung (3.5), die von $\bar{x} = \chi^B$ mit Gleichheit erfüllt wird. \square

Im Hinblick auf den Abschnitt 3.5.4 über hypermetrische Ungleichungen leiten wir noch eine weitere Darstellungsart für die Ungleichung (3.6) her. Ist $|C \setminus S| > |C \cap S|$, so definieren wir für alle $i \in V$

$$b_i := \begin{cases} 1 & \text{falls } i \in C \setminus S \\ -1 & \text{falls } i \in C \cap S \\ 0 & \text{falls } i \notin C \end{cases} \quad (3.7)$$

Daraus folgt direkt

$$\sum_{i \in V} b_i = \sum_{i \in C} b_i = |C \setminus S| - |C \cap S| = 1$$

Gilt jedoch $|C \cap S| > |C \setminus S|$, so vertauschen wir in der Definition von b_i die -1 mit der 1 und erhalten dann

$$\sum_{i \in V} b_i = \sum_{i \in C} b_i = |C \cap S| - |C \setminus S| = 1$$

In beiden Fällen haben wir

$$\sum_{i \in V} |b_i| = \sum_{i \in C} |b_i| = |C| = 2k + 1$$

und es gilt

$$b_i b_j = \begin{cases} 1 & \text{falls } ij \in E(C) \setminus B \\ -1 & \text{falls } ij \in E(C) \cap B \\ 0 & \text{falls } ij \notin E(C) \end{cases}$$

$$\Rightarrow \sum_{ij \in E} b_i b_j = \sum_{ij \in E(C)} b_i b_j = \sum_{e \in E(C) \setminus B} x_e - \sum_{e \in E(C) \cap B} x_e$$

Demnach lässt sich (3.6) durch Nummerierung der Knoten $C = \{1, \dots, 2k + 1\}$ mittels der Definition (3.7) auch darstellen als

$$\sum_{1 \leq i < j \leq 2k+1} b_i b_j \leq 0 \quad \text{mit} \quad \sum_{i=1}^{2k+1} b_i = 1 \quad \text{und} \quad \sum_{i=1}^{2k+1} |b_i| = 2k + 1 \quad (3.8)$$

Es stellt sich nun die Frage, ob das Problem, Clique-Ungleichungen für ungerade Cliques primal zu separieren, äquivalent dazu ist, Ungleichungen

des Typs (3.6) bzw. (3.8) standardmäßig zu separieren. Dies können wir durch folgende Überlegung positiv beantworten:

Für jeden Schnitt $B = \delta(S)$, der die Ungleichung (3.5) mit Gleichheit erfüllt, gilt

$$|E(C) \cap B| = \sum_{e \in E(C)} \bar{x}_e = k(k+1)$$

Sei nun ein Schnitt $B' = \delta(S')$ mit $|C \cap S'| < k$ gegeben, also $|C \cap S'| = k - l$ für $l = 1, \dots, k - 1$. Somit ist $|C \setminus S'| = k + 1 + l$, woraus folgt

$$\begin{aligned} |E(C) \cap B'| &= |C \cap S'| |C \setminus S'| = (k - l)(k + 1 + l) \\ &= k(k + 1) - l(l + 1) < k(k + 1) \end{aligned}$$

Also sind die gesuchten Cliques genau die mit $|C \cap S| = k$ oder $|C \cap S| = k + 1$. Diese in (3.6) aufgeführte Nebenbedingung bedeutet, dass man einen vollständigen bipartiten Subgraphen von G finden muss, dessen eine Partition genau ein Element mehr enthält als die andere. Da dieses Problem \mathcal{NP} -schwierig ist, lässt sich dies auch auf die primale Separierung der Clique-Ungleichungen für ungerade Cliques übertragen. Die Darstellungsform (3.8) wird sich in Abschnitt 3.5.4 als $(2k + 1)$ -gonale reine hypermetrische Ungleichung erweisen, deren Separierung sich ebenfalls als \mathcal{NP} -schwierig herausstellen wird.

Abschließend wollen wir unsere Heuristik vorstellen, mit deren Hilfe sich verletzte Clique-Ungleichungen primal separieren lassen. Die Knotenteilmenge $S \subseteq V$ und der dazugehörige Schnitt $B = \delta(S)$ seien gegeben. Wir versuchen, verletzte Ungleichungen des Typs (3.6) zu finden, um diese anschließend bzgl. B umschalten zu können. Dazu benutzen wir bereits bekannte Heuristiken, um uns eine möglichst große Clique C liefern zu lassen. Aus dieser entfernen wir sukzessiv die Knoten, um die Nebenbedingung aus (3.6) zu erfüllen. Um die dortige Ungleichung möglichst zu verletzen, werden dabei zuerst die Knoten aus C entfernt, die am wenigsten zur Verletzung der Ungleichung beitragen.

Unsere Heuristik lautet wie folgt:

- Schritt 1: Berechne eine möglichst große Clique C .
- Schritt 2: Falls $|C| < 3$, stoppe mit „Keine verletzte Ungleichung gefunden.“
- Schritt 3: Falls $|C \setminus S| - |C \cap S| = 0$, gehe zu Schritt 7.
- Schritt 4: Falls $|C \setminus S| - |C \cap S| > 1$, entferne den Knoten l aus C , für den gilt: $l \in C \setminus S$ und $\min_{i \in C \setminus S} \sum_{\substack{e \in E(C) \setminus B \\ e \in \delta(i)}} x_e - \sum_{\substack{e \in E(C) \cap B \\ e \in \delta(i)}} x_e$ wird für $i = l$ angenommen, und gehe zu Schritt 2.
- Schritt 5: Falls $|C \setminus S| - |C \cap S| < -1$, entferne den Knoten l aus C , für den gilt: $l \in C \cap S$ und $\min_{i \in C \cap S} \sum_{\substack{e \in E(C) \setminus B \\ e \in \delta(i)}} x_e - \sum_{\substack{e \in E(C) \cap B \\ e \in \delta(i)}} x_e$ wird für $i = l$ angenommen, und gehe zu Schritt 2.

- Schritt 6: Falls $\sum_{e \in E(C) \setminus B} x_e > \sum_{e \in E(C) \cap B} x_e$, gib die verletzte Ungleichung aus und stoppe.
- Schritt 7: Entferne den Knoten l aus C , für den gilt:

$$\min_{i \in C} \sum_{\substack{e \in E(C) \setminus B \\ e \in \delta(i)}} x_e - \sum_{\substack{e \in E(C) \cap B \\ e \in \delta(i)}} x_e$$
wird für $i = l$ angenommen. Gehe zu Schritt 2.

Wir vermuten, dass die Wahrscheinlichkeit, mittels dieser Heuristik eine verletzte Ungleichung zu finden, stark von der Anzahl der in C enthaltenen Elemente aus S abhängt. Da es gilt, die Nebenbedingung aus (3.6) zu erfüllen, ist es vorteilhaft, wenn $|C \cap S| \approx \frac{|C|}{2}$. In diesem Fall würde Schritt 6 recht früh aufgerufen werden. Nach dem ersten Aufruf von Schritt 6 werden jeweils zwei Knoten aus C entfernt, bevor Schritt 6 erneut aufgerufen wird. Ist $|C \cap S|$ allerdings sehr klein oder annähernd so groß wie $|C|$, so werden sehr viele Knoten aus C herausgenommen, bis die Überprüfung in Schritt 6 zum ersten Mal erfolgen kann. Da dieser Fall vermutlich häufiger auftritt als der Idealfall, gehen wir davon aus, dass diese Heuristik nur beschränkt einsetzbar ist.

3.5.4 Hypermetrische Ungleichungen

Sei G ein vollständiger Graph. Definiere $Q(b) : \mathbb{R}^n \rightarrow \mathbb{R}^m$ mit $Q(b)_{ij} := b_i b_j$. Die Ungleichung

$$Q(b)^T x = \sum_{1 \leq i < j \leq n} b_i b_j x_{ij} \leq 0 \text{ mit } b \in \mathbb{Z}^n \text{ und } \sum_{i=1}^n b_i = 1$$

nennt man die durch b definierte *hypermetrische Ungleichung*. Sie ist für $C^{cut}(G)$ und damit auch für $P^{cut}(G)$ gültig. Die Bedingung, dass G vollständig sein muss, kann eingeschränkt werden. Entweder definiert man die hypermetrische Ungleichung nur auf einem vollständigen Subgraphen von G , oder der Trägergraph, d.h., der Graph, der aus allen Knoten $i \in V$ gebildet wird, für die $b_i \neq 0$ ist, muss vollständig sein.

Gilt für alle $i \in V$, dass $b_i \in \{-1, 0, 1\}$, so spricht man von einer *reinen* hypermetrischen Ungleichung. Man nennt sie $(2k + 1)$ -gonal, wenn $\sum_{i=1}^n |b_i| = 2k + 1$. Diese Bedingung ist äquivalent dazu, dass $\sum_{i|b_i < 0} |b_i| = k$, denn mit $\sum_{i=1}^n b_i = 1$ gilt:

$$\begin{aligned} \sum_{i=1}^n |b_i| = 2k + 1 &\Leftrightarrow \sum_{i|b_i \geq 0} b_i - \sum_{i|b_i < 0} b_i = 2k + 1 \\ \Leftrightarrow -2 \sum_{i|b_i < 0} b_i = 2k &\Leftrightarrow \sum_{i|b_i < 0} |b_i| = k \end{aligned}$$

Bekannt ist, dass jede reine hypermetrische Ungleichung eine Facette definiert. Eine komplette Beschreibung aller facettendefinierenden hypermetrischen Ungleichungen gibt es zum jetzigen Zeitpunkt jedoch nicht. Bei Deza &

Laurent [10] finden sich einige notwendige und hinreichende Bedingungen für die Facettialeigenschaft von hypermetrischen Ungleichungen.

Ein Beispiel für eine reine 3-gonale hypermetrische Ungleichung ist folgende Dreiecksungleichung aus Abschnitt 3.5.1:

$$x_{ij} - x_{ik} - x_{jk} \leq 0$$

Man erhält sie, indem man $b_i = b_j := 1$, $b_k := -1$ und für $h \in V \setminus \{i, j, k\}$ $b_h := 0$ setzt.

Auch bei der umgeschalteten Clique-Ungleichung in ihrer Darstellungsform (3.8) handelt es sich um eine reine $(2k+1)$ -gonale hypermetrische Ungleichung. Die Separierung solcher Ungleichungen ist \mathcal{NP} -schwierig. Deza & Laurent [10] zeigten dies, indem sie das \mathcal{NP} -schwierige Problem, einen vollständigen bipartiten Subgraphen von G zu finden, dessen eine Partition genau ein Element mehr enthält als die andere, auf die Separierung $(2k+1)$ -gonaler hypermetrischer Ungleichungen reduzierten.

Die Komplexität der Separierung aller hypermetrischen Ungleichungen ist nicht bekannt, aber das Resultat zur Separierung der $(2k+1)$ -gonalen hypermetrischen Ungleichungen lässt vermuten, dass es sich hierbei ebenfalls um ein schwieriges Problem handelt. Wir wollen nun eine Heuristik zur primalen Separierung von hypermetrischen Ungleichungen entwickeln. Da sich in den letzten Abschnitten die Umschalt-Operation in diesem Zusammenhang bewährt hat, werden wir zuerst untersuchen, wie sich hypermetrische Ungleichungen unter Umschalten verhalten.

Seien dazu ein Schnitt $B = \delta(S)$ und eine hypermetrische Ungleichung in der Form $a^T x \leq 0$ gegeben, wobei $(a^T)_{ij} = b_i b_j$ für $1 \leq i < j \leq n$ und $\sum_{i=1}^n b_i = 1$, $b_i \in \mathbb{Z}$. Nach Umschalten bzgl. des Schnittes B erhält man $(a^B)^T x \leq -a^T \chi^B$. Da diese Ungleichung i.A. nicht homogen ist, sind die hypermetrischen Ungleichungen nicht abgeschlossen unter der Umschalt-Operation. Für alle Kanten $ij \in B$ wird in der umgeschalteten Ungleichung der Koeffizient von x_{ij} zu $-b_i b_j$. Um also weiterhin eine Darstellung der Art $\sum_{1 \leq i < j \leq n} b'_i b'_j x_{ij} \leq -a^T \chi^B$ benutzen zu können, kann man o.B.d.A. $b'_i = -b_i$ für alle $i \in S$ und $b'_i = b_i$ für alle $i \notin S$ setzen. Dadurch verliert aber die Bedingung $\sum_{i=1}^n b'_i = 1$ ihre Gültigkeit. Somit handelt es sich bei der umgeschalteten Ungleichung nicht mehr um eine hypermetrische Ungleichung, sondern um einen Spezialfall der in der Literatur als *Gap-Ungleichung* bekannten Ungleichungsklasse (siehe z.B. [10]). Von dieser weiß man, dass sie die hypermetrischen Ungleichungen enthält und dass sie abgeschlossen unter der Umschalt-Operation ist. Da es aufgrund der Definition dieser Ungleichungsklasse jedoch schon \mathcal{NP} -schwierig ist, eine Gap-Ungleichung konkret anzugeben, und über die Separierung dieser Ungleichungsklasse nichts bekannt ist, werden wir diesen Ansatz nicht weiter verfolgen.

Will man wenigsten die Homogenität einer hypermetrischen Ungleichung unter der Umschalt-Operation erhalten, so muss man die Bedingung $a^T \chi^B = 0$ erfüllen. Dies entspricht aber genau der Bedingung, die bei der primalen Separierung gegeben ist, nämlich dass \bar{x} die Ungleichung mit Gleichheit erfüllen soll. Demnach werden wir also auf direktem Wege versuchen, hypermetrische Ungleichungen primal zu separieren. Dazu betrachten wir zunächst

folgende Gleichung:

$$a^T \chi^B = \sum_{1 \leq i < j \leq n} b_i b_j \chi^B = \sum_{i \in S} b_i \cdot \sum_{j \notin S} b_j = \sum_{i \in S} b_i \left(1 - \sum_{i \in S} b_i\right)$$

Die Bedingung $a^T \chi^B = 0$ ist also dazu äquivalent, dass $\sum_{i \in S} b_i \in \{0, 1\}$.

De Simone & Rinaldi [9] stellten eine Heuristik vor, die hypermetrische Ungleichungen standardmäßig separiert. Durch Umformung der Nebenbedingung zu $b_n = 1 - \sum_{i=1}^{n-1} b_i$ und mit Hilfe einer symmetrischen $(n-1) \times (n-1)$ -Matrix Q und eines $(n-1)$ -dimensionalen Vektors d , die man beide mittels x^* definiert, schrieben sie die hypermetrische Ungleichung als

$$\sum_{1 \leq i < j \leq n} b_i b_j x_{ij}^* = b^T Q b + d^T b \quad (3.9)$$

wobei $b = (b_1, \dots, b_{n-1})^T$. Da wir hypermetrische Ungleichungen primal separieren wollen, müssen wir die Nebenbedingung $\sum_{i=1}^n b_i = 1$ verschärfen zu

$$\sum_{i \in S} b_i = 0 \wedge \sum_{j \notin S} b_j = 1 \quad \text{oder} \quad \sum_{i \in S} b_i = 1 \wedge \sum_{j \notin S} b_j = 0 \quad (3.10)$$

Dadurch können wir die hypermetrische Ungleichung ebenfalls in der Form (3.9) schreiben, allerdings mit $b \in \mathbb{Z}^{n-2}$, einer symmetrischen $(n-2) \times (n-2)$ -Matrix Q und einem $(n-2)$ -dimensionalen Vektor d , die wir beide mittels x^* definieren. Wir brauchen also durch die verschärfte Bedingung, die sich aus der primalen Separierung ergibt, eine Dimension weniger zur Darstellung der hypermetrischen Ungleichung als quadratische Form als im Standard-Fall. Wie dies im Detail aussieht, beschreibt unsere

Beobachtung 3.5.4: Im Folgenden ist der Übersichtlichkeit halber mit x immer x^* gemeint. Da der Graph G ungerichtet ist, gilt für alle Kanten $x_{ij} = x_{ji}$. Wir wählen zufällig ein $s \in S$ sowie ein $t \notin S$ und setzen, um die Bedingung (3.10) zu erfüllen,

$$\begin{aligned} b_s &= - \sum_{i \in S, i \neq s} b_i \\ b_t &= 1 - \sum_{i \notin S, i \neq t} b_i \end{aligned} \quad (3.11)$$

Da es sich bei (3.10) um eine Oder-Bedingung handelt, müssen wir ebenfalls betrachten, dass b_s und b_t gesetzt werden können als $b_s = 1 - \sum_{i \in S, i \neq s} b_i$ und $b_t = - \sum_{i \notin S, i \neq t} b_i$. Wir betrachten o.B.d.A. nur den ersten Fall und definieren die symmetrische $(n-2) \times (n-2)$ -Matrix Q mit den Einträgen q_{ij} für $i, j \in \{1, \dots, n\} \setminus \{s, t\}$, wobei

$$q_{ij} = \begin{cases} -x_{is} & \text{falls } i \in S \text{ und } i = j \\ -x_{it} & \text{falls } i \notin S \text{ und } i = j \\ \frac{1}{2}(x_{ij} - x_{is} - x_{js}) & \text{falls } i \in S, j \in S \text{ und } i \neq j \\ \frac{1}{2}(x_{ij} - x_{it} - x_{js} + x_{st}) & \text{falls } i \in S, j \notin S \\ \frac{1}{2}(x_{ij} - x_{is} - x_{jt} + x_{st}) & \text{falls } i \notin S, j \in S \\ \frac{1}{2}(x_{ij} - x_{it} - x_{jt}) & \text{falls } i \notin S, j \notin S \text{ und } i \neq j \end{cases}$$

Den $(n - 2)$ -dimensionalen Vektor d definieren wir für $i \in \{1, \dots, n\} \setminus \{s, t\}$ mittels

$$d_i = \begin{cases} x_{it} - x_{st} & \text{falls } i \in S \\ x_{it} & \text{falls } i \notin S \end{cases}$$

Damit gilt für den $(n - 2)$ -dimensionalen Vektor b mit den Einträgen b_i für $i \in \{1, \dots, n\} \setminus \{s, t\}$

$$b^T Q b + d^T b = \sum_{1 \leq i < j \leq n} b_i b_j x_{ij}$$

Beweis: Bei allen in der folgenden Rechnung vorkommenden Indizes i und j gilt die Bedingung $i, j \notin \{s, t\}$. Wir schreiben dies nicht explizit auf, um ein wenig die Übersichtlichkeit zu wahren. Es gilt

$$\begin{aligned} & b^T Q b + d^T b \\ &= \sum_{i \in S} -b_i^2 x_{is} + \sum_{i \notin S} -b_i^2 x_{it} + \sum_{\substack{i \in S, j \in S \\ i < j}} b_i b_j (x_{ij} - x_{is} - x_{js}) \\ & \quad + \sum_{i \in S, j \notin S} b_i b_j (x_{ij} - x_{it} - x_{js} + x_{st}) + \sum_{\substack{i \notin S, j \notin S \\ i < j}} b_i b_j (x_{ij} - x_{it} - x_{jt}) \\ & \quad \quad \quad + \sum_{i \in S} b_i (x_{it} - x_{st}) + \sum_{i \notin S} b_i x_{it} \\ &= \sum_{1 \leq i < j \leq n} b_i b_j x_{ij} + \sum_{i \in S} \underbrace{\left(-b_i^2 - \sum_{j \in S, i \neq j} b_i b_j \right)}_{*1} x_{is} + \sum_{i \notin S} \underbrace{\left(-\sum_{j \in S} b_i b_j \right)}_{*2} x_{is} \\ & \quad + \sum_{i \in S} \underbrace{\left(-\sum_{j \notin S} b_i b_j + b_i \right)}_{*3} x_{it} + \sum_{i \notin S} \underbrace{\left(-b_i^2 - \sum_{j \notin S, i \neq j} b_i b_j + b_i \right)}_{*4} x_{it} \\ & \quad \quad \quad + \underbrace{\left(\sum_{i \in S, j \notin S} b_i b_j - \sum_{i \in S} b_i \right)}_{*5} x_{st} \end{aligned}$$

Mittels (3.11) erhalten wir

$$\begin{aligned} *1 &= b_i \left(-b_i - \sum_{j \in S, i \neq j} b_j \right) = b_i b_s \\ *2 &= b_i \left(-\sum_{j \in S} b_j \right) = b_i b_s \\ *3 &= b_i \left(1 - \sum_{j \notin S} b_j \right) = b_i b_t \\ *4 &= b_i \left(1 - b_i - \sum_{j \notin S, i \neq j} b_j \right) = b_i b_t \\ *5 &= -\sum_{i \in S} b_i \left(1 - \sum_{j \notin S} b_j \right) = b_s b_t \end{aligned}$$

Damit können wir unsere Gleichungskette weiterführen:

$$\begin{aligned}
&= \sum_{1 \leq i < j \leq n} b_i b_j x_{ij} + \sum_{i \in S} b_i b_s x_{is} + \sum_{i \notin S} b_i b_s x_{is} \\
&\quad + \sum_{i \in S} b_i b_t x_{it} + \sum_{i \notin S} b_i b_t x_{it} + b_s b_t x_{st} \\
&= \sum_{\substack{1 \leq i < j \leq n \\ i, j \in \{s, t\}}} b_i b_j x_{ij}
\end{aligned}$$

□

Der Rest unserer Heuristik orientiert sich an der Vorgehensweise von De Simone & Rinaldi, die wir nun der Vollständigkeit halber mit den auf unsere Bedürfnisse angepassten Dimensionen skizzieren.

Um eine verletzte hypermetrische Ungleichung zu finden, reicht es nach der Beobachtung aus, einen Vektor $b \in \mathbb{Z}^{n-2}$ zu finden mit $b^T Q b + d^T b > 0$. Sei dazu $v \in \mathbb{R}^{n-2}$. Ist v ganzzahlig und gilt $v^T Q v + d^T v > 0$, dann wird die durch

$$\begin{aligned}
b_i &= v_i \text{ für } i \in \{1, \dots, n\} \setminus \{s, t\} \\
b_s &= - \sum_{i \in S, i \neq s} b_i \\
b_t &= 1 - \sum_{i \notin S, i \neq t} b_i
\end{aligned}$$

definierte hypermetrische Ungleichung von x^* verletzt. Ist v nicht ganzzahlig, so setzt man $w_i = \lfloor v_i \rfloor - \alpha_i + \beta_i$ für $i \in \{1, \dots, n\} \setminus \{s, t\}$, wobei α_i und β_i 0/1 Variablen sind. Da w durch Konstruktion ein ganzzahliger Vektor ist, werden der Richtungsvektor v und eine Belegung der Variablen α_i und β_i gesucht, für die $w^T Q w + d^T w > 0$ gilt. Den Richtungsvektor v setzt man anfangs auf 0 und berechnet damit eine Belegung von α_i und β_i . Erhält man keine verletzte Ungleichung, so berechnet man v mit Hilfe von Mitteln aus der Analysis und führt anschließend die Berechnung von α_i und β_i erneut durch.

Die Belegung der Variablen α_i und β_i lässt sich bei gegebenem v folgendermaßen berechnen: Setzt man den Vektor $w = \lfloor v \rfloor - \alpha + \beta$ in $b^T Q b + d^T b$ ein, so erhält man

$$w^T Q w + d^T w = x^T B x + a^T x + C$$

wobei

$$x = \left(\alpha_1, \dots, \alpha_{s-1}, \alpha_{s+1}, \dots, \alpha_{t-1}, \alpha_{t+1}, \dots, \alpha_n, \right. \\
\left. \beta_1, \dots, \beta_{s-1}, \beta_{s+1}, \dots, \beta_{t-1}, \beta_{t+1}, \dots, \beta_n \right)^T$$

und B eine symmetrische $2(n-2) \times 2(n-2)$ -Matrix, a ein $2(n-2)$ -dimensionaler Vektor und C eine Konstante ist. Also gilt

$$w^T Q w + d^T w > 0 \Leftrightarrow x^T B x + a^T x > -C$$

Man betrachtet nun das Problem

$$\begin{aligned}
\max \quad & x^T B x + a^T x \\
\text{s.t.} \quad & x \in \{0, 1\}^{2(n-2)}
\end{aligned} \tag{3.12}$$

Jede Lösung von (3.12), die einen Wert größer als $-C$ hat, liefert eine durch x^* verletzte hypermetrische Ungleichung. Da (3.12) ein unbeschränktes 0/1 quadratisches Problem ist, kann man es lösen, indem man es auf ein Max-Cut Problem auf dem vollständigen Graphen $K_{2(n-2)+1}$ mit geeignet definierten Kantenkosten transformiert (siehe De Simone [8]). Wir sind hier jedoch nicht an der optimalen Lösung sondern an den Lösungen interessiert, deren Kosten größer als $-C$ sind, weswegen zur Lösung dieses speziellen Max-Cut Problems Heuristiken zum Einsatz kommen.

Abschließend können wir sagen, dass die primale Separierung der hypermetrischen Ungleichungen in unserem Fall keinen Vorteil in Bezug auf die Geschwindigkeit bringt. Es wird lediglich eine Dimension weniger zur Darstellung von (3.9) als im Standard-Fall benötigt.

Kapitel 4

Gradbeschränkte minimale Spannbäume

In Abschnitt 4.1 dieses Kapitels wird das Problem, einen gradbeschränkten minimalen Spannbaum zu finden, zusammen mit einer Formulierung als lineares Programm präsentiert. Einige für das Problem bekannte Resultate aus der Polyedertheorie werden in 4.2 dargestellt. Über die Formulierung als lineares Programm hinaus sind zudem weitere Ungleichungsklassen bekannt, die in Abschnitt 4.3 zusammengefasst werden. Der Kern dieses Kapitels bildet Abschnitt 4.4. Hier leiten wir für die Zusammenhangsungleichungen, die speziellen sowie die allgemeinen Kreis-Eliminierungsungleichungen die primale Separierung her. Abschließend wird in 4.5 eine Heuristik zur Konstruktion eines gradbeschränkten Spannbaums beschrieben.

4.1 Die Problemstellung

Heutzutage ist der Einsatz des Computers aus einigen Bereichen des menschlichen Lebens nicht mehr wegzudenken. Er ist in der Wirtschaft und Industrie, bei Dienstleistungen und in vielen anderen Zweigen wie Medizin, Forschung etc. zu einem wichtigen Bestandteil geworden. Auch privat wird er immer häufiger unter anderem als Kommunikationsmittel eingesetzt. Sein Vorteil liegt im Austausch der Daten mit anderen, was über Netzwerke wie z.B. Firmennetzwerke oder dem Internet realisiert wird. Demzufolge hat der Entwurf solcher Netzwerke einen hohen Stellenwert bekommen. Um ein effizientes und zugleich kostengünstiges Netzwerk zu entwickeln, muss man häufig folgendes Grundproblem lösen:

Man hat n Computer oder Netzwerkknotenpunkte wie z.B. so genannte Hubs, Switches, Routers oder ähnliches. Diese sollen alle so miteinander verbunden werden, dass die Kosten, die unter anderem durch die Kabellänge entstehen, minimiert werden. Technische Einschränkungen wie die maximale Anzahl b_i der Ports eines Switches i müssen dabei berücksichtigt werden.

Es gibt noch einige andere Modelle, die zum Design von Netzwerken angewendet werden (siehe [17]). Bei allen liegt jedoch das oben beschriebene Problem zu Grunde, einen minimalen aufspannenden Baum zu finden, bei dem der Knotengrad für Knoten i nicht größer als b_i ist. Man sucht nach einem so

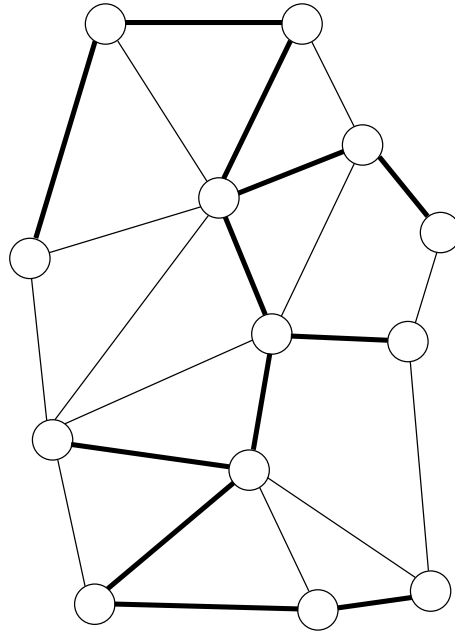


Abbildung 4.1: Beispiel eines gradbeschränkten Spannbaumes. Die fett gedruckten Linien bilden den Spannbaum. Die Gradbeschränkung ist hier für alle Knoten gleich und beträgt $b = 3$.

genannten *gradbeschränkten minimalen Spannbaum* (im Englischen „degree-constrained minimum spanning tree“). Wir wollen uns deswegen genauer mit diesem Problem beschäftigen und formulieren es dazu als lineares Programm:

$$\begin{aligned} \min \quad & c^T x \\ \text{s.t.} \quad & \sum_{e \in E(S)} x_e \leq |S| - 1 \quad \forall S \subset V: 2 \leq |S| < |V| \end{aligned} \quad (4.1)$$

$$\sum_{e \in E} x_e = |V| - 1 \quad (4.2)$$

$$\sum_{e \in \delta(i)} x_e \leq b_i \quad \forall i \in V \quad (4.3)$$

$$x_e \in \{0, 1\} \quad \forall e \in E$$

Für die Kantenkosten gilt hierbei $c_e \in \mathbb{R}$ für alle $e \in E$. Die Ungleichungen (4.1) verhindern, dass Kreise in der Lösungsmenge auftreten, weswegen sie *Kreis-Eliminierungsungleichungen* heißen. Dass der gefundene Baum wirklich ein Spannbaum ist, stellt die *Spannbaum-Gleichung* (4.2) sicher. Die Einschränkungen bzgl. der Anzahl der Verbindungen werden durch die *Knoten-grad-Ungleichungen* (4.3) beschrieben. Dabei muss gelten, dass $b_i \in \mathbb{Z}^{\geq 2}$ für alle Knoten $i \in V$. Dies stellt nur dann eine wirkliche Beschränkung dar, wenn $b_i \leq n - 1$ gilt. Es ist bei einigen Modellierungen auch möglich, dass $b_i = b$ für alle Knoten $i \in V$ gilt. Ist dies der Fall mit $b = 2$, so sucht man nach einem minimalen Hamiltonischen Pfad, was bekanntlich ein \mathcal{NP} -schwieriges Problem ist. Garey & Johnson [16] zeigten auch von dem Problem, einen gradbeschränk-

ten minimalen Spannbaum zu finden, dass es \mathcal{NP} -schwierig ist, indem sie das symmetrisches Handlungsreisendenproblem darauf reduzierten.

4.2 Resultate aus der Polyedertheorie

Das zu dem Problem, einen minimalen gradbeschränkten Spannbaum zu finden, gehörige Polytop

$$P^{dcmst}(G) := \left\{ x \in \mathbb{R}^m \mid \begin{aligned} \sum_{e \in E(S)} x_e &\leq |S| - 1 & \forall S \subset V: 2 \leq |S| < |V|, \\ \sum_{e \in E} x_e &= |V| - 1, \\ \sum_{e \in \delta(i)} x_e &\leq b_i & \forall i \in V, \\ 0 \leq x_e &\leq 1 & \forall e \in E \end{aligned} \right\}$$

ist offensichtlich ein 0/1 Polytop. Uns ist allerdings nicht bekannt, ob es weitere Resultate wie z.B. zur Dimension oder zu Facetten dieses Polytops gibt.

Entfernt man die Knotengrad-Ungleichungen (4.3) aus oben genanntem linearen Programm, so erhält man das als *minimaler Spannbaum* bekannte Problem. Dieses ist in polynomieller Zeit lösbar mit Hilfe der bekannten Algorithmen von Prim und Kruskal. Über das dazugehörige Polytop

$$P^{mst}(G) := \left\{ x \in \mathbb{R}^m \mid \begin{aligned} \sum_{e \in E(S)} x_e &\leq |S| - 1 & \forall S \subset V: 2 \leq |S| < |V|, \\ \sum_{e \in E} x_e &= |V| - 1, \\ 0 \leq x_e &\leq 1 & \forall e \in E \end{aligned} \right\}$$

gibt es einige Erkenntnisse. Man weiß, dass die Dimension dieses Polytops $m-1$ beträgt. Edmonds [11, 12] bewies, dass die Inzidenzvektoren von Spannbäumen eines zusammenhängenden Graphen G genau die Ecken des durch (4.1), (4.2) und $x_e \geq 0 \quad \forall e \in E$ beschriebenen Polytops sind. Damit gab er eine vollständige Ungleichungsbeschreibung des Minimalen-Spannbaum-Polytops. Die Ungleichungen

$$\sum_{e \in E(S)} x_e \leq |S| - 1 \quad \forall S \subseteq V: |S| \geq 2$$

stellen genau dann Facetten dieses Polytops dar, wenn entweder $|S| = 2$ und die Graphen $(S, E(S))$ und $(V \setminus S, E(V \setminus S))$ zusammenhängend sind oder $|S| \geq 3$ und $(S, E(S))$ 2-fach zusammenhängend ist.

4.3 Weitere gültige Ungleichungen

Es gibt noch weitere für gradbeschränkte minimale Spannbäume gültige Ungleichungen [38]. Zum einen darf kein Knoten isoliert sein, was durch

$$\sum_{e \in \delta(i)} x_e \geq 1 \quad \forall i \in V \tag{4.4}$$

überprüft wird. Des Weiteren dürfen keine unzusammenhängenden Strukturen auftreten. Dies stellen die *Zusammenhangsungleichungen*

$$\sum_{e \in \delta(S)} x_e \geq 1 \quad \forall S \subset V: 2 \leq |S| < |V| \quad (4.5)$$

sicher. Anstatt bei den Kreis-Eliminierungsungleichungen (4.1) alle möglichen Mengen S zuzulassen, kann man sich nur auf Kreise $(C, E(C))$ beschränken. Dies führt zu den *speziellen Kreis-Eliminierungsungleichungen*

$$\sum_{e \in E(C)} x_e \leq |C| - 1 \quad \forall C \subset V: |C| < |V| \text{ und } (C, E(C)) \text{ ist ein Kreis} \quad (4.6)$$

4.4 Primale Separierung für Ungleichungsklassen

4.4.1 Triviale Separierungen

Die Separierung der Gleichung (4.2) und der Ungleichungen (4.3) sowie (4.4) liegen auf der Hand. Wir erwähnen sie hier der Vollständigkeit halber.

Spannbaum-Gleichung

Wir zerlegen die Gleichung (4.2) in die zwei Ungleichungen

$$\begin{aligned} \sum_{e \in E} x_e &\leq |V| - 1 \\ - \sum_{e \in E} x_e &\leq -(|V| - 1) \end{aligned}$$

Eine exakte Separierung dieser Ungleichungen lässt sich durch Aufsummieren über alle Kanten in Zeit $O(m)$ erreichen. Dabei handelt es sich immer um eine primale Separierung, denn \bar{x} ist Inzidenzvektor eines gradbeschränkten minimalen Spannbäume und erfüllt deswegen die Gleichung (4.2).

Ungleichungen bzgl. des Knotengrads

Die Ungleichungen (4.3) und (4.4) können jeweils mit demselben Ansatz separiert werden. Man berechnet dazu für alle Knoten $i \in V$ die Werte $\sum_{e \in \delta(i)} x_e^*$ und $\sum_{e \in \delta(i)} \bar{x}_e$. Dies benötigt eine Gesamtlaufzeit von $O(m)$. Anschließend überprüft man bei der Standard-Variante jeden Knoten auf Verletzung bzgl. der aktuellen Lösung x^* und bei der primalen Separierung noch zusätzlich auf Gleichheit bzgl. \bar{x} . Beide Verfahren sind demnach exakt.

4.4.2 Zusammenhangsungleichungen

Betrachten wir nun die Zusammenhangsungleichungen (4.5). Wählt man S so, dass der durch S induzierte Subgraph $(S, E(S))$ nicht zusammenhängend ist, dann kann die dadurch gegebene Ungleichung nie von einem durch \bar{x} repräsentierten Spannbaum mit Gleichheit erfüllt werden. Somit sind die Zusammenhangsungleichungen nicht scharf. Sie können allerdings verschärft werden

zu

$$\sum_{e \in \delta(S)} x_e \geq s(S, E(S)) \quad \forall S \subset V: 2 \leq |S| < |V|$$

Diese Tatsache lässt vermuten, dass es zur Separierung von (4.5) ausreichend ist, nur diejenigen Knotenteilmengen $S \subset V$ zu betrachten, deren zugehöriger Subgraph $(S, E(S))$ zusammenhängend ist. Dies wird durch unsere folgende Beobachtung bestätigt:

Beobachtung 4.4.1: Wenn es eine Knotenteilmenge $S \subset V$ mit $\sum_{e \in \delta(S)} x_e^* < 1$ gibt, dann existiert eine Knotenteilmenge $S' \subset V$ mit $\sum_{e \in \delta(S')} x_e^* < 1$ und

$(S', E(S'))$ ist zusammenhängend.

Beweis: Sei $S \subset V$ so, dass $\sum_{e \in \delta(S)} x_e^* < 1$ gilt und $(S, E(S))$ nicht zusammenhängend ist. Weiter sei $S = S_1 \dot{\cup} \dots \dot{\cup} S_l$ die Zerlegung von S in diejenigen l Untermengen, deren Subgraphen $(S_k, E(S_k))$ mit $k = 1, \dots, l$ die l Zusammenhangskomponenten von $(S, E(S))$ darstellen. Angenommen, es existiert kein $k \in \{1, \dots, l\}$, für das $\sum_{e \in \delta(S_k)} x_e^* < 1$ gilt. Dann folgt mittels $\delta(S) = \delta(S_1) \dot{\cup} \dots \dot{\cup} \delta(S_l)$:

$$1 > \sum_{e \in \delta(S)} x_e^* = \sum_{k=1}^l \left(\sum_{e \in \delta(S_k)} x_e^* \right) \geq l$$

und damit $l < 1$, was ein Widerspruch ist. □

Die Standard-Separierung von (4.5) erfolgt durch die Berechnung eines minimalen Schnittes bzgl. den Kosten x^* . Sind dessen Kosten echt kleiner als 1, so hat man eine verletzte Ungleichung gefunden. Diese Separierung ist exakt und hat, wenn man den Phasenschnitt-Algorithmus zur Berechnung des minimalen Schnittes einsetzt, eine Laufzeit von $O(mn + n^2 \log n)$.

Zur Herleitung unserer primalen Separierung von (4.5) bestimmen wir zunächst diejenigen Knotenteilmengen $S \subset V$, für die bei einem durch \bar{x} gegebenem Spannbaum T die Gleichung $\sum_{e \in \delta(S)} \bar{x}_e = 1$ gilt. Laut unserer Beobachtung reicht es aus, nur die $S \subset V$ mit zusammenhängendem Subgraphen $(S, E(S))$ zu betrachten. S muss also so beschaffen sein, dass es eine Zusammenhangskomponente bildet und der Schnitt $\delta(S)$ im Spannbaum T genau eine Kante enthält. Demzufolge induziert jede Kante e des Spannbaumes T eine der gesuchten Mengen S wie folgt:

Sei $ij = e$ eine Kante von T und sei $T_i = (S_i, E_T(S_i))$ der größte in T liegende Teilbaum mit der Wurzel i , der die Kante e nicht enthält. Dann ist $S = S_i$. Da es sich bei T um einen Spannbaum handelt, gilt $S_j = V \setminus S$. Abbildung 4.2 verdeutlicht diese Situation.

In der Praxis kann man die zu einem gegebenen Knoten i gehörige Menge S_i durch Tiefensuche im Spannbaum T ermitteln. Man löscht die Kante e aus dem Spannbaum T und startet von der Wurzel i aus. Alle bei der Durchsuchung von T besuchten Knoten gehören zur Menge S_i . Da T ein Spannbaum ist, lässt

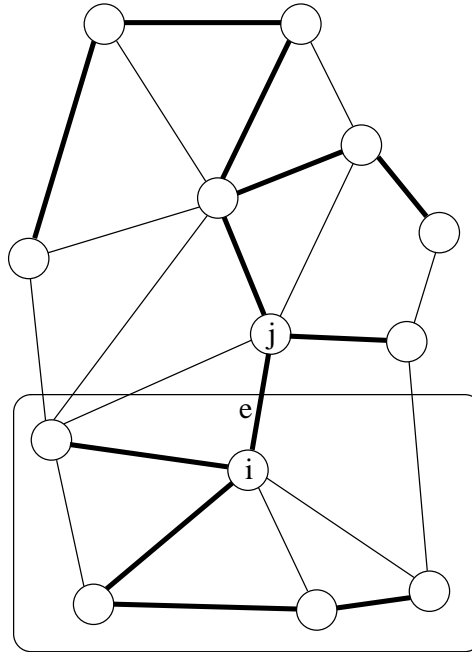


Abbildung 4.2: Beispiel einer primalen Zusammenhangsungleichung. Der Kasten enthält den Teilbaum T_i .

sich S_i somit in $O(n)$ bestimmen. Mit Hilfe dessen lässt sich dann auch der Wert des Schnittes $\delta(S_i)$ bzgl. x^* in $O(m)$ berechnen.

Unsere primale Separierung der Zusammenhangsungleichungen (4.5) muss also für jede Kante ij von T die Menge S_i und den Wert $\sum_{e \in \delta(S_i)} x_e^*$ berechnen und prüfen, ob die Ungleichung verletzt wird. Sie ist somit exakt und benötigt eine Gesamtlaufzeit von $O(mn + n^2)$, da T als Spannbaum $|n| - 1$ Kanten enthält. Gegenüber der Standard-Variante zeichnet sie sich durch eine schnellere Laufzeit aus.

4.4.3 Spezielle Kreis-Eliminierungsungleichungen

Die speziellen Kreis-Eliminierungsungleichungen (4.6) lassen sich wie folgt umformen:

$$\sum_{e \in E(C)} x_e \leq |C| - 1 \Leftrightarrow \sum_{e \in E(C)} 1 - x_e \geq 1$$

Raidl [38] schlug deswegen zur Standard-Separierung dieser Ungleichungsklasse vor, die Kosten aller Kanten e auf $c'(e) = 1 - x_e^*$ zu setzen. Sie sind dadurch weiterhin nicht negativ, da $0 \leq x_e^* \leq 1$ für alle Kanten $e \in E$ gilt. Dann sucht man nach einem minimalen Kreis C in (G, c') . Falls ein solcher existiert und dessen Kosten < 1 sind, gilt damit $\sum_{e \in E(C)} x_e^* > |C| - 1$, und man hat eine verletzte Ungleichung gefunden. Zur Suche nach einem minimalen Kreis C in (G, c') benutzte Raidl folgenden Algorithmus:

Setze $c'_{\min} = \infty$ und $C = \emptyset$.
Für alle Kanten e in G :

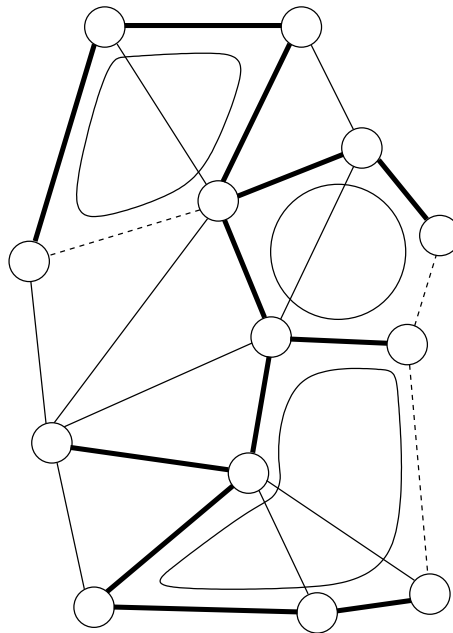


Abbildung 4.3: Drei Beispiele einer primalen speziellen Kreis-Eliminierungsungleichung. Die gestrichelten Kanten bilden jeweils mit den dazugehörigen Baumkanten die angedeuteten Fundamentalkreise.

Setze $ij = e$.

Entferne e aus G .

Finde den kürzesten Weg von i nach j mit den Kosten c'_{path} .

Falls dieser existiert und $c'_{\text{path}} + c'(e) < c'_{\text{min}}$:

Setze $c'_{\text{min}} = c'_{\text{path}} + c'(e)$ und speichere die zugehörige Menge C .

Füge e zu G hinzu.

Falls der Kürzeste-Wege-Algorithmus von Dijkstra zur Suche nach einem kürzesten Weg von i nach j benutzt wird, hat die oben genannte exakte Standard-Separierung von (4.6) über alle Kreise eine Laufzeit von $O(mn^2)$, da die Laufzeit von Dijkstra $O(n^2)$ beträgt und dieser für jede Kante aufgerufen werden muss.

Um die Ungleichungen (4.6) primal zu separieren, suchen wir nach Kreisen C , die die Bedingung $\sum_{e \in E(C)} \bar{x}_e = |C| - 1$ erfüllen. Da ein Kreis über $|C|$ Knoten $|C|$ viele Kanten enthält, muss folglich bei den gesuchten Kreisen genau eine Kante keine Baumkante sein. Es handelt sich somit um Fundamentalkreise, was durch die Abbildung 4.3 veranschaulicht wird. Unsere primale Separierung muss also für jede Kante e , die keine Baumkante von T ist, den dazugehörigen Fundamentalkreis finden. Unter diesen wird der bzgl. den Kosten $c'(e) = x^*$ maximale Fundamentalkreis C gesucht. Dazu verwenden wir diesen Algorithmus:

Setze $c'_{\text{max}} = 0$ und $C = \emptyset$.

Für alle Kanten e in G , die keine Baumkanten von T sind:

Setze $ij = e$.

Finde den Pfad von i nach j in T mit den Kosten c'_{path} .

Falls $c'_{\text{path}} + c'(e) > c'_{\text{max}}$:

Setze $c'_{\text{max}} = c'_{\text{path}} + c'(e)$ und speichere die zugehörige Menge C .

Falls ein maximaler Fundamentalkreis C gefunden wurde, überprüfen wir, ob für ihn $\sum_{e \in E(C)} x_e^* > |C| - 1$ gilt, und er damit eine verletzte Ungleichung darstellt.

Unser Verfahren zur primalen Separierung von (4.6) ist also exakt. Den Pfad von i nach j im Spannbaum T suchen wir mittels Breitensuche. Da T als Spannbaum nur $n - 1$ Kanten hat, benötigt die Breitensuche dazu $O(n)$ Zeit. Weiterhin müssen die Fundamentalkreise für alle Kanten berechnet werden, die keine Kanten des Spannbaumes T sind. Dies sind $m - (n - 1)$ viele, und deshalb beträgt die Gesamtlaufzeit unseres Verfahrens $O(mn)$. Es ist somit schneller als die vorgestellte Standard-Separierung.

4.4.4 Kreis-Eliminierungsungleichungen

Für die Kreis-Eliminierungsungleichungen (4.1) lässt sich die gleiche Beobachtung machen wie bei den Zusammenhangsungleichungen in Abschnitt 4.4.2:

Ist \bar{x} der Inzidenzvektor eines Spannbaumes und wird die Knotenteilmenge $S \subset V$ so gewählt, dass der durch sie induzierte Subgraph $(S, E(S))$ nicht zusammenhängend ist, so kann die dadurch gegebene Ungleichung nie von \bar{x} mit Gleichheit erfüllt werden. Folglich sind auch die Zusammenhangsungleichungen in der gegebenen Form nicht scharf. Man kann sie ebenfalls verschärfen. Sie lauten dann:

$$\sum_{e \in E(S)} x_e \leq |S| - s(S, E(S)) \quad \forall S \subset V: 2 \leq |S| < |V|$$

Dies führt erneut zu der Vermutung, dass es ausreicht, zur Separierung von (4.1) nur diejenigen Knotenteilmengen $S \subset V$ zu betrachten, deren zugehöriger Subgraph $(S, E(S))$ zusammenhängend ist. Unsere folgende Beobachtung, die in dieser Art auch bei Padberg & Wolsey [36] zu finden ist, belegt die Richtigkeit dieser Vermutung.

Beobachtung 4.4.2: Wenn es eine Knotenteilmenge $S \subset V$ mit

$$\sum_{e \in E(S)} x_e^* > |S| - 1 \text{ gibt, dann existiert eine Knotenteilmenge } S' \subset V \text{ mit}$$

$$\sum_{e \in E(S')} x_e^* > |S'| - 1 \text{ und } (S', E(S')) \text{ ist zusammenhängend.}$$

Beweis: Sei $S \subset V$ derart, dass $\sum_{e \in E(S)} x_e^* > |S| - 1$ gilt und $(S, E(S))$ nicht zusammenhängend ist. Erneut sei $S = S_1 \dot{\cup} \dots \dot{\cup} S_l$ die Zerlegung von S in diejenigen l Untermengen, deren Subgraphen $(S_k, E(S_k))$ mit $k = 1, \dots, l$ die l Zusammenhangskomponenten von $(S, E(S))$ darstellen. Wir nehmen an, dass es kein $k \in \{1, \dots, l\}$ mit $\sum_{e \in E(S_k)} x_e^* > |S_k| - 1$ gibt. Daraus folgt

$$|S| - 1 < \sum_{e \in E(S)} x_e^* = \sum_{k=1}^l \left(\sum_{e \in E(S_k)} x_e^* \right) \leq |S| - l$$

Dies führt zu dem Widerspruch $l < 1$.

□

Die Standard-Separierung, die Padberg & Wolsey [36] vorschlugen, separiert die Kreis-Eliminierungsungleichungen (4.1) in der gegebenen Form, d.h., es werden alle möglichen Knotenteilmengen $S \subset V$ berücksichtigt. Sie berechnen dazu das Minimum der submodularen Funktion $g(S) = |S| - \sum_{e \in E(S)} x_e^*$, indem sie ein geeignetes Netzwerk definieren. Auf diesem suchen sie nach einem speziellen minimalen Schnitt, den sie aus $n - 2$ Maximaler-Fluss-Berechnungen mit der jeweiligen Anpassung des Netzwerkes erhalten. Wird zur Berechnung der maximalen Flüsse die Maximum-Distanz-Version des Goldberg-Tarjan-Algorithmus verwendet, dann ergibt sich für die Separierung von Padberg & Wolsey, welche exakt ist, eine Gesamtlaufzeit von $O(n^4)$.

Um unsere primale Separierung der Kreis-Eliminierungsungleichungen (4.1) zu entwickeln, wollen wir uns zuerst wieder Gedanken zu den möglichen Knotenteilmengen $S \subset V$ machen. Die Einschränkung, dass die durch sie induzierten Subgraphen $(S, E(S))$ zusammenhängend sein sollen, wird wieder durch unsere Beobachtung gerechtfertigt. Außerdem soll für den durch \bar{x} gegebenen Spannbaum T die Gleichung $\sum_{e \in E(S)} \bar{x}_e = |S| - 1$ gelten. Deswegen enthält $(S, E(S))$ einen Teilbaum T' von T , der ein Spannbaum für $(S, E(S))$ ist.

Sei der Spannbaum T derart, dass er eine Wurzel und $n - 1$ Blätter hat. Daraus resultieren $\sum_{k=1}^n \binom{n}{k} = 2^n - 1$ Möglichkeiten für T' , wobei $k = |E(S)|$. Somit ist es nicht sinnvoll, alle möglichen Mengen S , deren durch sie induzierte Subgraphen $(S, E(S))$ einen zusammenhängenden Teil des Spannbaumes T enthalten, zu enumerieren.

Wir wenden uns nun wieder der von Padberg & Wolsey vorgeschlagenen Standard-Separierung zu und werden diese so verändern, dass sie primal separiert. Zuerst müssen wir die von ihnen benutzte submodulare Funktion $g(S) = |S| - \sum_{e \in E(S)} x_e^*$ modifizieren. Unsere Idee ist die folgende:

Wir wollen eine Funktion $f : S \rightarrow \mathbb{R}$ für $\emptyset \neq S \subseteq V$ definieren, für die Folgendes gilt:

$$f(S) = \begin{cases} g(S) & \text{falls } \sum_{e \in E(S)} \bar{x}_e = |S| - 1 \\ > 1 & \text{sonst} \end{cases}$$

Diese Funktion kann also nur dann < 1 werden, wenn $\sum_{e \in E(S)} x_e^* > |S| - 1$ und $\sum_{e \in E(S)} \bar{x}_e = |S| - 1$ gilt. Enthält der durch S induzierte Graph $(S, E(S))$ keinen zusammenhängenden Teilbaum von T , dann soll also $f(S) > 1$ gelten. Die Wahl einer Menge S , die nicht die zusätzliche Bedingung der primalen Separierung erfüllt, wird somit bestraft. Unsere primale Separierung läuft damit auf die Suche einer Menge S hinaus, die f minimiert. Hat dieses Minimum einen Wert < 1 , so hat man eine verletzte Ungleichung gefunden.

Um f die oben geforderte Eigenschaft zu geben, definieren wir

$$f(S) := |S| - \sum_{e \in E(S)} x_e^* + D \left(|S| - \sum_{e \in E(S)} \bar{x}_e - 1 \right),$$

wobei D eine positive Konstante ist. Diese muss demnach so gewählt werden, dass sie für den Fall $|S| - \sum_{e \in E(S)} \bar{x}_e \geq 2$ größer als $\sum_{e \in E(S)} x_e^*$ ist. Um D so

wählen zu können, bestimmen wir den größtmöglichen Wert, den $\sum_{e \in E(S)} x_e^*$ annehmen kann. Es gilt

$$\sum_{e \in E(S)} x_e^* \leq |E(S)| \leq \sum_{k=1}^{|S|-1} k = \frac{1}{2}(|S|^2 - |S|)$$

Gleichheit gilt hier, wenn G zusammenhängend und $x_e^* = 1$ für alle Kanten $e \in E$ ist. Wir nehmen dies o.B.d.A. an. Man erkennt leicht, dass die Funktion $\frac{1}{2}(|S|^2 - |S|)$ für $|S| \geq 1$ streng monoton steigend ist. Da $|S| \leq n$ gilt und folglich für $|S| = n$ der größtmögliche Wert angenommen wird, setzen wir $D := \frac{1}{2}(n^2 - n)$.

Da g submodular ist, lässt sich vermuten, dass dies auch auf f zutrifft. Dies bestätigt sich in unserer

Beobachtung 4.4.3: f ist submodular, d.h., es gilt

$$f(X \cup Y) + f(X \cap Y) \leq f(X) + f(Y) \quad \forall X, Y \subseteq V$$

Beweis: Die Funktion f lässt sich schreiben als $f(S) = g(S) + Dh(S) - D$ mit $h(S) = |S| - \sum_{e \in E(S)} \bar{x}_e$. h ist also von gleicher Form wie g . Da genau wie für x_e^* für alle Kanten e die Ungleichungen $0 \leq \bar{x}_e \leq 1$ gelten, ist auch h submodular. Die Addition einer Konstanten und auch die Multiplikation einer submodularen Funktion mit einer Konstanten erhält die Submodularität, wodurch auch $Dh(S) - D$ submodular ist. Die aus der Addition zweier submodularer Funktionen resultierende Funktion ist ebenfalls submodular, weswegen dies auch für f gilt. □

Um nach der Menge S zu suchen, die unsere submodulare Funktion f minimiert, orientieren wir uns an der Vorgehensweise von Padberg & Wolsey. Analog zu ihnen definieren wir anhand des Graphen $G = (V, E)$ ein Netzwerk (G^*, c') , das zusätzlich zu den n Knoten aus V eine Quelle 0 und eine Senke $n + 1$ erhält. Für alle Knoten $i \in V$ definieren wir $b_i := \sum_{e \in \delta(i)} x_e^* + D \sum_{e \in \delta(i)} \bar{x}_e$.

Die Kanten werden wie folgt gesetzt:

- Für jede ungerichtete Kante $e = ij$ aus E setzen wir in G^* zwei gerichtete Kanten (i, j) und (j, i) , die die Kosten $c'_{(i,j)} = c'_{(j,i)} = \frac{1}{2}(x_e^* + D\bar{x}_e)$ erhalten.
- Von der Quelle 0 aus setzen wir für jeden Knoten $i \in V$ eine gerichtete Kante $(0, i)$ mit den Kosten $c'_{(0,i)} = \max\{\frac{1}{2}b_i - (D + 1), 0\}$.
- Zur Senke $n + 1$ hin setzen wir von jedem Knoten $i \in V$ aus eine gerichtete Kante $(i, n + 1)$ mit den Kosten $c'_{(i,n+1)} = \max\{(D + 1) - \frac{1}{2}b_i, 0\}$.

Sei nun $S \subseteq V$. Wir betrachten in dem Netzwerk G^* einen $(0, n + 1)$ -Schnitt, d.h., einen gerichteten Schnitt von $S \cup \{0\}$ nach $(V \setminus S) \cup \{n + 1\}$. Dessen Kosten betragen:

$$\begin{aligned}
& c'(S \cup \{0\} : (V \setminus S) \cup \{n+1\}) \\
&= \sum_{i \in V \setminus S} \max\{\frac{1}{2}b_i - (D+1), 0\} + \sum_{i \in S} \max\{(D+1) - \frac{1}{2}b_i, 0\} \\
&\quad + c'(S : V \setminus S) \\
&= \sum_{i \in S} (\max\{(D+1) - \frac{1}{2}b_i, 0\} - \max\{\frac{1}{2}b_i - (D+1), 0\}) \\
&\quad + \frac{1}{2} \sum_{\substack{e=ij \\ i \in S, j \in V \setminus S}} (x_e^* + D\bar{x}_e) + \sum_{i \in V} \max\{\frac{1}{2}b_i - (D+1), 0\} \\
&= (D+1)|S| - \frac{1}{2} \sum_{i \in S} b_i + \frac{1}{2} \sum_{\substack{e=ij \\ i \in S, j \in V \setminus S}} (x_e^* + D\bar{x}_e) \\
&\quad + \sum_{i \in V} \max\{\frac{1}{2}b_i - (D+1), 0\} \\
&= (D+1)|S| - \sum_{e \in E(S)} (x_e^* + D\bar{x}_e) + \sum_{i \in V} \max\{\frac{1}{2}b_i - (D+1), 0\} \\
&= f(S) + D + \sum_{i \in V} \max\{\frac{1}{2}b_i - (D+1), 0\}
\end{aligned}$$

Da D und $\sum_{i \in V} \max\{\frac{1}{2}b_i - (D+1), 0\}$ von der Menge S unabhängig sind, korrespondiert eine die Funktion f minimierende Menge S zu einem minimalen $(0, n+1)$ -Schnitt in (G^*, c') . Diesen können wir mittels einer Maximaler-Fluss-Berechnung erhalten. Allerdings darf S nicht leer sein, weswegen wir nach der Idee von Padberg & Wolsey $n-2$ Maximaler-Fluss-Berechnungen benutzen. Die Kosten des k -ten Netzwerkes für $k = 1, \dots, n-2$ werden dabei wie in (G^*, c') definiert mit der Ausnahme, dass

- für $k = 1$ die Kosten $c'_{(0,1)} = \infty$ gesetzt werden,
- für $k \geq 2$ die Kosten $c'_{(0,k)} = \infty$ und für $v = 1, \dots, k-1$ die Kosten $c'_{(v,n+1)} = \infty$ gesetzt werden.

Die Schnittmenge S , die unter allen $n-2$ Berechnungen die geringsten Kosten hat, ist dann die gesuchte Menge, was durch folgende Überlegung bestätigt wird:

Falls S den $(0, n+1)$ -Schnitt minimiert und $|S| \geq 3$ gilt, dann gibt es ein $k \in \{1, \dots, n-2\}$, welches $k \in S$ und $\{1, \dots, k-1\} \subseteq V \setminus S$ erfüllt. Die Kosten jedes $(0, n+1)$ -Schnittes mit $|S| = 2$ sind $\geq D+1$. Wenn S nur einelementig ist, dann betragen die Schnittkosten $D+1$. Deswegen reichen die oben angegebenen $n-2$ Maximaler-Fluss-Berechnungen zum Finden der Menge S aus, die den $(0, n+1)$ -Schnitt minimiert.

Der von uns gemachte Ansatz zur primalen Separierung der Kreis-Eliminierungsungleichungen (4.1) hat somit die gleiche Laufzeit wie die von Padberg & Wolsey vorgeschlagene Standard-Separierung und ist ebenfalls exakt.

4.5 Konstruktion eines gradbeschränkten Spannbau- mes

Um die gewonnenen Erkenntnisse im Rahmen des primalen Schnittebenenalgorithmus aus Abschnitt 2.4 anwenden zu können, benötigt man noch ein Verfahren, um zu einem gegebenen Graphen $G = (V, E)$ mit Gradbeschränkungen $\sum_{e \in \delta(i)} x_e \leq b_i$ für alle Knoten $i \in V$ einen gradbeschränkten Spannbaum als Startlösung \bar{x} zu generieren. Wir gehen in diesem Zusammenhang davon aus, dass es solch eine zulässige Lösung gibt.

Bekannt ist Prim's Algorithmus zur Berechnung eines minimalen Spannbaumes bei gegebenen Kantenkosten $c(e)$:

Setze $T = (V_T, E_T) = (\{i\}, \emptyset)$, wobei $i \in V$ beliebig.

Solange $V_T \neq V$:

Füge diejenige Kante $e \in \delta(V_T)$ zu E_T hinzu, deren Kosten $c(e)$ minimal sind.

Setze $V_T = V(E_T)$.

Die Gesamtkosten des so gefundenen minimalen Spannbaumes können bei der Suche nach einem gradbeschränkten minimalen Spannbaum immer als untere Schranke benutzt werden.

Narula & Ho [30] veränderten Prim's Algorithmus geringfügig, um einen gradbeschränkten Spannbaum zu erhalten:

Setze $T = (V_T, E_T) = (\{i\}, \emptyset)$, wobei $i \in V$ beliebig.

Solange $V_T \neq V$:

Füge diejenige Kante $e \in \delta(V_T)$ zu E_T hinzu, deren Kosten $c(e)$ minimal sind und deren Hinzunahme keine der Gradbeschränkungen verletzt.

Setze $V_T = V(E_T)$.

Die Kosten des auf diese Weise erhaltenen gradbeschränkten Spannbaumes müssen nicht minimal sein und sind von der Wahl des Startknotens i abhängig. Deswegen schlugen Narula & Ho ein Verfahren vor, mit dessen Hilfe die Kosten des gradbeschränkten Spannbaumes durch Austausch gewisser Kanten verringert werden können. Die dadurch erhaltene Lösung bleibt allerdings weiterhin heuristisch.

Eine andere Art Heuristik, die von ihnen präsentiert wurde, basiert auf einem minimalen Spannbaum. Bei diesem werden nachträglich die Kanten ausgetauscht, die eine der Gradbeschränkungen verletzen. Auch Savelsbergh & Volgenant [39] stellten zwei Heuristiken vor, die mittels dieses Ansatzes arbeiten. Ihre beiden Ansätze haben eine Laufzeit von $O(n)$ und erzielten bessere Resultate als die von Narula & Ho.

Kapitel 5

Resultate

Dieses Kapitel fasst alle Resultate zusammen, die im Rahmen dieser Diplomarbeit erzielt wurden. In Abschnitt 5.1 werden die theoretischen Resultate zur primalen Separierung für Max-Cut und gradbeschränkte minimale Spannbäume besprochen. Die mit Hilfe unserer Implementierung gewonnenen Rechenresultate für das gradbeschränkte minimale Spannbaum Problem werden in Abschnitt 5.2 mit in der Literatur veröffentlichten Ergebnissen verglichen.

5.1 Theoretische Ergebnisse

Eine Motivation zur Untersuchung der primalen Separierung für das Max-Cut Problem und gradbeschränkte minimale Spannbäume war die Tatsache, dass Letchford & Lodi [27] für einige Probleme eine primale Separierung fanden, die schneller oder konzeptuell einfacher als die beste bekannte Standard-Separierung ist. Da die Separierung eine entscheidende Rolle bei Schnittebenenverfahren oder Branch & Cut spielt, legt dies die Vermutung nahe, dass eine schnellere Separierung auch zu einem schnelleren Verfahren zur Lösung eines Problems führen kann.

In Kapitel 3 wurde die primale Separierung für das Max-Cut Problem und in Kapitel 4 für gradbeschränkte minimale Spannbäume untersucht. Einen Vergleich der Laufzeiten der bisher bekannten Standard-Separierungen mit den von uns gewonnenen primalen Separierungen für die einzelnen Ungleichungsklassen zeigt Tabelle 5.1. Dabei handelt es sich bei allen mit einer Laufzeit aufgeführten Separierungen um exakte Separierungen. Auf den ersten Blick erkennt man, dass die primale Separierung in allen Fällen mindestens genauso schnell wie die Standard-Separierung ist. Wir betrachten nun beide Probleme im Einzelnen.

Für das Max-Cut Problem hat sich bei den von uns durchgeführten Untersuchungen gezeigt, dass die Laufzeiten der primalen und der Standard-Separierungen für die Dreiecks- und Kreisungleichungen durch das gleiche Polynom beschrieben werden. Trotz der Verlagerung der Probleme vom Schnitt-Polytop auf den Schnitt-Konus mit Hilfe der Umschalt-Operation, die selbst eine lineare Laufzeit hat, konnte keine schnellere Laufzeit erzielt werden. Dies lässt sich folgendermaßen begründen:

Problem	Ungleichungsklasse	Standard-Separierung	primale Separierung
Max-Cut	Dreiecksungl.	$O(n^3)$	$O(n^3)$
	Kreisungl.	$O(n^3)$	$O(n^3)$
	Clique-Ungl.	Heuristik	Heuristik
	Hypermetrische Ungl.	Heuristik	Heuristik
GBMSB	Spannbaumgl.	$O(m)$	$O(m)$
	Knotengradungl.	$O(m)$	$O(m)$
	Zusammenhangsungl.	$O(mn + n^2 \log n)$	$O(mn + n^2)$
	Spez. Kreiseliminierungsungl.	$O(mn^2)$	$O(mn)$
	Kreiseliminierungsungl.	$O(n^4)$	$O(n^4)$

Tabelle 5.1: Vergleich der Laufzeiten der primalen und der Standard-Separierungen für die jeweiligen Ungleichungsklassen

Für einen vollständigen Graphen können polynomiell viele Dreiecksungleichungen generiert werden. Betrachtet man nur die Subklasse derjenigen Dreiecksungleichungen, die von einem gegebenen Schnitt mit Gleichheit erfüllt werden, so reduziert das die Anzahl der zu generierenden Ungleichungen lediglich um einen konstanten Faktor. Somit sind die Ordnungen der Polynome, die die Anzahl der Ungleichungen der gesamten Klasse bzw. der eben beschriebenen Subklasse angeben, gleich.

Bei den Kreisungleichungen separiert man nach Umschalten auf die gleiche Art wie schon bei der Standard-Separierung, d.h., man sucht in einem Hilfsgraphen nach einem kürzesten Weg. Deswegen ist auch hier die primale Separierung genauso schnell wie die Standard-Separierung. Für die Clique-Ungleichungen und die hypermetrischen Ungleichungen sind die Standard-Separierungen und damit auch die primalen Separierungen \mathcal{NP} -schwierig. Aus diesem Grund haben wir uns bei diesen beiden Ungleichungsklassen auf die Entwicklung von Heuristiken zur primalen Separierung beschränkt. Insgesamt gesehen konnte für keine der betrachteten Ungleichungsklassen des Max-Cut Problems eine primale Separierung erreicht werden, die schneller als die jeweils bekannte Standard-Separierung ist.

Dies ist bei den gradbeschränkten minimalen Spann bäumen anders. Hier haben wir für zwei Ungleichungsklassen primale Separierungen hergeleitet, die schneller sind als die bekannten Standard-Separierungen. Bei den trivialen Separierungen für die Spannbaum-Gleichung und den Ungleichungen bzgl. des Knotengrads kann keine Verbesserung erzielt werden, da die Standard-Separierungen in linearer Zeit arbeiten.

Für die Zusammenhangs- und die speziellen Kreiseliminierungsgleichungen haben wir eine primale Separierung angegeben, deren Laufzeit schneller als die der Standard-Separierung ist. Der Grund dafür ist, dass der gegebene Punkt \bar{x} , der die zu separierenden Ungleichungen mit Gleichheit erfüllen soll, einen Spannbaum im Graphen darstellt. Diese Struktur kann genutzt werden, um die primale Separierung konzeptuell zu vereinfachen. So wird bei den Zusammenhangsgleichungen statt eines Minimum-Schnitt-Algorithmus, der bei der Standard-Separierung verwendet wird, nur die Tiefensuche

benutzt. Auch bei den speziellen Kreiseliminierungsungleichungen wird statt eines Kürzeste-Wege-Algorithmus nur die Tiefensuche bei der primalen Separierung verwendet. Daraus resultieren die Laufzeitverbesserungen. Bei den allgemeinen Kreiseliminierungsungleichungen wird das gleiche Verfahren wie zur Standard-Separierung benutzt, weswegen hier die Laufzeiten gleich sind.

5.2 Rechenresultate

Es sind viele Lösungsansätze bekannt, mit deren Hilfe man einen gradbeschränkten minimalen Spannbaum finden kann. Knowles & Corne [24] sowie Krishnamoorthy, Ernst & Sharaiha [25] vergleichen in ihren Veröffentlichungen einige Heuristiken, Simulated Annealing, genetische Ansätze, Lagrange Relaxierungen und Branch & Bound Verfahren anhand von verschiedenen Instanzklassen. Raidl hat seines Wissens nach [38] als Erster ein Standard Branch & Cut Verfahren zur exakten Lösung des gradbeschränkten minimalen Spannbaum Problems implementiert. Auch er benutzte dazu **ABACUS**. Wir werden im Folgenden Rechenresultate präsentieren, die wir mit unserem primalen Schnittebenenalgorithmus bzgl. oben genannter Instanzklassen erzielen konnten, und diese mit den besten bekannten Resultaten vergleichen. Unseres Wissens nach existiert in der Literatur bis zum jetzigen Zeitpunkt kein Lösungsversuch dieser Art.

Bei den von uns durchgeführten Berechnungen traten des Öfteren Probleme auf, die einen vorzeitigen Abbruch des Algorithmus zur Folge hatten. In einigen wenigen Fällen bereitete das zur Lösung der Tableaus benutzte Programm CPLEX Probleme. Es lieferte einen fehlerhaften Status einer Basisvariablen zurück oder benutzte Phase II des primalen Simplex, obwohl das Tableau primal zulässig war. Diese Effekte konnten auch von anderen Personen [15] bereits beobachtet werden.

Alle anderen Abbrüche wurden durch Rundungsfehler verursacht. Diese machten sich besonders bei der Generierung der gemischtganzzahligen Gomory-Schnitte bemerkbar. In vielen Fällen lag die beste bekannte Lösung \bar{x} nicht auf der durch den Gomory-Schnitt beschriebenen Schnittebene, sondern hatte von dieser einen Abstand ε in der Größenordnung $10^{-14} < \varepsilon < 10^{-7}$. Nach dem Hinzufügen solcher Schnittebenen wurde in einer anschließenden Iteration durch den primalen Simplex ein neuer Lösungsvektor x^* berechnet, der zu \bar{x} einen betragslichen Abstand von nur etwa ε hat und damit als neuer zulässiger Lösungsvektor galt. Dadurch begann der Algorithmus zu zyklern und konnte infolgedessen keinen besseren Lösungsvektor finden. Wir haben durch mehrere Ansätze versucht, die Abbrüche durch Rundungsfehler zu beseitigen. Zum einen haben wir die Generierung der Gomory-Schnitte dahingehend modifiziert, dass sie in einigen Fällen Schnittebenen liefert, die von \bar{x} mit Gleichheit erfüllt werden. Des Weiteren wird an einigen Stellen in unserem Algorithmus die Toleranzgrenze für die Maschinengenauigkeit dynamisch angepasst, um Rundungsfehlern entgegenzuwirken. Alle Berechnungen wurden auf einem Athlon MP 1800+ mit 1533MHz durchgeführt. Um die durch die Hardware verursachten Rundungsfehler zu überprüfen, haben wir unsere Berechnungen auf Systeme mit Pentium III 500Mhz, UltraSPARC-IIi 440MHz

und UltraSPARC-II 300MHz Prozessoren verlagert. Hierbei konnten wir bei einigen der problematischen Instanzen Veränderungen in der Anzahl der Iterationen, nach denen abgebrochen wurde, beobachten. Zuletzt wurde auf allen Maschinen die durch den Compiler vorgegebenen Optimierungsstufen variiert oder ausgeschaltet. Auch dies führte dazu, dass sich die Anzahl der Iterationen bis zum Abbruch bei einigen problematischen Instanzen veränderte. Trotz intensiver Nachforschungen konnten im Rahmen dieser Diplomarbeit nicht alle Fehler beseitigt werden.

Bei allen Berechnungen haben wir ein Iterationslimit von 10000 Iterationen gesetzt, da Beobachtungen gezeigt haben, dass ab etwa 5000 Iterationen keine signifikanten Verbesserungen mehr erzielt werden. Die Güte einer Lösung geben wir wie üblich durch den dazugehörigen *Gap*-Wert an, der sich berechnen lässt durch $\text{Gap} = (\text{primale Schranke} - \text{duale Schranke}) / \text{duale Schranke}$ und in Prozent angegeben wird.

Von den insgesamt 240 Instanzen, die in den 6 Klassen R, M, CRD, SYM, STR und SHRD gruppiert sind, konnten 52 von der Startheuristik und weitere 38 vom Schnittebenenalgorithmus optimal gelöst werden. Insgesamt wurden also 90 Instanzen optimal gelöst, was einem Anteil von 37,5% entspricht. Für 52 Instanzen (21,6%) wurden Lösungen berechnet, deren *Gap*-Wert zwischen 0 und 1% liegt. Weitere 20 Instanzen (8,3%) konnten mit einem *Gap*-Wert zwischen 1 und 5% gelöst werden. Insgesamt wurden also für 67,5% der Instanzen gute bis optimale Lösungen berechnet. Von den 78 Instanzen, für die keine guten Resultate erzielt werden konnten, brachen 75,64% der Berechnungen wegen oben genannten „Fehlern“ ab.

Bei den folgenden Tabellen gibt n immer die Anzahl der Knoten der jeweiligen Instanz an. Alle Instanzen haben eine globale Gradbeschränkung b , die wir übernommen haben. Wir wollen hier anmerken, dass unser Algorithmus auch Instanzen berechnen kann, bei denen jeder Knoten eine individuelle Gradbeschränkung hat. Die Spalte *CPU* gibt die nach oben gerundete CPU Zeit in Sekunden an. Unser primales Schnittebenenverfahren bezeichnen wir mit *Primal*. In der Spalte *Iter.* befindet sich die Anzahl der primalen Simplex-Iterationen, die der Algorithmus durchgeführt hat. Trat ein frühzeitiger Abbruch durch einen der oben erwähnten „Fehler“ auf, so ist die Anzahl der bis zum Abbruch durchgeführten Iterationen kursiv dargestellt.

Die R und M Instanzen stammen von Knowles & Corne [24]. Die von ihnen entwickelte *GA* Heuristik erzielt in den Vergleichen ihrer Veröffentlichung die besten Resultate für diese Instanzen. Sie geben die Güte mithilfe eines Quotienten Q . an, den sie als $Q. = \text{Kosten des gefundenen gradbeschränkten Spannbaums} / \text{Kosten des minimalen Spannbaums}$ definieren. Die Werte der von ihnen angegebenen Quotienten wurden als Mittelwert über 20 Durchläufe gebildet. Über die Laufzeit der *GA* Heuristik machen sie keine Angaben. Um ihre Resultate mit unseren vergleichen zu können, haben wir für unsere Lösungen ebenfalls diese Quotienten gebildet.

Tabelle 5.2 zeigt die für die Klasse der R-Instanzen erzielten Resultate. Obwohl unser Verfahren bei der Instanz r100n1 aufgrund eines „Fehlers“ vorzeitig abgebrochen ist, konnte auch hier eine sehr gute Lösung berechnet werden. Die 3 Instanzen mit 50 Knoten konnten alle optimal gelöst werden. Ein

n	b	Typ	Primal				GA
			Gap	CPU	Iter.	Q.	Q.
50	5	n1	0	1	45	1,61	1,74
50	5	n2	0	19	1528	1,69	1,83
50	5	n3	0	10	1111	1,62	1,78
100	5	n1	0,1652	38	7861	1,61	1,73
100	5	n2	0,3353	3145	10000	1,50	1,60
100	5	n3	0,0149	14	10000	1,55	1,69
200	5	n1	2,4273	140	10000	1,54	1,64
200	5	n2	0,9792	141	10000	1,49	1,59
200	5	n3	0,4945	148	10000	1,53	1,65

Tabelle 5.2: Vergleich der R-Instanzen

n	b	Typ	Primal				B & C			GA
			Gap	CPU	Iter.	Q.	Gap	CPU	Q.	Q.
50	5	n1	64,47	1	10000	3,65	0	320	2,45	3,15
50	5	n2	47,51	23	10000	3,24	0	1	2,21	2,99
50	5	n3	35,80	1	10000	3,20	0	1	2,36	2,58
100	5	n1	67,13	2	10000	3,29	0	3	1,98	2,44
100	5	n2	86,90	2	1886	3,86	0	1555	2,08	2,84
100	5	n3	90,62	2	10000	3,75	0	4	1,98	2,86
200	5	n1	48,75	121	3872	2,66	0	3	1,79	2,04
200	5	n2	78,10	125	10000	3,19	0	24	1,78	2,47
200	5	n3	0	4868	7416	1,68	0	22	1,68	1,71

Tabelle 5.3: Vergleich der M-Instanzen

Vergleich der Quotienten zeigt, dass unser primales Schnittebenenverfahren für alle Instanzen bessere Resultate erzielt als die GA Heuristik.

Die M-Instanzen wurden ebenfalls von Raidl [38] berechnet. Sein Standard Branch & Cut Algorithmus wird in der Tabelle 5.3 als *B & C* bezeichnet. Bei 2 Instanzen traten bei unseren Berechnungen „Fehler“ auf. Die von uns erzielten Ergebnisse sind bis auf die Ausnahme der Instanz m200n3, die optimal gelöst werden konnte, deutlich schlechter als diejenigen der GA Heuristik. Unser Algorithmus hatte mit dieser Instanzklasse die größten Probleme hinsichtlich der Qualität der Lösungen. Dies liegt an den von Knowles & Corne konstruierten M-Graphen. Anhand deren Struktur kann man erkennen, dass sie Heuristiken, die nach dem Greedy-Prinzip arbeiten und sich an den Kantenkosten orientieren wie z.B. die von uns eingesetzte Startheuristik von Narula & Ho, zu tendenziell ungünstigen Entscheidungen zwingen. Zusätzlich ist der Knotengrad des minimalen Spannbaums bei jeder Instanz sehr hoch, nämlich teilweise bis zu 15. Dadurch haben der minimale Spannbaum und der gesuchte gradbeschränkte minimale Spannbaum sehr unterschiedliche Strukturen.

Die Resultate zu den M-Instanzen bestätigen, dass die Wahl der Startheuristik ein entscheidender Punkt für die Geschwindigkeit und Güte von primalen Schnittebenenverfahren ist. Je besser die von der Heuristik gelieferte Startlösung ist, desto weniger Ecken des zugrunde liegenden Polytops muss

n	b	Primal				B & B	
		Gap	CPU	Fehler	Optimal	Gap	CPU
30	3	0,113	1	0	8	0,70	1
50	3	0,018	1	1	8	0,01	1
70	3	0,012	1	1	7	0,01	1
100	3	0,048	1,3	0	4	0,04	1

Tabelle 5.4: Vergleich der CRD-Instanzen

n	b	Primal				B & B	
		Gap	CPU	Fehler	Optimal	Gap	CPU
30	3	0,499	1,1	0	9	4,26	1
30	4	0,046	1	0	9	0,16	1
50	3	0,676	1,3	0	5	4,78	1
50	4	0,441	1	2	7	0,95	1
50	5	0,039	1	0	9	0,10	1
70	3	1,360	153,9	3	0	5,60	5,7
70	4	0,458	1,7	2	2	1,03	1
70	5	0,109	1	1	6	0,11	1

Tabelle 5.5: Vergleich der SYM-Instanzen

der Algorithmus zur Berechnung der optimalen Lösung besuchen. Startet man in einer Ecke, die sehr weit von der optimalen Ecke entfernt ist, so kann unser Algorithmus kaum Verbesserungen machen, was teilweise auch in Rundungsfehlern begründet liegt.

Raidl konnte alle M-Instanzen optimal lösen. Die von ihm erzielten Quotienten sind damit minimal. An ihnen kann man ebenfalls erkennen, dass sich der minimale Spannbaum vom jeweiligen gradbeschränkten minimalen Spannbaum stark unterscheidet. Raidl setzt zusätzlich zu einem Schnittebenenverfahren auch Branchen ein. Insbesondere bei Instanz m200n3 zeigt der Vergleich der Laufzeit seines Verfahrens mit der Laufzeit unseres Verfahrens, das Standard Branch & Cut deutlich schneller ist.

Die nun folgenden 4 Klassen CRD, SYM, STR und SHRD stammen aus der Veröffentlichung von Krishnamoorthy, Ernst & Sharaiha [25]. Wir vergleichen unsere Resultate mit dem Branch & Bound Ansatz von Krishnamoorthy, Ernst & Sharaiha, der in unseren Tabellen mit *B & B* bezeichnet wird.

Die in Tabelle 5.4 aufgeführten Gap-Werte und CPU Zeiten sind Mittelwerte über jeweils 10 unterschiedliche CRD-Instanzen mit derselben Knotenanzahl. Die Spalte *Fehler* gibt an, wie viele der 10 Berechnungen durch einen „Fehler“ vorzeitig abgebrochen wurden. Die Anzahl der Instanzen, die optimal gelöst wurden, ist in der Spalte *Optimal* ersichtlich. Die CRD-Instanzen sind zweidimensionale euklidische Graphen, die zufällig erzeugt wurden. Unser Algorithmus kann hier genauso gute Resultate aufweisen wie das Branch & Bound Verfahren. Für die Instanzen mit 30 Knoten sind unsere Ergebnisse sogar besser.

Bei den SYM-Instanzen handelt es sich um zufällig erzeugte mehrdimensionale euklidische Graphen. Auch hier sind die in Tabelle 5.5 angegebenen

n	b	Typ	Primal			B & B	
			Gap	CPU	Iter.	Gap	CPU
30	4	D3	8,79	1	<i>1646</i>	8,11	1
30	4	D4	13,13	1	10000	11,68	1
30	4	D5	0,10	1	10000	14,27	1
30	4	D6	0,15	1	10000	16,05	60
30	4	D7	0,14	1	10000	17,13	600
30	5	D3	4,44	1	10000	4,18	1
30	5	D4	9,68	1	10000	8,86	1
30	5	D5	0	1	240	11,67	1
30	5	D6	0,12	9	10000	13,95	47
30	5	D7	0,12	1	10000	15,34	600
50	4	D3	8,12	1	10000	7,62	1
50	4	D4	11,51	1	<i>1199</i>	10,47	1
50	4	D5	14,77	1	<i>3492</i>	12,89	2
50	4	D6	17,23	1	<i>1311</i>	14,86	125
50	4	D7	18,61	1	<i>2391</i>	15,72	600
50	5	D3	4,04	1	<i>35</i>	3,94	1
50	5	D4	8,43	1	10000	7,96	1
50	5	D5	11,63	1	10000	10,50	2
50	5	D6	14,55	1	<i>4191</i>	12,91	98
50	5	D7	16,50	46	10000	14,07	600
70	4	D3	7,31	1	10000	6,90	1
70	4	D4	10,95	1	<i>859</i>	10,00	1
70	4	D5	13,47	1	<i>1583</i>	12,04	4
70	4	D6	15,69	1	<i>3189</i>	13,57	221
70	4	D7	17,38	1	<i>3749</i>	14,92	600
70	5	D3	3,64	1	<i>1496</i>	3,56	1
70	5	D4	8,06	1	<i>1041</i>	7,61	1
70	5	D5	10,67	1	<i>1579</i>	9,78	3
70	5	D6	13,27	1	10000	11,75	173
70	5	D7	15,26	1	<i>4546</i>	13,35	600
100	3	D3	10,49	2	<i>153</i>	9,38	1
100	3	D4	13,49	1	<i>969</i>	11,78	1
100	3	D5	15,56	1	<i>1109</i>	13,42	1
100	3	D6	16,93	2	<i>490</i>	14,44	1
100	3	D7	18,00	2	10000	15,21	3
100	4	D3	6,69	1	<i>434</i>	6,27	1
100	4	D4	9,93	1	<i>598</i>	9,03	1
100	4	D5	12,78	1	<i>5348</i>	11,33	5
100	4	D6	14,58	2	<i>1418</i>	12,68	325
100	4	D7	15,69	2	10000	13,62	600
100	5	D3	3,33	2	<i>2440</i>	3,23	1
100	5	D4	7,41	2	<i>610</i>	6,90	1
100	5	D5	10,18	2	<i>1584</i>	9,24	4
100	5	D6	12,28	2	<i>1874</i>	10,94	255
100	5	D7	13,75	2	<i>3288</i>	12,10	600

Tabelle 5.6: Vergleich der STR-Instanzen

n	b	Primal			B & C		PSS	
		Gap	CPU	Iter.	Gap	CPU	Gap	CPU
15	3	0	1	103	0	1	1,72	2
15	4	0	1	25	0	1	2,08	2
15	5	0	1	13	0	1	0	3
20	3	0	1	127	0	1	0,45	6
20	4	0	1	118	0	1	0	4
20	5	0	1	62	0	1	0,47	5
25	3	0	1	73	0	1	0	13
25	4	0	1	68	0	1	0	13
25	5	0	1	46	0	1	1,07	10
30	3	0	1	304	0	1	0	19
30	4	0	1	294	0	1	0	20
30	5	0	1	227	0	1	0	19

Tabelle 5.7: Vergleich der SHRD-Instanzen

Gap-Werte und CPU Zeiten Mittelwerte über 10 unterschiedliche Instanzen mit derselben Knotenanzahl. Unser primales Schnittebenenverfahren erzielt für die SYM-Instanzen in fast allen Fällen deutlich bessere Resultate als das Branch & Bound Verfahren.

Die Struktur der Graphen der STR-Klasse ist durch Knoten-Cluster geprägt, die alle dieselbe Anzahl von Knoten enthalten. In Tabelle 5.6 sind die Resultate für jede STR-Instanz einzeln aufgeführt. Bei über der Hälfte der Berechnungen traten „Fehler“ auf, die meistens Rundungsfehler waren. Für über die Hälfte der Instanzen mit 30 Knoten konnten sehr gute Lösungen berechnet werden. Dies trifft auf die übrigen Instanzen nicht zu. Auch der Branch & Bound Ansatz, der nach maximal 600 Sekunden Laufzeit abgebrochen wurde, kann bei den STR-Instanzen keine guten Resultate erzielen. Im Mittel ist unser Verfahren hier geringfügig schlechter als Branch & Bound.

Die Instanzklasse SHRD wird von Krishnamoorthy, Ernst & Sharaiha als die schwerste bezeichnet, da keine dieser Instanzen von ihrem Branch & Bound Ansatz innerhalb von 600 Sekunden gelöst werden konnte. Deswegen verglichen wird in Tabelle 5.7 ihre beste Heuristik *PSS* mit unseren Ergebnissen. Ebenso wie Raidl [38] hat unser Verfahren keine Probleme mit diesen Instanzen, denn alle konnten innerhalb kürzester Zeit optimal gelöst werden.

Zusammenfassend lässt sich sagen, dass für 2 der 6 Instanzklassen, nämlich für die M- und STR-Instanzen, die Resultate unseres primalen Schnittebenenverfahrens schlechter sind als die Resultate derjenigen Verfahren, die uns zum Vergleich zur Verfügung standen. Bei den SHRD-Instanzen sind die Qualität der Lösungen und die Geschwindigkeit identisch mit denen des Standard Branch & Cut Verfahrens. Wir vermuten, dass sich bei dieser Instanzklasse Unterschiede erst ab Instanzen mit einer größeren Knotenanzahl ergeben. Für die beiden Instanzklassen CRD und SYM konnten wir bessere Resultate als der Branch & Bound Ansatz erzielen. Unsere Resultate für die R-Instanzen sind sogar besser als die besten bisher in der Literatur veröffentlichten Resultate.

Kapitel 6

Zusammenfassung & Ausblick

In dieser Diplomarbeit haben wir die primale Separierung für das Max-Cut und das gradbeschränkte minimale Spannbaum Problem untersucht. Für das Max-Cut Problem konnte für keine Ungleichungsklasse eine primale Separierung gefunden werden, die schneller als die beste bekannte Standard-Separierung ist. Im Gegensatz dazu ist uns dies für zwei Ungleichungsklassen des gradbeschränkten minimalen Spannbaum Problems gelungen. Wir konnten dadurch mit dieser Arbeit ein Beispiel dafür liefern, dass die primale Separierung in gewissen Fällen konzeptuell einfacher oder schneller als die Standard-Separierung ist. Dies wurde zuvor auch von anderen Autoren [27, 14] berichtet.

Um die von uns entwickelte primale Separierung für das gradbeschränkte minimale Spannbaum Problem anwenden zu können, haben wir einen primalen Schnittebenenalgorithmus für dieses Problem implementiert. Wir haben anhand von Instanzklassen die mit unserem Algorithmus erzielten Rechenergebnisse mit denen anderer Verfahren verglichen. Für einige Instanzklassen hat sich dabei gezeigt, dass die Qualität der Lösungen unseres Algorithmus mit denen der besten bisher bekannten Verfahren vergleichbar und teilweise sogar besser ist.

Zum direkten Vergleich der Mächtigkeit der primalen Separierung mit der der Standard-Separierung wären sicherlich die Rechenresultate eines Standard-Schnittebenenverfahrens geeignet gewesen. Diese lagen uns jedoch nicht vor. Wir konnten für zwei Instanzklassen unser primales Schnittebenenverfahren mit einem Standard Branch & Cut Verfahren vergleichen. Dieses war für die M-Instanzen unserem Algorithmus bzgl. der Geschwindigkeit und der Qualität der Lösungen deutlich überlegen.

Vermutlich kann unser primales Schnittebenenverfahren durch den Einsatz von Branchen von der Geschwindigkeit, zumindest aber von der Güte der Lösungen her weiter verbessert werden. Zudem können Rundungsfehler, die durch die Berechnung der gemischtganzzahligen Gomory-Schnitte entstehen, vermieden werden, indem man statt des Hinzufügens dieser Schnittebenen brancht, d.h., man erzeugt zwei oder mehrere Unterprobleme. Die Branching-Regeln, wie sie bei Standard Branch & Cut Verfahren eingesetzt werden, können für ein primales Branch & Cut Verfahren nicht genutzt werden, da sie i.A. nicht sicherstellen, dass die zurzeit beste bekannte Lösung \bar{x} in den Unterproblemen weiterhin enthalten ist. In der Literatur sind allerdings keine primalen

Branching-Regeln bekannt. Letchford & Lodi [26] haben eine Idee für eine primale Branching-Regel angegeben:

Angenommen, es gibt ein Paar Variablen x_i, x_j , so dass für die momentan beste bekannte Lösung \bar{x} gilt $\bar{x}_i = \bar{x}_j = 0$ und die aktuelle Lösung x^* erfüllt $0 < x_i^* < x_j^* \leq 1$. Dann erzeugt man aus dem jetzigen Problem zwei Unterprobleme, indem bei dem einen die Variable $x_i = 0$ gesetzt und bei dem anderen die Ungleichung $x_i \geq x_j$ hinzugefügt wird. In beiden Unterproblemen bleibt damit die Lösung \bar{x} erhalten, während zugleich x^* in beiden unzulässig wird.

Aufgrund der gemachten Voraussetzungen ist diese Regel nur in speziellen Situationen einsetzbar. Die Entwicklung allgemeiner primaler Branching Regeln ist ein interessanter Bereich, der für weitere Untersuchungen offen bleibt.

Abbildungsverzeichnis

2.1	Standard-Schnittebene	11
2.2	Primale Schnittebene	13
2.3	Beweiskette zur polynomiellen Äquivalenz	15
3.1	Beispiel-Instanz des Max-Cut Problems	32
3.2	Umschalt-Operation	34
3.3	Dreiecksungleichung	38
3.4	Kreisungleichung	39
3.5	Clique-Ungleichung	42
4.1	Gradbeschränkter Spannbaum	54
4.2	Zusammenhangsungleichung	58
4.3	Spezielle Kreis-Eliminierungsungleichung	59

Tabellenverzeichnis

5.1	Laufzeitvergleich der Separierungen	66
5.2	Vergleich der R-Instanzen	69
5.3	Vergleich der M-Instanzen	69
5.4	Vergleich der CRD-Instanzen	70
5.5	Vergleich der SYM-Instanzen	70
5.6	Vergleich der STR-Instanzen	71
5.7	Vergleich der SHRD-Instanzen	72

Literaturverzeichnis

- [1] L. R. Arnold, M. Bellmore: Iteration skipping in primal integer programming. *Operations Research* 22, 129-136, 1974.
- [2] L. R. Arnold, M. Bellmore: A generated cut for primal integer programming. *Operations Research* 22, 137-143, 1974.
- [3] L. R. Arnold, M. Bellmore: A bounding minimization problem for primal integer programming. *Operations Research* 22, 383-392, 1974.
- [4] F. Barahona, A. R. Mahjoub: On the cut polytope. *Mathematical Programming* 36, 157-173, 1986.
- [5] F. Barahona, H. Titan: Max Mean Cuts and Max Cuts. In: *Workshop on Applications of Combinatorial Optimization in Science and Technology COST*, DIMACS Technical Report 18, 30-45, 1991.
- [6] S. Chopra: On the spanning tree polyhedron. *Operation Research Letter* 8, 25-29, 1989.
- [7] W. J. Cook, W. H. Cunningham, W. R. Pulleyblank, A. Schrijver: *Combinatorial Optimization*. New York: Wiley-Interscience, 1998.
- [8] C. De Simone: The cut polytope and the boolean quadric polytope. *Discrete Mathematics* 79, 71-75, 1989.
- [9] C. De Simone, G. Rinaldi: A cutting plane algorithm for the max-cut problem. *Optimization Methods and Software* 3, 195-214, 1994.
- [10] M. M. Deza, M. Laurent: *Geometry of Cuts and Metrics*. Berlin: Springer, 1997.
- [11] J. Edmonds: Matroids and the greedy algorithm. *Mathematical programming* 1, 127-136, 1971.
- [12] J. Edmonds: Submodular functions, matroids and certain polyhedra. In: R. Guy, H. Hanani, N. Sauer, J. Schönheim (Hrsg.) *Combinatorial Structures and their Applications*, New York: Gordon and Breach, 69-87, 1970.
- [13] F. Eisenbrand, G. Rinaldi, P. Ventura: 0/1 primal separation and 0/1 optimization are equivalent. *Arbeitspapier*, Istituto di Analisi dei Sistemi ed Informatica, Rom, 2001.

-
- [14] F. Eisenbrand, G. Rinaldi, P. Ventura: Primal separation for 0/1 polytopes. *Technischer Bericht R. 546*, Istituto di Analisi dei Sistemi ed Informatica, Rom, 2001.
- [15] F. Eisenbrand: private Kommunikation. 2003.
- [16] M. R. Garey, D. S. Johnson: *Computers and Intractability, A Guide to the Theory of NP-Completeness*. San Francisco: Freeman, 1979.
- [17] B. Gavish: Topological Design of Centralized Computer Networks—Formulations and Algorithms. *Networks* 12, 355-377, 1982.
- [18] F. Glover: A new foundation for a simplified primal integer programming algorithm. *Operations Research* 16, 727-740, 1968.
- [19] M. Grötschel, L. Lovász: Combinatorial optimization. In: R. Graham, M. Grötschel, L. Lovász (Hrsg.) *Handbook of Combinatorics*, Vol. 2, North Holland, 1541-1597, 1995.
- [20] M. Grötschel, L. Lovász, A. Schrijver: The ellipsoid method and its consequences in combinatorial optimization. *Combinatorica* 1, 169-197, 1981.
- [21] ILOG: *CPLEX 7.1, User's Manual*. Gentilly: ILOG S.A., 2001.
- [22] M. Jünger, G. Reinelt, S. Thienel: Practical problem solving with cutting plane algorithms in combinatorial optimization. In: W. Cook, L. Lovász, P. Seymour (Hrsg.) *Combinatorial Optimization*, DIMACS Series in Discrete Mathematics and Theoretical Computer Science Vol. 20, American Mathematical Society, 111-152, 1995.
- [23] R. M. Karp, C. H. Papadimitriou: On linear characterizations of combinatorial optimization problems. *21st Annual Symposium on Foundations of Computer Science, Syracuse N.Y. 1980*, IEEE, New York, 1-9, 1980.
- [24] J. D. Knowles, D. W. Corne: A New Evolutionary Approach to the Degree-Constrained Minimum Spanning Tree Problem. *IEEE Transactions on Evolutionary Computation* 4 (2), 125-134, 2000.
- [25] M. Krishnamoorthy, A. T. Ernst, Y. M. Sharaiha: Comparison of Algorithms for the Degree Constrained Minimum Spanning Tree. *Journal of Heuristics* 7, 587-611, 2001.
- [26] A. N. Letchford, A. Lodi: Primal cutting plane algorithms revisited. *Mathematical Methods of Operations Research* 56 (1), 67-81, 2002.
- [27] A. N. Letchford, A. Lodi: Primal separation algorithms. *Technischer Bericht OR-01-5*, DEIS, Universität zu Bologna. Eingereicht, 2001.
- [28] A. N. Letchford: private Kommunikation. 2002.
- [29] K. Mehlhorn, S. Näher: *LEDA: A platform for combinatorial and geometric computing*. Cambridge: Cambridge University Press, 1999.

-
- [30] S. C. Narula, C. A. Ho: Degree-constrained minimum spanning tree. *Computers & Operations Research* 7, 239-249, 1980.
- [31] G. L. Nemhauser, L. A. Wolsey: *Integer and Combinatorial Optimization*. New York: Wiley, 1988.
- [32] OREAS: **ABACUS**, *A Branch-And-CUt System Version 2.3, User's Guide and Reference Manual*. Köln: OREAS GmbH, 1999.
- [33] M. W. Padberg, M. Grötschel: Polyhedral computations. In: E. L. Lawler, J. K. Lenstra, A. H. G. Rinnooy Kan, D. B. Shmoys (Hrsg.) *The Traveling Salesman Problem: A Guided Tour of Combinatorial Optimization*, Chichester: John Wiley & Sons, 307-360, 1985.
- [34] M. W. Padberg, S. Hong: On the symmetric travelling salesman problem: a computational study. *Mathematical Programming Study* 12, 78-107, 1980.
- [35] M. W. Padberg, M. R. Rao: The russian method for linear programming III: Bounded integer programming. *Technischer Bericht 81-39*, Graduate School of Business and Administration, New York University, 1981.
- [36] M. W. Padberg, L. A. Wolsey: Trees and cuts. *Annals of Discrete Mathematics* 17, 511-517, 1983.
- [37] S. Poljak, Z. Tuza: Maximum cuts and large bipartite subgraphs. In: W. Cook, L. Lovász, P. Seymour (Hrsg.) *Combinatorial Optimization*, DIMACS Series in Discrete Mathematics and Theoretical Computer Science Vol. 20, American Mathematical Society, 181-242, 1995.
- [38] G. Raidl: private Kommunikation. 2002.
- [39] M. Savelsbergh, T. Volgenant: Edge exchanges in the degree-constrained minimum spanning tree problem. *Computers & Operations Research* 12, 341-348, 1985.
- [40] A. Schrijver: *Theory of Linear and Integer Programming*. New York: Wiley, 1986.
- [41] A. S. Schulz, R. Weismantel, G. M. Ziegler: 0/1 Integer Programming: Optimization and Augmentation are equivalent. In: P. Spirakis (Hrsg.) *ESA '95 Lecture Notes in Computer Science* Vol. 979, Berlin: Springer, 473-483, 1995.
- [42] S. Sharma, B. Sharma: New technique for solving primal all-integer linear programming. *Opsearch* 34, 62-68, 1997.
- [43] S. Thienel: A Simple TSP-Solver: An **ABACUS** Tutorial. *Technischer Bericht 96.245*, Institut für Informatik, Universität zu Köln, 1996.
- [44] P. Ventura: Some contributions to primal and dual separation for integer programming. *Doktorarbeit*, Istituto di Analisi dei Sistemi ed Informatica, Rom, 2002.

- [45] A. Volgenant: A Lagrangean approach to the degree-constrained minimum spanning tree problem. *European Journal of Operational Research* 39, 325-331, 1989.
- [46] R. D. Young: A simplified primal (all-integer) integer programming algorithm. *Operations Research* 16, 750-782, 1968.

Danksagung

An dieser Stelle möchte ich allen danken, die mich beim Erstellen dieser Diplomarbeit unterstützt haben. In erster Linie gilt mein besonderer Dank meinen Eltern dafür, dass sie mir ein Studium ermöglicht haben.

Herrn Prof. Dr. Jünger danke ich für die von ihm gewährte Möglichkeit, meine Diplomarbeit in seinem Lehrstuhl schreiben zu können. Seine über diese Arbeit hinausgehende Unterstützung erwies sich als sehr hilfreich.

Des Weiteren möchte ich Frauke Liers meinen Dank aussprechen. Sie hat mich auf die primale Separierung aufmerksam gemacht und mein Interesse an diesem Gebiet geweckt. Die viele Zeit, die sie sich zum Diskutieren genommen hat, und ihre überaus nützlichen Verbesserungsvorschläge und Anmerkungen haben meine Arbeitsweise geprägt. Neben dem Fachlichen war ihr auch stets mein Befinden wichtig, was meine Motivation positiv beeinflusst hat.

Ebenfalls bedanke ich mich bei Matthias Elf für die reichhaltigen Hilfen und die vielen Ratschläge zur Programmierung mit **ABACUS**, die mir die Fehlersuche bei der Implementierung vereinfacht haben.

Schließlich möchte ich mich noch bei allen Mitarbeitern im Institut für das freundliche und inspirierende Arbeitsklima, das sie während meiner gesamten Zeit im Institut geschaffen haben, recht herzlich bedanken.

Eidesstattliche Versicherung

Hiermit versichere ich, dass ich die vorliegende Arbeit selbständig angefertigt und keine anderen als die angegebenen Hilfsmittel verwendet habe.

Köln, im Februar 2003

(Markus Behle)