Skriptausschnitt zur Vorlesung

Algorithmen und Datenstrukturen

Prof. Dr. Petra Mutzel

Wintersemester 2008/09

Lehrstuhl für Algorithm Engineering
Universität Dortmund

Kapitel 4

Lineare Programmierung und kombinatorische Optimierung

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 4.3 veranschaulicht den Zusammenhang zwischen kombinatorischen und 0/1-ganzzahligen Optimierungsproblemen.

4.1 Einführung

4.1.1 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 $c(F) := \sum_{e \in F} c(e)$, 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 $f(x) = 3x + 2$ 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

$$\begin{aligned} \max\ 2x_1 + 3x_2 & & & \quad (4.1)\\ s.t.\ x_1 + 2x_2 &\leq 3 & & \quad (4.2)\\ 3x_1 - x_2 &\leq 5 & & \quad (4.3)\\ x_1, x_2 &\in \mathbb{N} & & \quad (4.4) \end{aligned}$$

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 (4.4) durch die Bedingung

$$x_1, x_2 \in \{0, 1\}, \qquad (4.5)$$

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.

4.1.2 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 Problemsm die Bedingungen $Ax \leq b$ heißen auch *Restriktionen*. Jeder Vektor $x \in \mathbb{R}^n$, der alle Nebenbedingungen erfüllt, heißt *zulässige Lösung*. Die Menge $P = \{x \in \mathbb{R}^d \mid Ax \leq b\}$ heißt *Zulässigkeitsbereich*.

In Kurzform schreiben wir für ein LP:

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

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. 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 2,5 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:

$$\begin{aligned}
\min\ 3x_1 + 5x_2 & & & \qquad (4.6)\\
2x_1 + x_2 &\geq 3 & & \qquad (4.7)\\
2x_1 + 2x_2 &\geq 5 & & \qquad (4.8)\\
x_1 + 4x_2 &\geq 4 & & \qquad (4.9)\\
x_1 &\geq 0 & & \qquad (4.10)\\
x_2 &\geq 0 & & \qquad (4.11)
\end{aligned}$$

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:

$$110x_1 + 205x_2 + 160x_3 + 160x_4 + 420x_5 + 260x_6 \geq 2000 \tag{4.12}$$
$$4x_1 + 32x_2 + 13x_3 + 8x_4 + 4x_5 + 14x_6 \geq 55 \tag{4.13}$$
$$2x_1 + 12x_2 + 54x_3 + 285x_4 + 22x_5 + 80x_6 \geq 800 \tag{4.14}$$

Die Zielfunktion (Kosten minimieren) ist die folgende:

$$\min\ 3x_1 + 24x_2 + 13x_3 + 9x_4 + 20x_5 + 19x_6 \tag{4.15}$$

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

$$x_i \geq 0 \quad \text{für alle } i = 1, \ldots, 6. \tag{4.16}$$

Obere Schranken (d.h. Schranken der Art $x_i \leq M_i$ für alle $i = 1, \ldots, 6$) 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:

$$0.5x_4 \geq x_1\ . \tag{4.17}$$

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.

- 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:

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

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 4.2.

4.1.3 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

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

heißt *ganzzahliges lineares Optimierungsproblem* oder kurz *ganzzahliges Programm*. Zur Abkürzung schreiben wir machmal *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

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

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

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

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-Programme kommen meistens in folgender Form vor:

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

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 Resourcen 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, $i = 1, \ldots, n$ ein, die die Anzahl der eingepackten Güter der Art i angeben. Das Problem führt zum folgenden ganzzahligen linearen Programm:

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

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 *Algorithmen und Datenstrukturen 1* 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).

4.2 Lineare Programmierung

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

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

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.

4.2.1 Geometrische Interpretation

Wir beginnen die geometrische Interpretation mit dem Beispiel Ölraffinerie (s. Abschnitt 4.1.2).

$$
\begin{aligned}
\min 3x_1 + 5x_2 & & & (4.18)\\
2x_1 + x_2 &\geq 3 & & (4.19)\\
2x_1 + 2x_2 &\geq 5 & & (4.20)\\
x_1 + 4x_2 &\geq 4 & & (4.21)\\
x_1 &\geq 0 & & (4.22)\\
x_2 &\geq 0 & & (4.23)
\end{aligned}
$$

Wir können das LP, das durch die linearen Funktionen (4.18) bis (4.23) 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 (4.19) bis (4.23) 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 des Systems der Ungleichungen (4.19) bis (4.23). Jede dieser Ungleichungen definiert einen Halbraum oberhalb der dazugehörigen Geraden. Der Zulässigkeitsbereich ist genau der Durchschnitt aller dieser Halbräume und ist in Abbildung 4.1 graphisch dargestellt.

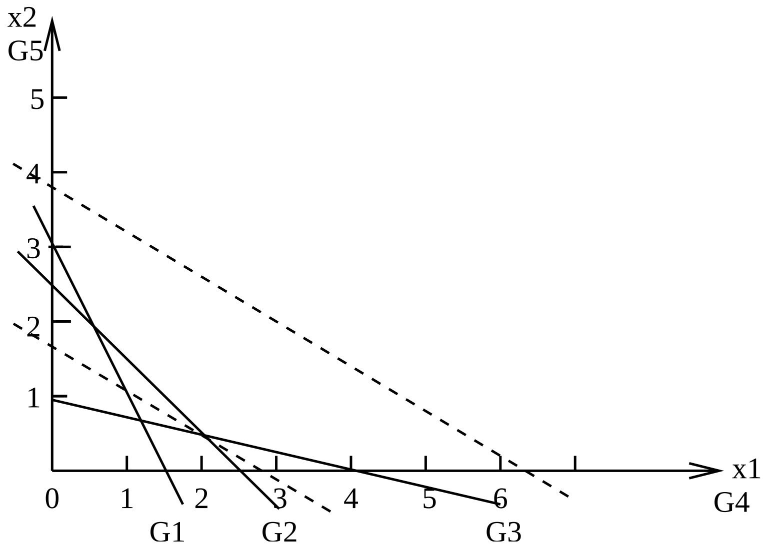


Abbildung 4.1: Graphische Interpretation der Lösungsmenge des Ölproblems

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 4.1 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 (4.19) bis (4.23) 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 4.3.

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 Behauptungen in den folgenden Abschnitten formalisieren.

4.2.2 Grundlagen der Polyedertheorie

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

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

heißt

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

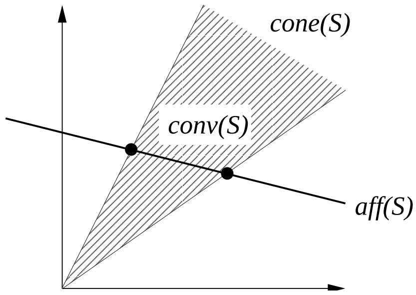


Abbildung 4.2: Illustration der Begriffe cone(S), conv(S) und aff(S)

Linearkombination von $x_1, \dots, x_k$. Gilt zusätzlich

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

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

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

d. h. die Menge aller Vektoren, die als entsprechende Kombination endlich vieler Vektoren aus S dargestellt werden können. Abbildung 4.2 veranschaulicht die verschiedenen Begriffe für die beiden dargestellten Punkte.

Falls

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

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 $i \in \{1, 2, \ldots, k\}$).

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

$$\left.\begin{array}{ll} \textit{Rang} & \mathrm{rang}(S) \\ \textit{affine Rang} & \mathrm{affrang}(S) \end{array}\right\} \text{ von } S$$

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

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

$$0 \in \mathrm{aff}(S) \Rightarrow dim(S) = rang(S)$$
$$0 \notin \mathrm{aff}(S) \Rightarrow dim(S) = rang(S) - 1$$

$S \subseteq R^n$ heißt *volldimensional* falls $dim(S) = n$.
Für $a \in \mathbb{R}^n$, $a \neq 0$, $a_o \in \mathbb{R}$ 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

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

Mit den Formulierungstricks aus Abschnitt 4.1.2 ($Dx \leq d$, $-Dx \leq -d$) 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 $\mathbb{R}^n = \{x \mid 0^T x \leq 1\}$. 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 4.1 [Minkowski 1896, Weyl 1935] *Jedes Polyeder $P \in \mathbb{R}^n$ besitzt eine Darstellung der Form $P = conv(V) + cone(E)$, 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

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

Polytope lassen sich durch $P = \mathrm{conv}(V)$ 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 4.3 veranschaulicht P.

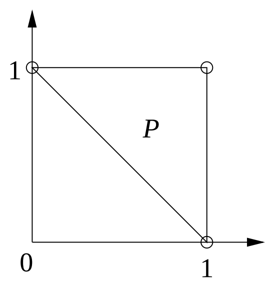


Abbildung 4.3: Ein Polytop des Beispiels zu den verschiedenen Darstellungen

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

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

und F Seitenfläche von P, so gibt es eine Teilmenge der Indizes $I \subseteq \{1, 2, \ldots, k\}$, 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 $F = \mathrm{conv}(R)$.

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

4.2.3 Dualität der Linearen Programmierung

Es ist vorteilhaft, Schranken für lineare Programme angeben zu können. Wir betrachten das folgende LP:

$$\min 4x_1 + 3x_2 \tag{4.24}$$
$$s.t. \quad x_1 + 2x_2 \geq 7 \tag{4.25}$$
$$2x_1 - x_2 \geq 5 \tag{4.26}$$
$$3x_1 + x_2 \geq -2 \tag{4.27}$$

Ein Punkt, der alle diese Nebenbedingungen erfüllt, erfüllt auch die positive Linearkombination, die man erhält, wenn man Ungleichung (4.25) mal 2 nimmt und das Ergebnis zu Ungleichung (4.26) addiert. Dadurch erhält man die Ungleichung

$$4x_1 + 3x_2 \geq 19. \tag{4.28}$$

Die Koeffizienten stimmen genau mit denen des Zielfunktionsvektors c überein. Daraus ergibt sich also eine untere Schranke für die Zielfunktion. Jeder zulässige Punkt kann also auf keinen Fall zu einem größeren Zielfunktionswert als zu 19 ergeben.

Wir probieren eine andere Linearkombination, indem wir Ungleichungen (4.25) und (4.27) addieren. Dies ergibt

$$4x_1 + 3x_2 \geq 5. \tag{4.29}$$

Auch hier stimmen die Koeffizienten mit denen der Zielfunktion überein, wir erhalten also eine neue, jedoch schwächere Schranke für den Zielfunktionswert, nämlich 5.

Wie erhalten wir die beste untere Schranke? Indem wir versuchen, den Zielfunktionsvektor c als positive Linearkombination der Ungleichungen darzustellen. Wir suchen also Skalierungsfaktoren y_i, $i = 1, 2, 3$, der Ungleichungen, die gewisse Bedingungen erfüllen müssen: Multiplizieren wir Ungleichung (4.25) mit y_1, (4.26) mit y_2 und (4.27) mit y_3, dann soll die Summe $y_1 + 2y_2 + 3y_3$ genau dem Wert des ersten Koeffizienten der Zielfunktion (also 4) entsprechen. Analog soll die Summe $2y_1 - y_2 + y_3$ genau den Wert 3 ergeben. Weiterhin sollen die Skalierungsfaktoren y_i nicht-negativ sein. Ferner soll die rechte Seite so groß wie möglich werden. Die rechte Seite kann man durch $7y_1 + 5y_2 - 2y_3$ berechnen. Daraus folgt, dass wir die folgende Aufgabe lösen müssen.

$$\max 7y_1 + 5y_2 - 2y_3 \tag{4.30}$$
$$s.t. \quad y_1 + 2y_2 + 3y_3 = 4 \tag{4.31}$$
$$2y_1 - y_2 + y_3 = 3 \tag{4.32}$$

Allgemein stellt sich also die Aufgabe, c als positive Linearkombination der a_i darzustellen, d. h. suche $y_i \geq 0$ mit $c = \sum_{i=1}^{m} y_i a_i$. Für die Zielfunktion folgt daraus

$$c^T x = (\sum_i y_i a_i)^T x = \sum_i y_i (a_i^T x) \geq \sum_i y_i b_i.$$

Die beste Schranke wird also erreicht, wenn die Funktion $y^T b$ maximiert wird. Die Bestimmung der besten unteren Schranke zu dem LP

$$(P) \qquad \min c^T x, \text{ so dass } Ax \geq b$$

ist also wiederum ein lineares Programm. Dieses heißt das *Duale Problem* und ist definiert als:

$$(D) \qquad \max y^T b, \text{ so dass } y^T A = c^T \text{ und } y \geq 0.$$

Es gilt: Das duale Problem eines dualen Problems zu (P) ist wiederum das primale Problem (P).

Satz 4.2 *(***Schwacher Dualitätssatz***): Sei $\bar{x}$ ein zulässiger Punkt für (P) (das primale Problem), und $\bar{y}$ zulässig für (D) (das duale Problem). Dann gilt:*

$$\bar{y}^T b \leq c^T \bar{x}.$$

Beweis:

$$A\bar{x} \geq b \Leftrightarrow \bar{y}^T (A\bar{x}) \geq \bar{y}^T b \Leftrightarrow (\bar{y}^T A)\bar{x} \geq \bar{y}^T b \Leftrightarrow c^T \bar{x} \geq \bar{y}^T b$$

$\square$

Korollar 4.1 *Ist das primale Problem unbeschränkt, dann ist das duale Problem unlösbar.*

Beweis: Zulässige Lösungen für das duale Problem wären untere Schranken für das primale Problem. Dieses ist jedoch unbeschränkt, also kann das duale Problem keine zulässige Lösung besitzen. $\square$

Der Zielfunktionswert des dualen Problems ist also immer kleiner gleich dem Zielfunktionswert des primalen Problems (wenn das primale Problem die Form $\min\{c^T x \mid Ax \geq b\}$ besitzt). Der starke Dualitätssatz gibt nun die Voraussetzungen an, unter denen diese beiden Werte genau gleich sind (ohne Beweis).

Satz 4.3 *(***Starker Dualitätssatz***): Seien x^* und y^* jeweils zulässige primale und duale Lösungen. Dann gilt:*

$$c^T x^* = (y^*)^T b \Leftrightarrow \text{ beide Lösungen } x^* \text{ und } y^* \text{ sind optimal.}$$

Es gilt also, dass die Zielfunktionswerte für zwei zulässige Lösungen x^* und y^* für (P) bzw. (D) genau dann gleich sind, wenn beide Optimallösungen für (P) bzw. (D) sind. Für jede Lösung eines linearen Programmes kann man somit ihre Optimalität beweisen, indem man auch das duale Programm dazu berechnet. Das am häufigsten in der Praxis zur Lösung von LPs eingesetzte Verfahren, das Simplexverfahren, berechnet implizit eine duale Lösung zusammen mit der primalen Lösung aus, so dass der Optimalitätsbeweis mit einer optimalen Lösung geliefert wird.

4.2.4 Einblick in den Simplex-Algorithmus

Wir betrachten den Simplex-Algorithmus zunächst an dem folgenden Beispiel.

$$\max\ 5x_1 + 4x_2 + 3x_3 \tag{4.33}$$
$$s.t.\quad 2x_1 + 3x_2 + x_3 \leq 5 \tag{4.34}$$
$$4x_1 + x_2 + 2x_3 \leq 11 \tag{4.35}$$
$$3x_1 + 4x_2 + 2x_3 \leq 8 \tag{4.36}$$
$$x_1, x_2, x_3 \geq 0 \tag{4.37}$$

Zunächst werden sogenannte Schlupfvariable x_4, x_5 und x_6 eingeführt, um Gleichungen zu erhalten. Das System sieht nun folgendermaßen aus:

$$\max\ z = 5x_1 + 4x_2 + 3x_3 \tag{4.38}$$
$$x_4 = 5 - 2x_1 - 3x_2 - x_3 \tag{4.39}$$
$$x_5 = 11 - 4x_1 - x_2 - 2x_3 \tag{4.40}$$
$$x_6 = 8 - 3x_1 - 4x_2 - 2x_3 \tag{4.41}$$
$$0 \leq x_1, x_2, x_3, x_4, x_5, x_6 \tag{4.42}$$

Die Strategie der Simplexmethode ist es, an einer Ecke des Lösungspolyeders zu beginnen, von dieser entlang einer Kante zu einer nächsten Ecke zu wandern, die den Zielfunktionswert auf jeden Fall verbessert. D. h. wir wandern von einer Lösung $x_1, x_2, x_3, x_4, x_5, x_6$ zu einer neuen Lösung $\bar{x}_1, \bar{x}_2, \bar{x}_3, \bar{x}_4, \bar{x}_5, \bar{x}_6$, so dass gilt

$$5\bar{x}_1 + 4\bar{x}_2 + 3\bar{x}_3 > 5x_1 + 4x_2 + 3x_3.$$

In diesem Fall ist es einfach, eine erste Lösung zu finden: wir setzen einfach

$$x_1 = x_2 = x_3 = 0.$$

Daraus ergeben sich die Werte für x_4, x_5 und x_6. Unsere erste Lösung ist also:

$$x_1 = 0, x_2 = 0, x_3 = 0, x_4 = 5, x_5 = 11, x_6 = 8 \text{ mit } z = 0.$$

Wie erhalten wir nun die nächste, bessere Lösung? Wir könnten z.B. x_2 und x_3 bei 0 festhalten, und nur x_1 auf 1 erhöhen. Dies ergäbe $z = 5$. Besser wäre es, x_1 auf 2 zu erhöhen, dies ergäbe $z = 10$. Würden wir x_1 auf 3 erhöhen, dann würde dies

$$x_4 = x_5 = x_6 = -1$$

erfordern, was nicht erlaubt ist. Das heißt also, wir dürfen x_1 nicht zu weit erhöhen. Die Frage lautet nun: wie weit dürfen wir x_1 erhöhen, um zulässig zu bleiben (während x_2 und x_3 unverändert bleiben)? Die Ungleichung

$$x_4 = 5 - 2x_1 - 3x_2 - x_3 \geq 0 \quad \text{ergibt} \quad x_1 \leq \frac{5}{2}.$$

Die Bedingung für

$$x_5 \geq 0 \quad \text{ergibt} \quad x_1 \leq \frac{11}{4}$$

und

$$x_6 \geq 0 \quad \text{ergibt} \quad x_1 \leq \frac{8}{3}.$$

Die erste Schranke ist die stärkste. Wir dürfen also x_1 um maximal $\frac{5}{2}$ erhöhen, was wir auch tun.

Dadurch erhalten wir neue Lösungswerte, die wir aus dem Gleichungssystems ablesen können. Dazu muß das System umformuliert werden: Das neue System soll die Werte x_1, x_5 und x_6 mit Hilfe der Werte x_2, x_3 und x_4 ausdrücken. Hier muß insbesondere x_1 von der rechten zur linken Seite wandern, da es seinen Wert von 0 nach 2.5 verändert hat.

Dazu betrachten wir zunächst diejenige Gleichung (4.39), für die die Schranke erreicht wurde. Diese lautet nun nach Umformulierung:

$$x_1 = \frac{5}{2} - \frac{3}{2}x_2 - \frac{1}{2}x_3 - \frac{1}{2}x_4$$

Nun setzen wir in dem restlichen Gleichungssystem alle Stellen, in denen x_1 vorkommt durch diesen Term. Die neue Formulierung sieht dann wie folgt aus (einfache Umformulierung und Substitution):

$$\begin{aligned}
x_1 &= \frac{5}{2} - \frac{3}{2}x_2 - \frac{1}{2}x_3 - \frac{1}{2}x_4 && (4.43)\\
x_5 &= 1 + 5x_2 + 2x_4 && (4.44)\\
x_6 &= \frac{1}{2} + \frac{1}{2}x_2 - \frac{1}{2}x_3 + \frac{3}{2}x_4 && (4.45)\\
z &= \frac{25}{2} - \frac{7}{2}x_2 + \frac{1}{2}x_3 - \frac{5}{2}x_4 && (4.46)
\end{aligned}$$

Nun können wir die neuen Lösungswerte einfach ablesen. Wir haben:

$$x_1 = \frac{5}{2}, x_2 = 0, x_3 = 0, x_4 = 0, x_5 = 1, x_6 = \frac{1}{2} \text{ mit } z = \frac{25}{2}.$$

Dies war die erste Simplex-Iteration. Für die zweite Simplex-Iteration fahren wir genauso fort, und versuchen den Zielfunktionswert durch Erhöhung eine Variablenwertes

zu erhöhen. Wir sehen, dass Erhöhungen von x_2 oder x_4 keine weitere Erhöhung, sondern eine Verminderung der Zielfunktion nach sich ziehen würden. Deswegen erhöhen wir denWert von x_3. Wie weit können wir x_3 erhöhen? Die Antwort können wir aus dem System direkt ablesen, da $x_2 = x_4 = 0$. Aus $x_1 \geq 0$ folgt $x_3 \leq 5$, die Bedingung $x_5 \geq 0$ ergibt keine Einschränkung für x_3, und die Bedingung $x_6 \geq 0$ ergibt $x_3 \leq 1$. Also erhöhen wir x_3 auf 1. Damit erhält die Variable x_6 den Wert 0.
Wie vorher, müssen wir unser System umschreiben, so dass sich alle positiven Variablen (die sogenannten Basisvariablen) auf der linken Seite befinden. Offensichtlich wandert bei jedem Austausch eine neue Variable, die vorher den Wert 0 hatte, in die Basis, während eine Variable, die vorher ungleich 0 war, den Wert 0 erhält, also zur Nichtbasisvariablen wird. Basisvariablen sind jetzt x_3, x_1, x_5, Nichtbasisvariablen sind x_2, x_4, x_6. Genauso wie vorher, betrachten wir die Gleichung, die zu der Variablen gehört, die aus der Basis ging, nämlich Gleichung (4.45). Diese lautet:

$$x_3 = 1 + x_2 + 3x_4 - 2x_6$$

Wir ersetzen jedes Vorkommen von x_3 durch diesen Term und erhalten somit:

$$x_1 = 2 - 2x_2 - 2x_4 + x_6 \tag{4.47}$$
$$x_5 = 1 + 5x_2 + 2x_4 \tag{4.48}$$
$$x_3 = 1 + x_2 + 3x_4 - 2x_6 \tag{4.49}$$
$$z = 13 - 3x_2 - x_4 - x_6 \tag{4.50}$$

Nun können wir die Werte der Lösung wieder aus dem System (das sogenannte *Simnplex-Tableau*) ablesen. Unsere neue Lösung sieht folgendermaßen aus:

$$x_1 = 2, x_2 = 0, x_3 = 1, x_4 = 0, x_5 = 1, x_6 = 0 \text{ mit } z = 13.$$

Nun wollen wir wieder eine Variable erhöhen, die zu einer Erhöhung der Zielfunktion führt. Allerdings führt nun jede Erhöhung einer Variablen zu einer Verminderung des aktuellen Zielfunktionswertes. Die bisher erreichte Lösung mit Zielfunktionswert 13 ist optimal.

Wie können wir überprüfen, ob die gefundene Lösung auch tatsächlich optimal ist? Die Gleichung (4.50) sagt, dass der Zielfunktionswert höchstens 13 sein kann, da wir wissen, dass x_2, x_4 und x_6 nicht-negativ sein müssen. Dies beweist die Optimalität.

Generell geht der Simplexalgorithmus wie beschrieben vor. Er beginnt mit einer zulässigen Lösung, und wandert von einer Lösung zu einer anderen Lösung, mit besserem bzw. mindest so gutem Zielfunktionswert. Die Lösungen heißen *Basislösungen*, da sie einer Ecke des Polyeders entsprechen.

Beispiel 2: Wir illustrieren anhand dieses Beispiels die Schritte eines Verlaufs des Simplexalgorithmus geometrisch anhand von Abbildung 4.4. Wir betrachten das folgende

Lineare Programm:

$$\begin{aligned}
\max \quad & 3x_1 + 2x_2 + 2x_3 && && (4.51)\\
s.t. \quad & x_1 + x_3 && \leq 8 && (4.52)\\
& x_1 + x_2 && \leq 7 && (4.53)\\
& x_1 + 2x_2 && \leq 12 && (4.54)\\
& x_1, x_2, x_3 && \geq 0 && (4.55)
\end{aligned}$$

Der Simplexalgorithmus beginnt mit dem zulässigen Punkt $(x_1, x_2, x_3) = (0, 0, 0)$. In der Basis befinden sich die Schlupfvariablen x_4, x_5 und x_6, die Nicht-Basis-Variablen sind x_1, x_2 und x_3. Das erste Simplex-Tableau sieht folgendermaßen aus:

$$\begin{aligned}
x_4 &= 8 - x_1 - x_3 && (4.56)\\
x_5 &= 7 - x_1 - x_2 && (4.57)\\
x_6 &= 12 - x_1 - 2x_2 && (4.58)\\
z &= 3x_1 + 2x_2 + 2x_3 && (4.59)
\end{aligned}$$

In der ersten Simplexiteration wird der Wert von x_1 erhöht. Dies geht bis zum Wert $x_1 = 7$. Nun findet ein Basiswechsel statt: x_1 geht in die Basis, x_5 geht hinaus (denn die Ungleichung, die zur Schlupfvariablen x_5 gehört, ist nun *tight*, d.h., sie wird mit Gleichheit erfüllt, damit wird x_5 zu Null). Das Simplex-Tableau wird nun so umgeschrieben, dass die Basisvariablen alle auf der linken Seite des Gleichheitszeichens stehen, und die Nicht-Basis-Variablen rechts. Wir beginnen mit der Ungleichung, die derjenigen Variablen entspricht, die aus der Basis geht, nämlich Ungleichung (4.57):

$$x_1 = 7 - x_2 - x_5 \qquad (4.60)$$

Nun wird im Tableau jedes Auftauchen von x_1 durch diesen Term ersetzt. Damit erhalten wir:

$$\begin{aligned}
x_4 &= 8 - (7 - x_2 - x_5) - x_3 = 1 + x_2 + x_5 - x_3 && (4.61)\\
x_1 &= 7 - x_2 - x_5 && (4.62)\\
x_6 &= 12 - (7 - x_2 - x_5) - 2x_2 = 5 - x_2 + x_5 && (4.63)\\
z &= 3(7 - x_2 - x_5) + 2x_2 + 2x_3 = 21 - x_2 + 2x_3 - 3x_5 && (4.64)
\end{aligned}$$

Wir können also nun den momentanen Lösungspunkt vom Tableau ablesen. Er ist

$$x_1 = 7, x_2 = 0, x_3 = 0, x_4 = 1, x_5 = 0, x_6 = 5$$

Bezogen auf den drei-dimensionale Raum unserer ursprünglichen Variablen (x_1, x_2, x_3) ist dies der Punkt $(7, 0, 0)$ mit Lösungswert 21 (s. Abbildung 4.4).

In der zweiten Simplexiteration wird der Wert von x_3 erhöht. Dies geht gut, bis $x_3 = 1$. Dann wird nämlich Ungleichung (4.61) *tight*. Das heißt, x_3 geht in die Basis, x_4 geht

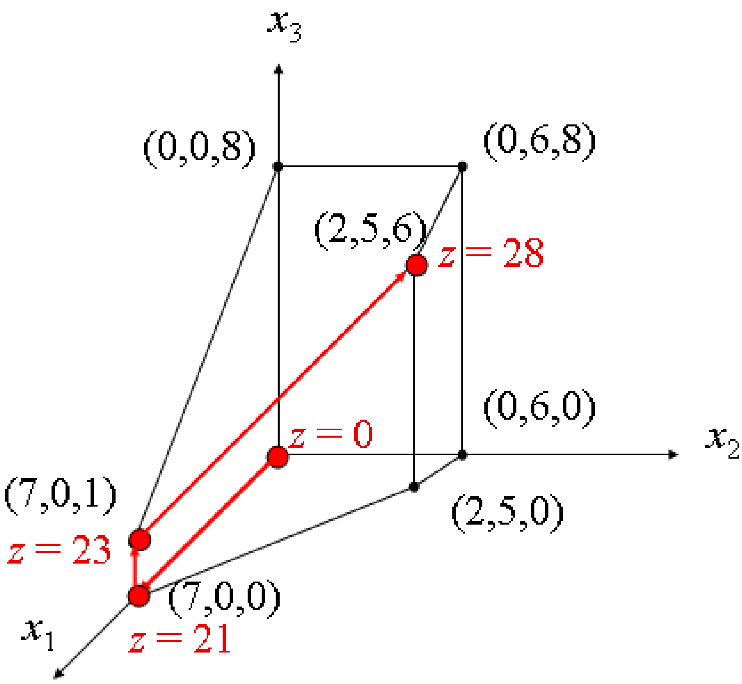


Abbildung 4.4: Illustration der Schritte des Simplexalgorithmus

aus der Basis. Wir schreiben das Tableau wieder um. Dazu formen wir zunächst die zu x_4 gehörige Ungleichung (4.61) um:

$$x_3 = 1 + x_2 + x_5 - x_4 \tag{4.65}$$

Das neue Tableau sieht folgendermassen aus:

$$x_3 = 1 + x_2 + x_5 - x_4 \tag{4.66}$$
$$x_1 = 7 - x_2 - x_5 \tag{4.67}$$
$$x_6 = 5 - x_2 + x_5 \tag{4.68}$$
$$z = 21 - x_2 + 2(1 + x_2 + x_5 - x_4) - 3x_5 = 23 + x_2 - 2x_4 - x_5 \tag{4.69}$$

Daraus können die Werte der Basisvariablen abgelesen werden: $x_1 = 7$ (wie vorher), $x_2 = 0$ (weil Nicht-Basis-Variable) und $x_3 = 1$. Der Zielfunktionswert ist 23. Dieser Schritt ist auch in Abbildung 4.4 veranschaulicht.

Die dritte Simplexiteration erhöht den Wert von x_2 auf 5. Dadurch wandert x_2 in die Basis, und x_6 geht hinaus. Die dazugehörige Gleichung (4.68) lautet:

$$x_2 = 5 + x_5 - x_6 \tag{4.70}$$

Das neue Tableau sieht folgendermassen aus:

$$x_3 = 1 + (5 + x_5 - x_6) + x_5 - x_4 = 6 - x_4 + 2x_5 - x_6 \quad (4.71)$$
$$x_1 = 7 - (5 + x_5 - x_6) - x_5 = 2 - 2x_5 + x_6 \quad (4.72)$$
$$x_2 = 5 + x_5 - x_6 \quad (4.73)$$
$$z = 23 + (5 + x_5 - x_6) - 2x_4 - x_5 = 28 - 2x_4 - x_6 \quad (4.74)$$

Daraus ergeben sich die neuen Werte der Basisvariablen: Nun ist der Wert von $x_1 = 2$, der Wert von $x_2 = 5$, und der Wert von $x_3 = 6$. Hier sieht man also, dass sich mehr als ein Wert der Variablen in einem Simplexschritt ändern kann. Der Zielfunktionswert ist nun 28 (s. auch Veranschaulichung in Abbildung 4.4).

Wir sehen, dass keine Erhöhung einer Variablen zu einer weiteren Erhöhung der Zielfunktion führen kann. Deswegen haben wir die optimale Lösung erreicht. Das Polytop sowie die einzelnen Schritte des Simplex-Algorithmus sind graphisch in der Abbildung 4.4 visualisiert.

Es gilt: Falls es eine zulässige Lösung des betrachteten LPs gibt, dann existiert auch eine Basis B und eine zugehörige Basislösung $\bar{x}$, die ebenfalls zulässig ist. D. h. es gelten $A\bar{x} \geq b$ und $A_B\bar{x} = b_B$. Dabei ist A_B die Teilmatrix, die diejenigen Zeilen der Matrix A enthält, die Nicht-Null sind. Eine Teilmenge B der Zeilenindizes heißt Basis, wenn A_B invertierbar ist. Der Simplexalgorithmus wechselt in jedem Schritt jeweils genau eine Basisvariable mit einer Nichtbasisvariablen aus.

Bemerkungen zum Simplexalgorithmus

Abschnitt 4.2.4 dient nur dazu, dass Sie einen groben Überblick über das Simplexverfahren bekommen. Es bleiben noch viele Fragen offen:

1. Wie erhält man eine erste Basislösung und damit ein gültiges Anfangs-Simplex-Tableau.

2. Welche Strategie verwendet man bei der Wahl der eintretenden Variablen in die Basis?

3. Welche Strategie verwendet man bei der Wahl der austretenden Variablen?

Diese Strategien sind wichtig, denn sonst könnte es sein, dass der Algorithmus immer wieder die gleiche Menge von Variablen in die Basis hinein- und kurze Zeit später wieder aus der Basis entfernt. Man sagt auch: Der Algorithmus *kreiselt.* Durch geschickte Strategien kann man dies verhindern. 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, 1983.

4.3 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

$$\chi_e^F = 1 \quad \Leftrightarrow \quad e \in F \tag{4.75}$$
$$\chi_e^F = 0 \quad \Leftrightarrow \quad e \notin F \tag{4.76}$$

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

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

gegebenen, so setzen wir:

$$E := \{1, 2, \ldots, n\} \tag{4.77}$$
$$I := \{F_x \subseteq E \mid x \in \{0,1\}^E, Ax \le b\} \tag{4.78}$$
$$c(F_x) := c^T \chi^{F_x}. \tag{4.79}$$

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 4.2). Fassen wir die Funktion $c : E \to \mathbb{R}$ 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. 4.5). 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 4.5 zeigt den dazugehörigen Graphen sowie die Menge aller möglichen Lösungen dieses MST-Problems.

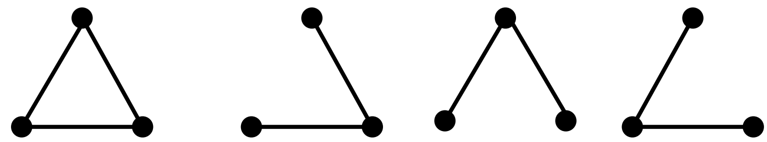


Abbildung 4.5: Graph K_3 und Menge aller zulässigen Spannbäume auf drei Punkten

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

$$x_1 + x_2 + x_3 = 2. \tag{4.80}$$

Und diese zusammen mit den Schranken

$$x_j \geq 0 \text{ und } x_j \leq 1 \text{ für alle } j = 1, 2, 3 \tag{4.81}$$

und den Ganzzahligkeitsbedingungen

$$x_j \in \mathbb{Z} \text{ für alle } j = 1, 2, 3 \tag{4.82}$$

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

$$\begin{aligned}
\min \quad d_1x_1 + d_2x_2 + d_3x_3 & & (4.83)\\
x_1 + x_2 + x_3 = 2 & & (4.84)\\
x_j \geq 0 & \quad \text{für } j = 1, 2, 3 & (4.85)\\
x_j \leq 1 & \quad \text{für } j = 1, 2, 3 & (4.86)
\end{aligned}$$

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 4.6.

Der Lösungsraum des linearen Programms (ohne Berücksichtigung der Ganzzahligkeitsbedingungen) ist die in Abbildung 4.6 markierte Fläche. Die optimale Lösung (über einer 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.

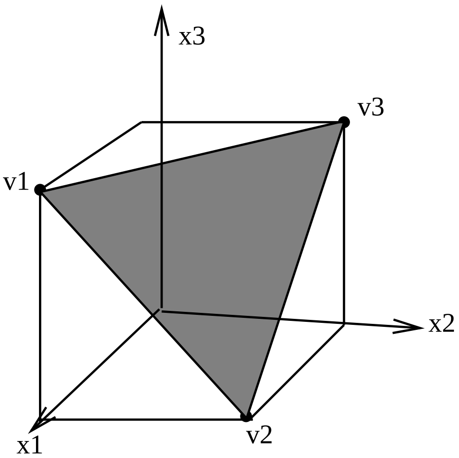


Abbildung 4.6: Der Zulässigkeitsbereich des MST-Problems auf K_3

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

Beispiel 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 4.7 zeigt als Beispiel den vollständigen gerichteten Digraphen D_4 und ein Turnier.

Abbildung 4.7: Vollständiger Digraph 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 4.8 zeigt ein azyklisches Turnier und die dazugehörige lineare Ordnung.

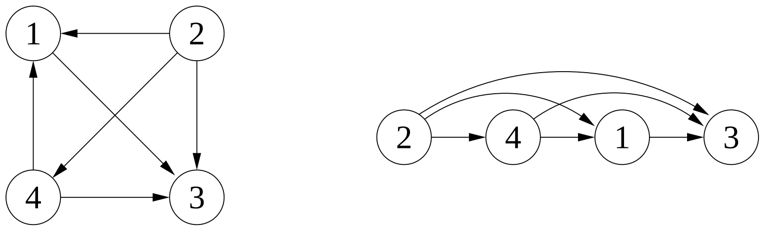


Abbildung 4.8: Ein azyklisches Turnier und die dazugehörige lineare Ordnung

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

$$x_{uv} + x_{vu} = 1 \text{ für alle Knotenpaare } u, v \in V. \tag{4.87}$$

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 $\sum_{(u,v)\in C} x_{uv}$.

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

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

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

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

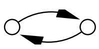
Dies ergibt wegen $x_{31} + x_{13} = 1$ und $x_{41} + x_{14} = 1$ schließlich

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

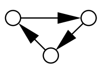
Genauso kann man die Ungleichung eines beliebigen anderen Kreises durch die 3-er Kreise erzeugen. Dies gilt in diesem speziellen Fall, weil der gegebene Graph vollständig und gerichtet ist, d.h. für jedes mögliche gerichtete Knotenpaar (u, v) existiert eine Kante in G, d.h. alle möglichen 3-er Kreise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

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

$$
\max \quad c^T x
$$



$$
x_{uv} + x_{vu} \quad = 1 \quad \forall\, (u, v) \in A
$$



$$
\begin{array}{rll}
x(C) & \leq 2 & \forall \text{ 3-er Kreise } C \subseteq A \\
0 \leq x_{uv} & \leq 1 & \forall\, (u, v) \in A \\
x_{uv} & \in \mathbb{Z} & \forall\, (u, v) \in A
\end{array}
$$

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 $u < v < w$:

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

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:

$$\max\ \bar{c}^T x \tag{4.88}$$
$$x_{uv} + x_{vw} - x_{uw} \geq 0 \quad \forall\, u < v < w \in V \tag{4.89}$$
$$x_{uv} + x_{vw} - x_{uw} \leq 1 \quad \forall\, u < v < w \in V \tag{4.90}$$
$$0 \leq x_{uv} \leq 1 \quad \forall\, (u,v) \in A, u < v \tag{4.91}$$
$$x_{uv} \in \mathbb{Z} \quad \forall\, (u,v) \in A, u < v \tag{4.92}$$

4.3.1 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:

$x_{12} = 1 \;\Leftrightarrow\;$ Kante$(1,2)$ ist in der Lösung enthalten, d.h. nicht die Kante (2,1)
$x_{13} = 1 \;\Leftrightarrow\;$ Kante$(1,3)$ ist in der Lösung enthalten, d.h. nicht die Kante (3,1)
$x_{23} = 1 \;\Leftrightarrow\;$ Kante$(2,3)$ 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
$< 1,2,3 >$	$(1,1,1)$
$< 2,1,3 >$	$(0,1,1)$
$< 2,3,1 >$	$(0,0,1)$
$< 1,3,2 >$	$(1,1,0)$
$< 3,1,2 >$	$(1,0,0)$
$< 3,2,1 >$	$(0,0,0)$

Wir befinden uns also im drei-dimensionalen Raum. Jede zulässige Permutation entspricht einem Punkt im drei-dimensionalen Raum. Die dazugehörigen Punkte sind in Abbildung 4.9 dargestellt. Man erhält das assoziierte Polytop, wenn man die konvexe Hülle über die Permutationen nimmt. Dies ist also ein Teil des Würfels im drei-dimensionaln 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 $n = 3$.

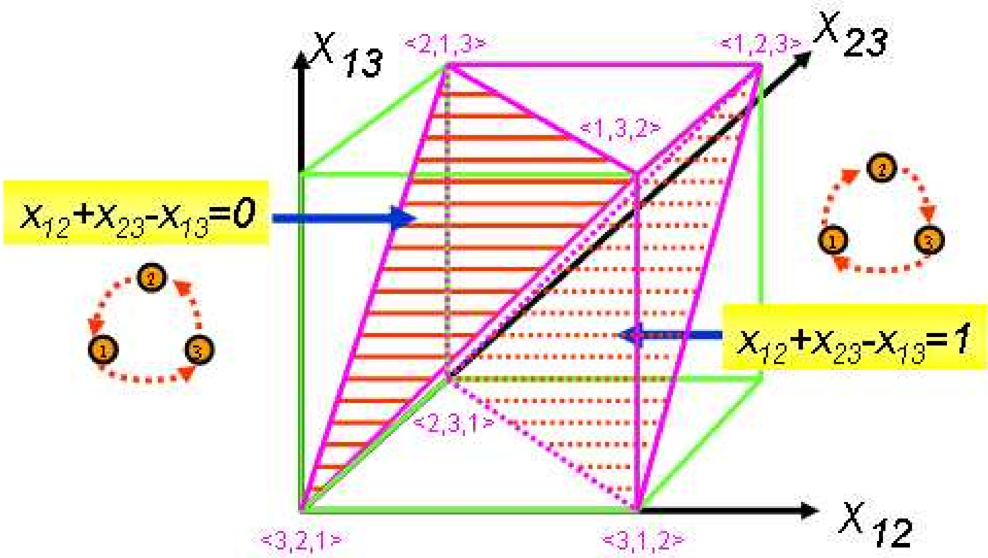


Abbildung 4.9: Das LOP-Polytop für n=3

Laut Minkowski und Weyl (s. Abschnitt 4.2.2) 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 (4.91). 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 (4.89) und (4.90), nämlich:

$$x_{12} + x_{23} - x_{13} \geq 0$$
$$x_{12} + x_{23} - x_{13} \leq 1$$

Abbildung 4.9 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.

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 4.10 zeigt eine Möbius-Leiter für $k = 5$, die in der Beschreibung des LOP-Polytops für alle $n \geq 10$ auftaucht.

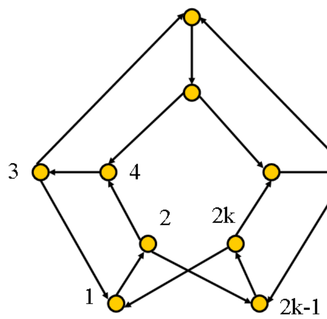


Abbildung 4.10: Eine Möbius-Leiter für $k = 5$

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 benötigter Ungleichungen. Einen Eindruck über die riesige Anzahl der benötigten Ungleichungen, die das LOP-Polytop beschreiben, erhalten sie in Tabelle 4.3.1.

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

Tabelle 4.1: 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 Programmiertechniken (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.

4.4 Weiterführende Literatur

- Eine sehr schöne Einführung in die Lineare Programmierung und in den Simplex-Algorithmus ist: V. Chvatal: “Linear Programming”, W.H. Freeman and Company, New York, 1983

- Ein gutes Buch zur algorithmischen Optimierung ist: C. H. Papadimitriou und K. Steiglitz: “Combinatorial Optimization: Algorithms and Complexity”, Prentice-Hall, Inc., Englewood Cliffs, New Jersey, 1982

- Ein Buch, das eher die mathematischen Aspekte behandelt, ist: W.J. Cook, W.H. Cunningham, W.R. Pulleyblank und A. Schrijver: “Combinatorial Optimization”, John Wiley & Sons, Chichester, 1998