
Skriptausschnitt zur Vorlesung
Algorithm Engineering

Prof. Dr. Petra Mutzel
Carsten Gutwenger
Markus Chimani
Karsten Klein

Kapitel 3

NP-schwierige kombinatorische Optimierungsprobleme

Optimierungsprobleme sind Probleme, die im Allgemeinen viele zulässige Lösungen besitzen. Jeder Lösung ist ein bestimmter Wert (Zielfunktionswert, Kosten) zugeordnet. Optimierungsalgorithmen suchen in der Menge aller zulässigen Lösungen diejenigen mit dem besten, dem *optimalen*, Wert.

Zunächst führen wir formale Definitionen spezieller Optimierungsprobleme wie kombinatorische, lineare, und ganzzahlige Optimierungsprobleme ein. Danach folgt ein kurzer Abschnitt zur linearen Programmierung. Abschnitt 3.2 veranschaulicht den Zusammenhang zwischen kombinatorischen und 0/1-ganzzahligen Optimierungsproblemen.

3.1 Einführung

3.1.1 Kombinatorische Optimierungsprobleme

Kombinatorische Optimierungsprobleme tauchen relativ häufig in der Praxis auf. Ob es darum geht einen kürzesten Weg (z.B. beste Bahnverbindungen zwischen zwei Städten zu finden) oder eine kürzeste Rundtour zu berechnen (das Handlungsreisendenproblem), einen Baum kleinsten Gewichts in einem Graphen zu berechnen (z.B. für Kommunikationsaufbau), oder die Lagerhaltung einer Firma zu optimieren, immer spielen kombinatorische Optimierungsprobleme eine wichtige Rolle. In diesem Kapitel werden wir noch einige andere Anwendungen und Anwendungsbereiche kennenlernen.

Ein *kombinatorisches Optimierungsproblem* ist formal folgendermaßen definiert:

Definition: Gegeben seien eine endliche Menge E (*Grundmenge*), eine Teilmenge I der Potenzmenge 2^E von E (die Elemente heißen *zulässige Mengen* oder *zulässige Lösungen*) und eine Funktion $c : E \rightarrow K$. Für jede Menge $F \subseteq E$ definieren wir ihren Wert durch

4 KAPITEL 3. NP-SCHWIERIGE KOMBINATORISCHE OPTIMIERUNGSPROBLEME

$c(F) := \sum_{e \in F} c(e)$, und wir suchen eine Menge $I^* \in I$, so dass $c(I^*)$ so groß (oder klein) wie möglich ist.

Beispiel TSP: Das *Handlungsreisendenproblem* (*Travelling Salesman Problem, TSP*) ist ein kombinatorisches Optimierungsproblem. Hierbei muss ein Handlungsreisender (z.B. eine Reisegruppe) eine Menge von vorgegebenen Städten (z.B. Sehenswürdigkeiten in Wien) besuchen und am Ende wieder zu seinem Ausgangspunkt (z.B. Hotel) zurückkehren. Dabei darf er jede Stadt nur genau einmal besuchen; die Distanz (bzw. Wegstrecke) zwischen je 2 Städten ist durch $c_e \geq 0$ gegeben. Das Ziel des Handlungsreisenden ist es, seine zurückgelegte Weglänge (bzw. Reisezeit) zu minimieren. Die Grundmenge E entspricht hierbei der Menge aller Knotenpaare bzw. Kanten in dem vollständigen Graphen $K_n = (V, E)$. Die Teilmenge I der zulässigen Lösungen entspricht der Menge aller zulässigen Touren in K_n . Dabei ist eine Menge $F \subseteq E$ genau dann eine Tour, wenn F zu jedem Knoten in V genau zweimal inzident ist, und der durch F induzierte Subgraph zusammenhängend ist.

Beispiel: Das Problem, das Minimum der Funktion $f(x) = 3x^2 + 2$ zu finden, ist ein Optimierungsproblem, aber kein kombinatorisches Optimierungsproblem.

Kombinatorische Optimierungsprobleme zeichnen sich dadurch aus, dass die Menge der zulässigen Lösungen **endlich** (also auch diskret, nicht kontinuierlich) ist.

Beispiel: Das Problem

$$\max 2x_1 + 3x_2 \tag{3.1}$$

$$s.t. x_1 + 2x_2 \leq 3 \tag{3.2}$$

$$3x_1 - x_2 \leq 5 \tag{3.3}$$

$$x_1, x_2 \in \mathbb{N} \tag{3.4}$$

besitzt zwar nur Lösungen, die diskret sind. Da jedoch die Grundmenge \mathbb{N} nicht endlich ist, handelt es sich hierbei nicht um ein kombinatorisches Optimierungsproblem. (Achtung: Die Menge der zulässigen Lösungen ist sehr wohl endlich, wegen $x_1 + 2x_2 \leq 3$ und $x_1, x_2 \in \mathbb{N}$. Die Grundmenge jedoch nicht.)

Ersetzt man die Bedingung (3.4) durch die Bedingung

$$x_1, x_2 \in \{0, 1\}, \tag{3.5}$$

dann erhält man ein kombinatorisches Optimierungsproblem.

Typischerweise haben die interessanten kombinatorischen Optimierungsprobleme eine Anzahl an zulässigen Lösungen, die exponentiell in n ist, wie z.B. $n!$ oder 2^n , wenn $n = |E|$ die Anzahl der Elemente in der Grundmenge ist.

Zum Beispiel ist die Menge aller zulässiger Touren des Handlungsreisendenproblems auf n Städten gegeben durch

$$\frac{(n-1)!}{2}.$$

Eine vollständige Enumeration der Elemente solcher Mengen ist auch auf den größten Rechnern (für z.B. $n \geq 40$) nicht durchführbar. Das **Ziel der kombinatorischen Optimierung** besteht darin, Algorithmen zu entwerfen, die erheblich schneller als die Enumeration aller Lösungen sind.

Eng verwandt mit den kombinatorischen Optimierungsproblemen sind die *ganzzahligen (linearen) Optimierungsprobleme*, die wiederum eng mit *linearen Optimierungsproblemen* zusammenhängen.

3.1.2 Lineare Optimierungsprobleme

Lineare Optimierungsprobleme tauchen in sehr vielfältiger Form in der Praxis auf. Es handelt sich dabei um die Optimierung einer linearen Zielfunktion über einem Zulässigkeitsbereich, der durch lineare Funktionen gegeben ist.

Im folgenden sind Vektoren $c \in \mathbb{R}^n$ immer Spaltenvektoren; der dazugehörige Zeilenvektor wird mit c^T bezeichnet.

Definition: Seien m, n positive ganze Zahlen, $b \in \mathbb{R}^m, c \in \mathbb{R}^n$ und A eine $m \times n$ Matrix mit Elementen $a_{ij} \in \mathbb{R}$. Eine Instanz eines *linearen Optimierungsproblems* (oder kurz: *lineares Programm, LP*) ist das Problem, einen Vektor $x \in \mathbb{R}^n$ zu finden, der unter allen Vektoren, die die Bedingungen $Ax \leq b$ erfüllen, derjenige ist, mit größtem (bzw. kleinstem Wert) $c^T x$. Dabei nennt man die zu minimierende Funktion die *Zielfunktion* des Problems, die Bedingungen $Ax \leq b$ heißen auch *Restriktionen*. Jeder Vektor $x \in \mathbb{R}^n$, der alle Nebenbedingungen erfüllt, heißt *zulässige Lösung*. Die Menge $P = \{x \in \mathbb{R}^n \mid Ax \leq b\}$ heißt *Zulässigkeitsbereich*.

In Kurzform schreiben wir für ein LP:

$$\begin{aligned} \max \quad & c^T x \\ & Ax \leq b \\ & x \in \mathbb{R}^n \end{aligned}$$

Viele Optimierungsprobleme aus der Praxis können als lineare Programme formuliert werden. Anwendungen umfassen z.B. Probleme in der Produktionsplanung, Portfolio Optimierung, oder Transportprobleme.

Die **Modellierung** von Optimierungsproblemen als lineare Programme bzw. ganzzahlige lineare Programme ist eine wichtige sowie nicht-triviale Aufgabe in der Optimierungspraxis.

6 KAPITEL 3. NP-SCHWIERIGE KOMBINATORISCHE OPTIMIERUNGSPROBLEME

Als nächstes modellieren wir ein wichtiges Problem, das heute tatsächlich in der Ölindustrie mit Hilfe von linearer Programmierung gelöst wird (natürlich in ganz anderen Dimensionen).

Beispiel Ölraffinerie: In Ölraffinerien wird angeliefertes Rohöl durch Anwendung von chemischen und/oder physikalischen Verfahren in gewisse gewünschte Komponenten zerlegt. Die Ausbeute an verschiedenen Komponenten hängt von dem eingesetzten Verfahren (Crackprozeß) ab. Wir nehmen an, dass eine Raffinerie aus Rohöl drei Komponenten (schweres Öl S , mittelschweres Öl M , leichtes Öl L) herstellen will. Sie hat zwei Crackverfahren zur Verfügung, die die folgenden Einheiten an Ausbeute sowie Kosten bezogen auf jeweils 10 Einheiten Rohöl liefern:

Crackprozeß 1: $2S, 2M, 1L$, Kosten: 3 EUR

Crackprozeß 2: $1S, 2M, 4L$, Kosten: 5 EUR

Aufgrund von Lieferbedingungen muss die Raffinerie folgende Mindestproduktion herstellen: $3S, 5M$ und $4L$. Die Mengen sollen so kostengünstig wie möglich hergestellt werden.

Das hieraus resultierende lineare Programm erhalten wir nach Einführung von Variablen x_1 und x_2 , die jeweils das Produktionsniveau der beiden Prozesse beschreiben. Zum Beispiel bedeutet ein Wert $x_1 = 2.5$, dass der Crackprozeß 1 mit 25 Einheiten Rohöl beschickt wird. Jeder Vektor $(x_1, x_2) \in \mathbb{R}^2$ mit $x \geq 0$ beschreibt also ein mögliches Produktionsniveau der beiden Prozesse.

Angenommen, durch (x_1, x_2) sei ein Produktionsniveau beschrieben, daraus erfolgt ein Ausstoß von schwerem Öl von $2x_1 + x_2$. Die Lieferbedingungen erfordern die folgende Nebenbedingung

$$2x_1 + x_2 \geq 3.$$

Die Bedingungen für mittelschweres und leichtes Öl ergeben sich aus ähnlichen Überlegungen. Die Kosten betragen

$$z = 3x_1 + 5x_2.$$

Dies ergibt das folgende lineare Programm:

$$\min 3x_1 + 5x_2 \tag{3.6}$$

$$2x_1 + x_2 \geq 3 \tag{3.7}$$

$$2x_1 + 2x_2 \geq 5 \tag{3.8}$$

$$x_1 + 4x_2 \geq 4 \tag{3.9}$$

$$x_1 \geq 0 \tag{3.10}$$

$$x_2 \geq 0 \tag{3.11}$$

Beispiel Diätproblem: Bob möchte sich möglichst billig ernähren, allerdings so, dass er mindestens 2000 kcal, 55 g Proteine und 800 mg Calcium erhält. In seinem Laden um die

Ecke ist die Auswahl nicht sehr groß.

Es gibt Haferflocken zu 28 g Packungen, die 110 kcal liefern, 4 g Protein, und 2 mg Calcium, und 3 Cent kosten. Es gibt auch Huhn zu 24 Cent in 100 g Packungen, was 205 kcal liefert sowie 32 g Proteine und 12 mg Calcium. Eier kosten 13 Cent und kommen jeweils in Doppelpacks. Sie liefern 160 kcal, 13 g Proteine, 54 mg Calcium. Eine Packung Milch enthält 237 ml, liefert 160 kcal, 8 g Proteine, 285 g Calcium und kostet 9 Cent. Eine Kirschkuchen kostet 20 Cent, liefert 420 kcal, kommt in der 179 g Packung, enthält 4 g Proteine und 22 mg Calcium. Schließlich gibt es noch Bohnen in der 260 g Packung zu 19 Cent, die 260 kcal enthalten und 14 g Proteine sowie 80 mg Calcium.

Wir formulieren das Diätproblem als lineares Programm. Als erstes führen wir für jede mögliche Mahlzeit eine Variable ein, also x_1 für Haferflocken, x_2 für Huhn, x_3 für Eier, x_4 für Milch, x_5 für Kirschkuchen und x_6 für Bohnen. $x_1 = 2.5$ bedeutet also, dass Bob 2.5 Packungen Haferflocken zu sich nimmt.

Die Nebenbedingungen, die aus den Ernährungsbedingungen folgen, sind:

$$110x_1 + 205x_2 + 160x_3 + 160x_4 + 420x_5 + 260x_6 \geq 2000 \quad (3.12)$$

$$4x_1 + 32x_2 + 13x_3 + 8x_4 + 4x_5 + 14x_6 \geq 55 \quad (3.13)$$

$$2x_1 + 12x_2 + 54x_3 + 285x_4 + 22x_5 + 80x_6 \geq 800 \quad (3.14)$$

Die Zielfunktion (Kosten minimieren) ist die folgende:

$$\min 3x_1 + 24x_2 + 13x_3 + 9x_4 + 20x_5 + 19x_6 \quad (3.15)$$

Um zu verhindern, dass die Variablen negative Werte annehmen, benötigen wir noch Schranken:

$$x_i \geq 0 \quad \text{für alle } i = 1, \dots, 6. \quad (3.16)$$

Obere Schranken (d.h. Schranken der Art $x_i \leq M_i$ für alle $i = 1, \dots, 6$) werden hier nicht benötigt, da die Zielfunktion automatisch dafür sorgen wird, die Werte möglichst klein zu halten (da minimiert wird und alle Zielfunktionskoeffizienten größer gleich 0 sind).

Bob überlegt sich, dass er Haferflocken nicht ohne Milch zu sich nehmen möchte. Für je eine Packung Haferflocken benötigt er je eine halbe Packung Milch. Dies führt zu der zusätzlichen Ungleichung:

$$0.5x_4 \geq x_1. \quad (3.17)$$

Falls allerdings Bob nicht mehr als 2 Portionen Bohnen täglich essen möchte, dann sollten diesbezügliche Schranken addiert werden.

Lineare Programme tauchen in **verschiedenen Formulierungen** auf, z.B.

8 KAPITEL 3. NP-SCHWIERIGE KOMBINATORISCHE OPTIMIERUNGSPROBLEME

- max oder min $c^T x : Ax \geq b$
- min $c^T x : Ax \geq b$ und $x \geq 0$
- min $c^T x : Ax = b$ und $x \geq 0$

Diese Formulierungen können alle ineinander übergeführt werden (s. auch Übung). Folgende Übergangsregeln helfen. Dabei bezeichnet a_i die i -te Zeile der Matrix A sowie b_i der i -te Eintrag von Vektor b :

1. $\max c^T x \Leftrightarrow \min(-c)^T x$
2. $a_i^T x \leq b_i \Leftrightarrow (-a_i)^T x \geq -b_i$
3. $a_i^T x = b_i \Leftrightarrow a_i^T x \geq b_i$ und $(-a_i)^T x \geq -b_i$
4. $a_i^T x \geq b_i \Leftrightarrow a_i^T x + s_i = b_i, s_i \geq 0$

Dabei wird s_i als *Schlupfvariable* (*slack variable*) bezeichnet.

In der allgemeinsten Form können also lineare Programme mit der folgenden Formulierung auftreten:

Sind Matrizen $A \in \mathbb{R}^{(p,r)}$, $B \in \mathbb{R}^{(p,s)}$, $C \in \mathbb{R}^{(q,r)}$, $D \in \mathbb{R}^{(q,s)}$ und Vektoren $a \in \mathbb{R}^r$, $b \in \mathbb{R}^s$, $c \in \mathbb{R}^p$, $d \in \mathbb{R}^q$ gegeben, dann wird durch die folgende Formulierung ein LP in seiner allgemeinsten Form beschrieben:

$$\begin{aligned} \max \quad & a^T x + b^T y \\ & Ax + By = c \\ & Cx + Dy \leq d \\ & x \geq 0 \end{aligned}$$

Das Lösen linearer Optimierungsprobleme ist in polynomieller Zeit möglich. Es gibt hierfür eine mächtige Theorie sowie effiziente Lösungsalgorithmen, die in der Lage sind, Instanzen bis zu einigen Millionen von Variablen und Restriktionen zu lösen. Wir geben einen kurzen Einblick in lineare Programmierung und ihre Lösungsmethoden in Abschnitt 3.1.4.

3.1.3 Ganzzahlige Lineare Optimierungsprobleme

Treten bei linearen Programmen zusätzlich zu den linearen Nebenbedingungen Forderungen nach Ganzzahligkeit aller oder eines Teils der Variablen auf, so nennt man die linearen Programme *ganzzahlig* oder *gemischt-ganzzahlig*. Genauer, eine Aufgabe der Form

$$\begin{aligned}
 \max \quad & a^T x + b^T y \\
 & Ax + By = c \\
 & Cx + Dy \leq d \\
 & x \geq 0 \\
 & x \in \mathbb{Z}^r \\
 & y \in \mathbb{Z}^r
 \end{aligned}$$

heißt *ganzzahliges lineares Optimierungsproblem* oder kurz *ganzzahliges Programm*. Zur Abkürzung schreiben wir manchmal *GLP* oder auch *ILP* bzw. *IP* (Kurznotation für: *Integer Program*).

Wird die Ganzzahligkeit nur für einen Teil der Variablen gefordert, dann heißt es ein *gemischt-ganzzahliges lineares Optimierungsproblem* oder kurz *gemischt-ganzzahliges Programm* (*GGLP* oder *MIP* für *Mixed Integer Program*).

Natürlich treten ganzzahlige Programme selten in dieser allgemeinsten Form auf. Z.B. werden wir bei den ganzzahligen Programmen meistens auf Programme der Form

$$\begin{aligned}
 \max \quad & c^T x \\
 & Ax = b \\
 & x \geq 0 \\
 & x \in \mathbb{Z}^n
 \end{aligned}$$

$$\begin{aligned}
 \max \quad & c^T x \\
 & Ax \leq b \\
 & x \in \mathbb{Z}^n
 \end{aligned}$$

$$\begin{aligned}
 \max \quad & c^T x \\
 & Ax \leq b \\
 & x \geq 0 \\
 & x \in \mathbb{Z}^n
 \end{aligned}$$

stoßen.

Dürfen die Variablen in einem linearen Programm nur die Werte 0 oder 1 annehmen, so spricht man von einem *binären linearen Programm* oder *0/1-Programm* (*BLP*). 0/1-

10KAPITEL 3. NP-SCHWIERIGE KOMBINATORISCHE OPTIMIERUNGSPROBLEME

Programme kommen meistens in folgender Form vor:

$$\begin{aligned} \max c^T x \\ Ax &\leq b \\ x &\in \{0, 1\}^n. \end{aligned}$$

Man beachte, dass die Nichtnegativitätsbedingung $x \geq 0$ hinzugefügt werden kann, ohne dass an der Menge der zulässigen Lösungen etwas geändert wird.

Ganzzahlige bzw. gemischt-ganzzahlige Optimierungsprobleme treten immer dann auf, wenn gewisse Produkte oder Ressourcen nicht beliebig geteilt werden können. Eine Fluggesellschaft kann nicht $\frac{7}{2}$ Flugzeuge kaufen, sondern nur 3 oder 4, und ebenso macht es keinen Sinn, $\frac{29}{2}$ Lokomotiven zu produzieren. Die in linearen Programmen zusätzlich auftretende Ganzzahligkeit führt zu erheblichen theoretischen und rechnerischen Problemen. Heutzutage kann man lineare Programme fast beliebiger Größenordnung lösen, dagegen können nicht selten ganzzahlige Programme mit 100 Variablen und 100 Nebenbedingungen selbst in Hunderten von Stunden CPU-Zeit nicht gelöst werden. Denn im Gegensatz zur linearen Programmierung ist ganzzahlige lineare Optimierung NP-schwierig.

Beispiel Rucksackproblem: Für den Stückguttransport von einem Ort A zu einem Ort B stehen n verschiedene Güterarten in unbegrenzter Menge zur Verfügung. Für den Transport eines Stücks der Güterart i , das ein Gewicht von $a_i > 0$ besitzt, werden c_i Geldeinheiten gezahlt. Die Kapazität eines Lastwagens, der zum Transport der Güter eingesetzt werden soll, beträgt b Gewichtseinheiten. Wie hat man den Lastwagen zu beladen, damit der Umsatz maximal wird? Wir führen Variable $x_i, i = 1, \dots, n$ ein, die die Anzahl der eingepackten Güter der Art i angeben. Das Problem führt zum folgenden ganzzahligen linearen Programm:

$$\begin{aligned} \max \quad & c_1x_1 + c_2x_2 + \dots + c_nx_n \\ & a_1x_1 + a_2x_2 + \dots + a_nx_n \leq b \\ & x_i \geq 0 \text{ für } i = 1, 2, \dots, n \\ & x_i \in \mathbb{Z} \text{ für } i = 1, 2, \dots, n. \end{aligned}$$

Ein GLP mit dieser Struktur (Vorzeichenbeschränkungen, Ganzzahligkeitsbedingungen und nur eine Ungleichung) heißt *Rucksackproblem*.

Das allgemeine Rucksackproblem ist also kein kombinatorisches Optimierungsproblem, da die Grundmenge nicht endlich ist. In dem Fall, dass es nur eine beschränkte Anzahl Güter für alle Güterarten gäbe, dann hätten wir das *binäre Rucksackproblem*, das wir bereits in *DAP 2* behandelt haben. Hierbei handelt es sich um ein kombinatorisches Optimierungsproblem.

Ganzzahlige bzw. gemischt-ganzzahlige Optimierungsprobleme tauchen sehr häufig in der Praxis auf. Sie sind z.B. inzwischen für Fluggesellschaften sehr wichtig geworden, um die Flugpläne zu gestalten, die Flugpreise festzulegen (s. auch Übung), und um das Flugpersonal einzuteilen. Aber auch das Transportwesen von Gütern und öffentliche Verkehrssysteme

lassen sich als ganzzahlige lineare Programme formulieren. Wichtige neue Anwendungen liegen im Finanzbereich (z.B. Portfolio-Optimierung).

3.1.4 Lineare Programmierung

Wir betrachten das folgende lineare Programm für $x \in \mathbb{R}^n$, $b \in \mathbb{R}^m$ und $A \in \mathbb{R}^{(m,n)}$.

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

Für den Zulässigkeitsbereich P eines linearen Programms gibt es drei verschiedene Möglichkeiten.

1. $P = \emptyset \Rightarrow$ In diesem Fall ist das LP unlösbar, denn es existiert keine einzige zulässige Lösung.
2. $P \neq \emptyset$, aber das $\inf\{c^T x \mid x \in P\}$ existiert nicht (z.B. $0x \geq -1$) \Rightarrow In diesem Fall ist das LP lösbar, aber es gibt keine optimale Lösung.
3. $P \neq \emptyset$ und das $\min\{c^T x \mid x \in P\}$ existiert \Rightarrow Das LP ist lösbar und hat eine endliche Lösung x^* mit $c^T x^* = \min\{c^T x \mid x \in P\}$.

Eine Aufgabe der linearen Programmierung besteht darin, herauszufinden, welcher der drei Fälle für das zu lösende LP zutrifft, und falls (3) zutrifft, die optimale Lösung zu finden.

Geometrische Interpretation

Eine anschauliche Möglichkeit lineare Programme und deren Lösungen zu studieren bietet die geometrische Interpretation. Wir beginnen die geometrische Interpretation mit dem Beispiel Öltraffinerie (s. Abschnitt 3.1.2).

$$\min 3x_1 + 5x_2 \tag{3.18}$$

$$2x_1 + x_2 \geq 3 \tag{3.19}$$

$$2x_1 + 2x_2 \geq 5 \tag{3.20}$$

$$x_1 + 4x_2 \geq 4 \tag{3.21}$$

$$x_1 \geq 0 \tag{3.22}$$

$$x_2 \geq 0 \tag{3.23}$$

Wir können das LP, das durch die linearen Funktionen (3.18) bis (3.23) definiert wird, graphisch interpretieren, da es nur zwei Variablen besitzt, und folglich sich die Lösungsmenge im 2-dimensionalen Raum befindet. Ersetzen wir das “ \geq ” Zeichen in (3.19) bis (3.23) durch ein Gleichheitszeichen, so erhalten wir 5 Gleichungen. Die Lösungsmenge einer Gleichung im \mathbb{R}^2 ist bekanntlich eine Gerade. Diese 5 Geraden beranden die Lösungsmenge

12KAPITEL 3. NP-SCHWIERIGE KOMBINATORISCHE OPTIMIERUNGSPROBLEME

des Systems der Ungleichungen (3.19) bis (3.23). Jede dieser Ungleichungen definiert einen Halbraum oberhalb der dazugehörigen Geraden. Der Zulässigkeitsbereich ist genau der Durchschnitt aller dieser Halbräume und ist in Abbildung 3.1 graphisch dargestellt.

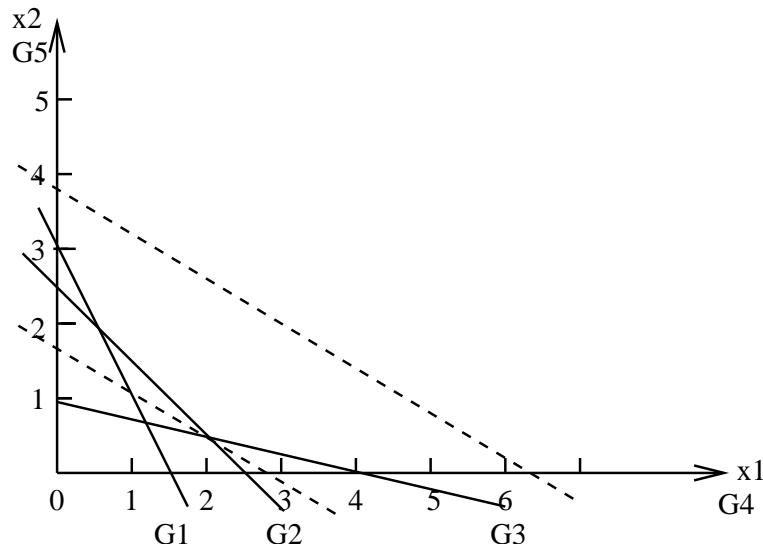


Abbildung 3.1: Graphische Interpretation der Lösungsmenge des Ölproblems

Die Zielfunktion ist keine Gerade, sie repräsentiert eine Schar paralleler Geraden. Nehmen wir z.B. den Punkt $x = (3, 2)$, der alle Ungleichungen erfüllt. Sein Zielfunktionswert ist 19, d.h. bei diesem Produktionsniveau treten Gesamtkosten in Höhe von 19 Einheiten auf. In Abbildung 3.1 ist die Gerade

$$G = \{x \mid 3x_1 + 5x_2 = 19\}$$

gestrichelt gezeichnet. Die Menge aller derjenigen Punkte, die auf dieser Geraden liegen und Ungleichungen (3.19) bis (3.23) erfüllen, stellen Produktionsniveaus mit Gesamtkosten 19 Einheiten dar.

Geometrisch ist nun klar, wie wir einen Punkt finden können, der alle Ungleichungen erfüllt und die Zielfunktion minimiert: Wir verschieben die Gerade G so lange parallel in Richtung auf den Punkt $(0,0)$, bis die verschobene Gerade die Zulässigkeitsbereich nur noch tangiert. Führen wir dies graphisch durch, so sehen wir, dass wir die Tangentialstellung im Punkt $x^* = (2, 0.5)$ erreichen. Die zu G parallele Gerade

$$G' = \{x \mid 3x_1 + 5x_2 = 8.5\}$$

berührt die Lösungsmenge in nur einem Punkt, nämlich x^* , jeder weitere Parallelverschiebung würde zu einem leeren Durchschnitt mit dieser Lösungsmenge führen. Wir

schließen daraus, dass $x^* = (2, 0.5)$ die Optimallösung unseres Problems ist, d.h. alle Lieferverpflichtungen können bei diesem Produktionsniveau erfüllt werden, und alle anderen Produktionsniveaus führen zu Gesamtkosten, die höher sind als die Kosten von 8.5 Einheiten, die beim Produktionsniveau von x^* anfallen.

Ein Beispiel eines Zulässigkeitsbereichs im 3-dimensionalen Raum finden Sie in Abschnitt 3.2.

Generell kann man sagen, dass die Lösungsmenge eines LPs mit n Variablen immer ein Polyeder im n -dimensionalen Raum darstellt. Denn jede Ungleichung definiert einen Halbraum im n -dimensionalen Raum. Der Schnitt von endlich vielen Halbräumen ist entweder leer oder definiert ein Polyeder. Die optimale Lösung eines LPs wird — unabhängig von der Zielfunktion — immer an einer Ecke des Polyeders angenommen (falls es eine optimale Lösung gibt). Genauer: es gibt immer eine optimale Ecklösung. Wir werden diese Begriffe in den folgenden Abschnitten formalisieren.

Bemerkungen zum Simplexalgorithmus

Lineare Programme werden heute in der Praxis mit Hilfe des Simplexalgorithmus gelöst. Die Idee hiervon ist, bei einer Ecke des Polyeders zu beginnen (zulässige Startlösung) und dann durch einen einfachen Austauschschritt zu einer benachbarten Ecke des Polyeders mit besserem Zielfunktionswert zu gelangen. Falls keine solche benachbarte Ecke existieren sollte, dann hat man eine optimale Lösung mit dem optimalen Zielfunktionswert erreicht. Mit einer Lösung erhält man auch immer gleichzeitig eine sogenannte duale Lösung (die Lösung des dualen linearen Programmes), die beweist, dass die gefundene Lösung auch tatsächlich optimal ist (man kann also nie in eine Sackgasse laufen).

Dennoch existieren *worst-case* Beispiele, bei denen der Simplexalgorithmus nicht in polynomieller Zeit läuft. Diese sind jedoch so selten, dass der Simplexalgorithmus heute noch der am weitesten verwendete Algorithmus in der Praxis ist (und der schnellste), obwohl es inzwischen auch Polynomialzeitalgorithmen gibt (z.B. die sogenannte *Ellipsoidmethode*).

Der Simplexalgorithmus wurde 1947 von George Dantzig entwickelt. Seitdem gab es große Fortschritte. Heute ist es möglich, LPs mit mehreren Millionen von Variablen und Nebenbedingungen zu lösen. Das am weitesten verbreitete Programm hierfür ist CPLEX. CPLEX benötigt z.B. zur Lösung eines LPs mit 1.584.000 Variablen und 401.640 Nebenbedingungen 59,1 Sekunden (2.0 GHz P4).

Lineare Programmierung ist heute eine wichtige Voraussetzung zum exakten Lösen von praktischen ganzzahligen bzw. gemischt-ganzzahligen linearen Programmen.

Eine sehr schöne Einführung in die lineare Programmierung und in den Simplexalgorithmus finden Sie in: V. Chvátal, *Linear Programming*, W.H. Freeman and Company, New York,

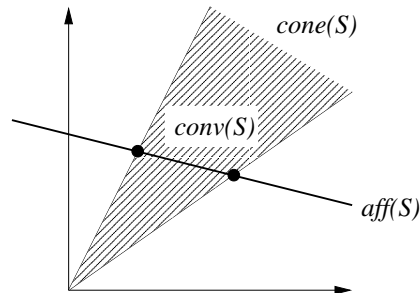


Abbildung 3.2: Illustration der Begriffe $cone(S)$, $conv(S)$ und $aff(S)$

1983.

3.1.5 Grundlagen der Polyedertheorie

Wir beginnen mit Grundbegriffen aus der Linearen Algebra. Für $x_1, x_2, \dots, x_k \in \mathbb{R}$ und

$$\lambda = \begin{pmatrix} \lambda_1 \\ \lambda_2 \\ \vdots \\ \lambda_k \end{pmatrix} \in \mathbb{R}^k$$

heißt

$$x = \sum_{i=1}^k \lambda_i x_i$$

Linearkombination von x_1, \dots, x_k . Gilt zusätzlich

$$\left. \begin{array}{l} \lambda \geq 0 \\ \sum_{i=1}^k \lambda_i = 1 \\ \sum_{i=1}^k \lambda_i = 1, \lambda \geq 0 \end{array} \right\} \text{ so heißt } x \left\{ \begin{array}{l} \text{konische} \\ \text{affine} \\ \text{konvexe} \end{array} \right\}$$

Kombination von x_1, x_2, \dots, x_k . Für $\emptyset \neq S \subseteq \mathbb{R}$ heißt

$$\left. \begin{array}{l} \text{lin}(S) \\ \text{cone}(S) \\ \text{aff}(S) \\ \text{conv}(S) \end{array} \right\} \text{ die } \left\{ \begin{array}{l} \text{lineare} \\ \text{konische} \\ \text{affine} \\ \text{konvexe} \end{array} \right\} \text{ Hülle von } S,$$

d.h. die Menge aller Vektoren, die als entsprechende Kombination endlich vieler Vektoren aus S dargestellt werden können. Abbildung 3.2 veranschaulicht die verschiedenen Begriffe für die beiden dargestellten Punkte.

Falls

$$\left. \begin{array}{l} S = \text{lin}(S) \\ S = \text{cone}(S) \\ S = \text{aff}(S) \\ S = \text{conv}(S) \end{array} \right\} \text{ so ist } S \text{ ein } \left\{ \begin{array}{l} \text{linearer Raum} \\ \text{Kegel} \\ \text{affiner Raum} \\ \text{konvexe Menge} \end{array} \right.$$

Es gilt: $L \subseteq \mathbb{R}^n$ ist ein *linearer Unterraum* des \mathbb{R}^n genau dann wenn ein $A \in \mathbb{R}^{m \times n}$ existiert, so dass $L = \{x \in \mathbb{R}^n \mid Ax = 0\}$. $L \subseteq \mathbb{R}^n$ ist ein *affiner Unterraum* des \mathbb{R}^n genau dann wenn ein $A \in \mathbb{R}^{m \times n}, b \in \mathbb{R}^m$ existiert, so dass $L = \{x \in \mathbb{R}^n \mid Ax = b\}$.

Für die lineare Programmierung sind spezielle affine Unterräume, die *Hyperebenen*, interessant, die dem Spezialfall $m = 1$ entsprechen. Eine Hyperebene ist also definiert als

$$\{x \in \mathbb{R}^n \mid a^T x = a_0\} \text{ für } a \in \mathbb{R}^n \setminus \{0\}, a_0 \in \mathbb{R}.$$

Eine Menge $S = \{x_1, x_2, \dots, x_k\} \subseteq \mathbb{R}^n$ von Vektoren heißt *linear unabhängig*, wenn aus $\sum_{i=1}^k \lambda_i x_i = 0$ folgt, dass für alle $\lambda_i = 0$ gilt (für alle $i \in \{1, 2, \dots, k\}$). Sie heißt *affin unabhängig*, wenn aus $\sum_{i=1}^k \lambda_i x_i = 0, \sum_{i=1}^k \lambda_i = 0$ folgt, dass für alle $\lambda_i = 0$ gilt (für alle $i \in \{1, 2, \dots, k\}$).

Für $S \subseteq \mathbb{R}^n$ ist der

$$\left. \begin{array}{l} \text{Rang} \quad \text{rang}(S) \\ \text{affine Rang} \quad \text{affrang}(S) \end{array} \right\} \text{ von } S$$

definiert durch $\left\{ \begin{array}{l} \max\{|T| \mid T \subseteq S \text{ ist linear unabhängig}\} \\ \max\{|T| \mid T \subseteq S \text{ ist affin unabhängig}\} \end{array} \right\}$

Die *Dimension* von S ist $\dim(S) = \text{affrang}(S) - 1$. Es gilt:

$$\begin{aligned} 0 \in \text{aff}(S) &\Rightarrow \dim(S) = \text{rang}(S) \\ 0 \notin \text{aff}(S) &\Rightarrow \dim(S) = \text{rang}(S) - 1 \end{aligned}$$

$S \subseteq \mathbb{R}^n$ heißt *volldimensional* falls $\dim(S) = n$.

Für $a \in \mathbb{R}^n, a \neq 0, a_0 \in \mathbb{R}$ heißt

$$H = \{x \in \mathbb{R}^n \mid a^T x \leq a_0\}$$

Halbraum, die durch H definierte Hyperebene ist $\{x \in \mathbb{R}^n \mid a^T x = a_0\}$. Eine Ungleichung $a^T x \leq a_0$ ist *gültig* bezüglich einer Teilmenge $S \subseteq \mathbb{R}^n$ genau dann, wenn $S \subseteq \{x \in \mathbb{R}^n \mid a^T x \leq a_0\}$.

Eine Teilmenge $P \subseteq \mathbb{R}^n$ heißt *Polyeder*, falls es eine Matrix $A \in \mathbb{R}^{(m,n)}$ und einen Vektor $b \in \mathbb{R}^m$ gibt mit

$$P = \{x \in \mathbb{R}^n \mid Ax \leq b\}.$$

16KAPITEL 3. NP-SCHWIERIGE KOMBINATORISCHE OPTIMIERUNGSPROBLEME

Mit den Formulierungstricks aus Abschnitt 3.1.2 ($Dx \leq d$, $-Dx \leq -d$) ist also auch

$$P = \{x \in \mathbb{R}^n \mid Ax \leq b, Dx = d\}$$

ein Polyeder. Halbräume sind offensichtlich Polyeder. Aber auch die leere Menge ist ein Polyeder, denn $\emptyset = \{x \mid 0^T x \leq -1\}$, und der gesamte Raum ist ein Polyeder, denn $\mathbb{R}^n = \{x \mid 0^T x \leq x\}$. Es gilt: Jedes Polyeder $P \neq \mathbb{R}^n$ ist der Durchschnitt von endlich vielen Halbräumen.

Wir haben es häufig auch mit *Polytopen* zu tun. Ein *Polytop* ist ein beschränktes Polyeder:

$$P \subseteq \{x \in \mathbb{R}^n \mid \|x\| \leq B\} \text{ für ein } B > 0.$$

Man kann zeigen, dass die lineare, die affine, die konvexe bzw. die konische Hülle einer endlichen Teilmenge des \mathbb{R}^n ein Polyeder ist. Aus den klassischen Theoremen von Minkowski und Weyl folgt:

SATZ 3.1 [Minkowski 1896, Weyl 1935] Jedes Polyeder $P \in \mathbb{R}^n$ besitzt eine Darstellung der Form $P = \text{conv}(V) + \text{cone}(E)$, wobei V und E endliche Teilmengen des \mathbb{R}^n entsprechen und umgekehrt.

Polyeder sind also genau die Teilmengen des \mathbb{R}^E , die sich als Summe von konvexen und konischen Hüllen endlicher Teilmengen des \mathbb{R}^E darstellen lassen. Deshalb existieren immer zwei Darstellungen von Polyedern

$$P = \{x \in \mathbb{R}^n \mid Ax \leq b\} = \text{conv}(V) + \text{cone}(E)$$

Polytope lassen sich durch $P = \text{conv}(V)$ darstellen.

Beispiel: Wir betrachten das durch

$$P = \left\{ \begin{pmatrix} 0 \\ 1 \end{pmatrix}, \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \begin{pmatrix} 1 \\ 1 \end{pmatrix} \right\}$$

definierte Polytop. Abbildung 3.3 veranschaulicht P .

Wir wissen also, dass es eine Darstellung durch Ungleichungen gibt. Diese ist graphisch leicht ablesbar:

$$P = \left\{ \begin{pmatrix} x \\ y \end{pmatrix} \in \mathbb{R}^2 \mid x \leq 1, y \leq 1, x + y \geq 1 \right\}$$

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 $F = \{x \in P \mid a^T x = a_0\}$. F ist *echte Seitenfläche* falls $F \neq P$. Ist

$$P = \{x \in \mathbb{R}^n \mid a_i^T x \leq b, i = 1, 2, \dots, k\} = \text{conv}(S)$$

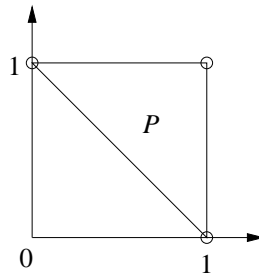


Abbildung 3.3: Ein Polytop des Beispiels zu den verschiedenen Darstellungen

und F Seitenfläche von P , so gibt es eine Teilmenge der Indizes $I \subseteq \{1, 2, \dots, k\}$, so dass

$$F = \{x \in \mathbb{R}^n \mid a_i^T \leq b_i, i \in I\}.$$

Weiterhin gibt es auch eine Teilmenge $R \subseteq S$, so dass $F = \text{conv}(R)$.

Eine *Ecke* eines Polyeders P ist definiert als eine einelementige Seitenfläche von P . Eine *Facette* von P ist definiert als eine maximale nicht leere echte Seitenfläche von P . Für jede Facette F von P gilt $\dim(F) = \dim(P) - 1$.

3.2 Kombinatorische vs. Ganzzahlige Optimierung

Kombinatorische und ganzzahlige lineare Optimierungsprobleme stehen in enger Beziehung. Sie sind sogar in einer gewissen Beziehung äquivalent. Man kann jedes kombinatorische Optimierungsproblem als ganzzahliges 0/1-Programm formulieren und umgekehrt. Hierzu benötigen wir die folgenden Definitionen.

Ist E eine endliche Menge, so bezeichnen wir mit \mathbb{R}^E den \mathbb{R} -Vektorraum der $|E|$ -Tupel (oder Spaltenvektoren der Länge $|E|$) $x = (x_e)_{e \in E}$, so dass jede Komponente von x mit einem Element von E indiziert ist. Das heißt, jedem Element $e \in E$ ist eine Komponente x_e von x zugeordnet und umgekehrt. Ist E eine endliche Menge und $F \subseteq E$, dann ist der *charakteristische Vektor* $\chi^F \in \mathbb{R}^E$ für F definiert als

$$\chi_e^F = 1 \Leftrightarrow e \in F \quad (3.24)$$

$$\chi_e^F = 0 \Leftrightarrow e \notin F \quad (3.25)$$

Umgekehrt, ist jeder 0/1-Vektor $x \in \{0, 1\}^E$ charakteristischer Vektor einer Teilmenge F_x von E , und zwar gilt:

$$F_x = \{e \in E \mid x_e = 1\}, \chi^{F_x} = x.$$

Ist nun ein binäres lineares Programm der Form

$$\max\{c^T x \mid Ax \leq b, x \in \{0, 1\}^n\}$$

gegebenen, so setzen wir:

$$E := \{1, 2, \dots, n\} \quad (3.26)$$

$$I := \{F_x \subseteq E \mid x \in \{0, 1\}^E, Ax \leq b\} \quad (3.27)$$

$$c(F_x) := c^T \chi^{F_x}. \quad (3.28)$$

E ist also die Grundmenge, I ist die Menge der zulässigen Lösungen, und c ist die Zielfunktion des dazugehörigen kombinatorischen Optimierungsproblems.

Ist umgekehrt ein kombinatorisches Optimierungsproblem mit Mengen e , I und Zielfunktion c gegeben, dann setzen wir

$$P_F = \text{conv}\{\chi^F \in \mathbb{R}^E \mid F \in I\}.$$

P_F ist also die konvexe Hülle endlich vieler Vektoren, also ein Polytop, dessen Ecken genau den charakteristischen Vektoren der zulässigen Mengen $F \in I$ entsprechen. Jedes Polytop P hat eine Darstellung durch Gleichungen und Ungleichungen (s. Abschnitt 3.1.4). Fassen wir die Funktion $c : E \rightarrow \mathbb{R}$ als Vektor $c \in \mathbb{R}^E$ auf, so ist jede optimale Ecklösung des linearen Programms

$$\max\{c^T x \mid x \in P_F\}$$

der charakteristische Vektor einer Optimallösung des gegebenen kombinatorischen Optimierungsproblems (und umgekehrt).

Wir haben somit das kombinatorische Optimierungsproblem sogar in ein lineares Programm umformuliert, jedoch hat diese Formulierung einige Haken. Denn, um ein lineares Programm über einem Polytop zu lösen, ist eine Formulierung mit Hilfe von Gleichungen und Ungleichungen nötig, die hier nicht gegeben ist. Unglücklicherweise existiert kein polynomieller Algorithmus, um eine solche Darstellung zu berechnen. Die Situation ist sogar noch schlimmer: Im allgemeinen hat eine solche Beschreibung exponentiell viele Ungleichungen und Gleichungen, die noch dazu im allgemeinen Koeffizienten exponentieller Größe haben können.

In der Praxis genügt es jedoch, nur einen kleinen Teil dieser Ungleichungen und Gleichungen zu kennen. Es gelingt relativ häufig, eine *einfache* Beschreibung des Polytops in Form eines ganzzahligen linearen Programmes zu bestimmen. In der Praxis wird diese ILP-Formulierung oft als Ausgangsbasis genommen, um eine partielle Formulierung des Polytops P_F zu bestimmen. Das relativ junge Forschungsgebiet der *polyedrischen Kombinatorik* beschäftigt sich mit dieser Aufgabenstellung. Es hat sich gezeigt, dass oft ein relativ kleiner Teil des Ungleichungssystems notwendig ist, um gegebene kombinatorische Optimierungsprobleme auf diese Weise beweisbar optimal zu lösen.

Beispiel MST: Betrachte eine Instanz des *Minimum Spanning Tree Problems*, (MST) mit drei Punkten (s. Abb. 3.4). Gesucht ist also ein spannender Baum in K_3 (d.h. ein kreisfreier

Untergraph von K_3) minimalen Gewichts bei gegebenen nicht-negativen Distanzen d_1, d_2 und d_3 . Abbildung 3.4 zeigt den dazugehörigen Graphen sowie die Menge aller möglichen Lösungen dieses MST-Problems.

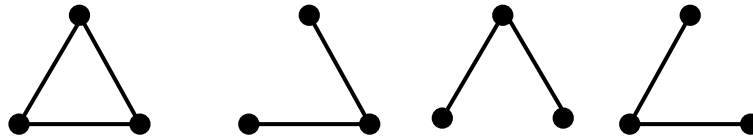


Abbildung 3.4: Graph K_3 und Menge aller zulässigen Spannblätter auf drei Punkten

Wie kann man nun diese Menge mit Hilfe von Variablen und linearen Ungleichungen beschreiben? Wir führen dazu drei Variablen im 3-dimensionalen Raum ein (für jede Kante genau eine), die wir $x_j = 1$ setzen, wenn die dazugehörige Kante e_j in der zulässigen Lösung (die also einem spannenden Baum entspricht) enthalten ist, und $x_j = 0$ sonst (für alle $j = 1, 2, 3$). Welche Bedingungen müssen die Variablen erfüllen, damit sie einem charakteristischen Vektor eines spannenden Baumes entsprechen? In diesem Beispiel ist das einfach: Wir sehen, dass jede zulässige Lösung genau zwei Variablen besitzt, die 1 gesetzt sind, d.h. wir haben die Gleichung

$$x_1 + x_2 + x_3 = 2. \quad (3.29)$$

Und diese zusammen mit den Schranken

$$x_j \geq 0 \text{ und } x_j \leq 1 \text{ für alle } j = 1, 2, 3 \quad (3.30)$$

und den Ganzzahligkeitsbedingungen

$$x_j \in \mathbb{Z} \text{ für alle } j = 1, 2, 3 \quad (3.31)$$

genügt in diesem Fall (MST in drei Punkten) bereits für eine gültige ILP-Formulierung. Damit ist das ILP gegeben durch:

$$\min \quad d_1x_1 + d_2x_2 + d_3x_3 \quad (3.32)$$

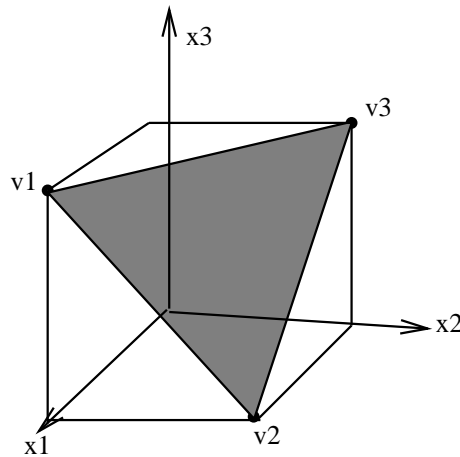
$$x_1 + x_2 + x_3 = 2 \quad (3.33)$$

$$x_j \geq 0 \quad \text{für } j = 1, 2, 3 \quad (3.34)$$

$$x_j \leq 1 \quad \text{für } j = 1, 2, 3 \quad (3.35)$$

Jede Lösung, die diese Bedingungen erfüllt, entspricht also einem charakteristischen Vektor eines spannenden Baums. Diese drei Lösungen entsprechen den ganzzahligen Punkten v_1, v_2 und v_3 in Abbildung 3.5.

Der Lösungsraum des linearen Programms (ohne Berücksichtigung der Ganzzahligkeitsbedingungen) ist die in Abbildung 3.5 markierte Fläche. Die optimale Lösung (über einer

Abbildung 3.5: Der Zulässigkeitsbereich des MST-Problems auf K_3

beliebigen linearen Zielfunktion) wird auch hier an den einzigen Ecken des Lösungsraumes angenommen. Diese stimmen genau mit den drei Punkten v_1 , v_2 und v_3 überein, die ganzzahlig sind. Deswegen können hier die Ganzzahligkeitsbedingungen aus dem ursprünglichen ILP weggelassen werden, ohne, dass sich die Lösung ändert.

Dies gilt allerdings nicht mehr, wenn wir das MST-Problem auf $N > 3$ Punkten betrachten. Dann müssen wir weitere Ungleichungen hinzunehmen.

3.3 Beispiel: Das Lineare Ordnungsproblem (LOP)

Das *Lineare Ordnungsproblem* (*Linear Ordering Problem*, (LOP)) ist gegeben durch einen vollständigen gerichteten Graphen $D_n = (V, A)$ mit Bogengewichten $c \in \mathbb{R}^A$. Gesucht ist eine lineare Ordnung der Knoten, so dass die Summe der Gewichte aller Bögen, die dieser Ordnung entsprechen, maximiert wird.

Diese Bedingung kann alternativ durch sogenannte azyklische Turniere formuliert werden. Eine Teilmenge $T \subseteq A$ heißt *Turnier* genau dann, wenn für alle $u, v \in V$, $u \neq v$ gilt entweder $(u, v) \in T$ oder $(v, u) \in T$, aber nicht beide. Abbildung 3.6 zeigt als Beispiel den vollständigen gerichteten Digraphen D_4 und ein Turnier.

Ein Turnier heißt *azyklisch*, wenn es keine gerichteten Kreise enthält. Die azyklischen Turniere in D_n entsprechen genau der Menge aller linearen Ordnungen von n Knoten. Abbildung 3.7 zeigt ein azyklisches Turnier und die dazugehörige lineare Ordnung.

Gesucht ist ein maximales azyklisches Turnier, d.h. $T \subseteq A$, T azyklisches Turnier, das die Bogengewichte maximiert. Wir wollen nun das kombinatorische Optimierungsproblem



Abbildung 3.6: Vollständiger Digraph D_4 und ein Turnier

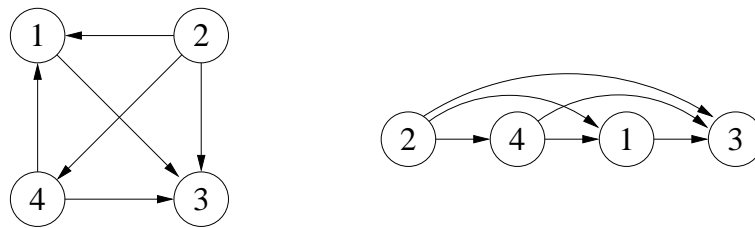


Abbildung 3.7: Ein azyklisches Turnier und die dazugehörige lineare Ordnung

(LOP) als 0/1-Programm schreiben. Hierzu führen wir einen Vektor $x \in \{0, 1\}^A$ ein, der mit den Bögen A assoziiert ist.

Welche Bedingungen muss nun x erfüllen, damit dieser als ein charakteristischer Vektor einer Menge F gedeutet werden kann, die einem azyklischen Turnier entspricht? Zunächst darf nur genau ein Bogen (u, v) oder (v, u) zwischen jedem Knotenpaar u, v gegeben sein. Dies führt zu einer Gleichung

$$x_{uv} + x_{vu} = 1 \text{ für alle Knotenpaare } u, v \in V. \quad (3.36)$$

Die Menge

$$\{x \in \{0, 1\}^A \mid x_{uv} + x_{vu} = 1 \text{ für alle } u, v \in V, u \neq v\}$$

entspricht also der Menge aller Turniere in $D_n = (V, A)$. Als nächstes müssen noch alle gerichteten Kreise verboten werden. Dies geht dadurch, in dem wir fordern, dass jeder gerichtete Kreis C mit k Bögen höchstens $k - 1$ Bögen mit $x_{uv} = 1$ enthalten darf. Die dazugehörige Ungleichung lautet:

$$\sum_{(u,v) \in C} x_{uv} \leq |C| - 1$$

für alle Kreise $C \subseteq A$, wobei $|C|$ die Anzahl der in C enthaltenen Bögen angibt. Man schreibt auch $x(C)$ für $\sum_{(u,v) \in C} x_{uv}$.

22 KAPITEL 3. NP-SCHWIERIGE KOMBINATORISCHE OPTIMIERUNGSPROBLEME

Man kann zeigen, dass es genügt, in diesem speziellen Fall nur alle Kreise der Länge 3 auszuschließen. Z.B. erhält man die Ungleichung für den 5-er Kreis

$$C_5 = \{1, 2, 3, 4, 5\}$$

durch die Addition der Ungleichungen der 3-er Kreise

$$C_1 = \{1, 2, 3\}, C_2 = \{3, 4, 1\}, C_3 = \{4, 5, 1\} :$$

$$\begin{array}{lll} C_1 : & x_{12} + x_{23} + x_{31} & \leq 2 \\ C_2 : & x_{34} + x_{41} + x_{13} & \leq 2 \\ C_3 : & x_{45} + x_{51} + x_{14} & \leq 2 \end{array}$$

$$\text{Summe: } x_{12} + x_{23} + (x_{31} + x_{13}) + x_{34} + (x_{41} + x_{14}) + x_{45} + x_{51} \leq 6$$

Dies ergibt wegen $x_{31} + x_{13} = 1$ und $x_{41} + x_{14} = 1$ schließlich

$$x_{12} + x_{23} + x_{34} + x_{45} + x_{51} \leq 4.$$

Genauso kann man die Ungleichung eines beliebigen anderen Kreises durch die 3-er Kreise erzeugen. Dies gilt in diesem speziellen Fall, weil der gegebene Graph vollständig und gerichtet ist, d.h. für jedes mögliche gerichtete Knotenpaar (u, v) existiert eine Kante in G , d.h. alle möglichen 3-er Kreise existieren in G .

Zusammenfassend ergeben die Gleichungen und Ungleichungen die folgende ILP-Formulierung. Die zulässigen Lösungen von

$$\begin{array}{ll} \max & c^T x \\ & x_{uv} + x_{vu} = 1 \quad \forall (u, v) \in A \\ & x(C) \leq 2 \quad \forall \text{ 3-er Kreise } C \subseteq A \\ & 0 \leq x_{uv} \leq 1 \quad \forall (u, v) \in A \\ & x_{uv} \in \mathbb{Z} \quad \forall (u, v) \in A \end{array}$$

sind genau die charakteristischen Vektoren von azyklischen Turnieren (linearen Ordnungen) in D_n . Die Optimallösung dieses ILPs ist also ein charakteristischer Vektor einer linearen Ordnung mit größtem Gewicht.

Nun haben wir hier die besondere Situation, dass wir den Wert von x_{vu} kennen, sobald wir den Wert von x_{uv} kennen (dies gilt wegen der Gleichungen für alle Knotenpaare $u \neq v$).

Das heißt, wir können auf die Hälfte der Variablen verzichten. Wir behalten also nur noch diejenigen Variablen x_{uv} , für die gilt, dass $u < v$. Was passiert nach der Projektion an den Gleichungen mit den 3-er Kreis Ungleichungen? Wir erhalten zwei verschiedene Versionen von Ungleichungen. Im folgenden sei $u < v < w$:

$$\begin{aligned} x_{uv} + x_{vw} + x_{wu} \leq 2 &\Leftrightarrow x_{uv} + x_{vw} + (1 - x_{uw}) \leq 2 \Leftrightarrow x_{uv} + x_{vw} - x_{uw} \leq 1 \\ x_{vu} + x_{wv} + x_{uw} \leq 2 &\Leftrightarrow (1 - x_{uv}) + (1 - x_{vw}) + x_{uw} \geq 0 \Leftrightarrow x_{uv} + x_{vw} - x_{uw} \geq 0 \end{aligned}$$

Auch die Zielfunktion muß geändert werden. Wir erhalten

$$\bar{c}(u, v) = c(u, v) - c(v, u) \text{ für alle Knotenpaare } (u, v) \text{ mit } u < v.$$

Damit erhalten wir das folgende ILP:

$$\max \bar{c}^T x \quad (3.37)$$

$$x_{uv} + x_{vw} - x_{uw} \geq 0 \quad \forall u < v < w \in V \quad (3.38)$$

$$x_{uv} + x_{vw} - x_{uw} \leq 1 \quad \forall u < v < w \in V \quad (3.39)$$

$$0 \leq x_{uv} \leq 1 \quad \forall (u, v) \in A, u < v \quad (3.40)$$

$$x_{uv} \in \mathbb{Z} \quad \forall (u, v) \in A, u < v \quad (3.41)$$

3.3.1 Geometrische Interpretation

Wir betrachten das LOP-Polytop für den Fall $n = 3$. Der Graph, in dem wir ein maximales azyklisches Turnier suchen, besteht also aus drei Knoten und sechs Kanten. Nach Projektion erhalten wir drei Variablen, die wir folgendermaßen interpretieren:

$$x_{12} = 1 \Leftrightarrow \text{Kante}(1, 2) \text{ ist in der Lösung enthalten, d.h. nicht die Kante } (2, 1)$$

$$x_{13} = 1 \Leftrightarrow \text{Kante}(1, 3) \text{ ist in der Lösung enthalten, d.h. nicht die Kante } (3, 1)$$

$$x_{23} = 1 \Leftrightarrow \text{Kante}(2, 3) \text{ ist in der Lösung enthalten, d.h. nicht die Kante } (3, 2)$$

Die Menge der zulässigen Lösungen läßt sich am leichtesten in der Sprache des *Linear Ordering Problems* bestimmen. Zulässig sind also alle linearen Ordnungen der Menge $\{1, 2, 3\}$. Jeder Permutation läßt sich ein 3-dimensionaler charakteristischer Vektor zuordnen. Die Menge der zulässigen Lösungen sowie die dazugehörigen charakteristischen Vektoren sind also:

Permutation	char. Vektor
$\langle 1, 2, 3 \rangle$	$(1, 1, 1)$
$\langle 2, 1, 3 \rangle$	$(0, 1, 1)$
$\langle 2, 3, 1 \rangle$	$(0, 0, 1)$
$\langle 1, 3, 2 \rangle$	$(1, 1, 0)$
$\langle 3, 1, 2 \rangle$	$(1, 0, 0)$
$\langle 3, 2, 1 \rangle$	$(0, 0, 0)$

Wir befinden uns also im drei-dimensionalen Raum. Jede zulässige Permutation entspricht einem Punkt im drei-dimensionalen Raum. Die dazugehörigen Punkte sind in Abbildung 3.8 dargestellt. Man erhält das assoziierte Polytop, wenn man die konvexe Hülle über die Permutationen nimmt. Dies ist also ein Teil des Würfels im drei-dimensionalen Raum, bei dem zwei Ecken ausgespart sind; nämlich genau diejenigen 0/1-Punkte, die keiner Permutation entsprechen: $(0, 1, 0)$ und $(1, 0, 1)$. Wir nennen dieses Polytop das *relaxierte LOP-Polytop* für $n = 3$.

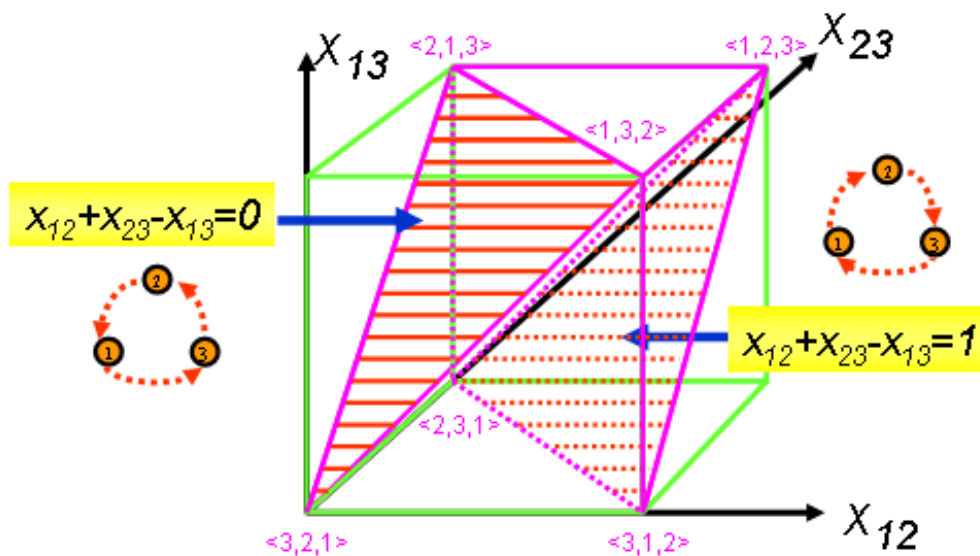


Abbildung 3.8: Das LOP-Polytop für $n=3$

Laut Minkowski und Weyl (s. Abschnitt 3.1.5) wissen wir aber, dass es für das Polytop auch eine Beschreibung in Form von Gleichungen und Ungleichungen gibt. Als Kandidaten hierfür sind generell diejenigen Restriktionen gut geeignet, die in der ILP-Formulierung vorkommen.

Wir betrachten zunächst die 0/1-Schranken: Ungleichungen (3.40). Diese schränken den Lösungsraum innerhalb des Einheitswürfels im 3-dimensionalen Raum ein. Für $n = 3$ gibt es genau zwei verschiedene 3-er Kreis Ungleichungen der Formen (3.38) und (3.39), nämlich:

$$\begin{aligned} x_{12} + x_{23} - x_{13} &\geq 0 \\ x_{12} + x_{23} - x_{13} &\leq 1 \end{aligned}$$

Abbildung 3.8 zeigt die beiden dazugehörigen Hyperebenen, die unter anderem auch diejenigen 0/1-Punkte vom Würfel abschneiden, die keinen Permutationen entsprechen. Dies ergibt genau das relaxierte LOP-Polytop (das wir durch die andere Beschreibung in Form der konvexen Hülle erhalten haben). Nehmen wir die Ganzzahligkeitsbedingungen

hinzu, dann bleiben genau diejenigen 0/1-Punkte übrig, die den Permutationen entsprechen.

Da wir jedoch das ganzzahlige lineare Programm mit Hilfe von linearer Programmierung lösen wollen, lassen wir erst einmal die Ganzzahligkeitsbedingungen weg. Wenn wir nun eine beliebige lineare Zielfunktion über dem relaxierten LOP-Polytop optimieren, so wird das Optimum an einer Ecke angenommen. Alle Ecken des relaxierten LOP-Polytops für $n = 3$ sind aber ganzzahlig. D.h., wir erhalten automatisch ganzzahlige Lösungen, auch wenn wir die Ganzzahligkeitsbedingungen weglassen. Dadurch können wir also optimale Lösungen (für jede gegebene Zielfunktion) des Linear Ordering Problems für $n = 3$ durch lineare Programmierung (in polynomieller Zeit) erhalten.

Auch für $n = 4$ und $n = 5$ macht das Weglassen der Ganzzahligkeitsbedingungen keinen Unterschied: das relaxierte LOP-Polytop besitzt auch in diesen Fällen nur ganzzahlige Ecken.

Leider ist dies nicht generell der Fall. Bereits für $n = 6$ besitzt das relaxierte LOP-Polytop auch nicht-ganzzahlige, sogenannte *fraktionale* Ecken. Wenn wir nun mit Hilfe von linearer Programmierung eine Zielfunktion optimieren, kann es sein, dass das Optimum an einer solchen nicht-ganzzahligen Ecke angenommen wird. In diesem Fall müssen wir zusätzliche Ungleichungen aus der vollständigen Beschreibung des LOP-Polytops (die i.a. unbekannt ist) hinzunehmen.

3.3.2 Zusätzliche Ungleichungen

Die Aufgabe, solche zusätzlichen Ungleichungen zu finden, ist aktuelles Forschungsgebiet der polyedrischen Kombinatorik. Hier soll nur ein Beispiel für eine solche neue Klasse von Ungleichungen gezeigt werden, nämlich die sogenannten *Möbius-Leiter Ungleichungen*. Diese beziehen sich auf einen Untergraphen von G , der folgendermaßen aufgebaut ist: k Kreise der Länge 4 (k ist ungerade) werden so zusammengefügt, dass sie jeweils mit ihren unmittelbaren beiden Nachbarn je genau eine gerichtete Kante gemeinsam haben. Abbildung 3.9 zeigt eine Möbius-Leiter für $k = 5$, die in der Beschreibung des LOP-Polytops für alle $n \geq 10$ auftaucht.

Wir erhalten die dazugehörige *Möbius-Leiter Ungleichung*, indem wir uns wieder fragen, wieviele Bögen mindestens aus dem Graphen entfernt werden müssen, um einen azyklischen Graphen zu erhalten. Zwei Kreise sind jeweils durch einen Bogen verbunden. Das heißt, wenn wir diesen entfernen, dann haben wir bereits zwei Kreise zerstört. Um alle Kreise zu entfernen, müssen also mindestens $(k+1)/2$ Bögen entfernt werden. Dies ergibt die *Möbius-Leiter Ungleichung*:

$$\sum_{x_{uv} \in A} x_{uv} \leq |A| - (k+1)/2$$

Die Möbius-Leiter-Ungleichungen sind nur ein Beispiel von einer Vielzahl sonstiger

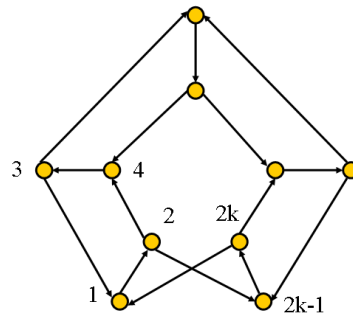


Abbildung 3.9: Eine Möbius-Leiter für $k = 5$

benötigter Ungleichungen. Einen Eindruck über die riesige Anzahl der benötigten Ungleichungen, die das LOP-Polytop beschreiben, erhalten sie in Tabelle 3.3.2.

n	Anzahl der Ungleichungen
3	8
4	20
5	40
6	910
7	87.472
8	>488.602.996

Tabelle 3.1: Anzahl der Ungleichungen des LOP-Polytops für $n \leq 8$

Diese Anzahl steigt rasant an. Und trotzdem ist es möglich, das Linear Ordering Problem für $n = 60$ mit Hilfe von ganzzahligen linearen Programmier-techniken (Schnittebenenverfahren) innerhalb von einer Sekunde beweisbar optimal zu lösen. In der Praxis genügt also bereits ein kleiner Teil der LOP-Beschreibung in Form von Ungleichungen und Gleichungen.

3.4 Exakte Verfahren für (Gemischt-) Ganzzahlige Optimierung

Dieser Abschnitt diskutiert die erfolgreichsten exakten Lösungstechniken für viele NP-schwierige kombinatorische Optimierungsprobleme und (gemischt-) ganzzahlige Optimierungsprobleme: Schnittebenenverfahren und deren Kombination mit Branch-and-Bound Verfahren, nämlich Branch-and-Cut Verfahren. Die Verfahren haben gemeinsam, dass sie (gemischt-) ganzzahlige Optimierungsprobleme mit Hilfe von linearer Programmierung lösen.

Abbildung 3.10 zeigt den Zulässigkeitsbereich eines ganzzahligen linearen Programms (die schwarzen Punkte). Wenn wir nun die Ganzzahligkeitsbedingungen weglassen, erhalten wir die sogenannte *LP-Relaxierung* des ILPs. Diese definiert ein Polyeder (hier ist dies ein Polytop). Die optimale Lösung der LP-Relaxierung wird an einer Ecke des dazugehörigen Polytops angenommen. Im allgemeinen ist diese nicht-ganzzahlig. Um nun zu einer ganzzahligen (also zulässigen Lösung zu kommen), kann man z.B. runden. Allerdings ist es meist nicht trivial, um durch Runden wieder zu einer zulässigen Lösung zu kommen, die alle Ungleichungen erfüllt.

Abbildung 3.10 zeigt, dass selbst wenn das Runden gelingt, die erreichte Lösung im allgemeinen nicht optimal ist. Die folgenden Abschnitte zeigen, wie man lineare Programmierung systematisch zur exakten Lösung (gemischt-) ganzzahliger Programme benutzen kann.

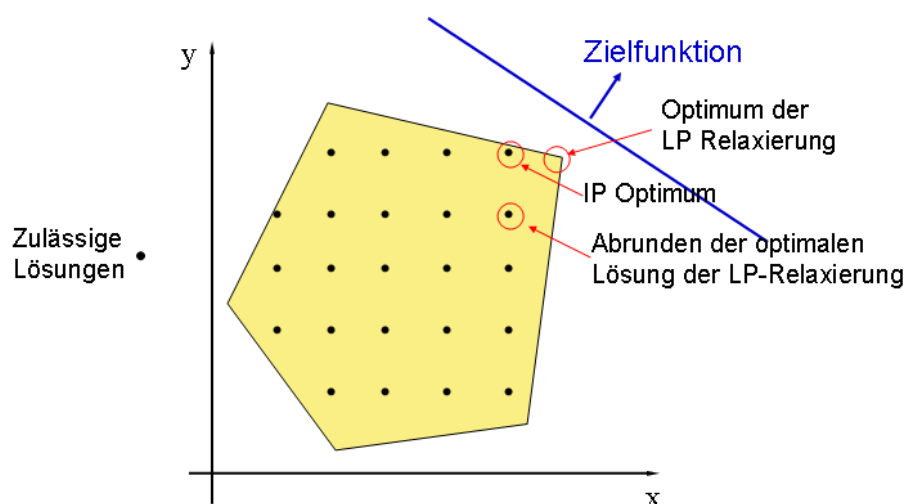


Abbildung 3.10: Ganzzahlige Optimierung

3.4.1 Branch-and-Bound für Gemischt-Ganzzahlige Optimierung

In diesem Abschnitt geben wir eine Kurzeinführung in *Branch-and-Bound*-Verfahren. Mit Branch-and-Bound bezeichnet man eine allgemeine Klasse von Verfahren, die generell für eine weitere Klasse von Problemen einsetzbar sind. Die prinzipielle Idee ist einfach (wenn nicht gar trivial). Die praktische Umsetzung erfordert allerdings weitergehende Überlegungen und effiziente Datenstrukturen.

Die Bestandteile eines Branch-and-Bound-Verfahrens sind:

- ein Schema zur Zerlegung eines Problems in Teilprobleme.
- Verfahren zur Berechnung oberer und unterer Schranken.

Wir betrachten im folgenden Maximierungsprobleme.

Das Vorgehen eines Branch-and-Bound-Verfahrens ist das folgende. Man versucht zunächst das gegebene Problem zu lösen. Hierbei liefern Relaxierungen obere Schranken und zulässige Lösungen untere Schranken. Fallen die Schranken zusammen, so ist das Problem (bewiesenermaßen) optimal gelöst. Gelingt dies nicht, so wird das Problem so in Teilprobleme zerlegt, dass die Gesamtheit der Lösungen der Teilprobleme die Lösung des Ausgangsproblems ergibt. Ein solches Teilproblem wird gelöst durch:

- (a) Bestimmen einer optimalen Lösung oder
- (b) Feststellung der Unzulässigkeit oder
- (c) Bestimmung einer oberen Schranke, die nicht besser als die bisher beste gefundene Lösung ist oder
- (d) Zerlegung in weitere Teilprobleme.

Bei endlicher Lösungsmenge unter *vernünftiger* Aufteilungsstrategie ist klar, dass dieses Verfahren zum Ziel führt.

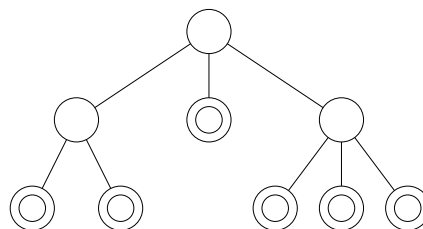


Abbildung 3.11: Illustration eines Branch-and-Bound Baumes

Dem Ablauf des Verfahrens läßt sich in natürlicher Weise der sogenannte *Branch-and-Bound-Baum* zuweisen. Jeder Knoten repräsentiert ein Teilproblem (die Wurzel das Ausgangsproblem) und die direkten Nachfolger eines Knotens entsprechen der Zerlegung des zugehörigen Problems. Die Blätter des Baumes repräsentieren Probleme, die im obigen Sinn (Fall (a) bis (c)) gelöst wurden. Ein Branch-and-Bound-Verfahren ist so organisiert, dass in jedem Schritt folgendes gilt: Eine zulässige Lösung S gehört entweder zu einem bereits gelösten Teilproblem oder ist zulässige Lösung eines noch nicht bearbeiteten Teilproblems.

Abschließend ist zu sagen, dass man typischerweise bei der Lösung schwieriger ganzzahliger oder kombinatorischer Optimierungsprobleme letztendlich um ein Branch-and-Bound-Verfahren nicht herumkommt. Durch fortgeschrittene Techniken (z.B. mittels Schnittebenenverfahren) zur Behandlung der Teilprobleme kann man die Größe des Baumes drastisch reduzieren und erhebliche Rechenzeiterparnis erzielen.

3.4.2 Schnittebenenverfahren

Schnittebenenverfahren, im englischen *Cutting Plane Methods* genannt, gehören zu den erfolgreichsten Verfahren zur Lösung (gemischt-) ganzzahliger Optimierungsprobleme.

Wir haben am Beispiel des LOP-Polytops (s. Abschnitt 3.2) bereits gesehen, dass es unmöglich ist, eine vollständige Beschreibung des assoziierten Polytops durch Ungleichungen zu erhalten. In der Praxis hat sich ausserdem herausgestellt, dass es meist genügt, nur einen kleinen Teil der Ungleichungen zu benützen.

Die Schnittebenenverfahren machen sich dies zunutze. Die grundlegende Idee dahinter ist es, mit nur einer kleinen Teilmenge der Restriktionen zu beginnen, dann das resultierende LP zu lösen. Man erhält einen Lösungspunkt. Nun muß man prüfen, ob dieser bereits für alle Restriktionen zulässig ist. Falls dies so ist, sind wir bereits optimal (denn wir hatten Restriktionen weggelassen, die wir aber anscheinend nicht benötigt hatten, denn sie sind sowieso erfüllt). Andernfalls finden wir eine Restriktion, die an dem jetzigen Lösungspunkt verletzt ist.

Wir betrachten das Beispiel in Abbildung 3.12. (Dies ist fast das gleiche wie in Abbildung 3.10, jedoch mit geänderter Zielfunktion.) Wir nehmen an, dass wir nur die beiden Restriktionen (1) und (2) in unser LP nehmen. Die optimale Lösung liefert den Punkt z_1 . Dieser verletzt jedoch die Restriktion (3), die wir zunächst weggelassen hatten. Im nächsten Schritt nehmen wir also genau diese Restriktion (3) zu unserem LP hinzu und re-optimieren. Durch diese Hinzunahme wird der Punkt z_1 von dem neuen Zulässigkeitsbereich abgeschnitten. Wir erhalten den neuen Lösungspunkt z_2 , der nun alle Restriktionen erfüllt, also Teil des Polytops ist. Die zu (3) gehörige Hyperebene bildet eine *Schnittebene* durch den alten Zulässigkeitsbereich. Deswegen heißt das Verfahren *Schnittebenenverfahren*.

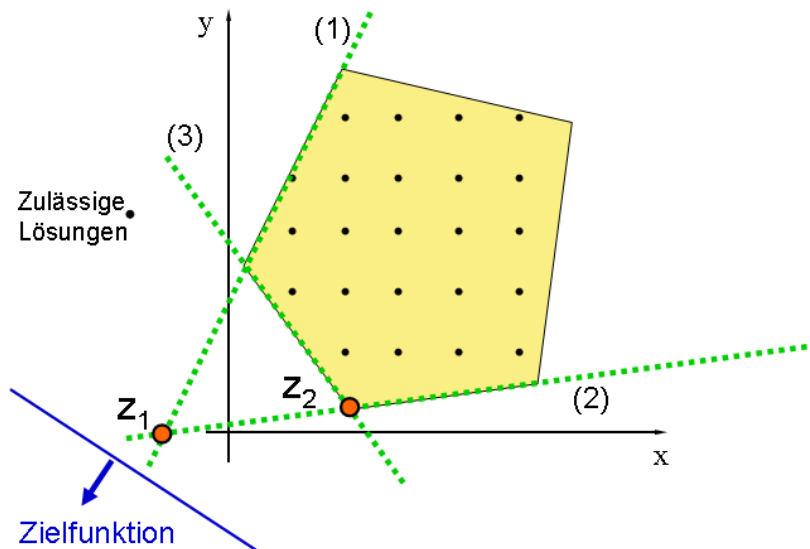


Abbildung 3.12: Schnittebenenverfahren: Ungleichungen werden nacheinander addiert

Idee von Schnittebenenverfahren

0. Starte mit einer Teilmenge der Restriktionen (z.B. nur den trivialen $0 \leq x \leq 1$).
1. Löse LP, sei x^* die gefundene Optimallösung.
2. Entscheide, ob es weggelassene Restriktionen $a^T x \leq a_0$ gibt, so dass $a^T x^* > a_0$.
 Falls nein: STOP (Relaxierung gelöst).
 Falls ja: Bestimme solche, füge sie zu LP hinzu und gehe zu 1.

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

Definition: Das *Separationsproblem* ist das folgende: Gegeben ist ein Punkt $\bar{x} \in \mathbb{R}^E$ und ein Polytop \mathcal{P} . Bestimme, ob $\bar{x} \in \mathcal{P}$, und falls $\bar{x} \notin \mathcal{P}$, finde eine Restriktion $a^T x \leq a_0$, die für alle Punkte $x \in \mathcal{P}$ gültig ist, jedoch für den Punkt \bar{x} verletzt ist, d.h. für alle Punkte $x \in \mathcal{P}$ muß gelten $a^T x \leq a_0$, und für \bar{x} muß gelten $a^T \bar{x} > a_0$.

Beispiel LOP: Können wir das Separationsproblem für das relaxierte LOP-Polytop lösen? Wir nehmen an, dass wir nur mit den Ungleichungen (3.40) beginnen und das dazugehörige LP lösen. Das heißt, wir beginnen mit dem System:

$$\max \bar{c}^T x \tag{3.42}$$

$$0 \leq x_{uv} \leq 1 \quad \forall (u, v) \in A, u < v \tag{3.43}$$

und lösen das Separationsproblem für die Klasse der 3-er Kreis Ungleichungen:

$$0 \leq x_{uv} + x_{vw} - x_{uw} \leq 1 \quad \forall u < v < w \in V \quad (3.44)$$

Optimierung über den Restriktionen (3.43) liefert einen optimalen Punkt \bar{x} . Können wir also in polynomieller Zeit herausfinden, ob der Punkt \bar{x} schon für das relaxierte LP-Polytop zulässig ist, oder falls nicht, können wir dann eine Ungleichung finden, die \bar{x} abschneidet, ohne jedoch einen zulässigen LOP-Punkt abzuschneiden?

Die Antwort ist ja: denn wir können der Reihe nach alle Ungleichungen aus der Menge (3.44) ausprobieren, denn es gibt höchstens $O(n^3)$ 3-er Kreise, also auch höchstens sovielen Ungleichungen dieser Klasse. In jede der $O(n^3)$ Ungleichungen müssen wir einfach nur \bar{x} einsetzen, um zu sehen, ob die Ungleichung am Punkt \bar{x} erfüllt oder verletzt ist. Sobald wir eine verletzte Ungleichung gefunden haben, haben wir das Separationsproblem gelöst. Wir fügen die gefundene Ungleichung (*) zu unserem LP hinzu, und optimieren neu. Der neue optimale Punkt muß vom bisherigen Punkt abweichen, denn die Ungleichung (*) muß ja nun erfüllt sein.

Der folgende Satz zeigt die polynomielle Äquivalenz von Optimierung und Separierung.

SATZ 3.2 (Grötschel, Lovasz, Schrijver 1981)

Wir können eine lineare Zielfunktion über einem Polytop genau dann in polynomieller Zeit lösen, wenn wir das dazugehörige Separationsproblem in polynomieller Zeit lösen können (ohne Beweis).

Beispiel Acyclic Subgraph: Eng verwandt mit dem *Linear Ordering Problem* ist das *Acyclic Subgraph Problem*. Hier ist ein gerichteter Graph $D = (V, A)$ mit Bogengewichten $c_e \in \mathbb{R}^+$ gegeben, und gesucht ist der Teilgraph maximalen Gewichts, der keine gerichteten Kreise enthält. Das *Acyclic Subgraph Problem* ist i.A. NP-schwierig.

Die ILP-Formulierung ist sehr ähnlich wie die Formulierung des LOP-Problems. Allerdings haben wir hier keinen vollständigen Graphen gegeben, d.h., der Ausschluß von 3-er Kreisen ist hier nicht genug. Hier müssen die Ungleichungen tatsächlich alle gerichteten Kreise in G ausschließen. Die Formulierung ist die folgende:

$$\max \sum_{uv \in A} c_{uv} x_{uv} \quad (3.45)$$

$$\sum_{uv \in C} x_{uv} \leq |C| - 1 \quad \forall \text{ gerichteten Kreise } C \text{ in } G \quad (3.46)$$

$$0 \leq x_{uv} \leq 1 \quad \forall (u, v) \in A \quad (3.47)$$

$$x_{uv} \in \mathbb{Z} \quad \forall (u, v) \in A \quad (3.48)$$

Im allgemeinen kann es in einem Graphen exponentiell viele Kreise geben. D.h., die Kreis-Ungleichungen (3.46) sind i.A. exponentiell viele, was zur Folge hat, dass man diese

Ungleichungen nicht von vornherein einem LP-Löser geben kann.

Obwohl das relaxierte LP (also nach Weglassen der Ganzzahligkeitsbedingung) exponentiell viele Ungleichungen besitzt, kann es in polynomieller Zeit gelöst werden. Dies geht mit Hilfe des obigen Satzes, die Äquivalenz von Optimierung und Separierung. Denn das Separationsproblem kann für diese Klasse von Ungleichungen folgendermaßen gelöst werden.

Wir nehmen an, dass wir zunächst das LP lösen, das nur aus den Ungleichungen (3.47) besteht. Wir erreichen den optimalen Punkt des LPs \bar{x} . Wie können wir nun für diesen Punkt und für die Klasse der Kreisungleichungen das Separationsproblem lösen?

Wir schreiben die Ungleichungen zunächst um:

$$\sum_{uv \in C} x_{uv} \leq |C| - 1 \Leftrightarrow |C| - \sum_{uv \in C} x_{uv} \geq 1 \Leftrightarrow \sum_{uv \in C} (1 - x_{uv}) \geq 1$$

Unser Ziel ist es, einen Kreis C zu finden, dessen Ungleichung

$$\sum_{uv \in C} (1 - \bar{x}_{uv}) \geq 1$$

am Punkt \bar{x} verletzt ist, d.h., wir wollen einen Kreis finden, für den gilt

$$\sum_{uv \in C} (1 - \bar{x}_{uv}) < 1.$$

Dies ist gleichbedeutend, in dem Graphen, in dem die Gewichte \bar{x}_{uv} durch neue Gewichte $1 - \bar{x}_{uv}$ ersetzt werden, nach einem Kreis zu suchen, der, wenn man alle Bögen darin aufaddiert, weniger als 1 ergibt.

Eine einfache Beobachtung ist nun: wenn der kürzeste Kreis bezüglich dieser Gewichte die Ungleichung nicht verletzt (d.h. die Summe der Werte ist größer gleich 1), dann existiert auch kein anderer Kreis, der die Ungleichung verletzt. Wenn aber der kürzeste Kreis die Ungleichung verletzt, dann haben wir eine Kreisungleichung gemäß dem Separationsproblem gefunden, die wir im nächsten Schritt zu unserem LP hinzunehmen können. D.h., selbst wenn wir die Suche auf den kürzesten Kreis beschränken, lösen wir auf jeden Fall das Separationsproblem.

Wir suchen also einen kürzesten Kreis in einem gewichteten gerichteten Graphen. Dieses Problem teilen wir in $|A|$ Probleme auf, die jeweils einen kürzesten Kreis suchen, der eine bestimmte Kante $f \in A$ enthält. Dazu führen wir für jede Kante $f \in A$ die folgenden Schritte aus:

- (a) Fixiere die Kante $f = (i, j)$

- (b) Berechne den kürzesten (j, i) -Weg in D bezüglich der neuen Kantenkosten $1 - \bar{x}_{uv}$ für alle Kanten $(u, v) \in A$. Dieser bildet zusammen mit der Kante $f = (i, j)$ einen Kreis, der f enthält.
- (c) Falls die Weglänge W plus das Gewicht $(1 - \bar{x}_{ij})$ der Kante $f = (i, j)$ kleiner als 1 ist, dann haben wir eine verletzte Ungleichung gefunden. Das Separationsproblem ist gelöst.
- (d) Sonst: ist dies der Beweis, dass keine verletzte Kreis-Ungleichung, die f enthält existiert.

Sobald wir eine verletzte Kreisungleichung für eine Kante f gefunden haben, fügen wir diese zu unserem LP hinzu und re-optimieren. Wir erhalten eine andere Lösung, für die wir wiederum das Separationsproblem lösen müssen. Dies machen wir solange, bis wir einen Punkt erreichen, für den keine verletzte Ungleichung gefunden wird. Dann wissen wir, dass wir das assoziierte Optimierungsproblem gelöst haben.

Wir haben also das relaxierte *Acyclic Subgraph* LP in polynomieller Zeit gelöst, obwohl es exponentiell viele Ungleichungen besitzt.

Ein anderes, sehr berühmtes Beispiel, ist das Handlungsreisendenproblem (*Traveling Salesman Problem*, TSP), das wir im folgenden betrachten werden.

3.4.3 Beispiel: Das Handlungsreisendenproblem (TSP)

Gegeben ist ein vollständiger Graph $K_n = (V, E)$ mit Kantengewichten $c \in \mathbb{R}^E$, gesucht ist eine Rundtour (bzw. Tour oder Hamiltonkreis, das sind Kreise, die jeden Knoten genau einmal besuchen) minimalen Gewichts.

Zunächst finden wir eine ILP-Formulierung für das TSP. Die Menge der zulässigen Lösungen ist die Menge aller Touren in K_n . Als Variable wählen wir die Menge der Kanten in K_n . Jeder Tour entspricht also ein 0/1-Vektor $x \in \mathbb{R}^E$ und umgekehrt. Wir assoziieren mit der Variablen x_e einen Wert von 1 genau dann, wenn die Kante e Teil der Tour ist; $x_e = 0$ bedeutet, dass diese Kante $e = (u, v)$ nicht Teil der Tour ist.

Was zeichnet nun Touren aus? Touren besuchen jeden Knoten genau einmal. D.h., die Anzahl derjenigen zu v inzidenten Kanten in einer Tour ist genau 2. Dies führt zu der folgenden Gleichung für jeden Knoten $v \in V$. Dabei ist $\delta(v)$ die Menge aller zu v inzidenten Kanten. $x(\delta(v))$ bezeichnet die Summe aller Werte x_e , deren Kanten sich in $\delta(v)$ befinden.

$$x(\delta(v)) = 2 \quad \text{für alle } v \in V$$

Diese Gleichungen heißen *Grad-Gleichungen*. Die Gleichungen alleine genügen noch nicht, um Touren zu charakterisieren. Eine wichtige Bedingung, die noch fehlt, ist die Eigenschaft

des Zusammenhangs. Wenn man eine beliebige Tour T gegeben hat, und eine Knotenmenge $W \subseteq V$ und alle Kanten betrachtet, die in der Knotenmenge W beginnen und außerhalb W enden, dann müssen davon mindestens zwei Kanten Teil von T sein. (Manche Touren enthalten auch 4 oder mehrere Kanten davon, aber es gibt auch Touren, die nur zwei Kanten enthalten). Dies führt zu den sogenannten *Subtour-Eliminations-Ungleichungen*:

$$x(\delta(W)) \geq 2 \quad \text{für alle } W \subseteq V, W \neq \emptyset, W \neq V$$

Ein Schnitt $\delta(W)$ einer Knotenmenge $W \subseteq V$ bezeichnet alle Kanten, die einen Endknoten in W , und einen Endknoten außerhalb von W besitzen. $x(\delta(W))$ bezeichnet die Summe aller Werte x_e , deren Kanten sich im Schnitt $\delta(W)$ befinden.

Zusammengefaßt entsprechen die zulässigen Lösungen des TSP-Problems genau den Lösungen des Systems:

$$\min c^T x \tag{3.49}$$

$$x(\delta(v)) = 2 \quad \text{für alle } v \in V \tag{3.50}$$

$$x(\delta(W)) \geq 2 \quad \text{für alle } W \subseteq V, W \neq \emptyset, W \neq V \tag{3.51}$$

$$0 \leq x_e \leq 1 \quad \text{für alle } e \in E \tag{3.52}$$

$$x_e \in \{0, 1\} \quad \text{für alle } e \in E \tag{3.53}$$

D.h., die Menge der zulässigen Lösungen dieses ILPs entsprechen genau den charakteristischen Vektoren von Touren. Eine Optimallösung ist charakteristischer Vektor einer kürzesten Tour. Das Weglassen der Ganzzahligkeitsbedingungen ergibt wieder eine LP-Relaxierung, mit deren Hilfe eine Schranke für den Optimalwert berechnet werden kann.

Können wir die LP-Relaxierung lösen? Die Anzahl der Grad-Gleichungen (3.50) ist genau n . Allerdings ist die Anzahl der *Subtour-Eliminations-Ungleichungen* exponentiell (denn es gibt eine Ungleichung für jede Teilmenge W von V). Allerdings können wir das Separationsproblem für die Ungleichungen (3.51) lösen.

Wir beginnen hier mit den Grad-Gleichungen (3.50) und den 0/1-Schranken (3.52) und lösen das erhaltene LP. Die optimale Lösung des LPs sei \bar{x} . Wir berechnen nun in dem Graphen mit den Kantengewichten \bar{x}_e (für alle Kanten $e \in E$) einen minimalen Schnitt. (Dies geht u.a. in polynomieller Zeit mittels des (s, t) -Netzwerkfluß-Algorithmus von Ford/Fulkerson; s. auch Übung).

Dieser gibt uns eine Menge $W \subseteq V$ und den Wert $\bar{x}(\delta(W))$ zurück. Falls $\bar{x}(\delta(W))$ kleiner als 2 ist, dann haben wir eine verletzte Ungleichung gefunden. Andernfalls ($\bar{x}(\delta(W))$ größer gleich 2) kann keine Menge W existieren, deren Schnitt am Punkt \bar{x} kleineren Wert als 2 hat. Damit haben wir das Separationsproblem für die *Subtour-Eliminations-Ungleichungen* gelöst, obwohl es exponentiell viele Ungleichungen sind.

Am Ende (wenn keine verletzte Ungleichung mehr gefunden wird) haben wir die optimale Lösung für unsere LP-Relaxierung erhalten. Und dies in polynomieller Zeit. D.h. wir können die *TSP*-Kurzyklenrelaxierung in polynomieller Zeit lösen, obwohl wir sie nicht in polynomieller Zeit aufschreiben können! Dies ist nur mit Hilfe des Separierungsproblems möglich.

Bemerkungen zum TSP-Polytop

Das TSP-Polytop ist folgendermaßen definiert:

$$STSP(n) = \text{conv}\{\chi^H \mid H \text{ Tour in } K_n\}$$

Nach Minkowski/Weyl existiert auch die folgende Formulierung:

$$STSP(n) = \{x \in \mathbb{R}^{E_n} \mid Ax = b, Dx \leq d\}$$

Würden wir sie kennen, so würde das *TSP* zu einem (riesigen) *LP*. Vollständige Facettialbeschreibungen sind nur für sehr kleine n bekannt:

n	Quelle	Anzahl Facetten
3		0
4		3
5		20
6	Norman [1955]	100
7	Boyd & Cunningham [1991]	3437
8	Christof, Jünger, Reinelt [1991]	194187
9	Christof/Padberg [1996]	42104442
10	Christof/Padberg [1996]	≥ 51043900866

Obwohl die Zahlen bereits für $n = 10$ sehr groß sind, so ist es trotzdem möglich, mit Hilfe von Schnittebenenverfahren (in Kombination mit Branch-And-Bound, s. nächster Abschnitt) sehr große Instanzen des TSPs zu lösen. Das größte beweisbar optimal gelöste TSP besitzt über 20.000 Städte. Das TSP-Polytop ist eines der am besten studierten Polytope. Es sind sehr viele Ungleichungsklassen bekannt, die Facetten des Polytops beschreiben, d.h. für die vollständige Beschreibung des TSP-Polytops notwendig sind. Dies ist auch der Grund, warum man so große TSP-Instanzen exakt lösen kann.

3.4.4 Branch-and-Cut Verfahren

Branch-and-Cut Verfahren verbinden Schnittebenenverfahren mit Branch-and-Bound. Jedoch versucht man hier, die einzelnen Teilprobleme (LP-Relaxierungen) mittels Schnittebenenverfahren zu lösen. Nur wenn man dort nicht mehr weiterkommt (weil man z.B. keine Ungleichungen mehr kennt, die verletzt sein könnten), teilt man die Probleme auf (*Branching*). Dies ist der Fall, falls die Lösung der LP-Relaxierung nicht ganzzahlig ist (also kein

charakteristischer Vektor einer zulässigen Lösung ist). Dann wähle eine nicht-ganzzahlige Variable x_e und generiere zwei neue Teilprobleme:

$P1$ mit zusätzlichen Restriktionen $x_e = 0$

$P2$ mit zusätzlichen Restriktionen $x_e = 1$

Jedes der neuen Teilprobleme versucht man nun wieder neu mit Hilfe des Schnittebenenverfahrens zu lösen. Dies ist der wesentliche Unterschied zu Branch-and-Bound, wo man nur gute Bounds für jedes Teilproblem berechnen möchte, ohne den Anspruch zu haben, diese exakt zu lösen.

Idealerweise gibt es bei Branch-and-Cut Verfahren einen relativ kleinen Branch-and-Bound Baum, während dieser bei Branch-and-Bound typischerweise groß ist.

Abbildung 3.13 zeigt ein Flußdiagramm für Minimierungsprobleme z.B. TSP .

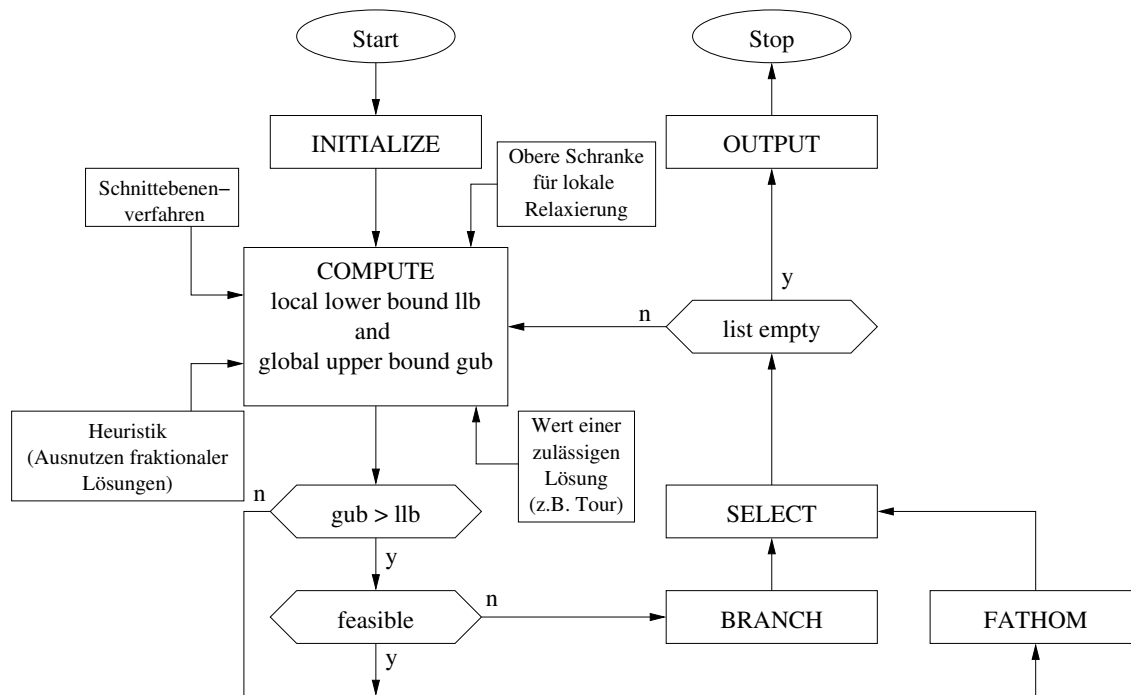


Abbildung 3.13: Ein Flußdiagramm für Minimierungsprobleme

Der Erfolg von Branch and Cut Algorithmen beruht auf

- (a) der Verwendung guter LP -Relaxierungen
- (b) schnellen Separationsalgorithmen für (a)
- (c) zahlreichen strategischen und algorithmischen „Tricks“