Einführung in die Optimierung
Sommersemester 2005

Anita Schöbel

9. Juli 2010

Vorwort

Das vorliegende Vorlesungsskript entstand aufgrund der Notizen der von mir im Sommersemester 2005 gehaltenen Vorlesung *Optimierung*. Die Vorlesung versucht, einen Überblick über Ergebnisse und Techniken im Gebiet der mathematischen Optimierung zu geben. Der Schwerpunkt liegt dabei auf der *linearen Optimierung* (es wird das Simplex-Verfahren ausführlich besprochen und es wird auf Dualität und Innere-Punkte Verfahren eingegangen). Anschließend wird kurz in die *Ganzzahlige Optimierung* eingeführt und in diesem Kontext besonders das Netzwerkflussproblem untersucht. Es folgt ein Überblick über einige Methoden der nichtlinearen Optimierung. Im letzten Kapitel wird das Gebiet der *Standortplanung* vorgestellt. Die Untersuchung einiger Standortprobleme bietet sich hervorragend als Anwendungsbeispiel einer Optimierungsvorlesung an, da zu ihrer Lösung sowohl Ergebnisse aus der linearen Optimierung als auch aus der nichtlinearen Optimierung verwendet werden.

Die Vorlesung wendet sich vorrangig an Mathematik - und Informatikstudierende, aber natürlich sind auch interessierte Studierende aus anderen Disziplinen willkommen! Das Skript wäre in der vorliegenden Form nicht entstanden, hätte Gert Lube es nicht in seiner Vorlesung verwendet und eine Fehlersammlung angelegt: Herzlichen Dank dafür an ihn und seine Studierenden! Außerdem möchte ich mich bei Keyu Wang für die vielen Zeichnungen bedanken und bei Horst Hamacher, bei dem ich so viel über Optimierung gelernt habe.

Göttingen, im Oktober 2007
Anita Schöbel

Inhaltsverzeichnis

Kapitel 1

Was ist Optimierung?

Viele Anwendungsprobleme in Industrie, Wirtschaft und sogar im täglichen Leben sind Optimierungsprobleme. Beispielsweise gehören dazu

- das Erstellen von Absatzprogrammen für eine Firma,
- die Reduktion von Fluglärm,
- aerodynamische Oberflächengestaltung,
- Probleme der Standortplanung z.B. von Schulen, Geschäften, Lagerhäusern,
- Tourenoptimierung,
- oder relativ allgemein Gewinnmaximierung in der Wirtschaft.

Die Lösung eines Anwendungsproblemes kann man sich wie in Abb. 1.1 veranschaulicht vorstellen.
Bei der *Modellierung* müssen die folgenden drei Punkte abgearbeitet werden:

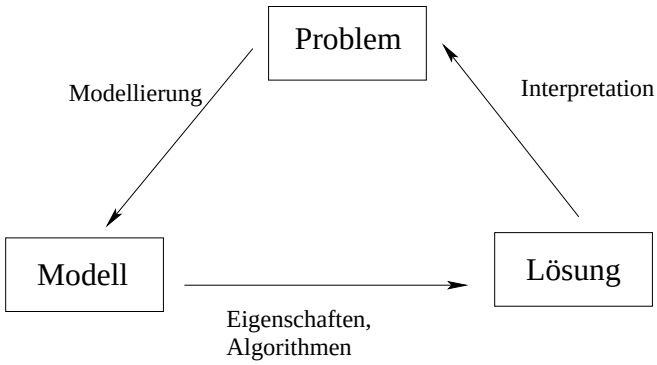


Abbildung 1.1: Lösen eines Anwendungsproblemes

1. Zunächst sollte man sich Gedanken machen, was genau entschieden werden kann. Daraus entstehen die *Variablen*, die typischerweise in einem Vektor $x \in \mathbb{R}^n$ zusammengefasst werden. Jede Belegung des Vektors x ist dann eine *Lösung* des Problems.

2. Als nächstes ist zu klären, welche Bedingungen unbedingt beachtet werden müssen. Daraus werden *Nebenbedingungen* in Form von Gleichungen und Ungleichungen formuliert, die man anhand der Werte für die im ersten Punkt definierten Variablen überprüfen muss. Sind die Nebenbedingungen für eine Lösung x erfüllt, so nennt man x eine *zulässige Lösung*.

3. Schließlich muss man sich Gedanken über eine Bewertung verschiedener zulässiger Lösungen machen, so dass man sie vergleichen und die beste auswählen kann. So ein Optimierungskriterium wird auch *Zielfunktion* (*objective*) genannt.

Das Ergebnis dieses Prozesses ist ein *mathematisches Modell* oder *mathematisches Programm* wie es gleich in Definition 1.1 eingeführt wird.

Obwohl das Erzeugen eines mathematischen Modells aus einem praktischen Problem sehr wichtig ist, liegt der Schwerpunkt dieses Textes auf der Entwicklung geeigneter Methoden und Verfahren zur Lösung schon bestehender mathematischer Modelle. Mit der Modellierung selbst werden wir uns daher nur in diesem Abschnitt beschäftigen.

Definition 1.1 *Ein* **Optimierungsproblem** *(oder "mathematisches Programm") ist gegeben durch*

$$\begin{array}{lrcll} (\mathbf{Opt}) & \min f(x) & & & \\ & \text{s.d.} \quad g_i(x) & \leq & 0 & \forall i \in I_1 \\ & g_i(x) & = & 0 & \forall i \in I_2 \\ & x \in \mathcal{B} & & & \end{array}$$

wobei

- $I = I_1 \cup I_2$ *eine endliche Indexmenge mit* $I_1 \cap I_2 = \emptyset$
- $\mathcal{B} \subseteq \mathbb{R}^n$
- *und* $f, g_i : \mathcal{B} \to \mathbb{R}$ *für alle* $i \in I$.

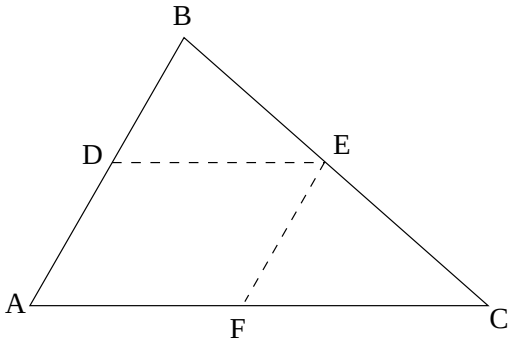
Bemerkungen:

- Das Minimum muss im allgemeinen nicht existieren, formal müsste man daher eigentlich „inf“ schreiben.

- Es gibt auch Anwendungen, bei denen $\mathcal{B} \subseteq \mathbb{R}^n$ nicht gilt (z.B. Bestimmung einer Funktion in der Approximationstheorie) oder bei denen I nicht endlich ist (semi-infinite Programme).

Im folgenden soll die Modellierung und Lösung von Optimierungsproblemen an drei einfachen Beispielen verdeutlicht werden, die alle dem Skript [Wer03] entnommen sind. Das erste Beispiel, das wir uns ansehen wird oft als „ältestes" Optimierungsproblem bezeichnet und findet sich in Euklids Buch der Elemente.

Beispiel 1.1 *Euklid's Elemente, Buch VI, Theorem 27*
Sei ein Dreieck ΔABC gegeben. Finde einen Punkt E auf $\overline{BC}$ so, dass das entstehende Parallelogramm ADEF mit D auf $\overline{AB}$ und F auf $\overline{AC}$ maximalen Flächeninhalt hat.

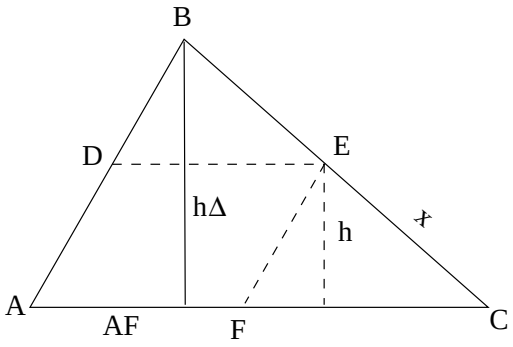


Wie modelliert man nun so ein Problem?

Dazu muss man zunächst überlegen, welche Entscheidungsfreiheiten man hat. In der Sprache der Optimierung heißt das, dass zunächst die Variablen festgelegt werden müssen. Im Fall von Euklids Problem können wir die Lage des Punktes E (auf der Strecke zwischen C und B) festlegen. Als Variable kann man also die Entfernung zwischen E und C wählen. Wir bezeichnen diese Entfernung nun mit x.

Als nächstes bestimmen wir die Zielfunktion. Das ist die Fläche des entstehenden Parallelogrammes ADEF. Damit ist uns aber noch nicht viel geholfen. Was wir brauchen, ist eine Funktion f, die in Abhängigkeit unserer Variablen x (also abhängig von der Lage von E) die Fläche des Parallelogrammes ADEF angibt.

Um diese zu bestimmen, bezeichne BC die Länge von $\overline{BC}$, AC die Länge von $\overline{AC}$, AF die Länge von $\overline{AF}$, h_Δ die Höhe des Δ(A,B,C) und h die Höhe des Δ(F,E,C).



Wir wenden den Strahlensatz *an und erhalten*

$$\begin{aligned}
& \frac{x}{BC} = \frac{h}{h_\Delta} = \frac{AC - AF}{AC} \\
\Rightarrow \quad & h = \frac{x \cdot h_\Delta}{BC}, AF = AC \cdot (1 - \frac{x}{BC}) \\
\Rightarrow \quad & f(x) = h \cdot AF = \frac{x}{BC} \cdot h_\Delta \cdot AC \cdot (1 - \frac{x}{BC}) \\
= \quad & Fläche \;\; von \;\; \Delta(A,B,C) \cdot 2 \cdot (1 - \frac{x}{BC}) \cdot \frac{x}{BC}.
\end{aligned}$$

Dementsprechend können wir nun ein mathematisches Modell für Euklids Problem angeben:

$$\begin{aligned}
\max \quad & 2\Delta(A,B,C)(1 - \frac{x}{BC}) \cdot \frac{x}{BC} \\
\text{s.d.} \quad & 0 \leq x \leq BC
\end{aligned}$$

Hat man eine mathematische Beschreibung eines Problems gefunden, ist der nächste Schritt natürlich der Versuch, das Problem zu lösen. Konkret heißt das, wir wollen einen Wert für x finden, der zulässig ist (also $0 \leq x \leq BC$ erfüllt) und zu einem möglichst großen Wert der Zielfunktion führt. Dazu gehen wir ganz klassisch vor, leiten die Funktion ab und suchen eine Nullstelle der Ableitung, die uns in diesem Fall auch direkt das gesuchte Maximum liefert.

$$\begin{aligned}
\max \quad f(x) = \quad & 2\Delta(A,B,C)(1 - \frac{x}{BC}) \cdot \frac{x}{BC} \\
f'(x) = \quad & 2\Delta(A,B,C) \left((\frac{1}{BC}) - \frac{2x}{BC^2} \right) \\
f'(x) = \quad & 0 \Longleftrightarrow x = \frac{BC}{2} \\
f''(\tfrac{BC}{2}) < \quad & 0 \\
\Rightarrow \quad & Maximum \;\; an \;\; x^* = \frac{BC}{2}
\end{aligned}$$

Das zweite Beispiel entstammt dem Gebiet der Standortplanung, auf das wir in in Kapitel 5 etwas ausführlicher eingehen werden. Es ist als Sylvesters Problem bekannt und stammt aus dem Jahr 1857.

Beispiel 1.2 *Standortplanung (Sylvesters Problem, 1857)*
Seien M Punkte $A_1, \ldots, A_M$ in der Ebene $\mathbb{R}^2$ gegeben, mit Koordinaten $A_m = (a_{m1}, a_{m2})$ für $m = 1, \ldots, n$. Finde den kleinsten Kreis, der diese Punkte enthält.

Auch hier müssen wir uns zunächst Klarheit über die Variablen und die Zielfunktion verschaffen. Gesucht ist ein Kreis, dementsprechend wählen wir drei Variablen: die beiden Koordinaten des Mittelpunktes $(x_1, x_2) \in \mathbb{R}^2$ und den Radius des Kreises $r \in \mathbb{R}$. Der Radius ist gleichzeitig unsere Zielfunktion. Die Zielfunktion ist hier also sehr einfach, dafür sind aber die Nebenbedingungen etwas komplizierter. Sie müssen sichern, dass alle Punkte A_m im Kreis liegen. Das können wir ausdrücken, indem wir fordern, dass für jeden der Punkte A_m seine Entfernung zum (gesuchten) Kreismittelpunkt x kleiner als der (gesuchte) Radius ist.

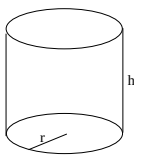
Das Modell lässt sich nun folgendermaßen aufschreiben:

$$\begin{array}{ll} \min & r \\ \text{s.d.} & \sqrt{(x_1 - a_{m1})^2 + (x_2 - a_{m2})^2} \leq r \quad m = 1, \cdots, M \\ & x_1, x_2 \in \mathbb{R}, r \in \mathbb{R}, r \geq 0 \end{array}$$

(Zur Lösung des Problems siehe Aufgabe ÜBUNG).

Das letzte der drei Beispiele spiegelt eine typische Optimierungsaufgabe aus dem Bereich der Wirtschaftsmathematik wieder. Hier geht es um die Konstruktion einer möglichst preisgünstigen Dose. Bei Optimierungsaufgaben aus dem Bereich der Betriebswirtschaftslehre geht es häufig um die Optimierung von Produktionsplänen, Lagerverwaltungen oder ähnlichem.

Beispiel 1.3
Konstruiere eine möglichst billige Dose (Kreiszylinder) mit Radius r und Höhe h, die mindestens das Volumen V hat. Dabei betragen die Kosten für Deckel und Boden C1 pro Flächeneinheit und die Kosten für den Mantel der Dose C2 pro Flächeneinheit.



Die Variablen müssen in diesem Beispiel so gewählt werden, dass sie den gesuchten Zylinder beschreiben. Naheliegend ist es also, den Radius des Zylinders r und seine Höhe h als Variablen zu wählen.

Die Zielfunktion soll die Kosten des Zylinder in Abhängigkeit von seinem Radius r und seiner Höhe h beschreiben. Sie ist dementsprechend gegeben als

$$f(r,h) = c_1 \cdot 2 \cdot \pi r^2 + c_2 \cdot h \cdot 2 \cdot \pi r$$

Würden wir nur diese Zielfunktion betrachten und wollten sie minimieren, so könnte man Kosten von Null erreichen, wenn man Radius und Höhe gleich Null wählt (also überhaupt keine Dose baut). Das wird dadurch verhindert, dass in der Aufgabenstellung gefordert ist, dass das Volumen mindestens V beträgt. d.h. $\pi r^2 \cdot h \geq V$. Zusammen erhalten wir folgendes Modell:

$$\begin{array}{ll} \min & 2c_1\pi r^2 + 2c_2\pi rh \\ \text{s.d.} & \pi r^2 h \geq V \\ & h, r \geq 0 \end{array}$$

Auch die Lösung dieser Aufgabe soll als Übungsaufgabe behandelt werden.

Klassifizierung von Optimierungsproblemen

Betrachte ein Optimierungsproblem mit $I = I_1 \cup I_2$ wie in Definition 1.1 eingeführt.

$$\begin{array}{rcl} \min f(x) & & \\ \text{s.d.} \quad g_i(x) & \leq & 0 \quad \forall i \in I_1 \\ g_i(x) & = & 0 \quad \forall i \in I_2 \\ x \in \mathcal{B} & & \end{array}$$

Man unterscheidet folgende (nicht disjunkte) Problemklassen von Optimierungsproblemen. Zunächst differenziert man zwischen Problemen mit eingeschränktem Lösungsraum und Problemen ohne Nebenbedingungen.

- Nicht-restringierte Probleme: Bei nicht-restringierten Problemen liegen keine Nebenbedingungen vor, d.h. jeder Punkt im $\mathbb{R}^n$ ist als Lösung des Problems erlaubt. Formal liegt ein nicht-restringiertes Problem vor, wenn $\mathcal{B} = \mathbb{R}^n$ und $I = \emptyset$.

- Programme mit Nebenbedingungen (Restriktionen): Hier dürfen nicht alle Punkte des $\mathbb{R}^n$ gewählt werden, sondern es müssen Nebenbedingungen

beachtet werden. Diese können vorliegen als Ungleichheitsbedingungen der Form $g_i(x) \leq 0$, als Gleichheitsnebenbedingungen der Form $g_i(x) = 0$, oder sie können aus einer Einschränkung des zulässigen Bereiches durch die Menge $\mathcal{B}$ bestehen.

- ganzzahlige (kombinatorische/diskrete) Programme: Diese sind ein wichtiger Spezialfall von Programmen mit Nebenbedingungen, in denen gefordert wird, dass die Lösung ganzzahlig sein soll, d.h. $\mathcal{B} = \mathbb{Z}^n$. Natürlich können sie auch weitere Nebenbedingungen beinhalten. Es ist zu bemerken, dass sehr viele in der Praxis auftretende Probleme mit Hilfe solcher ganzzahligen Programme modelliert werden können. Ein Spezialfall von ganzzahligen Programmen sind sogenannte boolesche Programme, in denen die Variablen weiter auf die Werte 0 und 1 eingeschränkt sind, d.h. $\mathcal{B} = \{0,1\}^n$. Auch diese Programme haben hohe Bedeutung in der Praxis und sind meistens erstaunlich schwierig zu lösen. Auch das Gebiet der Netzwerkoptimierung und viele Probleme aus dem Bereich des Operations Research fallen in die Klasse der ganzzahligen Optimierung.

Nach der Struktur der auftretenden Funktionen kann man Optimierungsprobleme weiter in folgende Klassen einteilen:

- Lineare Programme: f, g_i sind affin linear $\forall i \in I$ und $\mathcal{B} = \mathbb{R}^n$.
- konvexe Programme: f und g_i sind konvex $\forall i \in I_1$, g_i affin linear $\forall i \in I_2$ und $\mathcal{B} = \mathbb{R}^n$.
- glatte Programme: f, g_i sind differenzierbar $\forall i \in I$ und $\mathcal{B} = \mathbb{R}^n$.

Natürlich kann man die eben genannten Klassen auch kombinieren und weiter spezifizieren. Bedeutung haben unter anderem

1. differenzierbare konvexe Programme
2. konvexe ganzzahlige Programme
3. lineare ganzzahlige Programme

Die in diesem Abschnitt beschriebenen Klassen sind nicht disjunkt, sondern wohin ein Problem gehört, hängt von der jeweiligen Formulierung des Problems ab. Das wird im folgenden Beispiel gezeigt.

Beispiel 1.4 *Sei* $f : \mathbb{R}^n \to \mathbb{R}$ *linear.*

$$\begin{array}{lcl} \min\ f(x) & \textit{ist äquivalent} & \min\ f(x) \\ s.d.\ x \in \mathbb{Z}^n & \textit{zu} & s.d.\ \sin \pi x_i = 0\ \ i = 1, \cdots, n \\ & & \quad x \in \mathbb{R}^n \\ \uparrow & & \uparrow \\ \textit{ganzzahliges lineares Programm} & & \textit{nichtlineares glattes Programm} \end{array}$$

Wir führen jetzt ein paar wichtige Notationen formal ein, die zur Diskussion von Optimierungsproblemen nötig sind.

Definition 1.2 *Sei*

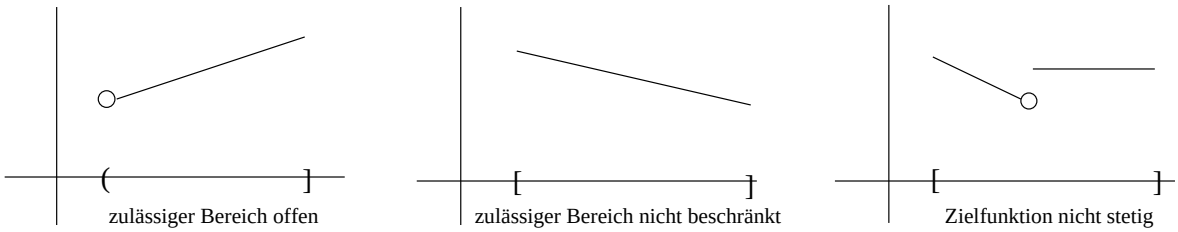
$$
\begin{array}{lllll}
(Opt) & \min & f(x) & & \\
& \text{s.d.} & g_i(x) & \leq & 0 \quad \forall i \in I_1 \\
& & g_i(x) & = & 0 \quad \forall i \in I_2 \\
& & x \in \mathcal{B} & &
\end{array}
$$

ein mathematisches Programm.

- $x \in \mathcal{B}$ *heißt* **zulässige Lösung** *(oder* **zulässig** *oder* **zulässiger Punkt***), falls* $g_i(x) \leq 0 \ \forall i \in I_1$ *und* $g_i(x) = 0 \ \forall i \in I_2$.
- *Der* **zulässige Bereich** *ist die Menge aller zulässigen Lösungen von (Opt), d.h.*
$$\mathcal{F} = \{x \in \mathcal{B} : g_i(x) \leq 0 \ \forall i \in I_1 \text{ und } g_i(x) = 0 \ \forall i \in I_2\}.$$
- *Eine zulässige Lösung* x *heißt* **(globale) Optimallösung**, *falls* $f(x) \leq f(y)$ *für alle zulässigen Lösungen* y.
- *Eine zulässige Lösung* x *heißt* **lokal optimal**, *falls es eine Nachbarschaft* $N(x)$ *gibt, so dass* $f(x) \leq f(y)$ *für alle* $y \in N(x) \cap \mathcal{F}$.

  Bemerkung: Insbesondere bei diskreten Problemen ist die Definition einer vernünftigen Nachbarschaft nicht eindeutig - die Definition von lokal optimal hängt also von der Nachbarschaft ab!

Wie schon erwähnt, müsste man in der Formulierung von (Opt) streng genommen das Minimum durch ein Infimum ersetzen, da von vornherein nicht klar ist, ob ein gegebenes Optimierungsproblem eine Optimallösung oder überhaupt eine (zulässige) Lösung hat. Wir beschäftigen uns daher zunächst mit der Frage, wann (Opt) eine Optimallösung hat. Das gilt sicher, wenn $\mathcal{F}$ eine endliche Menge ist und damit z.B. für alle booleschen Programme. Ist $\mathcal{F}$ dagegen eine unendliche Menge, so muss ein Optimum nicht existieren. Die folgende Abbildung zeigt drei eindimensionale Optimierungsprobleme $\min\{f(x) : x \in \mathcal{F}\}$, in denen der zulässige Bereich $\mathcal{F} \subseteq \mathbb{R}$ jeweils ein reelles Intervall ist. In allen drei Fällen existiert keine Optimallösung.



Die hier skizzierten Optimierungsprobleme werden ausgeschlossen, wenn f stetig ist und $\mathcal{F}$ kompakt. Dass diese beiden Bedingungen reichen, um die Existenz einer Optimallösung zu garantieren, zeigt der folgende, auf Weierstrass zurückgehende Satz.

Satz 1.3 (Satz von Weierstrass) *Sei f stetig und $\mathcal{F}$ nichtleer und kompakt. Dann hat das Problem*

$$\begin{array}{ll} \min & f(x) \\ s.d. & x \in \mathcal{F} \end{array}$$

eine Optimallösung.

Beweis: Dieser Beweis fällt üblicherweise in das Gebiet der Analysis. Er soll wegen seiner Bedeutung für die Optimierung dennoch kurz skizziert werden.

Sei $\alpha := \inf\{f(x) : x \in \mathcal{F}\}$.

Fall 1: $\alpha > -\infty$. Dann wähle Nullfolge $\varepsilon^k \to 0$ und definiere $\mathcal{F}_k = \{x \in \mathcal{F} : \alpha \leq f(x) \leq \alpha + \varepsilon^k\}$. Es gilt, dass $\mathcal{F}_k \neq \emptyset \ \ \forall k \in \mathbb{N}$, sonst wäre $\alpha + \varepsilon^k$ Infimum und nicht wie vorausgesetzt α.

Wähle nun ein $x_k \in \mathcal{F}_k, \forall k \in \mathbb{N}$. Man erhält eine Folge $(x_k) \subseteq \mathcal{F}$. Weil $\mathcal{F}$ beschränkt ist existiert eine konvergente Teilfolge $\{y_k\} \subseteq \{x_k\}$ die gegen einen Häufungspunkt $\overline{y}$ konvergiert,

$$y_l \to \overline{y} \text{ für } l \to \infty.$$

Weil $\mathcal{F}$ abgeschlossen ist folgt $\overline{y} \in \mathcal{F}$.

Nun nutzen wir aus, dass $y_l \in \mathcal{F}_l$ und erhalten $\alpha \leq f(y_l) \leq \alpha + \varepsilon^l$ nach der Definition von $\mathcal{F}_l$.

$$\Rightarrow \quad \alpha = \lim_{l\to\infty} f(y_l) \quad \underset{\substack{\uparrow \\ f\,stetig}}{=} \quad f(\lim_{l\to\infty} y_l) = f(\overline{y}),$$

also $\overline{y} \in \mathcal{F}$ und $f(\overline{y}) = \alpha$. Damit ist $\overline{y}$ ist das gesuchte Minimum.

Fall 2: $\alpha = -\infty$. In diesem Fall betrachte die Mengen $\mathcal{F}_k = \{x \in \mathcal{F} : f(x) \leq -k\}$. Diese sind für alle k nichtleer, sonst wäre das Infimum größer als $-\infty$. Das führt aber zu einem Widerspruch zur Beschränktheit von $\mathcal{F}$.

QED

Für diskrete Optimierungsprobleme kann man die Existenz einer optimalen Lösung oft leicht überprüfen: Enthält die zulässige Menge nämlich nur eine endliche Anzahl an Lösungen überhaupt, dann ist (mindestens) eine von ihnen optimal.

Lemma 1.4 *Ist $\mathcal{F}$ nichtleer und gilt $|\mathcal{F}| < \infty$, dann hat das Problem*

$$\begin{array}{ll} \min & f(x) \\ s.d. & x \in \mathcal{F} \end{array}$$

eine Optimallösung.

Zum Schluss sollten wir noch bemerken, dass es in diesem allgemeinen Teil egal ist, ob wir $\min f(x)$ oder $\max f(x)$ behandeln, da $\min f(x) = -\max(-f(x))$! Das gilt allerdings nicht für speziellere Optimierungsprobleme, deren Struktur durch das Negieren der Zielfunktion verloren geht. Zum Beispiel ist es im Fall der konvexen Optimierung von entscheidender Bedeutung, ob man maximiert oder minimiert.

Kapitel 2

Lineare Optimierung

2.1 Lineare Programme und graphisches Lösungsverfahren

Zur Illustration von Optimierungsproblemen betrachten wir zunächst ein einführendes Beispiel.

Beispiel 2.1 *Firma "Apfelrein" stellt die beiden folgenden Produkte her:*

- *Produkt 1 (Apfel-Duschgel): Eine Einheit bringt einen Erlös von 3 Euro.*
- *Produkt 2 (Apfel-Shampoo): Eine Einheit bringt einen Erlös von 5 Euro.*

Zur Herstellung der beiden Produkte sind drei Maschinen A, B und C nötig. Eine Einheit des Duschgels benötigt 1 Stunde auf Maschine A und 1 Stunde auf Maschine B. Die Produktion des Apfel-Shampoos ist etwas aufwändiger. Für die Herstellung einer Einheit davon ist eine Bearbeitungszeit von 2 Stunden auf Maschine A, eine Stunde auf Maschine B und 3 Stunden auf Maschine C nötig.
Da die Maschinen noch für andere Produkte verwendet werden, und außerdem auch gewartet werden müssen, stehen sie nicht die ganze Zeit zur Verfügung. Pro Monat können die Maschinen wie in der nächsten Tabelle angegeben verwendet werden.

Maschine	A	*steht* 170 *Stunden zur Verfügung*
	B	*steht* 150 *Stunden zur Verfügung*
	C	*steht* 180 *Stunden zur Verfügung*

Nach einer Marktanalyse geht Firma Apfelrein davon aus, dass alle von ihr produzierten Einheiten der beiden Produkte auch verkauft werden können. Wie viel soll die Firma von welchem Produkt produzieren, um ihren Gewinn zu maximieren?

Wie im vorangegangenen Abschnitt müssen wir uns zunächst Klarheit über die zur Beschreibung des Problems nötigen Variablen verschaffen. Wir definieren:

$$\begin{aligned} x_1 &= \textit{Menge an Apfel-Duschgel} \\ x_2 &= \textit{Menge an Apfel-Shampoo} \end{aligned}$$

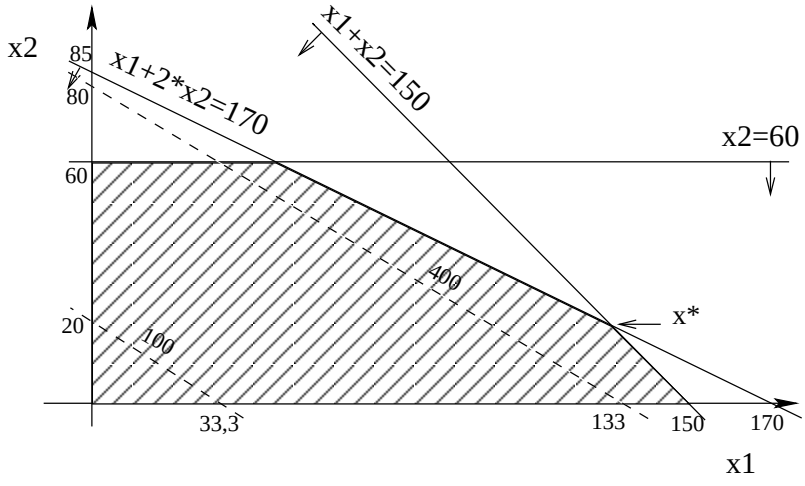
Die Zielfunktion beschreibt den Gewinn des Unternehmens Apfelrein. Er ergibt sich als $f(x_1, x_2) = 3x_1 + 5x_2$. Das Ziel ist, diese Funktion zu maximieren. Als Nebenbedingungen ist zu berücksichtigen, dass die nötigen Bearbeitungszeiten der Produkte die freien Zeiten auf den drei Maschinen nicht überschreiten. Wir formulieren also

$$\begin{aligned} x_1 + 2x_2 &\leq 170 \\ x_1 + x_2 &\leq 150 \\ 3x_2 &\leq 180 \\ x_1, x_2 &\geq 0. \end{aligned}$$

Die letzte Bedingung stellt dabei sicher, dass die beiden gesuchten Mengen nicht negativ sein dürfen. Im gesamten erhalten wir somit das folgende lineare Programm:

$$\begin{aligned} \max\ 3x_1 + 5x_2 & \\ x_1 + 2x_2 &\leq 170 \\ x_1 + x_2 &\leq 150 \\ 3x_2 &\leq 180 \\ x_1, x_2 &\geq 0. \end{aligned}$$

Die folgende Zeichnung veranschaulicht den zulässigen Bereich des Problems in Abhängigkeit der beiden Variablen x_1 und x_2.



Ein mögliches Vorgehen zur Lösung eines solchen linearen Programms in zwei Variablen ist das folgende:

1. *Zeichne den zulässigen Bereich.*

2. *Zeichne die Zielfunktion f für einen festen Wert z durch Markierung aller Punkte, die zu diesem Zielfunktionswert z führen. Die Menge dieser Punkte $L_=(z) = \{x \in \mathbb{R}^2 : f(x) = z\}$ wird auch die* **Niveaulinie** *der Funktion f von z genannt. In unserem Fall erhalten wir die Menge*

$$L_=(z) = \{(x_1, x_2) : z = 3x_1 + 5x_2\},$$

   die durch die Gerade $x_2 = \frac{z}{5} - \frac{3}{5}x_1$ dargestellt werden kann. In obiger Zeichnung sind die Geraden für $z = 100$ und $z = 400$ dargestellt.

3. *Verschiebe $L_=(z)$, bis z maximal und $L_=(z) \cap \mathcal{F} \neq \emptyset$. Optimal sind dann alle Punkte in der so erhaltenen Schnittmenge. In unserem Beispiel ergeben sich die optimalen Produktionsmengen zu ${x_1}^* = 130$ und ${x_2}^* = 20$. Der maximal mögliche Gewinn lässt sich als $z = 490$ ablesen.*

Beobachtung: Das Optimum liegt an einer Ecke des zulässigen Polyeders!

Die eben genannte Beobachtung ist für die Verfahren der linearen Optimierung sehr wichtig, wir werden uns mit ihr im Hauptsatz der linearen Optimierung (Satz 2.21 in Abschnitt 2.4) ausführlicher beschäftigen. Jetzt wenden wir uns zunächst der Formalisierung einiger grundlegender Begriffe der linearen Optimierung zu.

Definition 2.1 *Ein lineares Programm in* **allgemeiner Form** *ist gegeben durch:*

$$\begin{array}{rcl} \min & c^t x & (\textit{oder } \max\ c^t x) \\ \text{s.d. } A_1 x & = & b_1 \\ A_2 x & \leq & b_2 \\ A_3 x & \geq & b_3 \\ x_j & \geq & 0 \quad j = 1, \cdots, n_1 \\ x_j & \leq & 0 \quad j = n_1 + 1, \cdots, n_2 \\ x_j & \gtrless & 0 \quad j = n_2 + 1, \cdots, n \end{array}$$

$$\textit{mit } A_i \textit{ ist } m_i \times n - \textit{Matrix} \qquad b_i \in \mathbb{R}^{m_i} \qquad c \in \mathbb{R}^n \quad x \in \mathbb{R}^n$$

Die Schreibweise $x_j \gtrless 0$ drückt aus, dass keine Einschränkung an x_j gemacht wird, d.h. für x_j sind sowohl positive als auch negative Werte erlaubt (einschließlich des Wertes 0). Natürlich könnte man auch einfach $x_j \in \mathbb{R}$ schreiben, aber die Schreibweise $x_j \gtrless 0$ hat sich in vielen Arbeiten der Optimierung durchgesetzt.

Definition 2.2 *Ein lineares Programm in* **Standardform** *ist gegeben durch:*

$$\begin{array}{rcl} \min & c^t x & \\ \text{s.d. } Ax & = & b \\ x & \geq & 0, \end{array}$$

wobei A eine $m \times n$-Matrix ist, $b \in \mathbb{R}^m$, $c \in \mathbb{R}^n$, und die Variablen $x \in \mathbb{R}^n$.

Das im Beispiel aufgestellte und mit dem graphischen Lösungsverfahren bearbeitete lineare Programm liegt allerdings nicht in Standardform vor, sondern in der $\leq$–Form.

Definition 2.3 *Ein lineares Programm in* $\leq$**–Form** *ist gegeben durch:*

$$\begin{array}{rcl} \min & c^t x & \\ \text{s.d. } Ax & \leq & b \\ x & \gtrless & 0, \end{array}$$

wobei auch hier A eine $m \times n$-Matrix ist, $b \in \mathbb{R}^m$, $c \in \mathbb{R}^n$, und die Variablen $x \in \mathbb{R}^n$.

Für geometrische Überlegungen eignet sich die $\leq$–Form, während es bei der algebraischen Untersuchung von linearen Programmen und der Entwicklung von Lösungsverfahren angenehm ist, sich auf die einfache Form von linearen Programmen in Standardform zu beschränken. Wir werden diese Form vorrangig (im wesentlichen ab Abschnitt 2.3) verwenden. Dass man ohne Beschränkung der Allgemeinheit annehmen darf, dass ein gegebenes lineares Programm in Standardform vorliegt, zeigt das folgende Ergebnis.

Lemma 2.4 *Jedes lineare Programm in allgemeiner Form kann in ein äquivalentes lineares Programm in Standardform überführt werden.*

Beweis: Den Beweis führen wir konstruktiv, indem wir schrittweise zeigen, wie man ein lineares Programm in allgemeiner Form in ein äquivalentes lineares Programm in Standardform überführt.

- Falls ein Maximierungsproblem vorliegt, schreiben wir zunächst die Zielfunktion um: $\max c^t x \Longleftrightarrow -\min(-c^t)x$

- Jetzt beschäftigen wir uns mit den „$\geq$ Nebenbedingungen“ . Sei also $a_1x_1 + \cdots + a_nx_n \geq 0$. Durch Einführen einer so genannten *Überschussvariablen* (*surplus variable*), die wir hier mit y bezeichnen, können wir die Nebenbedingung äquivalent umschreiben zu:

$$a_1x_1 + \cdots + a_nx_n \geq 0 \Longleftrightarrow a_1x_1 + \cdots + a_nx_n - y = 0,\ y \geq 0.$$

- Ganz ähnlich verfahren wir mit „$\leq$ Nebenbedingungen“ der Form $a_1x_1 + \cdots + a_nx_n \leq 0$. Für jede solche Nebenbedingung führen wir eine *Schlupfvariable* (*slack variable*) $y \geq 0$ ein und erhalten:

$$a_1x_1 + \cdots + a_nx_n \leq 0 \Longleftrightarrow a_1x_1 + \cdots + a_nx_n + y = 0,\ y \geq 0.$$

- Jetzt müssen noch die Variablen betrachtet werden. Solche Variablen, für die $x_j \leq 0$ gefordert wird, ersetzt man durch neue Variablen $\hat{x}_j = -x_j$.

- Bei Variablen x_j, für die keine Beschränkung vorliegt, also $x_j \gtrless 0$, benötigt man zwei neue Variablen, x_j^+ und x_j^-. Man ersetzt x_j durch ${x_j}^+ - {x_j}^-$ und fordert $x_j^+, x_j^- \geq 0$.

QED

Es ist zu bemerken, dass sich bei der Transformation die Anzahl der Variablen erhöht, da man für jede Ungleichungsnebenbedingung eine neue Variable braucht und jede unbeschränkte Variable durch zwei beschränkte Variablen ersetzt. Die Transformation soll an dem folgenden Beispiel verdeutlicht werden, bei dem wir ein gegebenes LP mit drei Variablen in ein LP in Standardform (mit sechs Variablen) überführen.

Beispiel 2.2

$$\begin{array}{ll} \max & c_1x_1 - c_2x_2 + c_3x_3 \\ \text{s.d.} & x_1 + x_2 + x_3 \leq 4 \\ & 2x_1 - x_2 \geq 1 \\ & x_1 \leq 0 \qquad\qquad \hat{x}_1 = -x_1 \\ & x_2 \gtrless 0 \qquad\qquad x_2 = x_2^+ - x_2^- \\ & x_3 \geq 0 \end{array}$$

wird zu:

$$\begin{array}{ll} \min & c_1\hat{x}_1 + c_2(x_2^+ - x_2^-) - c_3x_3 \\ \text{s.d.} & -\hat{x}_1 + x_2^+ - x_2^- + x_3 + y_1 = 4 \\ & -2\hat{x}_1 - x_2^+ + x_2^- - y_2 = 1 \\ & \hat{x}_1 \geq 0\ x_2^+, x_2^- \geq 0\ x_3 \geq 0 \\ & y_1 \geq 0\ y_2 \geq 0. \end{array}$$

Noch viel einfacher lässt sich jedes lineare Programm in die $\leq$–Form überführen, indem man jede Gleichheits-Nebenbedingung $a^tx = b_i$ durch zwei Ungleichheitsnebenbedingungen $a^tx \leq b_i$ und $-a^tx \leq -b_i$ ersetzt, und vorzeichenbeschränkte Variablen als zusätzliche Nebenbedingungen auffasst.

2.2 Geometrische Grundlagen

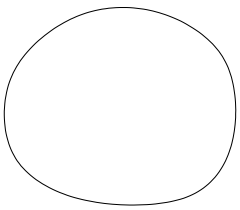
In diesem Abschnitt wollen wir einige geometrische Grundbegriffe einführen, die nicht nur für die lineare Optimierung, sondern auch für viele andere Bereiche der Optimierung von großer Bedeutung sind. Das Ziel dieses Abschnittes besteht jedoch darin, ein geometrisches Grundverständnis für die Struktur von linearen Optimierungsproblemen zu entwickeln, mit Hilfe dessen man sich später sehr viele – auch fortgeschrittene – Ergebnisse der linearen Optimierung veranschaulichen kann. Die hier vorgestellten Ergebnisse finden sich in ähnlicher Form in vielen Lehrbüchern. Wir verweisen hier insbesondere auf [JS04].

Wir betrachten zunächst Teilmengen $\mathcal{M}$ im $\mathbb{R}^n$, und führen zwei Klassen von Teilmengen mit besonders angenehmer Struktur ein: konvexe Mengen und Polyeder.

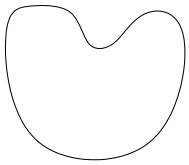
Definition 2.5 *Eine Menge $\mathcal{M} \subseteq \mathbb{R}^n$ heißt* **konvex**, *falls $\forall x, y \in \mathcal{M}$ und $\forall \lambda \in (0,1)$ gilt:*

$$\lambda x + (1-\lambda)y \in \mathcal{M}.$$

Geometrisch lässt sich diese Definition leicht veranschaulichen: Für zwei Punkte $x, y \in \mathbb{R}^n$ beschreibt die Menge $\{\lambda x + (1-\lambda)y : 0 < \lambda < 1\}$ genau die Strecke zwischen x und y. Eine Menge ist also konvex, falls mit je zwei Punkten aus der Menge auch die Verbindungsstrecke dieser Punkte in der Menge enthalten ist. Eine konvexe und eine nicht konvexe Menge sind in der folgenden Abbildung dargestellt.



konvexe Menge



nicht konvex

Zu den Mengen mit der einfachsten Struktur gehören lineare und affin lineare Mengen wie Geraden und Hyperebenen. Diese werden von linearen Gleichungen, wie sie z.B. in den Nebenbedingungen eines linearen Programms in Standardform auftreten, erzeugt. Dagegen erzeugen lineare Ungleichungen *Halbräume*.

Definition 2.6 *Sei $a \in \mathbb{R}^n$, $b \in \mathbb{R}$.*

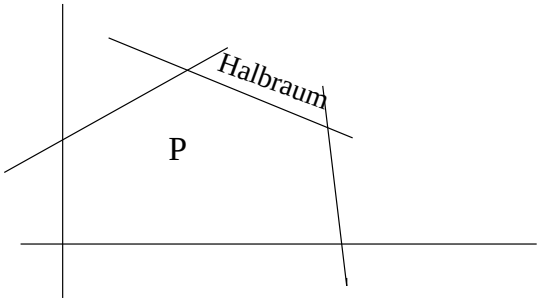
- *$H_{a,b} = \{x \in \mathbb{R}^n : a^t x = b\}$ ist die zu a senkrechte Hyperebene.*
- *$H^{\leq}_{a,b} = \{x \in \mathbb{R}^n : a^t x \leq b\}$ und $H^{\geq}_{a,b} = \{x \in \mathbb{R}^n : a^t x \geq b\}$ nennt man die von der Hyperebenen $H_{a,b}$ erzeugten Halbräume.*

Eine besondere Form von konvexen Mengen sind konvexe Polyeder, die man durch Halbräume konstruieren kann.

Definition 2.7 *Ein* **Polyeder** *P ist der Durchschnitt endlich vieler Halbräume, d.h.*

$$P = \bigcap_{j=1,\ldots,m} \{x \in \mathbb{R}^n : a_j^t x \leq b_j\}$$

mit $a_j \in \mathbb{R}^n, b_j \in \mathbb{R}$ für $j = 1, \ldots, m$.



Lemma 2.8 *Die folgenden Aussagen gelten.*

1. *Jeder Halbraum ist konvex.*
2. *Seien $\mathcal{M}_1, \cdots, \mathcal{M}_k$ konvex, dann ist $\bigcap_{i=1,\cdots,k} \mathcal{M}_i$ konvex.*
3. *Polyeder sind konvex.*

Beweis:

1. Sei $H = \{x \in \mathbb{R}^n : a^t x \leq b\}, \quad a \in \mathbb{R}^n, b \in \mathbb{R}$ ein Halbraum und seien $x_1, x_2 \in H$

   $$\Longrightarrow a^t x_1 \leq b, \;\; a^t x_2 \leq b.$$

   Sei $\lambda \in (0,1)$. Dann gilt:

   $$\begin{aligned} a^t(\lambda x_1 + (1-\lambda)x_2) &= \lambda a^t x_1 + (1-\lambda) a^t x_2 \\ &\leq \lambda b + (1-\lambda) b = b, \end{aligned}$$

   also $\lambda x_1 + (1-\lambda)x_2 \in H$.

2. Sei $x_1, x_2 \in \bigcap_{i=1,\ldots,k} \mathcal{M}_i$ und sei $\lambda \in (0,1)$:

   $$\begin{aligned} &\Longrightarrow \quad x_1, x_2 \in \mathcal{M}_i \;\forall i = 1, \ldots, k \\ &\Longrightarrow \quad \lambda x_1 + (1-\lambda)x_2 \in \mathcal{M}_i \;\forall i = 1, \ldots, k, \quad \text{weil } \mathcal{M}_i \text{ konvex} \\ &\Longrightarrow \quad \lambda x_1 + (1-\lambda)x_2 \in \bigcap_{i=1,\ldots,k} \mathcal{M}_i, \end{aligned}$$

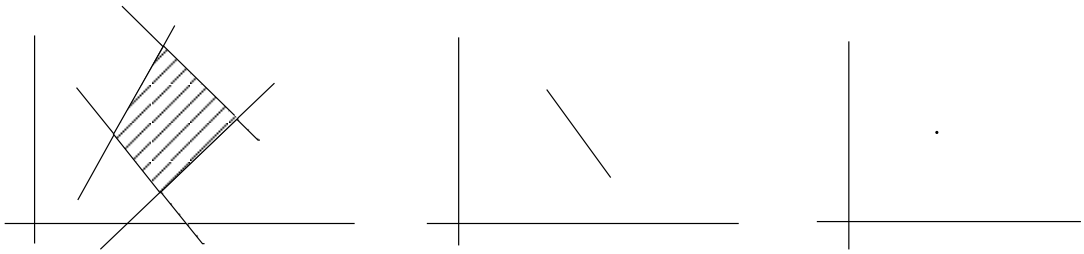
   also ist $\bigcap_{i=1,\ldots,k} \mathcal{M}_i$ konvex.

3. Die Aussage folgt direkt aus 1. und 2. QED

Der zulässige Bereich eines linearen Programms ist also ein (konvexes) Polyeder.
Man sieht das direkt, wenn die Nebenbedingungen als Ungleichungen formuliert sind, das lineare Programm also in $\leq$-Form vorliegt. Da man jede Gleichheitsnebenbedingung äquivalent durch zwei Ungleichheitsnebenbedingungen ersetzen kann,

$$a^t x = b \Leftrightarrow a^t x \leq b \quad \text{und} \quad a^t x \geq b$$

gilt diese Aussage also auch für einen zulässigen Bereich, der Gleichheits-Nebenbedingungen enthält und somit für alle linearen Programme.
Die folgende Abbildung zeigt die drei verschiedenen Typen von Polyedern, die im $\mathbb{R}^2$ auftreten können.



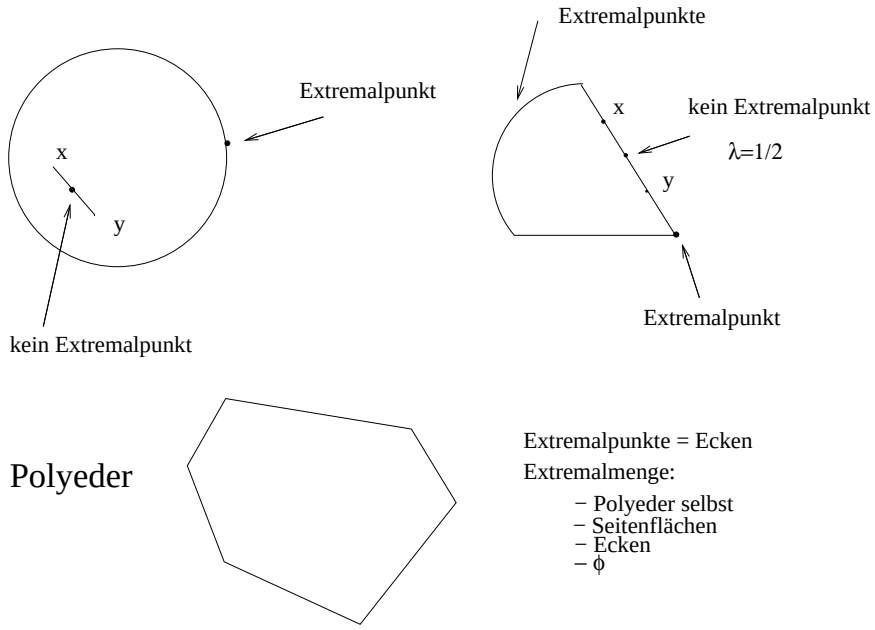
Definition 2.9 *Sei $\mathcal{M}$ eine konvexe Menge.*

- *Eine konvexe Teilmenge $\mathcal{E} \subseteq \mathcal{M}$ heißt* **Extremalmenge** *von $\mathcal{M}$, falls aus $a \in \mathcal{E}, x, y \in \mathcal{M}, \lambda \in (0,1)$ und $a = \lambda x + (1-\lambda)y$ folgt: $x, y \in \mathcal{E}$.*

- *Ein Punkt $a \in \mathcal{M}$ heißt* **Extremalpunkt**, *falls mit $x, y \in \mathcal{M}$ und $\lambda \in (0,1)$ aus $a = \lambda x + (1-\lambda)y$ folgt: $a = x = y$.*

Bemerkung: Extremalpunkte sind 0-dimensionale Extremalmengen.

Extremalpunkte und Extremalmengen werden in der folgenden Abbildung veranschaulicht.



Nun wollen wir die eben eingeführten Begriffe auf lineare Programme anwenden und auf geometrische Art einen Satz beweisen, dem wir in Abschnitt 2.3.4 als Hauptsatz der linearen Optimierung (Satz 2.21) wieder begegnen werden.

Satz 2.10 *Sei (LP) ein lineares Programm mit zulässigem Bereich $\mathcal{F}$. Sei $\mathcal{E}$ die Menge seiner Optimallösungen. Gilt $\mathcal{E} \neq \emptyset$, so ist $\mathcal{E}$ eine Extremalmenge von $\mathcal{F}$.*

Beweis: Wir schreiben (LP) als

$$(LP) \quad \min\{c^t x : x \in \mathcal{F}\}.$$

Nehmen wir nun an, dass $\mathcal{E} \neq \emptyset$. Das heißt, $\alpha = \min\{c^t x : x \in \mathcal{F}\}$ existiert und die Menge der Optimallösungen des (LP) ist gegeben als

$$\mathcal{E} = \{x \in \mathcal{F} | c^t x = \alpha\} \subseteq \mathcal{F}.$$

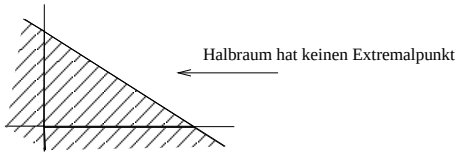
$\mathcal{E}$ ist als Durchschnitt des Polyeders $\mathcal{F}$ mit einer Hyperebene (beziehungsweise mit zwei Halbräumen) erneut ein Polyeder, welches in $\mathcal{F}$ enthalten ist.

Angenommen, $\mathcal{E}$ sei keine Extremalmenge von $\mathcal{F}$. Dann gibt es $a \in \mathcal{E}$ das sich schreiben lässt als $a = \lambda x + (1-\lambda)y$ mit $0 < \lambda < 1$ und $x, y \in \mathcal{F}$, wobei mindestens einer der beiden Vektoren x und y nicht in $\mathcal{E}$ liegt. Ohne Beschränkung der Allgemeinheit nehmen wir an, $x \notin \mathcal{E}$. Wir erhalten $c^t x > \alpha$ und $c^t y \geq \alpha$. Daraus ergibt sich (da $a \in \mathcal{E}$):

$$\begin{aligned} \alpha = c^t a &= c^t(\lambda x + (1-\lambda)y) \\ &= \lambda c^t x + (1-\lambda) c^t y \\ &> \lambda\alpha + (1-\lambda)\alpha = \alpha, \end{aligned}$$

ein Widerspruch. Also ist die Menge der Optimallösungen $\mathcal{E}$ eine Extremalmenge des zulässigen Polyeders. QED

Wie das folgende Bild zeigt, ist jedoch selbst bei Polyedern die Existenz von Extremalmengen (außer der leeren Menge und des Polyeders selbst) und die Existenz von Extremalpunkten nicht gesichert:



Aber für Polyeder P, die keine Gerade enthalten, gilt, dass jede Extremalmenge von P auch (mindestens einen) Extremalpunkt enthält. Weil jedes lineare Programm zu einem LP in Standardform äquivalent ist, und der zulässige Bereich $\mathcal{F} = \{x \in \mathbb{R}^n : Ax = b, x \geq 0\}$ eines linearen Programms in Standardform aufgrund der Nichtnegativitätsbedingungen keine Gerade enthält, lässt sich die folgende Aussage zeigen:

Hat ein lineares Programm überhaupt Optimallösungen,
dann gibt es unter ihnen auch Extremalpunkte(=Ecken).

Das ist die geometrische Form des Hauptsatzes der linearen Optimierung, den wir in Abschnitt 2.3.4 formal einführen und beweisen werden.

2.3 Basislösungen

2.3.1 Basen linearer Programme

In diesem und dem folgenden Abschnitt 2.4 wollen wir ein algebraisches Kriterium für die Optimalität einer zulässigen Lösung herleiten und darauf aufbauend das Simplex-Verfahren einführen. Der hier dargestellte Zugang ist angelehnt an das Lehrbuch von Hamacher und Klamroth ([HK04]). Dieses auf englisch und deutsch bilingual verfasste Buch ist insbesondere zu empfehlen, wenn man zum zukünftigen Umgang mit der englischsprachigen Fachliteratur parallel die entsprechenden englischen Begriffe lernen möchte.

Nach Lemma 2.4 reicht es, wenn wir uns hierbei auf lineare Optimierungsprobleme in Standardform beschränken, d.h. wir betrachten ein lineares Programm der Form

$$\begin{array}{rcl} \min & c^t x & \\ \text{s.d. } Ax & = & b \\ x & \geq & 0 \end{array}$$

mit einer $m \times n$-Matrix A und Vektoren $b \in \mathbb{R}^m$, $c \in \mathbb{R}^n$. Zusätzlich nehmen wir in Abschnitt 2.3 und Abschnitt 2.4 durchgehend an, dass $n \geq m$ gilt und dass die Matrix A $\text{Rang}(A) = m$ erfüllt, also vollen Zeilenrang hat. Das können wir deshalb ohne Beschränkung der Allgemeinheit tun, weil im Fall von $\text{Rang}(A) < m$ die Zeilen von A linear abhängig sind und wir durch Entfernen der abhängigen Gleichungen ein äquivalentes Gleichungssystem erhalten, das dann vollen Zeilenrang $m \leq n$ hat und somit beiden oben genannten Forderungen genügt.

Wir benötigen noch einige Bezeichnungen und Definitionen. Sei dazu A eine $m \times n$ Matrix. Wir bezeichnen mit $A_1, \ldots, A_n \in \mathbb{R}^m$ die Spalten von A. Für die Zeilen von A verwenden wir $A^1, \ldots, A^m$, wobei $(A^1)^t, \ldots, (A^m)^t \in \mathbb{R}^n$.

Definition 2.11 *Sei A eine $m \times n$-Matrix mit $m \leq n$ und Rang(A)=m. Eine Indexmenge $B = \{B(1) \cdots B(m)\} \subseteq \{1, 2, \ldots, n\}$ heißt* **Basis** *von A, falls ihre zugehörige Spaltenmenge $\{A_{B(1)} \cdots A_{B(m)}\}$ linear unabhängig ist. (Ungenauerweise wird auch die entsprechende Spaltenmenge oft als* **Basis von A** *bezeichnet.)*

Notation 2.12 *Sei B eine Basis der $m \times n$-Matrix A.*

- $N = \{N(1), \cdots, N(n-m)\} = \{j : j \notin B\}$ *bezeichnet die Menge der* **Nicht-Basisindizes**.
- $A_B = (A_{B(1)}, \cdots, A_{B(m)})$ *heißt der* **Basisanteil von A**, *analog bezeichnet $A_N = (A_{N(1)}, \cdots, A_{N(n-m)})$ den* **Nichtbasisanteil von A**.

- *Die Variablen* $x_B = (x_{B(1)}, \cdots, x_{B(m)})$ *nennt man* **Basisvariablen**, *die übrigen Variablen* $x_N = (x_{N(1)}, \cdots, x_{N(n-m)})$ **Nichtbasisvariablen**.

Von den Dimensionen her gilt entsprechend $A_B \in \mathbb{R}^{m,m}$ und $A_N \in \mathbb{R}^{m,n-m}$. Weiterhin wissen wir, dass die Inverse A_B^{-1} zu A_B existiert, weil A_B aus m unabhängigen Spalten besteht und somit regulär ist.

Lemma 2.13 *Sei* $x = (x_B, x_N)$. *Dann gilt* $Ax = b$ *genau dann, wenn*

$$x_B = A_B^{-1}b - A_B^{-1}A_N x_N. \tag{2.1}$$

Diese Umformulierung zu (2.1) werden wir im folgenden oft verwenden. Man nennt sie die **Basisdarstellung von x bezüglich B**.

Beweis:

$$b = A \cdot x = (A_B|A_N)\begin{pmatrix} x_B \\ x_N \end{pmatrix} = A_B x_B + A_N x_N$$
$$\iff x_B = A_B^{-1}b - A_B^{-1}A_N x_N$$

QED

Für beliebige Werte von x_N erhält man durch die Basisdarstellung die für x_B festgelegten Werte, so dass die Lösung $x = (x_B, x_N)$ die Bedingung $Ax = b$ erfüllt. Durch die Festlegung der Nichtbasisvariablen sind also die Werte der Basisvariablen aufgrund der Regularität von A_B eindeutig bestimmt.

Definition 2.14 *Sei* B *eine Basis der Matrix* A. *Ein Paar von Basis und Nichtbasisvariablen gegeben durch* $x = (x_B, x_N)$ *heißt* **Basislösung** *des Systems* $Ax = b$ *wenn* $x_N = 0$ *und* $x_B = A_B^{-1}b$ *gilt. Weiter nennt man eine Basislösung* **zulässig**, *wenn* $x_B \geq 0$ *gilt.*

Es ist wichtig, sich hier direkt klar zu machen, dass die zu B gehörende Basislösung x genau dann eine zulässige Lösung des Optimierungsproblems

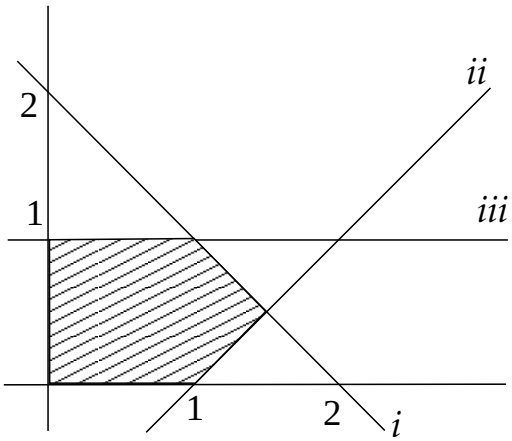
$$\min\{c^t x : Ax = b, x \geq 0\}$$

ist, wenn $x_B \geq 0$ gilt, sie also nach der eben eingeführten Definition eine *zulässige Basislösung* ist.

Beispiel 2.3 *Wir betrachten das folgende lineare Optimierungsproblem:*

$$\begin{aligned} \min \ & -x_1 + 2x_2 \\ \text{s.d. } & x_1 + x_2 \leq 2 \\ & x_1 - x_2 \leq 1 \\ & x_2 \leq 1 \\ & x_1, x_2 \geq 0 \end{aligned}$$

Graphisch lässt sich das Problem in zwei Variablen durch folgende Skizze veranschaulichen:



Bevor wir Basislösungen bestimmen können, formulieren wir das lineare Programm in Standardform.

$$\begin{aligned} \min \ & -x_1 + 2x_2 \\ \text{s.d. } & x_1 + x_2 + x_3 = 2 \\ & x_1 - x_2 + x_4 = 1 \\ & x_2 + x_5 = 1 \\ & x_1, x_2, x_3, x_4, x_5 \geq 0 \end{aligned}$$

Die Vektoren c, b und die Matrix A ergeben sich also wie folgt:

$$c^t = (-1, 2, 0, 0, 0)$$

$$A = \begin{pmatrix} 1 & 1 & 1 & 0 & 0 \\ 1 & -1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 \end{pmatrix}, b = \begin{pmatrix} 2 \\ 1 \\ 1 \end{pmatrix}.$$

Wir schauen uns nun einige Basislösungen an.

1. $B = (3, 4, 5)$

$$x_B = A_B^{-1} b = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 2 \\ 1 \\ 1 \end{pmatrix} = \begin{pmatrix} 2 \\ 1 \\ 1 \end{pmatrix} \quad x_N = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$$

$$\Rightarrow x = (0, 0, 2, 1, 1)^t \geq 0 \text{ zulässig.}$$

Der Zielfunktionswert von x beträgt $c^t x = (-1, 2, 0, 0, 0) \cdot x = 0$.

2. $B = (1, 2, 3)$

$$x_B = A_B^{-1} b = \begin{pmatrix} 0 & 1 & 1 \\ 0 & 0 & 1 \\ 1 & -1 & -2 \end{pmatrix} \begin{pmatrix} 2 \\ 1 \\ 1 \end{pmatrix} = \begin{pmatrix} 2 \\ 1 \\ -1 \end{pmatrix} \qquad x_N = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$$

$\Rightarrow x = (2, 1, -1, 0, 0)^t$ *keine zulässige Basislösung.*
Entsprechend ist x *nicht zulässig für das Optimierungsproblem.*

3. *Die Indexmenge* $B = (1, 3, 4)$ *ist keine Basis, weil die entsprechende Matrix*

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

einen Rang von $2 < m = 3$ *hat und somit nicht regulär ist. In diesem Fall existiert* A_B^{-1} *also nicht.*

4. $B = (1, 4, 5)$ *diskutieren wir ohne die Inverse der entsprechenden Matrix* A_B *zu bestimmen: Wir wissen, dass die Nichtbasisvariablen* x_2 *und* x_3 *beide Null sind. Daraus können wir die Lage der Lösung in der Skizze schon bestimmen: Die Lösung liegt auf der* x_1*-Achse (weil* $x_2 = 0$*) und auf der zur ersten Nebenbedingung gehörenden Gerade (weil* $x_3 = 0$ *und* x_3 *die Schlupfvariable dieser Nebenbedingung ist). Durch Schnitt der Gerade* $x_1 + x_2 = 2$ *mit der Gerade* $x_2 = 0$ *ergibt sich also* $x_1 = 2$*. Dass* $x_2 = x_3 = 0$ *gilt, war schon klar, weil* $N = \{2, 3\}$*. Die fehlenden Variablen lassen sich nun durch die zweite und dritte Nebenbedingung bestimmen:*

$$\begin{aligned} x_4 &= 1 - x_1 + x_2 = -1, \\ x_5 &= 1 - x_2 = 1 \end{aligned}$$

und die vollständige Basislösung x *in allen Variablen ist somit gegeben durch*

$$x = (2, 0, 0, -1, 1)^t.$$

Wie man leicht schon an dem Schnittpunkt der ersten Nebenbedingung mit der x_1*-Achse in der Zeichnung erkannt hat, sieht man nun auch an der Lösung, dass* $x_B \geq 0$ *nicht gilt und* x *somit unzulässig ist.*

5. $B = (1, 3, 5)$

$$x_B = A_B^{-1} b = \begin{pmatrix} 0 & 1 & 0 \\ 1 & -1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 2 \\ 1 \\ 1 \end{pmatrix} = \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix} \qquad x_N = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$$

$\Longrightarrow \quad x = (1, 0, 1, 0, 1)^t$ *zulässig.*
Der Zielfunktionswert für x *ergibt sich als* $c^t x = (-1, 2, 0, 0, 0) \cdot x = -1$*.*

Die beste der betrachteten Basislösungen ist also die durch die Basis $B = \{1, 3, 5\}$ definierte. Wir werden auf S. 27 zeigen, dass sie optimal ist.

Zum Abschluss des Beispiels wollen wir nun noch die vorhin angesprochene Abhängigkeit der Basisvariablen von der Wahl der Nichtbasisvariablen zeigen. Wählen wir dazu noch einmal die Basis $B = (3, 4, 5) = (B(1), B(2), B(3))$ und legen x_N fest durch

$$x_N = \begin{pmatrix} 0.5 \\ 0.5 \end{pmatrix}.$$

Dann lassen sich gemäß der Formel (2.1) die Werte für x_B berechnen durch:

$$x_B = A_B^{-1}b - A_B^{-1}A_N x_N = \begin{pmatrix} 2 \\ 1 \\ 1 \end{pmatrix} - \begin{pmatrix} 1 & 1 \\ 1 & -1 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} 0.5 \\ 0.5 \end{pmatrix} = \begin{pmatrix} 2 \\ 1 \\ 1 \end{pmatrix} - \begin{pmatrix} 1 \\ 0 \\ 0.5 \end{pmatrix} = \begin{pmatrix} 1 \\ 1 \\ 0.5 \end{pmatrix}$$

$$\begin{aligned} \text{d.h.} \quad x_{B(1)} &= x_3 = 1 \\ x_{B(2)} &= x_4 = 1 \\ x_{B(3)} &= x_5 = \tfrac{1}{2} \\ \Rightarrow \quad x &= (\frac{1}{2}, \frac{1}{2}, 1, 1, \frac{1}{2})^t \end{aligned}$$

Die folgende graphische Deutung der Beispiele sollten wir zur Erleichterung der Anschauung im Auge behalten.

1. Basislösungen entsprechen Schnittpunkten von n Restriktionen.
2. Zulässige Basislösungen entsprechen Ecken des zulässigen Bereiches.

Im nächsten Abschnitt werden wir der Frage nachgehen, wie sich erkennen lässt, ob eine zulässige Basislösung optimal ist.

2.3.2 Optimalitätskriterium

Sei B eine Basis der Matrix A. Wir partitionieren den Kostenvektor $c = (c_B, c_N) \in \mathbb{R}^n$ in einen Basisteil $c_B = (c_{B(1)}, \ldots, c_{B(m)})^t \in \mathbb{R}^m$ und einen Nichtbasisteil $c_N = (c_{N(1)}, \ldots, c_{N(n-m)})^t \in \mathbb{R}^{n-m}$ und nutzen die Basisdarstellung (2.1) von x, um die Zielfunktion $c^t x$ folgendermaßen umzuschreiben:

$$\begin{aligned} c^t x &= c_B^t x_B + c_N^t x_N \\ &= c_B^t(A_B^{-1}b - A_B^{-1}A_N x_N) + c_N^t x_N \\ &= c_B^t A_B^{-1} b + (c_N^t - c_B^t A_B^{-1} A_N) x_N \end{aligned} \tag{2.2}$$

Diese Umformulierung (2.2) gibt uns für eine Basis B den Wert der Zielfunktion $c^t x$ in Abhängigkeit der Nichtbasisvariablen x_N an. Wie man die Zielfunktion durch Veränderung der Nichtbasisvariablen verändern kann, wird also durch den Ausdruck $c_N^t - c_B^t A_B^{-1} A_N$ bestimmt. Das motiviert die folgende Definition.

Notation 2.15 *Sei B eine Basis von A. Als* **reduzierte Kosten** *der Nichtbasisvariablen $x_{N(j)}$ bezeichnet man*

$$\bar{c}_{N(j)} = c_{N(j)} - c_B^t A_B^{-1} A_{N(j)} = c_{N(j)} - \pi A_{N(j)}, j = 1, \ldots, n-m.$$

Dabei haben wir die Notation $\pi = c_B^t A_B^{-1} \in \mathbb{R}^m$ eingeführt, die später im Zusammenhang mit der Dualität linearer Programme noch wichtig werden wird, siehe Abschnitt 2.5. Wir können nun folgendes hinreichendes Kriterium für die Optimalität einer Basislösung angeben.

Satz 2.16 (Optimalitätskriterium für Basislösungen) *Ist x eine zulässige Basislösung bezüglich einer Basis B von A und gilt*

$$\bar{c}_{N(j)} = c_{N(j)} - c_B^t A_B^{-1} A_{N(j)} \geq 0 \quad \forall j = 1, \cdots, n-m$$

so ist x eine Optimallösung des linearen Programms $\min\{c^t x | Ax = b, x \geq 0\}$.

Beweis: Nach (2.2) ist eine Basislösung $x = (A_B^{-1} b, 0)$ optimal, wenn $x_N = 0$ die Optimierungsaufgabe

$$\min\{c_B^t A_B^{-1} b + (c_N^t - c_B^t A_B^{-1} A_N) x_N | x_N \geq 0\}$$

löst, d.h. wenn $x_N = 0$ optimal ist für

$$\min\{(c_N^t - c_B^t A_B^{-1} A_N) x_N | x_N \geq 0\}$$

also wenn $(c_{N(j)} - c_B^t A_B^{-1} A_{N(j)}) \geq 0 \quad \forall j \in \{1, \cdots, n-m\}$. QED

Sind die reduzierten Kosten aller Nichtbasisvariablen größer oder gleich Null, so ist die Optimalität der Lösung gewährleistet. Allerdings ist dieses Optimalitätskriterium nur hinreichend, nicht aber notwendig. Dementsprechend gilt der umgekehrte Fall nicht: Es gibt lineare Programme in denen reduzierte Kosten echt kleiner als Null auftreten, aber die entsprechende Basislösung dennoch optimal ist. Man darf im allgemeinen für eine optimale Basislösung x also nicht folgern, dass $\bar{c}_{N(j)} \geq 0$. (Beispiel dafür in ÜBUNG.)

Beispiel 2.4 *Wir kommen wieder auf Beispiel 2.3 (S. 22) zurück und untersuchen die beiden dort berechneten (zulässigen) Basislösungen.*

- *Ist die Basis $B = (3, 4, 5)$ mit Basislösung $x = (0, 0, 2, 1, 1)^t$ optimal? Wegen*

  $$c_B^t \cdot A_B^{-1} = (0, 0, 0) \cdot A_B^{-1} = (0, 0, 0)$$

  ergeben sich die reduzierten Kosten für die beiden Nichtbasisvariablen x_1 und x_2 als

  $$\begin{aligned} \bar{c_1} &= c_1 - c_B^t A_B^{-1} A_1 = c_1 = -1 \\ \bar{c_2} &= c_2 - c_B^t A_B^{-1} A_2 = c_2 = 2; \end{aligned}$$

das Optimalitätskriterium ist also nicht anwendbar. Weil die Basis $\{1,3,5\}$ *nach unseren vorhergehenden Berechnungen aber zu einem besseren Zielfunktionswert führt, wissen wir dennoch, dass* $x = (0,0,2,1,1)^t$ *nicht optimal ist.*

- *Untersuchen wir nun* $B = (1,3,5)$ *mit zugehöriger Basislösung* $x = (1,0,1,0,1)$. *Wiederum bestimmen wir zunächst*

$$c_B^t \cdot A_B^{-1} = (-1,0,0) \cdot \begin{pmatrix} 0 & 1 & 0 \\ 1 & -1 & 0 \\ 0 & 0 & 1 \end{pmatrix} = (0,-1,0)$$

mit dessen Hilfe wir nun die reduzierten Kosten berechnen können. Wir erhalten

$$\begin{aligned} \bar{c_2} &= c_2 - c_B^t A_B^{-1} A_2 \\ &= 2 - (0,-1,0) \cdot \begin{pmatrix} 1 \\ -1 \\ 1 \end{pmatrix} = 1 \geq 0 \\ \bar{c_4} &= c_4 - c_B^t A_B^{-1} A_4 \\ &= 0 - (0,-1,0) \cdot \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix} = 1 \geq 0, \end{aligned}$$

die beide größer als Null sind. Nach dem Optimalitätskriterium aus Satz 2.16 ist x *also optimal.*

2.3.3 Unbeschränktheit und Basiswechsel

Es soll nun untersucht werden, wie man eine nicht optimale Basis verbessern kann. Im ersten Fall vom Beispiel 2.3 sind die reduzierten Kosten für die Nichtbasisvariable x_1 negativ, $\bar{c_1} = -1$. Daher wird die Zielfunktion verbessert, wenn man x_1 erhöht. Aber wie weit darf man x_1 erhöhen?

Etwas allgemeiner formuliert, beschäftigen wir uns also mit folgender Idee. Wenn das Optimalitätskriterium für eine Basislösung nicht erfüllt ist, erhöhen wir eine Nichtbasisvariable $x_{N(s)}$ mit negativen reduzierten Kosten $\bar{c}_{N(s)} < 0$. Je weiter wir $x_{N(s)}$ erhöhen, desto kleiner wird $c^t x$. Aber wie weit darf $x_{N(s)}$ erhöht werden?

Dazu verwenden wir wieder die Basisdarstellung (2.1) von x bezüglich B:

$$x_B = A_B^{-1} b - A_B^{-1} A_N x_N$$

Sei nun $\bar{c}_{N(s)} < 0$. Wir konstruieren eine neue Lösung $\tilde{x}$ durch

$$\begin{aligned} \tilde{x}_{N(j)} &= 0 \quad \forall j \neq s \\ \tilde{x}_{N(s)} &= \delta \geq 0 \end{aligned}$$

und wollen herausfinden, wie groß δ werden darf.

Damit die resultierende Lösung $(\tilde{x}_B, \tilde{x}_N)$ zulässig ist, muss gelten:

$$\tilde{x}_B = A_B^{-1}b - A_B^{-1}A_N\tilde{x}_N = A_B^{-1}b - A_B^{-1}A_{N(s)}\delta \geq 0.$$

Wir verwenden die folgende Notation:

Notation 2.17 *Wir bezeichnen die i-te Komponente von $A_B^{-1}b$ mit $\tilde{b}_i$ und die i-te Komponente von $A_B^{-1}A_{N(s)}$ mit $\tilde{a}_{iN(s)}$.*

Dann ist die resultierende Lösung $(\tilde{x}_B, \tilde{x}_N)$ zulässig, genau dann, wenn

$$\tilde{x}_{B(i)} = \tilde{b}_i - \tilde{a}_{iN(s)} \cdot \delta \geq 0 \ \forall i = 1, \cdots, m.$$

Für solche i mit $\tilde{a}_{iN(i)} \leq 0$ ist obige Bedingung immer erfüllt (weil die Komponenten von $\tilde{b} = A_B^{-1}b$ den Werten x_B in der zulässigen Basislösung entsprechen und somit positiv sind). Gilt dagegen $\tilde{a}_{iN(i)} > 0$ so kann man die Bedingung nach δ auflösen und es ergibt sich $\delta \leq \frac{\tilde{b}_i}{\tilde{a}_{iN(s)}}$. Da δ so groß wie möglich werden soll, wählen wir das Maximum aller erlaubten Werte für δ. Es ergibt sich entsprechend als

$$\delta = x_{N(s)} := \min_{i=1,\cdots,m} \{\frac{\tilde{b}_i}{\tilde{a}_{iN(s)}} : \tilde{a}_{iN(s)} > 0\}. \quad (\textbf{Quotientenregel}) \tag{2.3}$$

Wir untersuchen im folgenden nun zwei Fälle, nämlich den Fall, dass alle $\tilde{a}_{iN(s)} \leq 0$ sind (und damit die Quotientenregel zu einem unbeschränkten Minimierungsproblem führt) und den Fall, in dem mindestens eines der $\tilde{a}_{iN(s)} > 0$ ist. Wir beginnen mit dem unbeschränkten Fall.

Fall 1: $\forall i = 1, \cdots, m \ : \tilde{a}_{iN(s)} \leq 0$
In diesem Fall kann δ nach den vorstehenden Überlegungen beliebig groß gemacht werden, ohne $x_B \geq 0$ zu verletzen, und es ergibt sich der folgende Satz.

Satz 2.18 *Sei x eine zulässige Basislösung bezüglich B und gelte:*

$$\exists N(s) \in N : \quad \bar{c}_{N(s)} < 0 \ \textit{und} \ A_B^{-1}A_{N(s)} \leq 0$$

Dann ist das LP $\min\{c^t x : \ Ax = b, \ x \geq 0\}$ unbeschränkt.

Beweis: Definiere die neue Lösung $\tilde{x}$ wie eben entwickelt, d.h.

$$\begin{aligned} \tilde{x}_{N(j)} &= 0 \ \forall j \neq s \\ \tilde{x}_{N(s)} &= \delta \geq 0 \\ \tilde{x}_B &= A_B^{-1}b - A_B^{-1}A_{N(s)}\delta. \end{aligned} \tag{2.4}$$

Wie eben bereits besprochen, gilt $A\tilde{x} = b$ zulässig und $\tilde{x}_N \geq 0$ nach Konstruktion. Für die Basisvariablen erhält man

$$\tilde{x}_B = \underbrace{A_B^{-1}b}_{\geq 0} - \underbrace{A_B^{-1}A_{N(s)}}_{\leq 0}\underbrace{\delta}_{\geq 0} \geq 0$$

Für die Zielfunktion gilt nach (2.2):

$$c^t\tilde{x} = c_B^t A_B^{-1}b + \bar{c}_N^t\tilde{x}_N = \underbrace{c_B^t A_B^{-1}b}_{konstant} + \underbrace{\bar{c}_{N(s)}}_{<0}\cdot\delta.$$

Das heißt, $c^t\tilde{x} \longrightarrow -\infty$ für $\delta \to \infty$, also ist das vorliegende lineare Programm unbeschränkt. QED

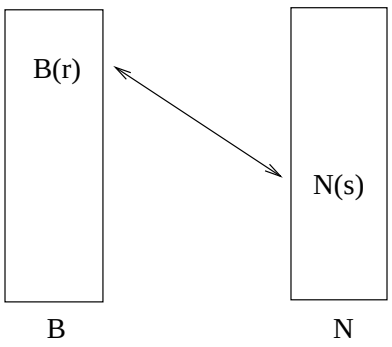
Fall 2: $\exists i \in \{1, \ldots, m\}: \tilde{a}_{iN(s)} > 0$
Wenden wir uns nun dem anderen Fall zu, in dem mindestens eine Komponente des Vektors $A_B^{-1}A_{N(s)}$ echt größer als Null ist. Hier erhöhen wir die fragliche Nichtbasisvariable von Null auf δ, wobei wir das entsprechende δ nach der Quotientenregel (2.3) wählen. Das Ergebnis ist (nach der Konstruktion von δ) zulässig und führt zu einer neuen Basislösung bezüglich einer zu B benachbarten Basis. Dieses Ergebnis wird im folgenden Satz zusammengefasst und formal bewiesen.

Satz 2.19 *Sei B eine Basis und $\bar{c}_{N(s)} < 0$, $\tilde{a}_{iN(s)} > 0$ für mindestens ein $i \in \{1, \cdots, m\}$. Sei weiterhin*

$$\delta = \frac{\tilde{b}_r}{\tilde{a}_{rN(s)}} = \min\{\frac{\tilde{b}_i}{\tilde{a}_{iN(s)}} : \tilde{a}_{iN(s)} > 0, i = 1, \cdots, m\}.$$

Dann ist $B' = B\backslash\{B(r)\} \cup \{N(s)\}$ eine Basis mit nicht schlechterem Zielfunktionswert und das in (2.4) definierte $\tilde{x}$ die zugehörige Basislösung.

Die im Satz beschriebene Transformation von einer Basis B zu einer benachbarten nicht-schlechteren Basis B' nennt man **Basiswechsel**. Die folgende Graphik veranschaulicht, dass eine Basisvariable $B(r)$ die Basis verlässt und die Nichtbasisvariable $N(s)$ dafür zur Basisvariablen wird.



Beweis: Nach der Quotientenregel gilt $\delta = \frac{\tilde{b}_r}{\tilde{a}_{rN(s)}}$ für ein $r \in \{1, \ldots, m\}$. Als neue Lösung ergibt sich folglich

$$\begin{aligned}
\tilde{x}_{N(s)} &= \frac{\tilde{b}_r}{\tilde{a}_{rN(s)}} \\
\tilde{x}_{N(j)} &= 0 \text{ für alle } j \neq s \\
\tilde{x}_{B(i)} &= \tilde{b}_i - \tilde{a}_{iN(s)} \cdot \tilde{x}_{N(s)} \\
&= \tilde{b}_i - \tilde{a}_{iN(s)} \cdot \frac{\tilde{b}_r}{\tilde{a}_{rN(s)}} \text{ für } i = 1, \cdots, m
\end{aligned}$$

Insbesondere gilt, dass $\tilde{x}_{B(r)} = 0$ ist. Die neue Lösung hat also erneut $m - n$ Variablen mit dem Wert Null, nämlich alle $x_{N(i)}$ für $i \neq s$, und $x_{B(r)}$. Wenn die Spalten der Nicht-Null-Variablen $B \setminus \{B(r)\} \cup \{N(s)\}$ also linear unabhängig sind, ist eine neue Basislösung erreicht. (Lineare Unabhängigkeit der Spalten: ÜBUNG)

Die neue Lösung $\tilde{x}$ ist zulässig, da $A\tilde{x} = b$ (Lemma 2.13), $\tilde{x}_N \geq 0$ nach Konstruktion, und $\tilde{x}_B \geq 0$ aufgrund der Quotientenregel (siehe (2.3)). Schließlich gilt für den Zielfunktionswert der neuen Lösung,

$$\begin{aligned}
c^t\tilde{x} &= c_B^t A_B^{-1} b + \bar{c}_N^t \tilde{x}_N \\
&= c_B^t x_B + \bar{c}_{N(s)} \cdot \delta \\
&= c^t x + \underbrace{\bar{c}_{N(s)}}_{<0} \cdot \underbrace{\delta}_{\geq 0} \\
&\leq c^t x.
\end{aligned}$$

QED

Es ist zu beachten, dass das Minimum bei der Quotientenregel im allgemeinen nicht eindeutig ist; der Satz gilt aber unabhängig davon, welches Minimum gewählt wird. Zum Abschluss zeigen wir einen Basiswechsel an dem in diesem Abschnitt schon mehrmals verwendeten Beispiel.

Beispiel 2.5 *Wir betrachten noch einmal das Beispiel 2.3 (siehe Seite 22 und Seite 26) und wählen die Basis* $B = \{3, 4, 5\}$. *Es sind also*

$$\begin{aligned}
B(1) &= 3 \\
B(2) &= 4 \\
B(3) &= 5
\end{aligned}$$

Wie schon auf Seite 26 bestimmt, gilt

$$\bar{c}_1 = -1 < 0,$$

wir wählen daher $s = 1$ und versuchen, den Wert der Nichtbasisvariable $x_N(s) = x_{N(1)} = x_1$ zu erhöhen, so dass x_1 zu einer Basisvariablen wird. Dazu berechnen wir zunächst die Werte $\tilde{a}_{i1}$ als jeweils i.te Komponente von dem Vektor

$$A_B^{-1} A_1 = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 \\ 1 \\ 0 \end{pmatrix} = \begin{pmatrix} 1 \\ 1 \\ 0 \end{pmatrix}$$

sowie $\tilde{b}_i$ als i-te Komponente des Vektors

$$A_B^{-1} b = \begin{pmatrix} 2 \\ 1 \\ 1 \end{pmatrix}.$$

Nach der Quotientenregel ergibt sich δ entsprechend als

$$\delta = \min\{\frac{2}{1}, \frac{1}{1}\} = 1,$$

d.h. $r = 2$. Von Satz 2.19 wissen wir, dass die Basisvariable $x_{B(2)} = x_4$ die Basis verlassen wird. Als neue Lösung $\tilde{x}$ erhält man

$$\begin{aligned} \tilde{x}_1 &= \delta = 1 \\ \tilde{x}_2 &= 0 \\ \tilde{x}_B &= A_B^{-1} b - A_B^{-1} A_{N(s)} \cdot \delta = \begin{pmatrix} 2 \\ 1 \\ 1 \end{pmatrix} - \begin{pmatrix} 1 \\ 1 \\ 0 \end{pmatrix} \delta = \begin{pmatrix} 1 \\ 0 \\ 1 \end{pmatrix}. \end{aligned}$$

$$\Longrightarrow \quad \tilde{x} = (1, 0, 1, 0, 1)^t$$

Die neue Basis B' ergibt sich als

$$B' = B \backslash \{B(r)\} \cup \{N(s)\} = B \backslash \{4\} \cup \{1\} = \{1, 3, 5\}.$$

Aus Satz 2.19 lässt sich leicht die Grundidee des Simplex-Verfahrens ableiten, das wir in Abschnitt 2.4.2 einführen werden: Das Simplex-Verfahren startet mit einer zulässigen Basislösung. In jedem Simplex-Schritt wechselt man von einer zulässigen Basislösung zu einer benachbarten, nicht schlechteren zulässigen Basislösung, bis man schließlich eine optimale Basislösung erreicht.

Um so ein Vorgehen zu rechtfertigen, werden wir im folgenden Abschnitt zunächst zeigen, dass es reicht, nur die Basislösungen zu untersuchen. Genauer beweisen wir, dass es unter allen Optimallösungen eines (lösbaren) linearen Programms auch immer eine Basislösung gibt.

2.3.4 Der Hauptsatz der linearen Optimierung

In diesem Abschnitt werden wir zeigen, dass wir bei der Suche nach einer Optimallösung eines linearen Programms nur Basislösungen untersuchen müssen. Dazu gehen wir in zwei Schritten vor.

1. Wir zeigen, dass jedes lineare Programm, das überhaupt eine zulässige Lösung besitzt, auch eine zulässige Basislösung hat.

2. Wir zeigen, dass jedes LP, das eine optimale Lösung hat, auch eine optimale Basislösung hat.

Beginnen wir mit dem ersten Teil. Natürlich hat jedes lineare Programm in Standardform eine Basislösung, weil wir als Generalvoraussetzung gefordert haben, dass die Koeffizientenmatrix A vollen Rang und damit m linear unabhängige Spalten (also eine Basis) hat. Allerdings ist damit noch nicht klar, ob es auch immer eine zulässige Basislösung gibt. Im nächsten Satz zeigen wir, dass das tatsächlich der Fall ist, falls es überhaupt eine zulässige Lösung gibt.

Satz 2.20 *Sei A eine $m \times n$ Matrix mit $Rang(A) = m \leq n$ und sei $b \in \mathbb{R}^m$. Weiter sei $P = \{x \in \mathbb{R}^n : Ax = b, x \geq 0\} \neq \emptyset$. Dann gibt es für jedes $c \in \mathbb{R}^n$ eine zulässige Basislösung des linearen Programms $\min\{c^t x : x \in P\}$.*

Beweis: Weil $P \neq \emptyset$, gibt es $x \in P$. Nehmen wir an, x hat k positive Komponenten, und sei ohne Beschränkung der Allgemeinheit die Nummerierung der Variablen so, dass

$$\begin{aligned} x_j &> 0 \text{ für } j = 1, \ldots, k \\ x_j &= 0 \text{ für } j = k+1, \ldots, n. \end{aligned}$$

Wir werden in diesem Beweis ausgehend von der Lösung x *konstruktiv* eine zulässige Basislösung erzeugen.

Fall 1: $\{A_1, \cdots, A_k\}$ sind linear unabhängig. Wegen $Rang(A) = m$ gilt $k \leq m$ und die Spalten $A_1, \ldots, A_m$ können durch $m - k$ Spalten zu einer Basis B ergänzt werden. Die zulässige Lösung $x \in P$ ist dann die Basislösung bezüglich dieser Basis, weil alle Nichtbasisvariablen Null sind.

Fall 2: $\{A_1, \cdots, A_k\}$ sind linear abhängig. In diesem Fall ist x keine Basislösung. Wir werden eine neue Lösung x' konstruieren, die nur noch $k - 1$ positive Komponenten hat, so dass wir iterieren können, bis Fall 1 eintritt.

Bezeichne $A_K = (A_1, \cdots, A_k)$ die $m \times k$ Matrix, die die ersten k Spalten der Matrix A enthält und sei $x_K = (x_1, \cdots, x_k)^t$. Weil

$$x_{k+1} = x_{k+2} = \cdots = x_n = 0$$

und x zulässig, gilt

$$A_K x_K = A_K x_K + (A_{k+1}, \cdots, A_n) \begin{pmatrix} x_{k+1} \\ \vdots \\ x_n \end{pmatrix} = Ax = b. \qquad (2.5)$$

Weiterhin sind $\{A_1, \cdots, A_k\}$ linear abhängig. Das heißt, es existiert ein $\alpha = (\alpha_1, \cdots, \alpha_k)^t \neq 0$ so dass $A_K \alpha = 0$. Es gilt also

$$A_K(\delta\alpha) = 0 \text{ für alle } \delta \in \mathbb{R}. \qquad (2.6)$$

Addiert man (2.5) und (2.6) so erhält man

$$A_K(x_K + \delta\alpha) = b \text{ für alle } \delta \in \mathbb{R}.$$

Wir definieren nun $x(\delta) \in \mathbb{R}^n$ mit Komponenten

$$x(\delta)_i = \begin{cases} x_i + \delta\alpha_i & \text{falls } i \in \{1, \cdots, k\} \\ 0 & \text{sonst.} \end{cases}$$

Wegen $Ax(\delta) = A_K(x_K + \delta\alpha) + 0 = b$ gilt dann

$$\begin{aligned} x(\delta) \in P \quad &\Longleftrightarrow \quad x(\delta)_i \geq 0 \ \ \forall i = 1, \ldots, n \\ &\Longleftrightarrow \quad \delta \geq -\frac{x_i}{\alpha_i} \ \ \forall i = 1, \ldots, k \ \ mit \ \ \alpha_i > 0 \\ &\text{und} \quad \delta \leq -\frac{x_i}{\alpha_i} \ \ \forall i = 1, \ldots, k \ \ mit \ \ \alpha_i < 0 \end{aligned}$$

Mindestens einer der beiden Werte $\delta = \max\{-\frac{x_i}{\alpha_i} : \ \alpha_i > 0\} < 0$ oder $\delta = \min\{-\frac{x_i}{\alpha_i} : \ \alpha_i < 0\} > 0$ existiert, da $\alpha \neq 0$. Sei ohne Beschränkung der Allgemeinheit

$$\bar{\delta} = -\frac{x_\tau}{\alpha_\tau} = \max\{-\frac{x_i}{\alpha_i} : \alpha_i > 0\} < 0.$$

Dann ist $x(\bar{\delta})$ zulässig, weil $\bar{\delta} \geq -\frac{x_i}{\alpha_i}$ für alle i mit $\alpha_i > 0$ und $\bar{\delta} < 0 \leq -\frac{x_i}{\alpha_i}$ für alle i mit $\alpha_i < 0$.

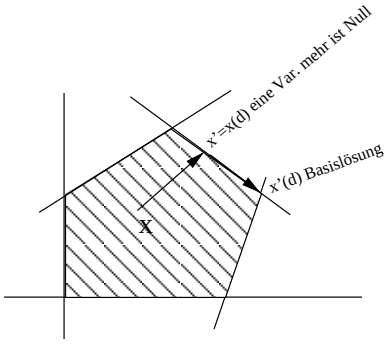
Weiterhin ist $x(\bar{\delta})_i = 0$ für $i = k+1, \ldots, n$ und

$$x(\bar{\delta})_\tau = x_\tau + \bar{\delta}\alpha_\tau = x_\tau + (-\frac{x_\tau}{\alpha_\tau})\alpha_\tau = 0,$$

also hat $\bar{x}(\bar{\delta})$ mindestens eine Komponente mit Wert Null mehr als x.

Wie zu Beginn des Beweises schon vermerkt, kann man dieses Verfahren nun für $x := x(\bar{\delta})$ wiederholen, bis die Spaltenvektoren der positiven Komponenten der aktuellen Lösung x linear unabhängig sind und man wie in Fall 1 eine zulässige Basislösung findet. QED

Anschaulich kann man sich das Verfahren wie in folgender Abbildung im zweidimensionalen angedeutet vorstellen: Man startet mit einer zulässigen Lösung x z.B. im Inneren von P, landet im nächsten Schritt dann auf einer Kante (indem die zugehörige Schlupfvariable auf Null gesetzt wurde) und schließlich in einer Ecke des Polyeders.



Der folgende Satz rechtfertigt nun endlich, dass wir im Simplex-Verfahren ausschließlich Basislösungen untersuchen, indem er zeigt, dass jedes zulässige lineare Programm eine optimale Lösung besitzt, die eine Basislösung ist.

Satz 2.21 (Hauptsatz der linearen Optimierung) *Sei A eine $m \times n$ Matrix mit $Rang(A) = m \leq n$ und sei $b \in \mathbb{R}^m$. Weiter sei $\mathcal{E}$ die Menge der Optimallösungen von $\min\{c^t x : Ax = b, x \in \mathbb{R}^n, x \geq 0\}$. Ist $\mathcal{E} \neq \emptyset$, dann gibt es eine Basislösung in $\mathcal{E}$.*

Beweis: Im Beweis verwenden wir das gleiche Vorgehen und die gleichen Notationen wie im vorhergehenden Beweis zu Satz 2.20. Sei dazu x optimal ($x \in \mathcal{E}$) und nehmen wir an, dass

$$\begin{aligned} x_j &> 0 \text{ für } j = 1, \ldots, k \\ x_j &= 0 \text{ für } j = k+1, \ldots, n. \end{aligned}$$

Wie im Beweis zu Satz 2.20 unterscheiden wir zwei Fälle:

Fall 1: $\{A_1, \ldots, A_k\}$ unabhängig. Dann ist x bereits eine Basislösung analog zum ersten Fall im Beweis zu Satz 2.20.

Fall 2: $\{A_1, \ldots, A_k\}$ linear abhängig, also existiert $\alpha \neq 0$ mit $A_K \alpha = 0$. Wie im Beweis zu Satz 2.20 definieren wir

$$x(\delta)_i = \begin{cases} x_i + \delta\alpha_i & \text{falls } i \in \{1, \cdots, k\}; \\ 0 & \text{sonst} \end{cases},$$

und wissen, dass

- $Ax(\delta) = b$

- $x(\delta) \geq 0$, wenn $|\delta|$ klein genug, genauer, für $\max\{-\frac{x_i}{\alpha_i} : \alpha_i > 0\} \leq \delta \leq \min\{-\frac{x_i}{\alpha_i} : \alpha_i < 0\}$,

also ist $x(\delta)$ für betragsmäßig kleine δ zulässig. Wie im vorhergehenden Beweis möchten wir δ so wählen, dass in $x(\delta)$ eine Variable mehr den Wert Null hat als in der aktuellen Lösung x, um so die Lösung nach und nach in eine Ecke des zulässigen Polyeders zu verschieben. Dabei dürfen wir aber den Zielfunktionswert nicht verschlechtern. Betrachten wir daher den Zielfunktionswert $c^t x(\delta)$:

$$c^t x(\delta) = c^t x + \delta \cdot (\alpha_1 c_1 + \cdots + \alpha_k c_k).$$

Angenommen, $\delta \cdot (\alpha_1 c_1 + \cdots + \alpha_k c_k) \neq 0$. Wäre $\delta \cdot (\alpha_1 c_1 + \cdots + \alpha_k c_k) > 0$, dann könnte man $c^t x$ durch Wahl eines $\delta < 0$ (echt) verbessern, im anderen Fall durch Wahl eines $\delta > 0$. Beide Fälle ergeben einen Widerspruch zu $x \in \mathcal{E}$. Wir haben also gezeigt, dass

$$\delta \cdot (\alpha_1 c_1 + \cdots + \alpha_k c_k) = 0, \tag{2.7}$$

oder $c^t x = c^t x(\delta)$. Wie im Beweis zum Satz 2.20 können wir $\delta = \bar{\delta} = -\frac{x_r}{a_r}$ also so wählen, dass $x(\delta)$ zulässig ist und nur noch $k-1$ positive Komponenten hat. Da $c^t x(\delta) = c^t x$ bleibt die konstruierte Lösung optimal und wir iterieren bis Fall 1 eintritt.

QED

2.4 Das Simplexverfahren

2.4.1 Das Simplextableau

Wie schon im letzten Abschnitt angedeutet, besteht das Simplex-Verfahren darin, sich durch Basiswechsel jeweils von einer Basislösung zu einer benachbarten besseren Basislösung zu bewegen, bis das Optimalitätskriterium (Satz 2.16 auf Seite 26) erfüllt ist oder das Problem als unbeschränkt erkannt wird (Satz 2.18 auf Seite 28). Um die dafür nötigen Basiswechsel (Abschnitt 2.3.3) effizient durchführen zu können und Optimalität und Unbeschränktheit schnell zu erkennen, führen wir nun zunächst so genannte *Simplextableaus* ein.

Wir wollen die Zielfunktion im Simplextableau analog zu den Nebenbedingungen behandeln. Dazu benötigen wir eine künstliche Variable z, die den Zielfunktionswert repräsentiert. Die Zielfunktion kann dann folgendermaßen geschrieben werden:

$$-z + c_1 x_1 + \cdots + c_n x_n = 0.$$

Diese Zielfunktion wird zusammen mit den Nebenbedingungen in eine Matrix geschrieben, die man das *Ausgangstableau* nennt. Die künstliche Variable z kommt nur in der Zielfunktion vor; daher besteht die erste Spalte aus einer 1 und ansonsten nur aus Nullen.

$$
\begin{array}{c}
\begin{array}{cccc} -z & x_1 & \cdots & x_n \end{array} \\
T = \left[\begin{array}{c|ccc|c} 1 & c_1 & \cdots & c_n & 0 \\ \hline 0 & a_{11} & \cdots & a_{1n} & b_1 \\ \vdots & \vdots & & \vdots & \vdots \\ 0 & a_{m1} & \cdots & a_{mn} & b_m \end{array}\right] = \left[\begin{array}{c|c|c} 1 & c^t & 0 \\ \hline 0 & A & \mathrm{b} \end{array}\right]
\end{array}
$$

Die Einträge im Ausgangstableau T bezeichnen wir mit t_{ij}, wobei der Zeilenindex i von 0 bis m läuft (die 0-te Zeile enthält die Koeffizienten der Zielfunktion) und der Spaltenindex j Werte von 0 bis $n+1$ annimmt. Das Ausgangstableau kann in ein Tableau für jede Basis B umgeschrieben werden.

Notation 2.22 *Ist B eine Basis, so bezeichnen wir mit T_B das Tableau, das nur die Spalten enthält, die zu der künstlichen Variable z und zu den Basisvariablen gehören, d.h.*

$$
T_B = \left(\begin{array}{c|c} 1 & c_B^t \\ \hline 0 & \\ \vdots & A_B \\ 0 & \end{array}\right),\ T_B \in \mathbb{R}^{(m+1)\times(m+1)}.
$$

Weil A_B nach Voraussetzung regulär ist, ist auch T_B regulär. Wie man durch Nachrechnen sofort sieht, hat die Inverse von T_B folgende Gestalt:

$$
T_B^{-1} = \left(\begin{array}{c|c} 1 & -c_B^t A_B^{-1} \\ \hline 0 & \\ \vdots & A_B^{-1} \\ 0 & \end{array}\right).
$$

Mit der Hilfe von T_B sind wir in der Lage, das Ausgangstableau T in die zur Basis B "passende" Form zu bringen.

Definition 2.23 *Ist B eine Basis, so ist*

$$
T(B) := T_B^{-1}T = \left(\begin{array}{c|c|c} 1 & c^t - c_B^t A_B^{-1} A & -c_B^t A_B^{-1} b \\ \hline 0 & & \\ \vdots & A_B^{-1}A & A_B^{-1}b \\ 0 & & \end{array}\right)
$$

das zu B gehörende **Simplextableau**.

Wozu braucht man nun dieses Darstellung $T(B)$? Zunächst bemerken wir, dass $T(B)$ dasselbe Gleichungssystem wie T repräsentiert, da die Multiplikation mit der regulären Matrix T_B^{-1} keinen Einfluss auf dessen Lösung hat. In der nun vorliegende Darstellung lassen sich aber die Werte der Variablen in der Basislösung B anhand der Einträge in der letzten Spalte von $T(B)$ ablesen! Das wird deutlich, wenn man sich die Einträge von $T(B)$ etwas näher anschaut.

- Die erste Spalte ist immer $(1, 0, \cdots, 0)^t$: Sie drückt den Gleichungscharakter der Zielfunktion aus. Da sie sich nie verändert, werden wir sie im folgenden vernachlässigen.

- Betrachten wir nun die Spalten, die zu den Basisvariablen gehören, z.B. die Spalte zur i-ten Basisvariablen, $j = B(i)$. Es gilt:

  - $A_B^{-1} A_j = e_i$,
  - $c_j - c_B^t A_B^{-1} A_j = c_j - \underbrace{c_B^t e_i}_{=c_{B(i)}} = 0.$

  Die zu einer Basis gehörenden Spalten erkennt man also daran, dass sie in der Zielfunktionszeile eine Null stehen haben und darunter ein Einheitsvektor folgt.

- Für die Spalte, die zur i-ten Nichtbasisvariablen gehört, also $j = N(i)$, gilt:

  - $A_B^{-1} A_j = (\tilde{a}_{1j}, \ldots, \tilde{a}_{mj})^t$
  - $c_j - c_B^t A_B^{-1} A_j = \bar{c}_j$.

  In den zu den Nichtbasisvariablen gehörenden Spalten kann man also die reduzierten Kosten (siehe Notation 2.15) und die in Notation 2.17 eingeführten Werte ablesen.

- Schließlich gilt für die letzte Spalte:

  - $A_B^{-1} b = x_B = \tilde{b}$
  - $-c_B^t A_B^{-1} b = -c_B^t x_B = -c^t x$

  Der negative Zielfunktionswert der Basislösung (x_B, x_N) lässt sich also in der Kostenzeile der letzten Spalte ablesen. Die Werte für die Basisvariablen $x_{B(1)}, \ldots, x_{B(m)}$ dieser Basislösung stehen in den Zeilen $1, \ldots, m$ der letzten Spalte.

Diese Beobachtungen sollen an dem schon mehrmals verwendeten Beispiel verdeutlicht werden.

Beispiel 2.6 *Betrachten wir nochmal Beispiel 2.3. Das Ausgangstableau in diesem Beispiel ist das folgende:*

$$T = \begin{array}{|c|cccccc|} \hline 1 & -1 & 2 & 0 & 0 & 0 & 0 \\ \hline 0 & 1 & 1 & 1 & 0 & 0 & 2 \\ 0 & 1 & -1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 1 & 1 \\ \hline \end{array}$$

Dieses Tableau ist schon ein Tableau bezüglich einer Basis, nämlich bezüglich der Basis $B = \{3, 4, 5\}$. Das erkennt man daran, dass die Spalten 3,4 und 5 den ersten, zweiten und dritten Einheitsvektor enthalten und in der Kostenzeile dieser Spalten jeweils eine Null steht. Da in der dritten Spalte der erste Einheitsvektor steht, gilt $B(1) = 3$ und der Wert der Basisvariablen $x_3 = x_{B(1)}$ lässt sich in der letzten Spalte ablesen, also $x_3 = x_{B(1)} = \tilde{b}_1 = t_{1n+1} = 2$. Für die anderen Werte der Basisvariablen ergibt sich entsprechend

$$\begin{aligned} x_4 &= x_{B(2)} = \tilde{b}_2 = t_{2n+1} = t_{26} = 1, \\ x_5 &= x_{B(3)} = \tilde{b}_3 = t_{3n+1} = t_{36} = 1. \end{aligned}$$

Der Zielfunktionswert der Basislösung ist $t_{0n+1} = t_{06} = 0$.

Jetzt formulieren wir das Ausgangstableau zum Tableau für die Basis $B = (1, 2, 4)$ um. Wir bestimmen

$$A_B = \begin{pmatrix} 1 & 1 & 0 \\ 1 & -1 & 1 \\ 0 & 1 & 0 \end{pmatrix}, \quad A_B^{-1} = \begin{pmatrix} 1 & 0 & -1 \\ 0 & 0 & 1 \\ -1 & 1 & 2 \end{pmatrix}$$

$$c_B^t A_B^{-1} = (-1, 2, 0) \begin{pmatrix} 1 & 0 & -1 \\ 0 & 0 & 1 \\ -1 & 1 & 2 \end{pmatrix} = (-1, 0, 3)$$

$$T_B^{-1} = \left(\begin{array}{c|ccc} 1 & 1 & 0 & -3 \\ \hline 0 & 1 & 0 & -1 \\ 0 & 0 & 0 & 1 \\ 0 & -1 & 1 & 2 \end{array}\right)$$

und können damit das Tableau $T(B)$ bilden als

$$T(B) = \begin{array}{|c|cccccc|l} \hline 1 & 0 & 0 & 1 & 0 & -3 & -1 & \\ \hline 0 & 1 & 0 & 1 & 0 & -1 & 1 & \leftarrow \tilde{b}_1 \\ 0 & 0 & 1 & 0 & 0 & 1 & 1 & \leftarrow \tilde{b}_2 \\ 0 & 0 & 0 & -1 & 1 & 2 & 1 & \leftarrow \tilde{b}_3 \\ \hline \end{array}$$

Um die zugehörige Basislösung abzulesen, machen wir uns zunächst die Basisvariablen und Nichtbasisvariablen klar:

Nichtbasisvariablen $N = \{3, 5\} \Rightarrow N(1) = 3,\ N(2) = 5$
Basisvariablen $B = \{1, 2, 4\} \Rightarrow B(1) = 1,\ B(2) = 2,\ B(3) = 4$
Die Werte der Basislösung lassen sich nun wie folgt ablesen:

$$\begin{aligned} x_1 &= x_{B(1)} = \tilde{b}_1 = t_{16} = 1, \\ x_2 &= x_{B(2)} = \tilde{b}_2 = t_{26} = 1, \\ x_4 &= x_{B(3)} = \tilde{b}_3 = t_{36} = 1. \end{aligned}$$

Da sie Nichtbasisvariablen sind, gilt $x_3 = x_5 = 0$. *Die Basislösung zur Basis* $B = (1, 2, 4)$ *ist also* $x = (1, 1, 0, 1, 0)^t$. *Den Zielfunktionswert der Lösung kann man entweder berechnen,*

$$c_1 \cdot x_1 + c_2 \cdot x_2 = -1 \cdot 1 + 2 \cdot 1 = 1,$$

oder direkt als das Negative von $t_{06} = -1$ *im Tableau ablesen.*

Bezüglich der Optimalität sieht man, dass beide Tableaus negative reduzierte Kosten in der 0-ten Zeile haben. Man kann also nicht folgern, dass eine der beiden Lösungen optimal ist und wird versuchen, sie durch Basiswechsel zu verbessern.

Die Berechnung der Inversen ist aufwändig, so dass die Bestimmung des Simplextableaus für eine Basis B anhand der Definition nicht zu empfehlen ist. Allerdings kann man ein Tableau zu einer Basis B auch ohne Kenntnis der Inversen bestimmen. Wie man das bei einem Basiswechsel effizient macht, soll im folgenden beschrieben werden. Wir betrachten dazu ein Tableau $T(B)$ zu einer Basis B, und nehmen an, dass das Optimalitätskriterium nicht erfüllt ist, d.h. dass ein $t_{0j} < 0$ für ein $j \in \{1, \ldots, n\}$ existiert. Eine Verbesserung durch einen Basiswechsel erscheint also möglich. Dazu soll die Spalte j in die Basis aufgenommen werden, d.h. wir wollen die derzeitige Nichtbasisvariable von $x_j = 0$ auf einen neuen Wert $x_j = \delta$ erhöhen. Dazu wenden wir die auf Seite 28 beschriebene Quotientenregel an:

$$\begin{aligned} \delta &= \min_{i=1,\ldots,m} \{\frac{\tilde{b}_i}{\tilde{a}_{ij}} : \tilde{a}_{ij} > 0\} \\ &= \min_{i=1,\ldots,m} \{\frac{t_{i(n+1)}}{t_{ij}} : t_{ij} > 0\}, \end{aligned}$$

die wir direkt anhand der Werte im Tableau durchführen können. Nach Satz 2.18 wissen wir, dass das Problem unbeschränkt ist, wenn alle $t_{ij} \leq 0$ sind. Nehmen wir also an, dass $\delta = \frac{t_{r(n+1)}}{t_{rj}}$. Das heißt, $B(r)$ soll die Basis verlassen.

Anstatt das Tableau zur neuen Basis $B' = B \backslash \{B(r)\} \cup \{j\}$ ausgehend vom Ausgangstableau T neu zu berechnen, führt man eine so genannte **Pivotoperation**

mit dem **Pivotelement** t_{rj} durch. Dabei verwandelt man durch elementare Zeilenoperationen (das heißt, durch Skalarmultiplikation und Addition von Zeilen) die j-te Spalte von $T(B)$ in den r-ten Einheitsvektor mit Kosten $t_{0j} = 0$. Das resultierende Tableau repräsentiert ein äquivalentes Gleichungssystem, da die Anwendung elementarer Zeilenoperationen nichts anderes als die Multiplikation mit regulären Matrizen ist. Da die anderen Basisspalten nicht verändert wurden, ergibt sich somit das Tableau $T(B')$.
Bevor wir das Vorgehen anhand des folgenden Lemmas 2.24 rechtfertigen, wollen wir es anhand eines Beispiels verdeutlichen.

Beispiel 2.7 *Wir setzen das vorhergehende Beispiel fort und starten mit dem Tableau zur Basis* $B = \{1, 2, 4\}$.

$T =$

1	*0*	*0*	*1*	*0*	*-3*	*-1*
0	*1*	*0*	*1*	*0*	*-1*	*1*
0	*0*	*1*	*0*	*0*	*1*	*1*
0	*0*	*0*	*-1*	*1*	**2**	*1*

Die Basislösung $x_1 = 1,\ x_2 = 1,\ x_3 = 0,\ x_4 = 1,\ x_5 = 0$ *mit Zielfunktionswert* $c^t x = 1$ *haben wir schon bestimmt. Da* $t_{05} < 0$, *also die reduzierten Kosten der Nichtbasisvariable* x_5 *negativ sind, wollen wir den Wert für* x_5 *erhöhen und diese Variable in die neue Basis bringen. Wir wählen also* $j = 5$. *Nach der Quotientenregel ergibt sich*

$$\delta = \min\{\frac{1}{1}, \frac{1}{2}\} = \frac{1}{2},$$

also $r = 3$ *und die Variable* $x_{B(3)} = x_4$ *soll die Basis verlassen. Das Pivotelement ist somit* $t_{rj} = t_{35} = 2$.

Um die fünfte Spalte zu dem dritten Einheitsvektor zu transformieren, ohne die Einheitsvektoren in den Spalten 1 und 2 zu verändern, geht man bei der **Pivotisierung** *folgendermaßen vor:*

1. *teile letzte Zeile durch 2*
2. *addiere die neu entstandene letzte Zeile*
   - *3 mal zu Zeile 0*
   - *1 mal zu Zeile 1*
   - *-1 mal zu Zeile 2*

Als neues Tableau ergibt sich

1	*0*	*0*	*-1/2*	*3/2*	*0*	*1/2*
0	*1*	*0*	*1/2*	*1/2*	*0*	*3/2*
0	*0*	*1*	**1/2**	*-1/2*	*0*	*1/2*
0	*0*	*0*	$-1/2$	*1/2*	*1*	*1/2*

Dieses Tableau ist das Tableau $T(B')$ zur Basis $B' = \{1, 2, 5\}$. Die Werte der Basisvariablen lassen sich wieder anhand der letzten Spalte ablesen, so dass sich mit $x_1 = 3/2,\ x_2 = 1/2,\ x_3 = 0,\ x_4 = 0,\ x_5 = 1/2$ die zugehörige Basislösung ergibt. Der Zielfunktionswert beträgt $c^t x = -1/2$. Da die reduzierten Kosten der Variablen x_3 mit $t_{03} = -\frac{1}{2} < 0$ negativ sind, wählen wir im nächsten Schritt $j = 3$. Nach der Quotientenregel erhält man $r = 2$, das Pivotelement ist also $t_{23} = 1/2$.

In diesem Fall führen wir bei der **Pivotisierung** *die folgende Zeilenoperationen aus:*

1. *addiere Zeile 2 zu Zeile 0*

2. *addiere Zeile 2 zu Zeile 3*

3. *ziehe Zeile 2 von Zeile 1 ab*

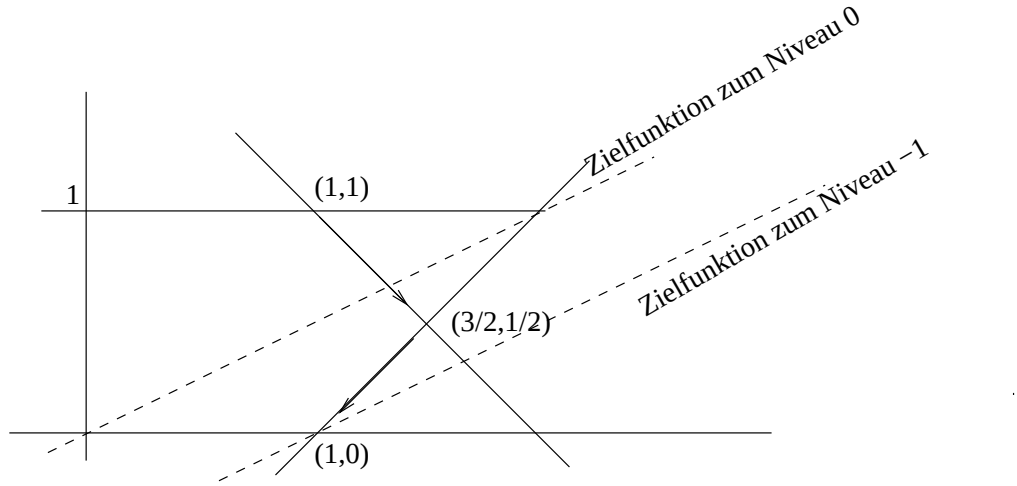
4. *multipliziere Zeile 2 mit 2*

Das Ergebnis ist im nächsten Tableau, das zur Basis $B'' = \{1, 3, 5\}$ gehört, dargestellt:

1	*0*	*1*	*0*	*1*	*0*	*1*
0	*1*	*-1*	*0*	*1*	*0*	*1*
0	*0*	*2*	*1*	*-1*	*0*	*1*
0	*0*	*1*	*0*	*0*	*1*	*1*

Die zugehörige Basislösung ist $x = (1, 0, 1, 0, 1)^t$ mit Zielfunktionswert -1. Da in diesem Tableau alle reduzierten Kosten t_{0j} größer oder gleich 0 sind, ist x nach Satz 2.16 optimal.

Geometrisch haben wir das Zulässigkeitspolyeder von Ecke zu Ecke abgesucht. Diese Wanderung am Rand entlang wird in der nächsten Abbildung verdeutlicht. Man beachte dabei, dass eine graphische Darstellung in zwei Variablen möglich ist, wenn wir den zulässigen Bereich anhand der beiden ursprünglichen Variablen x_1 und x_2 zeichnen und die Schlupfvariablen x_3, x_4 und x_5 vernachlässigen.



Wie bereits angekündigt, rechtfertigen wir abschließend noch die Korrektheit der Pivotisierung.

Lemma 2.24 *Sei B Basis und $B' = B\backslash\{B(r)\} \cup \{j\}$ eine benachbarte Basis. Dann kann man $T(B')$ durch Pivotisieren mit t_{rj} berechnen.*

Beweis:
Sei T das Ausgangstableau des Problems. Nach Definition gilt

$$\begin{aligned} T(B) &= T_B^{-1} \cdot T \\ T(B') &= T_{B'}^{-1} \cdot T, \end{aligned}$$

das heißt

$$T(B') = T_{B'}^{-1} \cdot T_B \cdot T(B) = H_1 \cdot T(B),$$

mit einer invertierbaren Matrix $H_1 := T_{B'}^{-1} \cdot T_B$. Sei T^p das durch Pivotisierung entstandene Tableau. Da das Pivotisieren die Anwendung elementarer Zeilenoperationen ist, lässt sich der Übergang von $T(B)$ zu T^p ebenfalls als Anwendung einer invertierbaren Matrix H_2 schreiben, die die Verkettung der einzelnen bei der Pivotisierung angewandten Zeilenoperationen repräsentiert:

$$T^p = H_2 \cdot T(B).$$

Wir wollen zeigen, dass $T^p = T(B')$.

Seien dazu $T_0, \ldots, T_{m+1}$ die Spalten von $T(B)$. $T^p = T(B')$, falls für alle Spalten T_j von $T(B)$ gilt, dass $H_2(T_j) = H_1(T_j)$. Erfreulicherweise müssen wir diese Aussage aber nicht für alle Spalten von $T(B)$ nachweisen, da aus der linearen Algebra bekannt ist, dass zwei lineare Abbildungen von $\mathbb{R}^{m+1} \longrightarrow \mathbb{R}^{m+1}$ bereits gleich sind, wenn sie auf einer Basis übereinstimmen. Auf welchen Spalten stimmen die beiden Abbildungen H_1 und H_2 nun überein?

- Die 0.Spalte von $T(B)$ bleibt sowohl bei Anwendung von H_1 als auch beim Pivotisieren unverändert, $H_1(A_0) = H_2(A_0) = \begin{pmatrix} 1 \\ 0 \\ \dots \\ 0 \end{pmatrix}$.

- Für die Basisspalten der *neuen* Basis B' gilt:

  - Die Basisspalten von B' haben im Tableau $T(B')$, also nach Anwendung von H_1 das auf Seite 37 beobachtete Aussehen.

  - Die durch H_2 repräsentierten Pivotoperationen sind so angelegt, dass die neue Basisspalte T_j zum Einheitsvektor mit Kostenkoeffizient 0 wird, $T_j = \begin{pmatrix} 0 \\ e_r \end{pmatrix}$. Weiterhin lässt sich leicht beobachten, dass sich die anderen "alten" Basisspalten T_i mit $i \in B \cap B' = B \setminus \{B(r)\}$ wegen $t_{ri} = 0$ bei den Pivotoperationen nicht verändern.

Damit stimmen H_1 und H_2 auf allen Spalten von $T(B)$ überein, sind somit gleich und es folgt, dass $T^p = T(B')$. QED

2.4.2 Beschreibung Algorithmus

Das Simplex-Verfahren beruht auf der im letzten Abschnitt entwickelten Idee des Basiswechsels, der im Simplextableau ausgeführt wird: Immer, wenn das Optimalitätskriterium nicht erfüllt ist, wird ein Basiswechsel durchgeführt und mittels einer Pivotoperation das zur nächsten Basis passende Simplextableau bestimmt. Das Verfahren lässt sich folgendermaßen formulieren.

Algorithmus 1: Simplex - Verfahren (Basisform)

Input: Basislösung (x_B, x_N) zu einer Basis B

Schritt 1: Berechne das Simplex-Tableau $T(B)$.

Schritt 2: Falls $t_{0j} \geq 0 \quad \forall j = 1, \ldots, n$

STOPP: Die zum Tableau gehörende Basislösung (x_B, x_N) mit $x_{B(i)} = t_{i(n+1)} \quad \forall i = 1, \ldots, m$ und $x_{N(j)} = 0 \quad \forall j = 1, \ldots, n-m$ mit Zielfunktionswert $-t_{0(n+1)}$ ist optimal.

Schritt 3: Wähle j mit $t_{0j} < 0$.

Schritt 4: Wenn $t_{ij} \leq 0 \quad \forall i = 1, \ldots, m$

STOPP: LP unbeschränkt.

Schritt 5: Bestimme $r \in \{1, \ldots, m\}$ mit $\frac{t_{r(n+1)}}{t_{rj}} = \min_{i=1,\ldots,m}\{\frac{t_{i(n+1)}}{t_{ij}} : t_{ij} > 0\}$.

Schritt 6: Pivotisiere mit t_{rj}, d.h. multipliziere Zeile r mit $\frac{1}{t_{rj}}$ und addiere für alle Zeilen $i = 1, \ldots, m$, $i \neq r$ das $-\frac{t_{ij}}{t_{rj}}$ fache von Zeile r zu Zeile i.

Schritt 7: Gehe zu Schritt 2.

Schritt 6 ergibt nach Lemma 2.24 einen korrekten Basiswechsel. Wenn das Verfahren also abbricht, ist die so erhaltene Lösung nach dem Optimalitätskriterium (Satz 2.16) optimal. Allerdings müssen wir noch diskutieren,

- ob das Verfahren wirklich abbricht, also endlich ist. Damit werden wir uns im folgenden Abschnitt 2.4.3 beschäftigen.
- Außerdem ist zu klären, wie man die im Input geforderte zulässige Startlösung erzeugen kann. Ein Verfahren, mit dem das möglich ist, wird in Abschnitt 2.4.4 beschrieben.

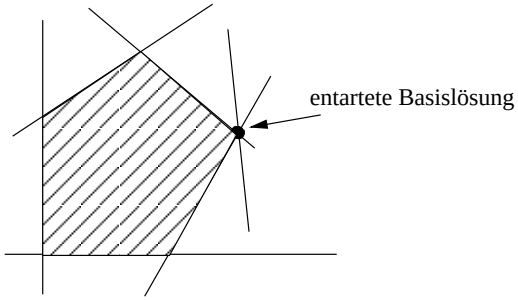
2.4.3 Degeneriertheit und Endlichkeit des Simplex-Verfahrens

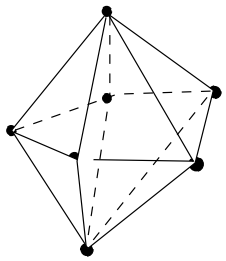
In diesem Abschnitt untersuchen wir, ob das Simplex-Verfahren wie in Algorithmus 1 definiert, endlich ist; also ob es irgendwann mit einer optimalen Basislösung oder dem Hinweis, dass das Problem unbeschränkt ist, abbricht, oder ob es ins Kreisen geraten kann.

Wir wissen, dass es maximal $\binom{n}{m}$ Basislösungen geben kann. Wird jede von ihnen im Simplex-Verfahren höchstens einmal besucht, so ist das Verfahren endlich. Das wäre gewährleistet, wenn wir zeigen können, dass sich der Zielfunktionswert in jedem Pivotschritt verbessert. Aber leider stimmt diese Aussage nicht, so dass

wir in diesem Abschnitt sogar das Simplex-Verfahren etwas modifizieren müssen, um seine Endlichkeit zu garantieren! Wir benötigen zunächst die folgende Definition.

Definition 2.25 *Eine zulässige Basislösung heißt* **degeneriert** *(oder* **entartet***), falls mindestens eine ihrer Basisvariablen den Wert* 0 *hat.*





alle Basislösungen
sind entartet

Wie die linke Abbildung zeigt, kann man im $\mathbb{R}^2$ entartete Basislösung durch das Entfernen von redundanten Nebenbedingungen verhindern. Die rechte Abbildung demonstriert, dass das im $\mathbb{R}^m$ mit $m \geq 3$ nicht gilt: Hier kann es durchaus sein, dass sich mehr als m Nebenbedingungen in einer Ecke des Polyeders treffen, man aber keine von ihnen weglassen kann, ohne den zulässigen Bereich zu verändern.

Im Fall, dass es keine degenerierten Basislösungen gibt, ist das Problem der Endlichkeit des Simplex-Verfahrens nach unseren Vorüberlegungen schnell geklärt.

Satz 2.26 *Sei (LP) ein lineares Programm, das nur nicht degenerierte Basislösungen besitzt. Dann endet das Simplex-Verfahren nach spätestens* $\binom{n}{m}$ *Pivotoperationen.*

Beweis: In jeder Pivotoperation wird der Zielfunktionswert um $x_{N(s)} \cdot \bar{c}_{N(s)}$ verbessert. Wir wollen zeigen, dass dieser Wert $x_{N(s)} \cdot \bar{c}_{N(s)}$ echt kleiner als Null ist. Nach der Quotientenregel gilt:

$$x_{N(s)} = \min\{\frac{\tilde{b}_i}{\tilde{a}_{iN(s)}} : \tilde{a}_{iN(s)} > 0\}$$

Entweder ist also das lineare Programm unbeschränkt (Satz 2.18 auf Seite 28), oder das Minimum existiert und

$$x_{N(s)} = \frac{\tilde{b}_i}{\tilde{a}_{iN(s)}} > 0,$$

weil wir wissen, dass $\tilde{b}_i$ als i-te Komponente von $A^{-1}b$ dem Wert der neuen Basisvariablen x_i entspricht und somit nach der Voraussetzung, dass B nicht

entartet ist, echt größer als Null ist. Weil sich der Zielfunktionswert in jedem Schritt echt verbessert, kann sich also keine Basislösung wiederholen, und das Verfahren ist endlich. QED

Bei entarteten linearen Programmen kann das Simplex-Verfahren dagegen kreisen! Das erste Beispiel dafür wurde von A. Hoffman im Jahr 1953 ([Hof53]) präsentiert. In den ÜBUNGEN wird ein (kleineres) Beispiel von Beale aus dem Jahr 1955 ([Bea55]) diskutiert.

Um das Kreisen des Simplex-Verfahrens zu verhindern, müssen wir die angewendete Pivotregel modifizieren. Bisher haben wir als Pivotspalte eine beliebige Spalte mit negativen reduzierten Kosten gewählt, und auch bei der Auswahl der Pivotzeile nicht darauf geachtet, welches r wir wählen, falls das Minimum in der Quotientenregel nicht eindeutig war. In der Praxis scheint bei der Auswahl der Pivotspalte eine Spalte mit betragsmäßig möglichst großen negativen Kosten zu den besten Ergebnissen zu führen. Um das Kreisen zu verhindern, sollte man dagegen die folgende Pivotregel von Bland ([Bla77]) anwenden.

Bland's Pivotregel:

- Wähle Spalte j mit negativen Kosten durch:
$$j = \min\{j' : t_{0j'} < 0\}.$$
- Wähle in der Quotientenregel r, so dass
$$B(r) = \min\{B(i) : t_{ij} > 0 \quad und \quad \frac{t_{i(n+1)}}{t_{ij}} \leq \frac{t_{k(n+1)}}{t_{kj}} \ \ \forall k \ \ mit \ \ t_{kj} > 0\}$$
d.h. die erste Basisvariable, an der das Minimum angenommen wird, muss die Basis verlassen.

Der Beweis dafür, dass das Simplex-Verfahren mit dieser Pivotregel endlich ist, kann z.B. in [HK04] nachgelesen werden. Der Nachweis der Endlichkeit lässt nun auch die Schlussfolgerung zu, dass jedes (endlich lösbare) lineare Programm mindestens eine Basislösung besitzt, die das Optimalitätskriterium aus Satz 2.16 erfüllt.

Satz 2.27 *Sei die Menge der Optimallösungen $\mathcal{E}$ von $\min\{c^t x : Ax = b, x \in \mathbb{R}^n, x \geq 0\}$ nicht leer. Dann kreist das Simplexverfahren bei Anwendung von Bland's Pivotregel nicht. Insbesonders gibt es also eine optimale Basislösung (x_B, x_N) bezüglich einer Basis B von A, die*

$$\bar{c}_{N(j)} = c_{N(j)} - c_B^t A_B^{-1} A_{N(j)} \geq 0 \ \ \forall j = 1, \cdots, n-m$$

erfüllt.

Es soll am Ende dieses Abschnittes noch darauf hingewiesen werden, dass in der Praxis sehr häufig — eigentlich in allen größeren linearen Optimierungsproblemen — entartete Basen auftreten. Dennoch wird ein Kreisen des Simplex-Verfahrens in praktischen Beispielen kaum beobachtet.

2.4.4 Finden einer zulässigen Startlösung

Im Simplex-Verfahren wie auf Seite 44 beschrieben, wird als Input eine zulässige Basislösung gefordert. In diesem Abschnitt wollen wir uns mit der Berechnung einer solchen zulässigen Basislösung beschäftigen. Im einfachsten Fall liegt unser lineares Programm in $\leq$-Form mit nicht-negativer rechter Seite b vor, also etwa

$$\begin{aligned} &\min c^t x \\ &\text{s.d. } Ax \;\leq\; b, \end{aligned}$$

mit einer $m \times n$-Matrix A, $c \in \mathbb{R}^n$ und $b \in \mathbb{R}^m$, wobei wir voraussetzen, dass $b_i \geq 0$ für alle $i = 1, \ldots, m$. In diesem Fall bringen wir das lineare Programm durch die Einführung von m Schlupfvariablen in Standardform und erhalten

$$\begin{aligned} \min c^t x & \\ \text{s.d. } Ax + Iy &= b \\ y &\geq 0. \end{aligned}$$

Eine Basis der Koeffizientenmatrix $\tilde{A} = (A|I)$ ist also durch die zu den Schlupfvariablen gehörenden Spalten $\tilde{A}_B = I$ gegeben. Die entsprechende Basislösung ergibt sich als

$$x_N = 0, \quad x_B = \tilde{A}_B^{-1} b = Ib = b \geq 0.$$

Sie entspricht genau dem Ursprung im Originalproblem in $\leq$-Form und weist den Zielfunktionswert Null auf.

Obwohl wir nicht immer von der $\leq$-Form mit positiver rechter Seite ausgehen können, gibt uns dieser einfache Fall die grundsätzliche Idee zum Erzeugen einer ersten zulässigen Basislösung. Die Idee soll zunächst an einem Beispiel geschildert werden.

Beispiel 2.8 *Betrachten wir das folgende System*

$$(A|b) = \left(\begin{array}{ccccc|c} 0 & 1 & 1 & 0 & 0 & 3 \\ -2 & 2 & 0 & 2 & 0 & 2 \\ -1 & -1 & 0 & 0 & 1 & -1 \end{array}\right).$$

Um eine nicht-negative rechte Seite zu erhalten, multiplizieren wir zunächst alle Zeilen i mit $b_i < 0$ mit (-1). Wir erhalten:

$$\longrightarrow \quad \left(\begin{array}{ccccc|c} 0 & 1 & \mathbf{1} & 0 & 0 & 3 \\ -2 & 2 & 0 & \mathbf{2} & 0 & 2 \\ 1 & 1 & 0 & 0 & -1 & 1 \end{array}\right).$$

Die beiden dick gedruckten Einträge können bereits als Schlupfvariablen interpretiert werden, da sie positiv sind und ihre Spalten sonst nur Nullen enthalten.

Das geht bei Zeile 3 nicht, da die Basis $B = \{3, 4, 5\}$ zu einer unzulässigen Basislösung $x_1 = x_2 = 0\ x_3 = 3\ x_4 = 1\ x_5 = -1$ führt. Daher führt man für die 3. Zeile eine neue Variable $\hat{x}_3$ ein, mit:

$$x_1 + x_2 + 0 \cdot x_3 + 0 \cdot x_4 - x_5 + \hat{x}_3 = 1,$$

und wir erhalten das neue System

$$(\tilde{A}|b) = \left(\begin{array}{cccccc|c} 0 & 1 & 1 & 0 & 0 & 0 & 3 \\ -2 & 2 & 0 & 2 & 0 & 0 & 2 \\ 1 & 1 & 0 & 0 & -1 & 1 & 1 \end{array}\right).$$

In diesem neuen System ist eine Basis bekannt, nämlich $B = \{3, 4, 6\}$. Allerdings haben wir durch die Einführung der neuen Variable $\hat{x}_3$ das Originalproblem verändert. Daher lösen wir im nächsten Schritt als Hilfsproblem das lineare Programm

$$\begin{aligned} \min \quad & \sum_{i=1}^{m} \hat{x}_i \\ \text{s.d.} : \tilde{A} & \begin{pmatrix} x \\ \hat{x} \end{pmatrix} = b \\ & \begin{pmatrix} x \\ \hat{x} \end{pmatrix} \geq 0 \quad , \end{aligned}$$

um die neuen Variablen wieder loszuwerden. Dieses Hilfsproblem können wir — im Gegensatz zum Originalproblem — mit dem Simplex-Verfahren lösen, da wir eine zulässige Basislösung aufgrund unserer Konstruktion bereits kennen.

Formal gehen wir zur Erzeugung einer zulässigen Startlösung also wie folgt vor:

1. Multipliziere alle Zeilen i für die $b_i < 0$ mit (-1). Sei $\min\{c^t x : Ax = b, x \geq 0\}$ das resultierende lineare Programm mit $b \geq 0$.

2. Löse das Hilfsproblem $\min\{\sum_{i=1}^{m} \hat{x}_i : Ax + I\hat{x} = b, x, \hat{x} \geq 0\}$.

Warum uns die Lösung des Hilfsproblemes weiter hilft, zeigt der folgende Satz. Wie schon in Schritt 2 nehmen wir auch hier der Einfachheit halber an, dass wir in jeder Zeile eine Hilfsvariable eingefügt haben. Wie das eben betrachtete Beispiel zeigt, kann man je nach Gegebenheit des Originalproblems oft auch mit weniger Hilfsvariablen auskommen.

Satz 2.28 *Sei das (LP) $\min\{c^t x : Ax = b, x \geq 0\}$ gegeben und sei $\min\{\sum_{i=1}^{m} \hat{x}_i : Ax + I\hat{x} = b,\ x, \hat{x} \geq 0\}$ das zugehörige Hilfsproblem. Dann gilt:*
(LP) ist zulässig $\iff$ $\min\{\sum_{i=1}^{m} \hat{x}_i : Ax + I\hat{x} = b,\ x, \hat{x} \geq 0\} = 0$.

Beweis:

"$\Longrightarrow$" Sei (LP) zulässig. Dann existiert $\bar{x}$ mit $A\bar{x} = b$, $\bar{x} \geq 0$. Eine zulässige Lösung des Hilfsproblems erhält man in diesem Fall durch $x := \bar{x}$, $\hat{x} := 0$.

"$\Longleftarrow$" Sei nun

$$\min\{\sum_{i=1}^{m} \hat{x}_i : Ax + I\hat{x} = b,\ x, \hat{x} \geq 0\} = 0.$$

Ist $\begin{pmatrix} x^* \\ \hat{x}^* \end{pmatrix}$ optimal so gilt $\hat{x}^* = 0$.

$$\Longrightarrow \exists x^* \geq 0 \quad mit \quad Ax^* = b.$$

Also ist (LP) zulässig.

QED

Eine zulässige Lösung des linearen Programms (LP) lässt sich durch Lösen des Hilfsproblems also leicht finden. Wie findet man aber eine Basislösung? Wir unterscheiden zwei Fälle:

- Fall 1: Alle $\hat{x}^*$-Variablen sind Nichtbasisvariablen. Dann sind die x^*-Variablen Basisvariablen und wir können sie als zulässige Basislösung des Originalproblems verwenden.
- Fall 2: Ist $\hat{x}_i^* = 0$ eine entartete Basisvariable (das heißt eine Basisvariable, die den Wert Null annimmt), so pivotisiert man sie aus der Basis heraus, bis man eine optimale Basislösung erhält, die keine künstlichen Variablen als Basisvariablen hat.

Dass man die künstlichen Variablen aus der Basis heraus pivotisieren kann (d.h. dass die entsprechenden Pivotelemente ungleich Null sind), ist in der ÜBUNG zu zeigen.

Die Minimierung des Hilfsproblemes nennt man auch Phase 1 des Simplex-Verfahrens. Der in Algorithmus 1 beschriebene eigentliche Simplex-Algorithmus heißt Phase 2 des Simplex-Verfahrens.

Bei der Umsetzung der Phase 1 sind die folgenden Hinweise unbedingt zu beachten:

- Im Ausgangstableau des Hilfsproblemes ist eine Basis B leicht ablesbar. Man darf aber nicht vergessen, die Kosten der Basisspalten in der obersten Zeile des Tableaus (durch elementare Zeilenoperationen) auf Null zu setzen, so dass das Tableau $T(B)$ der entsprechenden Basis B vorliegt.

- Ebenso müssen vor Beginn der Phase 2 die Kostenkoeffizienten der durch die Phase 1 bestimmten neuen Basisvariablen durch elementare Zeilenoperationen auf Null gesetzt werden.

Wir werden den Algorithmus erst angeben und dann an dem eben schon verwendeten Beispiel demonstrieren.

Algorithmus 2: 2-Phasen Simplex Verfahren

Input: Lineares Programm der Form $\min\{c^t x : Ax = b, x \geq 0\}$, wobei A eine $m \times n$-Matrix, $b \in \mathbb{R}^m$, $c \in \mathbb{R}^n$.

1. Erzeuge das Gleichungssystem $Ax + I\hat{x} = b$, mit $m \times m$ Einheitsmatrix I und $b \geq 0$.

1. Multipliziere Zeilen mit negativem b_i mit (-1)
2. Addiere die Variable $\hat{x}_i$ zu Zeile $i, i = 1, \ldots, m$

2. Erzeuge das entsprechende Tableau $T(B)$ zu Basis B, wobei B die Spalten von I enthält (Basislösung $\hat{x} = b, x = 0$).

Passe die Kostenzeile durch Anwendung elementarer Zeilenoperationen an die Basis an.

3. Löse $z = \min\{\sum_{i=1}^{m} \hat{x}_i : Ax + I\hat{x} = b, x, \hat{x} \geq 0\}$

4. Wenn $z > 0$, **STOPP:** Das LP ist unzulässig.

5. Wenn $z = 0$:

1. Pivotisiere alle künstlichen Variablen $\hat{x}_i$ aus der Basis.
2. Streiche die zu $\hat{x}$ gehörenden Spalten aus dem Tableau.
3. Ersetze die Zielfunktion durch $c^t x$.
4. Wende elementare Zeilenoperationen an, um die Koeffizienten der Basisvariablen in der Zielfunktion zu Null zu machen.

6. Wende Phase 2 des Simplexverfahrens (d.h. Algorithmus 1 auf Seite 44) an, um das so entstandene lineare Programm zu lösen.

Beispiel 2.9 *Wir setzten Beispiel 2.8 aus diesem Abschnitt fort. Das Originalproblem ist durch das folgende Tableau gegeben:*

-2	*-1*	*0*	*0*	*0*	*0*
0	*1*	*1*	*0*	*0*	*3*
-2	*2*	*0*	*2*	*0*	*2*
-1	*-1*	*0*	*0*	*1*	*-1*

.

Nach der Einführung einer künstlichen Variablen in Zeile 3 erhalten wir das folgende Hilfsproblem

0	*0*	*0*	*0*	*0*	*1*	*0*
0	*1*	*1*	*0*	*0*	*0*	*3*
-1	*1*	*0*	*1*	*0*	*0*	*1*
1	*1*	*0*	*0*	*-1*	*1*	*1*

,

in dem wir die zulässige Basis $B = \{3, 4, 6\}$ *kennen. Zunächst muss die Kostenzeile an diese Basis angepasst werden. Um in der 6. Spalte eine Null als Koeffizient zu erreichen, subtrahieren wir die 3. Zeile von der Kostenzeile. Wir erhalten das folgende Tableau*

-1	*-1*	*0*	*0*	*1*	*0*	*-1*
0	*1*	*1*	*0*	*0*	*0*	*3*
-1	*1*	*0*	*1*	*0*	*0*	*1*
1	*1*	*0*	*0*	*-1*	*1*	*1*

,

an dem wir die Basislösung $x_1 = x_2 = x_5 = 0$, $x_3 = 3$, $x_4 = 1$, $\hat{x}_3 = 1$ *mit Zielfunktionswert* $\hat{x}_3 = 1$ *ablesen können. Wir suchen nun mit dem Simplex-Verfahren die Lösung des Hilfsproblems. Als Pivotelement wählen wir* $t_{31} = 1$ *und erhalten nach der entsprechenden Pivotoperation das folgende Tableau*

0	*0*	*0*	*0*	*0*	*1*	*0*
0	*1*	*1*	*0*	*0*	*0*	*3*
0	*2*	*0*	*1*	*-1*	*1*	*2*
1	*1*	*0*	*0*	*-1*	*1*	*1*

mit der Basis $B(1) = 3, B(2) = 4, B(3) = 1$. *Der Zielfunktionswert lässt sich oben rechts als Null ablesen und ist damit optimal. Weiterhin sind alle künstlichen Variablen (in unserem Fall also* $\hat{x}_3$*) Nichtbasisvariable, so dass wir ohne weitere Pivotoperationen zum Originalproblem zurückkehren können, indem wir die zu den künstlichen Variablen gehörenden Spalten weglassen und die Kostenzeile des Hilfsproblemes durch die Kostenzeile des Originalproblems ersetzen,*

-2	*-1*	*0*	*0*	*0*	*0*
0	*1*	*1*	*0*	*0*	*3*
0	*2*	*0*	*1*	*-1*	*2*
1	*1*	*0*	*0*	*-1*	*1*

.

Im Originalproblem können wir nun als Startbasis die optimale Basis $B' = \{3, 4, 1\}$ *des Hilfsproblems verwenden. Zunächst müssen wir aber wieder die Kostenzeile an die neue Basis* B' *anpassen. Wir transformieren die* -2 *oben links zu* 0*, indem wir das Doppelte der 3. Zeile zu der Kostenzeile addieren und erhalten das*

Tableau

0	*1*	*0*	*0*	*-2*	*2*
0	*1*	*1*	*0*	*0*	*3*
0	*2*	*0*	*1*	*-1*	*2*
1	*1*	*0*	*0*	*-1*	*1*

mit Basislösung $x_1 = 1$, $x_2 = 0$, $x_3 = 3$, $x_4 = 2$, $x_5 = 0$ *und Zielfunktionswert* $-2x_1 - x_2 = -2$. *Weitere Pivotoperationen können wir uns sparen, da Spalte 5 des Tableaus zeigt, dass unser Problem unbeschränkt ist.*

Das Ergebnis lautet also: Das Originalproblem ist unbeschränkt.

2.4.5 Das revidierte Simplex-Verfahren

In diesem Abschnitt soll das so genannte *revidierte* Simplex-Verfahren skizziert werden. Es ist von großer praktischer Bedeutung bei linearen Problemen mit sehr vielen Variablen. Weiterhin ist es Grundlage für viele Verfahren der ganzzahligen Optimierung, z.B. für Spaltengenerierungsverfahren (*Column Generation*) und für Dekompositionsansätze (z.B. *Dantzig-Wolfe Dekomposition*).

Die Ausgangsüberlegung des revidierten Simplex-Verfahrens ist die folgende: Bei großen Tableaus wird das Pivotisieren des ganzen Tableaus aufwändig und das Simplex-Verfahren damit ineffizient. Im Fall, dass m sehr viel kleiner ist als n (man schreibt dann $m << n$), kann man das Simplex-Verfahren entscheidend verbessern. Erinnern wir uns an das Vorgehen zur Bestimmung des Pivotelements:

1. Entscheide, ob es eine Nichtbasisvariable mit negativen reduzierten Kosten gibt, also eine Spalte j mit $\bar{c}_j = c_j - c_B A_B^{-1} A_j < 0$.

2. Wenn es so eine Spalte j gibt, wähle sie als Pivotspalte und bestimme die Pivotzeile durch
$$\min\{\frac{\tilde{b}_i}{\tilde{a}_{ij}} : \tilde{a}_{ij} > 0\},$$
wobei $\tilde{b} = A_B^{-1} b$, $\tilde{A}_j = A_B^{-1} A_j$ ist.

Um letztere der beiden Informationen zu erhalten, ist also nicht das komplette Tableau nötig, sondern die Kenntnis von A_B^{-1} reicht aus. Das führt zu der Idee, die Pivotisierung immer nur in einem kleineren Tableau mit nur $m + 2$ Spalten durchzuführen, das aus den m Basisspalten besteht zuzüglich der neuen Spalte j. Betrachten wir dazu folgendes Ausgangstableau, in dem eine zulässige Basis bereits bekannt ist,

1	c^t	0	0
0	A	I	b

und erinnern wir uns an die Definition des Tableaus $T(B)$ zu einer beliebigen anderen Basis B (siehe Definition 2.23 auf Seite 36). Man erhält:

$$T(B) = \begin{array}{|c|c|c|c|} \hline 1 & \bar{c}^t = c^t - c_B^t A_B^{-1} A & -c_B^t A_B^{-1} & -c_B^t A_B^{-1} b \\ \hline 0 & \tilde{A} = A_B^{-1} A & A_B^{-1} I & \tilde{b} = A_B^{-1} b \\ \hline \end{array}.$$

Im folgenden Verfahren werden die Teile $\tilde{A}$ und $\bar{c}$ in den Pivotoperationen nicht vollständig berechnet.

Algorithmus 3: revidiertes Simplex - Verfahren (Basisform)

Input: Lineares Programm mit Ausgangstableau $\begin{array}{|c|c|c|c|} \hline 1 & c^t & 0 & 0 \\ \hline 0 & A & I & b \\ \hline \end{array}$, $T_{red}(B) = \begin{array}{|c|c|} \hline 0 & 0 \\ \hline I & b \\ \hline \end{array}$,

$B := \{n-m+1, \ldots, n\}$, $\tilde{b} = b$, $\tilde{A} = A$

Schritt 1: Falls $\bar{c}_j = c_j - c_B^t A_B^{-1} A_j \geq 0$ für alle $j \in N$ **STOPP**:

(x_B, x_N) mit $x_{B(i)} = \tilde{b}_i \;\; x_N = 0$ ist optimal.

Schritt 2: Sonst wähle $j \in N$ mit $\bar{c}_j < 0$. Erzeuge $\tilde{A}_j = A_B^{-1} A_j$ (A_j bezeichnet dabei die j-te Originalspalte) und füge die Spalte $\tilde{A}_j$ ganz rechts an das Tableau T_{red} an.

Schritt 3: Bestimme

$$\frac{\tilde{b}_r}{\tilde{a}_{rj}} = \min_{i=1\ldots m} \{\frac{\tilde{b}_i}{\tilde{a}_{ij}} : \tilde{a}_{ij} > 0\}.$$

Existiert der Ausdruck nicht, **STOPP**: Das lineare Programm ist unbeschränkt.

Sonst: Pivotisiere $(T_{red}|\tilde{A}_j)$ mit Pivotelement $\tilde{a}_{rj}$. Setze $B(r) = j$. Erhalte neues A_B^{-1} aus Spalten $1, \ldots, m$ von T_{red} und gehe zu Schritt 1.

Wie oben schon angesprochen, wird das revidierte Simplex-Verfahren bei der Bearbeitung großer linearer Programme eingesetzt. Es eignet sich besonders gut, wenn z.B. jede Spalte eine mögliche Lösung eines Problems repräsentiert, und man diese Lösungen nicht mitführen muss, sondern erzeugen kann. Das Teilproblem, eine Spalte mit möglichst kleinen reduzierten Kosten zu finden, oder zu bestätigen, dass es keine negativen reduzierten Kosten gibt, wird dabei als *Pricing Problem* bezeichnet. Die Effizienz des revidierte Simplex-Verfahrens wird also maßgeblich davon beeinflusst, wie effizient man das Pricing Problem lösen kann. In der Praxis wird das revidierte Simplex-Verfahren oft bei der Lösung großer ganzzahliger Programme eingesetzt, z.B. beim Bearbeiten von Personaleinsatzproblemen (*Crew Scheduling*).

2.4.6 Weitere Simplex-Varianten

Es gibt noch viele weitere Simplex-Varianten, von denen einige abschließend erwähnt werden sollen.

- Das *Simplex-Verfahren mit beschränkten Variablen* behandelt lineare Programme, in denen es untere und obere Schranken für die Variablen gibt, also lineare Programme mit Nebenbedingungen der Form

$$l_j \leq x_j \leq v_j \quad \forall j = 1, \ldots, n.$$

  Natürlich kann man diese Beschränkungen als Nebenbedingungen schreiben und mit dem normalen Simplex-Verfahren behandeln, das Simplex-Verfahren mit beschränkten Variablen bietet aber einen effizienteren Zugang. Das Verfahren kann z.B. in [HK04] nachgelesen werden.

- Ein Spezialfall des Simplex-Verfahrens mit beschränkten Variablen ist das *Netzwerk-Simplex-Verfahren* für Netzwerkflussprobleme. Netzwerkflussprobleme werden in Abschnitt 3.2 eingeführt, wobei wir aber nicht näher auf das spezielle Simplex-Verfahren zu ihrer Lösung eingehen. Hier verweisen wir wiederum z.B. auf [HK04].

- Das *duale Simplex-Verfahren* werden wir in Abschnitt 2.5.5 beschreiben.

- Auf *primal-duale Simplex-Verfahren* soll ebenfalls in Abschnitt 2.5.5 eingegangen werden.

2.5 Dualität bei linearen Programmen

2.5.1 Definition des dualen Programms

Zueinander duale Probleme spielen in der Optimierung eine große Rolle. Neben verschiedenen geometrischen Dualitäten ist vor allem die algebraische Dualität besonders wichtig. Wir führen sie bereits hier im Teil der linearen Optimierung ein.
Als Motivation kann man sich die Suche nach Schranken für Optimierungsprobleme vorstellen, d.h. wir wollen feststellen, wie gut der Zielfunktionswert eines linearen Programms im besten Fall werden kann. Sei dazu folgendes lineare Programm gegeben,

$$\begin{aligned} \min \quad & c^t x \\ \text{s.d. } & Ax = b \\ & x \geq 0 \end{aligned} \quad .$$

Wir suchen eine *untere Schranke* für dieses Problem.

Definition 2.29 *Eine* **untere Schranke** *eines linearen Programms* $\min\{c^t x : Ax = b, x \geq 0\}$ *ist ein Wert* $\underline{z}$*, so dass*

$$\underline{z} \leq c^t x \ \ \forall x \in P = \{x \in \mathbb{R}^n : Ax = b, x \geq 0\}.$$

Dazu betrachten wir zunächst ein Beispiel.

Beispiel 2.10 *Sei das folgende lineare Programm gegeben.*

$$\begin{array}{rl} \min & 5x_1 - x_2 \qquad + 6x_4 \\ \text{s.d.} & 2x_1 + x_2 + x_3 - x_4 = 4 \quad (1) \\ & x_1 - x_2 - x_3 + 2x_4 = 3 \quad (2) \\ & x_i \geq 0 \quad i = 1, 2, 3, 4 \end{array}$$

Wir wollen zeigen, dass der optimale Wert des linearen Programms z^*

$$z^* \geq 10$$

erfüllt, d.h. dass 10 eine untere Schranke des gegebenen linearen Programms ist. Anwender wissen somit von vornherein, dass ihre Zielfunktion (z.B. die erwarteten Kosten) $c^t x$ *nie besser als 10 werden kann.*

Um zu beweisen, dass 10 eine untere Schranke ist, gehen wir wie folgt vor: Wir addieren die erste Restriktion (1) *zu dem zweifachen der zweiten Restriktion* (2) *und erhalten*

$$(1) \cdot 1 + (2) \cdot 2 = 4x_1 - x_2 - x_3 + 3x_4 = 10 \text{ für alle } x = (x_1, x_2, x_3, x_4)^t \in P.$$

Nun können wir abschätzen, dass

$$5x_1 - x_2 + 0x_3 + 6x_4 \geq 4x_1 - x_2 - x_3 + 3x_4 = 10$$

für alle zulässigen x*, also ist* $c^t x \geq 10$ *für alle* $x \in P$ *und* 10 *ist eine untere Schranke des linearen Programms.*

Insbesondere gilt, dass der optimale Zielfunktionswert $z^* = c^t x^* \geq 10$ *ist.*

Etwas allgemeiner können wir für ein lineares Programm der Form

$$\begin{array}{rll} \min & c^t x & \\ \text{s.d.} & a_{i1}x_1 + \ldots + a_{in}x_n = b_i, & i = 1, \ldots, m \\ & x_j \geq 0 & j = 1, \ldots, n \end{array} \tag{2.8}$$

folgendes Vorgehen zur Erzeugung einer unteren Schranke anwenden:

1. Wähle Zahlen π_i für jede Nebenbedingung $i = 1, \ldots, m$.
2. Multipliziere Gleichung i mit π_i.
3. Addiere alle Gleichungen.

Man erhält, dass die folgende Gleichung

$$\sum_{i=1}^{m} a_{i1}\pi_i x_1 + \ldots + \sum_{i=1}^{m} a_{in}\pi_i x_n = \sum_{i=1}^{m} b_i \pi_i \tag{2.9}$$

für alle $x \in P$ erfüllt sein muss.

Lemma 2.30 *Sei $\pi = (\pi_1, \ldots, \pi_m)^t$. Dann ist $\pi^t b \in \mathbb{R}$ eine untere Schranke für $\min\{c^t x \colon Ax = b,\ x \geq 0\}$, falls $A^t\pi \leq c$.*

Beweis: Sei $A^t\pi \leq c$, d.h.

$$\sum_{i=1}^{m} a_{ij}\pi_i \leq c_j \quad \text{für alle } j = 1, \ldots, n$$

Wir erhalten für alle zulässigen x

$$\begin{aligned}
\pi^t b &= \sum_{i=1}^{m} \pi_i b_i \\
&= \sum_{i=1}^{m} a_{i1}\pi_i x_1 + \cdots + \sum_{i=1}^{m} a_{in}\pi_i x_n \quad \text{nach (2.9)} \\
&= x_1 \sum_{i=1}^{m} a_{i1}\pi_i + \cdots + x_n \sum_{i=1}^{m} a_{in}\pi_i \\
&\leq \sum_{j=1}^{n} c_j x_j = c^t x,
\end{aligned}$$

also ist $\pi^t b$ eine untere Schranke. QED

Erstrebenswert ist natürlich eine möglichst gute untere Schranke, d.h. eine möglichst große untere Schranke. Unter allen π mit $A^t\pi \leq c$ wollen wir also den Vektor π wählen, der zu einem möglichst großen Wert $\pi^t b$ der unteren Schranke führt. Das führt uns zu dem folgenden linearen Programm:

$$\begin{aligned}
\max \quad & b^t\pi \\
\text{s.d. } & A^t\pi \leq c \\
& \pi \gtrless 0.
\end{aligned}$$

Dieses Programm heißt das Duale zu dem in (2.8) gegebenen linearen Programm. Da dieser Begriff sehr wichtig ist, halten wir ihn in einer Definition fest.

Definition 2.31 *Sei ein lineares Programm (P) gegeben als*

$$\begin{array}{lll} \min & c^t x & \\ \text{s.d.} & Ax = b & (P) \\ & x \geq 0. & \end{array}$$

Das lineare Programm

$$\begin{array}{lll} \max & b^t \pi & \\ \text{s.d.} & A^t \pi \leq c & (D) \\ & \pi \gtrless 0 & \end{array}$$

heißt das **Duale** *zu (P). Das Originalproblem (P) wird auch das* **Primale** *genannt.*

In den folgenden Abschnitten werden wir zeigen, was für Eigenschaften das duale Programm hat, und für was man es benutzen kann.

2.5.2 Starke und schwache Dualität

Wir zeigen zunächst, dass das Duale des Dualen wieder das primale Programm ist.

Lemma 2.32 *Das Duale von (D) ist (P).*

Beweis: Den Beweis führen wir in drei Schritten.

Schritt 1: Da wir bisher nur das Duale eines linearen Programms *in Standardform* definiert haben, formen wir das Duale (D) zunächst in Standardform um. Mit $\pi = \pi^+ - \pi^-$ erhalten wir

$$\begin{array}{lrcl} - \min & -b^t\pi^+ + b^t\pi^- & & \\ \text{s.d.} & A^t(\pi^+) - A^t(\pi^-) + \gamma & = & c \\ & \pi^+, \pi^-, \gamma & \geq & 0. \end{array}$$

Schritt 2: Nun bestimmen wir das Duale von (D).

$$\begin{array}{lll} - \max & c^t y & \\ \text{s.d.} & \tilde{A}^t y^t \leq \tilde{c} & \end{array}$$

mit:

$$\tilde{A} = (A^t| - A^t|I), \tilde{A}^t = \begin{pmatrix} A \\ -A \\ I \end{pmatrix}, \tilde{c} = \begin{pmatrix} -b \\ b \\ 0 \end{pmatrix}.$$

Wir erhalten also

$$\begin{array}{rrcl} - & \max c^t y & & \\ & \text{s.d. } Ay & \leq & -b \\ & -Ay & \leq & b \\ & Iy & \leq & 0, \end{array}$$

und vereinfachen dieses lineare Programm zu

$$\begin{array}{rrcl} - & \max c^t y & & \\ & \text{s.d. } Ay & = & -b \\ & y & \leq & 0. \end{array}$$

Schritt 3: Im letzten Schritt machen wir uns klar, dass das in Schritt 2 bestimmte Duale von (D) wieder das Primale ist. Das Ersetzen von $y = -x$ führt zu

$$\begin{array}{rrcl} & \min c^t x & & \\ & \text{s.d. } Ax & = & b \\ & x & \geq & 0, \end{array}$$

und das ist (P). QED

Der nächste Satz zeigt eine erste einfache Aussage über die Beziehung der beiden durch (P) und (D) beschriebenen Optimierungsprobleme.

Satz 2.33 (Schwache Dualität) *Sei x zulässig für (P), π zulässig für (D). Dann gilt:*

$$b^t\pi \leq c^t x,$$

d.h. jede zulässige Lösung von (D) ist eine untere Schranke von (P).

Beweis:

$$\begin{array}{rcll} b^t\pi & = & (Ax)^t\pi & \text{weil } x \text{ für (P) zulässig ist} \\ & = & x^t A^t \pi & \\ & \leq & x^t c = c^t x, & \end{array}$$

wobei die Abschätzung gilt, weil

- π für (D) zulässig ist (also $A^t\pi \leq c$), und
- x für (P) zulässig ist (also $x \geq 0$). QED

Satz 2.34 (Starke Dualität) *Seien (P) und (D) zueinander duale lineare Programme. Dann gelten die folgenden Beziehungen:*

1. *Ist (P) unbeschränkt so ist (D) unzulässig.*
2. *Sei x zulässig für (P), π zulässig für (D) und gelte $c^t x = b^t \pi$. Dann sind x und π optimal.*
3. *Hat (P) eine endliche Optimallösung, so auch (D) und die optimalen Zielfunktionswerte von (P) und von (D) sind gleich.*

Die ersten beiden Aussagen gelten auch, wenn man (P) und (D) vertauscht.

Beweis:

1. Sei (P) unbeschränkt. Angenommen, es existiert eine für das Duale (D) zulässige Lösung π'. Dann gilt nach der schwachen Dualität (Satz 2.33) für alle primal zulässigen x:
$$c^t x \geq b\pi' \in \mathbb{R},$$
also ist $b\pi'$ eine untere Schranke für (P). Das ist jedoch ein Widerspruch, weil wir (P) als unbeschränkt angenommen hatten.

2. Zum Beweis der zweiten Aussage nehmen wir an, dass x zulässig ist für (P), π zulässig ist für (D), und $c^t x = b^t \pi$.
Wir zeigen, dass x optimal ist: Betrachten wir dazu ein zulässiges x'. Die schwache Dualität liefert $c^t x' \geq \pi^t b = c^t x$ und zeigt damit die Optimalität von x. Analog erhält man die Optimalität von π.

3. Wenn (P) endlich lösbar ist, so existiert nach dem Hauptsatz der linearen Optimierung (Satz 2.21) eine optimale Basislösung. Aufgrund der Endlichkeit des Simplex-Verfahrens können wir sogar von einer Basislösung $x = (x_B, x_N)$ ausgehen, deren reduzierte Kosten alle größer oder gleich Null sind (siehe Satz 2.27). Genauer wissen wir, dass $x_B = A_B^{-1} b$, $x_N = 0$ und dass
$$\bar{c}^t = c^t - c_B^t A_B^{-1} A \geq 0,$$
oder, in transponierter Form, dass
$$\bar{c} = c - A^t (A_B^{-1})^t c_B \geq 0 \tag{2.10}$$
gilt. Man definiert nun
$$\pi = (A_B^{-1})^t c_B$$
und zeigt, dass dieser Vektor eine Optimallösung des Dualen mit der geforderten Eigenschaft ist. Dazu sind die folgenden drei Punkte nachzuweisen:

- Zunächst ist zu zeigen, dass π zulässig ist, d.h. dass $A^t\pi \leq c$ gilt. Das gewährleistet (2.10):

$$c - A^t\pi = c - A^t(A_B^{-1})^t c_B \geq 0.$$

- Der Zielfunktionswert von π lässt sich nachrechnen zu

$$b^t\pi = \pi^t b = ((A_B^{-1})^t c_B)b = c_B^t A_B^{-1} b = c_B^t x_B = c_B^t x_B + c_N^t x_N = c^t x.$$

- Die Optimalität von π folgt nun aus der schon bewiesenen zweiten Aussage des Satzes.

Die Vertauschung von (P) und (D) in den ersten beiden Aussagen des Beweises gilt wegen Lemma 2.30. QED

Die Aussage des Satzes kann man an folgendem Schema verdeutlichen, das angibt, welche Konstellationen von primalen und dualen Programmen auftreten können.

	(D) optimal lösbar	(D) unbeschränkt	(D) unzulässig
(P) optimal lösbar	ja	nein	nein
(P) unbeschränkt	nein	nein	ja
(P) unzulässig	nein	ja	ja

Dabei folgt die Aussage, dass (P) und (D) gleichzeitig unzulässig sein können nicht aus dem Satz über die Starke Dualität, man kann sich dazu aber leicht Beispiele erzeugen (ÜBUNG).

Im folgenden untersuchen wir, wie man das duale Programm und seine Lösungen im Simplex-Tableau erkennen kann. Betrachten wir ein lineares Programm in Standardform,

$$\begin{aligned} \min \quad & c^t x \\ \text{s.d. } & Ax = b \qquad (P) \\ & x \geq 0, \end{aligned}$$

und sein Duales,

$$\begin{aligned} \max \quad & b^t\pi \\ \text{s.d. } & A^t\pi \leq c \qquad (D) \\ & \pi \gtreqless 0. \end{aligned}$$

Setzen wir voraus, dass das lineare Programm schon mit einer Basis gegeben ist und wir es in Form eines Simplextableaus vorliegen haben. Das heißt, wir nehmen

an, dass $c^t = (\mathbf{c}^t|0)$ und $A = (\mathbf{A}|I)$. Setzen wir diese Form voraus, so sind die Nebenbedingungen des Dualen äquivalent zu $\mathbf{A^t}\pi \leq \mathbf{c}$ und $\pi \leq 0$.

Sei nun B eine beliebige Basis. Das Tableau stellt sich wie folgt dar:

$$T(B) = \begin{array}{|c|c|c|c|} \hline 1 & c^t - c_B^t A_B^{-1}\mathbf{A} & 0 - c_B^t A_B^{-1} I & -c_B A_B^{-1} b \\ \hline 0 & A_B^{-1}\mathbf{A} & A_B^{-1} I & A_B^{-1} b \\ \hline \end{array}$$

Die primale Lösung zu B ist $(x_B, x_N) = (A_B^{-1}b, 0)$, die duale Lösung zur Basis B ist $\pi^t = c_B^t A_B^{-1}$ (wie wir sie im Beweis des Satzes über die starke Dualität konstruiert haben). Die primale Lösung kann also in der rechten Spalte, die duale Lösung in der obersten Zeile des Tableaus abgelesen werden.

Falls B eine zulässige Basis ist, können wir für die zugehörige zulässige Basislösung (x_B, x_N) und die duale Lösung π folgende Aussage treffen:

$$\text{Duale Lösung } \pi \text{ ist zulässig} \iff \text{primale Lösung } (x_B, x_N) \text{ ist optimal,}$$

und in diesem Fall gilt $b^t\pi = c_B A_B^{-1} b = c^t x$.

2.5.3 Dualität bei allgemeinen linearen Programmen

Weil man jedes lineare Programm zu einem äquivalenten linearen Programm in Standardform transformieren kann (siehe Lemma 2.4 auf Seite 14), gelten die vorhergehenden Resultate (Lemma 2.32, Satz 2.33 und Satz 2.34) entsprechend auch für beliebige lineare Programme. Dennoch ist es hilfreich, ein lineares Programm in allgemeiner Form direkt dualisieren zu können, ohne es vorher in Standardform zu bringen. Daher wollen wir im folgenden untersuchen, wie das Duale zu einem linearen Programm in allgemeiner Form aussieht.

Lemma 2.35 *Ein lineares Programm in allgemeiner Form*

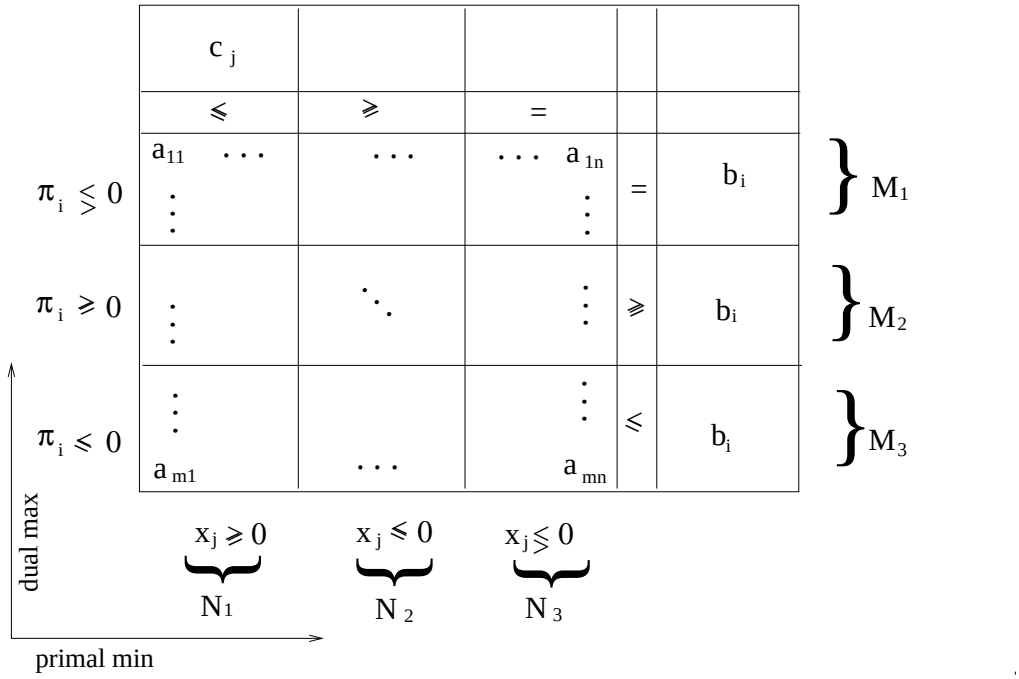
$$\begin{array}{rll} \min & c^t x & \\ \text{s.d. } A^i x = b_i & i \in M_1 & (A^i = i\text{-te Zeile von } A) \\ A^i x \geq b_i & i \in M_2 & \\ A^i x \leq b_i & i \in M_3 & \\ x_j \geq 0 & j \in N_1 & \\ x_j \leq 0 & j \in N_2 & \\ x_j \gtrless 0 & j \in N_3 & \end{array}$$

mit $n = |N_1|+|N_2|+|N_3|$ *Variablen hat das folgende lineare Programm als Duales:*

$$
\begin{aligned}
\max \quad & b^t\pi & & \\
\text{s.d. } & (A_j)^t\pi \leq c_j & \forall j \in N_1 \quad & ((A_j)^t = \textit{transponierte } j\textit{-te Spalte von } A) \\
& (A_j)^t\pi \geq c_j & \forall j \in N_2 & \\
& (A_j)^t\pi = c_j & \forall j \in N_3 & \\
& \pi_i \gtrless 0 & i \in M_1 & \\
& \pi_i \geq 0 & i \in M_2 & \\
& \pi_i \leq 0 & i \in M_3 &
\end{aligned}
$$

Der Beweis kann analog zu Lemma 2.32 geführt werden, indem man das lineare Programm zunächst von allgemeiner Form in Standardform überführt, dann dualisiert und das so erhaltene Duale umformuliert bis das Ergebnis des Lemmas gezeigt ist.
Zum schnellen Merken der Dualisierungsregeln für linearen Programme in allgemeiner Form bietet sich das so genannte *Tucker-Diagramm* an.

Tucker − Diagramm



Die Zeilen des Tucker-Diagramms entsprechen dabei den primalen Nebenbedingungen (von links nach rechts) die Spalten werden von unten nach oben gelesen und entsprechen den zugehörenden dualen Nebenbedingungen.

Beispiel 2.11 *Wir demonstrieren das Dualisieren eines linearen Programms in allgemeiner Form an folgendem Beispiel.*

$$\begin{array}{lrcll} \min & -2x_1 + 3x_2 & & & \\ \text{s.d.} & -x_1 + x_2 - x_3 & = 1 & \pi_1 & \\ & 3x_1 - x_2 + x_4 & \geq 8 & \pi_2 & (P) \\ & x_1, x_3 & \geq 0 & & \\ & x_2, x_4 & \gtrless 0 & & \end{array}$$

Am einfachsten ordnet man — wie hier geschehen — zunächst jeder Nebenbedingung eine duale Variable (hier π_1, π_2) zu. Das Duale des gegebenen linearen Programms ergibt sich als

$$\begin{array}{lrcl} \max & \pi_1 + 8\pi_2 & & \\ \text{s.d.} & -\pi_1 + 3\pi_2 & \leq -2 & \\ & \pi_1 - \pi_2 & = 3 & (D) \\ & -\pi_1 & \leq 0 & \\ & \pi_2 & = 0 & \\ & \pi_1 & \gtrless 0 & \\ & \pi_2 & \geq 0 & . \end{array}$$

In diesem Fall ist die Lösung des Dualen einfach. Zunächst nutzen wir $\pi_2 = 0$. Damit reduziert sich (D) zu

$$\begin{array}{lrcl} \max & \pi_1 & & \\ \text{s.d.} & \pi_1 & \geq 2 & \\ & \pi_1 & = 3 & \\ & \pi_1 & \geq 0 & . \end{array}$$

Als einzige zulässige Lösung ergibt sich folglich $\pi_1 = 3$ und $\pi_2 = 0$ mit Zielfunktionswert 3. Für das Primale folgt nach Satz 2.34, dass jede zulässige Lösung mit Zielfunktionswert 3 optimal ist. Durch Raten erhält man als Lösung des Primalen

$$x_1 = 0, \;\; x_2 = 1, \;\; x_3 = 0, \;\; x_4 = 9.$$

Eine Methode, wie man aus der Kenntnis einer dualen Optimallösung eine primale Lösung konstruieren kann, folgt im nächsten Abschnitt.

2.5.4 Folgerungen aus der Dualität

In diesem Abschnitt sollen einige Anwendungen der Dualitätstheorie für lineare Programme vorgestellt werden. Wir gehen zunächst auf eine für die lineare Optimierung klassische Folgerung, nämlich den Satz des komplementären Schlupfes ein. Dann schauen wir uns die so genannten *Alternativsätze* an und machen schließlich einen Exkurs zum Minmax-Theorem der Spieltheorie.

Komplementärer Schlupf

Satz 2.36 (Satz vom komplementären Schlupf) *Seien (P) und (D) zueinander duale lineare Programme. Sei x zulässig für (P) und π zulässig für (D). Definiere*

$$\begin{aligned} u_i &:= \pi_i(A^i x - b_i) \quad \forall i = 1, \ldots, m \text{ und} \\ v_j &:= x_j(c_j - (A_j)^t \pi) \quad \forall j = 1, \ldots, n. \end{aligned}$$

Dann ist x optimal für (P) und π optimal für (D) genau dann wenn

$$u_i = 0 \quad \text{für alle } i = 1, \ldots, m \text{ und} \quad v_j = 0 \quad \text{für alle } j = 1, \ldots, n.$$

Beweis: Wir führen den Beweis für lineare Programme in allgemeiner Form, obwohl es auch hier reichen würde, die vereinfachte Aussage des Satzes für lineare Programme in Standardform zu zeigen.

Durch Fallunterscheidung zeigen wir zunächst, dass $u_i \geq 0$ für alle $i = 1, \ldots, m$. Das gilt, denn

- Sei $i \in M_1 \quad \Rightarrow \quad A^i x = b_i \quad \Rightarrow \quad u_i = 0.$
- Sei $i \in M_2 \quad \Rightarrow \quad A^i x \geq b_i \;\; und \;\; \pi_i \geq 0 \;, \; also \; u_i \geq 0.$
- Sei $i \in M_3 \quad \Rightarrow \quad A^i x \leq b_i \;\; und \;\; \pi_i \leq 0 \;, \; also \; u_i \geq 0.$

Ganz analog zeigt man, dass $v_j \geq 0$.

Man definiert nun

$$\begin{aligned} u &:= \sum_{i=1}^{m} u_i = \pi^t(Ax - b) \geq 0, \\ v &:= \sum_{i=1}^{m} v_i = x^t(c - A^t \pi) \geq 0. \end{aligned}$$

Dann gilt

$$u_i = 0 \forall i = 1, \ldots, m \text{ und } v_j = 0 \text{ für alle } = 1, \ldots, n$$

$$
\begin{aligned}
&\iff u + v = 0 \\
&\iff \pi^t(Ax - b) + x^t(c - A^t\pi) = 0 \\
&\iff \pi^t Ax - \pi^t b + x^t c - x^t A^t \pi = 0 \\
&\iff \pi^t b = x^t c \\
&\iff \pi, x \text{ sind optimal.}
\end{aligned}
$$

Die letzte Aussage folgt aus dem Satz über die starke Dualität (Satz 2.34). QED

Liegt das lineare Programm (P) in Standardform vor, so gilt für jede primal zulässige Lösung x dass $A^i x = b_i$ für alle $i = 1, \ldots, m$. Dementsprechend ergibt sich

$$u_i = \pi_i(A^i x - b_i) = 0 \quad \forall i = 1, \ldots, m$$

für alle primal zulässigen Lösungen x und dual zulässigen Lösungen π. Dadurch vereinfacht sich die Aussage von Satz 2.36 für lineare Programme (P) in Standardform. Sie lautet entsprechend:

Sei x zulässig für (P) und π zulässig für (D). Dann ist x optimal für (P) und π optimal für (D) genau dann wenn

$$x^t(c - A^t\pi) = 0.$$

Man kann den Satz vom komplementären Schlupf verwenden, um aus der Kenntnis einer primalen optimalen Lösung eine duale optimale Lösung zu konstruieren, und umgekehrt. Das wird im nächsten Beispiel demonstriert.

Beispiel 2.12 *(Fortsetzung von Beispiel 2.11)*
Das lineare Programm und sein Duales werden im folgenden nochmals angegeben. Zur Anwendung des Satzes vom komplementären Schlupf empfiehlt es sich, die zugehörigen dualen Variablen hinter die primalen Nebenbedingungen (und umgekehrt) zu schreiben.

$$
\begin{aligned}
&(\mathrm{P}) \quad \min \quad -2x_1 + 3x_2 \\
&\text{s.d.} \quad -x_1 + x_2 - x_3 = 1 \qquad \pi_1 \\
&\qquad\quad 3x_1 - x_2 + x_4 \geq 8 \qquad \pi_2 \\
&\qquad\qquad\quad x_1, x_3 \geq 0, \quad x_2, x_4 \gtrless 0
\end{aligned}
$$

$$
\begin{aligned}
&(\mathrm{D}) \quad \max \quad \pi_1 + 8\pi_2 \\
&\text{s.d.} \quad -\pi_1 + 3\pi_2 \leq -2 \quad x_1
\end{aligned}
$$

$$\begin{aligned} \pi_1 - \pi_2 = 3 \quad & x_2 \\ -\pi_1 \leq 0 \quad & x_3 \\ \pi_2 = 0 \quad & x_4 \end{aligned}$$

Formuliert man die Bedingungen für den komplementären Schlupf, so erhält man folgendes nichtlineares Optimierungsproblem:

$$\begin{aligned} u_1 &= \pi_1(-x_1 + x_2 - x_3 - 1) = 0 \\ u_2 &= \pi_2(3x_1 - x_2 + x_4 - 8) = 0 \\ v_1 &= x_1(-\pi_1 + 3\pi_2 + 2) = 0 \\ v_2 &= x_2(\pi_1 - \pi_2 - 3) = 0 \\ v_3 &= x_3(-\pi_1) = 0 \\ v_4 &= x_4(\pi_2) = 0 \end{aligned}$$

Kennt man nun eine (duale) Optimallösung, so erhält man aus obigem nichtlinearen Programm ein neues lineares Programm. Für $\pi = (3, 0)^t$ *ergibt sich beispielsweise:* $x = (x_1, x_2, x_3, x_4)^t$ *ist optimal, genau dann wenn es das folgende lineare Programm löst:*

$$\begin{aligned} -x_1 + x_2 - x_3 - 1 &= 0 \\ 3x_1 - x_2 + x_4 - 8 &\geq 0 \\ x_1 &= 0 \\ x_3 &= 0. \end{aligned}$$

Setzt man $x_1 = x_3 = 0$ *in die ersten beiden Bedingungen ein, so erhält man:* $x = (x_1, x_2, x_3, x_4)$ *ist optimal genau dann wenn*

$$\begin{aligned} x_1 &= 0 \\ x_3 &= 0 \\ x_2 &= 1 \textit{ und} \\ -1 + x_4 - 8 &\geq 0. \end{aligned}$$

Also sind alle Vektoren der Form $x = (0, 1, 0, x_4)$ *mit* $x_4 \geq 9$ *Optimallösungen von (P).*

Alternativsätze

Eine weitere wichtige Anwendung der Dualität sind die Alternativsätze. Der bekannteste unter ihnen ist vermutlich der folgende Satz.

Satz 2.37 (Farkas' Lemma) *Sei A eine $m \times n$-Matrix und sei $c \in \mathbb{R}^n$. Dann existiert ein Vektor $y \in \mathbb{R}^m$ mit $A^t y = c$ und $y \geq 0$ genau dann, wenn für alle $x \in \mathbb{R}^n$ mit $Ax \leq 0$ gilt: $c^t x \leq 0$*

Man nennt diesen Satz *Alternativsatz*, weil immer genau eine der beiden Alternativen eintritt:

- Entweder hat das System $A^t y = c, y \geq 0$ eine Lösung,
- oder das System $Ax \leq 0, c^t x > 0$ hat eine Lösung.

Beweis: Betrachte die beiden zueineander dualen Programme

$$(P) \quad \max\{c^t x : Ax \leq b, , x \gtrless 0\} \qquad \text{und} \qquad \min\{0 : A^t y = c, y \geq 0\}$$

Mit Hilfe dieser beiden linearen Programme erhalten wir:

$\Longrightarrow$**:** Sei $A^t y = c, y \geq 0$ lösbar. Dann ist also (D) lösbar; der optimale Zielfunktionswert beträgt 0. Nach der starken Dualität (Satz 2.34) ist auch (P) lösbar und der optimale Zielfunktionswert ist ebenfalls 0. Das heißt, für alle $x \in \mathbb{R}^n$ mit $Ax \leq 0$ gilt $c^t x \leq 0$.

$\Longleftarrow$**:** Sei $A^t y = c, y \geq 0$ nicht lösbar. Dann ist (D) nicht lösbar. Nach der starken Dualität (Satz 2.34) ist also auch (P) nicht lösbar, d.h. (P) ist entweder unzulässig oder unbeschränkt. Weil $x = 0$ eine zulässige Lösung von (P) ist, ist (P) unbeschränkt. Insbesondere hat (P) eine zulässige Lösung x mit Zielfunktionswert > 0. Das wiederum heißt, es gibt ein x mit $Ax \leq 0$ und $c^t x > 0$.

QED

Eine direkte Folgerung liefert der folgende Satz, den wir in der nichtlinearen Optimierung zum Beweis der Fritz-John-Bedingungen (Satz 4.41 auf Seite 162) wieder verwenden werden.

Satz 2.38 (Gordan'scher Transpositionssatz) *Sei A eine $m \times n$-Matrix. Dann gilt:*

$$\nexists x \in \mathbb{R}^n : Ax < 0 \Longleftrightarrow \exists y \geq 0, y \neq 0 : A^t y = 0.$$

Auch dieser Satz ist ein Alternativsatz, weil genau eine der beiden folgenden Alternativen eintritt:

- Entweder es existiert ein $x \in \mathbb{R}^n$ so dass $Ax < 0$,
- oder es existiert ein $y \geq 0, y \neq 0$ so dass $A^t y = 0$.

Beweis: Man wendet Farkas' Lemma (Satz 2.37) an auf

$$A' = \begin{pmatrix} & & 1 \\ & A & \vdots \\ & & 1 \end{pmatrix}, \quad c = (0, \ldots, 0|1).$$

und erhält

$$\exists y \in \mathbb{R}^m, y \geq 0 \text{ mit } \begin{pmatrix} & A^t & \\ 1 & \ldots & 1 \end{pmatrix} y = \begin{pmatrix} 0 \\ \vdots \\ 0 \\ 1 \end{pmatrix} \tag{2.11}$$

$$\iff \forall x \in \mathbb{R}^{n+1} \text{ mit } \begin{pmatrix} & & 1 \\ & A & \vdots \\ & & 1 \end{pmatrix} x \leq 0 \text{ gilt: } x_{n+1} \leq 0. \tag{2.12}$$

Aussage (2.11) ist äquivalent zur rechten Seite

$$\exists y \in \mathbb{R}^m, y \geq 0 \text{ mit} A^t y = 0 \; y \neq 0$$

in Gordan's Transpositionssatz, weil aus $y \geq 0, A^t y = 0, y \neq 0$ auch $\alpha y \geq 0, A^t(\alpha y) = 0, \alpha y \neq 0$ für alle $\alpha > 0$ folgt.

Zu zeigen bleibt somit, dass

$$(2.12) \iff \nexists x \in \mathbb{R}^n : Ax < 0$$

gilt. Das sieht man folgendermaßen:

"$\Longrightarrow$" Gelte (2.12). Angenommen, es existiert ein $x' \in \mathbb{R}^n$ mit $Ax' < 0$. Dann gibt es $\epsilon > 0$ so dass

$$Ax' + \epsilon \begin{pmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{pmatrix} \leq 0,$$

also gibt es $x = (x', \epsilon)$ mit $Ax' \leq 0$ und $x_{n+1} = \epsilon > 0$, ein Widerspruch zu (2.12).

"$\Longleftarrow$" Sei $Ax' < 0$ nicht lösbar. Sei $x = (x', x_{n+1}) \in \mathbb{R}^{n+1}$. Gelte weiterhin

$$Ax' + x_{n+1} \begin{pmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{pmatrix} = \begin{pmatrix} & & 1 \\ & A & \vdots \\ & & 1 \end{pmatrix} \begin{pmatrix} x' \\ x_{n+1} \end{pmatrix} \leq 0. \tag{2.13}$$

Wir möchten zeigen, dass $x_{n+1} \leq 0$. Das gilt, denn wäre $x_{n+1} > 0$, so würde (2.13) implizieren, dass $Ax' < 0$, ein Widerspruch dazu, dass $Ax' < 0$ nicht lösbar ist.

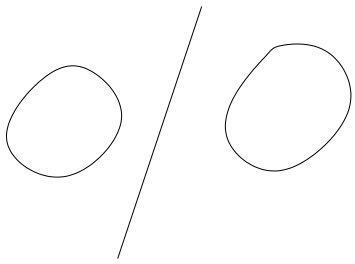
Es folgt also $x_{n+1} \leq 0$, also gilt (2.12). QED

Wir legen an dieser Stelle einen Exkurs ein, der einen anderen Zugang zur Dualitätstheorie beschreibt. Man kann nämlich zeigen, dass das Lemma von Farkas und die Aussage der starken Dualität äquivalent sind! D.h. es gilt:

$$\text{Farkas' Lemma (Satz 2.37)} \iff \text{Starke Dualität (Satz 2.34).}$$

Die eine Richtung dieser Aussage, nämlich wie man unter Verwendung des Satzes über die starke Dualität Farkas' Lemma beweist, haben wir schon gesehen: So haben wir auf Seite 67 das Lemma von Farkas bewiesen. Die andere Richtung ist ein alternativer Beweis für die starke Dualität ohne Verwendung des Hauptsatzes der linearen Optimierung, und bietet damit einen ganz anderen Zugang zu dieser Theorie als der hier geschilderte. Im folgenden geben wir eine Skizze dieses Beweises.

Zunächst benötigt man einen der so genannten *Trennungssätze*: Zwei konvexe, disjunkte, kompakte Mengen lassen sich wie in der folgenden Skizze dargestellt, durch eine Hyperebene trennen.



Aus dieser Aussage kann man Farkas' Lemma herleiten, ganz ohne die Dualitätstheorie zu verwenden. Aus Farkas' Lemma wiederum ergibt sich der folgende Satz von Tucker (Beweis ÜBUNG).

Satz 2.39 (Satz von Tucker) *Sei A eine schiefsymmetrische $n \times m$ Matrix, d.h. gelte $A^t = -A$. Dann gibt es $w \in \mathbb{R}^n$, $w \geq 0$ so dass $Aw \geq 0$ und $Aw + w > 0$.*

Unter Verwendung des Satzes von Tucker beweist man die starke Dualität folgendermaßen: Seien

$$\begin{aligned} &\text{(P)} \quad \max\{c^t x : Ax \leq b, x \geq 0\} \\ &\text{(D)} \quad \min\{b^t \pi : A^t \pi \geq c, \pi \geq 0\} \end{aligned}$$

zwei zueinander duale Programme. Wir wollen zeigen, dass entweder beide Programme endlich lösbar sind, und sie in diesem Fall den gleichen Zielfunktionswert haben, oder keines der beiden Programme (P) und (D) endlich lösbar ist. Im Beweis werden dazu die folgenden Aussagen verwendet:

[A1] Satz von Tucker

[A2] schwache Dualität, d.h. $c^t x \leq \pi^t b$ für alle x, π mit x zulässig für (P) und π zulässig für (D)

[A3] x, π zulässig und $c^t x = \pi^t b$. Dann folgt, dass x optimal für (P) und π optimal für (D).

Dazu ist zu bemerken, dass man alle drei Aussagen ohne den Hauptsatz der linearen Optimierung beweisen kann (wie im Fall der beiden letzteren auch geschehen).

Der Beweis für die starke Dualität sieht dann wie folgt aus: Definiere folgende schiefsymmetrische $(n+m+1) \times (n+m+1)$-Matrix

$$B = \begin{pmatrix} 0_n & A^t & -c \\ -A & 0_m & b \\ c^t & -b^t & 0 \end{pmatrix}.$$

Nach dem Satz von Tucker ([A1]) existiert $w = \begin{pmatrix} x \\ \pi \\ t \end{pmatrix} \geq 0$ so dass $Bw \geq 0$ und $Bw + w > 0$, wobei

$$Bw = \begin{pmatrix} A^t\pi - tc \\ -Ax + tb \\ c^t x - b^t \pi \end{pmatrix} \quad x \in \mathbb{R}^n,\ \pi \in \mathbb{R}^m,\ t \in \mathbb{R}.$$

Wegen $w \geq 0$, $Bw \geq 0$ und $Bw + w > 0$ gilt:

(1) $x \geq 0,\ \pi \geq 0,\ t \geq 0$

(2) $A^t\pi - t \cdot c \geq 0$

(3) $-Ax + t \cdot b \geq 0$

(4) $c^t x - b^t \pi \geq 0$

(5) $A^t\pi - t \cdot c + x > 0$

(6) $-Ax + t \cdot b + \pi > 0$

(7) $c^t x - b^t \pi + t > 0.$

Man unterscheidet jetzt die folgenden beiden Fälle:

Fall 1: $t > 0$ (Goldman 1955). In diesem Fall definiert man

$$\begin{aligned} x^* &= \frac{1}{t}x, \\ \pi^* &= \frac{1}{t}\pi. \end{aligned}$$

Wir zeigen, dass

1. x^* zulässig für (P) und π^* zulässig für (D), und
2. dass $c^t x^* = b^t \pi^*$.

Daraus folgt nach [A3], dass x^*, π^* optimal, also beide Programme endlich lösbar sind und den gleichen Zielfunktionswert haben.

ad 1. Zulässigkeit: Zunächst gilt $x^* \geq 0$ und $\pi^* \geq 0$ wegen (1). Weiterhin folgern wir

$$Ax^* \leq b$$

weil nach (3) $Ax \leq t \cdot b$ gilt, und analog

$$A^t\pi^* \geq c,$$

denn aus (2) wissen wir dass $A^t\pi \geq t \cdot c$.

ad 2. Die Gleichheit der Zielfunktionswerte $c^t x^* = b^t \pi^*$ folgt aus (4) und der schwachen Dualität [A2].

Fall 2: $t = 0$ (Goldman/Tucker 1956). In diesem Fall zeigt man, dass dass (P) und (D) nicht beide zulässige Lösung haben können. Angenommen doch, dann gibt es eine Lösung $\bar{x}$ von (P) und eine Lösung $\bar{\pi} \geq 0$ von (D), mit

$$\begin{aligned} A\bar{x} &\leq b, \\ A^t\bar{\pi} &\geq c. \end{aligned}$$

Dann gilt:

$$\begin{aligned} b^t\bar{\pi} &\geq (A\bar{x})^t\bar{\pi} \quad \text{weil } b \geq A\bar{x} \text{ und } \bar{\pi} \geq 0 \\ &= \bar{x}^t A^t \bar{\pi} \\ &\geq 0 \quad \text{weil } A^t\bar{\pi} \geq 0 \text{ wegen (2) und } \bar{x} \geq 0 \end{aligned}$$

Andererseits

$$\begin{aligned} b^t\bar{\pi} &< c^t\bar{x} \quad \text{wegen (7)} \\ &\leq (A^t\bar{\pi})^t\bar{x} \quad \text{weil } A^t\bar{\pi} \geq c \text{ und } \bar{x} \geq 0 \\ &= \bar{\pi}^t A\bar{x} \\ &\leq 0 \quad \text{weil } A\bar{x} \leq 0 \text{ nach (3) und } \bar{\pi} \geq 0\ , \end{aligned}$$

ein Widerspruch! QED

Sattelpunkte in der Spieltheorie

Als dritte Anwendung der Dualitätssätze soll hier noch ein grundlegendes Ergebnis der Spieltheorie vorgestellt werden. Wir betrachten zwei Spieler, Spieler 1 und Spieler 2. Spieler 1 hat die Strategien $S_1, \ldots, S_{m_1}$ zur Auswahl und Spieler 2 die Strategien $T_1, \ldots, T_{m_2}$. Beide Spieler wählen nun gleichzeitig (und heimlich) eine ihrer Strategien aus und erhalten danach eine Auszahlung, die aber von der Wahl beider Spieler anhängt. Man kann so ein Spiel anhand einer Tabelle darstellen.

		Spieler 2
		$T_1 \quad \cdots \quad T_{m_2}$
Spieler 1	S_1 $\vdots$ S_{m_1}	A

Dabei bezeichnen die Werte der Matrix $A = (a_{ij})$ die Auszahlungen der Spieler. Genauer ist a_{ij} die Auszahlung für Spieler 1, wenn er die Strategie S_i wählt und Spieler 2 die Strategie T_j. Gleichzeitig erhält in diesem Fall Spieler 2 die Auszahlung $-a_{ij}$. Die Auszahlungsmatrix ist den Spielern bekannt.

Das Spiel gehört in der Gruppe der **Matrixspiele**. Weil die Auszahlung von Spieler 1 immer das Negative der Auszahlung von Spieler 2 ist liegt hier ein **Nullsummen-Matrixspiel** für zwei Spieler vor. Im folgenden soll untersucht werden, welche Strategie die beiden Spieler wählen sollten, um ihre Auszahlung zu maximieren.

Beispiel 2.13 *Ein Beispiel für ein Nullsummen-Matrixspiel für zwei Spieler ist das folgende unter Schere-Stein-Papier bekannte Spiel. Die Auszahlungsmatrix hat folgende Gestalt:*

	Papier	Stein	Schere
Papier	0	1	-1
Stein	-1	0	1
Schere	1	-1	0

Als nächstes wollen wir "stabile" Zustände von Nullsummen-Matrixspielen untersuchen. Dazu benötigen wir die folgende Definition.

Definition 2.40 *Seien*

$$\begin{aligned} v_1 &= \max_{i=1,\ldots,m_1} \min_{j=1,\ldots,m_2} a_{ij} \\ v_2 &= \min_{j=1,\ldots,m_2} \max_{i=1,\ldots,m_1} a_{ij}. \end{aligned}$$

Gilt $v_1 = v_2$, so hat das Spiel einen **Sattelpunkt**.

Ein Sattelpunkt beschreibt einen *stabilen* Zustand des Spiels, in dem keiner der beiden Spieler rückwirkend seine Entscheidung ändern möchte. Ein Sattelpunkt wird daher auch als *Gleichgewicht in reinen Strategien* bezeichnet. Diese Bedeutung soll an dem folgenden Beispiel demonstriert werden.

Beispiel 2.14 *Wir betrachten das folgende Nullsummen-Matrixspiel für zwei Spieler. Spieler 1 hat drei Strategien, Spieler 2 kann unter vier Strategien wählen. Die Auszahlungsmatrix für Spieler 1 sieht wie folgt aus:*

	T_1	T_2	T_3	T_4	*worst case Spieler 1*
S_1	-4	*2*	*3*	-1	-4
S_2	*5*	-1	*2*	-2	-2
S_3	*1*	*2*	*3*	*0*	**0**
worst case Spieler 2	*5*	*2*	*3*	**0**	

In der Spalte ganz rechts ist zu jeder Strategie S_i der jeweils schlechteste Fall eingetragen, der Spieler 1 widerfahren kann, wenn er die Strategie S_i wählt. Wählt Spieler 1 also beispielsweise Strategie S_1, so würde er im Fall dass Spieler 2 Strategie T_1 wählt einen Verlust von -4 hinnehmen müssen. Das ist das schlimmste, was ihm bei Strategie S_1 passieren kann. (Wählt Spieler 2 nämlich Strategie T_2 so erhält Spieler 1 einen Gewinn von 2, bei T_3 sogar einen Gewinn von 3 und bei T_4 einen Verlust von -1.)
Denkt Spieler 1 also sicherheitsorientiert, so wird er versuchen, den schlimmsten Fall für ihn so gut wie möglich zu gestalten, und daher Strategie S_3 wählen, bei der er im schlimmsten Fall Null auf Null herauskommt. Der Wert v_1 für Spieler 1 ist also

$$v_1 = \max\{-4, -2, 0\} = 0.$$

Die gleiche Überlegung ist für den Spieler 2 in der untersten Zeile der Tabelle eingetragen. Da Spieler 2 immer das Negative der eingetragenen Werte a_{ij} ausgezahlt wird, minimiert er sein Risiko bei Wahl der Strategie T_4, und es gilt

$$v_2 = \min\{5, 2, 3, 0\} = 0.$$

Zusammen erhält man, dass $v_1 = v_2$, das Spiel hat also den Sattelpunkt (S_3, T_4).

Eine wichtige Bedeutung des Sattelpunktes wird hier klar: Angenommen, Spieler 1 wählt nun wirklich S_3 und Spieler 2 entscheidet sich für T_4. Dann sind rückwirkend beide Spieler mit ihren Entscheidungen zufrieden, in folgendem Sinne:
Wenn Spieler 1 von der Entscheidung des zweiten Spieler erfährt, wird er seine Strategie anhand der Auszahlungsmatrix überprüfen und feststellen, dass er sehr gut gewählt hat: Seine Auszahlung wäre sowohl bei S_1 als auch bei S_2 schlechter gewesen. Ebenso ergeht es Spieler 2: Wenn er nach Bekanntgabe der von Spieler 1 getroffenen Entscheidung seine Strategie überprüft, so sieht auch er, dass

er das bestmögliche Ergebnis erreicht hat. Keiner der beiden Spieler würde also rückblickend seine Entscheidung revidieren. Einen solchen Zustand nennt man **stabil** *oder ein* **Gleichgewicht**.

Es gibt nicht in jedem Matrixspiel Sattelpunkte. Ein Beispiel dafür ist das Spiel aus dem vorhergehenden Beispiel 2.13 (Papier-Stein-Schere), bei dem $v_1 = -1 \neq 1 = v_2$ gilt, und es entsprechend keinen stabilen Zustand gibt.

Um dennoch einen stabilen Zustand zu erreichen, erlaubt man in der Spieltheorie so genannte *gemischte Strategien*. Hier legt jeder Spieler eine Wahrscheinlichkeit für jede seiner Strategien fest und lässt sich dann überraschen. Bezeichne

- x_i die Wahrscheinlichkeit, dass Spieler 1 Strategie S_i wählt, $i = 1, \ldots, m_1$,
- y_j die Wahrscheinlichkeit, dass Spieler 2 Strategie T_j wählt, $j = 1, \ldots, m_2$.

Seien weiterhin

$$x = \begin{pmatrix} x_1 \\ \vdots \\ x_{m_1} \end{pmatrix} \quad \text{und} \quad y = \begin{pmatrix} y_1 \\ \vdots \\ y_{m_2} \end{pmatrix}$$

die Vektoren der Wahrscheinlichkeiten für Spieler 1 und Spieler 2. Man nennt diese Vektoren auch die *gemischten Strategien* der Spieler.

Die erwartete Auszahlung für Spieler 1 ergibt sich als

$$A(x, y) \;=\; \sum_{i=1}^{m_1} \sum_{j=1}^{m_2} x_i a_{ij} y_j \;=\; x^t A y.$$

Spieler 1 möchte nun eine gemischte Strategie x wählen. Dazu überlegt er zunächst, wie Spieler 2 auf eine gegebene Strategie x reagieren würde. Wenn Spieler 2 die gemischte Strategie x des ersten Spielers kennt, würde er seine erwartete Auszahlung berechnen als

$$V(x) \;= \min\{x^t A y : \sum_{j=1}^{m_2} y_j = 1, y_j \geq 0\}.$$

Das ist ein lineares Optimierungsproblem, nämlich

$$\begin{aligned} & \min c^t y \\ \text{s.d.} \quad \sum_{j=1}^{m_2} y_j &= 1 \\ y_j &\geq 0 \quad j = 1, \ldots, m_2, \end{aligned}$$

wobei der Kostenvektor $c^t = x^t A$. Das lineare Programm liegt in Standardform vor, hat nur eine einzige Zeile in der Koeffizientenmatrix und m_2 Variablen, also m_2 Spalten. Entsprechend besteht jede Basis aus einer einzigen Spalte, was

wiederum dazu führt, dass jede Basislösung $m_2 - 1$ Nichtbasisvariablen hat, die also den Wert Null annehmen. Nach dem Hauptsatz der linearen Optimierung (Satz 2.21) gibt es also eine Optimallösung, bei der alle Variablen y_j bis auf eine gleich Null sind, und die letzte den Wert 1 annimmt. Es ergibt sich also

$$V(x) = \min_{j=1,\ldots,m_2} \sum_{i=1}^{m_1} a_{ij} x_i$$

als Lösung des linearen Programms. Rein anschaulich kann man sich das auch ohne Kenntnis des Hauptsatzes klar machen: Spieler 2 untersucht das erwartete Verhalten aller seine Strategien, wenn Spieler 1 Strategie x wählt und wählt schließlich die Strategie mit dem für ihn besten Erwartungswert.

Nach diesen Überlegungen möchte Spieler 1 seine gemischte Strategie x so wählen, dass $V(x)$ maximal wird. Er muss also das Optimierungsproblem

$$V_1 = \max\{V(x) : \sum_{j=1}^{m_1} x_j = 1, x_j \geq 0\} \;=\; \max_x \{\min_j \sum_{i=1}^{m_1} a_{ij} x_i\}$$

lösen. Auch dieses Optimierungsproblem kann man als lineares Programm umschreiben, indem man einen oft verwendeten Trick anwendet und für lineare Funktionen $b_1(x), \ldots, b_{m_2}(x)$ den nichtlinearen Ausdruck

$$\max \min\{b_1(x), \ldots, b_{m_2}(x)\}$$

umschreibt zu

$$\begin{array}{rcl} \max v & & \\ \text{so dass } v & \leq & b_1(x) \\ v & \leq & b_2(x) \\ & \vdots & \\ v & \leq & b_{m_2}(x) \end{array}$$

Man benötigt dazu eine zusätzliche Variable v und m_2 linearen Nebenbedingungen.
Wendet man dieses Vorgehen auf das Optimierungsproblem von Spieler 1 an, so erhält man folgendes lineares Programm:

$$\begin{array}{lcl} \max & v & \\ \text{s.d.} & v \;\leq\; \displaystyle\sum_{i=1}^{m_1} a_{ij} x_i & \text{für alle } j = 1, \ldots, m_2 \\ \displaystyle\sum_{i=1}^{m_1} x_i & = & 1 \end{array}$$

$$\begin{aligned} x_i &\geq 0 \\ v &\gtrless 0. \end{aligned}$$

Wir ersetzten noch v durch $-u$ und erhalten

$$\begin{aligned} \text{(P1)} \qquad & \min \quad u \\ \text{s.d.} \quad & u + \sum_{i=1}^{m_1} a_{ij} x_i \geq 0 \quad \text{für alle } j = 1, \ldots, m_2 \\ & \sum_{i=1}^{m_1} x_i = 1 \\ & x_i \geq 0 \\ & u \gtrless 0. \end{aligned}$$

Eine analoge Überlegung kann man für Spieler 2 durchführen, der seinen erwarteten Verlust minimieren will. Durch Substitution mit einer negativen Variablen wie oben ergibt sich folgendes lineares Programm:

$$\begin{aligned} \text{(P2)} \qquad & \max \quad z \\ \text{s.d.} \quad & z + \sum_{j=1}^{m_2} a_{ij} y_j \leq 0 \quad \text{für alle } i = 1, \ldots, m_1 \\ & \sum_{j=1}^{m_2} y_j = 1 \\ & y_j \geq 0 \\ & z \lesseqgtr 0 \end{aligned}$$

Man sieht, dass die beide Programme (P1) und (P2) der Spieler 1 und 2 zueinander dual sind. Der starke Dualitätssatz (Satz 2.34) gibt uns daher die folgende Aussage.

Satz 2.41 (Minmax-Satz der Spieltheorie von J.v. Neumann) *Bei einem zwei Personen Matrix-Nullsummenspiel existieren gemischte Strategien x^* für Spieler 1 und y^* für Spieler 2 so, dass der maximale erwartete Gewinn von Spieler 1 gleich dem erwarteten minimalen Verlust von Spieler 2 ist.*

Einen solchen Zustand nennt man auch ein *Gleichgewicht in gemischten Strategien.*
ÜBUNG: Berechnen Sie den Sattelpunkt für das Spiel Schere-Stein-Papier.

2.5.5 Duale Simplex-Varianten

In diesem Abschnitt nutzen wir die Dualitätstheorie nun noch algorithmisch, um andere Simplex-Varianten herzuleiten. Diese können — je nach Struktur des linearen Optimierungsproblems — effizienter sein als das bisher diskutierte primale Simplex-Verfahren.

Duales Simplex-Verfahren

Die Idee des dualen Simplex-Verfahrens besteht darin, eine eventuell angenehmere Struktur des dualen Programms auszunutzen, und das Duale (D) anstatt des original gegebenen Primalen (P) zu lösen. Das könnte man natürlich machen, indem man das gegebene Problem (P) dualisiert und dann das Simplex-Verfahren auf sein Duales anwendet. Man kann sich das Dualisieren dabei aber auch sparen, und das duale Simplex-Verfahren direkt im primalen Tableau ausführen.

Sei dazu ein Tableau $T(B)$ zu einer Basis B gegeben. In Analogie zu unseren Überlegungen auf Seite 61 nennt man das Tableau **dual zulässig** falls die Kostenzeile $\bar{c} \geq 0$ erfüllt. Das duale Simplex-Verfahren lässt sich nun wie folgt beschreiben:

- falls $T(B)$ auch primal zulässig (d.h. $\tilde{b} \geq 0$), so ist $\pi^t = c_B A_B^{-1}$ optimal für (D) und (x_B, x_N) optimal für (P).
- falls es ein $\tilde{b}_i < 0$ gibt, wähle i als Pivotzeile und die Pivotspalte mittels der dualen Quotientenregel:

  $$\min\{\frac{\bar{c}}{-\tilde{a}_{ij}} : \tilde{a}_{ij} < 0\}$$

  Falls alle $\tilde{a}_{ij} \geq 0$ ist das Problem unbeschränkt,
  sonst führe eine Pivotoperation durch und iteriere wie im normalen Simplex-Verfahren.

Wir haben bisher gesehen, dass im primalen Simplex-Verfahren die Lösung in jeder Iteration primal zulässig ist. Dagegen ist die Lösung im dualen Simplex-Verfahren in jeder Iteration dual zulässig. Das duale Simplex-Verfahren bricht ab, sobald die Lösung dual optimal (und dementsprechend auch primal zulässig) ist.

Primal-duales Simplex-Verfahren

Wir haben im Laufe dieses Abschnittes nun schon mehrmals ausgenutzt, dass ein Paar von optimalen Lösungen x und π von zueinander dualen Programmen die Komplementaritätsbedingungen (siehe Satz 2.36 auf Seite 64) erfüllt. Von den drei möglichen Bedingungen

- primale Zulässigkeit
- duale Zulässigkeit
- Komplementaritätsbedingungen

erhält das primale Simplex-Verfahren in jedem Schritt die primale Zulässigkeit und das duale Simplex-Verfahren erhält die duale Zulässigkeit. In dem *primal-dualen Simplex-Verfahren*, das wir nun beschreiben wollen, ist die Lösung in jeder Iteration dual zulässig *und* sie erfüllt die Komplementaritätsbedingungen aus Satz 2.36.

Seien (P) und (D) wie folgt gegeben:

$$\begin{array}{lll} \text{(P) min } c^t x & & \text{(D) max } \quad b^t\pi \\ \quad \text{s.d. } Ax = b & & \quad \text{s.d. } \quad A^t\pi \leq c \\ \quad\quad x \geq 0 & & \quad\quad\quad \pi \gtrless 0 \end{array}$$

Sei ohne Beschränkung der Allgemeinheit $b \geq 0$. Die Komplementaritätsbedingungen der Programme (P) und (D) sind für eine Lösung x von (P) und eine Lösung π von (D) gegeben durch

$$x_j(c_j - (A_j)^t\pi) = 0 \quad \forall j = 1, \ldots, n. \tag{2.14}$$

Sei π dual zulässig. Wir betrachten die Indexmenge

$$\mathcal{I} := \{j : A_j^t\pi = c_j\}$$

der *zulässigen Spalten* von A, also der Spalten A_j, für die die Komplementaritätsbedingungen unabhängig von der Wahl von x auf jeden Fall erfüllt sind.

Lemma 2.42 *Sei π dual zulässig. π ist optimal für (D) genau dann, wenn*

$$H = \{x \in \mathbb{R}^{|\mathcal{I}|} : \sum_{j\in\mathcal{I}} x_j A_j = b, x_j \geq 0 \ \ \forall j \in \mathcal{I}\} \neq \emptyset$$

Beweis:

"$\Longrightarrow$" Sei π optimal. Nach Satz 2.34 existiert dann eine Optimallösung x^* für (P), für die gilt

$$\begin{array}{rcl} x^* & \geq & 0 \ \text{ und} \\ Ax^* & = & b. \end{array}$$

Nach Satz 2.36 erfüllt x^* außerdem Bedingung (2.14), d.h.

$$x_j^* = 0 \ \ \forall j \notin \mathcal{I}.$$

Wegen $b = Ax^* = \sum_{j=1}^n x_j^* A_j = \sum_{j\in\mathcal{I}} x_j^* A_j$ ist also $(x_i^*)_{i\in\mathcal{I}} \in H$ und $H \neq \emptyset$.

"$\Longleftarrow$" Sei nun $H \neq \emptyset$. Wähle $\bar{x} \in H$. Dann ist x mit

$$x_j := \begin{cases} \bar{x}_j & \text{falls } j \in \mathcal{I}; \\ 0 & \text{sonst.} \end{cases}$$

zulässig für (P) (denn $Ax = \sum_{j \in \mathcal{I}} x_j A_j = b$ und $x \geq 0$). Weiterhin gilt

$$x_j(c_j - A_j^t \pi) = 0 \quad \forall j = 1, \ldots, n,$$

also ist nach dem Satz von komplementären Schlupf (Satz 2.36) π optimal für (D) (und x ist optimal für (P)). QED

Im primal-dualen Algorithmus wird daher in jeder Iteration getestet, ob es ein $x \in \mathbb{R}^{|\mathcal{I}|}$ gibt mit

$$\sum_{j \in \mathcal{I}} x_j A_j = b, \quad x_j \geq 0 \quad \forall j \in \mathcal{I}.$$

Dazu löst man das folgende eingeschränkte Problem:

$$\begin{array}{lrcl} \text{(RP)} \qquad \min & w & = & \displaystyle\sum_{i=1}^{m} \hat{x}_i \\ \text{s.d.} & \displaystyle\sum_{j \in \mathcal{I}} x_j A_j + \hat{x} & = & b \\ & x_j & \geq & 0 \quad \text{für alle } j \in \mathcal{I} \\ & \hat{x} & \geq & 0 \end{array}$$

Das folgende Lemma ergibt sich direkt (ÜBUNG).

Lemma 2.43 *Sei π eine zulässige Lösung von (D) und $\mathcal{I}$ die Menge der zu π gehörenden Indizes von zulässigen Spalten von A. Dann ist π optimal für (D) genau dann wenn $w_{opt} = 0$ der optimale Zielfunktionswert von (RP) ist.*

Aus $w_{opt} = 0$ folgt also, dass π optimal ist, und wenn $w_{opt} > 0$ wissen wir, dass das nicht der Fall ist. Können wir daraus aber noch mehr Informationen ableiten?

Wir untersuchen den Fall $w_{opt} > 0$. Um π zu verbessern, betrachten wir das Duale (RD) des eingeschränkten linearen Programms (RP):

$$\begin{array}{lrcl} \text{(RD)} \qquad \max & v & = & b^t \alpha \\ \text{s.d.} & A_j^t \alpha & \leq & 0 \text{ für alle } j \in \mathcal{I} \\ & \alpha_i & \leq & 1 \quad \text{für alle } i = 1, \ldots, m \\ & \alpha_i & \gtrless & 0 \quad \text{für alle } i = 1, \ldots, m. \end{array}$$

Da $\alpha = 0$ eine zulässige Lösung ist, wissen wir dass (RD) zulässig ist. Weiterhin ist der Zielfunktionswert von (RD) beschränkt durch $\sum_{i=1}^{m} b_i$ denn laut Voraussetzung ist $b_i \geq 0$. (RD) ist also lösbar, d.h. es existiert ein optimales α^*, für das nach unserer Voraussetzung

$$b^t \alpha^* = w_{opt} > 0$$

gilt.

Man definiert nun für $\delta > 0$

$$\tilde{\pi} := \pi(\delta) := \pi + \delta \cdot \alpha^*.$$

Dann gilt

$$b^t \tilde{\pi} = b^t \pi + \underbrace{\delta}_{>0} \cdot \underbrace{b^t \alpha^*}_{>0} > b^t \pi,$$

das heißt, $\tilde{\pi}$ ist echt besser als π. Das hilft uns natürlich nur, wenn $\tilde{\pi}$ dual zulässig ist. Die duale Zulässigkeit von $\tilde{\pi}$ soll also im folgenden näher untersucht werden. Zunächst ist $\tilde{\pi}$ nach Definition dual zulässig, wenn $A_j^t \tilde{\pi} \leq c$. Das ist äquivalent zu der Bedingung

$$A_j^t \pi + \delta \cdot A_j^t \alpha^* \leq c_j \quad \text{für alle } j = 1, \ldots, n.$$

Wir betrachten diese Bedingung getrennt für $j \in \mathcal{I}$ und für $j \notin \mathcal{I}$.

$j \in \mathcal{I}$: Aufgrund der Zulässigkeit von α^* für (RD) wissen wir, dass

$$A_j^t \alpha^* \leq 0.$$

Daher gilt

$$A_j^t \pi + \delta \cdot A_j^t \alpha^* \leq A_j^t \pi \leq c_j \ \ \forall j \in \mathcal{I}$$

und die Bedingung ist für alle $j \in \mathcal{I}$ erfüllt.

$j \notin \mathcal{I}$: Hier unterscheiden wir wiederum zwei Fälle.

1. Fall: Falls $A_j^t \alpha^* \leq 0$ auch für alle $j \notin \mathcal{I}$ gilt, ist wie im eben untersuchten Fall $\pi(\delta)$ zulässig für alle $\delta > 0$. Das heißt aber,

   $$b^t \pi(\delta) \longrightarrow_{\delta \to \infty} \infty,$$

   und entsprechend ist (D) unbeschränkt und (P) daher unzulässig.

2. Fall: Nehmen wir nun an, dass $A_j^t \alpha^* > 0$ für mindestens ein $j \notin \mathcal{I}$. In diesem Fall setzt man

   $$\delta := \min\{\frac{c_j - A_j^t \pi}{A_j^t \alpha^*} : j \notin \mathcal{I}, A_j^t \alpha^* > 0\} > 0$$

   und wählt die dual zulässige Lösung $\pi(\delta)$ als Startwert für die nächste Iteration.

Dieses Vorgehen ist die Idee des primal-dualen Simplex-Verfahrens, das abschließend noch formal beschrieben werden soll.

Algorithmus: Primal-duales Simplex - Verfahren

Input: Dual zulässige Lösung π

Schritt 1: $\mathcal{I} := \{j : A_j^t \pi = c_j\}$

Schritt 2: Löse (RP) oder sein Duales (RD), sei v^* der Zielfunktionswert.

Schritt 3: Wenn $v^* = 0$ (**STOPP**): Wähle optimale Lösung $\bar{x}$ von (RP) und setze

$$x_j^* = \begin{cases} \bar{x}_j & \text{if } j \in \mathcal{I}; \\ 0 & \text{if } j \notin \mathcal{I}. \end{cases}$$

Output: x^* optimal.

Schritt 4: Bestimme eine optimale Lösung α^* von (RD).

Schritt 5: Wenn $A_j^t \alpha^* \leq 0 \quad \forall j \notin \mathcal{I}$ (**STOPP**): (P) unzulässig.
Sonst setze

$$\delta := \min\{\frac{c_j - A_j^t \pi}{A_j^t \alpha^*} : j \notin \mathcal{I}, A_j^t \alpha^* > 0\}$$

$$\pi := \pi + \delta \alpha^*$$

und gehe zu Schritt 1.

2.6 Karmarkars Innere-Punkte-Methode

2.6.1 Warum ein anderer Zugang?

Definition 2.44 *Ein Algorithmus hat die* **Komplexität** $O(g(n))$ *falls es eine Konstante C gibt, so dass die Gesamtzahl elementarer Rechnenoperatoren $(+, -, \cdot, <, \ldots)$ die von dem Algorithmus ausgeführt wird, nie größer als $O(g(n))$ ist. Dabei hängt g von der Größe des gegebenen Problems ab.*

Für ein lineares Programm kann seine Größe durch die Anzahl n seiner Variablen und durch die Anzahl m seiner Nebenbedingungen angegeben werden.

Man unterscheidet zwischen polynomialen Algorithmen mit Komplexität $O(n^p)$ für einen festen Wert p und Algorithmen, bei denen g exponentiell von der Größe der vorliegenden Eingabedaten abhängt. Der Unterschied zwischen polynomialen und exponentiellen Laufzeiten wird in der folgenden Tabelle (aus [GJ79]) verdeutlicht.

Laufzeit	$n = 10$	$n = 20$	$n = 30$	$n = 40$	$n = 50$	$n = 60$
n	$0,00001$ Sek	$0,00002$ Sek	$0,00003$ Sek	$0,00004$ Sek	$0,00005$ Sek	$0,00006$ Sek
n^2	$0,0001$ Sek	$0,0004$ Sek	$0,0009$ Sek	$0,0016$ Sek	$0,0025$ Sek	$0,0036$ Sek
n^3	$0,001$ Sek	$0,008$ Sek	$0,027$ Sek	$0,064$ Sek	$0,125$ Sek	$0,216$ Sek
n^5	$0,1$ Sek	$3,2$ Sek	$24,3$ Sek	$1,7$ Min	$5,2$ Min	$13,0$ Min
2^n	$0,001$ Sek	1 Sek	$17,9$ Min	$12,7$ Tage	$35,7$ Jahre	366 Jahrh.
3^n	$0,059$ Sek	58 Min	$6,5$ Jahre	3855 Jahrh.	2×10^8 Jahrh.	$1,3 \times 10^{13}$ Jahrh.

Wie die Tabelle zeigt, sind polynomiale Algorithmen also sehr wünschenswert!

Im folgenden werden wir ein — im Gegensatz zum Simplex-Verfahren — polynomielles Verfahren zum Lösen linearer Programme vorstellen. Vorher geben wir einen kurzen Überblick über die Entwicklung von Verfahren und Komplexitätsergebnissen zur linearen Optimierung.

1947: G. Dantzig entdeckt das Simplex-Verfahren.

ab 1960: Erste brauchbaren Implementationen sind verfügbar.

1971: Klee & Minty geben ein Beispiel an, bei dem das Simplex-Verfahren alle Basislösungen durchläuft.

1977: Borgwardt (Uni Kaiserslautern) zeigt, dass das Simplex-Verfahren "im Mittel" polynomial ist.

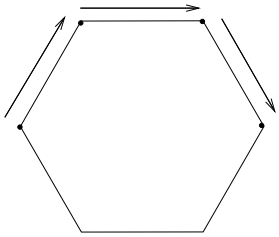
1979: Khachiyan entwickelt den "Ellipsoid Algorithmus" und zeigt, dass er polynomial ist. Damit liegt das erste polynomiale Verfahren zum Lösen linearer Programme vor. Leider ist es praktisch aber ineffizient.

1984: Karmarkar entwickelt mit seiner Inneren-Punkte-Methode ein effizientes, polynomiales Verfahren.

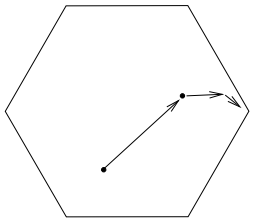
1984: Megiddo zeigt, dass lineare Programmierung bei fester Anzahl von Variablen in linearer Zeit ausführbar ist.

Im folgenden werden wir Karmarkars Innere-Punkte-Verfahren beschreiben. Seine Grundidee ist die folgende: Während das Simplex-Verfahren die Ecken des zulässigen Polyeders absucht und dabei immer am Rand von P bleibt, bewegt sich ein Iterationsschritt von Karmarkars Verfahren in die Richtung des steilsten Abstiegs im Inneren des Polyeders. Damit sind größere Schritte möglich; ein "Herumirren" außen wird also vermieden.

Simplex: laufe am Rand entlang.



Karmarkar: Bewege dich im Inneren von P
in die Richtung des steilsten Abstiegs.



In Kurzform lässt sich Karmarkars Verfahren wie folgt beschreiben:

Start: mit einem zulässigen $x \in int(P)$.

1. Transformation des zulässige Polyeders P mit einer projektiven Transformation, so dass
   - die aktuelle Lösung im "Zentrum" des tranformierten Polyeders liegt, und
   - das Problem möglichst wenig verändert wird.
2. Verbesserung der Lösung durch einen Schritt in Richtung des projizierten steilsten Abstiegs, der möglichst groß ist aber nicht bis zum Rand geht.
3. Rücktransformation des modifizierten Polyeders und der neuen Lösung. Gehe zu Schritt 1.

2.6.2 Karmarkars Form für LPs

Für das Verfahren brauchen wir ein LP in Karmarkar's Form.

Definition 2.45 *Ein LP ist in* **Karmarkar's Form**, *falls es durch*

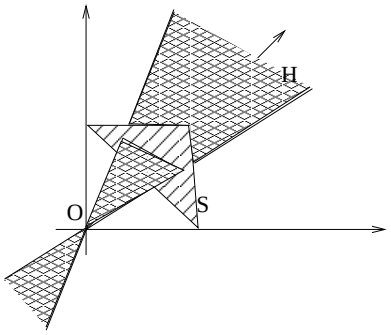
$$\begin{aligned} \text{(P) min } & c^t x \\ \text{s.d. : } & Ax = 0 \\ & e^t x = 1 \\ & x \geq 0 \end{aligned}$$

gegeben ist (für eine $m \times n$-Matrix A und den n-dimensionalen Vektor $e^t = (1, 1, \ldots, 1)$) und den folgenden Voraussetzungen genügt:

- *Alle Komponenten von A und c sind ganzzahlig.*
- *Ein zulässiger Punkt x^0 mit $x_i^0 > 0$ für alle $i = 1, \ldots, n$ ist bekannt.*
- *Der optimale. Zielfunktionswert ist Null.*

Die Bedingung $e^t x = 1$ nennt man auch **Normalisierungsbedingung**.

Die Normalisierungsbedingung für ein LP mit $n = 3$ Variablen ist gegeben durch $x_1 + x_2 + x_3 = 1$. Sie definiert einen *Simplex*.



$\longrightarrow$ Simplex

$\{x|Ax = 0\}$ ist Hyperebene H durch 0, die den Simplex S im zulässigen Gebiet schneidet.

Wir zeigen zuerst, wie man ein beliebiges lineares Programm in Karmarkars Form überführen kann.

Sei also ein lineares Programm gegeben. Wir kümmern ums zunächst um die seltsamste Forderung, nämlich, dass der Zielfunktionswert des LPs bekannt und gleich Null sein soll. Dazu verwenden wir die Dualitätstheorie. Ein gegebenes Problem

$$\begin{array}{lrcl} \text{(P)} & \min c^t x & & \\ & \text{s.d.} : Ax & = & b \\ & x & \geq & 0 \end{array}$$

formulieren wir um zu:

$$\begin{array}{lrcl} \text{(P')} & \min c^t x - b^t \pi^t & & \\ & \text{s.d.} : Ax & = & b \\ & x & \geq & 0 \\ & A^t \pi^t & \leq & c^t \\ & \pi & \gtrless & 0. \end{array}$$

Es gilt die folgende Aussage.

Lemma 2.46 *(P) ist lösbar genau dann wenn (P') lösbar ist. Weiter hat jede Optimallösung von (P') den Zielfunktionswert Null.*

Beweis: Die Aussage folgt direkt aus den Sätzen 2.34 und 2.36. QED

Im folgenden nehmen wir oBdA an, dass das umformulierte Programm (P') in Standardform vorliegt. Wir zeigen nun, wie man es in Karmarkars Form transformieren kann.

Sei also für ein (LP) der Form

$$\begin{aligned} \min \quad & c^t x & & \\ \text{s.d. } & Ax & = & \ b \\ & x & \geq & \ 0 \end{aligned}$$

der Zielfunktionswert bekannt und gleich Null. Wir nehmen an, dass die Koeffizienten des LPs durch Multiplikation der Zeilen mit ihren jeweiligen Hauptnennern ganzzahlig sind. Um es nun in Karmarkars Form zu überführen, benötigen wir die folgenden drei Schritte.

Schritt 1. Wir benutzen, dass

$$Ax = b \iff Ax - bx_{n+1} = 0 \text{ und } x_{n+1} = 1$$

um das gegebene LP in das folgende äquivalente lineare Programm umzuschreiben,

$$\begin{aligned} \min \quad & c^t x & & \\ \text{s.d. } & Ax - bx_{n+1} & = & \ 1 \\ & x_{n+1} & = & \ 1 \\ & x, x_{n+1} & \geq & \ 0. \end{aligned}$$

Schritt 2. Da wir voraussetzen, dass das lineare Programm lösbar ist, gibt es eine obere Schranke Q für die Summe der Variablen, die an der Optimallösung nichts ändert. Wir können die Nebenbedingung

$$e^t x \leq Q$$

also einfach mit in das lineare Programm aufnehmen und erhalten unter Einbeziehung der zusätzlichen Variablen x_{n+1} aus Schritt 1 zunächst

$$\begin{aligned} \min\ & c^t x \\ \text{s.d. } & Ax - bx_{n+1} = 1 \\ & x_{n+1} = 1 \\ & x \geq 0 \end{aligned} \qquad \Longleftrightarrow \qquad \begin{aligned} \min\ & c^t x \\ \text{s.d. } & Ax - bx_{n+1} = 1 \\ & x_{n+1} = 1 \\ & e^t x + x_{n+1} \leq Q + 1 \\ & x \geq 0. \end{aligned}$$

Einführen der Schlupfvariablen x_{n+2} ergibt

$$\begin{aligned} \text{(HP1)} \quad \min\ & c^t x \\ \text{s.d. } & Ax - bx_{n+1} &=& \ 0 \\ & e^t x + x_{n+1} + x_{n+2} &=& \ Q + 1 \\ & x_{n+1} &=& \ 1 \\ & x, x_{n+1}, x_{n+2} &\geq& \ 0. \end{aligned}$$

Die letzte Umformung in Schritt 2 führt uns zu

$$\begin{aligned} \text{(HP2)} \quad \min\ & c^t x \\ \text{s.d. } & Ax - bx_{n+1} &=& \ 0 \\ & e^t x - Qx_{n+1} + x_{n+2} &=& \ 0 \qquad (2.15) \\ & e^t x + x_{n+1} + x_{n+2} &=& \ Q + 1 \qquad (2.16) \\ & x, x_{n+1}, x_{n+2} &\geq& \ 0. \end{aligned}$$

Wir machen uns klar, dass (HP1) und (HP2) äqivalent sind:

"(HP1) $\Longrightarrow$ (HP2)": Eine zulässige Lösung von (HP1) ist wegen $x_{n+1} = 1$ auch für die neue Nebenbedingung (2.15) und damit für (HP2) zulässig.

"(HP2) $\Longrightarrow$ (HP1)": Sei eine zulässige Lösung von (HP2) gegeben. Die Differenz der Nebenbedingungen (2.16) - (2.15) ergibt $(Q+1)x_{n+1} = Q+1$. Daraus folgt $x_{n+1} = 1$. Das gewährleistet die Zulässigkeit für (HP1).

Schritt 3. Substituiere

$$y_j = \frac{x_j}{Q+1} \quad j = 1, \ldots, n+2$$

Das ergibt:

$$\text{(HP3)} \qquad \min c^t y$$

$$\begin{aligned} \text{s.d. } Ay - by_{n+1} &= 0 \\ e^t y - Qy_{n+1} + y_{n+2} &= 0 \\ e^t y + y_{n+1} + y_{n+2} &= 1 \\ y, y_{n+1}, y_{n+2} &\geq 0 \end{aligned}$$

Damit sieht unser lineares Programm endlich aus wie in Karmarkars Form. Von den drei Forderungen sind die letzten beiden bereits erfüllt. Es fehlt also nur noch ein zulässiger innerer Punkt. Dazu schreiben wir das lineare Programm noch einmal weiter um zu

$$\begin{aligned} \text{(HP4)} \qquad \min c^t y + My_{n+3} & \\ \text{s.d. } Ay - by_{n+1} - (Ae - b)y_{n+3} &= 0 \\ e^t y - Qy_{n+1} + y_{n+2} - (n + 1 - Q)y_{n+3} &= 0 \\ e^t y + y_{n+1} + y_{n+2} + y_{n+3} &= 1 \\ y_i &\geq 0 \text{ für alle } i = 1, \ldots, n+3 \end{aligned}$$

wobei wir $M > c_i Q$ für alle $i = 1, \ldots, n$ wählen.

Das folgende Lemma gibt uns nun den gesuchten zulässigen inneren Punkt.

Lemma 2.47 .

1. *$(HP4)$ hat eine Optimallösung mit $y^*_{n+3} = 0$ genau dann, wenn $(HP3)$ zulässig.*
2. *$(y_1, \ldots, y_{n+3})^t = (\frac{1}{n+3}, \ldots, \frac{1}{n+3})^t$ ist ein zulässiger innerer Punkt von $(HP4)$*

Beweis: ÜBUNG.

2.6.3 Karmarkars Algorithmus

Anmerkung: Dieser Abschnitt muss noch überarbeitet werden! Aktuell verweise ich auf das Buch [Ham95]; die Präsentation in diesem (und im letzten) Abschnitt ist von dem entsprechenden Kapitel dort übernommen und sollte sich daher leicht nachvollziehen lassen!

1. Projektive Transformation: Ziel: Transformiere das Problem so, dass der Punkt $\in int(P)$ in der Mitte liegt.

Sei x^k die Lösung des Algorithmus in Iteration k, mit $x^k_i > 0 \;\; \forall i = 1, \ldots, n$

$$T(x) := \frac{(D^k)^{-1}x}{e(D^k)^{-1}x} \qquad \forall x \in P$$

wobei $D^k = diag(x^k) = \begin{pmatrix} x_1^k & 0 & \cdots & \cdots & \cdots & 0 \\ 0 & x_2^k & 0 & \cdots & \cdots & 0 \\ : & : & \cdots & x_i^k & \cdots & : \\ 0 & \cdots & \cdots & \cdots & 0 & x_n^k \end{pmatrix}$

Es gilt:

$$T(x)_i = \frac{1}{\sum_{j=1}^n \frac{x_j}{x_j^k}} \cdot \frac{x_i}{x_i^k} \ , \qquad \sum_{i=1}^n T(x)_i = 1$$

$$T(x^k) = \begin{pmatrix} \frac{1}{n} \\ : \\ \frac{1}{n} \end{pmatrix}$$

Da alle $x \in T(P) = \{T(x) : x \in P\}$ Komponentensumme 1 haben, liegt $T(x^k)$ "in der Mitte".
Die zu T inverse Abbildung ist:

$$T^{-1}(y) = \frac{D^k y}{eD^k y} \quad \forall y = T(x),\ x \in P$$

Nachrechnen ergibt, dass $T^{-1}(T(x)) = x \ \ \forall x \in P$:

$$\begin{aligned} T^{-1}(T(x)) &= T^{-1}(\tfrac{(D^k)^{-1}x}{e(D^k)^{-1}x}) \\ &= D^k(\tfrac{(D^k)^{-1}x}{e(D^k)^{-1}x}) \cdot \tfrac{1}{eD^k(\frac{(D^k)^{-1}x}{e(D^k)^{-1}x})} \\ &= \tfrac{x}{e(D^k)^{-1}x} \cdot \tfrac{e(D^k)^{-1}x}{ex} \\ &= x, \end{aligned}$$

weil $ex = 1$ wegen der Noramliesierungsbedingung.
wir wenden T auf unser LP an, d.h. wir ersetzen $x = T^{-1}(y)$.

$$\begin{aligned} \min\ & c^t x \\ \text{s.d.} :\ & Ax = 0 \\ & ex = 1 \\ & x \geq 0 \end{aligned}$$

wird also zu:

$$\begin{aligned} \min \quad & c^t T^{-1}(y) = \tfrac{c^t D^k y}{eD^k y} && \leftarrow \textit{nicht lineare Zielfunktion} \\ \text{s.d.} : \quad & A\tfrac{D^k y}{eD^k y} = 0 \\ & e\tfrac{D^k y}{eD^k y} = 1 && \leftarrow \textit{redundant} \\ & \tfrac{D^k y}{eD^k y} \geq 0 \end{aligned}$$

Vereinfachen:
Sei $y = T(x),\ x \in P$. Dann gilt

$$eD^k y = eD^k T(x) = eD^k \frac{(D^k)^{-1}x}{e(D^k)^{-1}x} = \frac{ex}{e(D^k)^{-1}x} > 0$$

Da $T(P)$ äquivalent zu (P) ist, hat es Zielfunktionswert 0.
Wir können also einfach $\min\ c^t D^k y$ schreiben, und die Nenner $\neq 0$ weglassen.
Man erhält:

$$\begin{aligned} \min \quad & c^t D^k y \\ \text{s.d. :} \quad & AD^k y = 0 \\ & ey = 1 \\ & y \geq 0 \end{aligned}$$

$ey = 1$ ist redundant, weil

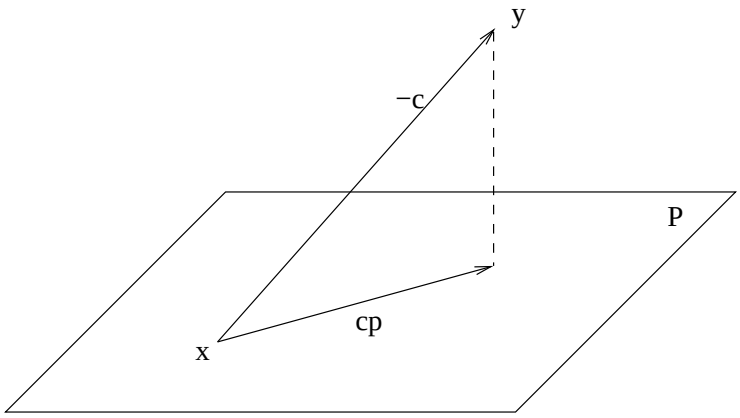
$$ey = eT(x) = e\frac{(D^k)^{-1}x}{e(D^k)^{-1}x} = 1 \quad \forall y = T(x)$$

2. Verbesserung des Zielfunktionswertes von y^k im transformieten Problem. Idee: Gehe in Richtung des steilsten Abstiegs.

Betrachte ein LP in Standardform:

$$\begin{aligned} \min\ & c^t x \\ \text{s.d. :}\ & Ax = 0 \\ & x \geq 0 \end{aligned}$$

Richtung des negativen Gradienten: $-c$
Aber $y = x - \delta \cdot c,\ \ \delta > 0$ erfüllt in allgemein nicht dass $Ay = Ax - \delta Ac = b$.



Daher bestimme orthogonale Projektion von y auf P. d.h. projiziere c auf

$$c_p \in \{c \in \mathbb{R}^n : Ax = 0\},\ y_p = x - \delta c_p$$

dann

$$Ay_p = Ax - \delta Ac_p = b - \delta \cdot 0 = b$$

Man bestimmt:

$$\begin{aligned} c_p^t &= c^t - A^t(AA^t)^{-1}Ac^t \\ &= (I - A^t(AA^t)^{-1}A)c^t \end{aligned}$$

Bemerkung: Für jede Matrix A mit vollem Zeilenrang existiert $(AA^t)^{-1}$.

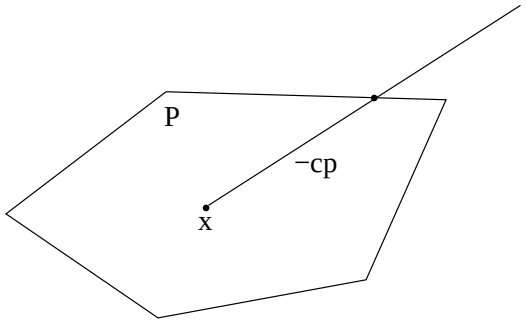
Lemma 2.48 *$Ac_p^t = 0$, also $A(x - \delta c_p^t) = b$*

Beweis:

$$\begin{aligned} Ac_p^t &= A(I - A^t(AA^t)^{-1}A)c^t \\ &= (A - AA^t(AA^t)^{-1}A)c^t \\ &= (A - A)c^t = 0 \end{aligned}$$

QED

Aber wie weit darf man gehen?



In unserem Fall ergibt sich:

$$\begin{array}{lll} \text{(TP)} & \min & c^t D^k y \\ & \text{s.d.} & AD^k y = 0 \\ & & ey = 1 \\ & & y \geq 0 \end{array}$$

d.h. $\bar{c} = cD^k$, $\bar{A} = \begin{pmatrix} AD^k \\ e \end{pmatrix}$, $\bar{b} = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$
und es gilt:

Lemma 2.49 .

$$r = \frac{1}{\sqrt{n(n-1)}} \ , \quad c_p = \bar{A}(I - \bar{A}^t(\bar{A}\bar{A}^t)^{-1}\bar{A})\bar{c}^t$$

$$y^{k+1} := y^k - \alpha \cdot r \cdot \frac{c_p}{||c_p||}$$

ist zulässig für (TP) *für alle* $0 \leq \alpha \leq 1$

Beweis:
Idee: Lege Kugel mit Radius r um $\left(\frac{1}{n}, \ldots, \frac{1}{n}\right)$ und zeige, dass diese komplett in $T(P)$ enthalten. QED

Wir können jetzt Karmarkars Algorithmus angeben.

Karmarkars Verfahren

Input: LP in Karmarkar's Form mit innerem zulässigen Punkt x_0, großer ganzen Zahl L, Schrittlänge α mit $0 < \alpha < 1$ und $r = \frac{1}{\sqrt{n(n-1)}}$

$$\begin{array}{ll} \min & c^t x \\ \text{s.d.} & Ax = 0 \\ & ex = 1 \\ & x \geq 0 \end{array}$$

Schritt 1: Falls $cx^k < 2^{-L}$ gehe zu Schritt 5.

Schritt 2: Projektive Transformation:
Transformiere das LP zu

$$\begin{array}{ll} \min & c^t D^k y \\ \text{s.d.} & AD^k y = 0 \\ & ey = 1 \\ & y \geq 0 \end{array}$$

mit $\bar{c} = cD^k$, $\bar{A} = \begin{pmatrix} AD^k \\ e \end{pmatrix}$, $\bar{b} = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$, $y^k = \left(\frac{1}{n}, \ldots, \frac{1}{n}\right)^t$

Schritt 3: Schritt in Richtung des steilsten Abstiegs:

$$\bar{c_p} = (I - \bar{A}^t(\bar{A}\bar{A}^t)^{-1}\bar{A})\bar{c}^t; \quad y^{k+1} := y^k - \alpha \cdot r \cdot \frac{\bar{c_p}}{||\bar{c_p}||}$$

Schritt 4: Inverse Transformation:

$$x^{k+1} = \frac{D^k y^{k+1}}{eD^k y^{k+1}}$$

$k := k + 1$ und gehe zu (1).

Schritt 5: Optimales Runden: Bestimme aus x^k eine optimale Basislösung x^* mit $cx^* < cx^k < 2^{-L}$.

Zu diskutieren bleiben noch die folgenden beiden Punkte:

- Korrektheit
- Schritt 5.

Korrektheit:

Satz 2.50 *Sei*

$$L = \lfloor 2 + log(1 + |c^{max}|) + log(|det^{max}|) \rfloor, \quad \alpha = \frac{n-1}{3n}$$

wobei $c^{max} = \max_{j=1,\ldots,n} c_j$, $det^{max} = \max\{|det(\bar{A}_B)| : \bar{A}_B \ \textit{Basismatrix von } \bar{A}\}$. *Dann finden die Schritte* $(1)-(4)$ *des Algorithmus von Karmarkar eine Lösung* x^k *mit Zielfunktionswert* $cx^k < 2^{-L}$, *mit einer Komplexität von* $O(n^{3.5}L)$.

Beweis: Wir zeigen im Beweis nur eine Komplexität $O(n^4L)$.
oBdA: $x^0 = (\frac{1}{n}, \ldots, \frac{1}{n})$ Startlösung.
Komplexität einer Iteration wird durch Bestimmung von $\bar{c_p}$ dominiert: $O(n^3)$
[Hier ist eine Verbesserung zu $O(n^{2.5})$ möglich, weil sich nur die Elemente von D^k ändern].
Zeige: Die Anzahl k der richtigen Iterationen um eine Lösung x^k mit $c^tx^k < 2^{-L}$ zu finden kann durch $O(nL)$ abgeschätzt werden.
Definiere

$$f(x) = \sum_{i=1}^{n} ln(\frac{c^tx}{x_i}) = n \cdot ln(c^tx) - \sum_{i=1}^{n} ln(x_i)$$

Wir wollen zeigen, dass

$$(*) \quad f(x^{j+1}) \leq f(x^j) - \delta \quad for\ \delta > 0 \ \ \forall j = 0, \ldots, k-1$$

Dann folgt:
$(*) \implies f(x^k) \leq f(x^0) - k \cdot \delta$
$\implies n \cdot ln(c^tx^k) - \sum_{i=1}^{n} ln(x_i^k) \leq n \cdot ln(c^tx^0) - \sum_{i=1}^{n} ln(\frac{1}{n}) - k \cdot \delta$

Wegen $max\{\sum_{i=1}^{n} ln(x_i) : ex = 1\} = \sum_{i=1}^{n} ln(\frac{1}{n})$ folgt:

$$\begin{aligned} n \cdot ln(c^tx^k) &\leq n \cdot ln(c^tx^0) + \sum_{i=1}^{n} ln(x_i^k) - \sum_{i=1}^{n} ln(\tfrac{1}{n}) - k \cdot \delta \\ &\leq n \cdot ln(c^tx^0) - k \cdot \delta \\ \implies \quad ln(c^tx^k) - ln(c^tx^0) &\leq -\tfrac{k \cdot \delta}{n} \\ \implies \quad ln(\tfrac{c^tx^k}{c^tx^0}) &\leq -\tfrac{k \cdot \delta}{n} \\ \implies \quad \tfrac{c^tx^k}{c^tx^0} &\leq e^{-\frac{k \cdot \delta}{n}} = 2^{\frac{-(\log e) \cdot k \cdot \delta}{n}} \end{aligned}$$

Das bedeutet, dass noch $k = \frac{2nL}{\delta \cdot \log(e)} = O(nL)$ Iterationen eine Lösung x^k vorliegt mit:

$$c^tx^k \leq c^tx^0 \cdot 2^{-2L} \leq 2^{-L},$$

denn $c^tx^0 \leq 2^{-L}$ (Übung).

Bleibt zu zeigen, dass $(*)$ gilt.
Sei T die zu x^j gehörende Abbildung:

$$\begin{aligned} f(x) &= f(T^{-1}(y)) = f(\frac{D^j y}{eD^j y}) \\ &= n \cdot \ln(c^t \frac{D^j y}{eD^j y}) - \sum_{i=1}^{n} \ln(\frac{x_i^j y_i}{eD^j y}) \\ &= n \cdot \ln(c^t D^j y) - n \cdot \ln(ec^t D^j y) - \sum_{i=1}^{n} \ln(x_i^j) - \sum_{i=1}^{n} \ln(y_i) + \sum_{i=1}^{n} \ln(eD^j y) \\ &= n \cdot \ln(c^t D^j y) - \sum_{i=1}^{n} \ln(y_i) - \sum_{i=1}^{n} \ln(x_i^j) \end{aligned}$$

Definiere

$$g(y) := n \cdot \ln(c^t D^j y) - \sum_{i=1}^{n} \ln(y_i)$$

Sei nun x^j die Lösung in Iteration j, x^{j+1}die in Iteration $j+1$. Dann definiere:

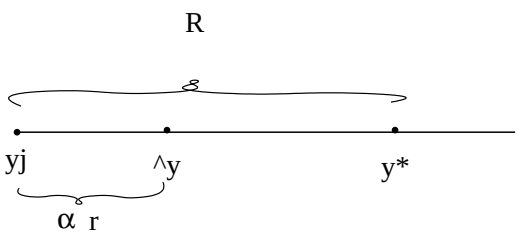
$$\begin{aligned} y^j &= T(x^j) \\ \hat{y} &= y^j - \alpha \cdot r \cdot \frac{c_p}{||c_p||} \\ r &= \frac{1}{\sqrt{n(n-1)}} \end{aligned}$$

und es gilt: $T^{-1}(\hat{y}) = x^{j+1}$ bzw. $T(x^{j+1}) = \hat{y}$.
Weiterhin:

$$f(x^j) - f(x^{j+1}) = g(y^j) - g(\hat{y}) + const - const$$

d.h. es reicht zu zeigen, dass

$$g(\hat{y}) \leq g(y^j) - \delta \ , \qquad y^j = (\frac{1}{n}, \ldots, \frac{1}{n})$$



$$R = \sqrt{\frac{n-1}{n}} \ , \qquad und \ \ n \cdot r = \frac{\sqrt{n}}{\sqrt{n-1}} > \frac{\sqrt{n-1}}{\sqrt{n}} = R$$
$$\Longrightarrow \ \frac{r}{R} > \frac{1}{n}$$

d.h. die Verbesserung beträgt:

$$
\begin{aligned}
& c^t(y^j - \hat{y}) = c^t(y^j - y^*) \cdot \frac{\alpha r}{R} = c^t y^j \cdot \frac{\alpha r}{R} \ , \ da \ c^t y^* = 0 \\
\Longrightarrow \quad & c^t y^j (1 - \frac{\alpha r}{R} = c^t \hat{y} \\
\Longrightarrow \quad & c^t \hat{y} = c^t y^j (1 - \frac{\alpha r}{R}) < c^t y^j (1 - \frac{\alpha}{n}) \leq c^t y^j \cdot e^{-\frac{\alpha}{n}} \\
& Weil \ \ \frac{r}{R} > \frac{1}{n} \ und \ \ln(1-x) \leq -x \ \Rightarrow \ e^{-x} \geq 1 - x \\
\Longrightarrow \quad & n \cdot \ln(c^t D^j \hat{y}) < n \cdot \ln(c^t D^j y^j \cdot c^{-\frac{\alpha}{n}}) \\
& \qquad = n \cdot \ln(c^t D^j y^j) - \alpha
\end{aligned}
$$

Der zweite Teil von $g(y)$ hebt das nicht auf. ... $\longrightarrow$ $g(\hat{y}) - g(y^j) \leq -\delta$ QED

Als letztes diskutieren wir Schritt (5) des Verfahrens, das optimale Runden. (Purifikationsschema)

Satz 2.51 *Sei x^k mit $c^t x^k < 2^{-L}$ die nach k Iterationen gefundene Lösung von Karmarkar's Algorithmus. Dann kann eine zulässige Basislösung x^* mit $c^t x^* < c^t x^k$ in polynomialer Zeit bestimmt werden.*

Beweis: (wie Hauptsatz linearer Optimierung.)
Sei x^k mit $c^t x^k < 2^{-L}$ die zulässige Lösung nach k Iterationen.
Sei $\hat{A} = \begin{pmatrix} A \\ e \end{pmatrix}$, $\hat{b} = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$, also

$$
\begin{aligned}
\min \ & c^t x \\
\text{s.d.} \ & \hat{A} x = \hat{b} \\
& x \geq 0
\end{aligned}
$$

oBdA:

$$
\begin{aligned}
x_i^k &> 0 \qquad \text{für } i = 1, \ldots, l \\
x_i^k &= 0 \qquad \text{für } i = l+1, \ldots, n
\end{aligned}
$$

$$
\hat{b} \neq 0 \quad \Longrightarrow \quad 1 \leq l \leq n
$$

Sind die zu den l positiven Komponenten von x^k gehörenden Spalten von $\hat{A}$ unabhängig, so ist x^k zulässige Basislösung.
Sonst existiert $\alpha = (\alpha_1, \ldots, \alpha_l)^t \neq 0$ mit $\hat{A}_L \cdot \alpha = \hat{A}_L \cdot (\delta\alpha) = 0$ ($\hat{A}_L = (\hat{A}_1, \ldots, \hat{A}_L)$)
Sei für $\delta \in \mathbb{R}$

$$
x^k(\delta)_i = \begin{cases} x_i^k + \delta\alpha_i & \text{i=1,...,l;} \\ 0 & \text{sonst.} \end{cases}
$$

Für $\delta_1 = \max\{\frac{-x_i^k}{\alpha_i} : \alpha_i > 0\} < 0$
oder $\delta_2 = \max\{\frac{-x_i^k}{\alpha_i} : \alpha_i < 0\} > 0$ gilt $\delta_j \in \mathbb{R}$.
Für die entsprechende Lösung $x^k(\delta_j)$ gilt:

- $x^k(\delta_i)$ hat höchstens $l-1$ positive Komponenten
- $\hat{A}(x^k(\delta_j)) = \hat{b}$
- $x^k(\delta_i) \geq 0$
- $c^t x^k(\delta_j) = c^t x^k + \delta_j(\alpha_1 c_1 + \ldots + \alpha_l c_l)$
- Ist $\alpha_1 c_1 + \ldots + \alpha_l c_l > 0$
  so wähle $\delta = \delta_1 < 0$ (d.h. $\delta_1 \in \mathbb{R}$ oder Problem unbeschränkt).
- für $\alpha_1 c_1 + \ldots + \alpha_l c_l < 0$
  wähle $\delta = \delta_2 > 0$ (d.h. $\delta_2 \in \mathbb{R}$ oder Problem unbeschränkt).
  Es gilt: $c^t x^k(\delta_j) \leq c^t x^k$ und $x^k(\delta_j)$ hat höchstens noch $n-1$ positiven Komponenten.
  $\Longrightarrow$ nach $l-1$ Iterationen erhält man zulässige Basislösung x^* mit $c^t x^* < c^t x^k$.

Der folgende Satz sagt, dass diese Lösung sogar optimal ist! QED

Satz 2.52 *Die durch das Purifikationsschema in Satz 2.51 gefundene zulässige Basislösung x^* mit $c^t x^* < 2^{-L}$ ist optimal.*

Beweis: .
Sei B eine beliebige zulässige Basis mit $x_B = A_B^{-1} b$, $x_N = 0$ mit Zielfunktionswert

$$\begin{aligned} c^t x = c_B x_B &= c_B A_B^{-1} b \\ &= c_B \frac{Adj(A_B)}{\det(A_B)} \cdot b \end{aligned}$$

wobei $A^* = Adj(A)$ die Adjungierte bezeichnet mit $a_{ij}^* = (-1)^{i+j} \cdot \det A_{ji}$
und A_{ji} ist Matrix A ohne Zeile j und ohne Spalte i.
z.B. :

$$Adj(A) = Adj \begin{pmatrix} 1 & 2 & 3 \\ 1 & 0 & 0 \\ 0 & 1 & 1 \end{pmatrix} = \begin{pmatrix} 0 & 1 & 0 \\ -1 & 1 & 3 \\ 1 & -1 & -2 \end{pmatrix}$$

$\Longrightarrow \quad A^{-1} = \frac{1}{det(A)} \cdot Adj(A) = Adj(A)$ *weil* $\det(A) = 1$

Alle Komponenten von c_B, $Adj(A_B)$, b ganzzahlig

$$\Longrightarrow \quad c^t x = \frac{z}{\det(A_B)} \ , \quad z \in \mathbb{Z}$$

Wegen $|\det(A_B)| \leq |\det^{max}| < 2^L$ (Übung) folgt

$$\begin{aligned} c^t x = \frac{z}{\det(A_B)} &\geq z \cdot 2^{-L} \\ &\geq 2^{-L} \qquad\qquad weil \;\; z \geq 1 \end{aligned}$$

d.h. jede nicht-optimale Basislösung x erfüllt $c^t x \geq 2^{-L}$. Karmarkar's Verfahren findet aber eine Basislösung x^* mit

$$c^t x^* \leq c^t x^k < 2^{-L}$$

$\Longrightarrow$ x^* optimal mit Zielfunktionswert Null. QED

Kapitel 3

Ganzzahlige Programmierung & Netzwerkflussprobleme

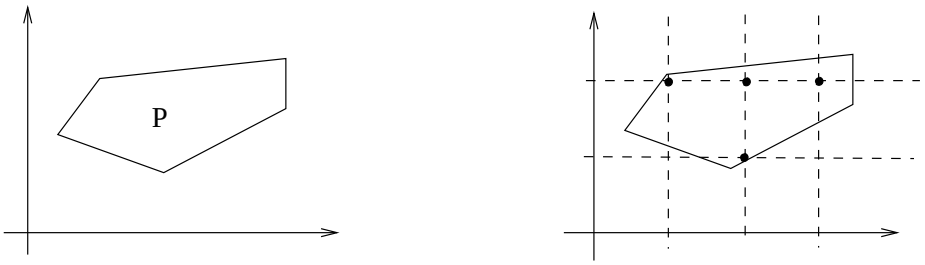
3.1 Ganzzahlige Programmierung

Definition 3.1 *Ein* **ganzzahliges lineares Programm** *(IP) ist ein lineares Programm mit der zusätzlichen Bedingung, dass alle Variablen ganzzahlig sein sollen. Wir können es z.B. in Standardform betrachten, also als*

$$\begin{array}{lrcl} \text{(IP)} & \min c^t x & & \\ & \text{s.d. } Ax & = & b \\ & x & \in & \mathbb{Z}^n_+. \end{array}$$

Zum Zeichnen bietet sich die $\leq$-Form an:

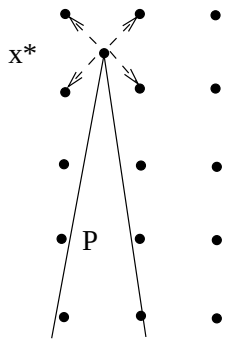
$$\begin{array}{rr} \text{(LP) } \min c^t x & \text{(IP) } \min c^t x \\ \text{s.d. } Ax \leq b & \text{s.d. } Ax \leq b \\ x \in \mathbb{R}^n & x \in \mathbb{Z}^n \end{array}$$



In der ersten Abbildung (links) sieht man, dass der zulässige Bereich P für (LP) ein Polyeder ist, während der zulässige Bereich $X = P \cap \mathbb{Z}^n$ in der rechten Abbildung eine (diskrete) Menge von Gitterpunkten ist.

Eine erste Idee, die man haben könnte, ist die folgende: Anstatt des ganzzahligen Programms (IP) löst man das dazu passende lineare Programm (LP) und rundet die gefundene Lösung. Dabei treten allerdings folgende Schwierigkeiten auf.

1. Es kann sein, dass die gerundete Lösung unzulässig ist, selbst dann, wenn man beliebiges Auf- und Abrunden erlaubt. Das wird an folgendem Beispiel $\in \mathbb{R}^2$ verdeutlicht:



2. Selbst wenn man durch Auf- oder Abrunden eine zulässige Lösung erhält, kann der optimale Zielfunktionswert des ganzzahligen Programms (IP) von dem gerundeten Zielfunktionswert des linearen Programms (LP) beliebig weit weg liegen.

3. Insbesondere wenn man es mit booleschen Variablen $x \in \{0,1\}^n$ zu tun hat, hilft Runden nicht weiter.

Man muss also andere Ideen entwickeln, um ganzzahlige Programme zu lösen - derzeit ein aktives Forschungsfeld. Es ist leider nicht nicht einmal bekannt, ob es überhaupt Lösungsverfahren mit polynomialer Laufzeit für beliebige ganzzahlige Programme gibt. (Die Mehrzahl der Mathematiker vermutet, dass das nicht der Fall ist.)

Um ganzzahlige Programme zu bearbeiten, versucht man, den Zielfunktionswert nach oben und nach unten zu beschränken. Dazu sind *Relaxationen* hilfreich.

Definition 3.2 *Sei ein mathematisches Programm*

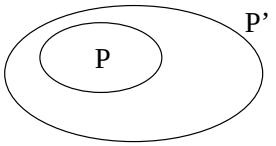
$$(P) \quad \min\{f(x) : x \in P\}$$

gegeben. Das Programm

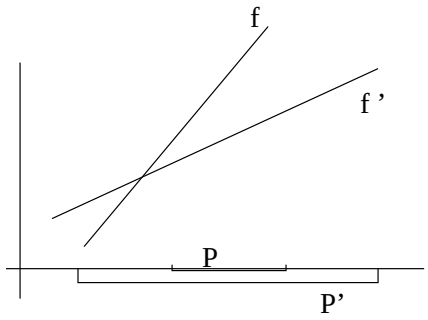
$$(P') \quad \min\{f'(x) : x \in P'\}$$

heißt **Relaxation** *von (P), falls*

$$\begin{aligned} P &\subseteq P' \text{ und} \\ f'(x) &\leq f(x) \text{ für alle } x \in P \end{aligned}$$



und



Eine häufig verwendete Relaxation ist die *LP-Relaxation*, bei der man die Zielfunktion eines linearen Programms nicht verändert, und ausschließlich die Ganzzahligkeitsbedingung relaxiert. Formal ist die LP-Relaxation von

$$\begin{aligned} \min\ & c^t x \\ \text{s.d.} :\ & Ax = b \\ \text{s.d.} :\ & x \geq 0 \\ & x \in \mathbb{Z}^n \end{aligned}$$

gegeben durch

$$\begin{aligned} \min\ & c^t x \\ \text{s.d.} :\ & Ax = b \\ \text{s.d.} :\ & x \geq 0 \\ & x \in \mathbb{R}^n \end{aligned}$$

Die folgende Aussage gilt für jede Relaxation.

Lemma 3.3 *Sei (P') Relaxation von (P) mit*

$$\begin{aligned} (\mathrm{P}') \quad & \min\{f'(x) : x \in P'\} \\ (\mathrm{P}) \quad & \min\{f(x) : x \in P\}. \end{aligned}$$

Sei x' optimal für (P') und x^* optimal für (P). Dann gilt:

1. *$f'(x') \leq f(x^*)$.*
2. *Falls $x' \in P$ und $f'(x') = f(x')$ ist x' optimal für (P).*

Beweis:

ad 1. $f'(x') = \min_{x \in P'} f'(x) \leq \min_{x \in P} f'(x) \leq \min_{x \in P} f(x) = f(x^*)$.

ad 2. Sei also $x' \in P$ optimal für die Relaxation (P'). Dann gilt wegen der Voraussetzung, dem ersten Teil des Lemmas und weil $x' \in P$:

$$f(x') = f'(x') \leq f(x^*) \leq f(x'),$$

es folgt also $f(x') = \min_{x \in P} f(x)$ und x' ist optimal für (P). QED

Speziell für die (LP)-Relaxation bedeutet das: Sei x^R eine Optimallösung der Relaxation $\min\{c^t x : Ax \leq b, x \in \mathbb{R}^n\}$.

1. Dann ist $c^t x^R$ eine untere Schranke für den Zielfunktionswert des ganzzahligen Programms (IP) $\min\{c^t x : Ax \leq b, x \in \mathbb{Z}^n\}$.
2. Falls $x^R \in \mathbb{Z}^n$ ist x^R sogar optimal für das ganzzahlige Programm (IP).

Man beachte, dass die Relaxation bei Maximierungsproblemen nicht zu unteren, sondern zu oberen Schranken führt. Im folgenden Beispiel betrachten wir ein Maximierungsproblem und seine Relaxation.

Beispiel 3.1

$$\begin{array}{rr} \text{(IP) max } 4x_1 - x_2 & \text{(LP) max } 4x_1 - x_2 \\ \text{s.d. } 7x_1 - 2x_2 \leq 14 & \text{s.d. } 7x_1 - 2x_2 \leq 14 \\ x_2 \leq 3 & x_2 \leq 3 \\ 2x_1 - 2x_2 \leq 3 & 2x_1 - 2x_2 \leq 3 \\ x \in \mathbb{Z}^2_+ & x \in \mathbb{R}^2_+ \end{array}$$

Wir erraten eine zulässige Lösung $x = (2, 1)^t$ und wollen ihre Qualität abschätzen. Der Zielfunktionswert von x ergibt sich zu $c^t x = 7$, also gilt für den Wert der (unbekannten) optimalen Lösung x^*:

$$c^t x^* \geq 7.$$

Dann lösen wir die LP-Relaxation und erhalten als optimale Lösung $x^R = (\frac{20}{7}, 3)$ mit Zielfunktionswert $z^R = \frac{59}{7}$. Das ergibt eine obere Schranke für das ganzzahlige Programm, nämlich

$$c^t x^* \leq \frac{59}{7}.$$

Man kann die obere Schranke sogar noch verschärfen, weil wir wissen, dass der Zielfunktionswert einer ganzzahligen Lösung (bei ganzzahligen Kosten c) auch wieder ganzzahlig sein muss. Damit können wir also abrunden und erhalten

$$c^t x^* \leq \lfloor \frac{59}{7} \rfloor = 8.$$

Zusammen mit der unteren Schranke ergibt sich also $c^t x^* \in \{7, 8\}$. Damit ist der Zielfunktionswert schon "fast" bekannt.

Im folgenden möchten wir gerne Fälle identifizieren, in denen der zweite Fall aus Lemma 3.3 eintritt, in denen also $x^R \in \mathbb{Z}$ gilt. Dabei hilft uns einmal mehr der Hauptsatz der linearen Optimierung, der besagt, dass es für lösbare lineare Programme immer eine optimale Basislösung gibt. Um $x^R \in P \cap \mathbb{Z}^n$ zu erreichen, fordern wir also, dass alle Basislösungen ganzzahlig sind. Um lineare Programme mit ausschließlich ganzzahligen Basislösungen zu erkennen, benötigen wir den folgenden Begriff.

Definition 3.4 *Sei A eine $m \times n$-Matrix. A ist* **total unimodular** (**TU**)*, wenn für jede (quadratische) Submatrix A' von A gilt:*

$$\det(A') \in \{0, 1, -1\}.$$

Dabei definieren wir als *Submatrizen von A* alle Matrizen, die durch Streichen von Zeilen und Spalten von A entstehen.

Beispiel 3.2 *. Betrachte folgende Matrix*

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

Die quadratischen Submatrizen von A sind

- *jedes Element von A (als 1×1-Matrix)*
- *die neun 2×2-Matrizen, sie sich durch Streichen jeweils einer Zeile und einer Spalte ergeben, z.B.*

  $$A_{11} = \begin{vmatrix} 1 & 0 \\ -1 & -1 \end{vmatrix} = -1$$

  wenn man Zeile 1 und Spalte 1 streicht, oder

  $$A_{23} = \begin{vmatrix} -1 & 0 \\ 1 & -1 \end{vmatrix} = -1,$$

  die durch das Streichen von Zeile 2 und Spalte 3 entsteht.
- *die Matrix A selbst.*

Beispiele für TU-Matrizen beziehungsweise nicht TU-Matrizen sind die folgenden:

1. Die Matrix A aus obigem Beispiel ist TU.

2. Für die Matrix $B = \begin{pmatrix} 1 & 1 & 0 \\ -1 & 1 & 1 \end{pmatrix}$ gilt, dass $\det(B') = \det \begin{pmatrix} 1 & 1 \\ -1 & 1 \end{pmatrix} = 2$, also ist B ist nicht TU.

3. Die Matrix $C = \begin{pmatrix} 2 & 1 \\ 1 & 1 \end{pmatrix}$ ist auch nicht TU, weil $a_{11} = 2$.

Es wird sofort klar, dass jede total unimodulare Matrix A $a_{ij} \in \{0, 1, -1\}$ für alle Koeffizienten a_{ij} erfüllt, und dass aus totaler Unimodularität von A $\det(A) \in \{0, 1, -1\}$ folgt. Die folgenden Aussagen sind nützlich.

Lemma 3.5 .

(i) A ist TU $\Longleftrightarrow$ A^t ist TU.

(ii) A ist TU $\Longleftrightarrow$ $(A|I)$ ist TU.

Beweis:

ad (i) Die Aussage folgt direkt wegen $|\det(A)| = |\det(A^t)|$.

ad (ii) "$\Longleftarrow$": Diese Richtung gilt, weil jede Submatrix B von A auch Submatrix von $(A|I)$ ist.

"$\Longrightarrow$": Sei A TU. Betrachte eine Submatrix B von $(A|I)$. Ist B sogar Submatrix von A gilt $\det(B) \in \{0, 1, -1\}$. Ansonsten enthält B mindestens eine Spalte aus I, z.B. e_k. Wir entwickeln nach dieser Spalte und erhalten $|\det(B)| = |\det(B_{kk})|$, wobei B_{kk} die Teilmatrix von B ohne Spalte k und ohne Zeile k bezeichnet. Ist B_{kk} Submatrix von A, folgt wiederum $\det(B_{kk}) \in \{0, 1, -1\}$, also $\det(B) \in \{0, 1, -1\}$. Anderenfalls enthält B_{kk} eine weitere Spalte von I und wir führen das Verfahren fort, bis wir

$$|\det(B)| = |\det(\tilde{B})|$$

mit einer Submatrix $\tilde{B}$ von A erhalten, also $\det(B) \in \{0, 1, -1\}$.

QED

Der folgende Satz bestätigt nun, dass sich ganzzahlige Programme mit total unimodularer Koeffizientenmatrix leicht (d.h. durch lineare Programmierung) lösen lassen.

Satz 3.6 *Sei A eine ganzzahlige $m \times n$ - Matrix. Dann sind folgende Aussagen äquivalent:*

- *A ist TU.*

- *Für alle $b \in \mathbb{Z}^m$ gilt: Jede Basislösung von $P := \{x \in \mathbb{R}^n : Ax \leq b, x \geq 0\}$ ist ganzzahlig.*

Beweis:

"$\Longrightarrow$" Sei A total unimodular. Wir formulieren den zulässigen Bereich P als ein lineares Programm in Standardform,

$$P := \{x \in \mathbb{R}^{n+m} : (A|I)x = b, x \geq 0\}$$

und erhalten die nach Lemma 3.5 total unimodulare Koeffizientenmatrix $A' = (A|I)$. Sei nun x^R eine Basislösung. Wir möchten zeigen, dass $x^R \in \mathbb{Z}^{n+m}$. Der Nichtbasis-Anteil $x_N = 0$ ist auf jeden Fall ganzzahlig. Für den Basisanteil x_B gilt nach der Cramerschen Regel

$$x_B = A_B^{-1} b = \frac{\mathrm{Adj}(A_B)}{\det(A_B)} \cdot b$$

Die Elemente h_{ij} der Adjungierten $\mathrm{Adj}(A_B) = (h_{ij})$ zu A_B entsprechen der Determinante der Matrix $(A_B)_{ij}$, die durch Streichen der $i$ten Zeile und der $j$ten Spalte aus A_B entsteht, $h_{ij} = \det((A_B)_{ij})$. Weil diese Matrizen $(A_B)_{ij}$ Submatrizen von $(A|I)$ sind, gilt $h_{ij} \in \{0, 1, -1\}$. Weiterhin ist $|\det(A_B)| = 1$, weil A_B Basis einer total unimodularen Matrix ist. Damit ist x_B ganzzahlig, also auch x^R.

"$\Longleftarrow$" Sei A_1 eine beliebige nicht-singuläre $k \times k$ Submatrix von A. Wir wollen $|\det(A_1)| = 1$ zeigen. Wir vervollständigen die Spalten in A_1 und fügen noch $m - k$ Spalten aus I dazu, so dass wir eine $m \times m$-Matrix

$$\tilde{A} = \begin{pmatrix} A_1 & 0 \\ A_2 & I_{m-k} \end{pmatrix}$$

erhalten, die

$$|\det(\tilde{A})| = |\det(A_1)| \neq 0. \tag{3.1}$$

erfüllt. Also bilden die Spalten der Matrix $\tilde{A}$ für jedes $b \in \mathbb{R}^m$ eine Basis des zulässigen Bereiches

$$P := \{x \in \mathbb{R}^{n+m} : (A|I)x = b, x \geq 0\}$$

Sei B die entsprechende Indexmenge. Weil es gewohnter ist schreiben wir ab jetzt $A_B = \tilde{A}$.

Definiere $b = A_B z + e_i$ für einen ganzzahligen Vektor $z \in \mathbb{Z}^m$ und den i.ten Einheitsvektor e_i. Dann gilt $A_B^{-1} b = z + (A_B^{-1})_i$ wobei $q_i := (A_B^{-1})_i$

die i.te Spalte von A_B^{-1} ist. Wähle z so, dass $z + q_i \geq 0$. Dann ist $z + q_i$ der Basisanteil der zu B gehörenden Basislösung. Nach der Voraussetzung wissen wir, dass $z + q_i \in \mathbb{Z}_+^n$ gilt. Weil $z \in \mathbb{Z}^m$ muss also auch $q_i \in \mathbb{Z}^m$ sein. Das können wir für alle $i = 1, \ldots, m$ machen und erhalten so, dass A_B^{-1} eine ganzzahlige $m \times m$ Matrix ist.

Also ist $\det(A_B^{-1}) \in \mathbb{Z}$. Nach Voraussetzung ist auch $\det(A_B) \in \mathbb{Z}$. Schließlich gilt

$$|\det(A_B)||\det(A_B^{-1})| = \det(I) = 1,$$

woraus folgt, dass $|det\tilde{A}| = |\det(A_B)| = 1$, also wegen (3.1) $|\det(A_1)| = 1$.

QED

Sei

$$\begin{array}{ll} IP(A,b,c) & \min\{c^t x : Ax \leq b, x \in \mathbb{Z}_+^n\} \\ LP(A,b,c) & \min\{c^t x : Ax \leq b, x \in \mathbb{R}_+^n\} \end{array}$$

Angewendet auf ganzzahlige Programme $IP(A,b,c)$ sagt der eben bewiesene Satz also:

- Ist A TU, dann gibt es für alle $c \in \mathbb{R}^n, b \in \mathbb{Z}^m$ eine Lösung von $LP(A,b,c)$, die auch Lösung des ganzzahligen Programmes $IP(A,b,c)$ ist. Das heißt, man kann eine Lösung des ganzzahligen Programmes finden, indem man seine LP-Relaxation löst (solange die gefundene Lösung der LP-Relaxation eine Basislösung ist!)
- Ist A nicht TU, dann gibt es $c \in \mathbb{R}^n, b \in \mathbb{Z}^m$, so dass $LP(A,b,c)$ keine ganzzahlige optimale Lösung hat.

Es soll noch erwähnt werden, dass die Äquivalenz in Satz 3.6 für den Fall von (IP) $\min\{c^t x : \ Ax = b, \ x \in \mathbb{Z}_+^n\}$ nicht gilt. Zwar gilt noch, dass auch in diesem Fall aus der totalen Unimodularität von A die Ganzzahligkeit der Lösung der LP-Relaxation folgt, die Rückrichtung des Satzes ist dann aber im allgemeinen falsch.

Beispiel 3.3 *Ein einfaches Beispiel hierfür liefern ganzzahlige Matrizen, die nicht total unimodular sind, aber ganzzahlige Inverse haben wie z.B.*

$$A = \begin{pmatrix} 1 & -2 \\ 0 & 1 \end{pmatrix}.$$

Diese ist nicht TU, aber für jede rechte Seite $b \in \mathbb{Z}^2$ ist die Menge der zulässigen Lösungen $\{x : Ax = b, x \geq 0\}$ entweder leer oder enthält genau einen (ganzzahligen) Punkt, nämlich $x = (b_1 + 2b_2, b_2)^t$ (der entsprechend auch für jede Wahl der Zielfunktion optimal ist).

Das folgende Lemma gibt ein Kriterium, an dem man TU-Matrizen erkennen kann.

Lemma 3.7 *Eine $m \times n$-Matrix A ist TU, falls die folgenden drei Bedingungen gleichzeitig gelten:*

1. $a_{ij} \in \{1, -1, 0\}$ *für alle* $i \in \{1, \ldots, m\}, j \in \{1, \ldots, n\}$
2. *Jede Spalte von A enthält höchstens zwei von Null verschiedene Elemente.*
3. *Es gibt eine Partition $M_1 \cup M_2 = \{1, \ldots, m\}$ der Zeilen von A, so dass jede Spalte j mit genau zwei Nicht-Nullelementen*

$$\sum_{i \in M_1} a_{ij} - \sum_{i \in M_2} a_{ij} = 0$$

erfüllt.

Beispiel 3.4 *Die folgende Matrix ist nach dem Kriterium TU; die Partition ist durch eine Leerzeile angedeutet.*

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

Auch diese Matrix ist TU, als Partition kann man $M_1 = \{1, 2, 3, 4, 5\}$ und $M_2 = \emptyset$ wählen.

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

Beweis: Wir beweisen das Lemma durch Widerspruch und nehmen an, dass A die drei Bedingungen erfüllt, aber nicht TU ist. Dann wählen wir B als eine kleinste quadratische Submatrix von A, die der totalen Unimodularität widerspricht, also mit $\det(B) \notin \{0, 1, -1\}$.

B kann keine Spalte mit nur einem Nicht-Null-Element enthalten, somit gäbe es eine kleinere quadratische Submatrix, deren Determinante auch nicht in $\{0, 1 - 1\}$ ist.

Bezeichne I die Indizes der Zeilen von B und $b^j, j \in I$ ihre Zeilen. Wir addieren nun alle Zeilen von B mit Indizes in $M_1 \cap I$ und subtrahieren die Zeilen von B mit Indizes aus $M_2 \cap I$,

$$b := \sum_{j \in M_1 \cap I} b^j - \sum_{j \in M_2 \cap I} b^j.$$

Weil jede Spalte von B zwei (also alle!) Nicht-Null-Elemente der Spalten von A enthält ergibt diese Addition einen Zeilenvektor $b = 0$. Damit sind die Zeilen von B aber linear abhängig — ein Widerspruch zu $\det(B) \neq 0$. QED

Das wesentliche Ergebnis dieses Abschnittes ist, dass man ganzzahlige Programme mit TU-Matrizen effizient durch lineare Programmierung lösen kann. Für beliebige ganzzahlige Programme muss man Methoden der *ganzzahligen Programmierung* anwenden, wie sie z.B. in [NW88] ausführlich beschrieben sind. Zwei solcher Verfahren sollen abschließend kurz skizziert werden.

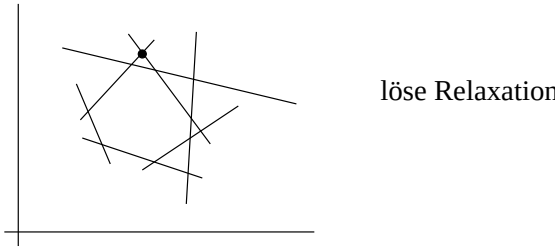
- Ein häufig angewendetes Verfahren ist das *Brauch & Bound*-Verfahren. Es basiert auf folgender Idee:

  Man löst zunächst die LP-Relaxation und erhält eine Optimallösung x^R. (Im Fall, dass die Relaxation nicht lösbar ist, ist auch das ganzzahlige Programm nicht lösbar.) Ist $x^R \in \mathbb{Z}^n$ haben wir die Optimallösung des ganzzahligen Programms bereits gefunden. Anderenfalls wähle eine nichtganzzahlige Komponente x_i^R von x^R und verzweige ("*to branch*") das ganzzahlige Programm in zwei neue, die durch Anfügen jeweils einer der folgenden Bedingungen entstehen:

  a) $x_i \geq \lfloor x_i^R \rfloor + 1$
  b) $x_i \leq \lfloor x_i^R \rfloor$

  Man untersucht dann die beiden entstehenden Probleme weiter. Dabei muss man aber nicht alle so entstehenden Äste bis zum Ende durchgehen, sondern versucht, durch gute Schranken ("*bounds*") ganze Teilbereiche von der Untersuchung auszuschließen. Die Einsetzbarkeit eines solchen Branch & Bound Verfahrens hängt wesentlich von der Qualität der Schranken ab.

- Schnittebenen-Verfahren (*cutting planes*)

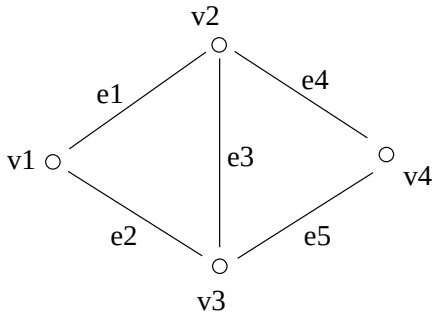


Auch hier löst man zunächst die LP-Relaxation des ganzzahligen Programms. Ist sie lösbar, aber die Lösung x^R nicht ganzzahlig fügt man eine neue Ungleichung zu der LP-Relaxation hinzu. Diese wird so gewählt, dass sie x^R vom zulässigen Bereich ausschließt, aber keine zulässige *ganzzahlige* Lösung abschneidet. Man wiederholt das Verfahren bis man eine ganzzahlige Lösung findet. (Wählt man die Schnittebenen geschickt, kann man in jedem Schritt eine weitere ganzzahlige Ecke erreichen, so dass sich das Polyeder immer mehr der Ganzzahligkeit nähert.)

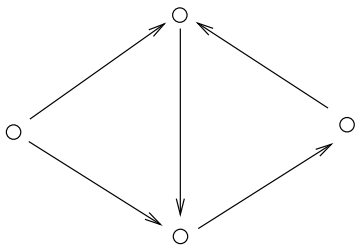
3.2 Netzwerkflussprobleme

Zunächst brauchen wir einige grundlegende Definitionen aus der Graphentheorie.

Definition 3.8 *Ein Tupel $G = (V, E)$ mit nichtleerer, endlicher* **Knotenmenge** *V und* **Kantenmenge** *$E \subseteq \{\{i, j\} : i, j \in V\}$ heißt* **ungerichteter Graph**.



Ein Tupel $G = (V, E)$ mit nichtleerer, endlicher **Knotenmenge** *V und* **gerichteter Kantenmenge** *$E \subseteq V \times V = \{(i, j) : i, j \in V\}$ heißt* **gerichteter Graph** *oder* **Digraph**.



Dabei zeichnen wir ungerichtete Kanten $\{i, j\}$ als Striche und gerichtete Kanten (i, j) als Pfeile.

Wir geben zunächst eine Möglichkeit an, Graphen oder Digraphen in Form von Matrizen zu speichern.

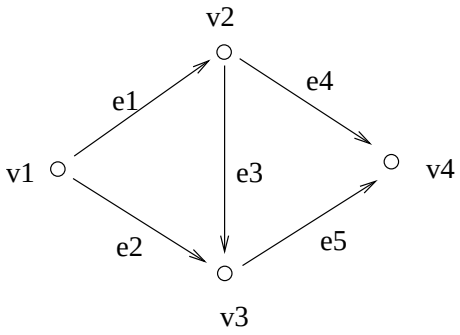
Definition 3.9 *Die* **Knoten-Kanten-Inzidenzmatrix** *$A = A(G) = (a_{ie})_{i \in V, e \in E}$ eines schlingenfreien Digraphen $G = (V, E)$ ist gegeben durch:*

$$a_{ie} = \begin{cases} 1 & \text{falls } e = (i, k) \in E \text{ für ein } k \in V \\ -1 & \text{falls } e = (k, i) \in E \text{ für ein } k \in V \\ 0 & \text{sonst.} \end{cases}$$

Für einen (ungerichteten) Graphen $G = (V, E)$ definiert man

$$a_{ie} = \begin{cases} 1 & \text{falls } e = \{i, k\} \in E \text{ für ein } k \in V \\ 0 & \text{sonst.} \end{cases}$$

Beispiel 3.5 $m = 5, \quad n = 4$



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

Bemerkungen:

- Für jeden Digraphen G gilt $det(A(G)) = 0$. Das liegt daran, dass für alle $e \in E$ gilt: $\sum_{i=1}^{n} a_{ie} = 0$, also sind die Zeilen der Knoten-Kanten-Inzidenzmatrix linear abhängig.
- Die Knoten-Kanten-Inzidenzmatrix A ist für jeden Digraphen total unimodular.

Definition 3.10 *Sei $G = (V, E)$ ein Graph oder Digraph, und sei $c_e \in \mathbb{R}$ die Länge der Kante $e \in E$. Eine endliche Knotenfolge $P = (v_{i_1}, \ldots, v_{i_l})$ heißt* **Weg** *von v_{i_1} nach v_{i_l}, falls*

- $e_{i_p} = \{v_{i_p}, v_{i_{p+1}}\} \in E$ *für Graphen, beziehungsweise*
- $e_{i_p} = (v_{i_p}, v_{i_{p+1}}) \in E$ *oder* $e_{i_p} = (v_{i_p+1}, v_{i_p}) \in E$ *für Digraphen.*

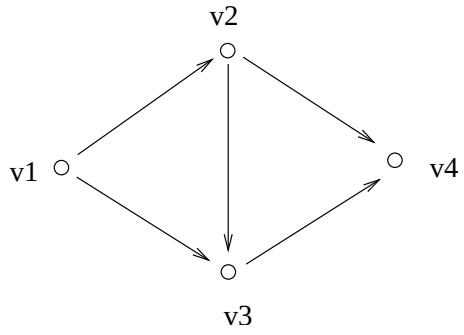
Die **Länge** *des Weges P ist*

$$l(P) = \sum_{e \in P} c_e.$$

Ein Weg P in einem Digraphen heißt **gerichtet**, *wenn er keine Kante $e_{i_p} = (v_{i_{p+1}}, v_{i_p})$ enthält. Ein Weg mit $v_{i_l} = v_{i_1}$ heißt* **Kreis**, *ein gerichteter Weg mit $v_{i_l} = v_{i_1}$ heißt* **Dikreis** *Ein (Di-)Graph heißt* **zusammenhängend** *wenn je zwei Knoten in G durch einen Weg verbunden sind.*

Obwohl Wege eigentlich als Kantenmengen definiert sind, gibt man sie auch oft durch eine geordnete Folge von Knoten an.

Beispiel 3.6 .



In diesem Graphen ist $P_1 = (v_1, v_2, v_4)$ *ein gerichteter Weg mit zwei Kanten,* $P_2 = (v_1, v_2, v_1, v_3)$ *ein Weg (aber kein gerichteter Weg) mit drei Kanten, während* $P_3 = (v_2, v_4, v_1, v_3)$ *kein Weg ist. Ein Kreis ist z.B.* $P = (v_1, v_2, v_4, v_3, v_1)$*. Der Graph ist zusammenhängend.*

Wir wollen nun **Netzwerkflussprobleme** einführen. Dazu sei ein zusammenhängender Digraph $G = (V, E)$ mit n Knoten und m Kanten gegeben. Wir benötigen weiter

- einen Vektor $b \in \mathbb{R}^n$, der jedem Knoten i einen Bedarf ($b_i < 0$) oder einen Vorrat ($b_i > 0$) zuordnet, oder der angibt, dass in i weder Bedarf noch Vorrat besteht ($b_i = 0$),
- obere und untere Schranken $l_{ij} \leq u_{ij}$ für alle Kanten $(i, j) \in E$,
- und Kosten c_{ij} für alle Kanten $(i, j) \in E$.

Das Netzwerk wird beschrieben durch $N = (V, E, b, l, u, c)$.

Wir partitionieren die Knotenmenge in folgende Teilmengen:

- $S = \{i \in V : b_i > 0\}$ sei die Menge der Vorratsknoten,
- $D = \{i \in V : b_i < 0\}$ sei die Menge der Bedarfsknoten, und
- $T = \{i \in V : b_i = 0\}$ sei die Menge der Durchflussknoten.

Das Netzwerkfkussproblem besteht nun darin, den überschüssigen Vorrat aus den Vorratsknoten mit möglichst geringen Transportkosten und unter Einhaltung der Kapazitätsbedingungen an die Bedarfsknoten zu verteilen.

Definition 3.11 *Eine Abbildung $x : E \to \mathbb{R}$ heißt* **Fluss**, *falls die folgenden Flusserhaltungsbedingungen*

$$\sum_{j:(i,j)\in E} x_{ij} - \sum_{j:(j,i)\in E} x_{ji} = b_i$$

in allen Knoten $i \in V$ erfüllt sind. Ein Fluss x heißt **zulässig** *falls*

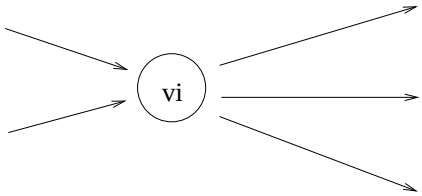
$$l_{ij} \leq x_{ij} \leq u_{ij} \text{ für alle } (i,j) \in E.$$

Die Kosten des Flusses x sind gegeben durch

$$c(x) := c^t x = \sum_{(i,j)\in E} c_{ij} x_{ij}.$$

Die Flusserhaltungsgleichung gewährleistet, dass in jedem Knoten gilt:

einfließender Fluss + Vorrat = ausfließender Fluss.



Als lineares Programm kann man ein **Netzwerkflussproblem** (NFP) mit den gegebenen Daten $N = (V, E, b, l, u, c)$ wie folgt formulieren:

$$\begin{array}{rc} \min & \sum_{(i,j)\in E} c_{ij} x_{ij} \\ \text{u.d.N.} & Ax = b \\ & l \leq x \leq u \end{array} \quad ,$$

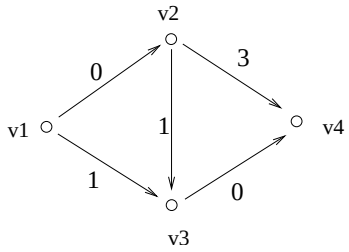
wobei A die Knoten-Kanten-Inzidenzmatrix von G ist.

Das Programm hat also n Nebenbedingungen (für jeden Knoten eine) und m Variablen (für jede Kante den Fluss x_{ij}). Setzt man die Definition der Knoten-Kanten-Inzidenzmatrix ein, so beschreibt die $i$te Nebenbedingung $A^i x = b_i$ gerade die Flusserhaltungsgleichung im Knoten i, nämlich

$$A^i x = \sum_{e\in E} a_{ie} x_e = \sum_{j:(i,j)\in E} x_{ij} - \sum_{j:(j,i)\in E} x_{ji} = b_i.$$

Die folgenden bekannten Probleme lassen sich als Netzwerkflussprobleme formulieren und lösen.

Beispiel 3.7 (Kürzeste Wege Problem) *Gegeben sei ein Digraph $G = (V, E)$ mit Entfernungen c_{ij} für jede Kante $(i, j) \in E$ und zwei ausgezeichneten Knoten s und t. Die Aufgabe besteht darin, einen möglichst kurzen Weg von s nach t zu finden, also einen Weg P von v_1 nach v_4 mit minimaler Länge $l(P)$.*



Kürzeste Wege Probleme können folgendermaßen als Flussprobleme modelliert werden: Definiere ein Netzwerk $N = (V, E, b, l, u, c)$ durch die gegebenen Daten V, E, c und setze

$$b_s := 1,\ b_t = -1, \quad \textit{und } b_i = 0 \textit{ für alle } i \notin \{s, t\}.$$

Es soll also genau eine Flusseinheit von s nach t geschickt werden. Weiterhin sollen die Kapazitätsbedingungen z.B. durch $l_e = 0, u_e = 1$ für alle $e \in E$ vernachlässigt werden. Der sich ergebende Fluss entspricht dann wegen der Flusserhaltung einem Weg, und ein kostenminimaler Fluss entspricht einem kürzesten Weg.

In unserem konkreten Beispiel erhält man das folgende lineare Programm zur Bestimmung eines kürzesten Weges in G,

$$\begin{array}{lll}
\min & 0 \cdot x_{12} + 1 \cdot x_{13} + 1 \cdot x_{23} + 3 \cdot x_{24} + 0 \cdot x_{34} & \\
\text{s.d.} & x_{12} + x_{13} & = 1 \\
& -x_{12} + x_{23} + x_{24} & = 0 \\
& -x_{13} - x_{23} + x_{34} & = 0 \\
& -x_{24} - x_{34} & = -1 \\
& 0 \le x_{ij} \le 1 & \forall (i, j) \in E
\end{array}$$

Beispiel 3.8 (Maximales Flussproblem) *Gegeben sei erneut ein Digraph $G = (V, E)$, zwei ausgezeichnete Knoten s und t sowie Kapazitätsbeschränkungen u_e für alle $e \in E$. Die Aufgabe besteht darin, so viel Fluss wie möglich von s nach t zu senden.*

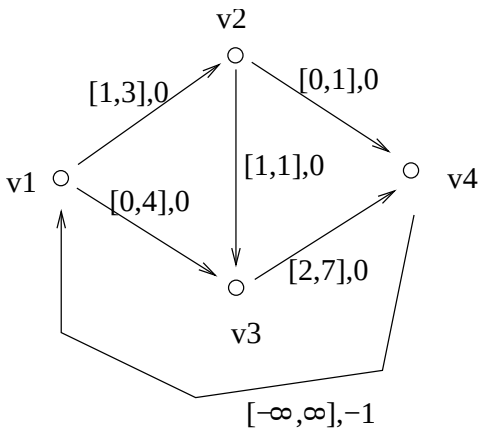
Als lineares Programm lässt sich das maximale Flussproblem durch die Knoten-Kanten-Inzidenzmatrix A von G wie folgt angeben:

$$\begin{aligned} &\max v \\ &\text{s.d. } Ax = b \\ &0 \leq x \leq u, \end{aligned}$$

wobei der Vektor b auf der rechten Seite definiert ist als

$$b_i = \begin{cases} v & , \textit{falls } i = s \\ -v & , \textit{falls } i = t \\ 0 & , \textit{falls } i \neq s, t. \end{cases}$$

Auch ein maximales Flussproblem kann man als Netzwerkflussproblem modellieren. Wir definieren Kosten $c_{ij} = 0$ für alle $(i, j) \in E$, und setzen die unteren Schranken $l_{ij} = 0$ auf Null, und ebenso den Bedarf $b_i = 0$ für alle $i \in V$.



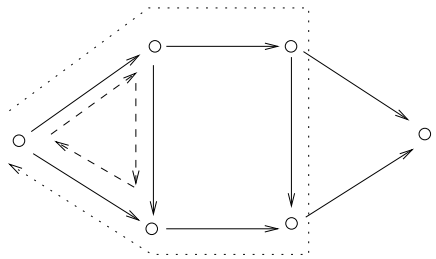
Schließlich erweitert man G um eine neue Kante (t, s), für die man definiert:

$$\begin{aligned} l_{ts} &= -\infty \\ u_{ts} &= \infty \\ c_{ts} &= -1 \end{aligned}$$

Das Ziel besteht nun also darin, einen möglichst hohen Fluss durch die neue Kante $(t, s) \in E$ zu schicken. Das geht, so lange dieser Fluss auch wieder von s nach t zurückfließen kann, ohne die oberen Kapazitäten zu verletzten und löst damit das maximale Flussproblem.

Nach diesen Beispielen beschäftigen wir uns nun mit der Lösung von Netzwerkflussproblemen. Die Idee unseres Ansatzes besteht darin, die Zielfunktion zu verbessern ohne die Flusserhaltungsgleichungen und die Kapazitätsbeschränkungen zu verletzen.

Definition 3.12 *Ein Fluss x heißt* **Zirkulation**, *wenn $Ax = 0$. Eine Zirkulation heißt* **zulässig**, *wenn $l \leq x \leq u$ gilt. Ein Netzwerk mit $b = 0$ heißt* **Zirkulationsnetzwerk**.



Satz 3.13 *Jedes (NFP) ist äquivalent zu einem Zirkulationsproblem, d.h. zu einem (NFP) mit $b = 0$.*

Beweis: Wir konstruieren aus den gegebenen Daten ein Zirkulationsproblem, in dem wir eine Quelle s und eine Senke t einfügen, die wir mit Vorrats- beziehungsweise Bedarfsknoten i verbinden. Als Rückflusskante nehmen wir die Kante (t, s) dazu. Formal sieht das wie folgt aus:

Definiere $\hat{G} = (\hat{V}, \hat{E})$ mit $\hat{V} = V \cup \{s, t\}$ und

$$\hat{E} = E \cup \{(s, i) : i \in S\} \cup \{(i, t) : i \in D\} \cup \{(t, s)\}.$$

Setze $v = \sum_{i \in S} b_i = \sum_{i \in D} -b_i$ und definiere für alle $i \in V$: $\hat{b}_i = 0$. Für alle $(i, j) \in \hat{E}$ setzen wir weiter

$$\hat{u}_{ij} = \begin{cases} u_{ij} & \text{falls } (i,j) \in E \\ b_j & \text{falls } (i,j) = (s,j) \\ -b_i & \text{falls } (i,j) = (i,t) \\ v & \text{falls } (i,j) = (t,s) \end{cases}$$

$$\hat{l}_{ij} = \begin{cases} l_{ij} & \text{falls } (i,j) \in E \\ b_j & \text{falls } (i,j) = (s,j) \\ -b_i & \text{falls } (i,j) = (i,t) \\ v & \text{falls } (i,j) = (t,s) \end{cases}$$

$$\hat{c}_{ij} = \begin{cases} c_{ij} & \text{falls } (i,j) \in E \\ 0 & \text{sonst} \end{cases}$$

Dann ist x ein Fluss in $N = (V, E, b, l, u, c)$ genau dann, falls $\hat{x}$ mit

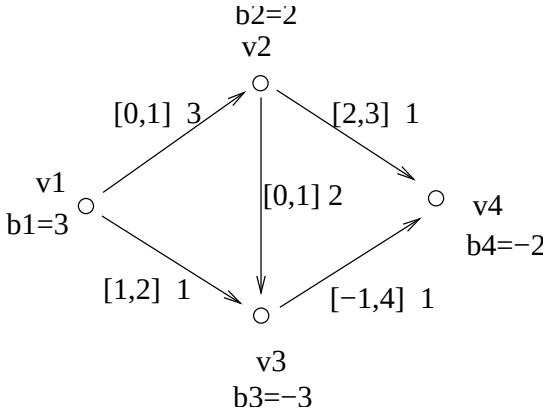
$$\hat{x}_{ij} = \begin{cases} x_{ij} & \text{falls } (i,j) \in E \\ b_j & \text{falls } (i,j) = (s,j) \\ -b_i & \text{falls } (i,j) = (i,t) \\ v & \text{falls } (i,j) = (s,t) \end{cases}$$

ein Fluss in $\hat{N} = (\hat{V}, \hat{E}, \hat{b}, \hat{l}, \hat{u}, \hat{c})$ ist. Weiterhin haben x und $\hat{x}$ die gleichen Kosten, nämlich

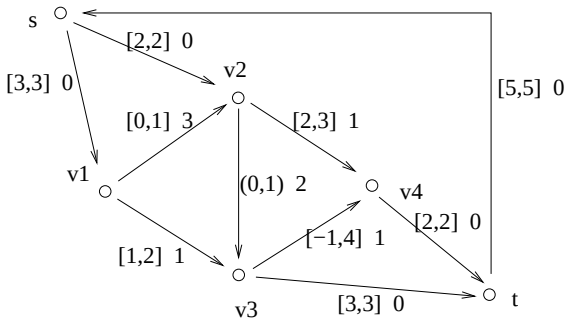
$$c^t x = \hat{c}^t \hat{x}.$$

QED

Beispiel 3.9 *Betrachte das folgende* (NFP)



Das äquivalente Zirkulationsproblem sieht folgendermaßen aus:



Im folgenden zeigen wir, dass sich jede Zirkulation in kleinere Kreise, so genannte Dikreisflüsse, zerlegen lässt. Diese Aussage brauchen wir zum Beweis des Hauptergebnisses später in diesem Abschnitt.

Definition 3.14 *Sei C ein Dikreis und $\mathcal{E} \in \mathbb{R}$ mit $0 \leq \mathcal{E} \leq u_{ij}$ für alle $(i,j) \in C$. Der zu C gehörende* **Dikreisfluss** *ist definiert durch $\mathcal{E}(C)$ mit*

$$\mathcal{E}(C)_{ij} := \begin{cases} \mathcal{E} & \text{falls } (i,j) \in C \\ 0 & \text{sonst} \end{cases}$$

Aufgrund der Wahl von $\mathcal{E}$ ist ein Dikreisfluss immer eine zulässige Zirkulation.

Satz 3.15 (Dekompositionssatz für Zirkulationen) *Sei x eine zulässige Zirkulation, dann existieren Dikreise $C_1, \ldots, C_k$ und $\mathcal{E}_1, \ldots, \mathcal{E}_k \in \mathbb{R}$, so dass gilt:*

1. *$\mathcal{E}_i(C_i)$ ist ein Dikreisfluss für alle $i = 1, \ldots, k$*

2. *$x = \sum_{i=1}^{k} \mathcal{E}_i(C_i)$*

3. *$k \leq m$*

Beweis:
Falls $x = 0$ die zulässige Zirkulation ist, sind alle Behauptungen erfüllt. Nehmen wir also an, dass $x \neq 0$. Dann wähle $(i_1, i_2) \in E$ mit $x_{i_1,i_2} > 0$. Wegen $b_{i_2} = 0$, existiert $(i_2, i_3) > 0$. Durch Iteration erhält man einen Diweg $(i_1, \ldots, i_p, \ldots, i_q)$, so dass $x_{i_j,i_{j+1}} > 0$ für alle $j = 1, \ldots, q-1$ und $i_p = i_q$ für $q > p \geq 1$, wobei wir annehmen, dass q minimal mit dieser Eigenschaft ist. Setze $C = (i_p, i_q)$ und

$$\mathcal{E} := \min\{x_{ij} : (i,j) \in C\} > 0.$$

Der Fluss $x := x - \mathcal{E}(c)$ ist dann wieder eine Zirkulation. Ist $x = 0$ endet das Verfahren, ist $x \neq 0$ wird der Vorgang iteriert. Da in jeder Iteration mindestens eine Kante den Wert $x_{ij} = 0$ erhält, bricht das Verfahren nach maximal m Iterationen ab. Daher findet es maximal m Dikreise, also $k \leq m$. QED

Wir wollen Zirkulationsprobleme wie folgt ausnutzen: Ändert man einen gegebenen Fluss entlang einer Zirkulation (unter Beachtung der Kapazitätsbeschränkung), dann bleiben die Flusserhaltungsgleichungen unverletzt, und der neu entstehende Fluss ist also zulässig. Das Ziel ist es also, einen gegebenen Fluss entlang einer Zirkulation zu verbessern. Um Verbesserungen leichter erkennen zu können, führen wir noch Inkrementnetzwerke ein.

Definition 3.16 *Sei N ein Zirkulationsnetzwerk und sei x eine zulässige Zirkulation. Das* **Inkrementnetzwerk** *N_x zu N und zu x ist gegeben durch folgendes Zirkulations-Netzwerk*

$$N_x := (V, E_x, 0, l_x, u_x, c_x).$$

Die Menge der Kanten im Inkrementnetzwerk ist gegeben durch $E_x = E_x^+ \dot{\cup} E_x^-$, wobei

$$E_x^+ := \{(i,j) \in E : x_{ij} < u_{ij}\} \tag{3.2}$$
$$E_x^- := \{(j,i) : (i,j) \in E \text{ und } x_{ij} > l_{ij}\}. \tag{3.3}$$

Weiterhin definiert man als untere Schranken $l_x(i,j) := 0$ für alle $(i,j) \in E$, und obere Schranken

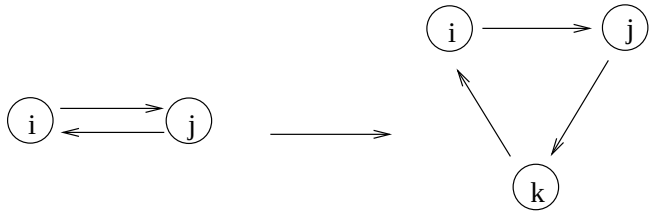
$$u_x(i,j) := \begin{cases} u_{ij} - x_{ij} & \text{falls } (i,j) \in E_x^+ \\ x_{ji} - l_{ji} & \text{falls } (i,j) \in E_x^-. \end{cases}$$

Schließlich braucht man noch Kosten

$$c_x(i,j) := \begin{cases} c_{ij} & \text{falls } (i,j) \in E_x^+ \\ -c_{ji} & \text{falls } (i,j) \in E_x^-. \end{cases}$$

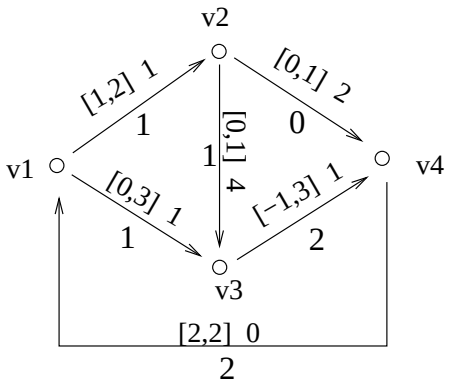
Zulässige Zirkulationen in N_x nennt man **Inkrementzirkulationen**.

Wegen $E_x = E_x^+ \cup E_x^-$ muss ein antisymmetrischer Digraph (d.h. $(i,j) \in E \Rightarrow (j,i) \notin E$) vorausgesetzt werden, sonst ist der Inkrementgraph kein einfacher Graph. Ist G nicht antisymmetrisch, so kann durch folgende folgende Transformation ein antisymmetrischer Graph erzwungen werden.

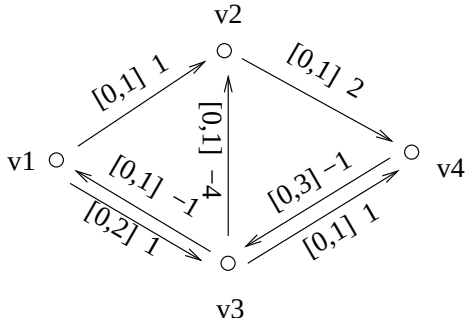


nicht antisymmetrisch antisymmetrisch

Beispiel 3.10 *Das Beispiel zeigt ein Netzwerk N mit einem Fluss x_{ij} (dargestellt jeweils unter der Kante), und den Inkrementgraphen N_x bezüglich des dargestellten Flusses x.*



Inkrementnetzwerk



Sei ξ eine Inkrementzirkulationen. Wir möchten diese Zirkulation gerne zu anderen Zirkulationen oder Flüssen addieren. Leider unterscheiden sich im allgemeinen aber die Kantenmengen E_{x_1} und E_{x_2} der Inkrementgraphen bezüglich verschiedener Flüsse x_1 und x_2. Um diese Schwierigkeit zu umgehen, erweitern wir eine Inkrementzirkulation ξ' in N_x auf alle Kanten

$$\{(i,j) \in V \times V : (i,j) \in E \text{ oder } (j,i) \in E\},$$

indem wir für die *erweiterte Zirkulation* ξ definieren:

$$\xi_{ij} = \begin{cases} \xi_{ij} & \text{falls } (i,j) \in E_x \\ 0 & \text{sonst} \end{cases}$$

Wir partitionieren die erweiterte Zirkulation $\xi = (\xi^+, \xi^-)$ dann in zwei Teile: ξ^+ definiert die Werte für die Kanten aus E und ξ^- für die Kanten aus $\{(j,i) : (i,j) \in E\}$. Wir erhalten die folgende Aussage.

Lemma 3.17 *$\xi = (\xi^+, \xi^-)$ ist eine* **Inkrementzirkulation** *bezüglich x, falls gilt:*

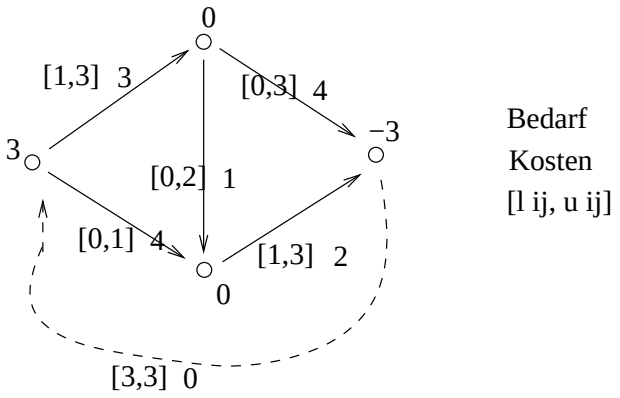
- $A \cdot \xi^+ - A \cdot \xi^- = 0$
- $l_x(i,j) \leq \xi \leq u_x(i,j) \quad \forall (i,j) \in E_x$
- $\xi_{ij} = 0 \qquad \forall (i,j) \notin E_x$

Beweis: Sei A die Knoten-Kanten-Inzidenzmatrix von G. Da die erweiterte Inkrementzirkulation alle Kanten

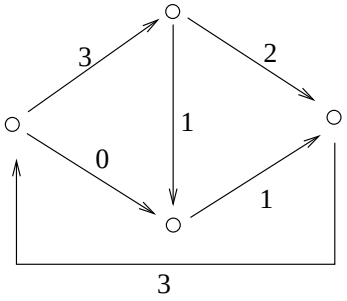
$$\bar{E} = \{(i,j) \in V \times V : (i,j) \in E \text{ oder } (j,i) \in E\},$$

betrifft, ist die zugehörige Inzidenzmatrix von $G = (V, \bar{E})$ gegeben durch $(A, -A)$. Das ergibt die Behauptung. QED

Beispiel 3.11 *Wir illustrieren die Begriffe an einem Beispiel. Sei folgendes zu einem Zirkulationsproblem transformierte Netzwerkflussproblem gegeben:*

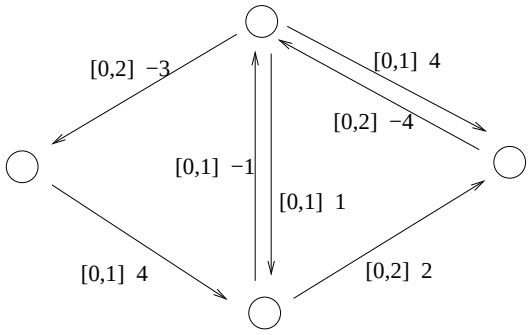


Eine zulässige Zirkulation x ist zum Beispiel:



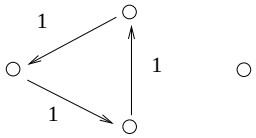
Kosten: 20

Im folgenden ist das Inkrementnetzwerk $N_x = (V, E_x, u_x, l_x, c_x)$ bezüglich der Zirkulation x angegeben:



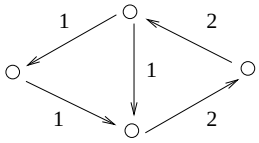
Beispielhaft betrachten wir drei Inkrementzirkulationen in N_x mit unterschiedlichen Kosten.

ξ1:



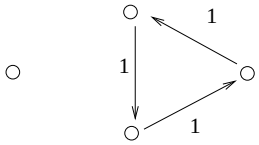
Kosten: −3+4−1=0

ξ2:



Kosten: −3+4+1+2*2−2*4= −2

ξ3:



Kosten: 2−4+1= −1

.

- Die Zirkulation ξ_1 besteht aus drei Kanten, entlang derer jeweils eine Einheit fließt. Die Kosten ergeben sich entsprechend als Summe der Kosten der drei Kanten.
- Dagegen ergeben sich die Kosten der Inkrementzirkulation ξ_2 als die Summe der Produkte der Flusswerte mit den Kantenkosten.
- Die Inkrementzirkulation ξ_3 führt zu negativen Kosten. Das ist zur Verbesserung des ursprünglichen Flusses x von Bedeutung.

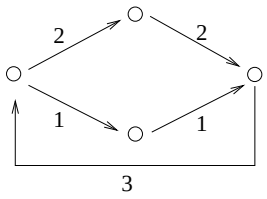
Wir wollen im folgenden möglichst einfache(!) Inkrementzirkulationen nutzen, um den gegebenen Fluss x zu verbessern. Dazu definieren wir die Addition einer Inkrementzirkulation ξ zu dem Fluss x wie folgt:
Sei x Zirkulation in N und sei ξ eine Inkrementzirkulation aus N_x. Wir ergänzen ξ zunächst auf den in N_x fehlenden Kanten durch 0. Die addierte Zirkulation $x' = x \oplus \xi$ ist dann definiert durch:

$$x'_{ij} = x_{ij} + \xi_{ij} - \xi_{ji}.$$

Die Addition einer Inkrementzirkulation zu dem gegebenen Fluss x soll an den drei Inkrementzirkulationen aus dem obigen Beispiel demonstriert werden.

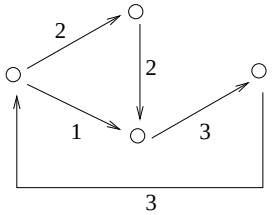
Beispiel 3.12

x⊕ ξ1:



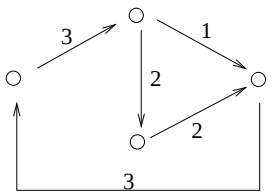
Kosten: 20+0=20

x⊕ ξ2:



Kosten: 20−2=18

x⊕ξ3:



Kosten: 20−1=19

Dargestellt sind die sich jeweils ergebenden Flüsse mit ihren Kosten. Man sieht, dass die Kosten der addierten Zirkulationen $x' = x \oplus \xi$ sich aus den alten Kosten für x (nämlich 20) und den Kosten der jeweiligen Inkrementzirkulation zusammensetzen.

Die Beobachtung aus den drei Beispielen werden wir nun formal nachvollziehen.

Satz 3.18 *Sei x zulässige Zirkulation in N und ξ eine zulässige Inkrementzirkulation. Dann ist $x' = x \oplus \xi$ eine zulässige Zirkulation in N mit*

$$c^t x' = c^t x + c_x^t \cdot \xi$$

Beweis:

- $Ax' = Ax + A\xi^+ - A\xi^- = 0$, weil x zulässig und $A\xi^+ - A\xi^- = 0$ nach Lemma 3.17.
- $l \leq x' \leq u$, denn

$$\begin{aligned} 0 \leq \xi_{ij} &\leq u_{ij} - x_{ij} && (3.4)\\ 0 \leq \xi_{ji} &\leq x_{ij} - l_{ij} && (3.5) \end{aligned}$$

$$\begin{aligned} \Longrightarrow \quad & x_{ij} - \xi_{ji} + \xi_{ij} \leq x_{ij} + \xi_{ij} \leq u_{ij} && \text{nach (3.4)}\\ & x_{ij} - \xi_{ji} + \xi_{ij} \geq x_{ij} - xi_{ji} \geq l_{ij} && \text{nach (3.5).} \end{aligned}$$

- Für die Kosten rechnen wir nach:

$$\begin{aligned} c^t x' &= \textstyle\sum_{(i,j)\in E} c_{ij}(x_{ij} + \xi_{ij} - \xi_{ji})\\ &= c^t x + \textstyle\sum_{(i,j)\in E_x} c_x(i,j)\xi_{ij}\\ &= c^t x + c_x^t \xi \end{aligned}$$

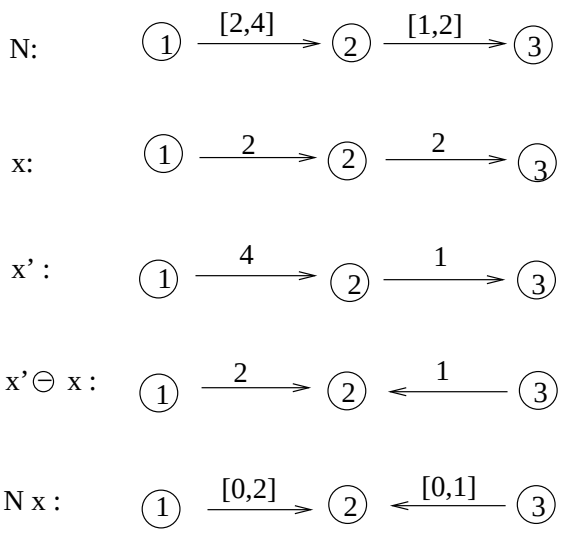
  denn $c_x(i,j) = -c_{ji}$ falls $(i,j) \in E_x$. QED

Man kann Zirkulationen auch voneinander abziehen, indem man definiert:
$\xi := x' \ominus x$, falls

$$\xi_{ij} = \begin{cases} \max\{0, x'_{ij} - x_{ij}\} & \text{falls } (i,j) \in E_x^+ \\ \max\{0, x_{ji} - x'_{ji}\} & \text{falls } (i,j) \in E_x^- \end{cases}$$

Satz 3.19 *$\xi := x' \ominus x$ ist eine zulässige Inkrementzirkulation in N_x die $x' = x \oplus \xi$ erfüllt.*

Beispiel 3.13 *Wir illustrieren die Subtraktion von Zirkulationen an folgendem Beispiel. (Das Inkrementnetzwerk N_x zu dem Netzwerk N und der Zirkulation x ist in der letzten Zeile angegeben.)*



Wir kommen nun endlich zum Hauptergebnis dieses Abschnittes.

Satz 3.20 (Optimalitätsbedingung für Zirkulationen) *x ist optimale Zirkulation $\Longleftrightarrow$ Es gibt keinen Dikreis negativer Länge (bzgl. c_x) in N_x.*

Beweis:

$''\Rightarrow''$: Annahme: Es existiert ein negativer Dikreis in N_x.
Zu zeigen: x ist nicht optimal.
Weil K ein negativer Dikreis ist, gilt

$$c_x(K) = \sum_{(i,j)\in K} c_x(i,j) < 0$$

Sei $\mathcal{E} := \min\{u_x(i,j) : (i,j) \in K\}$. Weil $(i,j) \in E_x$ für alle $(i,j) \in K$ gilt $\mathcal{E} > 0$.

Definiere $\xi = \mathcal{E}(K)$ als den zulässigen Dikreisfluss in N_x, d.h. $\xi_{ij} = \mathcal{E}$ für alle $(i,j) \in K$. Dann ergeben sich die Kosten von ξ zu

$$\begin{aligned} c_x^t\xi &= \textstyle\sum_{(i,j)\in K} c_x^t(i,j)\xi_{ij} \\ &= \xi \cdot \textstyle\sum_{(i,j)\in K} c_x^t(i,j) \quad < 0. \end{aligned}$$

Nach Satz 3.18 wissen wir, dass $x \oplus \xi$ eine zulässige Zirkulation in N ist, mit

$$c^t(x \oplus \xi) = c^t x + c_x^t\xi < c^t x,$$

also ist x nicht optimal.

$''\Leftarrow''$: Sei x' beliebige Zirkulation. Wir wollen zeigen, dass $c^t x \leq c^t x'$. Von Satz 3.19 wissen wir, dass $\xi = x' \ominus x$ eine zulässige Zirkulation in N_x ist. Nach Satz 3.15 zerfällt ξ also in Dikreisflüsse

$$\mathcal{E}_1(K_1) + \mathcal{E}_2(K_2) + \ldots + \mathcal{E}_l(K_l).$$

Nach unserer Voraussetzung existiert kein negativer. Dikreis, also gilt $c_x^t(K_i) \geq 0$ für alle $i = 1, \ldots, l$.

Zusammen ergibt sich:

$$c^t x' = c^t(x \oplus \xi) = c^t x + c_x^t \xi = c^t x + \mathcal{E}_1 c_x^t(K_1) + \cdots + \mathcal{E}_l c_x^t(K_l) \geq c^t x$$

QED

Weil die Zirkulationskante im Inkrementnetzwerk nicht mehr auftaucht, erhalten wir das folgende Korollar.

Korollar 3.21 *Satz 3.20 gilt auch für beliebige Flüsse.*

Aus Satz 3.20 ergibt sich das folgende Verfahren zur Bestimmung eines optimalen Flusses.

Algorithmus: Zum Auffinden eines optimalen Flusses.
Input: $N = (V, E, b, l, u, c)$ mit zulässigem Fluss x.

1. Bestimme N_x.
2. Solange ein Dikreis K negativer Länge in N_x existiert
   - 2.1 $\mathcal{E} := \min\{u_x(i,j) : (i,j) \in K\}$
   - 2.2 $x := x \oplus \mathcal{E}(K)$
   - 2.3 Bestimme N_x

Weil sich der Zielfunktionswert in jeder Iteration strikt verbessert, ist das Verfahren endlich.

Es sollte erwähnt werden, dass es noch andere Möglichkeiten gibt, Netzwerkflussprobleme zu lösen; beispielsweise durch das Netzwerk-Simplex-Verfahren. Dieses ist eine sehr effiziente Variante des Simplex-Verfahrens mit beschränkten Variablen, in der speziell ausgenutzt wird, dass Basislösungen im Netzwerkfkussproblem spannenden Bäumen entsprechen und somit effizient berechnet werden können.

Als nette Anwendung der Dualitätstheorie in der Netzwerkoptimierung besprechen wir exemplarisch noch zwei klassische Netzwerkprobleme mit ihren jeweiligen dualen Problemen. Zunächst charakterisieren wir maximale Flüsse durch minimale Schnitte, dann gehen wir auf das Transportproblem ein.

Maximale Flüsse und minimale Schnitte

Wir beginnen mit der graphentheoretischen Definition eines *Schnittes*.

Definition 3.22 *Sei $G = (V, E)$ ein Digraph. Eine Menge $Q \subseteq E$ nennt man einen* **s-t-Schnitt** *falls es eine Menge $X \subseteq V$ gibt, so dass $s \in X$, $t \notin X$ und*

$$Q = (X, V \backslash X) := \{(i, j) \in E : i \in X, j \notin X \text{ oder } i \notin X, j \in X\}.$$

Ist Q ein Schnitt, so definiert man weiter

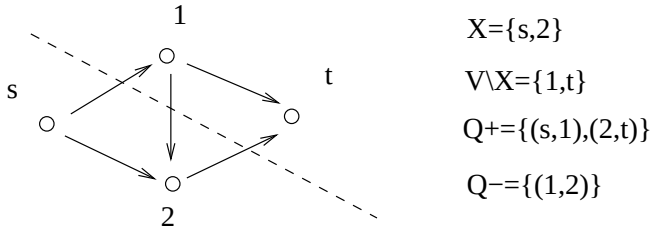
$$\begin{aligned} Q^+ &:= \{(i, j) \in Q : i \in X, j \notin X\} \\ Q^- &:= \{(i, j) \in Q : i \notin X, j \in X\}. \end{aligned}$$

Die **Kapazität** *des Schnittes ist gegeben durch*

$$C(Q) = \sum_{(i,j) \in Q^+} u_{ij}.$$

Ein s-t-Schnitt Q heißt **minimal**, *wenn er (unter allen möglichen s-t-Schnitten) $C(Q)$ minimiert.*

Die eingeführten Begriffe veranschaulicht die folgende Skizze.

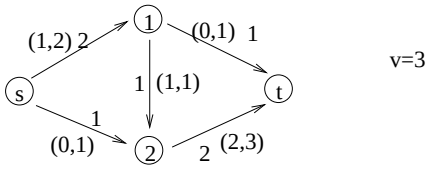


Ein s-t-Schnitt Q *trennt* die beiden Knoten s und t in folgendem Sinn: Entfernt man aus G alle Kanten aus Q, so gibt es keinen Weg von s nach t in dem reduzierten Graphen $G' = (V, E \setminus Q)$. Q wird daher auch *trennende Menge* genannt. Allerdings ist nicht jede trennende Menge von Kanten auch ein Schnitt.

Formuliert man das minimale Schnitt-Problem als lineares Programm und bildet dann das Duale dazu (Übung!), so erhält man ein maximales Flussproblem. Die Dualitätstheorie (Satz 2.34) liefert entsprechend den folgenden Satz (der von Ford und Fulkerson übrigens ohne Verwendung des Dualitätssatzes gezeigt wurde).

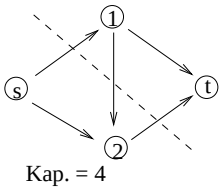
Satz 3.23 (Ford und Fulkerson: MaxFlow=MinCut) *Ein Fluss von s nach t ist maximal mit Wert v genau dann, wenn es einen minimalen Schnitt $Q = (X, X \setminus V)$ mit $C(Q) = v$ gibt.*

Bsp.:

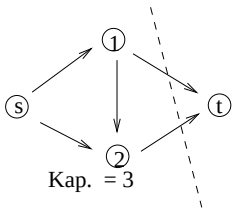


v=3

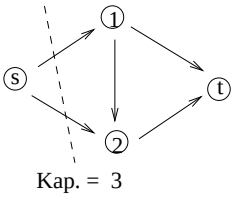
Cuts:



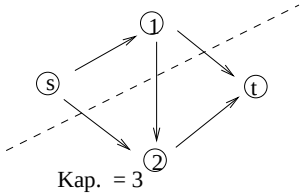
Kap. = 4



Kap. = 3



Kap. = 3



Kap. = 3

Es gilt also: Minimale Kapazität = 3 = maximaler Fluss.

Satz 3.23 wurde von Ford und Fulkerson veröffentlicht und ist daher unter ihrem Namen bekannt. Das Resultat wurde aber in einem geheimen Bericht schon vorher von Harris und Ross gezeigt, die für das Militär untersuchten, wie man den Transport von Material (maximaler Fluss) von der Sowjetunion nach Osteuropa durch Zerstörung möglichst weniger Brücken (minimaler Schnitt) blockieren kann.

Satz 3.23 hat verschiedene Anwendungen, u.a. folgt aus ihm direkt der folgende Satz von Menger.

Satz 3.24 (Satz von Menger) *Die maximale Anzahl der kantendisjunkten Wege von s nach t in einem gerichteten Graphen ist gleich der minimalen Anzahl an Kanten, die man entfernen muss, um s von t zu trennen.*

Andere Folgerungen aus Satz 3.23 sind der Satz von Dilworth über die Größe von Antiketten in partiell geordneten Mengen oder der Satz von Hall über perfekte Matchings in bipartiten Graphen.

Das Transportproblem

Zur Definition des *Transportproblems* seien m_1 Quellen $Q = \{q_1, \ldots, q_{m_1}\}$ und m_2 Senken $S = \{s_1, \ldots, s_{m_2}\}$ gegeben. Wir nehmen an, dass Quelle q_i bis zu a_i Einheiten eines Produktes produzieren kann und dass Senke s_j einen Bedarf von b_j Einheiten des gleichen Produktes hat. Weiter seien die Transportkosten w_{ij} einer Einheit des Produktes von q_i nach s_j bekannt.

Das Transportproblem $T = (Q, S, a, b, w)$ besteht darin, zu entscheiden wie viele Einheiten des Produktes von welcher Quelle zu welcher Senke transportiert werden sollen, so dass der Bedarf jeder Senke gedeckt ist und die Transportkosten minimal sind.

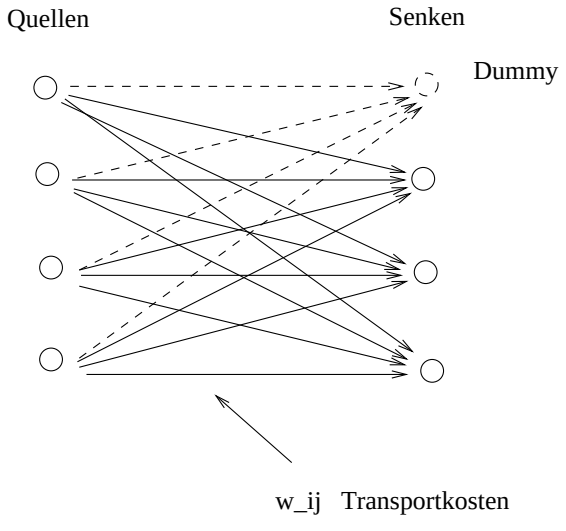
Wir beginnen damit, das Problem zu vereinfachen. Zunächst stellt man fest, dass eine zulässige Lösung des Transportproblems genau dann existiert, wenn $\sum_{i \in Q} a_i \geq \sum_{j \in S} b_j$, da der Bedarf der Senken andernfalls nicht gedeckt werden kann. Ohne Beschränkung der Allgemeinheit dürfen wir dann sogar annehmen, dass

$$\sum_{i \in Q} a_i = \sum_{j \in S} b_j, \tag{3.6}$$

da wir im Fall $\sum_{i \in Q} a_i > \sum_{j \in S} b_j$ eine *Dummy-Senke* s_{m_2+1} einfügen können, für die wir

$$\begin{aligned} b_{m_2+1} &= \sum_{i \in Q} a_i - \sum_{j \in S} b_j \quad \text{und} \\ w_{im_2+1} &= 0 \text{ für alle } i \in Q \end{aligned}$$

setzen und so ein äquivalentes Transportproblem mit nur einem zusätzlichen Knoten und $\sum_{i \in Q} a_i = \sum_{j \in S \cup \{s_{m_2+1}\}} b_j$ erhalten, siehe die folgende Skizze.



$b_{dummy} = \sum a_i - \sum b_j$

Im folgenden beschränken wir uns also auf Transportprobleme, die der Bedingung (3.6) genügen. Zunächst geben wir eine Formulierung als lineares Programm an. Dazu beschreiben wir die Menge, die von i nach j transportiert werden soll, durch Variablen x_{ij} für $i \in Q, j \in S$. Man erhält die folgende Formulierung.

$$\begin{aligned} \min \quad & \sum_{i \in Q} \sum_{j \in S} w_{ij} x_{ij} \\ \text{s.d.} \quad & \sum_{j \in S} x_{ij} = a_i \text{ für alle } i \in Q \\ & \sum_{i \in Q} x_{ij} = b_j \text{ für alle } j \in S \\ & x_{ij} \geq 0 \text{ für alle } i \in Q, j \in S. \end{aligned}$$

Eine zweite Vereinfachung beschreibt das folgende Lemma.

Lemma 3.25 *Jedes Transportproblem $T = (Q, S, a, b, w)$ ist äquivalent zu einem Transportproblem mit $w_{ij} \geq 0$ für alle $i \in Q, j \in S$.*

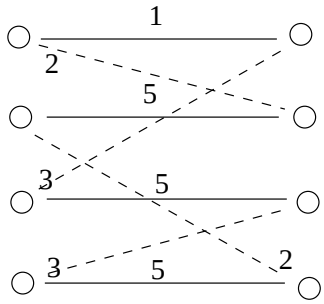
Beweis: Sei $T = (Q, S, a, b, w)$ ein Transportproblem mit $w_{ij} < 0$ für mindestens ein $i \in Q, j \in S$. Wir wählen $w \in \mathbb{R}$ so groß, dass $w_{ij} + w \geq 0$ für alle $i \in Q, j \in S$ und definieren $\bar{T} = (Q, S, a, b, \bar{w})$ mit $\bar{w}_{ij} := w_{ij} + w$. Dann gilt

$$\begin{aligned} \sum_{i \in Q} \sum_{j \in S} \bar{w}_{ij} x_{ij} &= \sum_{i \in Q} \sum_{j \in S} w_{ij} x_{ij} + w x_{ij} \\ &= \sum_{i \in Q} \sum_{j \in S} w_{ij} x_{ij} + w \sum_{i \in Q} \sum_{j \in S} x_{ij} \\ &= \sum_{i \in Q} \sum_{j \in S} w_{ij} x_{ij} + w \sum_{i \in Q} a_i, \end{aligned}$$

die Zielfunktionen von $T(Q, S, a, b, w)$ und $T(Q, S, a, b, \bar{w})$ unterscheiden sich also nur um eine additive Konstante und haben daher die gleichen Minimierer. QED

Der Spezialfall $m_1 = m_2$, $a_i = 1$ für alle $i \in Q$ und $b_j = 1$ für alle $j \in S$ ist unter dem Namen *Zuordnungsproblem* (oder *Assignmentproblem*) bekannt und hat z.B. Anwendungen, wenn man einer Gruppe von Personen Q eine Menge von Aufgaben S zuordnen möchte, so dass beispielsweise die Gesamtzeit für die Ausführung aller Aufgaben (wenn w_{ij} angibt, wie lange Person i zum Erledigen der Aufgabe j benötigt) minimal wird.

z.B. Personen − Aufgaben



Transportprobleme lassen sich effizient mit einem primal-dualen Verfahren lösen, das abschließend noch kurz skizziert werden soll. Einzelheiten findet man z.B. in [NW88].

Mit Hilfe des Tucker-Diagramms erhält man das Duale $D(Q, S, a, b, w)$ von dem Transportproblem $T(Q, S, a, b, w)$:

$$\begin{array}{ll} \max & \sum_{i \in Q} a_i u_i + \sum_{j \in S} b_j v_j \\ \text{s.d.} & u_i + v_j \leq w_{ij} \text{ für alle } i \in Q, j \in S \\ & u_i, v_j \gtrless 0 \text{ für alle } i \in Q, j \in S \end{array}$$

mit den Dualvariablen u_i für alle $i \in Q$ und v_j für alle $j \in S$.

Die Bedingung vom komplementären Schlupf führt zu

$$x_{ij}(w_{ij} - u_i - v_j) = 0 \ \forall i \in Q, j \in S$$

und nach dem Satz vom komplementären Schlupf (Satz 2.36) ergibt sich die folgende Aussage.

Lemma 3.26 *Seien $x_{ij} \geq 0$ für alle i, j. Dann gilt:*

x ist optimal für $T(Q, S, a, b, w)$ und u, v sind optimal für $D(Q, S, a, b, w)$ genau dann, wenn x, u, v die folgenden drei Bedingungen erfüllen:

(T1) $x_{ij}(w_{ij} - u_i - v_j) = 0$ *(Komplementaritätsbedingung)*

(T2) $w_{ij} - u_i - v_j \geq 0$ *(duale Zulässigkeit)*

(T3) $\sum_{j \in S} x_{ij} = a_i \ \forall i \in Q, \sum_{i \in Q} x_{ij} = b_j \ \forall j \in S$ *(primale Zulässigkeit)*

In dem primal-dualen Verfahren für $T(Q, S, a, b, w)$ verlangt man in jedem Schritt, dass die Komplementaritätsbedingungen (T1) und die duale Zulässigkeit (T2) erhalten bleiben und außerdem

$$\sum_{j \in S} x_{ij} \leq a_i \text{ für alle } i \in Q \text{ und } \sum_{i \in Q} x_{ij} \leq b_j \text{ für alle } j \in S \tag{3.7}$$

gilt. Man versucht nun, ein x zu konstruieren, das auch (T3) erfüllt. Dazu geht man iterativ vor:

Man beginnt mit einer dual zulässigen Startlösung, z.B.

$$\begin{aligned} u_i^0 &= \min_{j \in S} w_{ij} \text{ für alle } i \in Q \\ v_j^0 &= \min_{i \in Q} (w_{ij} - u_i^0) \text{ für alle } j \in S. \end{aligned}$$

Diese wird im Laufe des Verfahrens verändert, bis man eine Optimallösung erreicht. Sei dazu in einer Iteration des Verfahrens die dual zulässige Lösung u, v gegeben. Wir möchten herausfinden, ob man x so wählen kann, dass (T3) erfüllt ist. Dazu definiert man

$$J = \{(i, j) : w_{ij} - u_i - v_j = 0\}$$

und setzt $x_{ij} = 0$ für alle $(i, j) \notin J$. Man definiert ein *eingeschränktes Problem* $RT(Q, S, a, b, w)$ als

$$\begin{aligned} \max & \sum_{(i,j) \in J} x_{ij} \\ \text{s.d.} & \sum_{j \in S:(i,j) \in J} x_{ij} \leq a_i \text{ für alle } i \in Q \\ & \sum_{i \in Q:(i,j) \in J} x_{ij} \leq b_j \text{ für alle } j \in S \\ & x_{ij} \geq 0 \quad \forall (i, j) \in J. \end{aligned}$$

Sei x^* eine optimale Lösung von $RT(Q, S, a, b, w)$ und sei $z^* = \sum_{i \in Q, j \in S} x_{ij} = \sum_{(i,j) \in J} x_{ij}$ ihr Zielfunktionswert. Falls

$$z^* = \sum_{i \in Q} a_i = \sum_{j \in S} b_j \tag{3.8}$$

sind die drei Optimalitätsbedingungen für x^*, u, v erfüllt. Entsprechend ist x^* optimal für $T(Q, S, a, b, w)$ und u, v sind optimal für das duale $D(Q, S, a, b, w)$. Ist das nicht der Fall, so muss man die dualen Variablen u und v ändern.

Erfreulicherweise lässt sich $RT(Q, S, a, b, w)$ effizient als maximales Flussproblem lösen, indem man den folgenden Digraph konstruiert:

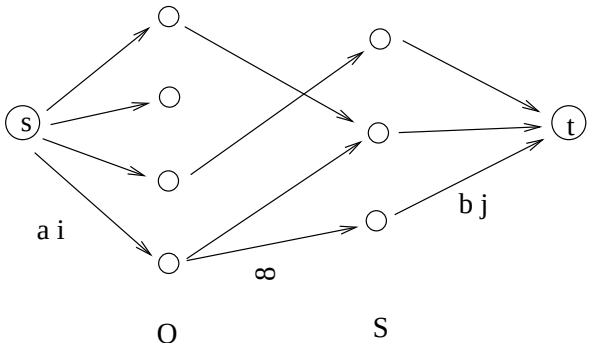
$$G_{RP} = (Q \cup S \cup \{s, t\}, E_{RP})$$

mit

$$E_{RP} = \{(i,j) : i \in Q, j \in S \text{ und } w_{ij} - u_i - v_j = 0\} \cup \{(s,i) : i \in Q\} \cup \{(j,t) : j \in S\}.$$

Als Kapazitäten setzt man

$$c(i,j) = \begin{cases} a_j & \text{falls } i = s, j \in Q \\ b_i & \text{falls } j = t, i \in S \\ \infty & \text{sonst.} \end{cases}$$



Die Lösung des daraus resultierenden maximalen Flussproblems ist dann eine Lösung von $RT(Q, S, a, b, w)$ und umgekehrt. Ist die Bedingung (3.8) erfüllt, so ist die Optimallösung erreicht. Anderenfalls kann man das maximale Flussproblem dazu nutzen, die dualen Variablen u und v so anzupassen, dass $\sum_{i \in Q} a_i$ im nächsten Schritt echt größer wird. Auf die Details werden wir hier nicht näher eingehen.

Kapitel 4

Nichtlineare Optimierung

4.1 Einführung und Begriffe

Nichtlineare Programme schreiben wir allgemein in der Form

$$(\mathbf{NL}) \qquad \min\{f(x) : x \in P\}$$

mit *zulässiger Menge* $P \subseteq \mathbb{R}^n$ und *Zielfunktion* $f : P \to \mathbb{R}$.

Die Aufgabe besteht darin, eine zulässige Lösung $x \in P$ zu finden, für die $f(x)$ möglichst klein ist. Dabei unterscheiden wir zwischen *globalen* und *lokalen* Minima, die folgendermaßen definiert sind.

Definition 4.1

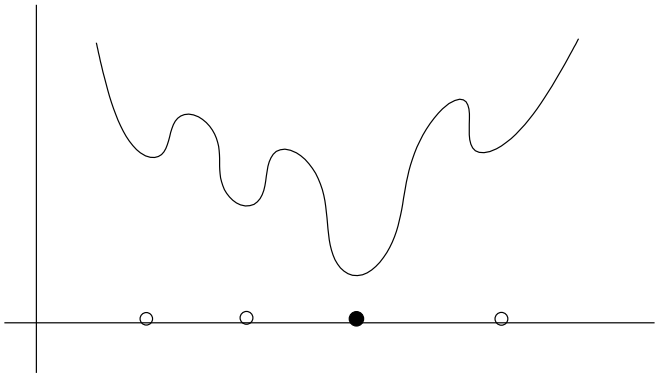
- $x \in P$ *ist ein* **globales Minimum von (NL)**, *falls*

$$f(y) \geq f(x) \text{ für alle } y \in P.$$

- $x \in P$ *ist ein* **lokales Minimum von (NL)**, *falls es ein* $\epsilon \geq 0$ *und eine Umgebung* $N_\epsilon(x) = \{y \in \mathbb{R}^n : \|y - x\| \leq y\}$ *gibt, so dass*

$$f(y) \geq f(x) \text{ für alle } y \in P \cap N_\epsilon(x).$$

Dabei ist es unerheblich, welche Norm $\|\cdot\|$ man in der Definition verwendet, da auf dem $\mathbb{R}^n$ alle Normen äquivalent sind.



Jedes globale Minimum ist auch ein lokales Minimum; die Umkehrung gilt aber nicht.

Bemerkung: Die Unterscheidung zwischen lokalen und globalen Minima war in der linearen Optimierung nicht nötig, weil in linearen Programmen jedes lokale Optimum gleichzeitig auch ein globales Optimum ist. In etwas allgemeinerer Form werden wir das in Satz 4.14 nachweisen.

Ein wichtiges Konzept in der nicht-linearen Optimierung ist Differenzierbarkeit.

Definition 4.2 *Sei $P \subset \mathbb{R}^n$ und sei $f : P \to \mathbb{R}$. Sei weiter ein $x \in P$ und ein Vektor $d \in \mathbb{R}^n \backslash \{0\}$ gegeben, so dass $x + \lambda d \in P$ für alle $\lambda < \lambda^0$ für ein $\lambda^0 \in \mathbb{R}^+$. Falls der folgende Limes*

$$f'(x,d) = \lim_{\lambda \to 0} \frac{f(x+\lambda d) - f(x)}{\lambda}$$

existiert, bezeichnet man $f'(x,d)$ als **die Richtungsableitung** *von f an x bezüglich d. Weiter definieren wir im Falle ihrer Existenz*

$$\frac{\partial f(x)}{\partial x_i} = f'(x, e_i)$$

als **i-te partielle Ableitung** *von f.*

Eine stärkere Forderung ist *(totale) Differenzierbarkeit.*

Definition 4.3 *Sei $f : P \to \mathbb{R}$, $x \in int(P)$. f heißt* **differenzierbar** *an x, wenn es einen Vektor $\nabla f(x)^t \in (\mathbb{R}^n)$ und eine Funktion $\alpha_x : \mathbb{R} \to \mathbb{R}$ gibt, so dass*

$$f(y) = f(x) + \nabla f(x)(y-x) + \|y-x\|\alpha_x(y-x) \text{ für alle } y \in P$$

und

$$\lim_{y \to x} \alpha_x(y-x) = 0.$$

$\nabla f(x)$ bezeichnet man dann als **Gradient** *von f an x.*

Bemerkung: Ist f differenzierbar in x, so ist der Gradient (an der Stelle x) eindeutig und es gilt:

$$\nabla f(x) = \left(\frac{\partial f(x)}{\partial x_1}, \cdots, \frac{\partial f(x)}{\partial x_n} \right).$$

Definition 4.4 *$f : P \to \mathbb{R}$ heißt* **zwei mal differenzierbar** *an $x \in int(P)$, wenn es $\nabla f(x)^t \in \mathbb{R}^n$, eine symmetrische $n \times n$-Matrix $H(x)$ und eine Funktion $\alpha_x : \mathbb{R}^n \to \mathbb{R}$ gibt, so dass*

$$f(y) = f(x) + \nabla f(x)(y-x) + \frac{1}{2}(y-x)^t H(x)(y-x) + \|y-x\|^2 \alpha_x(y-x) \text{ für alle } y \in P$$

und

$$\lim_{y \to x} \alpha_x(y-x) = 0.$$

$H(x)$ heißt **Hesse-Matrix von f an x.**

Bemerkung: Ist f zwei mal differenzierbar, so hat die Hesse-Matrix die folgende Gestalt:

$$H(x) = \begin{bmatrix} \frac{\partial^2 f(x)}{\partial x_1^2} & \frac{\partial^2 f(x)}{\partial x_1 \partial x_2} & \cdots & \frac{\partial^2 f(x)}{\partial x_1 \partial x_n} \\ \vdots & & & \vdots \\ \frac{\partial^2 f(x)}{\partial x_n \partial x_1} & \cdots & \cdots & \frac{\partial^2 f(x)}{\partial x_n^2} \end{bmatrix}.$$

Aus der Analysis übernehmen wir den folgenden Satz über Optimallösungen bei differenzierbaren Funktionen für den Fall $P = \mathbb{R}^n$, d.h. für den Fall dass alle Lösungen aus dem $\mathbb{R}^n$ erlaubt sind. Solche Probleme bezeichnet man als *nichtrestringierte Probleme*. Wir erinnern zunächst an die folgende Bezeichnung.

Notation 4.5

- *Eine Matrix H heißt positiv definit, falls $d^t H d > 0$ für alle $d \in \mathbb{R}^n \setminus \{0\}$.*
- *Eine Matrix H heißt positiv semi-definit, falls $d^t H d \geq 0$ für alle $d \in \mathbb{R}^n \setminus \{0\}$.*

Der folgende Satz ist aus der Analysis bekannt.

Satz 4.6 *Betrachte $\min\{f(x) : x \in \mathbb{R}^n\}$ mit einer zwei mal differenzierbaren Funktion f. Dann gelten die folgenden Aussagen:*

- *Ist x ein lokales Minimum, so ist gilt $\nabla f(x) = 0$ und $H(x)$ ist positiv semi-definit.*

- *Ist $\nabla f(x) = 0$ und $H(x)$ ist positiv definit, so ist x ein lokales Minimum von f.*

Im folgenden werden wir uns mit der Frage beschäftigen, für welche Funktionen f ein globales Minimum existiert, und wie man es findet. Dabei gehen wir insbesondere auf *restringierte* Probleme ein, also Optimierungsprobleme mit $P \subset \mathbb{R}^n$.

Als erstes verallgemeinern wir das graphische Verfahren für lineare Programme aus Abschnitt 2.1. Dazu führen wir die folgenden Begriffe ein.

Definition 4.7 *Sei $f : \mathbb{R}^n \to \mathbb{R}, \quad z \in \mathbb{R}$.*
Die **Niveaumenge** *von f an z ist gegeben durch*

$$L_{\leq}(z) = \{x \in \mathbb{R}^n : f(x) \leq z\}.$$

Die **Konturlinie** *(oder* **Nivealinie***) von f an z ist*

$$L_{=}(z) = \{x \in \mathbb{R}^n : f(x) = z\}.$$

Das graphische Verfahren für nichtlineare Programme (*Niveaumengen-Verfahren*) beruht darauf, dass man die Niveaumenge um die Lösung x^* des unrestringierten Optimums wachsen lässt, bis ein zulässiger Punkt erreicht ist. Formal beruht das Ergebnis auf der folgenden Beobachtung.

Lemma 4.8 *Für das nichtlineare Optimierungsproblem* $\min\{f(x) : x \in P\}$ *gilt:*

$$\inf\{f(x) : x \in P\} = \inf\{z' : L_{\leq}(z') \cap P \neq \emptyset\}.$$

Ist $z^* = \min\{f(x) : x \in P\}$ so ist jedes $x \in L_{\leq}(z^*) \cap P$ optimal.

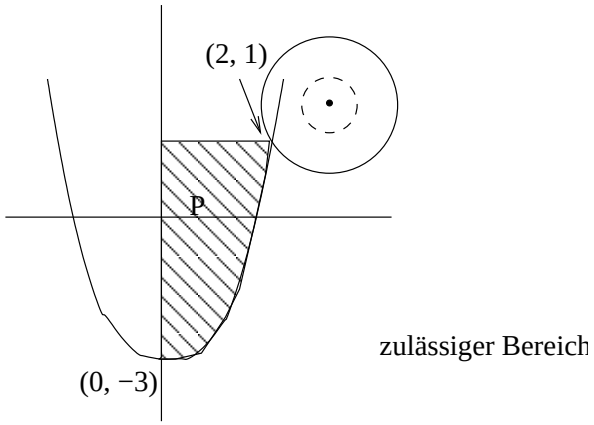
Beweis: Der Beweis folgt direkt aus der Definition der Niveaumengen. QED

Im graphischen Verfahren zur Lösung nichtlinearer Programme bestimmt man zunächst eine Optimallösung des nicht-restringierten Problems $\min\{f(x) : x \in \mathbb{R}^n\}$ (falls sie existiert). Dann zeichnet man die Niveaulinien um diese Lösung zu einem kleinen Niveau z ein, und lässt z so lange wachsen, bis die Niveaumenge einen zulässigen Punkt aus P enthält. Wir demonstrieren das Verfahren an folgendem Beispiel eines quadratischen, restringierten Optimierungsproblems.

Beispiel 4.1

$$\begin{array}{lll} \min & f(x_1, x_2) = (x_1 - 3)^2 + (x_2 - 2)^2 & \\ \text{s.d.} & x_1^2 - x_2 - 3 \leq 0 & (d.h.\ x_2 \geq x_1^2 - 3) \\ & x_2 - 1 \leq 0 & \\ & -x_1 \leq 0 & \end{array}$$

Graph:



Die Niveaumengen

$$L_{\leq}(z) = \{x \in \mathbb{R}^2 : (x_1 - 3)^2 + (x_2 - 2)^2 \leq z\}$$

sind gegeben durch Kreise um $\begin{pmatrix} 3 \\ 2 \end{pmatrix}$ *mit Radius* $\sqrt{z}$*.*
Graphisch erhält man die Optimallösung $x^* = \begin{pmatrix} 2 \\ 1 \end{pmatrix} \in P$ *als Schnittpunkt des kleinsten Kreises um* $\begin{pmatrix} 3 \\ 2 \end{pmatrix}$ *mit* P*. Der Zielfunktionswert ergibt sich zu* $f(x^*) = (2-3)^2+(1-2)^2 = 2$*; der Kreis, der die entsprechende Niveaumenge repräsentiert, hat also Radius* $\sqrt{2}$*.*

Da das graphische Verfahren (aufgrund der oft unangenehmen Struktur der Niveaumengen) keine numerische Bedeutung hat, verzichten wir auf eine formale Beschreibung. Zur Veranschaulichung von Optimierungsproblemen und zur Herleitung endlicher Kandidatenmengen für kontinuierliche Probleme (so genannter *finite dominating sets*) erweist es sich aber in vielen Optimierungsproblemen als nützlich. Insbesondere bei geometrischen Problemen, z.B. aus der Standortplanung kann die Struktur der Niveaumengen zur Bestimmung von Minima ausgenutzt werden.

Es soll noch bemerkt werden, dass für eine lineare Funktion f die Niveaumengen $L_{\leq}(z)$ Halbräume sind. Das graphische Verfahren aus Abschnitt 2.1 ist also ein Spezialfall des Niveaumengen-Verfahrens.

4.2 Konvexe Programmierung

In der konvexen Optimierung beschäftigen wir uns, wie der Name schon andeutet, mit *konvexen Funktionen* auf *konvexen Mengen*.

In Definition 2.5 auf Seite 16 haben wir *konvexe Mengen* folgendermaßen definiert: Eine Menge $\mathcal{M} \subseteq \mathbb{R}^n$ heißt **konvex**, falls $\forall x, y \in \mathcal{M}$ und $\forall \lambda \in (0,1)$ gilt:

$$\lambda x + (1-\lambda)y \in \mathcal{M}.$$

Für reellwertige Funktionen definiert man *Konvexität* wie folgt.

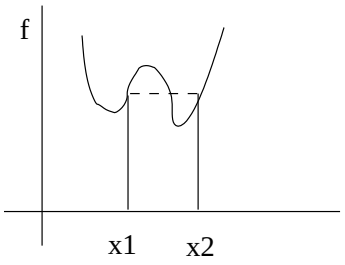
Definition 4.9 *Sei $\mathcal{M} \subseteq \mathbb{R}^n$ konvex und $f : \mathcal{M} \to \mathbb{R}$ eine reellwertige Funktion. f heißt* **konvex**, *falls $\forall x, y \in \mathcal{M}$ und $\forall \lambda \in (0,1)$ gilt:*

$$f(\lambda x + (1-\lambda)y) \leq \lambda f(x) + (1-\lambda) f(y).$$

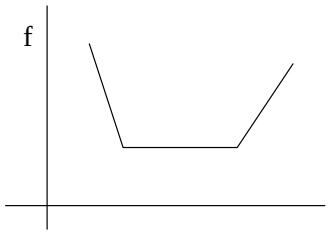
f heißt **streng konvex**, *falls $\forall x, y \in \mathcal{M}$ und $\forall \lambda \in (0,1)$ gilt:*

$$f(\lambda x + (1-\lambda)y) < \lambda f(x) + (1-\lambda) f(y)$$

f heißt **konkav**, *falls $-f$ konvex ist.*
$f : \mathcal{M} \mapsto \mathbb{R}$ heißt **streng konkav**, *falls $-f$ streng konvex ist.*



f nicht konvex



f konvex, aber nicht streng konvex

Jetzt können wir beschreiben, wann ein konvexes Optimierungsproblem vorliegt.

Definition 4.10 *Sei $P \subseteq \mathbb{R}^n$ eine konvexe Menge und $f : P \to \mathbb{R}$ eine konvexe Funktion. Dann heißt das nichtlineare Problem*

$$\min\{f(x) : x \in P\} \tag{4.1}$$

konvexes Optimierungsproblem (konvexes Programm).

Wir stellen zunächst einige Eigenschaften von konvexen Funktionen zusammen.

Lemma 4.11

a) Sei $P \subseteq \mathbb{R}^n$ konvexe Menge, $f_1, \ldots, f_m$: $P \to \mathbb{R}$ konvexe Funktionen und $\alpha_1, \ldots, \alpha_m \geq 0$. Dann sind

$$f(x) = \sum_{i=1}^{m} \alpha_i f_i(x) \text{ und } f(x) = \max_{i=1,\ldots,m} \alpha_i f_i(x)$$

konvexe Funktionen.

b) Sei $h : \mathbb{R}^n \to \mathbb{R}$ konvex und $g : \mathbb{R} \to \mathbb{R}$ monoton wachsend und konvex. Dann ist $f(x) = g \circ h(x)$ konvex.

c) Sei $h : \mathbb{R}^n \to \mathbb{R}$ konkav und $g : \mathbb{R} \to \mathbb{R}$ monoton fallend und konvex. Dann ist $f(x) = g \circ h(x)$ konvex.

d) Sei h linear und g konvex. Dann ist $f(x) = g \circ h(x)$ konvex.

Beweis: Übung!

Insbesondere folgt aus dem Satz die manchmal nützliche Tatsache, dass der Quotient

$$f(x) = \frac{1}{g(x)}$$

einer konkaven Funktion $g : \mathbb{R}^n \to \mathbb{R}$ über $P = \{x : g(x) > 0\}$ konvex ist.

Wir notieren weiter:

Lemma 4.12 *Sei $f : P \to \mathbb{R}$ eine konvexe Funktion auf der konvexen Menge P. Dann sind ihre Niveaumengen $L_{\leq}(z)$ konvex für alle $z \in \mathbb{R}$.*

Beweis: Sei $z \in \mathbb{R}$ und seien $x, y \in L_{\leq}(z)$. Dann gilt

$$f(\lambda x + (1 - \lambda y)) \leq \lambda \underbrace{f(x)}_{\leq z} + (1 - \lambda) \underbrace{f(y)}_{\leq z} \leq \lambda z + (1 - \lambda)z = z,$$

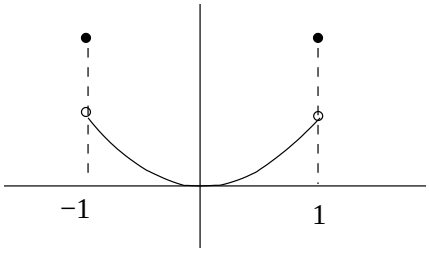
also ist $\lambda x + (1 - \lambda)y \in L_{\leq}(z)$. QED

Satz 4.13 *Sei $P \subseteq \mathbb{R}^n$ konvex. Eine konvexe Funktion $f : P \to \mathbb{R}$ ist auf $int(P)$ stetig.*

Wir verzichten auf den Beweis, vermerken aber, dass die Voraussetzung $int(P)$ wichtig ist: Auf dem Rand des Definitionsbereichs muss eine konvexe Funktion nicht stetig sein, wie das folgende Beispiel demonstriert.

Beispiel 4.2 *Sei $P = \{x : |x| \leq 1\}$ und*

$$f(x) = \begin{cases} x^2 & \text{für } |x| < 1 \\ 2 & \text{für } |x| = 1 \end{cases}$$



Dann ist f konvex, aber nicht stetig auf P.

Satz 4.14 *Sei $P \subseteq \mathbb{R}^n$ konvex, $P \neq \emptyset$, $f : P \to \mathbb{R}$ konvex. Wir betrachten das konvexe Optimierungsproblem (4.1), also* $\min\{f(x) : x \in P\}$. *Es gilt:*

1. *Jedes lokale Minimum von (4.1) ist ein globales Minimum von (4.1).*
2. *Die Menge der Minimierer*

$$M^* = \{x \in P : f(x) \leq f(y) \text{für alle } y \in P\}$$

von (4.1) ist konvex.

Beweis:

1. Sei $\bar{x}$ lokales Minimum. Dann existiert $\varepsilon > 0$ so dass $f(y) \geq f(\bar{x})$ für alle $y \in P \cap N_\varepsilon(\bar{x})$. Wir nehmen an, dass $\bar{x}$ kein globales Optimum ist. Dann existiert $x^* \in P$ mit $f(x^*) < f(\bar{x})$. Betrachte nun für $\lambda \in (0, 1)$ den Punkt

$$y_\lambda := \lambda x^* + (1 - \lambda)\bar{x}.$$

   Dann ist y_λ zulässig für (4.1), denn wegen $\bar{x}, x^* \in P$ folgt aus der Konvexität von P dass auch $y_\lambda \in P$. Für den Zielfunktionswert von y_λ ergibt sich:

$$\begin{aligned} f(y_\lambda) &= f(\lambda x^* + (1 - \lambda)\bar{x}) \\ &\leq \lambda f(x^*) + (1 - \lambda) f(\bar{x}) \\ &< \lambda f(\bar{x}) + (1 - \lambda) f(\bar{x}) = f(\bar{x}). \end{aligned}$$

   Für $\lambda \to 0$ erhält man

$$\|y_\lambda - \bar{x}\| < \varepsilon,$$

   also gilt $y_\lambda \in N_\varepsilon(\bar{x})$ und somit $f(y_\lambda) \geq f(\bar{x})$. Das ist ein Widerspruch zu $f(y_\lambda) < f(\bar{x})$.

2. Für $z^* = \min\{f(x) : x \in P\}$ ist die Menge der Optimallösungen $M^* = L_\leq(z^*) \cap P$, daher ist M^* nach Lemma 4.12 und als Schnitt von zwei konvexen Mengen konvex. QED

Insbesondere folgt aus dem zweiten Teil des Satzes, dass die Menge der Optimallösungen eine zusammenhängende Menge ist. Für streng konvexe Funktionen kann man obige Aussage noch verschärfen.

Lemma 4.15 *Sei $P \subseteq \mathbb{R}^n$ konvex, und $f : P \to \mathbb{R}$ streng konvex. Dann hat das konvexe Optimierungsproblem (4.1) höchstens ein Minimum.*

Beweis: Angenommen, x_1, x_2 sind beide Minima mit $x_1 \neq x_2$ und $f(x_1) = f(x_2) = z^*$. Betrachte einen beliebigen Punkt zwischen x^1 und x^2, z.B. $\tilde{x} := \frac{1}{2}x_1 + \frac{1}{2}x_2$. Weil P eine konvexe Menge ist, gilt $\tilde{x} \in P$. Wegen der strengen Konvexität folgt weiter

$$f(\tilde{x}) = f(\frac{1}{2}x_1 + \frac{1}{2}x_2) < \frac{1}{2}f(x_1) + \frac{1}{2}f(x_2) = z^*,$$

aber das ist ein Widerspruch zur Optimalität von x_1 und x_2. QED

Man beachte, dass das Minimum nicht existieren muss, selbst wenn P eine konvexe und kompakte Menge ist. Ein einfaches Gegenbeispiel hierfür liefert die folgende Funktion

$$f(x) = \begin{cases} x & 0 < x \leq 1 \\ 1 & x = 0 \end{cases},$$

die auf der konvexen und kompakten Menge $P = [0, 1]$ kein Minimum hat.

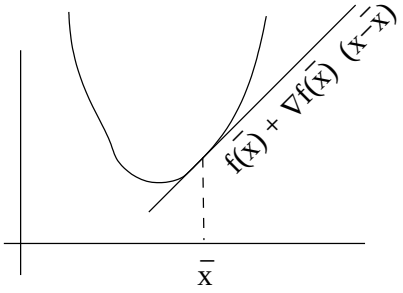
Im folgenden leiten wir ein Verfahren her, mit dem man das Minimum einer konvexen Funktion numerisch bestimmen kann. Dieses Verfahren ähnelt einem *Steepest descent* Verfahren für stetige Funktionen, verwendet aber statt des (nicht notwendigerweise existierenden Gradienten) einen *Subgradienten* der zu minimierenden Funktion. Es ist daher unter dem Namen *Subgradienten-Verfahren bekannt*. Wir benötigen zunächst einige Begriffe.

Definition 4.16 *Sei $P \subseteq \mathbb{R}^n$ konvex, sei $f : P \to \mathbb{R}$ eine konvexe Funktion und $\bar{x} \in P$. Dann nennt man $\xi \in (\mathbb{R}^n)^*$* **Subgradient** *von f an $\bar{x}$, falls*

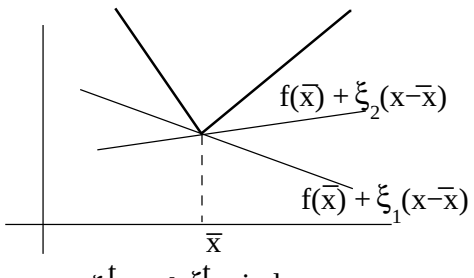
$$f(x) \geq f(\bar{x}) + \xi(x - \bar{x}) \text{ für alle } x \in P.$$

Die Menge aller Subgradienten ξ heißt **Subdifferential** *von f an $\bar{x}$.*

Die Funktion $f(\bar{x}) + \xi(x - \bar{x})$ ist eine lineare Funktion durch den Punkt $(\bar{x}, f(\bar{x}))$, die ganz unterhalb der Funktion f liegt. Die folgende Abbildung verdeutlicht, dass diese nicht eindeutig sein muss.



∇f(x̄): Steigung von f an x̄



ξ_1^t und ξ_2^t sind beides Subgradienten.

Lemma 4.17 *Das Subdifferential von f an $\bar{x}$ ist konvex.*

Beweis: Seien ξ^1, ξ^2 beides Subgradienten von f an $\bar{x}$ und sei $\lambda \in (0,1)$. Wir möchten zeigen, dass

$$\xi = \lambda\xi^1 + (1-\lambda)\xi^2$$

ebenfalls ein Subgradient von f an $\bar{x}$ ist. Sei dazu $y \in P$. Dann gilt:

$$\begin{aligned}
\xi(y-\bar{x}) + f(\bar{x}) &= [\lambda\xi^1 + (1-\lambda)\xi^2](y-\bar{x}) + f(\bar{x}) \\
&= \lambda\xi^1(y-\bar{x}) + \lambda f(\bar{x}) + (1-\lambda)\xi^2(y-\bar{x}) + (1-\lambda)f(\bar{x}) \\
&= \lambda\left(\xi^1(y-\bar{x}) + f(\bar{x})\right) + (1-\lambda)\left(\xi^2(y-\bar{x}) + f(\bar{x})\right) \\
&\leq \lambda f(y) + (1-\lambda)f(y) = f(y),
\end{aligned}$$

also ist ξ tatsächlich im Subdifferential. QED

Bevor wir zum Subgradientenverfahren kommen, diskutieren wir, wann Subgradienten existieren, welche Beziehung zwischen Subgradient und Gradient besteht, und wie man mit ihrer Hilfe Minima charakterisieren kann. Dazu benötigen wir den folgenden Trennungssatz.

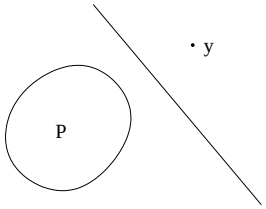
Satz 4.18 (Trennungssatz für abgeschlossene, konvexe Mengen) *Sei $y \in \mathbb{R}^n$ und sei $P \subseteq \mathbb{R}^n$ eine konvexe und abgeschlossene Menge, mit $y \notin P$. Dann existiert ein $\alpha \in \mathbb{R}$ und ein Vektor $q \in \mathbb{R}^n$, so dass die folgenden beiden Bedingungen erfüllt sind:*

$$\begin{aligned}
q^t y &> \alpha \text{ und} \\
q^t x &\leq \alpha \text{ für alle } x \in P.
\end{aligned}$$

Geometrisch bedeutet die Aussage des Satzes, dass die Hyperebene

$$H = H(q,\alpha) = \{x \in \mathbb{R}^n : q^t x = \alpha\}$$

den Punkt y von der konvexen Menge P *trennt*. Man bezeichnet H daher auch als *trennende Hyperebene*.



Beweis: Um den Satz zu beweisen, konstruieren wir eine Hyperebene $H = H(q,\alpha)$, die

- durch den Punkt $\bar{x} \in P$ geht, der am nächsten an y liegt, und die

- senkrecht zu $\bar{x} - y$ ist

und beweisen dann, dass diese Hyperebene y von P trennt.

Dazu behaupten wir zunächst, dass es ein $\bar{x}$ gibt, das die folgenden Bedingungen erfüllt:

$$\begin{aligned} (\bar{x} - y)^t(\bar{x} - y) &\leq (x - y)^t(x - y) \text{ für alle } x \in P, \text{ und} && (4.2)\\ (x - \bar{x})^t(y - \bar{x}) &\leq 0 \text{ für alle } x \in P. && (4.3) \end{aligned}$$

Zum Beweis dieser Behauptung nutzen wir, dass $f(x) = (x - y)^t(x - y)$ stetig ist und für ein beliebiges $\hat{x} \in P$ gilt, dass

$$\hat{P} = \{x \in P : \|x - y\| \leq \|\hat{x} - y\|\}$$

eine kompakte Menge ist. Wegen

$$\inf_{x \in P}\{f(x)\} = \inf_{x \in \hat{P}}\{f(x)\} = \min_{x \in \hat{P}}\{f(x)\}$$

existiert ein Minimierer $\bar{x}$, für den $f(\bar{x}) \leq f(x)$ für alle $x \in P$ gilt; $\bar{x}$ erfüllt also (4.2). (Man bezeichnet $\bar{x}$ auch als die *Projektion* von y auf die Menge P).

Wir rechnen weiter nach, dass $\bar{x}$ auch (4.3) erfüllt: Angenommen, nein. Dann existiert ein x', so dass $(x' - \bar{x})^t(y - \bar{x}) > 0$. Wir setzen

$$x_\lambda := \lambda x' + (1 - \lambda)\bar{x} = \lambda(x' - \bar{x}) + \bar{x}$$

als Konvexkombination des Gegenbeispiels x' und der Projektion $\bar{x}$. Für $\lambda \to 0$ gilt:

$$\begin{aligned} (x_\lambda - y)^t(x_\lambda - y) &= \underbrace{\lambda^2(x' - \bar{x})^t(x' - \bar{x})}_{\to 0} + \underbrace{2\lambda(x' - \bar{x})^t(\bar{x} - y)}_{<0} + (\bar{x} - y)^t(\bar{x} - y)\\ &< (\bar{x} - y)^t(\bar{x} - y) \text{ für } \lambda \to 0; \end{aligned}$$

also $f(x_\lambda) < f(\bar{x}_\lambda)$ für λ hinreichend klein. Das ist ein Widerspruch, weil $\bar{x}$ als Minimierer von f vorausgesetzt wurde.

Es fehlt noch die Konstruktion von q und α. Hierzu definieren wir

$$\begin{aligned} q &:= (y - \bar{x}) \text{ und}\\ \alpha &:= \bar{x}^t(y - \bar{x}) = q^t\bar{x}. \end{aligned}$$

Zunächst gilt $q \neq 0$, denn $\bar{x} \in P$, $y \notin P$. Mit den gesetzten Werten für q und α erhält man weiter

$$\begin{aligned} q^t x &= x^t(y - \bar{x}) \leq \bar{x}^t(y - \bar{x}) = \alpha \text{ für alle } x \in P \text{ (wegen (4.3))}, \text{ und}\\ q^t y - \alpha &= (y - \bar{x})^t y - (y - \bar{x})^t\bar{x} = (y - \bar{x})^t(y - \bar{x}) = \|y - \bar{x}\|^2 > 0, \end{aligned}$$

also $q^t y > \alpha$. QED

Der Trennungssatz gilt auch ohne die Voraussetzung, dass P abgeschlossen ist:

Satz 4.19 (Trennungssatz) *Sei $y \in \mathbb{R}^n$ und sei $P \subseteq \mathbb{R}^n$ eine konvexe Menge, mit $y \notin cl(P)$. Dann existiert ein $\alpha \in \mathbb{R}$ und ein Vektor $q \in \mathbb{R}^n$, so dass die folgenden beiden Bedingungen erfüllt sind:*

$$\begin{aligned} q^t y &> \alpha \text{ und} \\ q^t x &\leq \alpha \text{ für alle } x \in P. \end{aligned}$$

Beweis: Definiere $P' := cl(P)$ als den Abschluss von P. Nach Satz 4.18 existieren dann α und q, so dass

$$\begin{aligned} q^t y &> \alpha \text{ und} \\ q^t x &\leq \alpha \text{ für alle } x \in P'. \end{aligned}$$

Weil $P \subseteq P'$ sind die in Satz 4.19 geforderten Bedingungen erfüllt. QED

Eine weitere Folgerung aus dem Trennungssatz ist die Existenz von *unterstützenden Hyperebenen* (*supporting hyperplanes*) an konvexe Mengen.

Korollar 4.20 *Sei $P \subseteq \mathbb{R}^n$ konvex und sei $y \in \partial P$ ein Punkt auf dem Rand von P. Dann gibt es $q \in \mathbb{R}^n \backslash \{0\}$, so dass*

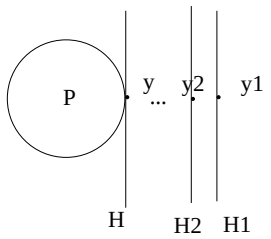
$$q^t(x - y) \leq 0 \text{ für alle } x \in P.$$

Man sagt, $H = \{x \in \mathbb{R}^n : q^t(x - y) = 0\}$ *unterstützt* P an y und bezeichnet H als *unterstützende Hyperebene.*

Beweis: Wir skizzieren die Idee des Beweises. Weil $y \in \partial P$ gibt es eine Folge $(y_k) \subseteq \mathbb{R}^n \setminus cl(P)$ mit $y_k \to y$. Für jedes $k \in \mathbb{N}$ liefert Satz 4.19 eine Hyperebene

$$H_k = \{x : q_k^t x = \alpha_k\},$$

die y_k von P trennt. Durch den Grenzübergang $k \to \infty$ erhält man q und α als Grenzwerte der Folgen q_k und α_k, und die resultierende Hyperebene $H = \{x : q^t x = \alpha = q^t y\}$ erfüllt $q^t(x - y) \leq 0 \ \forall x \in P$.



QED

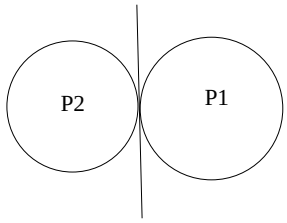
Wir nennen noch eine weitere Folgerung aus dem Trennungssatz.

Korollar 4.21 *Seien P_1, P_2 konvexe Mengen und sei $P_1 \cap P_2 = \emptyset$. Dann existiert ein $q \in \mathbb{R}^n$ und ein $\alpha \in \mathbb{R}$, so dass*

$$\begin{aligned} q^t x &\leq \alpha \text{ für alle } x \in P_1, \text{ und} \\ q^t x &\geq \alpha \text{ für alle } x \in P_2. \end{aligned}$$

Beweis: Wir gehen auch bei diesem Beweis nicht auf jedes Detail ein. Wie beim Beweis des Trennungssatzes behandelt man zunächst den Fall von abgeschlossenen Mengen, und macht sich danach klar, dass das Ergebnis bei offenen Mengen gültig bleibt. Man geht in folgenden Schritten vor.

1. Wähle $x_1 \in P_1$ und $x_2 \in P_2$ so dass $\|x_2 - x_1\| \leq \|x_2' - x_1'\|$ für alle $x_2' \in P_2, x_1' \in P_1$.



2. Benutze den Trennungssatz (Satz 4.19), um P_2 von x_1 zu trennen. Es gibt also eine Hyperebene $H(q, \alpha)$ mit $q = x_1 - x_2$ und $\alpha = x_2^t(x_1 - x_2)$, so dass

$$\begin{aligned} q^t x_1 &> \alpha \text{ und} \\ q^t x &\leq \alpha \text{ für alle } x \in P_2 \end{aligned}$$

3. Um zu zeigen, dass H auch P_2 von P_1 trennt, verifiziert man wie im Beweis vom Satz 4.18 dass

$$\begin{aligned} (x - x_1)^t(x_2 - x_1) &\leq 0 \text{ für alle } x \in P_1 \\ (x - x_2)^t(x_1 - x_2) &\leq 0 \text{ für alle } x \in P_2. \end{aligned}$$

Daraus folgt:

$$(x - x_2)^t q \leq 0 \leq (x' - x_1)^t q \quad \forall x \in P_2,\ x' \in P_1,$$

also

$$\begin{aligned} x^t q &\leq x_2^t q = \alpha \text{ für alle } x \in P_2, \text{ und} \\ x'^t q &\geq x_1^t q > \alpha \text{ für alle } x' \in P_1. \end{aligned}$$

QED

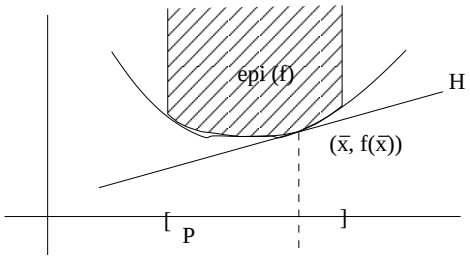
Jetzt kommen wir zu den Subgradienten zurück. Wir erinnern an die Definition: $\xi \in (\mathbb{R}^n)^*$ ist Subgradient von f an $\bar{x}$, falls $f(x) \geq f(\bar{x}) + \xi(x - \bar{x})$ für alle $x \in P$.

Satz 4.22 *Sei $P \subseteq \mathbb{R}^n$ nichtleer und konvex und sei $f : P \to \mathbb{R}$ konvex. Sei weiter $\bar{x} \in int(P)$. Dann existiert ein Subgradient ξ von f an $\bar{x}$.*

Beweis: Sei $\bar{x} \in int(P)$. Da f konvex ist, ist

$$\text{epi}(f) := \{(x, z) : x \in P \text{ und } z \geq f(x)\}$$

eine konvexe Menge (Übung).



Wir wenden Korollar 4.20 auf $P = \text{epi}(f)$ und $\bar{y} = (\bar{x}, f(\bar{x}))$ an und erhalten ein $\tilde{q} \in \mathbb{R}^{n+1}$, $\tilde{q} \neq 0$ so dass

$$\tilde{q}^t \left(\begin{pmatrix} x \\ z \end{pmatrix} - \begin{pmatrix} \bar{x} \\ f(\bar{x}) \end{pmatrix} \right) \leq 0 \text{ für alle } (x, z) \in \text{epi}(f).$$

Mit $\tilde{q} = \begin{pmatrix} q \\ \mu \end{pmatrix}$ erhält man

$$q^t(x - \bar{x}) + \mu(z - f(\bar{x})) \leq 0 \text{ für alle } (x, z) \in \text{epi}(f). \tag{4.4}$$

Behauptung: $\mu < 0$, denn:

- Wäre $\mu > 0$, so könnte (4.4) für festes x und hinreichend großes $z > f(x)$ nicht erfüllt sein.
- Wäre $\mu = 0$, so ist (4.4) äquivalent zu $q^t(x - \bar{x}) \leq 0$ für alle $x \in P$.

  Wir betrachten $x := \bar{x} + \lambda q$. Für $\lambda > 0$ klein genug gilt $x \in int(P)$. Wegen (4.4) gilt weiterhin
  $$q^t(\lambda q) = q^t(x - \bar{x}) \leq 0.$$
  Das aber bedeutet, dass $\lambda \|q\|^2 \leq 0$, was wegen $\lambda > 0$ zu $q = 0$ führt. Damit ergibt sich also $(q, \mu) = 0$, aber das ist ein Widerspruch zu $\tilde{q} \neq 0$.

Wegen $\mu < 0$ erhält man jetzt aus (4.4), dass alle $(x, z) \in \text{epi}(f)$

$$\frac{q^t}{|\mu|}(x - \bar{x}) + (-1)(z - f(\bar{x})) \leq 0$$

erfüllen, also gilt

$$\frac{q^t}{|\mu|}(x - \bar{x}) - z + f(\bar{x}) \leq 0 \text{ für alle } (x, z) \in \text{epi}(f).$$

Speziell für $\xi := \frac{1}{|\mu|}q^t$ gilt für alle $x \in P$ (also für $(x, f(x)) \in \text{epi}(f)$), dass

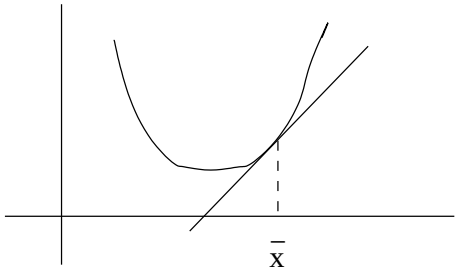
$$\xi(x - \bar{x}) + f(\bar{x}) \leq f(x),$$

also ist ξ Subgradient von f an $\bar{x}$. QED

Bemerkung: Im Fall von offenen, nichtleeren und konvexen Mengen P existiert also ein Subgradient ξ für alle $x \in P$. In diesem Fall kann man auch die Rückrichtung von Satz 4.22 zeigen:
Sei P offen, nichtleer und konvex und sei $f : P \to \mathbb{R}$. Wenn für alle $x \in P$ ein Subgradient von f existiert, dann ist f konvex.

Wir wollen als nächstes Subgradienten von differenzierbaren Funktionen untersuchen.

Satz 4.23 *Sei P eine konvexe und nichtleere Menge im $\mathbb{R}^n$ und sei $f : P \to \mathbb{R}$ konvex und differenzierbar an einem Punkt $\bar{x} \in int(P)$. Dann ist das Subdifferential von f an $\bar{x}$ gegeben durch $\{\nabla f(\bar{x})\}$, d.h. der einzige Subgradient von f an $\bar{x}$ ist der Gradient von f.*



einziger Subgradient ξ mit

$g(x) = f(\bar{x}) + \xi\,(x - \bar{x})$

ist Gradient $\xi = \nabla f(\bar{x})$

Beweis: Nach Satz 4.22 existiert ein Subgradient ξ an $\bar{x}$, d.h.

$$f(x) \geq f(\bar{x}) + \xi(x - \bar{x}) \text{ für alle } x \in P.$$

Wähle nun $x = \bar{x} + \lambda d$ für ein beliebiges $d \in \mathbb{R}^n \setminus \{0\}$. Ist $\lambda > 0$ klein genug, so ist $x \in P$ und

$$f(\bar{x} + \lambda d) \geq f(\bar{x}) + \lambda \xi d. \tag{4.5}$$

Da f an $\bar{x}$ differenzierbar ist, erhält man

$$f(\bar{x}+\lambda d) = f(\bar{x}) + \lambda(\nabla f(\bar{x})^t)d + \lambda\|d\|\alpha_{\bar{x}}(\lambda d)). \tag{4.6}$$

mit $\lim_{\lambda\to 0}\alpha_{\bar{x}}(\lambda d) = 0$.
Wir subtrahieren nun (4.5)-(4.6) und erhalten für kleine λ:

$$0 \geq \lambda(\xi - \nabla f(\bar{x}))d - \lambda\|d\|\alpha_{\bar{x}}(\lambda d),$$

oder, wegen $\lambda > 0$

$$0 \geq (\xi - \nabla f(\bar{x}))d - \|d\|\alpha_{\bar{x}}(\lambda d).$$

Da $\|d\|\alpha_{\bar{x}}(\lambda d) \to 0$ für $\lambda \to 0$ erhalten wir daraus $0 \geq (\xi - \nabla f(\bar{x}))d$ für alle $d \in \mathbb{R}^n \setminus \{0\}$. Mit $d = (\xi - \nabla f(\bar{x}))^t$ ergibt sich $0 \geq \|\xi - \nabla f(\bar{x})\|^2$, also $\xi = \nabla f(\bar{x})$. QED

Für offene Mengen erhalten wir daraus folgende Charakterisierung von Konvexität für differenzierbare Funktionen.

Korollar 4.24 *Sei $P \subseteq \mathbb{R}^n$ konvex, offen und nichtleer, und sei $f : P \to \mathbb{R}$ differenzierbar. Dann ist f konvex genau dann wenn*

$$f(x) \geq f(\bar{x}) + \nabla f(\bar{x})(x - \bar{x}) \text{ für alle } x, \bar{x} \in P.$$

Jetzt können wir folgende Charakterisierung von Minima durch Subgradienten beweisen.

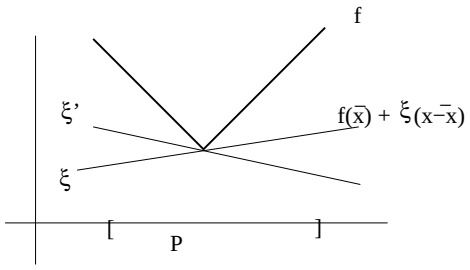
Satz 4.25 *Sei $f : \mathbb{R}^n \to \mathbb{R}$ konvex und sei $P \subseteq \mathbb{R}^n$ eine nichtleere, konvexe Menge. Betrachte das konvexe Optimierungsproblem*

$$\text{(P)} \quad \min\{f(x) : x \in P\}.$$

Dann ist $\bar{x}$ ein Minimum von (P) genau dann wenn f einen Subgradienten ξ an $\bar{x}$ hat mit

$$\xi(x - \bar{x}) \geq 0 \text{ für alle } x \in P.$$

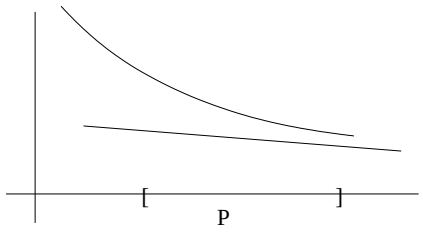
Wir verdeutlichen diese Aussage zunächst an den folgenden Skizzen.



$\xi' < 0 \Rightarrow$ für $x > \bar{x}$ gilt nicht

$\xi^t(x-\bar{x}) \geq 0$

$\xi = 0 \Rightarrow \xi^t(x-\bar{x}) \geq 0 \ \forall\ x \in P$



$\xi < 0$

$\forall\ x \in P \quad x \leq \bar{x}$

$\Rightarrow \xi^t(x-\bar{x}) \geq 0$

Beweis:

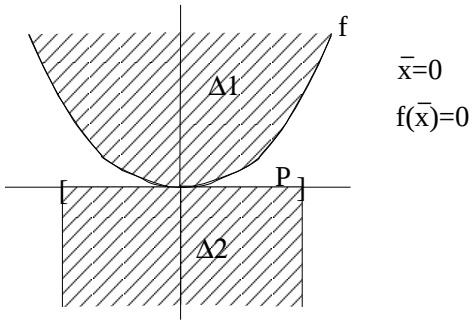
"⇐": Sei ξ Subgradient an $\bar{x}$ und $\xi(x-\bar{x}) \geq 0$ für alle $x \in P$. Weil ξ Subgradient gilt

$$f(x) \geq f(\bar{x}) + \underbrace{\xi(x-\bar{x})}_{\geq 0} \geq f(\bar{x}) \text{ für alle } x \in P,$$

also ist $\bar{x}$ (globales) Minimum.

"⇒": Sei $\bar{x}$ optimal. Definiere

$$\begin{aligned} \Delta_1 &= \{(x-\bar{x}, y) : x \in \mathbb{R}^n, y > f(x) - f(\bar{x})\} \subseteq \mathbb{R}^{n+1} \\ \Delta_2 &= \{(x-\bar{x}, y) : x \in P, y \leq 0\} \subseteq \mathbb{R}^{n+1} \end{aligned}$$



Es gilt:

- $\Delta_1,\ \Delta_2$ konvex
- $\Delta_1 \cap \Delta_2 = \emptyset$: Sonst gäbe es $(x-\bar{x}, y)$ mit $x \in P, f(x) - f(\bar{x}) < y \leq 0$. Das bedeutet aber $f(x) < f(\bar{x})$ für $x \in P$; ein Widerspruch, da $\bar{x}$ globales Minimum ist.

Aus dem Trennungssatz (Satz 4.19) folgt die Existenz einer Hyperebene $H \subseteq \mathbb{R}^{n+1}$, die Δ_1 von Δ_2 trennt, d.h. es existiert $(\xi_0, \mu) \neq 0$ und $\alpha \in \mathbb{R}$, so dass

$$\xi_0(x-\bar{x}) + \mu y \leq \alpha \text{ für alle } (x-\bar{x}, y) \in \Delta_1, \tag{4.7}$$

$$\xi_0(x-\bar{x}) + \mu y \geq \alpha \text{ für alle } (x-\bar{x}, y) \in \Delta_2. \tag{4.8}$$

Wir wollen zeigen, dass der gesuchte Subgradient $\xi = \frac{\xi_0}{-\mu}$ ist. Dazu brauchen wir zunächst die folgende Aussage.

Behauptung: $\mu < 0, \alpha = 0$.

- Wegen $(\bar{x}-\bar{x}, 0) \in \Delta_2$ gilt nach (4.8) $0 \geq \alpha$.

- Weiter ist $(\bar{x} - \bar{x}, p) \in \Delta_1$ für alle $p > 0$. Eingesetzt in (4.7) ergibt das $\mu p \leq \alpha$ für alle $p > 0$. Daraus folgt $\mu \leq 0$ und daraus wiederum $\alpha \geq 0$.
- Zusammen ergibt das $\mu \leq 0, \alpha = 0$.
- Es bleibt noch zu zeigen, dass $\mu \neq 0$. Angenommen, doch. Dann wähle $x = \bar{x} + \xi_0^t$. Mit y groß genug ist $(x - \bar{x}, y) \in \Delta_1$. Aus (4.7) ergibt sich

  $$\xi_0 \xi_0^t + 0 \leq \alpha = 0,$$

  also $\|\xi_0\|^2 \leq 0$ und entsprechend $\xi_0 = 0$. Das ist ein Widerspruch, weil $(\xi_0, \mu) \neq 0$.

Wir setzen nun wie geplant $\xi := \frac{1}{-\mu}\xi_0$ und müssen nachrechnen, dass ξ wirklich Subgradient ist. Dazu teilen wir (4.7) und (4.8) durch $(-\mu)$ und erhalten

$$\xi(x - \bar{x}) - y \leq 0 \text{ für alle } (x - \bar{x}, y) \in \Delta_1, \tag{4.9}$$

$$\xi(x - \bar{x}) - y \geq 0 \text{ für alle } (x - \bar{x}, y) \in \Delta_2. \tag{4.10}$$

Für $\varepsilon > 0$ gilt $y := f(x) - f(\bar{x}) + \varepsilon > f(x) - f(\bar{x})$, also ist $(x - \bar{x}, y) \in \Delta_1$ und aus (4.9) folgt

$$\xi(x - \bar{x}) \leq f(x) - f(\bar{x}) + \varepsilon \text{ für alle } \varepsilon > 0, x \in P.$$

Weil das für alle $\varepsilon > 0$ gilt folgt daraus $\xi(x - \bar{x}) \leq f(x) - f(\bar{x})$, also

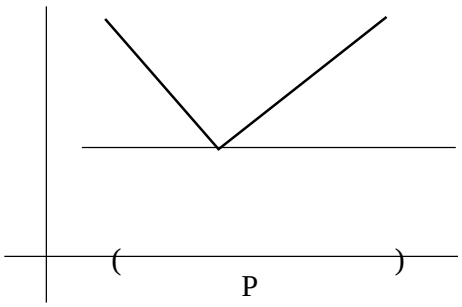
$$f(x) \geq \xi(x - \bar{x}) + f(\bar{x}) \text{ für alle } x \in P,$$

und ξ ist Subgradient.

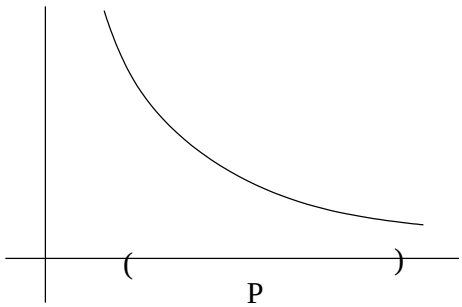
Die im Satz für alle $x \in P$ geforderte Bedingung $\xi(x - \bar{x}) \geq 0$ erhält man schließlich für $y = 0$ aus (4.10), weil $(x - \bar{x}, 0) \in \Delta_2$ für alle $x \in P$. QED

Korollar 4.26 *Sei $P \subseteq \mathbb{R}^n$ eine nichtleere, konvexe und offene Menge und sei $f : P \to \mathbb{R}$ konvex. Dann gilt: $\bar{x}$ ist minimal für $\min\{f(x) : x \in P\}$ genau dann wenn $\xi = 0$ ein Subgradient von f an $\bar{x}$ ist.*

Es können die folgenden zwei Fälle auftreten.



ξ =0 ist Subgradient



kein Min. ex.

Beweis: Wir wenden Satz 4.25 auf offene Mengen an.

"$\Leftarrow$": Sei $\xi = 0$ Subgradient von f an $\bar{x}$. Dann gilt $\xi(x - \bar{x}) \geq 0$ für alle $x \in P$, also folgt die Optimalität von $\bar{x}$ wegen Satz 4.25.

"$\Rightarrow$": Sei andererseits $\bar{x}$ optimal. Dann existiert ein Subgradient ξ an $\bar{x}$ mit

$$\xi(x - \bar{x}) \geq 0 \text{ für alle } x \in P \tag{4.11}$$

(Satz 4.25). Weil P offen ist, ist $x = \bar{x} - \lambda\xi^t \in P$ für ein hinreichend kleines $\lambda > 0$. Für dieses x ergibt (4.11):

$$\xi(-\lambda\xi^t) \geq 0,$$

woraus $-\lambda\|\xi\|^2 \geq 0$ und daher $\xi = 0$ folgt. QED

Korollar 4.27 *Sei $P \subseteq \mathbb{R}^n$ konvex und sei $f : P \to \mathbb{R}$ konvex und differenzierbar. Dann ist $\bar{x}$ optimal für*

$$\min\{f(x) : x \in P\}$$

genau dann wenn $(\nabla f(\bar{x}))(x - \bar{x}) \geq 0$ für alle $x \in P$.
Ist P außerdem offen, so ist $\bar{x}$ optimal genau dann wenn $\nabla f(\bar{x}) = 0$.

Wir wenden die vorhergehenden Ergebnisse für *differenzierbare* Funktionen an. Sei also f differenzierbar und konvex, P konvex. Dann:

1. Ist P offen, so gilt: $\bar{x}$ ist genau dann optimal wenn $\nabla f(\bar{x}) = 0$.

   Insbesondere heißt das, dass man zwei Fälle unterscheiden muss. Entweder gilt $\nabla f(\bar{x}) = 0$ für $\bar{x} \in P$, dann ist $\bar{x}$ optimal. Im anderen Fall ist $\nabla f(\bar{x}) \neq 0$ für alle $\bar{x} \in P$ und ein Minimum des konvexen Optimierungsproblemes (4.1) existiert nicht.

2. Ist P abgeschlossen, dann gilt: $\bar{x}$ ist genau dann optimal wenn

   $$(\nabla f(\bar{x}))(x - \bar{x}) \geq 0 \text{ für alle } x \in P.$$

   Das heißt, entweder ist $\nabla f(\bar{x}) = 0$ für $\bar{x} \in P$, dann ist $\bar{x}$ optimal, oder $\nabla f(x) \neq 0$ für alle $x \in P$, dann gibt es ein $\bar{x} \in \partial P$, das optimal ist.

Wir können nun das **Subgradientenverfahren** zur Minimierung einer konvexen Funktion über einer konvexen Menge beschreiben.

Man startet mit einer zulässigen Lösung $\bar{x} \in P$ und dem zugehörigen Subgradienten ξ. Um die Lösung zu verbessern würde man gerne

$$\bar{x}_{neu} := \bar{x} - \lambda \xi^t \tag{4.12}$$

mit $\lambda > 0$ definieren, ähnlich zu der Methode des steilsten Abstiegs. Die Motivation besteht also auch hier darin, den Gradienten für differenzierbare Funktionen durch einen Subgradienten zu ersetzen. Für den Gradienten gilt nämlich

$$f(\bar{x}_{neu}) = f(\bar{x} - \lambda \nabla(\bar{x})) \leq f(\bar{x}),$$

der Zielfunktionswert der neuen Lösung ist also besser als der der alten, siehe Lemma 4.36 (auf Seite 158 in Abschnitt 4.36). Allerdings wird in (4.12) meistens gelten, dass $\bar{x}_{neu} = \bar{x} - \lambda \xi^t \notin P$, der neue Wert also nicht zulässig ist. Um in jedem Iterationsschritt Zulässigkeit zu erreichen, projiziert man in einem weiteren Schritt $\bar{x}_{neu}$ auf P, das heißt, man wählt $x_{neu} \in P$ so, dass $\|x_{neu} - \bar{x}_{neu}\| = \|x_{neu} - (\bar{x} - \lambda \xi^t)\|$ möglichst klein wird.

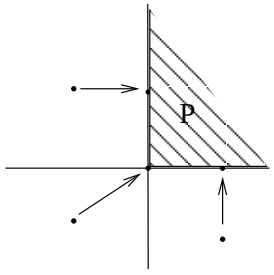
Notation 4.28 *Sei $P \subseteq \mathbb{R}^n$ und $y \in \mathbb{R}^n$. Die Projektion von y auf P ist definiert durch*

$$\mathcal{P}_P(y) := \operatorname{argmin}\{\|x - y\| : x \in P\}$$

Beispiel 4.3 *Ist $P = \{x \in \mathbb{R}^n : x \geq 0\}$, so gilt*

$$\mathcal{P}_P \begin{pmatrix} y_1 \\ \vdots \\ y_n \end{pmatrix} = \begin{pmatrix} \max\{y_1, 0\} \\ \vdots \\ \max\{y_n, 0\} \end{pmatrix},$$

also zum Beispiel $\mathcal{P}_P \begin{pmatrix} 1 \\ 0 \\ -2 \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}$



Der Subgradienten-Algorithmus wird formal wie folgt beschrieben.

Algorithmus: Subgradienten-Verfahren

Input: Startpunkt $x_1 \in P$, Folge $\lambda_1 \geq \lambda_2 \geq \cdots$ mit $\lambda_k \to 0$ und $\sum \lambda_k = \infty$.

Schritt 1: $UB_1 = f(x_1),\ x^* = x_1, k = 1$

Schritt 2: Sei ξ_k ein Subgradient von f an x_k.
wenn $\xi_k = 0$, STOP: x^* optimal.
sonst setze $d_k := \frac{-\xi_k^t}{\|\xi_k\|},\ \bar{x}_{k+1} := x_k + \lambda_k d_k,\ x_{k+1} := \mathcal{P}_P(\bar{x}_{k+1})$.

Schritt 3: Wenn $f(x_{k+1}) < UB_k$: $UB_{k+1} = f(x_{k+1}), x^* = x_{k+1}$.
sonst: $UB_{k+1} = UB_k$.

Schritt 4: $k \to k+1$, gehe zu Schritt 2.

Weil ξ_k aus dem Subdifferential beliebig gewählt wird, ist es unwahrscheinlich, gerade $\xi_k = 0$ zu treffen. Daher beendet man das Verfahren in der Praxis meistens nach einer vorgegebenen Anzahl an Iterationen oder falls $x_k \approx x_{k+1}$.

Satz 4.29 *Sei ein konvexes Optimierungsproblem* $\min\{f(x) : x \in P\}$ *mit konvexer Menge* $P \subseteq \mathbb{R}^n$ *und konvexer Funktion* $f : P \to \mathbb{R}$ *wie in (4.1) gegeben. Wenn ein Minimum* x^* *für das konvexe Optimierungsproblem existiert, dann endet das Subgradientenverfahren entweder nach endlichen vielen Schritten mit der Optimallösung oder es erzeugt eine Folge* (UB_k)*, die gegen* $f^* = f(x^*) = \min\{f(x) : x \in P\}$ *konvergiert.*

Beweis: Falls $\xi_k = 0$ in Schritt 2 des Subgradientenverfahrens erreicht wird, so gilt, dass $x_k = x^*$ nach Satz 4.25 optimal ist.
Ist das nicht der Fall, so entsteht eine Folge $\{UB_k\}$. Zunächst gilt $UB_k \geq UB_{k+1} \geq f^*$ für alle k, also ist die Folge monoton fallend und nach unten beschränkt. Die Folge ist also konvergent gegen einen Grenzwert

$$UB_k \to \overline{f}.$$

Wir möchten nun zeigen, dass $\overline{f} = f^*$. Angenommen, nein. Dann gilt $\overline{f} > \alpha > f^*$ für ein $\alpha \in \mathbb{R}$, also notieren wir

$$f(x_k) > \alpha \text{ für alle } k. \tag{4.13}$$

Wir wählen nun $\hat{x} \in P$ mit $f(\hat{x}) < \alpha$ (das existiert, z.B. ist $\hat{x} = x^*$ eine mögliche Wahl.) Es gilt also, dass $\hat{x} \in int(L_{\leq}(\alpha))$, es existiert sogar ein $\delta > 0$ so dass eine kleine Umgebung $N_\delta(\hat{x})$ um $\hat{x}$ ganz in der Niveaumenge liegt, also $N_\delta(\hat{x}) \subseteq L_{\leq}(\alpha)$. Damit erhalten wir

$$x_B := \hat{x} + \delta \frac{\xi_k^t}{\|\xi_k\|} \in L_{\leq}(\alpha), \tag{4.14}$$

wobei ξ_k der im Algorithmus gewählte Subgradient von f an x_k ist. Es gilt also

$$\begin{aligned}
& f(x_B) \geq f(x_k) + \xi_k(x_B - x_k) \\
\Rightarrow\ & \xi_k(x_B - x_k) \leq \underbrace{f(x_B)}_{\leq \alpha \text{ nach (4.14)}} - \underbrace{f(x_k)}_{> \alpha \text{ nach (4.13)}} < 0 \\
\Rightarrow\ & \xi_k \left(\hat{x} + \delta \frac{\xi_k^t}{\|\xi_k\|} - x_k \right) < 0 \\
\Rightarrow\ & \xi_k(\hat{x} - x_k) < -\delta \frac{\xi_k \xi_k^t}{\|\xi_k\|} = -\delta \|\xi_k\|.
\end{aligned}$$

Mit $d_k = \frac{-\xi_k^t}{\|\xi_k\|}$ folgt:

$$(x_k - \hat{x})^t d_k < -\delta \text{ für alle } k \tag{4.15}$$

Weil $x_{k+1} = \text{argmin}_{x \in P} \|\overline{x}_{k+1} - x\|$ gilt außerdem (wie im Beweis von Satz 4.18)

$$(\overline{x}_{k+1} - x_{k+1})^t (\hat{x} - x_{k+1}) \leq 0. \tag{4.16}$$

Daraus folgt

$$\begin{aligned}
\|\overline{x}_{k+1} - \hat{x}\|^2 &= \|(\overline{x}_{k+1} - x_{k+1}) + (x_{k+1} - \hat{x})\|^2 \\
&= \|\overline{x}_{k+1} - x_{k+1}\|^2 + \|x_{k+1} - \hat{x}\|^2 + 2 \underbrace{(\overline{x}_{k+1} - x_{k+1})^t (x_{k+1} - \hat{x})}_{\geq 0 \text{ nach (4.16)}} \\
&\geq \|x_{k+1} - \hat{x}\|^2
\end{aligned} \tag{4.17}$$

Wir nutzen diese Ungleichung, um $\|x_{k+1} - \hat{x}\|^2$ weiter nach oben abschätzen:

$$\begin{aligned}
\|x_{k+1} - \hat{x}\|^2 &\leq \|\overline{x}_{k+1} - \hat{x}\|^2 \\
&= \|x_k + \lambda_k d_k - \hat{x}\|^2 \\
&= \|x_k - \hat{x}\|^2 + \lambda_k^2 \|d_k\|^2 + 2\lambda_k \cdot \underbrace{d_k^t (x_k - \hat{x})}_{< -\delta \text{ nach (4.15)}} \\
&\leq \|x_k - \hat{x}\|^2 + \lambda_k^2 + 2\lambda_k(-\delta) \\
&= \|x_k - \hat{x}\|^2 + \lambda_k(\lambda_k - 2\delta).
\end{aligned}$$

Nach Voraussetzung geht $\lambda_k \to 0$, also existiert ein K, so dass für alle $k \geq K$ gilt: $\lambda_k < \delta$, oder auch $\lambda_k - 2\delta < -\delta$. Das setzen wir in unsere Abschätzung für $\|x_{k+1} - \hat{x}\|^2$ ein und erhalten

$$\|x_{k+1} - \hat{x}\|^2 \leq \|x_k - \hat{x}\|^2 - \lambda_k \delta.$$

Jetzt summieren wir auf beiden Seiten von $k = K$ bis $K + r$ für ein beliebiges $r \in \mathbb{N}$:

$$\begin{aligned}
\delta \sum_{k=K}^{K+r} \lambda_k &\leq \|x_K - \hat{x}\|^2 - \|x_{K+r+1} - \hat{x}\|^2 \\
&\leq \|x_K - \hat{x}\|^2,
\end{aligned}$$

und erhalten für $r \to \infty$ auf der linken Seite der Ungleichung nach Voraussetzung ∞, aber auf der rechten Seite $\|x_k - \hat{x}\|^2 < \infty$, ein Widerspruch. QED

4.3 Konkave Programmierung

In diesem Abschnitt betrachten wir nun die Maximierung einer konvexen Funktion,

$$\max\{f(x) : x \in P\} \text{ mit } f \text{ konvex.}$$

Dieses Problem kann in ein äquivalentes Minimierungsproblem

$$-\min\{-f(x) : x \in P\} \text{ mit } f \text{ konvex.}$$

transformiert werden, allerdings geht dabei die Konvexität der zu minimierenden Funktion $g(x) = -f(x)$ verloren: g ist konkav, wenn f konvex ist. Man spricht daher auch von *konkaver Programmierung.*

Im folgenden werden wir sehen, dass sich konkave und konvexe Programmierung grundlegend unterscheiden, sowohl bei der Charakterisierung der Minima als auch bei den Lösungsansätzen. Wir nennen den folgenden Satz ohne Beweis.

Satz 4.30 *Sei $P \subseteq \mathbb{R}^n$ nichtleer und konvex und sei $f : P \to \mathbb{R}$ eine konvexe Funktion. Sei $\bar{x}$ lokal optimal für $\max\{f(x) : x \in P\}$. Dann gilt für jeden Subgradienten ξ von f an $\overline{x}$:*

$$\xi(x - \overline{x}) \leq 0 \text{ für alle } x \in P.$$

Es muss beachtet werden, dass im Gegensatz zu dem analogen Kriterium in der konvexen Minimierung (Satz 4.25 auf Seite 4.25) die Rückrichtung von Satz 4.30 nicht gilt. Das deutet an, dass es schwieriger ist, eine konvexe Funktion zu maximieren als sie zu minimieren. Es gibt aber einen leicht lösbaren Fall, nämlich wenn der Rand der zulässigen Menge P "einfach" ist.

Satz 4.31 *Sei $f : \mathbb{R}^n \to \mathbb{R}$ eine konvexe Funktion und sei $P \subseteq \mathbb{R}^n$ konvex und kompakt. Dann existiert ein (globales) Maximum von*

$$\max\{f(x) : x \in P\}$$

und eine optimale Lösung ist gegeben durch einen Extremalpunkt von P.

Beweis: Zunächst wissen wir von Satz 4.13, dass f stetig auf dem $\mathbb{R}^n$ ist. Nach dem Satz von Weierstrass existiert auf der kompakten Menge P also ein Maximum.
Weiter benötigt man den *Repräsentations-Satz*, der besagt, dass man jeden Punkt x einer konvexen Menge P als endliche Konvexkombination von Extremalpunkten der Menge darstellen kann, also

$$x = \sum_{j=1}^{k} \lambda_j x_j \text{ mit } \lambda_j \geq 0, \sum_{j=1}^{k} \lambda_j = 1$$

und alle $x_j, j = 1, \ldots, k$ sind Extremalpunkte von P. Sei nun $x \in P$. Wir benutzen den Repräsentationssatz und schreiben x als Konvexkombination $x = \sum_{j=1}^{k} \lambda_j x_j$ mit Extremalpunkten x_j der Menge P. Es ergibt sich

$$\begin{aligned} f(x) &= f(\sum_{j=1}^{k} \lambda_j x_j) \\ &\leq \sum_{j=1}^{k} \lambda_j f(x_j) \text{ weil } f \text{ konvex, (Übung!)} \\ &\leq \sum_{j=1}^{k} \lambda_j f(x') \text{ mit } f(x') = \max\{f(x_j), j = 1, \ldots, k\} \\ &= f(x'), \end{aligned}$$

es gibt also zu jedem Punkt $x \in P$ einen besseren Extremalpunkt von P. QED

Der Satz hat besondere Bedeutung, wenn P nur endlich viele (am besten wenige) Extremalpunkte hat, die man leicht durchprobieren kann. Das ist der Fall für polyedrische Mengen P.

Die Eigenschaft "es gibt einen Extremalpunkt, der optimal ist" gilt auch noch für *pseudokonvexe* und stetige *quasikonvexe* Funktionen. Diese sind wie folgt definiert.

Definition 4.32 *Sei $P \subseteq \mathbb{R}^n$ eine konvexe Menge und sei $f : P \to \mathbb{R}$.*

- *Man nennt f* **quasikonvex**, *falls für alle $x_1, x_2 \in P$ und für alle $\lambda \in (0, 1)$ das folgende gilt:*

$$f(\lambda x_1 + (1 - \lambda)x_2) \leq \max\{f(x_1), f(x_2)\}.$$

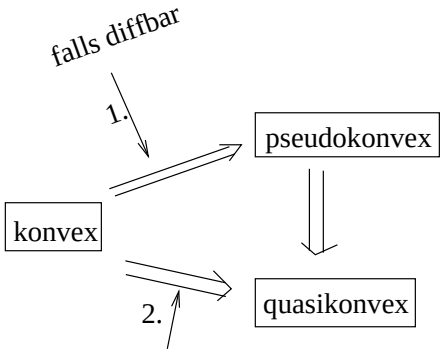
- *Sei f zusätzlich differenzierbar. Dann ist f* **pseudokonvex**, *falls für alle $x_1, x_2 \in P$ mit $\nabla f(x_1)(x_2 - x_1) \geq 0$ gilt*

$$f(x_2) \geq f(x_1).$$

Den Zusammenhang zu Konvexität klärt das folgende Lemma.

Lemma 4.33 *Sei $P \subseteq \mathbb{R}^n$ konvex und $f : P \to \mathbb{R}$ konvex. Dann gilt:*

1. *Ist f differenzierbar, so ist f pseudokonvex.*
2. *f ist quasikonvex.*
3. *Die Umkehrung der beiden Aussagen gilt nicht.*



Beweis:

ad 1. Sei f konvex und differenzierbar. Angenommen, $\nabla f(x_1)(x_2 - x_1) \geq 0$ aber $f(x_1) > f(x_2)$. Weil $\nabla f(x_1)$ Subgradient von f an x_1 ist, gilt

$$f(x_1) > f(x_2) \geq f(x_1) + \nabla f(x_1)(x_2 - x_1) \geq f(x_1),$$

ein Widerspruch. Also ist f pseudokonvex.

ad 2. Sei nun f konvex. Betrachte $x_1, x_2 \in P$, $\lambda \in (0, 1)$. Wegen der Konvexität von f gilt

$$f(\lambda x_1 + (1 - \lambda)x_2) \leq \lambda f(x_1) + (1 - \lambda)f(x_2) \leq \max\{f(x_1), f(x_2)\}.$$

Folglich ist f quasikonvex.

ad 3. Hier lassen sich leicht Gegenbeispiele finden (Übung). QED

Es gelten noch weitere Aussagen, die wir ohne Beweise nennen:

- Ist f pseudokonvex, so ist f quasikonvex.
- Sei $P \subseteq \mathbb{R}$. Ist $f : P \to \mathbb{R}$ monoton, dann ist f quasikonvex.
- Ist f differenzierbar, so gilt: f ist quasikonvex genau dann wenn $\nabla f(x_1)(x_2 - x_1) \leq 0$ für alle $x_1, x_2 \in P$ mit $f(x_1) \geq f(x_2)$.

Wir bemerken, dass für pseudokonvexe Funktionen auch viele der Aussagen aus dem Abschnitt über konvexe Programmierung gelten. So ist z.B. bei pseudokonvexen Funktionen jedes lokale Minimum ein globales Minimum, und gilt $\nabla f(x) = 0$ so ist x ein (globales) Minimum. Diese beiden Aussagen gelten für quasikonvexe Funktionen im allgemeinen nicht.

4.4 Restringierte Minimierung differenzierbarer Funktionen

Möchte man eine differenzierbare Funktion minimieren, ohne dass Nebenbedingungen zu beachten sind, so ist das folgende notwendige Kriterium bekannt:

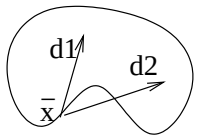
$$x \text{ lokales Minimum } \Longrightarrow \nabla f(x) = 0.$$

In diesem Abschnitt verallgemeinern wir diese Optimalitätsbedingung für die Minimierung differenzierbarer Funktionen unter (differenzierbaren) Nebenbedingungen. Wir betrachten zunächst das folgende allgemeine nichtlineare Programm

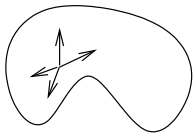
$$\min\{f(x) : x \in P\}. \tag{4.18}$$

Notation 4.34 *Sei $P \subseteq \mathbb{R}^n, P \neq \emptyset$ und sei $\bar{x} \in cl(P)$. Der Kegel der* **zulässigen Richtungen** *von P an $\bar{x}$ ist*

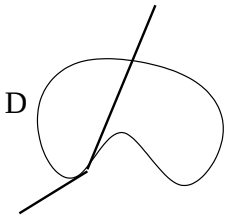
$$D = \{d \in \mathbb{R}^n \backslash \{0\} : \text{ Es existiert ein } \delta > 0 \text{ so dass } \bar{x} + \lambda d \in P \text{ für alle } \lambda \in (0, \delta)\}.$$



d1: zulässige Richtung
d2: nicht zul. Richtung



x ∈ int(P)
D = $\mathbb{R}^n \backslash \{0\}$



D Kegel

Notation 4.35 *Der Kegel der* **verbessernden Richtungen** *an $\bar{x}$ ist*

$$F = \{d \in \mathbb{R}^n \setminus \{0\} \quad : \quad \text{Es existiert ein } \delta > 0 \text{ so dass } f(\bar{x} + \lambda d) < f(\bar{x}) \text{ für alle } \lambda \in (0, \delta)\}.$$

Der Name *verbessernde Richtung* bezieht sich auf das Optimierungsproblem (4.18): Wenn man sich von $\bar{x}$ aus in eine Richtung $d \in F$ bewegt, verbessert sich der Zielfunktionswert. Verbessernde Richtungen kann man mit Hilfe des folgenden Kriteriums erkennen.

Lemma 4.36 *Sei f differenzierbar an $\bar{x}$ und sei $d \in \mathbb{R}^n$. Dann gilt:*

$$\nabla f(\bar{x}) d < 0 \Rightarrow d \in F,$$

insbesonders ist also $-\nabla f(\bar{x}) \in F$.

Beweis: f ist differenzierbar an $\bar{x}$, also gilt

$$f(\bar{x}+\lambda d) = f(\bar{x}) + \lambda \nabla f(\bar{x}) d + \lambda \|d\| \alpha_{\bar{x}}(\lambda d)$$

mit $\alpha_{\bar{x}}(\lambda d) \to 0$ für $\lambda \to 0$. Division durch $\lambda \neq 0$ ergibt:

$$\frac{f(\overline{x}+\lambda d) - f(\overline{x})}{\lambda} = \underbrace{\nabla f(\bar{x}) d}_{<0 \text{ nach Vor.}} + \underbrace{\|d\| \alpha_{\overline{x}}(\lambda d)}_{\to 0 \text{ für } \lambda \to 0}.$$

Also existiert ein $\delta > 0$ so dass

$$\frac{f(\bar{x}+\lambda d) - f(\bar{x})}{\lambda} < 0 \text{ für alle } 0 < \lambda < \delta,$$

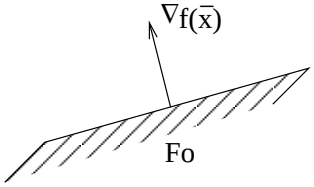
somit gilt $f(\bar{x}+\lambda d) < f(\bar{x})$ für alle $0 < \lambda < \delta$ und entsprechend $d \in F$. QED

Man macht sich leicht klar, dass aus $\nabla f(\bar{x}) d > 0$ folgt, dass $d \notin F$. Dagegen lässt $\nabla f(\bar{x}) d = 0$ keine weiteren Schlüsse zu.

Wir definieren

$$F_0 := \{d \in \mathbb{R}^n : \nabla f(\bar{x}) d < 0\}. \tag{4.19}$$

Wegen Lemma 4.36 gilt dann $F_0 \subseteq F$. Der Vorteil der Menge F_0 besteht darin, dass $d \in F_0$ viel einfacher zu überprüfen ist als $d \in F$.



Mit Hilfe von F_0 und D erhält man folgende notwendige Bedingung für lokale Optimalität.

Satz 4.37 *Sei $P \subseteq \mathbb{R}^n$ nichtleer und sei $f : \mathbb{R}^n \to \mathbb{R}$ differenzierbar an $\bar{x} \in P$. Dann gilt:*

$$\bar{x} \text{ lokales Minimum für (4.18)} \Rightarrow F_0 \cap D = \emptyset.$$

Beweis: Sei $\bar{x}$ ein lokales Minimum von (4.18), d.h. es existiert $\varepsilon > 0$ so dass für alle $x \in N_\varepsilon(\bar{x}) \cap P$ gilt:

$$f(x) \geq f(\bar{x}). \tag{4.20}$$

Angenommen, $F_0 \cap D \neq \emptyset$. Dann wähle eine Richtung $d \in F_0 \cap D$ mit $\|d\| = 1$. Es gilt:

- $d \in F_0$, also existiert $\delta_1 > 0$ so dass $f(\bar{x}+\lambda d) < f(\bar{x})$ für alle $\lambda \in (0, \delta_1)$.

- $d \in D$, also existiert $\delta_2 > 0$ so dass $\bar{x} + \lambda d \in P$ für alle $\lambda \in (0, \delta_2)$.

Wir wählen $0 < \lambda \leq \min\{\varepsilon, \delta_1, \delta_2\}$ und $x := \bar{x} + \lambda d$. Dann folgt $f(x) < f(\bar{x})$ (weil $\lambda < \delta_1$) und $x \in N_\varepsilon(\bar{x}) \cap P$ (weil $\lambda < \delta_2$ und $\lambda < \varepsilon$). Das ist aber ein Widerspruch zu (4.20). QED

Die Idee der folgenden Ergebnisse besteht darin, dieses geometrische Kriterium zu nutzen, um zu einem algebraischen Kriterium zu gelangen.

Dazu betrachten wir zunächst Probleme mit Ungleichungen als Nebenbedingungen, also Probleme der folgenden Form:

$$\begin{aligned} &\min f(x) \\ &\text{s.d. } g_i(x) \leq 0 \text{ für alle } i = 1, \ldots, m, \end{aligned} \tag{4.21}$$

das heißt, die zulässige Menge P lässt sich schreiben als

$$P = \{x \in \mathbb{R}^n : g_i(x) \leq 0 \text{ für alle } i = 1, \ldots, m\}.$$

Wir führen eine weitere Bezeichnung ein.

Notation 4.38 *Sei $\bar{x} \in P$. Dann heißt*

$$I(\bar{x}) = \{i \in \{1, \ldots, m\} : g_i(\bar{x}) = 0\}$$

die Menge der **bindenden Nebenbedingungen** *an $\bar{x}$. (Ist klar, auf welches $\bar{x}$ sich die Menge bezieht, schreiben wir auch einfach I.)*

Sei $\bar{x} \in P$ und sei g_i differenzierbar an $\bar{x}$ für alle $i \in I(\bar{x})$ und zumindest stetig an $\bar{x}$ für alle $i \notin I(\bar{x})$. Dann definieren wir

$$D_0 := \{d \in \mathbb{R}^n : \nabla g_i(\bar{x}) d < 0 \text{ für alle } i \in I(\bar{x})\}. \tag{4.22}$$

Ähnlich zu der Aussage $F_0 \subseteq F$ zeigen wir nun, dass $D_0 \subseteq D$, um im folgenden D_0 als Approximation von D heranziehen zu können.

Lemma 4.39 *$D_0 \subseteq D$.*

Beweis: Sei $d \in D_0$. Wir möchten zeigen, dass es ein $\delta > 0$ gibt, so dass $\bar{x} + \lambda d \in P$ für alle $\lambda \in (0, \delta)$. Dazu verwenden wir

$$\bar{x} + \lambda d \in P \Longleftrightarrow g_i(\bar{x} + \lambda d) \leq 0, i = 1, \ldots, m,$$

und analysieren die folgenden beiden Fälle. Sei dazu $I = I(\bar{x})$.

$i \notin I$**:** Dann gilt $g_i(\bar{x}) < 0$. Wegen der Stetigkeit von g_i existiert $\delta_1 > 0$ so dass

$$g_i(\bar{x} + \lambda d) < 0 \text{ für alle } \lambda \in (0, \delta_1).$$

$i \in I$**:** Dann ist g_i differenzierbar an $\bar{x}$ und es gilt wegen $d \in D_0$, dass $\nabla g_i(\bar{x})d < 0$. Also existiert $\delta_2 > 0$ so dass analog zu Lemma 4.36

$$g_i(\bar{x} + \lambda d) < g_i(\bar{x}) \leq 0 \text{ für alle } \lambda \in (0, \delta_2).$$

Für $\delta \leq \min\{\delta_1, \delta_2\}$ gilt entsprechend

$$\bar{x} + \lambda d \in P \text{ für alle } \lambda \in (0, \delta),$$

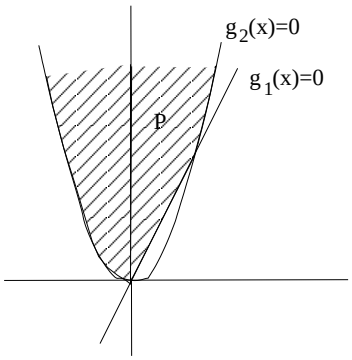
also ist $d \in D$. QED

Beispiel 4.4 *Wir betrachten*

$$\begin{aligned} g_1(x) &= 2x_1 - x_2, \\ g_2(x) &= x_1^2 - x_2 \text{ und} \\ P &= \{x \in \mathbb{R}^2 : g_1(x) \leq 0 \text{ und } g_2(x) \leq 0\}. \end{aligned}$$

Die Gradienten ergeben sich als

$$\nabla g_1(x) = (2, -1) \text{ und } \nabla g_2(x) = (2x_1, -1).$$



Wir bilden die Mengen D, D_0 in drei verschiedenen Punkten:

1. $\bar{x} = \begin{pmatrix} -1 \\ 2 \end{pmatrix}$. *Dann ist die Menge der bindenden Ungleichungen in $\bar{x}$ leer, d.h. $I(\bar{x}) = \emptyset$. Es folgt dass $D = D_0 = \mathbb{R}^2$.*

2. $\bar{x} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$. *In diesem Fall erhält man $I(\bar{x}) = \{1, 2\}$ und $\nabla g_2(\bar{x}) = (0, -1)$. Es ergibt sich*

$$\begin{aligned} D &= \{d \in \mathbb{R}^2 : 2d_1 - d_2 \leq 0 \text{ und } d_2 > 0\} \\ D_0 &= \{d \in \mathbb{R}^2 : 2d_1 - d_2 < 0 \text{ und } d_2 > 0\}. \end{aligned}$$

3. $\bar{x} = \begin{pmatrix} -1 \\ 1 \end{pmatrix}$. *Hier ist* $I(\bar{x}) = \{2\}$ *und* $\nabla g_2(\bar{x}) = (-2, -1)$. *Wir erhalten*

$$D_0 = \{d \in \mathbb{R}^2 : -2d_1 - d_2 < 0\} = D.$$

Satz 4.40 *Sei* $f : \mathbb{R}^n \to \mathbb{R}$ *und* $g_i : \mathbb{R}^n \to \mathbb{R}$ *für* $i = 1, \ldots, m$. *Sei* $\bar{x}$ *so, dass* $g_i(\bar{x}) \leq 0$ *für alle* $i = 1, \ldots, m$. *Weiter seien* f *und* g_i *differenzierbar an* $\bar{x}$ *für alle* $i \in I(\bar{x})$ *und stetig an* $\bar{x}$ *für alle* $i \notin I(\bar{x})$. *Dann gilt:*

$$\bar{x} \text{ lokales Minimum für (4.21)} \Rightarrow F_0 \cap D_0 = \emptyset.$$

Beweis: Sei $\bar{x}$ ein lokales Minimum von (4.21). Nach Satz 4.37 gilt dann $F_0 \cap D = \emptyset$. Wegen $D_0 \subseteq D$ folgt $F_0 \cap D_0 \subseteq F_0 \cap D = \emptyset$. QED

Bemerkung: Sei zusätzlich f konvex und alle g_i streng konvex auf $N_\varepsilon(\bar{x})$. Dann gilt sogar:

$$\bar{x} \text{ lokales Minimum von (4.21)} \Leftrightarrow F_0 \cap D_0 = \emptyset.$$

Wir können nun das geometrische Kriterium aus dem letzten Satz in ein algebraisches Kriterium umformulieren. Zum Beweis der Aussage benötigen wir den Satz von Gordan, den wir bereits in der linearen Optimierung in Zusammenhang mit den Alternativsätzen kennen gelernt haben.

Satz 4.41 (Fritz John Bedingungen) *Sei* $P = \{x \in \mathbb{R}^n : g_i(x) \leq 0, i = 1, \ldots, m\}$ *und sei* $\bar{x} \in P$. *Sei* $I = I(\bar{x})$ *und seien en* f, g_i *differenzierbar an* $\bar{x}$ *für alle* $i \in I$ *und stetig an* $\bar{x}$ *für alle* $i \notin I$.
Sei $\bar{x}$ *lokales Minimum von (4.21). Dann gilt:*

1. *Es existieren Skalare* u_0, u_i *für alle* $i \in I$ *so dass*

$$\begin{aligned} u_0 \nabla f(\bar{x}) + \sum_{i \in I} u_i \nabla g_i(\bar{x}) &= 0 \\ (u_0, u_I) &\neq 0 \\ (u_0, u_I) &\geq 0, \end{aligned}$$

   wobei $u_I = (u_i | i \in I)$.

2. *Seien zusätzlich alle* g_i *differenzierbar an* $\bar{x}$. *Dann existieren Skalare* $u_0, u_i, i = 1, \ldots, m$, *so dass*

$$\begin{aligned} u_0 \nabla f(\bar{x}) + \sum_{i=1}^{m} u_i \nabla g_i(\bar{x}) &= 0 \\ u_i g_i(\bar{x}) &= 0, \text{ für } i = 1, \ldots, m \\ u_0, u_i &\geq 0 \\ (u_0, u_1, \ldots, u_m) &\neq 0. \end{aligned}$$

Beweis:

ad 1. Wir definieren A als eine $(|I|+1) \times n-$ Matrix, die den Gradienten der Zielfunktion in ihrer ersten Zeile und die Gradienten $\nabla g_i(\bar{x})$ für alle $i \in I$ in ihren weiteren Zeilen enthält, d.h.

$$A = \begin{pmatrix} \nabla f(\bar{x}) \\ \vdots \\ \nabla g_i(\bar{x}) \\ \vdots \end{pmatrix}.$$

Sei nun $\bar{x}$ ein lokales Minimum von (4.21). Wegen Satz 4.40 folgt $F_0 \cap D_0 = \emptyset$. Daher existiert kein $d \in \mathbb{R}^n$, so dass $\nabla f(\bar{x})d < 0$ und $\nabla g_i(\bar{x})d < 0$ für alle $i \in I$. Es existiert also kein $d \in \mathbb{R}^n$ so dass $Ad < 0$.

Nach dem Satz von Gordan (Satz 2.38 auf Seite 67) folgt: Es gibt $p \in \mathbb{R}^{|I|+1}$, so dass

$$\begin{aligned} A^t p &= 0 \\ p &\geq 0 \\ p &\neq 0. \end{aligned}$$

Definiere die gesuchten Skalare als die Komponenten von P, d.h. $(u_0, \ldots, u_i, \ldots) := p^t$. Dann gilt

$$0 = p^t A = (u_0, u_I) \begin{pmatrix} \nabla f(\bar{x}) \\ \vdots \\ \nabla g_i(\bar{x}) \\ \vdots \end{pmatrix} = u_0 \nabla f(\bar{x}) + \sum_{i \in I} u_i \nabla g_i(\bar{x}).$$

ad 2. Der zweite Teil des Satzes folgt direkt, wenn man $u_i = 0 \; \forall i \notin I$ setzt. QED

Analog zu den Begriffen in der linearen Optimierung fasst man die Fritz-John-Bedingungen (für differenzierbare Funktionen $g_i, i = 1, \ldots, m$) folgendermaßen zusammen:

primale Zulässigkeit:

$$g_i(\bar{x}) \leq 0, \; i = 1, \ldots, m \tag{4.23}$$

duale Zulässigkeit:

$$\begin{aligned} u_0, u_i &\geq 0, \; i = 1, \ldots, m \\ (u_0, u_1, \ldots, u_m) &\neq 0 \\ u_0 \nabla f(\bar{x}) + \sum_{i=1}^{m} u_i \nabla g_i(\bar{x}) &= 0 \end{aligned} \tag{4.24}$$

komplementärer Schlupf:

$$u_i g_i(\bar{x}) = 0, i = 1, \ldots, m \tag{4.25}$$

Die folgenden Bezeichnungen werden häufig verwendet.

Notation 4.42 *Die Skalare $u_0, u_1, \ldots, u_m$ aus Satz 4.41 nennt man* **Lagrange-Multiplikatoren**. *Ein Punkt $x \in P$ heißt* **FJ-Punkt** *(oder* **Fritz-John-Punkt***), falls Skalare $u_0, u_i, i = 1, \ldots, m$ existieren, die (4.23), (4.24) und (4.25) erfüllen.*

Mit diesen Begriffen können wir Satz 4.41 für differenzierbare Funktionen g_o, $i = 1 \ldots, m$ kurz fassen: Es gilt

$$x \text{ lokales Minimum von } (4.21) \Rightarrow x \text{ ist FJ-Punkt.}$$

Die Rückrichtung dieser Aussage gilt nicht. Wir untersuchen zunächst FJ-Punkte an einigen Beispielen.

Beispiel 4.5 .

$$\begin{array}{ll} \min & -x_1 \\ \text{s.d.} & x_1 + x_2 - 1 \leq 0 \\ & -x_2 \leq 0 \end{array}$$

Die Gradienten an $(x_1, x_2)^t$ ergeben sich als

$$\begin{aligned} \nabla f(x) &= (-1, 0), \\ \nabla g_1(x) &= (1, 1), \\ \nabla g_2(x) &= (0, -1). \end{aligned}$$

Daraus erhält man die folgenden FJ-Bedingungen:

$$\begin{aligned} x_1 + x_2 - 1 &\leq 0 \\ -x_2 &\leq 0 \\ u_0 &\geq 0 \\ u_1 &\geq 0 \\ u_2 &\geq 0 \\ (u_0, u_1, u_2) &\neq 0 \\ -u_0 + u_1 &= 0 \\ u_1 - u_2 &= 0 \\ u_1(x_1 + x_2 - 1) &= 0 \\ u_2(-x_2) &= 0 \end{aligned}$$

Aus den Bedingungen folgt, dass $u_0 = u_1 = u_2$, also $u_i > 0, i = 0, 1, 2$. Daraus ergibt sich $x_2 = 0$ und die Bedingung $x_1 + x_2 - 1 = 0$, die wiederum zu $x_1 = 1, x_2 = 0$ als einzigem FJ-Punkt und der Optimallösung führt.

Allerdings kann man nicht erwarten, so einfach wie im letzten Beispiel mit Hilfe der FJ-Punkte eine Optimallösung zu finden. Dazu geben wir ein paar Beispiele.

Beispiel 4.6 Alle Punkte sind FJ-Punkte: *Wir betrachten*

$$\begin{array}{rcl} \min f(x) & & \\ s.d.\ g(x) & \leq & 0 \\ -g(x) & \leq & 0 \end{array}$$

Die FJ-Bedingungen in diesem Beispiel sind

$$\begin{array}{rcl} g(x) & = & 0 \\ u_0\nabla f(x) + u_1\nabla g(x) - u_2\nabla g(x) & = & 0 \\ u_1 g(x) & = & 0 \\ -u_2 g(x) & = & 0 \\ u_0, u_1, u_2 & \geq & 0 \\ (u_0, u_1, u_2) & \neq & 0 \end{array}$$

In diesem Beispiel sind also unabhängig von der Zielfunktion **alle** *zulässigen Punkte x FJ-Punkte, nämlich zu den Lagrange-Multiplikatoren $u_0 = 0, u_1 = u_2 = \alpha > 0$.*

Unabhängigkeit von der Zielfunktion: *In dem folgenden Beispiel spielt die Zielfunktion ebenfalls keine Rolle für die Eigenschaft, ein FJ-Punkt zu sein: Sei $x \in P$ und $\nabla g_i(x) = 0$ für eine bindende Bedingung $i \in I$. Dann ist x für jede Zielfunktion f ein FJ-Punkt, nämlich mit $u_i > 0$ und $u_0, u_j = 0$ für alle $j \neq i$.*

Unabhängigkeit von den Nebenbedingungen: *Hier betrachten wir den Fall $\nabla f(x) = 0$. Dann ist mit $u_0 > 0, u_i = 0, i = 1, \ldots, m$ jeder zulässige Punkt FJ-Punkt, unabhängig von den Nebenbedingungen.*

Während im letzten Beispiel die Zielfunktion konstant ist (und daher auch alle Punkte lokal optimal), ist die Unabhängigkeit von f in den ersten beiden Beispielen sicher nicht wünschenswert. Das Problem besteht darin, dass die Gradienten der Nebenbedingungen linear abhängig sind. Damit kann es sogar bei linearen Programmen auftreten, wie unser letztes Beispiel demonstriert.

Beispiel 4.7 *Wir betrachten das folgende lineare Optimierungsproblem*

$$\begin{aligned} \min\ & c^t x \\ s.d.\ x_1 + x_2 - 1 &\leq 0 \\ -x_2 + 1 &\leq 0 \\ -x_1 &\leq 0 \\ -x_2 &\leq 0 \end{aligned}$$

Die Gradienten der g_i sind:

$$\begin{aligned} \nabla g_1(x) &= (1,1), \\ \nabla g_2(x) &= (0,-1), \\ \nabla g_3(x) &= (-1,0), \\ \nabla g_4(x) &= (0,-1) \end{aligned}$$

Für $x = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$ erhält man $I = \{1,2,3\}$ als Menge der bindenden Ungleichungen. Setzt man nun $u_0 = 0$, $u_1 = 1$, $u_2 = 1$, $u_3 = 1$ und $u_4 = 0$ so erhält man

$$u_1(1,1) + u_2(0,1) + u_3(-1,0) = 0,$$

also ist $\begin{pmatrix} 0 \\ 1 \end{pmatrix}$ FJ-Punkt, und auch hier ist das unabhängig von der Zielfunktion. Der Grund liegt darin, dass die Gradienten der bindenden Ungleichungen in dem untersuchten Punkt x linear abhängig sind.

Als Ausweg aus diesen Schwierigkeiten geht man den folgenden Weg: Man verlangt $u_0 > 0$. Das erreicht man, indem man z.B. $u_0 = 1$ setzt. Allerdings erkauft man sich diese Bedingung damit, dass das daraus resultierende Kriterium im allgemeinen nicht mehr notwendig für lokale Optima ist. Um das Ergebnis des Satzes 4.41 zu übertragen, benötigt man daher eine zusätzliche Bedingung. So eine Bedingung wird *constraint qualification* genannt.

Satz 4.43 (Karush-Kuhn-Tucker Bedingungen) *Sei $P = \{x \in \mathbb{R}^n : g_i(x) \leq 0, i = 1,\ldots,m\}$ und sei $\bar{x} \in P$. Seien f, g_i differenzierbar an $\bar{x}$ für alle $i \in I$ und stetig an $\bar{x}$ für alle $i \notin I$. Sei weiterhin $\{\nabla g_i(\bar{x}) : i \in I\}$ linear unabhängig.*
Sei $\bar{x}$ lokales Minimum von (4.21). Dann gilt:

1. *Es existieren Skalare u_i für alle $i \in I$ so dass*

$$\begin{aligned} \nabla f(\bar{x}) + \sum_{i \in I} u_i \nabla g_i(\bar{x}) &= 0 \\ u_i &\geq 0 \text{ für alle } i \in I. \end{aligned}$$

2. *Seien zusätzlich alle g_i differenzierbar an $\bar{x}$. Dann existieren Skalare $u_i, i = 1, \ldots, m$, so dass*

$$\begin{aligned} \nabla f(\bar{x}) + \sum_{i=1}^{m} u_i \nabla g_i(\bar{x}) &= 0 \\ u_i g_i(\bar{x}) &= 0, \text{ für } i = 1, \ldots, m \\ u_i &\geq 0 \end{aligned}$$

Beweis: Sei $\bar{x} \in P$ ein lokales Minimum von (4.21), das heißt $g_i(\bar{x}) \leq 0$ für alle $i = 1, \ldots, m$. Satz 4.41 liefert die Existenz von Lagrange-Multiplikatoren $u_0, \hat{u} = (\hat{u}_i, i \in I)$, so dass

$$\begin{aligned} (u_0, \hat{u}) &\neq 0 \\ u_0 &\geq 0 \\ \hat{u}_i &\geq 0 \text{ und} \\ u_0 \nabla f(\bar{x}) + \sum_{i \in I} \hat{u}_i \nabla g_i(\bar{x}) &= 0 \end{aligned}$$

Wir zeigen nun, dass $u_0 > 0$. Angenommen, nein. Dann wäre

$$\sum_{i \in I} \hat{u}_i \nabla g_i(\bar{x}) = 0$$

für Skalare $\hat{u}_i, i \in I$, die nicht alle gleich Null sind, also wäre $\{\nabla g_i(\bar{x}) : i \in I\}$ linear abhängig, ein Widerspruch.

Weil $u_0 > 0$ definieren wir

$$u_i = \frac{\hat{u}_i}{u_0} \text{ für alle } i \in I$$

und erhalten damit den ersten Teil des Satzes. Der zweite Teil folgt wie im Beweis von Satz 4.41 durch die Ergänzung $u_i := 0 \; \forall i \notin I$. QED

Notation 4.44

- *Man nennt x einen KKT-Punkt, wenn x zulässig ist und die KKT-Bedingungen aus Satz 4.43 erfüllt.*
- *Genau wie bei den FJ-Bedingungen spricht man auch bei den KKT-Bedingungen von primaler Zulässigkeit, dualer Zulässigkeit und komplementärem Schlupf.*

Wir schließen zwei Bemerkungen an.

- Jeder KKT-Punkt ist auch ein FJ-Punkt. Wie man an dem Punkt $x = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$ in Beispiel 4.7 sieht, gilt die Umkehrung im allgemeinen nicht.

- In der linearen Programmierung entsprechen die KKT-Bedingungen genau der dort definierten primalen beziehungsweise dualen Zulässigkeit und der Bedingung vom komplementären Schlupf. Man beachte, dass die KKT-Bedingungen bei linearen Programmen nach Satz 2.34 über die starke Dualität sogar hinreichend für Optimalität sind!

Nun wollen wir auf die Frage eingehen, wann denn ein lokales Minimum x ein KKT-Punkt ist. Von Satz 4.43 wissen wir bereits, dass diese Aussage gilt, falls $\{\nabla g_i(x) : i \in I\}$ linear unabhängig ist. Es gibt aber noch andere Bedingungen, aus denen man

$$x \text{ lokales Minimum für (4.21)} \Longrightarrow x \text{ KKT-Punkt} \tag{4.26}$$

folgern kann. Man nennt solche Bedingungen "constraint qualification". Beispiele hierfür sind:

- (4.26) ist erfüllt, falls $g_i(x)$ linear $\forall i$.

- (4.26) ist erfüllt, falls $g_i(x)$ konvex $\forall i$ und $int(\{x \in \mathbb{R}^n : g_i(x) \leq 0\}) \neq \emptyset$.

Das rechtfertigt nun zumindest, warum in der linearen Programmierung jedes (lokale) Optimum ein KKT-Punkt ist.
Betrachten wir abschließend noch ein nichtlineares Problem, in dem nicht nur Ungleichungen sondern auch Gleichungen als Nebenbedingungen auftreten, also

$$\begin{array}{ll} \min & f(x) \\ \text{s.d.} & g_i(x) \leq 0 \ \text{ für alle } i = 1, \ldots, m \\ & h_j(x) = 0 \ \text{ für alle } j = 1, \ldots, l. \end{array} \tag{4.27}$$

Die Verallgemeinerung von Satz 4.43 auf solche Optimierungsprobleme lautet wie folgt.

Satz 4.45 *Sei*

$$P = \{x \in \mathbb{R}^n : g_i(x) \leq 0, i = 1, \ldots, m \text{ und } h_j(x) = 0, j = 1, \ldots, l\}$$

und sei $\bar{x} \in P$. Seien f, g_i differenzierbar an $\bar{x}$ für alle $i = 1, \ldots, m$, h_j stetig differenzierbar an $\bar{x}$ für alle $j = 1, \ldots, l$ und seien die beiden Mengen

$$\{\nabla g_i(\bar{x}) : i \in I\} \text{ und } \{\nabla h_j(\bar{x}) : j = 1, \ldots, l\}$$

jeweils linear unabhängig.

Sei $\bar{x}$ lokales Minimum von (4.27). Dann existieren (eindeutig bestimmte) Skalare $u_i, \quad i = 1, \ldots, m$ und $v_j, \quad j = 1, \ldots, l$, so dass

$$\begin{aligned} \nabla f(\bar{x}) + \sum_{i=1}^{m} u_i \nabla g_i(\bar{x}) + \sum_{j=1}^{l} v_j \nabla h_j(\bar{x}) &= 0 \\ u_i g_i(\bar{x}) &= 0 \text{ für } i = 1, \ldots, m \\ u_i &\geq 0 \text{ für } i = 1, \ldots, m \\ v_j &\gtrless 0 \text{ für } j = 1, \ldots, l \end{aligned}$$

Algorithmen für nichtlineare Programme

Wir betrachten zunächst das einfache Problem, eine eindimensionale Funktion $f : \mathbb{R} \to \mathbb{R}$ über einem unbeschränktem Gebiet (also über $\mathbb{R}$) zu minimieren. Die entsprechenden Verfahren sind unter dem Namen *Line Search Procedures* bekannt. Die Idee besteht darin, das so genannte *Intervall der Unsicherheit* (*interval of uncertainty*), also der Bereich, in dem man einen Minimierer vermutet, in jeden Schritt zu verkleinern. Man unterscheidet hierbei zwei Fälle

- Falls f nicht differenzierbar ist, kann man eines der folgenden Verfahren verwenden:
  - Uniforme Suche: Hier wird das Intervall in gleichmäßige Stücke geteilt und die Funktion wird an allen Intervallgrenzen ausgewertet. Sei x^* die Intervallgrenze mit dem kleinsten Zielfunktionswert. Als neues Intervall für den nächsten Schritt wählt man die beiden Intervalle rechts und links von x^*.
  - Dichotomous Suche: Sei $I = [a_k, b_k]$ das aktuelle Intervall der Unsicherheit im $k$ten Schnitt. Setze $\lambda_k = \frac{a_k + b_k}{2} - \varepsilon$, $\mu_k = \frac{a_k + b_k}{2} + \varepsilon$.
    Wenn $f(\lambda_k) < f(\mu_k)$ dann setze $a_{k+1} = a_k, b_{k+1} = \mu_k$, sonst setze $a_{k+1} = \lambda_k, b_{k+1} = b_k$.
  - Verfahren des Goldenen Schnitts: Sei wiederum $I = [a_k, b_k]$ das Intervall der Unsicherheit im $k$ten Schritt. In diesem Fall verwenden wir $\lambda_k = a_k + (1 - \alpha)(b_k - a_k)$ und $\mu_k = a_k + \alpha(b_k - a_k)$.
    Falls $f(\lambda_k) > f(\mu_k)$ setze $\lambda_{k+1} = \mu_k$, sonst setze $\mu_{k+1} = \lambda_k$.
- Falls f differenzierbar ist, kann man die Ableitung mit verwenden. Hier sind die folgenden beiden Verfahren geläufig.
  - Bisection Search: Sei $I = [a_k, b_k]$ das Intervall der Unsicherheit im $k$ten Schritt, und sei $\lambda_k = \frac{a_k + b_k}{2}$. Wir testen die Ableitung $f'(\lambda_k)$. Falls $f'(\lambda_k) = 0$ ist λ_k ein lokales Minimum. Ist $f'(\lambda_k) > 0$, dann setzen wir $a_{k+1} = a_k, b_{k+1} = \lambda_k$, für den Fall $f'(\lambda_k) < 0$ wählen wir $a_{k+1} = \lambda_k, b_{k+1} = b_k$.

– Newton: Die Idee des Newton-Verfahrens besteht darin, die quadratische Approximation von f an λ_k zu minimieren. Das geschieht durch Wahl der Nullstelle der quadratischen Approximation. In Formeln erhält man $\lambda_{k+1} = \lambda_k - \frac{f'(\lambda_k)}{f''(\lambda_k)}$.

Jetzt beschäftigen wir uns mit Funktionen

$$f : \mathbb{R}^n \to \mathbb{R}.$$

Wir verfolgen dabei die folgende Idee: Finde für einen gegebenen Punkt $x \in \mathbb{R}^n$ eine Richtung $d \in \mathbb{R}^n$ und minimiere die eindimensionale Funktion

$$\theta(\lambda) = f(x + \lambda d) \text{ so dass } \lambda \in [a, b]$$

mit Hilfe einer der oben genannten Line-Search-Prozeduren. Wieder unterscheiden wir zwei Fälle.

- Falls f nicht differenzierbar ist, bieten sich die folgenden Verfahren an.

  – Zyklische Koordinaten: Wähle dazu $d_1, \ldots, d_n$ als Koordinatenachsen, und suche nacheinander entlang jeder der Achsen mit einer eindimensionalen Line-Search Prozedur. Für die erste Koordinate löst man also zunächst

  $$\min_x f \begin{pmatrix} x \\ x_2 \\ \vdots \\ x_n \end{pmatrix}$$

  und erhält dadurch eine optimale Lösung $x = x_1^*$. Im nächsten Schritt hält man x_1^* fest und löst

  $$\min_x f \begin{pmatrix} x_1^* \\ x \\ x_3 \\ \vdots \\ x_n \end{pmatrix}$$

  mit Optimallösung $x = x_2^*$, die man dann im dritten Schritt einsetzt. Das führt man so lange fort, bis sich in keiner Koordinate noch eine Verbesserung ergibt.

  – Hooke and Jeeves: Das Verfahren ist eine Verbesserung der zyklischen Koordinaten, in der man nach einer Suche entlang aller Koordinatenachsen jeweils einen *Gesamtschritt* einsetzt.

  Sei also x_k die Lösung im k-ten Schritt von Hooke and Jeeves. Man verbessert nun x_k zunächst entlang aller Koordinatenachsen $1, \ldots, n$

wie im Verfahren der zyklischen Koordinaten und erhält x'_{k+1}. Dann setzt man

$$d = x'_{k+1} - x_k$$

und optimiert in diese Richtung, um so einen größeren Schritt zurückzulegen. Man erhält x_{k+1} und fährt dafür wieder mit den zyklischen Koordinaten fort.

- Rosenbrock: Das Verfahren funktioniert ähnlich wie das Verfahren der zyklischen Koordinaten, aber in jedem Schritt wird ein neues orthogonales Koordinatensystem gewählt.

Ist f dagegen differenzierbar, so sind die folgenden beiden Verfahren üblich.

- steilster Abstieg (*steepest descent*) : Wähle eine Richtung $d \in \mathbb{R}^n$, so dass die Richtungsableitung von x_k in Richtung d $f'(x_k, d) < 0$ erfüllt. Dann ist d eine verbessernde Richtung (siehe Lemma 4.36). Man verbessert x_k entlang der Richtung d und sucht im nächsten Punkt x_{k+1} erneut eine verbessernde Richtung. Das entspricht in jedem Schritt einer linearen Taylor-Approximation der Funktion.
- Newton-Verfahren: Hier verwendet man nicht die lineare, sondern die quadratische Approximation von f an x_k, und erhält

$$x_{k+1} = x_k - (H(x_k))^{-1}(\nabla f(x_k))^t.$$

Als letzte Klasse von Algorithmen skizzieren wir kurz Algorithmen für restringierte Probleme der Form

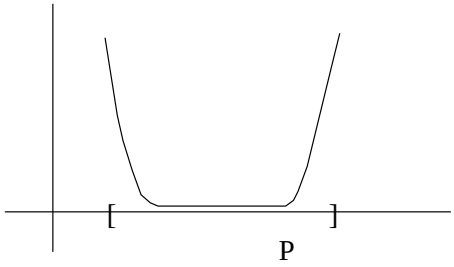
$$\min\{f(x) : x \in P\}.$$

Wir kennen in einigen Spezialfällen schon gute Verfahren:

- Ist f linear und $P =$ ein Polyeder, so liegt ein lineares Programm vor und wir können beispielsweise das Simplex-Verfahren oder Innere-Punkt-Methoden verwenden.
- Für konvexe Funktionen f über konvexen Mengen P haben wir das Subgradienten-Verfahren kennen gelernt.
- Ist f dagegen eine beliebige Funktion, die über einer beliebigen Menge P minimiert werden soll, müssen wir relativ allgemeine Verfahren verwenden. Die Idee besteht darin, nicht erfüllte Nebenbedingungen in der Zielfunktion zu "bestrafen" und das restringierte Optimierungsproblem damit in jedem Schritt auf ein Optimierungsproblem ohne Nebenbedingungen zurückzuführen. Bekannt sind die folgenden beiden Methoden.
  - Barriere-Methode: Hier werden Lösungen, die in der Nähe des verbotenen Gebietes liegen, bestraft.

– Strafverfahren: Hier bestraft man ausschließlich unzulässige Lösungen.

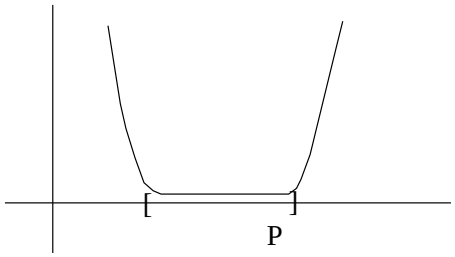
Der Unterschied zwischen den beiden Verfahren ist in der folgenden Skizze verdeutlicht.



Min. Zielfunktion,

die Unzulässigkeit

verhindet!



Kapitel 5

Standortplanung als Anwendung in der Optimierung

5.1 Was sind Standortprobleme?

Ein Standortproblem ist wie folgt definiert.

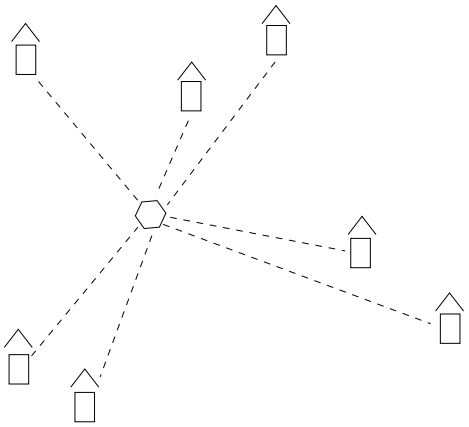
- Gegeben sind *existierende Standorte* $\mathcal{A} = \{A_1, \cdots, A_M\}$, die eventuell mit Gewichten $w_m \geq 0$ bezüglich ihrer Bedeutung bewertet sind. Weiter benötigt man eine Entfernungsfunktion, z.B. die Euklidische Entfernung.

- Gesucht sind ein oder mehrere neue Standorte. Sie sollen so gewählt werden, dass die Entfernungen zwischen den neuen und den existierenden Standorten möglichst klein sind.

Wir beschreiben zunächst einige (stark vereinfachte) Anwendungen, um zu illustrieren, wo Standortprobleme überall auftreten können. Weitere Anwendungen, Theorie und zahlreiche Verfahren finden sich beispielsweise in den Lehrbüchern von [Ham95, LMW88].

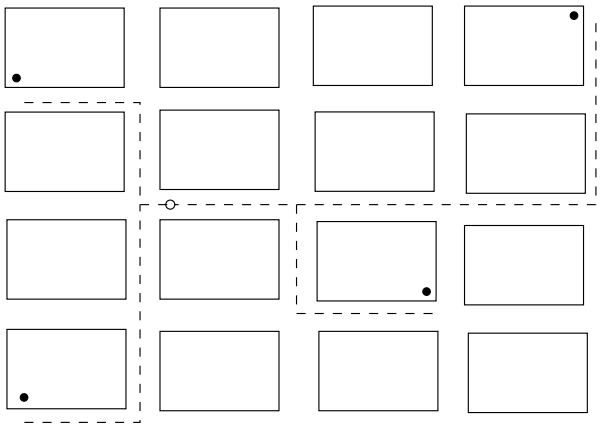
Wüste. Wo in der Wüste soll ein Anbieter einen Brunnen bauen, so dass er einen möglichst geringen Weg zurücklegen muss, um alle existierenden Standorte mit Wasser zu versorgen?

Das Beispiel *Wüste* wird gerne herangezogen, weil es die Euklidische Entfernung rechtfertigt. Zu minimieren ist hier die Summe der Entfernungen von dem gesuchten neuen Standort zu den existierenden Standorten. Standortprobleme, in denen die Summe der Entfernungen minimiert werden soll, werden *Medianprobleme* oder *Minisum-Probleme* genannt.



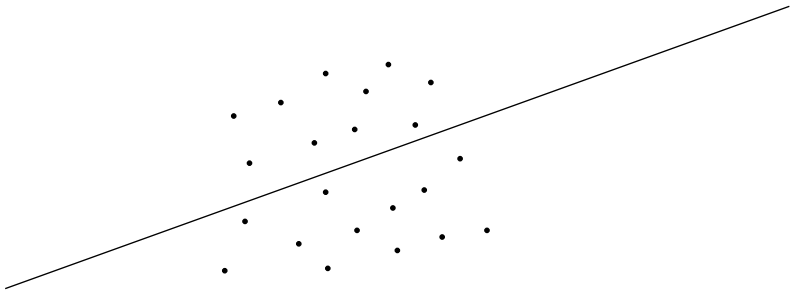
Manhattan. Was ist ein guter Standort für eine Feuerwache oder einen privaten Einbruchschutz in Manhattan?

In Manhattan macht es wenig Sinn, Entfernungen mit der Euklidischen Metrik zu messen, sondern es bietet sich die l_1 Entfernung (auch *Manhattan-Entfernung* genannt) an. Eine sinnvolle Zielfunktion in diesem Beispiel bewertet nicht die Summe der Entfernungen von der Feuerwache zu den existierenden Standorten sondern misst die maximale Entfernung, d.h. also die Entfernung von der Feuerwache zu dem am weitesten entfernt liegenden existierenden Standort. Solche Standortprobleme werden als *Center-Probleme* bezeichnet.

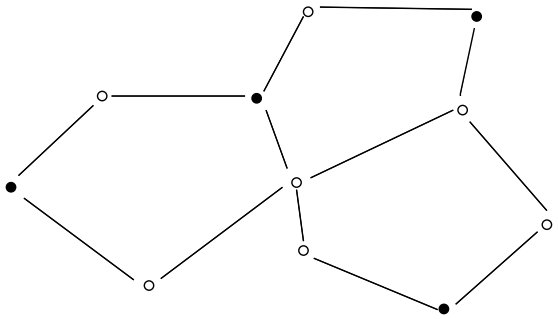


Regressionsgerade. Auch in der Statistik tauchen Standortprobleme auf. Beispielsweise besteht das lineare Ausgleichsproblem darin, eine Gerade so zu platzieren, dass eine Menge an existierenden Messpunkten möglichst gut approximiert wird. Die Entfernung von den existierenden Standorten (also den Messpunkten) zu der Regressionsgerade wird in der Statistik meistens mittels des *vertikalen Abstandes* zwischen den Punkten und der Geraden

gemessen; bei Verwendung der Euklidischen Entfernung wählt man als Zugang zur Geraden das Lot (bekannt als *orthogonale Regression*). Meistens betrachtet man die Summe der quadrierten Abstände.



Straßennetz. Nicht immer ist der Grundraum für Standortprobleme die Ebene $\mathbb{R}^2$. Beispielsweise treten oft Standortfragen in Netzwerken auf. Eine typische Fragestellung sucht z.B. den besten Platz für ein oder mehrere Warenlager innerhalb eines gegebenen Straßennetzes. Auch hier soll der neue Standort so gewählt werden, dass z.B. die Summe der Entfernungen zu den Kunden möglichst gering ist. Entfernungen werden in diesem Problem mit Hilfe der Netzwerkentfernung entlang der Kanten des gegebenen Netzwerkes gemessen.



Um die Vielfalt von Standortproblemen zu sortieren, wurde von Hamacher und Nickel [HN98] ein Klassifikationsschema eingeführt. Es besteht aus den folgenden fünf Positionen

Neue Standorte / Grundraum / Besonderheiten / Entfernung / Zielfunktion

Als Einträge für die fünf Positionen sind die folgenden Angaben möglich:

- *Neue Standorte* gibt Typ und Anzahl der zu platzierenden Objekte an. Möglichkeiten sind z.B.
  - 1 Punkt,

  - N Punkte,
  - 1 Gerade,
  - 2 Hyperebenen,
  - ...

- Der *Grundraum* bezeichnet die Menge, in der die neuen Standorte gesucht werden. Das kann der $\mathbb{R}^n$ sein, oder die Ebene $P = \mathbb{R}^2$ (in letzterem Fall spricht man von *planaren* Standortproblemen). Liegt ein Problem in einem Netzwerk vor, so wird ein N eingetragen. Ein D steht für ein *diskretes* Standortproblem: Hier ist eine endliche Menge an Standorten vorgegeben, unter denen man die neuen Standorte wählt.

- In die Position *Besonderheiten* wird alles eingetragen, was man als Modellerweiterung mit betrachtet. Eine häufig betrachtete Erweiterung ist es, z.B. *verbotene Gebiete* zuzulassen, die für die Platzierung neuer Standorte nicht zulässig sind (man schreibt dann R). Auch so genannte *Barrieren*, durch die man noch nicht einmal durchreisen darf, können als Besonderheit vermerkt werden.

- Als Entfernungsmaße sind alle aus Normen abgeleiteten Metriken (z.B. l_2, l_1, l_∞) üblich, aber auch mit anderen Abstandsmaßen werden Standortprobleme diskutiert. Ein Trend scheint es derzeit zu sein, planare Entfernungen und Netzwerkentfernungen integriert zu behandeln.

Die vier genannten Standortprobleme kann man folgendermaßen klassifizieren.

Wüste. $1/\mathbb{R}^n/./l_2/\sum$

Manhattan. $1/\mathbb{R}^2/R =$Häuser$/l_1/\max$

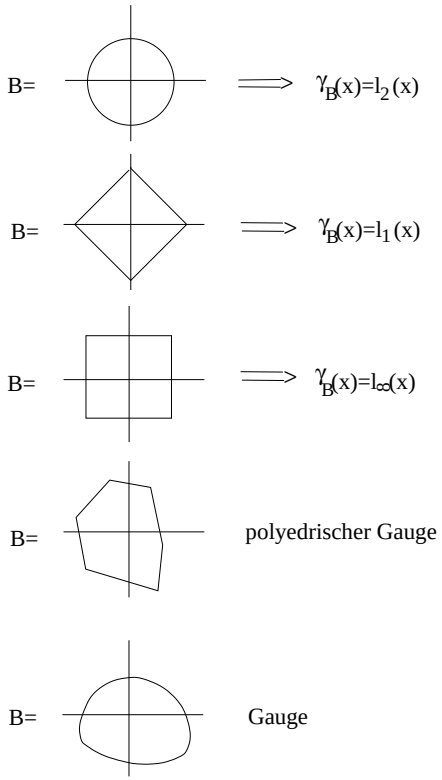
Regressionsgerade. $1l/\mathbb{R}^2/./(d_{ver})^2/\sum$

Straßennetz. $N/N/./d/\sum$

Als Entfernungsmaße betrachten wir eine Verallgemeinerung von Normen, so genannte *Gauges*. Diese Entfernungen haben den Vorteil, dass sie auch unsymmetrische Gegebenheiten abbilden können. In der Standortplanung macht es Sinn, sich die Definition von Entfernungen anhand eines gegebenen Einheitskreises zu verdeutlichen — ein Vorgehen, dass zuerst von Minkowski (als die so genannten *Minkowski-Funktionale*) eingeführt wurde. Sei dazu $\|\cdot\| : \mathbb{R}^n \to \mathbb{R}$ eine Norm. Der Einheitskreis $B_{\|\cdot\|}$ zu Norm $\|\cdot\|$ ist definiert als

$$B_{\|\cdot\|} = \{x \in \mathbb{R}^n : \|x\| \leq 1\}.$$

Die ersten drei Beispiele der folgenden Abbildung zeigen die Einheitskreise der Euklidischen Norm, der Manhattan-Norm und der Maximum-Norm.



Die letzten zwei Abbildungen zeigen konvexe Mengen, aus denen man sich wie folgt *Gauges* definieren kann.

Definition 5.1 *Sei $B \subseteq \mathbb{R}^n$ kompakt und konvex und $0 \in int(B)$. Der* **Gauge** *zu B ist definiert als die Menge*

$$\gamma_B(x) = \inf\{\eta \geq 0 : x \in \eta B\}.$$

B heißt der **Einheitskreis** *von γ_B. Ist B ein Polyeder, so nennt man γ_B* **polyedrischen Gauge**.

Wir untersuchen zunächst die Existenz von γ_B an einer Stelle $x \in \mathbb{R}^n$.

Satz 5.2 *Sei $B \subseteq \mathbb{R}^n$ kompakt und konvex und $0 \in int(B)$ Dann existiert $\gamma_B(x)$ für alle $x \in \mathbb{R}^n$, d.h.*

$$\inf\{\eta \geq 0 : x \in \eta B\} = \min\{\eta \geq 0 : x \in \eta B\}.$$

Beweis: Zunächst ist $P = \{\eta \geq 0 : x \in \eta B\} \subseteq \mathbb{R}$ ein Intervall. Weil $0 \in int(B)$ gilt weiter $P \neq \emptyset$. Außerdem ist P nach unten beschränkt. Aus der Kompaktheit von B folgt weiter, dass P abgeschlossen ist.
Weil die Funktion $f(\eta) = \eta$ stetig ist existiert nach dem Satz von Weierstrass (Satz 1.3) also ein Minimum des Optimierungsproblems

$$\min\{\eta : \eta \in P\}.$$

QED

Der Zusammenhang zwischen Gauges und Normen wird durch die folgende, äquivalente Definition von Gauges deutlich.

Satz 5.3 *γ_B ist ein Gauge genau dann wenn die folgenden drei Bedingungen für alle $x, y \in \mathbb{R}^n$ gelten:*

1. $\gamma_B(x) \geq 0$, $\gamma_B(x) = 0 \Leftrightarrow x = 0$.
2. $\gamma_B(\lambda x) = \lambda \gamma_B(x)$ *für alle* $\lambda \geq 0$.
3. $\gamma_B(x + y) \leq \gamma_B(x) + \gamma_B(y)$.

Beweis:

$\Longrightarrow$**:** Sei γ_B ein Gauge wie in Definition 5.1 eingeführt.

Wir führen hier nur den Beweis für die 3. Aussage (die Dreiecksungleichung) vor; die ersten beiden Aussagen sollen als ÜBUNG bewiesen werden.

Definieren wir für den Beweis $\eta_x := \gamma_B(x),\ \eta_y := \gamma_B(y)$. Unser Ziel ist es, zu zeigen dass

$$x + y \in (\eta_x + \eta_y)B, \tag{5.1}$$

denn daraus folgt sofort, dass

$$\gamma_B(x + y) = \min\{\eta \geq 0 : x + y \in \eta B\} \leq \eta_x + \eta_y.$$

Sei also $x \in \eta_x B$, $y \in \eta_y B$. Daraus folgt zunächst, dass $\frac{x}{\eta_x} \in B$, $\frac{y}{\eta_y} \in B$. Weil (5.1) äquivalent ist zu $\frac{x+y}{\eta_x+\eta_y} \in B$ rechnen wir nach, dass

$$\frac{x + y}{\eta_x + \eta_y} = \underbrace{\frac{\eta_x}{\eta_x + \eta_y}}_{=:\mu_1}\left(\frac{x}{\eta_x}\right) + \underbrace{\frac{\eta_y}{\eta_x + \eta_y}}_{=:\mu_2}\left(\frac{y}{\eta_y}\right).$$

Die beiden Faktoren μ_1 und μ_2 erfüllen $0 \leq \mu_1, \mu_2 \leq 1$ und außerdem $\mu_1 + \mu_2 = 1$. Also ist $\frac{x+y}{\eta_x+\eta_y}$ eine Konvexkombination von Elementen aus B. Aus der Konvexität von B folgt

$$\frac{x + y}{\eta_x + \eta_y} \in B$$

und daraus (5.1).

$\Longleftarrow$: Sei nun γ_B eine Funktion, die den Bedingungen des obigen Satzes genügt. Wir definieren

$$B := \{x \in \mathbb{R}^n : \gamma_B(x) \leq 1\}$$

und möchten nachweisen, dass

- B kompakt und konvex ist und die Null in seinem Inneren enthält, und
- $y_B(x) = \inf\{\lambda \geq 0 : x \in \lambda B\}$ gilt.

Man kann wie folgt vorgehen: Als erstes weist man die Stetigkeit von γ nach, indem man aus (2) zunächst die Stetigkeit im Ursprung erhält und diese mittels (3) dann für beliebige Punkte zeigt. Aus der Stetigkeit von γ_B folgt dann die Existenz einer Umgebung um den Ursprung, die komplett in B enthalten ist und die Abgeschlossenheit von B.

Die Beschränktheit von B erhält man durch Gegenannahme: Wäre B unbeschränkt, so gäbe es einen Strahl $\delta\eta$ mit $\eta \in \mathbb{R}^n$, so dass für alle $\delta > 0$ gilt: $\delta\eta \in B$, also $\gamma_B(\delta\eta) \leq 1$. Zusammen mit (2) würde daraus aber folgen, dass

$$\delta \underbrace{\gamma_B(\eta)}_{=:r\leq 1} = \gamma_B(\delta\eta) \leq 1$$

für alle $\delta > 0$, ein Widerspruch.

Die Konvexität von B folgt durch folgende Ungleichungskette: Seien $x, y \in B$ und $\lambda \in (0, 1)$. Dann gilt

$$\begin{aligned} \gamma_B(\lambda x + (1-\lambda y)) &\leq \gamma_B(\lambda x) + \gamma_B((1-\lambda)y) \text{ nach (3)} \\ &= \lambda \underbrace{\gamma_B(x)}_{\leq 1} + (1-\lambda) \underbrace{\gamma_B(y)}_{\leq 1} \text{ nach (2)} \\ &\leq \lambda + (1-\lambda) = 1 \end{aligned}$$

Als letztes folgt $\gamma_B(x) = \inf\{\lambda \geq 0 : x \in \lambda B\}$ durch folgende Überlegung: Sei $r := \gamma_B(x)$. Dann ist

$$\gamma_B(\frac{x}{r}) = 1 \text{ und } \gamma_B(\frac{x}{r'}) > 1 \text{ für alle } r' > r.$$

also gilt $\frac{x}{r} \in B$ und $\frac{x}{r} \notin B$ für alle $r' > r$, oder,

$$x \in rB \text{ und } x \notin r'B \text{ für alle } r' > r.$$

Entsprechend folgt die Behauptung. QED

Der letzte Satz zeigt, dass jede Norm ein Gauge ist, da für Normen $\|\cdot\|$ die Bedingung

$$\|\lambda x\| = |\lambda|\|x\|$$

gefordert wird, aus der die zweite Bedingung des obigen Satzes folgt. Daraus ergibt sich der folgende Zusammenhang zwischen Gauges und Normen.

Satz 5.4 .

1. *Sei γ eine Norm. Dann ist $B(\gamma) := \{x \in \mathbb{R}^n : \gamma(x) \leq 1\}$ kompakt, konvex und enthält 0 in seinem Inneren.*

2. *Sei B kompakt, konvex, $0 \in int(B)$ und B punktsymmetrisch zu 0. Dann ist γ_B Norm.*

Beweis: Der erste Teil des Satzes folgt direkt aus Satz 5.3, da jede Norm ein Gauge ist. Für die zweite Aussage muss man nachweisen, dass aus

$$x \in B \Longrightarrow -x \in B$$

die zweite Normeigenschaft, also

$$\gamma_B(\lambda x) = |\lambda|\gamma_B(x)$$

folgt. Weil γ ein Gauge ist, wissen wir, dass diese Aussage für positive λ gilt. Sei nun also $\lambda < 0$. Dann

$$\begin{aligned}
\gamma_B(\lambda x) &= \inf\{\eta \geq 0 : \lambda x \in \eta B\} \\
&= \inf\{\eta \geq 0 : -(\lambda x) \in \eta B\} \text{ weil } B \text{ punktsymmetrisch} \\
&= \gamma_B(-\lambda x) \text{ nach Definition der Gauge-Entfernung} \\
&= -\lambda\gamma_B(x) \text{ weil } -\lambda > 0 \\
&= |\lambda|\gamma_B(x)
\end{aligned}$$

QED

Definition 5.5 *Eine Entfernung $d : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}$ heißt* **Gauge-Entfernung**, *falls es einen Gauge γ gibt, so dass für alle $x, y \in \mathbb{R}^n$*

$$d(x, y) = \gamma(y - x)$$

erfüllt ist.

5.2 Lösung einiger 1-Standortprobleme

In diesem Abschnitt betrachten wir exemplarisch 1-Standort-Medianprobleme des Typs $1/\mathbb{R}^2/./d/\sum$. In solchen Problemen ist eine Menge existierender Standorte

$$\mathcal{A} = \{A_1, \ldots, A_M\} \subseteq \mathbb{R}^2$$

mit Koordinaten $A_m = (a_{m1}, a_{m2})$ und Gewichten $w_m \geq 0$ gegeben. Die Positivität der Gewichte wollen wir dabei fest voraussetzen. Gesucht ist ein neuer Standort $x \in \mathbb{R}^2$, der die Summe der Abstände zu den existierenden Standorten minimiert. Unser Problem ist also gegeben als

$$\min_{x \in \mathbb{R}^2} f(x) = \sum_{m=1}^{M} w_m d(A_m, x).$$

Zunächst gilt die folgende Aussage.

Lemma 5.6 *Sei d eine Gauge-Entfernung. Dann ist $f(x)$ konvex.*

Beweis: Von Lemma 4.11 (Teil a) (siehe Seite 137) wissen wir, dass positive Linearkombinationen konvexer Funktionen selbst wieder konvex sind. Weil wir $w_m \geq 0$ vorausgesetzt haben, reicht es also, die Konvexität von $d(A, x)$ nachzuweisen. Dazu unterscheiden wir zwei Fälle.

Fall 1: Sei $A = 0$. Dann ist $d(A, x) = d(0, x) = \gamma_B(x)$. Wie wir im Beweis von Satz 5.3 bereits gesehen haben, gilt für $0 \leq \lambda \leq 1$, dass

$$\begin{aligned} \gamma_B(\lambda x_1 + (1-\lambda)x_2) &\leq \gamma_B(\lambda x_1) + \gamma_B((1-\lambda)x_2) \text{ Satz 5.3, (3)} \\ &= \lambda\gamma_B(x_1) + (1-\lambda)\gamma_B(x_2) \text{ Satz 5.3, (2).} \end{aligned}$$

Fall 2: Sei nun $A \in \mathbb{R}^2$ beliebig. Wir definieren

$$k(x) := d(A, x) = \gamma_B(x - A).$$

Es ist $x - A$ linear und (nach dem ersten Fall) γ_B konvex. Daraus folgt die Konvexität der Funktion k, siehe Teil d) von Lemma 4.11.

QED

Aus Satz 4.14 ergibt sich damit sofort die folgende Aussage.

Korollar 5.7 *Standortprobleme der Form $1/\mathbb{R}^2/./\gamma_B/\sum$ sind konvex, d.h. jedes lokale Minimum ist ein globales Minimum.*

Wir betrachten nun 1-Standort-Medianprobleme mit einigen speziellen Abständen genauer.

5.2.1 Standortprobleme mit $\gamma_B = l_1$

Das Optimierungsproblem in diesem Spezialfall lautet

$$\min_{x \in \mathbb{R}^2} f(x) = \sum_{m=1}^{M} w_m(|a_{m1} - x_1) + |a_{m2} - x_2|).$$

Erfreulicherweise ist die Zielfunktion $f(x)$ separierbar,

$$f(x) = \sum_{m=1}^{M} w_m(|a_{m1} - x_1| + |a_{m2} - x_2|) = \sum_{m=1}^{M} w_m|a_{m1} - x_1| + \sum_{m=1}^{M} w_m|a_{m2} - x_2|$$

so dass wir das Optimierungsproblem in zwei voneinander unabhängige, eindimensionale Probleme des gleichen Typs zerlegen können. Diese lauten

$$\min_{x_1 \in \mathbb{R}} \sum_{m=1}^{M} w_m|a_{m1} - x_1| \quad \text{und} \quad \min_{x_2 \in \mathbb{R}} \sum_{m=1}^{M} w_m|a_{m2} - x_2|.$$

Wir betrachten nun ein solches eindimensionales Optimierungsproblem

$$\min_{x \in \mathbb{R}} h(x) := \sum_{m=1}^{M} w_m|a_m - x| \tag{5.2}$$

mit der eindimensionalen Zielfunktion $h : \mathbb{R} \to \mathbb{R}$. Wir lassen zunächst alle a_m weg, für die $w_m = 0$ ist und sortieren die verbliebenen a_m nach ihrer Größe. Durch Zusammenfassen von identischen Koordinaten a_m (bei Addition der entsprechenden Gewichte) erhalten wir sogar eine strikte Ordnung $a_1 < a_2 < \cdots < a_{M'}$. Im folgenden setzen wir diese (strikte) Sortierung für unser Optimierungsproblem (5.2) voraus.

Lemma 5.8 *Die Funktion h hat die folgenden Eigenschaften.*

- *h ist linear auf dem Intervall $(-\infty, a_1]$, auf jedem Intervall $[a_{m-1}, a_m]$ für $m = 2, \ldots, M$ und auf $[a_M, \infty)$.*
- *h ist konvex.*

Beweis: Die erste Aussage folgt, da in den entsprechenden Intervallen keiner der Ausdrücke $(a_m - x)$ sein Vorzeichen wechselt, die zweite Aussage folgt aus Lemma 5.6. QED

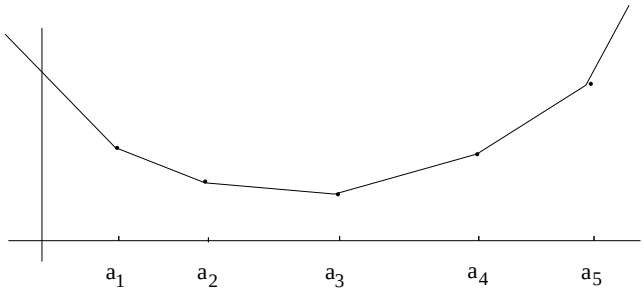
Beispielsweise aus dem Hauptsatz der linearen Optimierung (Satz 2.21) folgt nun:

Lemma 5.9 *Es gibt einen Minimierer $x^* = a_m$ für ein $m = 1, \ldots, M$.*

Weil $\mathbb{R}$ offen ist folgt aus Korollar 4.26 weiter:

Lemma 5.10 *x^* ist ein Minimum von $h(x)$ genau dann wenn ein Subgradient $\xi = 0$ an x^* existiert.*

Um herauszufinden, welches der a_m das Minimum ist, betrachten wir die Steigungen der Funktion h auf den Intervallen $[a_{m-1}, a_m]$.



- Betrachten wir zunächst ein Intervall $[a_m, a_{m+1}]$, $m = 1, \ldots, M-1$: Die Zielfunktion auf diesem Intervall lässt sich betragsfrei schreiben als lineare Funktion
$$h(x) = \sum_{i=1}^{m} w_i(x - a_i) + \sum_{i=m+1}^{M} w_i(a_i - x).$$
Ihre Ableitung ist auf dem Inneren des Intervalls, also auf (a_m, a_{m+1}) konstant und beträgt $h'_m = \sum_{i=1}^{m} w_i - \sum_{i=m+1}^{M} w_i$.
- Die Ableitung auf dem Intervall $(-\infty, a_1)$ ganz links ergibt sich für alle $x < a_1$ konstant zu $h'_0 = -\sum_{i=1}^{M} w_i < 0$,
- die Ableitung auf dem letzten Intervall (a_M, ∞) beträgt $h'_M = \sum_{i=1}^{M} w_i > 0$.

Weiterhin sind die Ableitungen auf den Intervallen von links nach rechts streng monoton wachsend, also $h'_0 < h'_1 < \cdots < h'_M$. Daraus folgt:

$$x = a_m \text{ hat einen Subgradienten von Null } \iff h'_{m-1} \leq 0 \text{ und } h'_m \geq 0. \quad (5.3)$$

Es ergeben sich die folgenden beiden Fälle für einen Subgradienten von Null:

Fall 1: Es gibt ein $m \in 0, 1, \ldots, M$ mit $h'_m = 0$. Dann haben alle $x \in [a_m, a_{m+1}]$ einen Subgradienten von Null, und entsprechend sind alle $x \in [a_m, a_{m+1}]$ optimal.

Fall 2: $h'_m \neq 0$ für alle $m = 0, \ldots, M$. Kein $x \notin \{a_1, \ldots, a_M\}$ kann dann einen Subgradienten von Null haben, d.h. es gibt genau ein a_m mit Subgradienten Null und $x = a_m$ ist die eindeutige Optimallösung.

Abschließend schreiben wir Bedingung (5.3) um: a_m hat einen Subgradienten von Null genau dann wenn

$$\sum_{i=1}^{m-1} w_i - \sum_{i=m}^{M} w_i \leq 0 \quad \text{und} \quad \sum_{i=1}^{m} w_i - \sum_{i=m+1}^{M} w_i \geq 0$$

was äquivalent ist zu

$$m = \min\{m' : \sum_{i=1}^{m'} w_i \geq \frac{1}{2} \sum_{i=1}^{M} w_i\}.$$

Die Lösungen von (5.3) entsprechen also dem aus der Statistik bekannten *gewichteten Median* bezüglich der Daten $a_1, \ldots, a_M$ mit positiven Gewichten $w_m, m = 1, \ldots, M$.

Wir fassen dieses Ergebnis in dem folgenden Satz zusammen.

Satz 5.11 *Sei* $m^* = \min\{m : \sum_{i=1}^{m} w_i \geq \frac{1}{2} \sum_{i=1}^{M} w_i\}$. *Dann gilt:*

1. $x^* = a_m^*$ *ist eindeutiges Min von* $h(x)$, *falls* $\sum_{i=1}^{x^*} w_i > \frac{1}{2} \sum_{i=1}^{M} w_i$.
2. *Sonst ist* $[a_m^*, a_{m^*+1}]$ *die Menge der Minimierer von* $h(x)$.

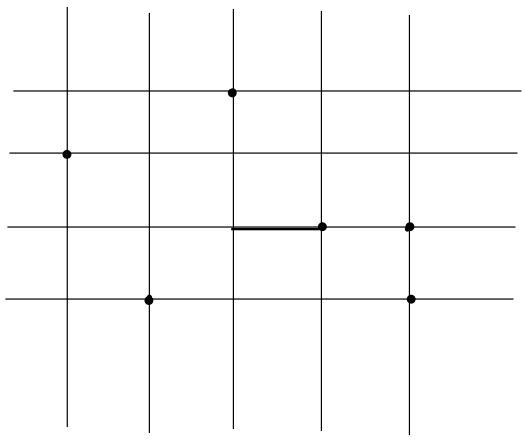
Wir nutzen dieses Ergebnis, um die Menge der Optimallösungen für das 1-Medianproblem mit Manhattan-Entfernung zu beschreiben.

Satz 5.12 *Die Menge der optimalen Standorte für das Standortproblem* $1/\mathbb{R}^2/./l_1/\sum$ *mit existierenden Standorten* $A_m = (a_{m1}, a_{m2})$, $m = 1, \ldots, M$ *ist ein (im allgemeinen degeneriertes) Rechteck der Form*

$$[a_{i1}, a_{j1}] \times [a_{k2}, a_{l2}]$$

mit $i, j, k, l \in \{1, 2, \ldots, M\}$.

Beispiel 5.1 *Im folgenden Beispiel mit sechs existierenden Standorten und Gewichten* $w_m = 1$ *für alle* $m = 1, 2, \ldots, 6$ *ergibt sich eine horizontale Strecke als die Menge der optimalen Standorte.*



Die in der Zeichnung skizzierten vertikalen und horizontalen Linien durch die existierenden Standorte A_m heißen auch *Konstruktionslinien*. Sie zerlegen den $\mathbb{R}^2$ in *Zellen*. Auf jeder Zelle ist die Zielfunktion f linear.

5.2.2 Standortprobleme mit $d = l_\infty$

Standortprobleme mit der Maximum-Norm als Entfernung lassen sich leicht auf die im letzten Abschnitt behandelten Standortprobleme mit Manhattan-Entfernung zurückführen. Dazu betrachten wir die folgende Transformation $T : \mathbb{R}^2 \to \mathbb{R}^2$,

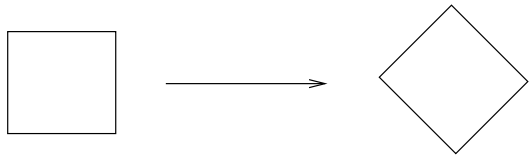
$$T = \frac{1}{2} \begin{pmatrix} 1 & 1 \\ -1 & 1 \end{pmatrix}.$$

Diese Transformation erlaubt es, die l_∞ und die l_1 Entfernung ineinander zu überführen. Genauer gilt die durch Fallunterscheidung leicht zu bestätigende Aussage:

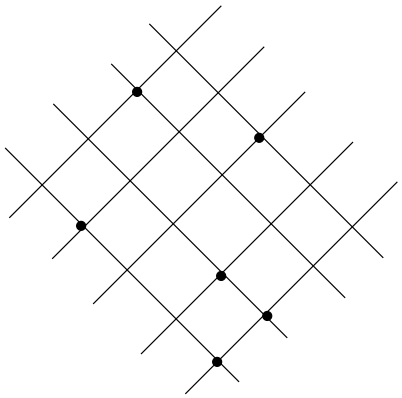
$$l_\infty(x, y) = l_1(T(x), T(y)).$$

Unter Anwendung der Transformation T können wir also das Problem mit Maximum-Entfernung auf ein äquivalentes Standortproblem mit l_1-Entfernung transformieren.

Geometrisch bildet die Transformation T den Einheitskreis der Maximum-Entfernung auf den l_1 Einheitskreis ab. Das entspricht einer Drehstreckung.



Um das Problem mit der Maximum-Entfernung auch ohne die Transformation direkt zu lösen, benutzt man die um 45 Grad gedrehten Konstruktionslinien der Manhattan-Entfernung. Man zeichnet also Geraden mit Steigung +1 und Geraden mit Steigung -1 durch alle existierenden Standorte. Die Zielfunktion des Standortproblems ist auf den dabei entstehenden Zellen linear, und die Optimallösung ergibt sich als gewichteter Median entlang dieser gedrehten Konstruktionslinien.



5.2.3 Standortprobleme mit $d = l_2^2$

Bevor wir uns mit der Euklidischen Entfernung beschäftigen, betrachten wir zunächst die quadrierte Euklidische Entfernung. (Vorsicht: Diese Entfernung ist keine Gauge-Entfernung!) Das 1-Median-Problem ergibt sich zu

$$\min_{x \in \mathbb{R}^2} f(x) = \sum_{m=1}^{M} w_m((a_{m1} - x_1)^2 + (a_{m2} - x_2)^2)$$

Wegen

$$f(x) = \sum_{m=1}^{M} w_m(a_{m1} - x_1)^2 + \sum_{m=1}^{M} w_m(a_{m2} - x_2)^2$$

lässt sich dieses Problem wie das Problem mit der Manhattan-Entfernung in zwei voneinander unabhängige, eindimensionale Probleme des Typs

$$\min_{x \in \mathbb{R}} h(x) = \sum_{m=1}^{M} w_m(a_m - x)^2$$

separieren. Die Funktion $h(x)$ ist differenzierbar und konvex. Wir leiten also ab und erhalten

$$h'(x) = -\sum_{m=1}^{M} 2w_m(a_m - x) = 2\sum_{m=1}^{M} w_m(x - a_m).$$

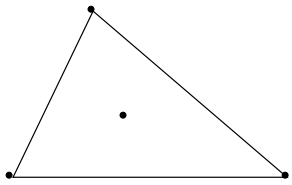
Null setzen der Ableitung ergibt entsprechend einen (eindeutigen) Minimierer:

$$h'(x) = 0 \Leftrightarrow x = \frac{2\sum_{m=1}^{M} w_m a_m}{2\sum_{m=1}^{M} w_m}.$$

Auch hier fassen wir das Ergebnis als Satz zusammen.

Satz 5.13 $x^* = (\frac{\sum w_m a_{m1}}{\sum w_m}, \frac{\sum w_m a_{m2}}{\sum w_m})$ *ist eindeutige Lösung von* $1/\mathbb{R}^2/./l_2^2/\sum$.

Geometrisch ist die optimale Lösung gerade der Schwerpunkt der gegebenen Standorte. In einem Dreieck entspricht sie also dem gemeinsamen Schnittpunkt der drei Seitenhalbierenden, oder dem Punkt auf dem man das Dreieck auf einer Bleistiftspitze (oder vielleicht besser auf dem Daumen) balancieren könnte.



5.2.4 Standortproblem mit Euklidischer Entfernung l_2

Jetzt betrachten wir Standortprobleme mit Euklidischer Entfernung. Hier treten gleich mehrere Schwierigkeiten auf. Zunächst ist die sich ergebende Zielfunktion

$$\min_{x\in\mathbb{R}^2} f(x) = \sum_{m=1}^{M} w_m \sqrt{(a_{m1}-x_1)^2 + (a_{m2}-x_2)^2}$$

nicht differenzierbar in den existierenden Standorten, also für alle $a \in \{A_1, \ldots, A_M\}$. Weiterhin kann man zeigen, dass sich eine Lösung des Standortproblemes ab $M \geq 5$ nicht mehr konstruieren oder in geschlossener Form angeben lässt. Wir sind also auf approximative Verfahren angewiesen, um den optimalen Standort zu berechnen. Aufgrund der Konvexität der Zielfunktion wissen wir aber immerhin, dass es reicht, ein lokales Optimum zu finden. Wir nutzen das aus, indem wir zunächst die nicht differenzierbaren Punkte untersuchen und dann ein Iterationsverfahren anwenden, um einen Punkt mit Gradienten Null zu erhalten. Zunächst untersuchen wir die Punkte, in denen unsere Zielfunktion nicht differenzierbar ist; also die existierenden Standorte.

Satz 5.14 *Die Zielfunktion $f(x)$ von $1/P/\cdot/\sum/l_2$ hat ein Minimum in $x^* = A_m$ für ein $m = 1, \ldots, M$ genau dann, wenn*

$$Test_i = \sqrt{\sum_{m=1,\cdots,M,m\neq i} w_m \frac{a_{i1}-a_{m1}}{l_2(A_i,A_m)} + \sum_{m=1,\cdots,M,m\neq i} w_m \frac{a_{i2}-a_{m2}}{l_2(A_i,A_m)}} \leq w_i$$

Der Beweis ist z.B. in [Ham95] ausgeführt.

In allen anderen Punkten $x \neq A_m$ ist f differenzierbar. Aufgrund der Konvexität wissen wir daher, dass x^* ein Minimierer ist genau dann wenn $\nabla f(x^*) = 0$ gilt. Um den Gradienten zu bestimmen, berechnen wir die partiellen Ableitungen nach x_1 und x_2 als

$$\frac{\partial f}{\partial x_k} = \sum_{m=1}^{M} w_m \frac{x_k - a_{mk}}{l_2(A_m, x)} \quad \text{für } k = 1, 2.$$

Dann gilt

$$\nabla f(x) = 0 \Leftrightarrow x_k = \frac{\sum_{m=1}^{M} w_m \frac{a_{mk}}{l_2(A_m,x)}}{\sum_{m=1}^{M} w_m \frac{1}{l_2(A_m,x)}} \quad \text{für } k = 1, 2.$$

Diese Gleichung lässt sich analytisch nicht nach x_1 und x_2 auflösen. Daher verwendet man das *Iterationsverfahren von Weiszfeld*, bei dem man eine Lösung mittels sukzessiver Approximation bestimmt. Zu einem gegebenen Punkt $x^{(l)}$ nach l Schritten berechnet man in Schritt $l+1$ einen neuen Punkt $x^{(l+1)} = (x_1^{(l+1)}, x_2^{(l+1)})$ durch

$$x_k^{(l+1)} = \frac{\sum w_m \frac{a_{mk}}{l_2(A_m,x^{(l)})}}{\sum w_m \frac{1}{l_2(A_m,x^{(l)})}} \quad \text{für } k = 1, 2.$$

Man kann zeigen, dass dieses Verfahren gegen einen Punkt x^* mit $\nabla(x^*) = 0$ konvergiert.

Es soll noch erwähnt werden, dass man für $M = 3$ existierende Standorte die optimale Lösung (für beliebige Gewichte) konstruieren kann. Konstruktive Lösungen sind auch für $M = 4$ existierende Standorte möglich, wenn die Gewichte für alle Standorte gleich sind, also $w_m = 1$ für alle $m = 1, 2, 3, 4$ gilt. Ab fünf existierenden Standorten lässt sich eine Optimallösung dagegen nicht mehr konstruieren.

5.2.5 Standortproblem mit polyedrischen Gauges γ_B

Als letztes deuten wir noch an, wie man 1-Medianprobleme mit beliebigen polyedrischen Gauges angehen kann. Sei γ_B ein polyedrischer Gauge. Das planare Standortproblem Problem mit Median-Zielfunktion und Gauge-Entfernung lässt sich formulieren als

$$\min_{x \in \mathbb{R}^2} f(x) = \sum_{m=1}^{M} w_m \gamma_B(A_m, x)$$

Man kann für polyedrische Gauges die folgende Aussage zeigen.

Lemma 5.15 *Sei B ein Polyeder mit Ecken $d_1, \ldots, d_p$ und sei*

$$x = \sum_{i=1}^{p} \lambda_i d_i$$

mit $\lambda_i \geq 0$ für $i = 1, \ldots, p$. Dann ist $\gamma_B(x) \leq \sum_{i=1}^{p} \lambda_i$. Genauer gilt

$$\gamma_B(x) = \min\{\sum_{i=1}^{p} \lambda_i : x = \sum_{i=1}^{p} \lambda_i d_i, \lambda_i \geq 0 \text{ für } i = 1, \ldots, p\}.$$

Mit Hilfe dieser Aussage kann man Standortprobleme mit polyedrischen Gauges als lineare Programme formulieren. Dazu nutzen wir aus, dass $\gamma_B(x, A_m) \leq \lambda_1 + \lambda_2 + \cdots + \lambda_p$, falls $x - A_m = \lambda_1 d_1 + \cdots + \lambda_p d_p$. Wir erhalten

$$\begin{aligned} \min & \sum_{m=1}^{M} (w_m \sum_{j=1}^{d} \lambda_j^m) \\ \text{s.d.} & \sum_{j=1}^{p} \lambda_j^m d_j + A_m = x \text{ für alle } m = 1, \ldots, M \\ & \lambda_j^m \geq 0 \text{ für alle } m = 1, \ldots, M,\ j = 1, \ldots, p \\ & x_1, x_2 \gtrless 0. \end{aligned}$$

Es ist zu bemerken, dass die erste Nebenbedingung für Vektoren im $\mathbb{R}^2$ aufgeschrieben ist; sie entspricht also für jedes m zwei Gleichheitsnebenbedingungen. Entsprechend erhält man $2M$ Nebenbedingungen und $pM + 2$ Variablen.

Anstatt des angegebenen linearen Programms ist auch für polyedrische Gauges ein geometrischer Ansatz möglich: Man skizziert um jeden der existierenden Standorte den polyedrischen Einheitskreis des Gauges B und zeichnet dann Strahlen ausgehend von dem existierenden Standort durch die Ecken des Polyeders. Diese Strahlen zerlegen den $\mathbb{R}^2$ in konvexe Zellen. Es lässt sich auch hier zeigen, dass die Zielfunktion auf jeder dieser Zellen linear ist; ein Optimum wird also nach dem Hauptsatz der linearen Optimierung an mindestens einer *Zellecke* angenommen. Die Zellecken wiederum entsprechen Schnittpunkten von zwei Konstruktionslinien, so dass man sie (zumindest theoretisch) berechnen kann.

Literaturverzeichnis

[Bea55] E.M.L. Beale. Cycling in the dual simplex method. *Naval Research Logistics Quaterly*, pages 269–275, 1955.

[Bla77] Bland. New finite pivoting rules for the simplex method. *Mathematics of Operarations Research*, 2:103–107, 1977.

[GJ79] M.R. Garey and D.S. Johnson. *Computers and Intractability — A Guide to the Theory of NP-Completeness*. Freeman, San Francisco, 1979.

[Ham95] H.W. Hamacher. *Mathematische Lösungsverfahren für planare Standortprobleme*. Vieweg, Braunschweig, 1995.

[HK04] H.W. Hamacher and K. Klamroth. *Lineare und Netzwerk-Optimierung — a bilingual textbook*. Vieweg, 2004. ISBN 3-528-03155-7.

[HN98] H. W. Hamacher and Nickel. Classification of location models. *Locations Science*, 6:229–242, 1998.

[Hof53] A.J. Hoffman. Cycling in the simplex algorithm. Technical Report No. 2974, national Bureau of Standards, Gaithersburg, MD, 1953.

[JS04] F. Jarre and J. Stoer. *Optimierung*. Springer, 2004. ISBN 3-540-43575-1.

[LMW88] R.F. Love, J.G. Morris, and G.O. Wesolowsky. *Facilities Location*, chapter 3.3, pages 51–60. North-Holland, Amsterdam, 1988.

[NW88] G. L. Nemhauser and L. A. Wolsey. *Integer and Combinatorial Optimization*. Wiley, 1988.

[Wer03] J. Werner. Optimierung. Georg-August-Universität Göttingen, 2003. Vorlesungsskript.