Alexander Martin, Sven O. Krumke

Diskrete Optimierung

SPIN Springer's internal project number, if known

14. April 2009

Springer

Berlin Heidelberg New York
Hong Kong London
Milan Paris Tokyo

Wir widmen dieses Buch blablabla

Vorwort

Unsere goldenen Worte.

Darmstadt, Kaiserslautern,
Mai 2006

Alexander Martin
Sven O. Krumke

Inhaltsverzeichnis

Teil I

Lineare Optimierung

1

Ein erster Blick auf das Simplexverfahren

1.1 Neulich auf der Studentenparty

Auf einer Studentparty werden drei alkoholische Getränke zum Mixen mit verschiedene Preisen und mit unterschiedlichem Zuckergehalt angeboten (siehe Tabelle 1.1). Der Austauschstudent *Simple Ex*[1] hat noch 3€ übrig, die er für ein Mischgetränk aus den drei Angeboten vollständig ausgeben möchte. Da *Simple Ex* Diabetiker ist, muss er auf seinen Zuckerspiegel achten und dafür sorgen, dass er noch genau 2 Broteinheiten[2] (BE) zu sich nimmt.

	Sugarfree Obstler	Bubblegum Sprit	Maschfasch Flash
Kosten/10ml	2€	1€	2€
Zuckergehalt	0 BE	2 BE	1 BE

Tabelle 1.1. Getränke auf der Studentenparty

Wir bezeichnen mit x_1, x_2 und x_3 die Menge (in 10ml) der drei jeweiligen Getränke. *Simple Ex* erhält damit für diese Mengen das folgende lineare Gleichungssystem:

$$2x_1 + x_2 + 2x_3 = 3 \tag{1.1a}$$
$$2x_2 + x_3 = 2 \tag{1.1b}$$

Wir bringen das Gleichungssystem $Ax = b$ aus (1.1) (mit Hilfe des Gauß-Algorithmus) in eine für unsere Zwecke

[1] aus *Leppland*, kurz LP

[2] 1 Broteinheit (BE) entspricht 12 gr. an Kohlenhydraten

geeignete Form. Dazu subtrahieren wir das 1/2-fache der Gleichung (1.1b) von von (1.1a):

$$2x_1 \quad +\tfrac{3}{2}x_3 = 2 \tag{1.2a}$$
$$2x_2 + \; x_3 = 2 \tag{1.2b}$$

Anschließend multiplizieren wir beide Zeilen noch mit 1/2. Dies liefert:

$$x_1 \quad +\tfrac{3}{4}x_3 = 1 \tag{1.3a}$$
$$x_2 + \tfrac{1}{2}x_3 = 1 \tag{1.3b}$$

Die Lösungsmenge des ursprünglichen Systems (1.1) wird damit durch

$$x_1 = 1 - \tfrac{3}{4}x_3 \tag{1.4a}$$
$$x_2 = 1 - \tfrac{1}{2}x_3. \tag{1.4b}$$

beschrieben. Wir sehen, dass es beliebig viele Lösungen dieses Systems gibt, die sich mit Hilfe von x_3 parametrisieren lassen. Eine spezielle Lösung erhalten wir für $\bar{x}_3 := 0$ als $\bar{x} = (\bar{x}_1, \bar{x}_2, \bar{x}_3) = (1, 1, 0)^T$. Nachdem wir mehr als eine zulässige Lösung zur Verfügung haben, stellt sich die Frage, welche dieser Lösungen eine gegebene (lineare) Zielfunktion $c^T x = \sum_{i=1}^{3} c_i x_i$ optimiert.

Der Beste Cocktail (Teil 1)

In unserem speziellen Fall nehmen wir an, dass *Simple Ex* gerne möglichst nüchtern bleiben und daher den Akoholgehalt seines Mischgetränks minimieren möchte.[3] Sei dazu x eine beliebige zulässig Lösung für (1.1). Dann können wir unter Zuhilfenahme von (1.4) die Zielfunktion wie folgt umschreiben:

$$\begin{aligned} c^T x &= c_1(1 - \tfrac{3}{4}x_3) + c_2(1 - \tfrac{1}{2}x_3) + c_3 x_3 \\ &= \underbrace{c_1 + c_2}_{=:\alpha} + \underbrace{(-\tfrac{3}{4}c_1 - \tfrac{1}{2}c_2 + c_3)}_{=:\beta} x_3 \\ &= \alpha + \beta x_3 \end{aligned} \tag{1.5}$$

[3] Möglicherweise ist bei einer größeren Anzahl von Studenten/Studentinnen die Zielfunktion zwar die die selbe wie hier, das Ziel ist aber die Maximierung und nicht die Minimierung des Alkoholgehalts.

Mit Hilfe von (1.4) erhalten wir also die folgende äquivalente Formulierung des Problems von *Simple Ex*:

$$\min \alpha + \beta x_3 \tag{1.6a}$$
$$x_1 = 1 - \tfrac{3}{4} x_3 \tag{1.6b}$$
$$x_2 = 1 - \tfrac{1}{2} x_3 \tag{1.6c}$$

Ist $\beta = 0$, so ist die Zielfunktion auf der Menge der zulässigen Lösungen konstant gleich α (insbesondere ist dann die spezielle Lösung $\bar{x} = (1, 1, 0)^T$ optimal). Ist $\beta > 0$, so ist die Zielfunktion auf der Menge der zulässigen Lösungen nach unten unbeschränkt, da wir $x_3 < 0$ beliebig klein wählen können. Auch für $\beta < 0$ ist die Zielfunktion nach unten unbeschränkt, da hier $x_3 > 0$ beliebig groß gewählt werden kann. Unser Alkohol-Minimierungsproblem

$$\min\{\, c^T x : x \text{ erfüllt } (1.1) \,\}$$

ist damit gelöst. Lineare Optimierung über linearen Gleichungssystemen ist nicht besonders spannend: entweder sind alle zulässigen Lösungen optimal oder es gibt überhaupt keine Optimallösung (wie wir später sehen werden, gilt dies nicht nur für unser spezielles Beispiel, sondern allgemein).

Der Beste Cocktail (Teil 2)

Wenn wir unsere Modellierung der Optimierungsaufgabe von *Simple Ex* nochmal etwas genauer betrachten, stellen wir fest, dass wir wichtige Nebenbedingungen übersehen haben: die Mengen x_1, x_2, x_3 der Getränke sollten nichtnegativ sein![4] Damit sieht das Alkohol-Minimierungsproblem wie folgt aus:

$$\begin{aligned}
&\min\{\, c^T x : x \text{ erfüllt } (1.1) \text{ und } x \geq 0 \,\} \\
\Leftrightarrow \min \quad & c_1 x_1 + \quad c_2 x_2 + \quad c_3 x_3 \\
& 2x_1 + \quad x_2 + \quad 2x_3 = \quad 3 \\
& \quad 2x_2 + \quad x_3 = \quad 2 \\
& x_1 \geq 0 \quad x_2 \geq 0 \quad x_3 \geq 0
\end{aligned} \tag{1.7}$$

Abbildung **??** zeigt die Menge der zulässigen Lösungen des Optimierungsproblems (1.7).

[4] Es gibt mögliche Interpretation negativer Getränkemengen, diese mögen sich die Leser selbst vorstellen.

Unsere Überlegungen von oben, die zu (1.5) bleiben nach wie vor gültig: Wir können wieder nach x_1 und x_2 auflösen und in die Zielfunktion einsetzen, womit wir wieder eine äquivalente Formulierung des Alkohol-Minimierungsproblems (1.7) erhalten:

$$\min \alpha + \beta x_3 \tag{1.8a}$$
$$x_1 = 1 - \tfrac{3}{4}x_3 \tag{1.8b}$$
$$x_2 = 1 - \tfrac{1}{2}x_3 \tag{1.8c}$$
$$x_1 \geq 0, x_2 \geq 0, x_3 \geq 0 \tag{1.8d}$$

Die spezielle Lösung $\bar{x} = (\bar{x}_1, \bar{x}_2, \bar{x}_3) = (1, 1, 0)^T$, die wir für wir für $\bar{x}_3 := 0$ aus (1.8) erhalten, ist wieder zulässig und hat den Zielfunktionswert α (in Abbildung **??**(a) ist diese Lösung innerhalb der Menge der zulässigen Lösungen hervorgehoben). Das Einzige, das sich gegenüber der Variante ohne die Vorzeichenrestriktionen ändert, ist, dass wir unsere Fallunterscheidung nach β neu durchführen müssen.

Falls $\beta = 0$, so ist für alle Lösungen von (1.1), also insbesondere auch alle nichtnegativen Lösungen x von (1.1) weiterhin $c^T x = \alpha$. Die Lösung $\bar{x} = (1, 1, 0)^T$ mit $c^T \bar{x} = \alpha$ bleibt optimal. Für $\beta > 0$ konnten wir vorher $x_3 < 0$ beliebig klein wählen. Dies ist nun nicht mehr der Fall. In der Tat ist für jede zulässige Lösung x mit $x_3 \geq 0$ wegen (1.5) jetzt $c^T x \geq \alpha$. Damit ist $\bar{x}$ erneut als optimal nachgewiesen.

Es verbleibt der Fall, dass $\beta < 0$. Wie anfangs führt eine Erhöhung von x_3 dann zu einer Verringerung der Zielfunktion. Allerdings wird diese Erhöhung durch $x_1 \geq 0$, $x_2 \geq 0$ und (1.4) (bzw. (1.8b) und (1.8c)) begrenzt. Wegen (1.4a) muss

$$1 - 3/4 x_3 \geq 0,$$

also $x_3 \leq 4/3$ gelten. Entsprechend erhalten wir aus (1.4b) die Bedingung, dass

$$1 - \tfrac{1}{2}x_3 \geq 0,$$

also $x_3 \leq 2$. Insgesamt ist die maximale Erhöhung γ gegeben durch:

$$\gamma = \min \left\{ \frac{1}{3/4}, \frac{1}{1/2} \right\} = \frac{4}{3}. \tag{1.9}$$

Wenn wir x_3 auf $\gamma = 4/3$ erhöhen und die Werte von x_1, x_2 nach (1.4) anpassen, so ergibt sich für uns die neue

spezielle Lösung $\bar{x}^+ := (0, 1/3, 4/3)^T$ mit dem Zielfunktionswert $c^T\bar{x}^+ = \alpha + 4/3\beta$. Da $\beta < 0$ war, gilt insbesondere $c^T\bar{x}^+ < c^T\bar{x}$ und wir haben eine bessere Lösung gefunden.

In unserem konkreten Beispiel nehmen wir an, dass die Alkoholgehalte der drei Getränke $c_1 = 4$, $c_2 = 2$ und $c_3 = 3$ sind. Tabelle 1.2 fasst die Daten für die Getränke noch einmal zusammen.

	Sugarfree Obstler	Bubblegum Sprit	Maschfasch Flash
Alkoholgehalt	4	2	3
Kosten/10ml	2€	1€	2€
Zuckergehalt	0 BE	2 BE	1 BE

Tabelle 1.2. Getränke auf der Studentenparty und ihr Alkoholgehalt

Dann erhalten wir α und β für die umgeschriebene Zielfunktion nach (1.5) als:

$$\alpha = c_1 + c_2 = 4 + 2 = 6$$
$$\beta = -\tfrac{3}{4}c_1 - \tfrac{1}{2}c_2 + c_3 = -3 - 1 + 3 = -1 < 0.$$

Hier lohnt es sich demnach, den Wert der Variablen x_3 (also die gekaufte Menge von *Maschfasch Flash*) zu erhöhen. Wie wir bereits in (1.9) ausgerechnet hatten, können wir x_3 maximal auf den Wert $4/3$ erhöhen und wir erhalten die neue Lösung $\bar{x}^+ := (0, 1/3, 4/3)^T$ mit dem Zielfunktionswert

$$c^T\bar{x}^+ = \alpha + 4/3\beta = 6 - 4/3 = 17/3.$$

Abbildung **??**(b) zeigt $\bar{x}^+$ innerhalb der Menge der zulässigen Lösungen als dicken Punkt. Wie können wir jetzt weiterarbeiten? Ist $\bar{x}^+$ bereits eine optimale Lösung?

Was die Vorgehensweise bei $\bar{x}$ so einfach machte, war, dass wir die zwei Variablen x_1 und x_2 in Abhängigkeit der Variablen x_3 ausgedrückt hatten, wobei x_3 in $\bar{x} = (1, 1, 0)^T$ den Wert 0 besass. Im Prinzip vollzieht sich beim Übergang von $\bar{x}$ zu $\bar{x}^+$ ein Rollentausch der Variablen x_1 und x_3: x_1 erhält den Wert 0 und x_3 wird auf $4/3$ erhöht. Wir stellen daher das System der linearen Gleichungen so um, dass x_2 und x_3 in Abhängigkeit von x_1 ausgedrückt werden (der genaue Rechenweg ist hier momentan nicht

ganz so interessant, wir werden im nächsten Abschnitt das systematische Vorgehen genauer untersuchen):

$$x_3 = \tfrac{4}{3} - \tfrac{4}{3}x_1 \tag{1.10a}$$
$$x_2 = \tfrac{1}{3} + \tfrac{2}{3}x_1. \tag{1.10b}$$

Für jede zulässige Lösung $x = (x_1, x_2, x_3)^T$ gilt also:

$$\begin{aligned} c^T x &= c_1 x_1 + c_2(\tfrac{1}{3} + \tfrac{2}{3}x_1) + c_3(\tfrac{4}{3} - \tfrac{4}{3}x_1) \\ &= 4x_1 + 2(\tfrac{1}{3} + \tfrac{2}{3}x1) + 3(\tfrac{4}{3} - \tfrac{4}{3}x_1) \\ &= \tfrac{17}{3} + (4 + \tfrac{4}{3} - 4)x_1 \\ &= \tfrac{17}{3} + \tfrac{1}{3}x_1. \end{aligned} \tag{1.11}$$

Außerdem erfüllt jede zulässige Lösung $x = (x_1, x_2, x_3)^T$ auch die Vorzeichenbedingung $x_3 \geq 0$. Daher gilt dann wegen (1.11) auch $c^T x \geq 17/3 = c^T \bar{x}^+$ für jede zulässige Lösung x. Die spezielle Lösung $\bar{x}^+ = (0, 1/3, 4/3)^T$ ist damit eine Optimallösung und unsere Vorgehensweise liefert dafür einen mathematischen Beweis!

1.2 Die Grundform des Simplex-Verfahrens

Wir verallgemeinern unser Beispiel von oben nun auf beliebige lineare Gleichungsrestriktionen

$$Ax = b, \tag{1.12}$$

wobei A eine $m \times n$-Matrix und $b \in \mathbb{R}^m$ ist. Wir nehmen zunächst an, dass $\operatorname{Rang} A = m$ (und damit insbesondere $m \leq n$) ist. Dadurch ist das System (1.12) überbestimmt und es gibt Optimierungspotential. In unserem Beispiel oben war $n = 3$ und $m = 2$ (siehe (1.1)):

$$A = \begin{pmatrix} 2 & 1 & 2 \\ 0 & 2 & 2 \end{pmatrix}, \qquad b = \begin{pmatrix} 3 \\ 2 \end{pmatrix} \tag{1.13}$$

Wegen $\operatorname{Rang} A = m$ gibt es eine $m \times m$-Teilmatrix A' von A, die nichtsingulär ist. Die Indizes der Spalten dieser Teilmatrix A' und ihre Reihenfolge spielen eine besondere Rolle. Wir führen daher folgende Notation ein:

Notation 1.1. Für eine $m \times n$-Matrix A mit Spalten $A_{\cdot 1}, \ldots, A_{\cdot n}$ und ein Tupel $J = (j_1, \ldots, j_k)$ von paarweise verschiedenen Spaltenindizes $j_i \in \{1, \ldots, n\}$ sei

$A_{\cdot J} = (A_{\cdot j_1}, \ldots, A_{\cdot j_k})$ die Matrix, die aus den Spalten mit Indizes in J (in dieser Reihenfolge) entsteht. Wir schreiben dann auch $|J| := k$ für die Anzahl der Indizes in J und $J \subseteq \{1, \ldots, n\}$.

Definition 1.2 (Basis, Basismatrix). *Ist $|B| = m$ und $A_{\cdot B}$ nichtsingulär, so nennen wir nennen wir B eine* Basis *von A und $A_{\cdot B}$* Basismatrix.

Basis

Basismatrix

Im Alkohol-Minimierungsproblem von *Simple Ex* ist beispielsweise ist $B = (1, 2)$ eine Basis und $A' = A_{\cdot B} = \left(\begin{smallmatrix} 2 & 1 \\ 0 & 1 \end{smallmatrix}\right)$ eine Basismatrix.

Ist B eine Basis von A, so können wir jeden Vektor $x \in \mathbb{R}^n$ in die *Basisvariablen* x_B und die *Nicht-Basisvariablen* x_N partitionieren: $x = \binom{x_B}{x_N}$. Es gilt dann:

Basisvariablen

Nicht-Basisvariablen

$$\begin{aligned} Ax = b &\Leftrightarrow A_{\cdot B}x_B + A_{\cdot N}x_N = b \\ &\Leftrightarrow A_{\cdot B}x_B = b - A_{\cdot N}x_N \\ &\Leftrightarrow x_B = A_{\cdot B}^{-1}(b - A_{\cdot N}x_N). \end{aligned} \tag{1.14}$$

Insbesondere ist $\bar{x} = \binom{\bar{x}_B}{\bar{x}_N} = \binom{A_{\cdot B}^{-1}b}{0}$ eine spezielle Lösung des Systems $Ax = b$ in (1.12).

Definition 1.3 (Basislösung). *Ist B eine Basis von A, so heisst der Vektor $\bar{x} = \binom{\bar{x}_B}{\bar{x}_N}$ mit*

$$\begin{aligned} \bar{x}_B &:= A_{\cdot B}^{-1}b \\ \bar{x}_N &:= 0 \end{aligned}$$

die zur Basis B gehörende Basislösung.

Basislösung

Im Beispiel des Alkohol minimierenden Studenten *Simple Ex* ist $B = (1, 2)$ eine Basis mit zugehöriger Basislösung $\bar{x} = (\bar{x}_1, \bar{x}_2, \bar{x}_3) = (1, 1, 0)^T$. Wie in (1.5) können wir die Zielfunktion für eine Lösung $x = \binom{x_B}{x_N}$ von (1.12) umschreiben:

$$\begin{aligned} c^T x &= c_B^T x_B + c_N^T x_N \\ &\overset{(1.14)}{=} c_B^T A_{\cdot B}^{-1}(b - A_{\cdot N}x_N) + c_N^T x_N \\ &= c_B^T A_{\cdot B}^{-1} b + \underbrace{(c_N^T - c_B^T A_{\cdot B}^{-1} A_{\cdot N})}_{=:\bar{z}_N^T} x_N \\ &= \begin{pmatrix} c_B \\ c_N \end{pmatrix}^T \begin{pmatrix} A_{\cdot B}^{-1} b \\ 0 \end{pmatrix} + \bar{z}_N^T x_N \\ &= c^T \bar{x} + \bar{z}_N^T x_N, \end{aligned} \tag{1.15}$$

wobei $\bar{x}$ die zu B gehörende Basislösung ist. Der Vektor

$$\bar{z}_N = c_N - A_{\cdot N}^T A_{\cdot B}^{-T} c_B$$

heißt Vektor der *reduzierten Kosten* (oder einfach nur *reduzierte Kosten*).

reduzierte Kosten

Wie in unserem Alkohol-Minimierungsproblem ergeben sich unterschiedliche Fälle in Abhängigkeit von $\bar{z}_N$. Ist $\bar{z}_N = 0$ der Nullvektor, so ist die Zielfunktion auf der Menge der Lösungen von (1.12) konstant gleich $c^T\bar{x}$. Insbesondere ist dann die Basislösung $\bar{x}$ optimal.

Sei daher nun $\bar{z}_N \neq 0$, also $\bar{z}_j \neq 0$ für ein $j \in N$. Wir betrachten für ein Skalar t den Vektor $x(t)$, definiert durch

$$\begin{aligned} x_j(t) &:= t \\ x_i(t) &:= 0 \text{ für alle } i \in N \setminus \{j\} \\ x_B(t) &:= A_{\cdot B}^{-1}(b - A_{\cdot N} x_N(t)). \end{aligned}$$

Offenbar gilt $Ax(t) = b$ und nach unserer Rechnung in (1.15) gilt daher:

$$c^T x(t) = c^T\bar{x} + \bar{z}_N^T x_N(t) = c^T\bar{x} + \bar{z}_j t. \tag{1.16}$$

Gilt $\bar{z}_j > 0$, so ist die Zielfunktion auf der Menge der zulässigen Lösungen nach unten unbeschränkt, da wir $x_j(t) = t < 0$ beliebig klein wählen können. Analog ist die Zielfunktion auch nach unten unbeschränkt, falls $\bar{z}_j < 0$ für ein $j \in N$, da hier $x_j(t) = t > 0$ beliebig groß gewählt werden kann. Damit ergibt sich im allgemeinen Fall die gleiche Situation wie im Alkohol-Minimierungsproblem. Das Optimierungsproblem

$$\min \{\, c^T x : Ax = b \,\}$$

ist entweder unbeschränkt oder die Zielfunktion ist auf der Menge $\{\, x : Ax = b \,\}$ der zulässigen Lösungen konstant. Beide Fälle sind nicht sonderlich spannend.

1.2.1 Lineare Programme

Im Alkohol-Minimierungsproblem hatten wir als nächstes die (für die konkrete Anwendung überaus sinnvollen) Vorzeichenrestriktionen $x \geq 0$ eingeführt. Im allgemeinen Fall führt dies nun auf das Optimierungsproblem

$$\begin{aligned} \min \quad & c^T x && \text{(1.17a)} \\ & Ax = b && \text{(1.17b)} \\ & x \geq 0 && \text{(1.17c)} \end{aligned}$$

Definition 1.4 (Lineares Programm in Standardform). *Ein lineares Programm der Form (1.17) nennen wir auch* lineares Programm in Standardform. lineares Programm in Standardform

Für die Basislösungen ergibt sich eine neue Situation, da nun nicht mehr jede solche Lösung automatisch zulässig ist. Beispielsweise liefert die Basis $B = (1, 3)$ im Alkohol-Minimierungsproblem aus

$$x_B = \begin{pmatrix} x_1 \\ x_3 \end{pmatrix} = A_{\cdot B}^{-1} b = \begin{pmatrix} 2 & 2 \\ 0 & 1 \end{pmatrix}^{-1} \begin{pmatrix} 3 \\ 2 \end{pmatrix} = \begin{pmatrix} -\frac{1}{2} \\ 2 \end{pmatrix}$$

die Basislösung $(-1/2, 0, 2)^T$, die negative Einträge besitzt.

Definition 1.5 ((Primal) zulässige Basis). *Eine Basis B heisst* (primal) zulässige Basis *und die zugehörige Basislösung $\binom{x_B}{x_N}$ heisst* zulässige Basislösung*, falls $x_B \geq 0$ gilt.* (primal) zulässige Basis; zulässige Basislösung

Wie wir bereits im Beispiel weiter vorne gesehen haben, ist die Basis $B = (1, 2)$ zulässig, da die zugehörige Basislösung $\bar{x} = (1, 1, 0)^T$ nichtnegativ ist.

Wir gehen zunächst davon aus, dass wir eine (primal) zulässige Basis B bereits gegeben haben. Wie wir eine solche erhalten, ist Thema eines späteren Abschnitts. Sei $\bar{x}$ die zugehörige zulässige Basislösung. Wie im Beispiel bleiben unsere Überlegungen, die zu (1.15) führten nach wie vor gültig, nur die Fallunterscheidung bezüglich $\bar{z}_N$ muss neu überdacht werden.

Falls $\bar{z}_N = 0$, so ist die Zielfunktion wieder konstant auf der Menge aller zulässigen Lösungen und die Basislösung $\bar{x}$ ist insbesondere optimal. Ist allgemeiner $\bar{z}_N \geq 0$, so gilt wegen $x_N \geq 0$ für jede zulässige Lösung $x = \binom{x_B}{x_N}$ von (1.17):

$$c^T x \overset{(1.15)}{=} c^T \bar{x} + \underbrace{\bar{z}_N^T}_{\geq 0} \underbrace{x_N}_{\geq 0} \geq c^T \bar{x}. \tag{1.18}$$

Daher folgt aus $\bar{z}_N \geq 0$ wieder die Optimalität der Basislösung $\bar{x}$. Ist andererseits $\bar{z}_j < 0$ für ein $j \in N$, so bringt nach (1.16) jede Erhöhung des Wertes von x_j eine Verbesserung der Zielfunktion.

Wir wollen x_j nun soweit wie möglich erhöhen, so dass der resultierende Vektor noch zulässig ist. Dabei lassen

wir die anderen Nichtbasiseinträge in $N \setminus \{j\}$ auf 0 fixiert. Wegen (1.14) gilt:

$$\begin{aligned} x_B &= A_{\cdot B}^{-1} b - A_{\cdot B}^{-1} A_{\cdot N} x_N \\ &= \underbrace{A_{\cdot B}^{-1} b}_{=\bar{x}_B} - A_{\cdot B}^{-1} A_{\cdot j} x_j - A_{\cdot B}^{-1} A_{\cdot N\setminus\{j\}} \underbrace{x_{N\setminus\{j\}}}_{=0} \qquad (1.19) \\ &= \bar{x}_B - A_{\cdot B}^{-1} A_{\cdot j} x_j. \qquad (1.20) \end{aligned}$$

Der Einfluss von x_j auf die Basisvariablen x_B wird daher durch den Vektor

$$w = A_{\cdot B}^{-1} A_{\cdot j} \qquad (1.21)$$

gesteuert. Gilt $w \leq 0$, so können wir x_j beliebig erhöhen ohne die Zulässigkeit zu verlieren, da invariant $x \geq 0$ gilt. Da zusätzlich $\bar{z}_j < 0$ gilt, wird dadurch auch die Zielfunktion beliebig verringert (vgl. (1.15)), d.h. das Optimierungsproblem (1.17) ist nach unten unbeschränkt.

Nehmen wir deshalb $w \not\leq 0$ an und betrachten diejenigen Indizes $i \in B$ mit $w_i > 0$. Damit x zulässig bleibt, muss $x_B \geq 0$ bleiben, d.h. wir können x_j maximal soweit erhöhen, bis eine der Variablen aus x_B die Null erreicht. Damit x zulässig bleibt, muss $x_B \geq 0$ bleiben:

$$x_B = \bar{x}_B - x_j w \geq 0.$$

Für alle Indizes $i \in B$ mit $w_i > 0$ muss also gelten:

$$x_j w_i \leq \bar{x}_i \Leftrightarrow x_j \leq \frac{\bar{x}_i}{w_i}.$$

Die maximale Erhöhung γ lässt sich in folgender Form schreiben:

$$\begin{aligned} \gamma &= \min \left\{ \frac{\bar{x}_i}{w_i} : w_i > 0 \text{ und } i \in B \right\} \qquad (1.22a) \\ &= \min \left\{ \frac{(A_{\cdot B}^{-1} b)_i}{w_i} : w_i > 0 \text{ und } i \in B \right\}. \qquad (1.22b) \end{aligned}$$

Wir betrachten nun den entsprechenden Vektor $\bar{x}^+$ mit $\bar{x}_j^+ := \gamma$ und $\bar{x}_B^+ := \bar{x} - \gamma w$, den wir erhalten haben. Es gilt nach Konstruktion:

$$\begin{aligned} A\bar{x}^+ &= b \qquad (1.23a) \\ \bar{x}^+ &\geq 0 \qquad (1.23b) \\ c^T \bar{x}^+ &= c^T \bar{x} + \gamma \bar{z}_j \leq c^T \bar{x}, \qquad (1.23c) \end{aligned}$$

wobei in (1.23c) sogar strikte Ungleichung gilt, falls $\gamma > 0$ gewählt werden konnte, da $\bar{z}_j < 0$ ist (wir werden später noch sehen, dass dem Fall $\gamma = 0$ eine besondere Bedeutung innerhalb des Simplex-Verfahrens zukommt).

Ist nun $\bar{x}^+$ eine optimale Lösung unseres Optimierungsproblems (1.17)? Um dies zu festzustellen bzw. um eine bessere Lösung zu finden würden wir gerne mit $\bar{x}^+$ genauso verfahren, wie wir dies bei $\bar{x}$ getan haben (vgl (1.18)). Allerdings ist nicht klar, wie dies geschehen soll. Was die Vorgehensweise bei $\bar{x}$ so einfach machte, war, dass $\bar{x}$ eine *Basislösung* (zur Basis B) ist und wir dadurch die Basisvariablen in Abhängigkeit von den Nichtbasisvariablen ausdrücken konnten (1.14). Dies ermöglichte und dann das Umschreiben der Zielfunktion in (1.15) und das Finden von $\bar{x}^+$.

Im Alkohol-Minimierungsproblem von *Simple Ex* hatten wir einfach die Rollen der beiden Variablen x_1 und x_3 beim Übergang von $\bar{x} = (1, 1, 0)^T$ zu $\bar{x}^+ = (0, 1/3, 4/3)^T$ vertauscht, da x_1 in der neuen Lösung $\bar{x}^+$ den Wert 0 erhielt. Das folgende Lemma zeigt, dass diese Vorgehensweise auch im allgemeinen Fall gilt: Wenn wir die neue Variable x_j in die Basis aufnehmen und eine beliebige Variable x_i aus der Basis entfernen, die in $\bar{x}^+$ den Wert 0 hat, so haben wir wieder eine Basis und $\bar{x}^+$ ist die zugehörige Basislösung.

Lemma 1.6. *Sei $i \in B$ beliebig mit $w_i > 0$ sowie $\bar{x}^+_i = 0$. Sei B^+ definiert als $B^+ := B \setminus \{i\} \cup \{j\}$ (d.h. B^+ entsteht aus B, indem wir den Index $i \in B$ durch den Index $j \in N$ ersetzen). Dann ist B^+ eine zulässige Basis von A und der Vektor $\bar{x}^+$ die zugehörige Basislösung.*

Beweis. Sei $N^+ := N \setminus \{j\} \cup \{i\}$. Wir müssen folgende Eigenschaften zeigen:

(i) $A_{\cdot B^+}$ ist nichtsingulär.
(ii) $\bar{x}^+_{B^+} \geq 0$ und $\bar{x}^+_{N^+} = 0$.
(iii) $\bar{x}^+_{B^+} = A^{-1}_{\cdot B^+} b$.

Wir zeigen zunächst, dass mit (i) die beiden Eigenschaften (ii) und (iii) folgen.

Die Aussage $\bar{x}^+_{N^+} = 0$ ist offensichtlich nach Konstruktion, da nach Voraussetzung $\bar{x}^+_i = 0$ und alle Nichtbasisvariablen in $N \setminus \{j\}$ den Wert Null behalten. Wegen $\bar{x}^+_j = \gamma$ und $\gamma \geq 0$, folgt $\bar{x}^+_j \geq 0$. Wir müssen also nur noch zeigen, dass $\bar{x}^+_k \geq 0$ für alle $k \in B^+ \setminus \{j\} = B \setminus \{i\}$. Für diese k gilt nach (1.20) und (1.21):

$$\bar{x}_k^+ = \bar{x}_k - \gamma w_k.$$

Ist $w_k \leq 0$, dann gilt $\bar{x}_k^+ \geq 0$, da $\bar{x}_k \geq 0$. Falls $w_k > 0$ so haben wir nach der Wahl von γ in (1.22) die Ungleichung $\gamma \leq \bar{x}_k/w_k$ und damit

$$\bar{x}_k^+ = \bar{x}_k - \gamma w_k \overset{(1.22)}{\geq} \bar{x}_k - \frac{\bar{x}_k}{w_k} w_k = 0. \tag{1.24}$$

Dies zeigt (ii).

Die Eigenschaft (iii) ergibt sich wie folgt. Nach Konstruktion von $\bar{x}^+$ in (1.20) gilt $A\bar{x}^+ = b$. Da nach (ii) $\bar{x}_{N^+}^+ = 0$ und nach (i) $A_{.B^+}$ regulär ist, folgt $\bar{x}_{B^+}^+ = A_{.B^+}^{-1} b$ aus $b = A_{.B^+}\bar{x}_{B^+}^+ + A_{.N^+}\bar{x}_{N^+}^+ = A_{.B^+}\bar{x}_{B^+}^+$.

Es verbleibt, die Nichtsingularität von $A_{.B^+}$ zu zeigen. Angenommen, $A_{.B^+}$ wäre singulär. Da $A_{.B}$ regulär war und $B^+ \setminus \{j\} = B \setminus \{i\}$, sind die Spalten aus $A_{.B^+\setminus\{j\}}$ linear unabhängig. Daher folgt, dass $A_{.j}$ eine Linearkombination der Spalten aus $A_{.B^+\setminus\{j\}}$ sein muss und es (wegen der linearen Unabhängigkeit der Spalten aus $A_{.B^+\setminus\{j\}}$) einen eindeutigen Vektor λ gibt, so dass $A_{.B^+\setminus\{j\}}\lambda = A_{.j}$. Setzen wir $\bar{\lambda}_i := 0$ und $\bar{\lambda}_k := \lambda_k$ sonst, so haben wir dann $A_{.B}\bar{\lambda} = A_{.j}$, also $\bar{\lambda} = A_{.B}^{-1}A_{.j} = w$, wobei w wie in (1.21) definiert ist. Dann ist aber $w_i = \bar{\lambda}_i = 0$ im Widerspruch zur Voraussetzung $w_i \neq 0$ (wegen $w_i > 0$). Dies zeigt (i). □

Aus dem Beweis des obigen Lemmas ergibt sich eine wichtige Konsequenz, die wir später in Abschnitt **??** verwenden werden. Die einzige Stelle, an der wir $w_i > 0$ benutzt haben, ist um in (1.24) die Eigenschaft $\bar{x}_k^+ \geq 0$ zu zeigen. Gilt $w_i < 0$ und gilt für

$$\gamma' := \min\left\{\frac{\bar{x}_i}{w_i} : w_i \neq 0 \text{ und } i \in B\right\} \tag{1.25}$$

die Bedingung $\gamma' = 0$, so bleiben alle Aussagen von Lemma 1.6 auch für $\gamma := \gamma' = 0$ und jedes $i \in B$ mit $w_i < 0$ richtig, da sich der Wert keiner Variablen beim Übergang von B zu B^+ ändert.

Beobachtung 1.7. *Falls wir γ gemäß (1.25) definieren, so gilt Lemma 1.6 für alle $i \in B$ mit $w_i \neq 0$.*

Lemma 1.6 ermöglicht nun eine Fortsetzung des Verfahrens mit B^+ und $\bar{x}^+$ anstelle von B und $\bar{x}$. Das resultierende Verfahren ist in Algorithmus 1.1 noch einmal zusammengefasst:

Algorithmus 1.1 : Grundform des Simplex-Verfahrens

Input : Eine primal zulässige Basis $B = (B_1, \ldots, B_m)$
Output : Eine Optimallösung $\bar{x}$ von (1.17) oder die Information, dass (1.17) unbeschränkt ist.

1 **BTRAN:** `/* Backward TRANsformation */`
Löse $\bar{y}^T A_{\cdot B} = c_B^T$;

2 **Pricing:** `/* Auspreisen */`
Berechne $\bar{z}_N = c_N - A_{\cdot N}^T \bar{y}$;
if $\bar{z}_N \geq 0$ **then**
stop, B ist optimale Basis und $\bar{x} = \binom{\bar{x}_B}{\bar{x}_N}$ ist eine Optimallösung;
end
Wähle ein $j \in N$ mit $\bar{z}_j < 0$; `/*` x_j `heißt` *`eintretende Variable (engl.`* `entering variable)` `*/`

3 **FTRAN:** `/* Forward TRANsformation */`
Löse $A_{\cdot B} w = A_{\cdot j}$;

4 **Ratio-Test:** `/* Quotiententest */`
if $w \leq 0$ **then**
stop, das Problem (1.17) ist unbeschränkt;
end
Berechne

$$\gamma = \frac{\bar{x}_{B_i}}{w_i} = \min\left\{ \frac{\bar{x}_{B_k}}{w_k} : w_k > 0 \text{ und } k \in \{1, \ldots, m\} \right\}$$

wobei $i \in \{1, 2, \ldots, m\}$ und $w_i > 0$ ist ; `/*` $\bar{x}_i$ `heißt die` *`die Basis verlassende Variable (engl.`* `leaving). */`

5 **Update:** `/* Aktualisieren */`
Setze

$$\begin{aligned} \bar{x}_B &= \bar{x}_B - \gamma w \\ N &= N \setminus \{j\} \cup \{B_i\} \\ B_i &= j \\ \bar{x}_j &= \gamma \end{aligned}$$

und **goto** Schritt 1 ; `/* weiter bei BTRAN */`

Beispiel 1.8. Als Beispiel betrachten wir folgendes lineare Programm:

$$\begin{aligned} \min \quad & -3x_1 - 2x_2 - 2x_3 \\ & x_1 \qquad + x_3 \leq 8 \\ & x_1 + x_2 \qquad \leq 7 \\ & x_1 + 2x_2 \qquad \leq 12 \\ & x_1, \quad x_2, \quad x_3 \geq 0 \end{aligned}$$

Die Menge der zulässigen Lösungen ist in Abbildung **??** dargestellt. Wir bringen das LP in Standardform, indem wir Schlupfvariablen x_4, x_5, x_6 einführen:

$$\begin{aligned} \min \quad & -3x_1 - 2x_2 - 2x_3 \\ & x_1 \qquad + x_3 + x_4 \qquad\qquad = 8 \\ & x_1 + x_2 \qquad\qquad + x_5 \qquad = 7 \\ & x_1 + 2x_2 \qquad\qquad\qquad + x_6 = 12 \\ & x_1, \quad x_2, \quad x_3, \quad x_4, \quad x_5, \quad x_6 \geq 0 \end{aligned}$$

Eine primal zulässige Basis ist $B = (4, 5, 6)$ und die zugehörige Basislösung ist gegeben durch

$$\bar{x}_B = \begin{pmatrix} \bar{x}_4 \\ \bar{x}_5 \\ \bar{x}_6 \end{pmatrix} = A_{\cdot B}^{-1} b = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}^{-1} \begin{pmatrix} 8 \\ 7 \\ 12 \end{pmatrix} = \begin{pmatrix} 7 \\ 8 \\ 12 \end{pmatrix}$$

$$\bar{x}_N = \begin{pmatrix} \bar{x}_1 \\ \bar{x}_2 \\ \bar{x}_3 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix},$$

d.h. alle Schlupfvariablen sind in der Basis und der Zielfunktionswert ist

$$c^T \bar{x} = c_B^T \bar{x}_B = (0, 0, 0) \begin{pmatrix} 7 \\ 8 \\ 12 \end{pmatrix} = 0.$$

Die einzelnen Teilschritte des Algorithmus 1.1 sehen dann wie folgt aus:

(1) **BTRAN**: Löse $\bar{y}^T A_{\cdot B} = c_B^T$, d.h.

$$\bar{y}^T \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} = (0, 0, 0) \quad \Rightarrow \bar{y}^T = (0, 0, 0)^T.$$

(2) **Pricing**: Berechne $\bar{z}_N = c_N - A_{\cdot N}^T \bar{y}$.

$$\bar{z}_N = \begin{pmatrix} -3 \\ -2 \\ -2 \end{pmatrix} - \begin{pmatrix} 1 & 1 & 1 \\ 0 & 1 & 2 \\ 1 & 0 & 0 \end{pmatrix} \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix} = \begin{pmatrix} -3 \\ -2 \\ -2 \end{pmatrix}$$

Für $j = 1$ gilt $\bar{z}_j = -3 < 0$.

(3) **FTRAN**: Löse $A_{\cdot B}w = A_{\cdot j}$.

$$\begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} w = \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix} \qquad \Rightarrow w = \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix}$$

(4) **Ratio-Test**: Berechne $\gamma = \min\left\{ \frac{\bar{x}_{B_k}}{w_k} : w_k > 0 \text{ und } k \in \{1, \ldots, m\} \right\}$.

$$\gamma = \min\left\{ \frac{8}{1}, \frac{7}{1}, \frac{12}{1} \right\} = 7 = \frac{\bar{x}_5}{w_2} = \frac{\bar{x}_{B_i}}{w_i}.$$

mit $i = 2$ und $B_2 = 5$.

(5) **Update**:

$$\bar{x}_B = \begin{pmatrix} \bar{x}_4 \\ \bar{x}_5 \\ \bar{x}_6 \end{pmatrix} - \gamma w = \begin{pmatrix} 8 \\ 7 \\ 12 \end{pmatrix} - 7 \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \\ 5 \end{pmatrix}$$

$N := N \setminus \{j\} \cup \{B_i\} = \{1, 2, 3\} \setminus \{1\} \cup \{5\} = \{2, 3, 5\}$

$B_2 := 1$

$\bar{x}_1 := \gamma = 7$

Wir erhalten die neue Basislösung $\bar{x}$ zur Basis $B = (4, 1, 6)$:

$$\bar{x}_B = \begin{pmatrix} \bar{x}_4 \\ \bar{x}_1 \\ \bar{x}_6 \end{pmatrix} = \begin{pmatrix} 1 \\ 7 \\ 5 \end{pmatrix}, \quad \bar{x}_N = \begin{pmatrix} \bar{x}_2 \\ \bar{x}_3 \\ \bar{x}_5 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}.$$

Der Zielfunktionswert erniedrigt sich um $\gamma \bar{z}_j = 7 \cdot (-3) = -21$, so dass jetzt gilt:

$$c^T \bar{x} = -21.$$

Iteration	Basis	Zielfunktions-wert	Eintretende Variable	Verlassende Variable
1	$B = (4, 5, 6)$	0	x_1	x_5
2	$B = (4, 1, 6)$	-21	x_3	x_4
3	$B = (3, 1, 6)$	-23	x_2	x_6
4	$B = (3, 1, 2)$	-28		

Tabelle 1.3. Simplex-Algorithmus auf dem LP aus Beispiel 1.8.

Die Fortsetzung des Algorithmus 1.1 liefert die Ergebnisse aus Tabelle 1.3. Nach der dritten Iteration gilt $B = (3, 1, 2)$, $N = \{4, 5, 6\}$. Die aktuelle Basislösung ist

$$\bar{x}_B = \begin{pmatrix} \bar{x}_3 \\ \bar{x}_1 \\ \bar{x}_2 \end{pmatrix} = \begin{pmatrix} 6 \\ 2 \\ 5 \end{pmatrix}, \qquad \bar{x}_N = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}.$$

In der vierten Iteration sind dann die einzelnen Teilschritte wie folgt:

(1) **BTRAN**: Löse $\bar{y}^T A_{\cdot B} = c_B^T$, d.h.

$$\bar{y}^T \begin{pmatrix} 1 & 1 & 0 \\ 0 & 1 & 1 \\ 0 & 1 & 2 \end{pmatrix} = (-2, -3, -2) \quad \Rightarrow \bar{y}^T = (-2, 0, -1).$$

(2) **Pricing**: Berechne $\bar{z}_N = c_N - A_{\cdot N}^T \bar{y}$.

$$\bar{z}_N = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix} - \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} -2 \\ 0 \\ 1 \end{pmatrix} = \begin{pmatrix} 2 \\ 0 \\ 1 \end{pmatrix}$$

Es gilt nun $\bar{z}_N \geq 0$ und der Algorithmus terminiert mit der Optimallösung $x^* = (2, 5, 6, 0, 0, 0)^T$.

Wir halten das folgende (partielle) Ergebnis über die Korrektheit des Simplex-Verfahrens fest:

Theorem 1.9. *Terminiert Algorithmus 1.1, so liefert er das korrekte Ergebnis.*

Beweis. Der Algorithmus terminiert entweder in Schritt 2 mit einer Lösung oder in Schritt 4 mit der Auskunft, dass das Problem unbeschränkt ist.

Stoppt das Verfahren in Schritt 2 mit der aktuellen Basislösung $\bar{x}$, so gilt $\bar{z}_N \geq 0$. Wie wir bereits in (1.18) gezeigt haben, gilt für jede zulässige Lösung x des Linearen Programms (1.17): $c^T x = c^T \bar{x} + \bar{z}_N^T x_N \geq c^T \bar{x}$. Somit ist die Basislösung $\bar{x}$ auch tatsächlich eine optimale Lösung.

Falls das Verfahren in Schritt 4 terminiert, so ist der Vektor $w = A_{\cdot B}^{-1} A_{\cdot j}$, der im FTRAN-Schritt berechnet wird nichtpositiv: $w \leq 0$. Wie wir bei der Herleitung in (1.21) gezeigt haben, ist in diesem Fall das Optimierungsproblem (1.17) nach unten unbeschränkt. □

Der Simplex-Algorithmus 1.1 liefert beim Abbruch entweder eine Optimallösung oder die korrekte Information, dass das Optimierungsproblem unbeschränkt ist. Aber terminiert der Algorithmus immer? Dies ist eine der Fragen, mit denen wir uns noch beschäftigen müssen. Insgesamt müsen wir noch folgende Punkte klären, um das Verfahren zu einem vollständigen Optimierungsalgorithmus zu machen:

Terminieren: Bricht der Simplex-Algorithmus immer ab oder kann es vorkommen, dass unendlich viele Iterationen durchgeführt werden?

Initialisierung: Wie finden wir eine erste zulässige Basis?
In unserer bisherigen Grundversion des Simplex-Algorithmus sind wir davon ausgegangen, dass zu Beginn bereits eine primal zulässige Basis vorliegt. Im Allgemeinen kann man aber eine zulässige Basis nicht einfach ablesen bzw. es ist überhaupt nicht klar, ob eine solche Basis überhaupt existiert (man denke nur an den Fall, dass das Lineare Programm keine zulässigen Lösungen besitzt).

1.3 Standardform von Linearen Programmen, eine Einschränkung?

Bisher haben wir uns mit Linearen Programmen in der Standardform

$$\begin{aligned} \min \quad & c^T x \\ & Ax = b \\ & x \geq 0 \end{aligned}$$

beschäftigt. Wir zeigen in diesem Abschnitt kurz, dass man jedes „allgemeine Lineare Programm“ auf diese Standardform bringen kann. Im Prinzip genügt es daher dann, sich mit Lösungsmethoden für Lineare Programme in Standardform zu beschäftigen. Es sollte aber erwähnt werden, dass man in der Praxis durchaus die Struktur von bestimmten Linearen Programmen ausnutzen kann, um spezielle Formen der Simplex-Methode zu erhalten, die möglichst effizient arbeiten.

Minimieren und Maximieren

Da für jede Menge S die Beziehung:

$$\min\{c^T x : x \in S\} = -\max\{-c^T x : x \in S\}$$

gilt, sind Minimierungs- und Maximierungsprobleme äquivalent. Wir können uns also ohne Einschränkung auf Lineare Programme beschränken, bei denen die Zielfunktion minimiert wird.

Lineare Programme in allgemeiner Form

Wir betrachten ein Lineares Programm in allgemeiner Form, d.h. mit linearen Ungleichungen und Gleichungen sowie vorzeichenbeschränkten und freien Variablen, bei dem die Zielfunktion minimiert wird:

$$\begin{aligned} \min \quad & c^T x + d^T y && && (1.26a)\\ & Ax + By && = b && (1.26b)\\ & Cx + Dy && \leq d && (1.26c)\\ & Ex + Fy && \geq e && (1.26d)\\ & x \geq 0 && && (1.26e) \end{aligned}$$

Die Variablen x, bei denen wir Nichtnegativität $x \geq 0$ fordern, nennt man *vorzeichenbeschränkte Variablen*, die y Variablen heißen *freie Variablen*.

Schlupf- und Überschussvariablen

Zunächst beschäftigen wir uns mit den Ungleichungen $Cx + Dy \leq d$ und $Ex + Fy \geq e$ und zeigen, wie wir sie durch Einführen von neuen Variablen in Gleichungsrestriktionen überführen können.

Für eine Ungleichung der Form

$$C_{i\cdot}x + D_{i\cdot}y \leq d_i$$

können wir eine neue vorzeichenbeschränkte Variable $s_i \geq 0$ einführen, und die Ungleichung dann äquivalent als:

$$C_{i\cdot}x + D_{i\cdot}y + s_i = d_i, \quad s_i \geq 0$$

schreiben. Die Variable s_i nennt man *Schlupfvariable*. Sie misst quasi den freien Platz oder Schlupf bis zur oberen Schranke d_i. Fassen wir die s_i-Variablen in einem Vektor $s \geq 0$ zusammen, so erhalten wir die Gleichungsrestriktionen

$$Cx + Dy + s = d.$$

Analog können wir für jede Ungleichung

$$E_{i\cdot}x + F_{i\cdot}y \geq e_i$$

eine Schlupfvariable $u_i \geq 0$ einführen, so dass wir die Ungleichungen $Ex + Fy \geq e$ äquivalent als

$$Ex + Fy - u = e$$

formulieren können. Manchmal nennt man die u-Variablen auch *Überschussvariablen*, da u_i den Überschuss bis zur unteren Schranke in der $i$ten Restriktion misst.

Mit Hilfe der Schlupfvariablen erhalten wir folgende äquivalente Form von (1.26)

$$\begin{aligned} \min \quad & c^T x + d^T y && && (1.27a)\\ & Ax + By && = b && (1.27b)\\ & Cx + Dy + s && = d && (1.27c)\\ & Ex + Fy - u && = e && (1.27d)\\ & x \geq 0, s \geq 0, u \geq 0. && && (1.27e) \end{aligned}$$

Das Problem (1.27) ist fast in Standardform. Der einzige Unterschied ist, dass die Variablen y freie Variablen und keine vorzeichenbeschränkte Variablen sind. Es besitzt also die Form:

$$\begin{aligned} \min \quad & \hat{c}^T \hat{x} && (1.28a)\\ & \hat{A}x = \hat{b} && (1.28b)\\ & \hat{x}_i \geq 0, \text{ für } i \in I, && (1.28c) \end{aligned}$$

wobei $I \subseteq \{1, \ldots, n\}$ eine Teilmenge der Variablen ist. Wir beschäftigen

Freie Variablen (1. Möglichkeit)

Eine einfache Möglichkeit, eine freie Variable $\hat{x}_j$ zu eliminieren, ist sie durch zwei vorzeichenbeschränkte Variablen $\hat{x}_j^+ \geq 0$ und $\hat{x}_j^- \geq 0$ zu ersetzen. Wir setzen

$$\hat{x}_j = \hat{x}_j^+ - \hat{x}_j^- \tag{1.29}$$

und fordern, wie bereits erwähnt $\hat{x}_j^+ \geq 0$ und $\hat{x}_j^- \geq 0$. Wenn wir überall im Linearen Programm (1.28) $\hat{x}_j$ durch $\hat{x}_j^+ - \hat{x}_j^-$ ersetzen, bleibt die lineare Struktur in allen Nebenbedingungen sowie der Zielfunktion erhalten.

Freie Variablen (2. Möglichkeit)

Bei der oben genannten einfachen Methode zur Elimination von freien Variablen erhöht sich die Anzahl der Variablen, ein Umstand den man gerne vermeiden möchte, um das Problem „so klein wie möglich“ zu halten. Außerdem führt die Transformation (1.29) zu einer gewissen Art von

Redundanz: Wenn man sowohl zu $\hat{x}_j^+$ als auch zu $\hat{x}_j^-$ eine Konstante addiert, ändert sich die Differenz $\hat{x}_j^+ - \hat{x}_j^-$ nicht. Mit anderen Worten, die Darstellung von $\hat{x}_j$ als Differenz von zwei nichtnegativen Werten ist nicht eindeutig. Dies stört zwar in der Simplex-Methode nicht, ist aber aus theoretischen Überlegungen heraus etwas „unschön".

Eine zweite elegantere Möglichkeit zu Elimination einer freien Variable $\hat{x}_j$ ist, es die Nebenbedingungen $\hat{A}\hat{x} = \hat{b}$ zu benutzen. Wir betrachten dazu die $j$te Spalte $\hat{A}_{\cdot j}$ der Matrix $\hat{A}$.

Falls $\hat{A}_{\cdot j} = 0$ ist, also in keiner Nebenbedingung die Variable $\hat{x}_j$ überhaupt vorkommt, so unterscheiden wir zwei Fälle: $\hat{c}_j = 0$ und $\hat{c}_j \neq 0$. Ist $\hat{c}_j = 0$, so sind nicht nur die Nebenbedingungen von $\hat{x}_j$ unabhängig, sondern auch die Zielfunktion. Wir können also die Variable $\hat{x}_j$ aus dem Linearen Programm (1.28) entfernen und erhalten ein äquivalentes kleineres Lineares Programm.

Ist hingegen $\hat{c}_j \neq 0$ und besitzt das Lineare Programm (1.28) überhaupt eine zulässige Lösung $\bar{x}$, so ist wegen $\hat{A}_{\cdot j} = 0$ auch für jeden Wert $t \in \mathbb{R}$ der Vektor $x(t)$, definiert durch

$$x_k(t) := \begin{cases} \bar{x}_k, & \text{falls } k \neq j \\ t, & \text{falls } k = j \end{cases}$$

eine zulässige Lösung. Es gilt dann $\hat{c}^T x(t) = \hat{c}^T \bar{x} + t\hat{c}_j$. Da $\hat{c}_j \neq 0$ können wir die Zielfunktion durch geeignete Wahl von $t \in \mathbb{R}$ beliebig klein machen. Wir können also schließen, dass das Problem (1.28) keine Optimallösung, da es entweder unzulässig oder unbeschränkt ist.

Es verbleibt noch der Fall, dass $\hat{A}_{\cdot j} \neq 0$. Dann gibt es mindestens eine Gleichung von $\hat{A}\hat{x} = \hat{b}$, in der $\hat{x}_j$ mit einem Koeffizienten ungleich 0 vorkommt, beispielsweise

$$\begin{aligned} &\hat{A}_{k\cdot}\hat{x} = \hat{b}_k \\ \Leftrightarrow &a_{k,1}\hat{x}_1 + a_{k,2}\hat{x}_2 + \cdots + a_{k,j}\hat{x}_j + \cdots + a_{k,n}\hat{x}_n = \hat{b}_k, \end{aligned} \tag{1.30}$$

wobei $a_{k,j} \neq 0$. Wir können dann (1.30) nach $\hat{x}_j$ auflösen:

$$\hat{x}_j = \frac{1}{a_{k,j}} \left(\hat{b}_k - \sum_{\ell \neq j} a_{k,\ell}\hat{x}_\ell \right) \tag{1.31}$$

und in jeder Gleichung von $\hat{A}\hat{x} = \hat{b}$ dann $\hat{x}_j$ durch die rechte Seite von (1.31) ersetzen. Wir erhalten dann ein reduziertes äquivalentes Lineares Programm ohne die freie

Variable $\hat{x}_j$, das sogar eine Variable weniger besitzt als das ursprüngliche Problem (1.28). Jede Optimallösung des neuen Problems kann mittels (1.31) zu einer Optimallösung des Originalproblems (1.28) erweitert werden.

2

Geometrischer Hintergrund

Bei unserem Linearen Programm in Kapitel 1, das wir als erstes Beispiel für das Simplex-Verfahren betrachtet haben, gab es eine Optimallösung, die auf einer „Ecke“ des zulässigen Bereichs lag. In diesem Kapitel werden wir den Begriff der „Ecke“ formalisieren und beweisen, dass diese Beobachtung kein Zufall ist.

2.1 Polyeder und Polytope

Betrachten wir das Lineare Programm

$$\begin{aligned} \min \quad & 3x_1 + 5x_2 && \text{(2.1a)} \\ -2x_1 - x_2 &\leq -3 && \text{(2.1b)} \\ -2x_1 - 2x_2 &\leq -5 && \text{(2.1c)} \\ -x_1 - 4x_2 &\leq -4 && \text{(2.1d)} \\ x_1, x_2 &\geq 0 && \text{(2.1e)} \end{aligned}$$

Definition 2.1 (Hyperebene, Halbraum). *Für einen Vektor $a \in \mathbb{K}^n$ mit $a \neq 0$ und $\alpha \in \mathbb{K}$ heisst die Menge*

$$G := G_{a,\alpha} := \left\{ x \in \mathbb{K}^n : a^T x = \alpha \right\}$$

die durch a und α bestimmte Hyperebene. *Analog nennen wir die Menge* Hyperebene

$$H := H_{a,\alpha} := \left\{ x \in \mathbb{K}^n : a^T x \leq \alpha \right\}$$

den durch $a \neq 0$ und α bestimmten Halbraum. Halbraum

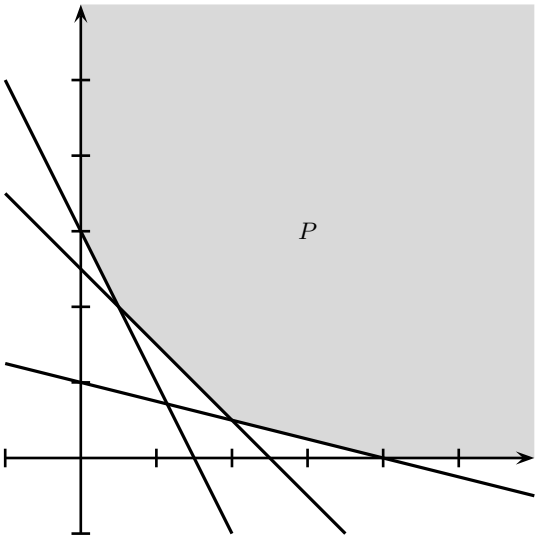


Abb. 2.1. Polyeder zum Linearen Programm (**??**)

In unserem Beispiel bildet der Schnitt der fünf Halbräume

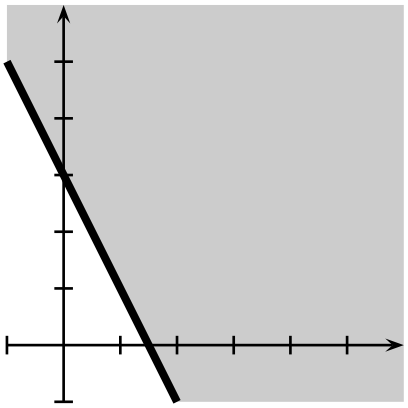
$$
\begin{aligned}
H_{\binom{-2}{-1},-2} &= \left\{ x \in \mathbb{R}^2 : -2x_1 - x_2 \leq -3 \right\} \\
H_{\binom{-2}{-2},-5} &= \left\{ x \in \mathbb{R}^2 : -2x_1 - 2x_2 \leq -5 \right\} \\
H_{\binom{-1}{-4},-4} &= \left\{ x \in \mathbb{R}^2 : -x_1 - 4x_2 \leq -4 \right\} \\
H_{\binom{-1}{0},0} &= \left\{ x \in \mathbb{R}^2 : -x_1 \leq 0 \right\} \\
H_{\binom{0}{-1},0} &= \left\{ x \in \mathbb{R}^2 : -x_2 \leq 0 \right\}
\end{aligned}
$$

die Menge der zulässigen Lösungen für das Lineare Programm (**??**). Den Schnitt von endlich vielen Halbräumen nennen wir ein *Polyeder*, allerdings stellt es sich als nützlich heraus, auch den ganzen Raum als Polyeder zu bezeichnen.

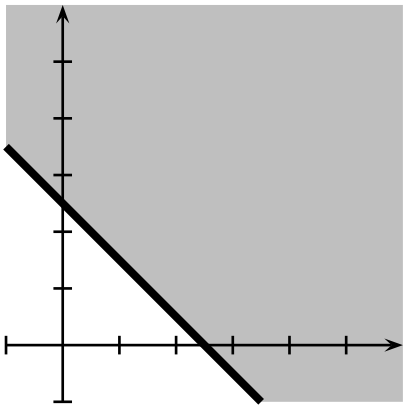
Polyeder

Definition 2.2 (Polyeder, Polytop). *Wir nennen eine Menge $P \subseteq \mathbb{K}^n$ ein* Polyeder*, falls entweder $P = \mathbb{K}^n$ gilt oder P der Durchschitt endlich vieler Halbräume ist. Ein Polyeder P heißt* Polytop*, wenn es beschränkt ist, d.h. falls es eine Zahl $K > 0$ gibt, so dass $P \subseteq \{ x \in \mathbb{K}^n : \|x\|_2 \leq K \}$ gilt.*

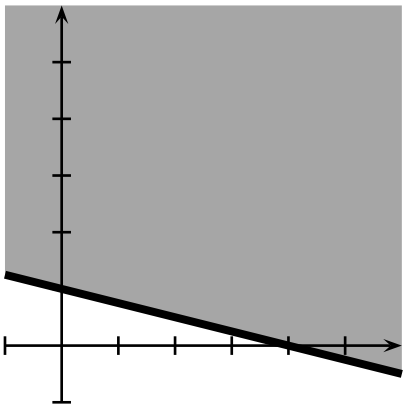
Ist $P \neq \mathbb{K}^n$ ein Polyeder, also



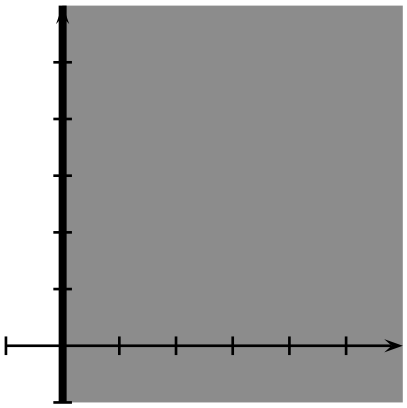
(a) Halbraum $H_{\binom{-2}{-1},-3}$



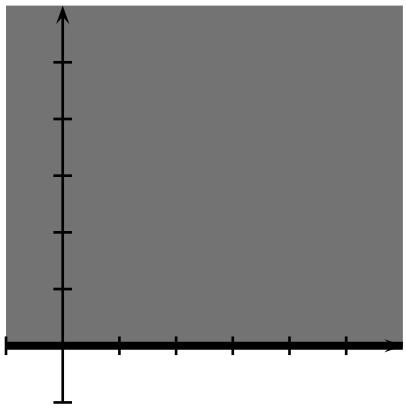
(b) Halbraum $H_{\binom{-2}{-2},-5}$



(c) Halbraum $H_{\binom{-1}{-4},-4}$



(d) Halbraum $H_{\binom{-1}{0},0}$



(e) Halbraum $H_{\binom{0}{-1},0}$

Abb. 2.2.

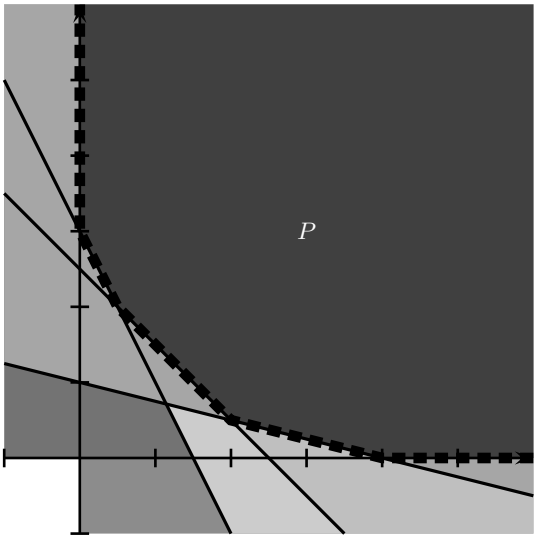


Abb. 2.3. Das Polyeder als Schnitt der fünf Halbräume

$$P = \bigcap_{i=1}^{m} \{\, x \in \mathbb{K}^n : a_i^T x \leq \alpha_i \,\}, \tag{2.2}$$

so können wir die Vektoren $a_1^T, \ldots, a_m^T$ als Zeilen einer $m \times n$-Matrix

$$A := \begin{pmatrix} a_1^T \\ \vdots \\ a_m^T \end{pmatrix}$$

auffassen und einen Vektor $b := (\alpha_1, \ldots, \alpha_m)^T \in \mathbb{K}^m$ definieren. Damit können wir das Polyeder P aus (**??**) äquivalent auch als

$$P = P(A, b) = \{\, x \in \mathbb{K}^n : Ax \leq b \,\} \tag{2.3}$$

schreiben. Ist $P = \mathbb{K}^n$, so können wir P ebenfalls in der Form (**??**) schreiben, indem wir die $1 \times n$-Nullmatrix und den Nullvektor $0 \in \mathbb{K}^1$ benutzen:

$$\mathbb{K}^n = \left\{\, x \in \mathbb{K}^n : 0^T x \leq 0 \,\right\}.$$

Daher ist jedes Polyeder in der Form (**??**) darstellbar. Umgekehrt ist natürlich auch jede Menge der Form (**??**) ein Polyeder, da

$$P(A, b) = \bigcap_{\{i : A_{i\cdot} \neq 0\}} \{\, x \in \mathbb{K}^n : A_{i\cdot} x \leq b_i \,\},$$

wobei $A_{i\cdot}$ die $i$te Zeile der Matrix A bezeichnet.

Damit erhalten wir folgendes grundlegende Ergebnis:

Beobachtung 2.3. *Eine Menge $P \subseteq \mathbb{K}^n$ ist genau dann ein Polyeder, wenn es eine $m \times n$-Matrix A und einen Vektor $b \in \mathbb{K}^n$ gibt, so dass*

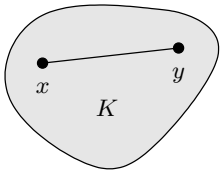
$$P = P(A, b) = \{ x \in \mathbb{K}^n : Ax \leq b \}. \tag{2.4}$$

Wir schreiben im Folgenden kurz $P(A, b)$, um das in (**??**) über eine Matrix A und einen Vektor b definierte Polyeder zu bezeichnen. Man beachte, dass es für ein Polyeder P immer mehr als eine Matrix A und einen Vektor b gibt, so dass $P = P(A, b)$ gilt. So sind für jedes skalare $\lambda > 0$ die Polyeder $P(A, b)$ und $P(\lambda A, \lambda b)$ identisch Darüberhinaus sind auch die Dimension von A und b nicht eindeutig.

2.2 Konvexität, Extremalmengen und Ecken von Polyedern

Definition 2.4 (Konvexe Menge). *Eine Menge $K \subseteq \mathbb{K}^n$ heißt* konvex*, wenn für alle $x, y \in K$ und $\lambda \in \mathbb{K}$ mit $0 \leq \lambda \leq 1$ gilt: $\lambda x + (1 - \lambda)y \in K$.*

konvex



Für eine konvexe Menge K und $x, y \in K$ liegt immer auch die Verbindungsstrecke zwischen x und y ganz in K.

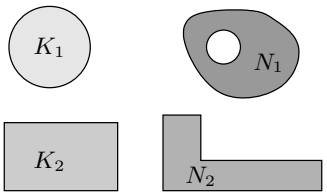
Anschaulich bedeutet die Definition, dass für eine konvexe Menge K und $x, y \in K$ auch die Verbindungsstrecke zwischen x und y ganz in K liegen muss.

Jeder Halbraum $H := H_{a,\alpha} := \{ x \in \mathbb{K}^n : a^T x \leq \alpha \}$ ist konvex, da für $x, y \in H$ und $0 \leq \lambda \leq 1$ gilt:

$$c^T(\lambda x + (1-\lambda)y) = \lambda c^T x + (1-\lambda)c^T y \leq \lambda\alpha + (1-\lambda)\alpha = \alpha.$$

Lemma 2.5. *Der Schnitt beliebig vieler konvexer Mengen ist eine konvexe Menge.*

Beweis. Seien K_i, $i \in I$ konvex und $K = \bigcup_{i \in I} K_i$. Für $x, y \in K$ gilt $x, y \in K_i$ für alle $i \in I$. Somit ist für $0 \leq \lambda \leq 1$ dann wegen der Konvexität jeder Menge K_i auch $z := \lambda x + (1 - \lambda)y \in K_i$, also $z \in K$. □



K_1 und K_2 sind konvex, N_1 und N_2 hingegen nicht.

Korollar 2.6. *Jedes Polyeder ist eine konvexe Menge.*

Beweis. Folgt aus Lemma **??**, da jedes Polyeder entweder gleich $\mathbb{K}^n$ oder der Schnitt endlich vieler Halbräume sind. □

Beispiel 2.7. Sei A eine $m \times n$-matrix und die Menge K definiert durch:

$$\begin{aligned} K &:= \{\, z : \text{ es gibt ein } x \in \mathbb{R}^n \text{ mit } x \geq 0 \text{ und } Ax = z \,\} \\ &= \{\, Ax : x \in \mathbb{R}^n, x \geq 0 \,\} \subseteq \mathbb{R}^m. \end{aligned}$$

Dann ist K konvex. Sind nämlich $z_1 = Ax_1 \in K$ und $z_2 = Ax_2 \in K$ mit $x_1, x_2 \geq 0$ und $0 \leq \lambda \leq 1$, so haben wir $\lambda x_1 + (1-\lambda)x_2 \geq 0$ und $A(\lambda x_1 + (1-\lambda)x_2) = \lambda Ax_1 + (1-\lambda)Ax_2 = \lambda z_1 + (1 - \lambda z_2)$, also $\lambda z_1 + (1-\lambda)z_2 \in K$. ◁

Definition 2.8 (Extremalmenge, Extremalpunkt). *Eine konvexe Teilmenge E einer konvexen Menge $K \subseteq \mathbb{K}^n$ heißt* Extremalmenge *von K, falls aus $z \in E$ mit $z = \lambda x + (1-\lambda)y$ mit $0 < \lambda < 1$, $\lambda \in \mathbb{K}$ für $x, y \in K$ folgt, dass $x, y \in E$. Ein* Extremalpunkt *ist eine einelementige Extremalmenge.*

Extremalmenge

Extremalpunkt

Ein Extremalpunkt z eines Polyeders ist also ein Punkt $z \in P$, so dass aus $z = \lambda x + (1-\lambda)y$ mit $x, y \in P$ und $0 < \lambda < 1$, folgt, dass $x = y = z$. Ein Extremalpunkt eines Polyeders P lässt sich also nicht als echte Konvexkombination verschiedener Punkte $x, y \in P$ schreiben. Anschaulich ist klar, dass die Extremalpunkte tatsächlich die „geometrischen“ Ecken sind. Wir werden später sehen, dass diese Anschauung richtig ist und die oben definierten Ecken genau die geometrisch erwarteten Eigenschaften besitzen.

Theorem 2.9. *Sei $P \subseteq \mathbb{K}^n$ ein Polyeder. Falls das Lineare Programm*

$$\min \{\, c^T x : x \in P \,\}$$

Optimallösungen hat, so ist die Menge der Optimallösungen eine Extremalmenge von P.

Beweis. Falls das Lineare Programm eine Optimallösung hat, so existieren $\gamma = \max \{\, c^T x : x \in P \,\}$ und $x^* \in P$ mit $c^T x^* = \gamma$. Die Menge der Optimallösungen ist dann

$$E = \{\, x \in P : c^T x = \gamma \,\} = P \cap \{\, x : -c^T x \leq -\gamma \,\} \cap \{\, x : c^T x \leq \gamma \,\}.$$

Wir haben hier E als Schnitt dreier Halbräume dargestellt. Somit ist E wieder ein Polyeder und damit auch konvex.

Sei $z \in E$. Wir nehmen an, dass es $x, y \in P$ gibt, die nicht beide in E liegen, und $0 < \lambda < 1$ mit $z = \lambda x + (1-\lambda)y$. O.B.d.A. sei $x \notin E$, also $c^T x > \gamma$. Dann folgt aber mit $c^T y \geq \gamma$

$$c^T z = \lambda \underbrace{c^T x}_{>\gamma} + (1-\lambda) \underbrace{c^T y}_{\geq\gamma} > \lambda\gamma + (1-\lambda)\gamma = \gamma,$$

im Widerspruch zu $z \in E$. Also muss $x, y \in E$ gelten. □

2.3 Basislösungen und Optimierung

Wir betrachten das Lineare Programm in Standardform:

$$\begin{aligned} \min \quad & c^T x && \text{(2.5a)} \\ & Ax = b && \text{(2.5b)} \\ & x \geq 0 && \text{(2.5c)} \end{aligned}$$

Wir nehmen dabei zunächst an, dass die $m \times n$-Matrix $A \in \mathbb{K}^{m\times n}$ vollen Zeilenrang besitzt, also $\operatorname{Rang} A = m$ gilt. Es besteht ein wichtiger Zusammenhang zwischen den im letzten Abschnitt definierten Extremalpunkten eines Polyeders und Basislösungen.

Lemma 2.10. *Sei $P := \{x : Ax = b, x \geq 0\}$ das Polyeder der zulässigen Lösungen des Linearen Programms (**??**). Dann ist x genau dann ein Extremalpunkt von P, wenn x eine Basislösung von (**??**) ist.*

Beweis. „⇒“: Sei x ein Extremalpunkt von P und $I(x) := \{\, i : x_i > 0 \,\}$ die Menge der Indizes i, so dass x_i strik positiv ist. Falls die Vektoren $\{\, A_{\cdot i} : i \in I(x) \,\}$ linear unabhängig sind, dann können wir wegen $\operatorname{Rang} A = m$ noch $m - |I(x)|$ weitere Spaltenindizes I' finden, so dass $B := I(x) \cup I'$ eine Basis von A ist. Da $A_{\cdot B} x = b$ und $A_{\cdot B}$ nichtsingulär ist, ist x die eindeutige zu B gehörende Basislösung.

Falls die Vektoren $\{\, A_{\cdot i} : i \in I(x) \,\}$ nicht linear unabhängig sind, dann finden wir Skalare λ_i, $i \in I(x)$, die nicht alle 0 sind, so dass $\sum_{i\in I(x)} \lambda_i A_{\cdot i} = 0$. Wir definieren den Vektor $y \in \mathbb{R}^n$ durch

$$y_i := \begin{cases} \lambda_i, & \text{falls } i \in I(x) \\ 0, & \text{sonst.} \end{cases}$$

Dann gilt nach Konstruktion

$$Ay = \sum_{i=1}^{n} A_{\cdot i} y_i = \sum_{i\in I(x)} A_{\cdot i} \lambda_i = 0.$$

Wir betrachten für $\delta \in \mathbb{K}$ den Vektor $x + \delta y$, für den gilt:

$$x_i + \delta y_i := \begin{cases} x_i + \delta\lambda_i, & \text{falls } i \in I(x) \\ 0, & \text{sonst.} \end{cases} \tag{2.6}$$

Dann gilt $A(x + \delta y) = Ax + \delta Ay = Ax + \delta 0 = b$ für alle $\delta \in \mathbb{K}$. Da $x_i > 0$ für $i \in I(x)$ finden wir $\bar{\delta} \neq 0$, so dass

$$x + \bar{\delta} y \geq 0 \quad \text{und} \quad x - \bar{\delta} y \geq 0.$$

Also sind $x + \bar{\delta} y \in P$ und $x - \bar{\delta} y \in P$ wegen $y \neq 0$ zwei von x verschiedene Vektoren mit $x = \frac{1}{2}(x + \bar{\delta} y) + \frac{1}{2}(x - \bar{\delta} y)$ im Widerspruch zur Voraussetzung, dass x ein Extremalpunkt von P ist.

„$\Leftarrow$“: Sei $x = \binom{x_B}{x_N}$ Basislösung zur Basis B. Wir nehmen an, dass $x = \lambda y + (1-\lambda) z$ für $0 < \lambda < 1$ und $y, z \in P$ gilt. Für $j \in N$ ist

$$0 = x_j = \lambda \underbrace{y_j}_{\geq 0} + (1 - \lambda) \underbrace{z_j}_{\geq 0}.$$

Daher folgt $y_N = z_N = 0$. Dann folgt aber wegen

$$b = Ay = A_{\cdot B} y_B + A_{\cdot N} \underbrace{y_N}_{=0} = A_{\cdot B} y_b$$

und der Nichtsingularität von $A_{\cdot B}$, dass $y_B = A_{\cdot B}^{-1} b = x_B$. Somit ist $y = x$. Analog folgt $z = x$. Also ist x Extremalpunkt von P. □

Theorem 2.11 (Fundamentalsatz der Linearen Optimierung I). *Für das Lineare Programm in Standardform (**??**) mit* $\operatorname{Rang} A = m$ *gelten folgende Aussagen:*

*(i) Falls (**??**) eine zulässige Lösung besitzt, dann hat das Lineare Programm auch eine zulässige Basislösung.*
*(ii) Falls (**??**) eine optimale Lösung besitzt, dann hat das Lineare Programm auch eine optimale Basislösung (d.h. eine optimale Lösung die auch Basislösung ist).*

Beweis. (i) Der Beweis verläuft ähnlich wie der Beweis von Lemma **??**. Sei x eine zulässige Lösung von (**??**) und $I(x) := \{ i : x_i > 0 \}$. Falls die Vektoren $\{ A_{\cdot i} : i \in I(x) \}$ linear unabhängig sind, so können wir wie im Beweis von Lemma **??** schließen, dass x eine Basislösung ist. Falls die Vektoren $\{ A_{\cdot i} : i \in I(x) \}$ nicht linear unabhängig sind, dann finden wir Skalare λ_i, $i \in I(x)$,

die nicht alle 0 sind, so dass $\sum_{i \in I(x)} \lambda_i A_{\cdot i} = 0$. Wir können o.B.d.A. annehmen, dass mindestens ein Wert λ_i strikt größer als 0 ist (wir können notfalls alle Werte λ_i mit -1 multiplizieren). Wir betrachten für $\delta \in \mathbb{K}$ den Vektor $x + \delta y$ aus (**??**). Wieder gilt $A(x + \delta y) = b$ für alle $\delta \in \mathbb{K}$. Sei

$$\bar{\delta} := \min \left\{ \frac{x_i}{y_i} : i \in I(x) \text{ und } y_i > 0 \right\}. \tag{2.7}$$

Dann gilt $x - \bar{\delta} y \geq 0$ und $x - \bar{\delta} y$ hat höchstens $|I(x)| - 1$ positive Komponenten. Wir setzen $x := x - \bar{\delta} y$. Wenn wir diesen Prozess fortsetzen, so gelangen wir nach höchstens n Schritten in die Situation, dass die Vektoren $\{ A_{\cdot i} : i \in I(x) \}$ linear unabhängig sind und wir erhalten wie oben eine zulässige Basislösung.

(ii) Sei nun x eine Optimallösung von (**??**). Falls die Vektoren $\{ A_{\cdot i} : i \in I(x) \}$ linear unabhängig sind, so ist x wie oben bereits eine Basislösung und es ist nichts mehr zu zeigen.
Im zweiten Fall ist $x + \delta y$ für alle kleinen $|\delta|$ zulässig für (**??**). Wir haben:

$$c^T(x + \delta y) = c^T x + \delta \underbrace{\sum_{i \in I(x)} c^T a_i}_{=:t} = c^T x + \delta t. \tag{2.8}$$

Es folgt $t = 0$, da ansonsten entweder $c^T(x + \delta y) > c^T x$ oder $c^T(x - \delta y) > c^T x$ für kleines $\delta > 0$ wäre. Damit ist dann aber der Vektor $x + \bar{\delta} y$ mit $\bar{\delta}$ aus (**??**) wegen $c^T(x + \bar{\delta} y) = c^T x$ ebenfalls eine Optimallösung von (**??**), die weniger positive Komponenten als x hat. Wir können also wie in (i) verfahren und gelangen nach höchstens n Schritten zu einer optimalen Lösung, die auch Basislösung ist. □

Aus dem obigen Fundamentalsatz und der Charakterisierung von Extremalpunkten eines Polyeders in Lemma **??** ergibt sich nun unmittelbar:

Korollar 2.12. *Falls das Lineare Programm in Standardform (**??**) mit* $\operatorname{Rang} A = m$ *eine Optimallösung hat, so hat es auch eine Optimallösung, die Ecke des Polyeders* $P = \{x : Ax = b, x \geq 0\}$ *ist.*

Betrachten wir die Technik, die wir beim Beweis von Satz **??** Teil (ii) angewendet haben, noch einmal genauer.

Sei diesmal x eine beliebige zulässige Lösung von (P), die noch keine Basislösung ist. Wir finden dann $y \neq 0$ mit $Ay = 0$, so dass $x + \delta y$ für alle kleinen $|\delta|$ zulässig ist. Wie wir in (**??**) ausgerechnet hatten, gilt $c^T(x+\delta y) = c^T x + \delta t$ mit $t := \sum_{i \in I(x)} c^T a_i$. Ist $t < 0$, so haben wir $c^T(x+\delta y) < c^T x$ für alle $\delta > 0$. Falls (P) nicht unbeschränkt ist, so existiert ein maximales $\bar{\delta} > 0$, so dass $x + \bar{\delta} y$ zulässig ist, und es folgt wieder, dass $x + \bar{\delta} y$ weniger strikt positive Einträge hat als x. Wir können aus x also bessere Lösung mit weniger strikt positiven Einträgen konstruieren. Ist $t > 0$, so funktioniert die Konstruktion analog mit $x + \bar{\delta} y$ für ein geignetes $\bar{\delta} < 0$. Ist $t = 0$, so sind wir in der Situation, die wir in Teil (ii) des Satzes betrachtet hatten. Insgesamt können wir also aus einer Nicht-Basislösung x eine neue Lösung $x + \bar{\delta} y$ konstruieren, so dass $c^T(x + \bar{\delta} y) \leq c^T x$ (wobei für $t \neq 0$ sogar strikte Ungleichung gilt) und $x + \bar{\delta} y$ weniger strikt positive Einträge hat als x. Diese Beobachtung liefert ein Verfahren, um eine zulässige Lösung x für (P) in eine Basislösung mit nicht schlechterem Zielfunktionswert zu konvertieren.

Wir starten mit einer zulässigen Lösung x und testen, ob x bereits eine Basislösung ist. Falls ja, dann stoppen wir. Ansonsten berechnen wir aus x einen neuen Vektor $x + \bar{\delta} y$ wie oben. Wenn wir das Verfahren fortsetzen, so haben wir nach maximal n Iterationen eine Basislösung, deren Zielfunktionswert nicht schlechter ist, als die Ausgangslösung. In jeder Iteration müssen wir ein Lineares Gleichungssystem lösen, um die lineare Unabhängigkeit der Spalten $\{ A_{\cdot i} : i \in I(x) \}$ zu testen. Dies ist in $\mathcal{O}(n^2)$ Zeit durchführbar, etwa mit dem Gauß-Algorithmus. Wir halten dieses nützliche Ergebnis fest.

Lemma 2.13. *Das Lineare Programm in Standardform (**??**) mit* $\operatorname{Rang} A = m$ *habe eine Optimallösung. Zu einer beliebigen zulässigen Lösung x von (P) können wir in $\mathcal{O}(n^3)$ Zeit eine Basislösung $\bar{x}$ von (P) konstruieren, so dass $c^T \bar{x} \leq c^T x$.* □

3

Dualität

3.1 Untere Schranken für optimale Lösung eines LPs

Versetzen wir uns noch einmal in unseren Studenten *Simple Ex*, der auf der Studentenparty seinen Alkoholkonsums minimieren möchte. Wie wir in Abschnitt 1.1 bereits gesehen haben, führt diese Aufgabe zum Linearen Programm

$$\begin{aligned}
\min \quad & 12x_1 + & 2x_2 + & 5x_3 & & & (3.1a)\\
& 2x_1 + & x_2 + & 2x_3 = & 3 & & (3.1b)\\
& & 2x_2 + & x_3 = & 2 & & (3.1c)\\
& x_1 \geq 0 & x_2 \geq 0 & x_3 \geq 0. & & & (3.1d)
\end{aligned}$$

(Wir haben hier im Gegensatz Abschnitt 1.1 konkrete Werte $c_1 = 12$, $c_2 = 2$, $c_3 = 5$ für die Zielfunktionskoeffizienten benutzt, um in der Motivation besser rechnen zu können).

Nehmen wir an, dass *Simple Ex* zunächst gar nicht die wirkliche Optimallösung von (**??**) bestimmen möchte, sondern zunächst einmal (möglichst rasch) abschätzen möchte, wie viel Alkohol er mindestens zu sich nehmen wird, wenn er dem Ergebnis des LPs folgt und sein komplettes Geld ausgibt. Mit anderen Worten, *Simple Ex* möchte zunächst einmal eine *untere Schranke* für den optimalen Zielfunktionswert z^* bestimmen.

untere Schranke

Wenn wir die Gleichung (**??**) mit 1/2 multiplizieren und zu Gleichung (**??**) addieren, so erhalten wir

$$1 \cdot (2x_1 + x_2 + 2x_3) + \frac{1}{2} \cdot (2x_2 + x_3) = 1 \cdot 3 + \frac{1}{2} \cdot 2 = 4.$$

Somit gilt für jede zulässige Lösung $x = (x_1, x_2, x_3)^T$ von (**??**) die Gleichung

$$2x_1 + 2x_2 + \frac{5}{2}x_3 = 4. \tag{3.2}$$

Betrachten wir die Koeffizienten vor den Variablen in (**??**). Da $2 \leq 12 = c_1$, $2 \leq 2 = c_2$ und $\frac{5}{2} \leq 5 = c_3$ ist und $x \geq 0$ gilt, folgt aus (**??**) dann:

$$\begin{aligned} c^T x &= c_1 x_1 + c_2 x_2 + c_3 x_3 \\ &= 12x_1 + 2x_2 + 5x_3 \\ &\geq 2x_1 + 2x_2 + \frac{5}{2}x_3 \overset{(??)}{=} 4. \end{aligned}$$

Wir haben damit gezeigt, dass $c^T x \geq 4$ für jede zulässige Lösung x von (**??**) ist. Insbesondere gilt dies natürlich auch für die Optimallösung x^* von (**??**), so dass *Simple Ex* durch die kurze Rechnung sicher sein kann, dass er auf jeden Fall mindestens 4 Alkoholeinheiten zu sich nehmen wird.

Können wir noch eine bessere untere Schranke herleiten? Wenn wir Gleichung (**??**) mit 3 multiplizieren, Gleichung (**??**) mit -2 multiplizieren und die beiden erhaltenen Gleichungen wieder addieren, so ergibt dies:

$$3 \cdot (2x_1 + x_2 + 2x_3) - 2 \cdot (2x_2 + x_3) = 3 \cdot 3 - 2 \cdot 2 = 5.$$

Also folgt

$$6x_1 - x_2 + 4x_3 = 5 \tag{3.3}$$

für jede zulässige Lösung x von (**??**). Wir haben wieder die Situation, dass die Koeffizienten vor den Variablen in (**??**) kleiner oder gleich den entsprechenden Koeffizienten in der Zielfunktion von (**??**) sind: $6 \leq 12 = c_1$, $-1 \leq 2 = c_2$ und $4 \leq 5 = c_3$. Wie oben können wir daher schliessen, dass für jede zulässige Lösung x von (**??**) gilt:

$$c^T x = 12x_1 + 2x_2 + 5x_3 \leq 6x_1 - x_2 + 4x_3 \overset{(??)}{=} 5. \tag{3.4}$$

Simple Ex kann also durch die zweite Rechnung sogar beweisen, dass er mindestens 5 Alkoholeinheiten trinken muss.

Wir systematisieren nun unseren *ad-hoc* Ansatz für die unteren Schranken. Im Prinzip haben wir Linearkombinationen der Nebenbedingungen des Linearen Programms betrachtet: Wir haben die erste Gleichung (**??**) mit einem

Wert $y_1 \in \mathbb{R}$ multipliziert, die zweite Gleichung (**??**) mit einem Wert $y_2 \in \mathbb{R}$ multipliziert und dann addiert. Dies ergibt dann im allgemeinen Fall die Gleichung

$$y_1 \cdot (2x_1 + x_2 + 2x_3) + y_2 \cdot (2x_2 + x_3) = y_1 \cdot 3 + y_2 \cdot 2,$$

bzw. nach Umsortieren der Terme:

$$(2y_1 + 0y_2)x_1 + (y_1 + 2y_2)x_2 + (2y_1 + y_2)x_3 = 3y_1 + 2y_2. \tag{3.5}$$

Damit wir den Zielfunktionswert jeder zulässigen Lösung x von (**??**) durch die linke Seite der obigen Gleichung nach unten abschätzen können, müssen die Koeffizienten auf der rechten Seite von (**??**) kleiner oder gleich den Koeffizienten in der Zielfunktion sein, d.h. es muss gelten:

$$2y_1 + 0y_2 \leq c_1 \tag{3.6a}$$
$$y_1 + 2y_2 \leq c_2 \tag{3.6b}$$
$$2y_1 + y_2 \leq c_3 \tag{3.6c}$$

Haben wir also Werte $y_1, y_2 \in \mathbb{R}$ gefunden, welche die Ungleichungen (**??**) erfüllten, so ist nach (**??**) dann $3y_1 + 2y_2$ eine untere Schranke für den optimalen Zielfunktionswert z^* von (**??**). Bei unserem ersten Versuch hatten wir $y_1 = 1$, $y_2 = \frac{1}{2}$, was die untere Schranke $3 \cdot 1 + 2 \cdot \frac{1}{2} = 4$ ergibt. Im zweiten Versuch hatten wir mit $y_1 = 3$ und $y_2 = -2$ dann die bessere Schranke $3 \cdot 3 + 2 \cdot (-2) = 5$ erhalten.

Wir können nun das Problem, eine größtmögliche untere Schranke mit unserem „Linearkombinationsansatz“ zu finden, als Optimierungsproblem formulieren. Wir wollen $y_1, y_2 \in \mathbb{R}$ finden, so dass $3y_1 + 2y_2$ möglichst groß ist und die Ungleichungen (**??**) erfüllt sind:

$$\begin{aligned}
\max \quad & 3y_1 + && 2y_2 && && (3.7a)\\
& 2y_1 && && \leq && 12 && (3.7b)\\
& y_1 + && 2y_2 && \leq && 1 && (3.7c)\\
& 2y_1 + && y_2 && \leq && 5 && (3.7d)
\end{aligned}$$

Das Problem (**??**) ist wieder ein Lineares Programm!

3.2 Das Duale Programm

Wir betrachten nun ein allgemeines Lineares Programm in Standardform:

$$\begin{aligned} \min \quad & c^T x && \text{(3.8a)} \\ & Ax = b && \text{(3.8b)} \\ & x \geq 0, && \text{(3.8c)} \end{aligned}$$

wobei $A \in \mathbb{R}^{m \times n}$ und $b \in \mathbb{R}^m$. In unseren Ansatz für die unteren Schranken im letzten Abschnitt haben wir Linearkombinationen der Gleichungen in (**??**) gebildet. Wir bezeichnen für $i = 1, \ldots, m$ mit y_i den Multiplikator für die $i$te Gleichung $A_{i\cdot}x = b_i$ und setzen $y := (y_1, \ldots, y_m)^T$. Dann ergibt die Linearkombination mit den Koeffizienten $y_1, \ldots, y_m$:

$$\begin{aligned} & \sum_{i=1}^{m} y_i A_{i\cdot} x = \sum_{i=1}^{m} y_i b_i \\ \Leftrightarrow & \left(\sum_{i=1}^{m} y_i A_{i\cdot} \right) x = b^T y \\ \Leftrightarrow & (y^T A)x = b^T y \\ \Leftrightarrow & (A^T y)^T x = b^T y \end{aligned} \qquad (3.9)$$

Die Gleichung $(A^T y)^T x = b^T y$ gilt also für alle x, die zulässig für (**??**) sind. Damit wir $b^T y$ als untere Schranke für $c^T x$ verwenden können, muss der Vektor $A^T y$ komponentenweise kleiner oder gleich dem Vektor c sein, es muss also

$$A^T y \leq c \qquad (3.10)$$

gelten. Haben wir einen Vektor y, der die Ungleichungen (**??**) erfüllt, so folgt dann für jede zulässige Lösung x von (**??**) wegen $x \geq 0$:

$$c^T x \geq (A^T y)x \overset{(??)}{=} b^T y. \qquad (3.11)$$

Die bestmögliche untere Schranke für unseren Ansatz erhalten wir dann, indem wir y so wählen, dass (**??**) gilt und $b^T y$ möglichst groß wird, d.h. indem wir das Lineare Programm

$$\begin{aligned} \max \quad & b^T y && \text{(3.12a)} \\ & A^T y \leq c && \text{(3.12b)} \end{aligned}$$

duales Programm

lösen. Das Lineare Programm (**??**) wird *duales Programm* zu (**??**) genannt. Unsere Herleitung der unteren Schranken für (**??**) zeigt das folgende wichtige Ergebnis:

Lemma 3.1 (Schwacher Dualitätssatz). *Ist x eine zulässige Lösung des* primalen Programms *(??) und y eine zulässige Lösung des* dualen Programms *(??) so gilt:*

$$b^T y \leq c^T x.$$

Beweis. Siehe (??). □

Aus dem schwachen Dualitätssatz ergibt sich sofort folgende Konsequenz:

Korollar 3.2. *Ist x zulässig für das primale Programm (??) und y zulässig für das duale Programm (??) und $c^T x = b^T y$, so sind x und y optimal für die entsprechenden Probleme.* □

3.3 Das Lemma von Farkas und seine Konsequenzen

Aus der Linearen Algebra kennen wir folgenden sogenannten Alternativsatz: Alternativsatz

Theorem 3.3. *Sei $A \in \mathbb{R}^{m \times n}$ eine Matrix und $b \in \mathbb{R}^m$. Dann hat genau eines der beiden Systeme*

$$Ax = b \tag{3.13}$$

und

$$A^T y = 0 \tag{3.14a}$$
$$b^T y \neq 0 \tag{3.14b}$$

eine Lösung.

Beweis. Man sieht leicht, dass nicht beide Systeme eine Lösung haben können. Ist nämlich $Ax = b$ und $A^T y = 0$, so folgt $b^T y = (Ax)^T y = x^T A^T y = x^T 0 = 0$.

Aus der Linearen Algebra ist bekannt, dass der Nullraum $N(A^T) := \{y \in \mathbb{R}^m : A^T y = 0\}$ von A^T und der Bildraum $R(A) := \{Ax : x \in \mathbb{R}^n\}$ orthogonale Komplemente sind.[1] Daher können wir b eindeutig zerlegen in $b = b_1 + b_2$ mit $b_1 \in N(A^T)$ und $b_2 \in R(A)$. Wir nehmen

[1] Ein Beweis dieser Tatsache findet sich auch in Lemma **??**, bei dem dann auch die expliziten Darstellungen der Projektionen auf $N(A^T)$ und $R(A)$ berechnet werden.

an, dass das System (**??**) unlösbar ist. Wäre $b \notin R(A)$, so wäre $b_1 \neq 0$ und $A^T b_1 = 0$ (wegen $b_1 \in N(A^T)$) und $b^T b_1 = b_1^T b_1 + b_1^T b_2 = \|b_1\|_2^2 > 0$ im Widerspruch zu Unlösbarkeit von (**??**). Also ist $b \in R(A)$, mit anderen Worten, es existiert ein $x \in \mathbb{R}^n$ mit $Ax = b$. □

Ziel dieses Abschnitts wird es sein, entsprechende Resultate für Systeme von linearen Ungleichungen herzuleiten.

Theorem 3.4 (Farkas' Lemma). *Sei $A \in \mathbb{R}^{m\times n}$ eine Matrix und $b \in \mathbb{R}^m$. Dann hat genau eines der beiden Systeme*

$$Ax = b \tag{3.15a}$$
$$x \geq 0 \tag{3.15b}$$

und

$$A^T y \geq 0 \tag{3.16a}$$
$$b^T y < 0 \tag{3.16b}$$

eine Lösung.

Beweis. Offensichtlich können nicht beide Systeme (**??**) und (**??**) eine Lösung haben, denn andernfalls wäre

$$0 > b^T y = (Ax)^T y = \underbrace{x^T}_{\geq 0} \underbrace{A^T y}_{\geq 0} \geq 0.$$

Wir müssen noch zeigen, dass immer mindestens eines der beiden Systeme eine Lösung besitzt. Sei (**??**) unlösbar, d.h. es gebe kein x mit $Ax = b$ und $x \geq 0$.

Mir bemerken zunächst, dass wir ohne Beschränkung der Allgemeinheit annehmen könnben, dass in der Matrix A keine Spalte mehr als einmal vorkommt. Wir nennen nun eine Spalte $A_{\cdot j}$ *komplementär*, wenn auch $-A_{\cdot j}$ eine Spalte von A ist und beweisen die Behauptung durch Induktion nach der Anzahl k der *nicht*-komplementären Spalten.

Ist $k = 0$, so können wir die jeweils komplementären Spalten zusammenfassen und erhalten ein System $\tilde{A}\tilde{x} = b$, wobei $\tilde{x}$ keine Vorzeichenrestriktionen hat. Dieses System ist unlösbar, daher gibt es nach Satz **??** ein y mit $A^T y = 0$ und $b^T y = \alpha \neq 0$. Wenn $\alpha < 0$ ist, so haben wir eine Lösung von (**??**), für $\alpha > 0$ löst $-y$ das System (**??**).

Wir nehmen nun an, dass die Aussage für alle System mit höchstens k nicht-komplementären Spalten bereits bewiesen und A eine Matrix mit $k+1$ nicht-komplementären Spalten ist. Ohne Beschränkung sei die Spalte $A_{\cdot n}$ nicht-komplementär. Wir setzen $J := \{1, \ldots, n-1\}$. Nach Konstruktion hat dann $A_{\cdot J}$ genau k nicht-komplementäre Spalten.

Das System $A_{\cdot J}x = b$, $x \geq 0$ ist unlösbar, da sich jede Lösung über $x_n := 0$ zu einer Lösung von (**??**) fortsetzen lässt. Daher existiert nach Induktionsvoraussetzung ein y mit $A_{\cdot J}^T y \geq 0$ (also $A_{\cdot j}^T y \geq 0$ für alle $j \in J$) und $b^T y < 0$.

Wenn auch $A_{\cdot n}^T y \geq 0$ gilt, dann erfüllt y die Bedingungen $A^T y \geq 0$ und $b^T y$, ist also die geforderte Lösung für (**??**) und wir sind fertig. Andernfalls ist y eine Lösung von

$$(A_{\cdot 1}, \ldots, A_{\cdot n-1}, -A_{\cdot n})^T y \geq 0 \tag{3.17a}$$

$$b^T y < 0. \tag{3.17b}$$

Wir betrachten nun die Matrix

$$\bar{A} := (A_{\cdot 1}, \ldots, A_{\cdot n-1}, A_{\cdot n}, -A_{\cdot n}).$$

Nach Konstruktion hat $\bar{A}$ höchstens k nicht-komplementäre Spalten. Nach Induktionsvoraussetzung tritt genau einer der beiden folgenden Fälle ein:

(1) Es gibt ein $\bar{x} \in \mathbb{R}^{n+1}$ mit $\bar{A}\bar{x} = b$ und $\bar{x} \geq 0$.
(2) Es gibt ein $\bar{y} \in \mathbb{R}^m$, so dass $\bar{A}^T \bar{y} \geq 0$ und $b^T \bar{y} < 0$.

Im Fall (2) gilt $A_{\cdot j}^T \bar{y} \geq 0$ für $j = 1, \ldots, n-1$, sowie $A_{\cdot n}^T \bar{y} \geq 0$ und $-A_{\cdot n}^T \bar{y} \geq 0$, also $A_{\cdot n}^T \bar{y} = 0$. Daher ist $\bar{y}$ eine Lösung von (**??**) und wir sind fertig.

Wir definieren den Vektor $x \in \mathbb{R}^n$ durch

$$x_i := \begin{cases} \bar{x}_i, & \text{für } i = 1, \ldots, n-1 \\ \bar{x}_n - \bar{x}_{n+1}, & \text{für } i = n. \end{cases}$$

Dann gilt $Ax = b$. Es kann nicht $x \geq 0$ gelten, da wir sonst eine Lösung von (**??**) gefunden hätten im Widerspruch zur Unlösbarkeit des Systems. Wegen $\bar{x} \geq 0$ muss $x_n < 0$, also $-x_n > 0$ gelten. Für die Lösung y aus (**??**) folgt dann:

$$
\begin{aligned}
0 > b^T y &= y^T b \\
&= y^T A x = y^T \left(A_{\cdot 1}, \ldots, A_{\cdot n-1}, A_{\cdot n}, A_{\cdot n}\right)(x_1, \ldots, x_n) \\
&= y^T \left(A_{\cdot 1}, \ldots, A_{\cdot n-1}, A_{\cdot n}, -A_{\cdot n}\right)(x_1, \ldots, -x_n)^T \\
&= \left(\underbrace{(A_{\cdot 1}, \ldots, A_{\cdot n-1}, A_{\cdot n}, -A_{\cdot n})^T y}_{\geq 0 \text{ nach } (??)} \right) \underbrace{(x_1, \ldots, -x_n)^T}_{\geq 0} \\
&\geq 0,
\end{aligned}
$$

also ein Widerspruch. □

Farkas' Lemma hat folgende geometrische Interpretation: Ist $Ax = b$ für ein $x \geq 0$, so liegt b im Kegel, der von den Spalten $A_{\cdot 1}, \ldots, A_{\cdot n}$ der Matrix A aufgespannt wird. Gibt es andererseits ein y mit $A^T y \geq 0$, also $A_{\cdot i}^T y \leq 0$ für $i = 1, \ldots, n$ und $y^T b < 0$, so ist y der Normalenvektor einer Hyperebene, die b von $A_{\cdot 1}, \ldots, A_{\cdot n}$ trennt. Die geometrische Formulierung des Farkas' Lemma lautet also:

> Entweder liegt b im Kegel, der von den Vektoren $A_{\cdot 1}, \ldots, A_{\cdot n}$ aufgespannt wird, oder es gibt eine Hyperebene, die b von $A_{\cdot 1}, \ldots, A_{\cdot n}$ trennt.

Beispiel 3.5. Wir betrachten die Matrix

$$A = \begin{pmatrix} 1 & 2 \\ 2 & 1 \end{pmatrix}.$$

(1) Für $b_1 = \binom{3}{3}$ ist $\bar{x} = \binom{1}{1}$ eine Lösung von $Ax = b_1$, das heißt, b_1 liegt im Kegel, der von $A_{\cdot 1} = \binom{1}{2}$ und $A_{\cdot 2} = \binom{2}{1}$ aufgespannt wird, vgl. Abbildung **??**.

(2) Für $b_2 = \binom{3}{0}$ betrachten wir den Vektor $y = \binom{-1}{2}$. Dann hat y mit $A_{\cdot 1} = \binom{1}{2}$ und $A_{\cdot 2} = \binom{2}{1}$ ein positives Skalarprodukt (einen spitzen Winkel) und mit b_2 ein negatives Skalarprodukt (einen stumpfen Winkel), vgl. Abbildung **??**.

Korollar 3.6 (Farkas' Lemma, Variante). *Sei $A \in \mathbb{R}^{m \times n}$ eine Matrix und $b \in \mathbb{R}^m$. Dann hat genau eines der beiden Systeme*

$$Ax \leq b \tag{3.18a}$$

und

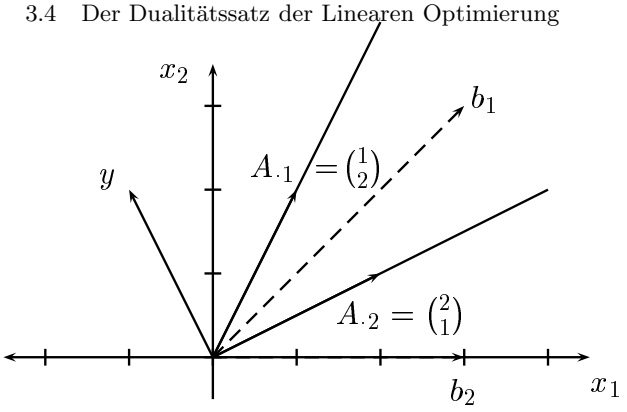


Abb. 3.1. Geometrische Interpretation des Farkas-Lemma

$$A^T y = 0 \tag{3.19a}$$
$$y \geq 0 \tag{3.19b}$$
$$b^T y < 0 \tag{3.19c}$$

eine Lösung.

Beweis. Wir führen im System $Ax \leq b$ eine Schlupfvariable $s \geq 0$ ein und ersetzen x durch $x^+ - x^-$, wobei $x^+, x^- \geq 0$. Dann ist $Ax \leq b$ genau dann lösbar, wenn $Ax^+ - Ax^+ + s = b$, $x^+, x^-, s \geq 0$ lösbar ist.

Sei $A' = (A, -A, I)$. Nach obigen Überlegungen hat $Ax \leq b$ genau dann eine Lösung, wenn $A' \begin{pmatrix} x^+ \\ x^- \\ s \end{pmatrix} = b$, $x^+, x^-, s \geq 0$ eine Lösung besitzt. Dies ist nach der ersten Variante von Farkas' Lemma (Satz **??**) genau dann der Fall, wenn $(A')^T y \geq 0$, $b^T y < 0$ unlösbar ist. Dies ist aber mit $A' = (A, -A, I)$ wiederum äquivalent zu $A^T y \geq 0$, $-A^T y \geq 0$, $y \geq 0$, $b^T y < 0$, also zu (**??**). □

3.4 Der Dualitätssatz der Linearen Optimierung

Wir haben jetzt alle Hilfsmittel beisammen, um den Dualiätssatz der Linearen Optimierung zu beweisen:

Theorem 3.7 (Dualitätssatz der Linearen Optimierung). *Für das Paar*

$$\begin{array}{llll} \textit{(P)} & \min & c^T x & \\ & & Ax = b & \\ & & x \geq 0 & \end{array} \qquad \begin{array}{llll} \textit{(D)} & \max & b^T y & \\ & & A^T y \leq c & \end{array}$$

von dualen Linearen Programmen sind folgende Aussagen äquivalent:

(i) (P) und (D) haben beide zulässige Lösungen
(ii) (P) und (D) haben beide optimale Lösungen x^* bzw. y^* und es gilt $c^T x^* = b^T y^*$.
(iii) (P) oder (D) hat eine endliche Optimallösung.

Beweis. „(i)⇒(ii)“: Nach Korollar **??** genügt es zu zeigen, dass es zulässige Lösungen x^* von (P) und y^* von (D) gibt mit $c^T x^* = b^T y^*$. Da nach der schwachen Dualität (Lemma **??**) sowieso $c^T x^* \geq b^T y^*$ gilt, reicht es aus, zu zeigen, dass für geeignete zulässige Lösungen x^* und y^* gilt: $c^T x^* \leq c^T y^*$. Dies ist genau dann der Fall, wenn folgendes System lösbar ist:

$$Ax = b \tag{3.20a}$$
$$A^T y \leq c \tag{3.20b}$$
$$c^T x - b^T y \leq 0 \tag{3.20c}$$
$$x \geq 0. \tag{3.20d}$$

Ersetzen wir y durch $y^+ - y^-$, wobei $y^+, y^- \geq 0$, führen wir in (**??**) die Schlupfvariable $s \geq 0$ und in (**??**) die Schlupfvariable $\alpha \geq 0$ ein, so erhalten wir die äquvialente Form von (**??**):

$$\left[\begin{array}{r} Ax = b \\ A^T y^+ - A^T y^- + s = c \\ c^T x - b^T y^+ + b^T y^- + \alpha = 0 \\ x, y^+, y^-, s \geq 0 \end{array}\right]$$

$$\Leftrightarrow \begin{pmatrix} A & 0 & 0 & 0 & 0 \\ 0 & A^T & -A^T & I & 0 \\ c^T & -b^T & b^T & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} x \\ y^+ \\ y^- \\ s \\ \alpha \end{pmatrix} = \begin{pmatrix} b \\ c \\ 0 \end{pmatrix} \tag{3.21a}$$

$$x, y^+, y^-, s, \alpha \geq 0 \tag{3.21b}$$

Wir nehmen an, dass (**??**) keine Lösung besitzt. Nach Farkas' Lemma (Satz **??**) ist dann folgendes System lösbar:

$$\begin{pmatrix} A & 0 & 0 & 0\,0 \\ 0 & A^T & -A^T & I\,0 \\ c^T & -b^T & b^T & 0\,1 \end{pmatrix}^T \cdot \begin{pmatrix} u \\ v \\ \gamma \end{pmatrix} \geq 0$$

$$\begin{pmatrix} b \\ c \\ 0 \end{pmatrix}^T \begin{pmatrix} u \\ v \\ \gamma \end{pmatrix} < 0$$

$$\Leftrightarrow \begin{bmatrix} A^T u + c\gamma \geq 0 \\ Av - b\gamma \geq 0 \\ -Av + b\gamma \geq 0 \\ v \geq 0 \\ \gamma \geq 0 \\ b^T u + c^T v < 0 \end{bmatrix} \Leftrightarrow \begin{bmatrix} A^T u + c\gamma \geq 0 \\ Av - b\gamma = 0 \\ v \geq 0 \\ \gamma \geq 0 \\ b^T u + c^T v < 0 \end{bmatrix} \tag{3.22}$$

Sei u, v, γ eine Lösung von (**??**). Ist $\gamma = 0$, so bilden u und v eine Lösung von

$$\begin{bmatrix} A^T u \geq 0 \\ Av- \geq 0 \\ -Av+ \geq 0 \\ v \geq 0 \\ b^T u + c^T v < 0 \end{bmatrix} \Leftrightarrow \begin{pmatrix} A^T & 0 \\ 0 & A \\ 0 & -A \\ 0 & I \end{pmatrix} \begin{pmatrix} u \\ v \end{pmatrix} \geq 0$$

$$\begin{pmatrix} b \\ c \end{pmatrix}^T \begin{pmatrix} u \\ v \end{pmatrix} < 0$$

und nach Farkas' Lemma (Satz ??) wäre dann das System

$$\begin{aligned} Ax &= b \\ A^T y^+ - A^T y^- + s &= c \\ x, y^+, y^-, s &\geq 0 \end{aligned}$$

unlösbar, was der Voraussetzung widerspricht, dass sowohl (P) als auch (D) zulässige Lösungen haben.

Es gilt also $\gamma > 0$. Wir können dann in (**??**) jede Zeile durch $\gamma > 0$ teilen und sehen, dass u/γ, v/γ, 1 ebenfalls eine Lösung von (**??**) bildet. Daher können wir ohne Einschränkung $\gamma = 1$ annehmen, so dass u und v folgendes System lösen:

$$A^T u \geq -c \tag{3.23a}$$
$$Av = b \tag{3.23b}$$
$$v \geq 0 \tag{3.23c}$$
$$b^T u + c^T v < 0 \tag{3.23d}$$

Dann folgt aber

$$0 \overset{(??)}{>} b^T u + c^T v \overset{(??)}{=} u^T A v + c^T v \overset{(??),(??)}{\geq} (-A^T u)v = u^T A v - u^T A v = 0,$$

also ein Widerspruch. Folglich ist (**??**) unlösbar und damit (**??**) wie gewünscht lösbar.
„(ii)⇒(iii)“: trivial
„(iii)⇒(i)“: Es habe zunächst (D) eine endliche Optimallösung y^*. Wäre (P) unzulässig, so gäbe es nach Farkas' Lemma (Satz ??) ein y mit $A^T y \geq 0$ und $b^T y < 0$. Für $u := -y$ gilt dann $A^T u \leq 0$ und $b^T u > 0$. Dann ist aber für beliebiges $\lambda > 0$ der Vektor $y^* + \lambda u$ wegen

$$A^T(y^* + \lambda u) = \underbrace{A^T y^*}_{\leq c} + \lambda \underbrace{A^T u}_{\leq 0} \leq c$$

zulässig für (D) und es gilt $b^T(y^* + \lambda u) = b^T y^* + \lambda b^T u > b^T y^*$ im Widerspruch zur Optimalität von y^*.

Es habe nun (P) eine Optimallösung x^*. Hätte (D) keine zulässige Lösung, so wäre $A^T y \leq c$ unlösbar. Nach Korollar **??** gibt es dann ein $x \geq 0$ mit $Ax = 0$ und $c^T x < 0$. Für $\lambda > 0$ ist dann $x^* + \lambda x$ zulässig für (P) und $c^T(x^* + \lambda x) = c^T x^* + \lambda c^T x < c^T x^*$ im Widerspruch zur Optimalität von x^*. □

Korollar 3.8. *Sei P bzw. D die Lösungsmenge von (P) bzw. (D):*

$$\begin{array}{llll} \textit{(P)} & \min & c^T x & \\ & & Ax = b & \\ & & x \geq 0 & \end{array} \qquad \begin{array}{llll} \textit{(D)} & \max & b^T y & \\ & & A^T y \leq c & \end{array}$$

Setzen wir

$$z^* := \begin{cases} -\infty, & \textit{falls (P) unbeschränkt ist,} \\ +\infty, & \textit{falls } P = \emptyset, \\ \max\left\{ c^T x : x \in P \right\}, & \textit{sonst,} \end{cases}$$

$$w^* := \begin{cases} +\infty, & \textit{falls (D) unbeschränkt,} \\ -\infty, & \textit{falls } D = \emptyset, \\ \min\left\{ b^T y : y \in D \right\}, & \textit{sonst.} \end{cases}$$

Dann gilt:

(a) $z^* = -\infty \Rightarrow D = \emptyset$
(b) $w^* = +\infty \Rightarrow P = \emptyset$
(c) $P = \emptyset \Rightarrow (D = \emptyset$ *oder* $w^* = +\infty)$
(d) $D = \emptyset \Rightarrow (P = \emptyset$ *oder* $z^* = -\infty)$

Beweis. (a) Wäre $y \in D \neq \emptyset$, so folgt aus dem schwachen Dualitätssatz (Lemma **??**), dass $z^* \geq b^T y$ gilt, was $z^* = -\infty$ widerspricht.
(b) Der Beweis erfolgt analog zu (a).
(c) Nach dem Dualitätssatz (Satz **??**) kann nicht gleichzeitig $P = \emptyset$ und $|w^*| < \infty$ eintreten. Daraus folgt die Behauptung.
(d) Der Beweis erfolgt analog zu (c). □

Die Umkehrung des obigen Satzes gilt im Allgemeinen nicht, denn es kann gleichzeitig $P = \emptyset$ und $D = \emptyset$ gelten, wie folgendes Beispiel zeigt.

Beispiel 3.9. Sei

$$P = \left\{ x \in \mathbb{R}^2 \;\middle|\; \begin{array}{c} x_1 - x_2 = 1 \\ -x_1 + x_2 = 1 \end{array}, x \geq 0 \right\}$$
$$D = \left\{ y \in \mathbb{R}^2 \;\middle|\; \begin{array}{c} y_1 - y_2 \leq 0 \\ -y_1 + y_2 \leq -1 \end{array} \right\}.$$

Dann sind die linearen Programme

$$\text{(P)} \quad \begin{array}{c} \min -x_2 \\ x \in P \end{array} \quad \text{und} \quad \text{(D)} \quad \begin{array}{c} \max y_1 + y_2 \\ y \in D \end{array}$$

dual zueinander und es gilt:

(a) $D = \emptyset$, da nach der Variante des Farkas' Lemmas aus Korollar **??** das System

$$\begin{aligned} u_1 - u_2 &= 0 \\ -u_1 + u_2 &= 0 \\ -u_2 &< 0 \\ u_1,\, u_2 &\geq 0 \end{aligned}$$

eine Lösung hat, nämlich: $u_1 = u_2 = 1$.
(b) $P = \emptyset$, da nach Farkas' Lemma (Satz **??**) das System

$$\begin{aligned} u_1 - u_2 &\geq 0 \\ -u_1 + u_2 &\geq 0 \\ u_1 + u_2 &< 0 \end{aligned}$$

eine Lösung hat, nämlich: $u_1 = u_2 = -1$.

Betrachten wir noch einmal das Paar (P) und (D) von dualen Linearen Programmen:

$$\begin{array}{llll} \text{(P)} & \min & c^T x & \\ & & Ax = b & \\ & & x \geq 0 & \end{array} \qquad \begin{array}{llll} \text{(D)} & \max & b^T y & \\ & & A^T y \leq c & \end{array}$$

Wir sind in unserer Herleitung der Dualität von (P) ausgegangen und haben dann das duale Programm (D) erhalten. Wir zeigen jetzt, dass die Dualisierung in gewisser Weise symmetrisch ist, wir also umgekehrt bei der Dualisierung von (D) auch wieder (P) erhalten. Dazu schreiben wir (D) äquivalent durch Einführung von Schlupfvariablen $s \geq 0$ und Ersetzen von y durch $y = y^+ - y^-$ um:

$$\begin{aligned} \text{(D)} \quad -\min \quad & \begin{pmatrix} -b \\ b \\ 0 \end{pmatrix}^T \begin{pmatrix} y^+ \\ y^- \\ s \end{pmatrix} \\ & \begin{pmatrix} A^T & -A^T & I \end{pmatrix} \begin{pmatrix} y^+ \\ y^- \\ s \end{pmatrix} = c \\ & \begin{pmatrix} y^+ \\ y^- \\ s \end{pmatrix} \geq 0 \end{aligned}$$

Damit erhalten wir ein Lineares Programm in Standardform, zu dem wir rein mechanisch das Duale Lineare Programm bilden können. Dies lautet dann:

$$\begin{array}{llll} -\max & c^T x & \Leftrightarrow -\max & -c^T(-x) \\ & \begin{pmatrix} A \\ -A \\ I \end{pmatrix} x \leq \begin{pmatrix} -b \\ b \\ 0 \end{pmatrix} & & A(-x) = b \\ & & & (-x) \leq 0 \end{array} \quad \begin{array}{ll} \Leftrightarrow \min & c^T x \\ & Ax = b \\ & x \geq 0 \end{array}$$

Dies zeigt, dass das Duale zum Dualen Linearen Programm wieder das primale Lineare Programm (P) ist.

Das bislang behandelte Konzept der Dualität lässt sich mit den oben benutzten „Tricks“ (Einführen von Schlupfvariablen etc.) auch auf beliebige lineare Programme verallgemeinern.

Theorem 3.10. *Gegeben seien dimensionsverträgliche Matrizen A, B, C, D, und Vektoren a, b, c, d. Betrachte das (primale) lineare Programm*

$$\begin{aligned} \min \quad & c^T x + d^T y \\ & Ax + By \geq a \\ & Cx + Dy = b \\ & x \geq 0. \end{aligned} \tag{3.24}$$

Dann gilt:

(a) Das zu (??) duale Programm lautet

$$\begin{aligned} \max \quad & a^T u + b^T v \\ & A^T u + C^T v \leq c \\ & B^T u + D^T v = d \\ & u \geq 0. \end{aligned} \tag{3.25}$$

(b) Das zu (??) duale Programm ist (??).
(c) Satz ?? und Korollar ?? gelten analog für (??) und (??).

Beweis. (a) Mit Hilfe der Transformationsregeln schreiben wir (??) als

$$\begin{aligned} \min \quad & c^T x + d^T y^+ - d^T y^- \\ & Ax + By^+ - By^- - Is = a \\ & Cx + Dy^+ - Dy^- + 0s = b \\ & x, s, y^+, y^+ \geq 0, \end{aligned}$$

also in Standardform. Das hierzu duale Lineare Programm ist nach Definition

$$\begin{aligned} \max \quad & a^T u + b^T v \\ & A^T u + C^T v \leq c \\ & B^T u + D^T v \leq d \\ & -B^T u - D^T v \leq -d \\ & -u \leq 0 \end{aligned}$$

Dies entspricht (??).

(b) Der Beweis erfolgt analog zu (a).

(c) Durch Anwendung der Transformationsregeln folgt die Behauptung analog zu Teil (a) unter Verwendung von Satz ?? und Korollar ??. □

In Tabelle ?? sind einige Merkregeln zusammengefasst, wie man aus einem primalen linearen Programm das zugehörige duale ableitet und umgekehrt. Die dargestellten Transformationsregeln sind aus den bisherigen primal-dualen Relationen abgeleitet.

Primal	Dual
Gleichung / Ungleichung	Variable
Ungleichung	nichtnegative Variable
Gleichung	nicht vorzeichenbeschränkte Variable
nichtnegative Variable	Ungleichung
nicht vorzeichenbeschränkte Variable	Gleichung

Tabelle 3.1. Transformationsregeln

3.5 Komplementarität

Theorem 3.11 (Satz vom schwachen komplementären Schlupf). *Gegeben seien dimensionsverträgliche Matrizen A, B, C, D und Vektoren a, b, c, d. Betrachte das zueinander gehörende Paar dualer Programme Gegeben seien dimensionsverträgliche Matrizen A, B, C, D, und Vektoren a, b, c, d.*

Betrachte das zueinander gehörende Paar dualer linearer Programme (??) und (??):

$$\begin{array}{lll} (P) & \min & c^T x + d^T y \\ & & Ax + By \geq a \\ & & Cx + Dy = b \\ & & x \geq 0. \end{array} \quad \textit{und} \quad \begin{array}{lll} (D) & \max & a^T u + b^T v \\ & & A^T u + C^T v \leq c \\ & & B^T u + D^T v = d \\ & & u \geq 0. \end{array}$$

Die Vektoren $\binom{\bar{x}}{\bar{y}}$ und $\binom{\bar{u}}{\bar{v}}$ seien zulässig für (P) bzw. (D). Dann sind folgende Aussagen äquivalent:

(i) $\binom{\bar{x}}{\bar{y}}$ ist optimal für (P), $\binom{\bar{u}}{\bar{v}}$ ist optimal für (D).

(ii) $\big(c - (A^T \bar{u} + C^T \bar{v})\big)^T \bar{x} + \bar{u}^T \big((A\bar{x} + B\bar{y} - a)\big) = 0.$

(iii) Für alle Komponenten $\bar{u}_i$ der Duallösung gilt:

$$\bar{u}_i > 0 \Rightarrow A_{i\cdot}\bar{x} + B_{i\cdot}\bar{x} = a_i$$

(d.h., ist $\bar{u}_i > 0$, so ist die ite Ungleichung in $A\bar{x} + B\bar{y} \geq a$ mit Gleichheit erfüllt).
Für alle Komponenten $\bar{x}_j$ der Primallösung gilt:

$$\bar{x}_j > 0 \Rightarrow A_{\cdot j}\bar{u} + C_{\cdot j}v = c_j$$

(d.h., ist $\bar{x}_j > 0$, so ist die jte Ungleichung in $A^T\bar{u} + C^T\bar{v} \leq c$ mit Gleichheit erfüllt).

(iv) Für alle Zeilenindizes i von A und B gilt:

$$A_{i\cdot}\bar{x} + B_{i\cdot}\bar{x} > a_i \Rightarrow \bar{u}_i = 0$$

(d.h., ist die ite Ungleichung strikt erfüllt, so ist die zugehörige Dualvariable $\bar{u}_i$ gleich 0)
Für alle Spaltenindizes j von A und C gilt:

$$A_{\cdot j}\bar{u} + C_{\cdot j}\bar{v} < c_j \Rightarrow \bar{x}_j = 0$$

(d.h., ist die jte Ungleichung strikt erfüllt, so ist die zugehörige Variable $\bar{x}_j$ gleich 0).

Beweis. „(i)⇔(ii)“: Nach dem Dualitätssatz **??** ist $\binom{\bar{x}}{\bar{y}}$ optimal für (P) und $\binom{\bar{u}}{\bar{v}}$ optimal für (D) genau dann, wenn $c^T\bar{x} + d^T\bar{y} = a^T\bar{u} + b^T\bar{v}$. Also haben wir

$$\begin{aligned}
\text{(i)} &\Leftrightarrow c^T\bar{x} + d^T\bar{y} = a^T\bar{u} + b^T\bar{v} && \text{(nach Satz **??**)}\\
&\Leftrightarrow c^T\bar{x} + (B^T\bar{u} + D^T\bar{v})^T\bar{y} = a^T\bar{u} + (C\bar{x} + D\bar{y})^T\bar{v} && \text{(da } B^T\bar{u} + D^Tv = d \text{ und } C\bar{x} + D\bar{y} = b\text{)}\\
&\Leftrightarrow (c - C^T\bar{v})^T\bar{x} + \bar{u}^T(B\bar{y} - a) = 0\\
&\Leftrightarrow (c - C^T\bar{v})^T\bar{x} - (A^T\bar{u})^T\bar{x} + \bar{u}^T(B\bar{y} - a) + \bar{u}^TAx\\
&\Leftrightarrow (c - C^T\bar{v} - A^T\bar{u})^T\bar{x} + \bar{u}^T(B\bar{y} + A\bar{x} - a) = 0\\
&\Leftrightarrow \text{(ii)}
\end{aligned}$$

„(ii)⇒(iii)“:
Sei $t := c - (A^T\bar{u} + C^T\bar{v})$ und $s := A\bar{x} + B\bar{y} - a$. Da $\binom{\bar{x}}{\bar{y}}$ zulässig für (P) ist, gilt $s \geq 0$. Analog folgt aus der Zulässigkeit von $\binom{\bar{u}}{\bar{v}}$ für (D), dass $t \geq 0$. Aus $\bar{x} \geq 0$, $\bar{u} \geq 0$, $s \geq 0$, $t \geq 0$ folgt daher $t^T\bar{x} + \bar{u}^Ts \geq 0$. Nach Voraussetzung ist $t^T\bar{x} + \bar{u}^Ts = 0$ und dies kann nur gelten, falls $t^T\bar{x} = \bar{u}^Ts = 0$ gilt. Dies impliziert (iii).
„(iii)⇒(ii)“: Klar
„(iv)⇔(iii)“: Beweis folgt sofort durch Negation. □

Für das Lineare Programm in Standardform (**??**) und sein Duales (**??**) besitzen die Komplementaritätsbedingungen eine etwas einfachere Form:

Korollar 3.12 (Satz vom schwachen komplementären Schlupf). *Betrachte das Paar dualer Linearer Programme:*

$$\begin{array}{llll}
\textit{(P)} \quad \min & c^Tx & \textit{(D)} \quad \max & b^Ty\\
& Ax = b & & A^Ty \leq c\\
& x \geq 0 & &
\end{array}$$

und sei $\bar{x}$ zulässig für (P) und $\bar{y}$ zulässig für (D).
Dann sind folgende Aussagen äquivalent:

(i) $\bar{x}$ *ist optimal für (P),* $\bar{y}$ *ist optimal für (D).*
(ii) $(c - A^T\bar{y})^T\bar{x} = 0$
(iii) Für alle Komponenten $\bar{x}_j$ *der Primallösung gilt:*

$$\bar{x}_j > 0 \Rightarrow A_{\cdot j}\bar{y} = c_j$$

(d.h., ist $\bar{x}_j > 0$*, so ist die* j*te Ungleichung in* $A^T\bar{y} \leq c$ *mit Gleichheit erfüllt).*
(iv) Für alle Spaltenindizes j *von* A

$$A_{\cdot j}\bar{y} < c_j \Rightarrow \bar{x}_j = 0$$

(d.h., ist die j*te Ungleichung in (D) für* $\bar{y}$ *strikt erfüllt, so ist die zugehörige Variable* $\bar{x}_j$ *gleich* 0*).*

Beweis. Unmittelbar aus Satz **??**. □

Anmerkung 3.13. Die Umkehrungen in (iii) und (iv) gelten im Allgemeinen nicht. Es gibt jedoch immer Paare von Lösungen, die diese erfüllen.

Theorem 3.14 (Satz vom starken komplementären Schlupf). *Betrachte das Paar dualer Linearer Programme:*

$$\begin{array}{llll} (P) & \min & c^T x & \\ & & Ax = b & \\ & & x \geq 0 & \end{array} \qquad \begin{array}{llll} (D) & \max & b^T y & \\ & & A^T y \leq c & \end{array}$$

Besitzen sowohl (P) als auch (D) zulässige Lösungen, so existieren Optimallösungen $\bar{x}$ *von (P) und* $\bar{y}$ *von (D) mit*

$$\bar{x}_j > 0 \Leftrightarrow A_{\cdot j}\bar{y} = c_j$$
$$\bar{x}_j = 0 \Leftrightarrow A_{\cdot j}\bar{y} < c_j$$

Beweis. Aus dem Dualiätssatz (Satz **??**) folgt, dass sowohl (P) als auch (D) Optimallösungen besitzen. Nach Korollar **??** genügt es zu zeigen, dass es Optimallösungen $\bar{x}$ von (P) und $\bar{y}$ von (D) gibt mit:

$$\bar{x}_j > 0 \Leftarrow A_{\cdot j}\bar{y} = c_j$$
$$\bar{x}_j = 0 \Rightarrow A_{\cdot j}\bar{y} < c_j$$

□

3.6 Dualität und die Simplex-Methode

Wir werfen einen weiteren Blick auf die Dualität, diesmal von der Seite der Simplex-Methode. Dazu betrachten wir wieder das Paar dualer Linearer Programme:

$$\begin{array}{llll} \text{(P)} & \min & c^T x & \\ & & Ax = b & \\ & & x \geq 0, & \end{array} \qquad \begin{array}{llll} \text{(D)} & \max & b^T y & \\ & & A^T y \leq c & \end{array}$$

wobei wir diesmal annehmen, dass $\operatorname{Rang} A = m$ gilt, also A vollen Zeilenrang besitzt und die Menge der zulässigen Lösungen von (P) nichtleer ist (Diese Voraussetzungen können wir mit Hilfe der Phase I der Simplexmethode wie in Abschnitt **??** sicherstellen).

Unter diesen Voraussetzungen können wir den Fundamentalsatz der Linearen Optimierung (Satz ??) anwenden, der besagt, dass (P) eine zulässige Basislösung besitzt. Wir wenden die Simplex-Methode auf (P) an, wobei wir in der existierenden zulässigen Basislösung starten. Die Simplex-Methode kann mit zwei Ergebnissen abbrechen:

1. Mit einer optimalen Basislösung $\bar{x} = \binom{\bar{x}_B}{\bar{x}_N} = \binom{A_{\cdot B}^{-1} b}{0}$, wobei B die zugehörige Basis ist.
2. Mit der Information, dass (P) unbeschränkt ist.

Behandeln wir zunächt den ersten Fall. Wenn die Simplex-Methode mit der Information abbricht, dass die aktuelle Basislösung $\bar{x}$ optimal ist, so gilt nach Konstruktion für den Vektor $\bar{z}_N = c_N - A_{\cdot N}^T A_{\cdot B}^{-T} c_B$ die Bedingung $\bar{z}_N \geq 0$, also

$$A_{\cdot N}^T A_{\cdot B}^{-T} c_B \leq c_N \tag{3.26}$$

Wir betrachten den Vektor

$$\bar{y} := A_{\cdot B}^{-T} c_B, \tag{3.27}$$

der im Schritt 1 („**BTRAN**“) berechnet wird. Für diesen Vektor $\bar{y}$ gilt:

$$\begin{aligned} A^T \bar{y} &= \begin{pmatrix} A_{\cdot B}^T \\ A_{\cdot N}^T \end{pmatrix} A_{\cdot B}^{-T} c_B = \begin{pmatrix} A_{\cdot B}^T A_{\cdot B}^{-T} c_B \\ A_{\cdot N}^T A_{\cdot B}^{-T} c_B \end{pmatrix} \\ &= \begin{pmatrix} c_B \\ A_{\cdot N}^T A_{\cdot B}^{-T} c_B \end{pmatrix} \overset{(??)}{\leq} \begin{pmatrix} c_B \\ c_N \end{pmatrix} = c \end{aligned}$$

Der Vektor $\bar{y}$ ist also zulässig für das duale Lineare Programm (D). Für seinen Zielfunktionswert $b^T\bar{y}$ ergibt sich:

$$b^T\bar{y} = b^T(A_{.B}^{-T}c_B) = c_B^T A_{.B}^{-1}b = c_B^T\bar{x}_B = c^T\bar{x},$$

und daher ist nach Korollar **??** dann $\bar{y}$ eine optimale Lösung von (D)!

Damit erhalten wir folgendes Resultat:

Theorem 3.15. *Terminiert die Simplex-Methode (Algorithmus 1.1) mit einer optimalen Basislösung* $\bar{x} = \binom{\bar{x}_B}{\bar{x}_N} = \binom{A_{.B}^{-1}b}{0}$ *von (P), so ist* $\bar{y} := A_{.B}^{-T}c_B$ *eine Optimallösung von (D).* □

3.7 Dualität und Sensitivität

Wir sind bisher davon ausgegangen, dass die Daten A, b und c eines linearen Programms der Form

$$\begin{aligned} \min \quad & c^T x && \text{(3.28a)}\\ & Ax = b && \text{(3.28b)}\\ & x \geq 0, && \text{(3.28c)} \end{aligned}$$

fest vorgegeben waren. Häufig ist es jedoch der Fall, dass sich diese Daten im Laufe der Zeit ändern, wenn Bedingungen und/oder Variablen hinzukommen und damit nachoptimiert werden muss.

Gehen wir einmal davon aus, dass wir ein lineares Programm der Form (**??**) bereits gelöst haben und eine optimale Basis B mit zugehöriger Basislösung $\bar{x}_B = A_{.B}^{-1}b$ und $\bar{x}_N = 0$ kennen. Wir nehmen ebenfalls an, dass die Basislösung $\bar{x}$ nicht degeneriert ist, d.h. dass $\bar{x}_B > 0$ gilt. In diesem Fall gilt wie im letzten Abschnitt für den Vektor $\bar{z}_N = c_N - A_{.N}^T A_{.B}^{-T}c_B$ der reduzierten Kosten, dass $\bar{z}_N \geq 0$ ist, und der Vektor $\bar{y} := A_{.B}^{-T}c_B$ aus (**??**) eine Optimallösung des dualen Linearen Programms ist.

Wir betrachten die Situation, wenn sich die rechte Seite b um eine „kleine Störung“ Δb ändert, d.h. wir betrachten das neue Lineare Programm

$$\begin{aligned} \min \quad & c^T x && \text{(3.29a)}\\ & Ax = b + \Delta b && \text{(3.29b)}\\ & x \geq 0 && \text{(3.29c)} \end{aligned}$$

Offenbar ist B immer noch eine Basis des geänderten Linearen Programms (**??**) und

$$\begin{aligned}\bar{x}' = \begin{pmatrix} \bar{x}'_B \\ \bar{x}'_N \end{pmatrix} &= \begin{pmatrix} A_{\cdot B}^{-1} b + A_{\cdot B}^{-1} \Delta b \\ 0 \end{pmatrix} \\ &= \bar{x} + \begin{pmatrix} A_{\cdot B}^{-1} \Delta b \\ 0 \end{pmatrix} =: \bar{x} + \begin{pmatrix} \Delta x \\ 0 \end{pmatrix}.\end{aligned}$$

eine Basislösung. Da nach unserer Voraussetzung B nichtdegeneriert ist, also $\bar{x}_B = A_{\cdot B}^{-1} b > 0$ gilt, ist für kleines Δb auch $\bar{x}_B + \Delta x = \bar{x}_B + A_{\cdot B}^{-1} \Delta b \geq 0$, also $\bar{x}'$ eine zulässige Basislösung für (**??**).

Da der Vektor $\bar{z}_N$ der reduzierten Kosten unverändert bleibt, ist $\bar{x}'$ somit für kleines Δb eine Optimallösung von (**??**). Die entsprechende Änderung in der Zielfunktion errechnet sich als

$$c_B^T(\bar{x}_B + \Delta x) - c_B^T \bar{x}_B = c_B^T \Delta x = c_B^T A_{\cdot B}^{-1} \Delta b = \bar{y}^T \Delta b. \tag{3.30}$$

Gleichung (**??**) zeigt, dass die optimale Duallösung $\bar{y}$ die *Sensitivität* bezüglich kleiner Änderungen in der rechten Seite b misst: Wenn wir b durch $b + \Delta b$ ersetzen, so ändert sich (für kleines Δb) der optimale Zielfunktionswert um $\bar{y}^T \Delta b$.

4

Die Simplexmethode

In Abschnitt 1.2 haben wir die Grundform des Simplex-Verfahrens zur Lösung des Linearen Programms

$$\begin{aligned} \min \quad & c^T x && \text{(4.1a)} \\ & Ax = b && \text{(4.1b)} \\ & x \geq 0 && \text{(4.1c)} \end{aligned}$$

in Standardform kennengelernt. Wir haben gesehen, wie wir ausgehend von einer aktuellen zulässigen Basislösung $x = (x_B, x_N)^T$ mit $x_B := A_{\cdot B}^{-1} b \geq 0$ und $x_N := 0$ entweder eine neue Basislösung mit höchstens besserem Zielfunktionswert finden können oder korrekt schließen können, dass das Problem (**??**) unbeschränkt ist, d.h. Lösungen mit beliebig kleinem Zielfunktionswert besitzt.

In diesem Kapitel beschäftigen wir uns mit den zwei wichtigen Punkten, die noch dazu fehlen, um das Simplex-Verfahren zu einem vollständigen Optimierungsalgorithmus zu machen:

Terminieren: Bricht der Simplex-Algorithmus immer ab oder kann es vorkommen, dass unendlich viele Iterationen durchgeführt werden?

Mit dieser Frage beschäftigen wir uns in Abschnitt **??**. Hier werden wir feststellen, dass die Wiederholung einer Basislösung die einzige Möglichkeit ist, die einen Abbruch nach endlich vielen Schritten verhindert. Wir zeigen dann, dass durch geeignete Wahl der eintretenden bzw. die Basis verlassenden Variablen, dieses sogenannte *Kreiseln* ausgeschlossen werden kann.

Initialisierung: Wie finden wir eine erste zulässige Basis?

Die Initialisierung der Simplex-Methode ist Gegenstand von Abschnitt **??**. Wir zeigen, wie man für jedes Problem (**??**) in Standardform ein sogenanntes *Phase I Problem* aufstellen kann, das wiederum ein Lineares Programm ist und für das man leicht eine zulässige Basis angeben kann. Die Lösung des Phase I-Problems mit der Simplex-Methode liefert dann entweder eine zulässige Startbasis für (**??**) oder die Information, dass (**??**) keine zulässige Lösung besitzt.

4.1 Terminieren

Wir betrachten das folgende Lineare Programm in Standardform:

$$\begin{array}{lrrrrrrrl}
\min & -2.3x_1 & -2.15x_2 & +13.55x_3 & +0.4x_4 & & & & \\
& 0.4x_1 & +0.2x_2 & -1.4x_3 & -0.2x_4 & +x_5 & & = & 0 \\
& -7.8x_1 & +1.4x_2 & +7.8x_3 & +0.4x_4 & & +x_6 & = & 0 \\
& x_1, & x_2, & x_3, & x_4, & x_5, & x_6 & \geq & 0
\end{array}$$

In der Simplex-Methode gibt es im allgemeinen mehrere Möglichkeiten, eine Variable in die Basis aufzunehmen bzw. aus der Basis zu entfernen. Für das Beispiel setzen wir folgende Regeln fest:

- Falls beim **Pricing** es mehrere Nichtbasisvariable j mit $\bar{z}_j < 0$ gibt, so wählen wir diejenige mit kleinstem $\bar{z}_j$ aus, um in die Basis aufgenommen zu werden.
- Falls es beim **Ratio-Test** mehrere Basisvariablen gibt, für die das Minimum angenommen wird, so entfernen wir eine Basisvariable x_{B_k} mit größtem Wert w_k.

Für unser Beispiel ist eine primal zulässige Basis $B = (5, 6)$ und die zugehörige Basislösung gegeben durch

$$\bar{x}_B = \begin{pmatrix} \bar{x}_5 \\ \bar{x}_6 \end{pmatrix} = A_{\cdot B}^{-1} b = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}^{-1} \begin{pmatrix} 0 \\ 0 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$$

$$\bar{x}_N = \begin{pmatrix} \bar{x}_1 \\ \bar{x}_2 \\ \bar{x}_3 \\ \bar{x}_4 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}.$$

Der Zielfunktionswert ist $c^T \bar{x} = c^T 0 = 0$. Wir führen nun die einzelnen Schritte des Simplex-Algorithmus aus.

(1) **BTRAN**: Löse $\bar{y}^T A_{\cdot B} = c_B^T$, d.h.

$$\bar{y}^T \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} = (0,0) \quad \Rightarrow \bar{y}^T = (0,0)^T.$$

(2) **Pricing**: Berechne $\bar{z}_N = c_N - A_{\cdot N}^T \bar{y}$.
Wir haben

$$\bar{z}_N = c_N - A_{\cdot N}^T \begin{pmatrix} 0 \\ 0 \end{pmatrix} = c_N = \begin{pmatrix} -2.3 \\ -2.15 \\ 13.55 \\ 0.4 \end{pmatrix}$$

Die Nichtbasisvariable mit dem kleinsten negativen Wert $\bar{z}_1 = -2.3$ ist x_1. Somit wird x_1 in die Basis aufgenommen und im Simplex-Algorithmus wird $j = 1$ gewählt.

(3) **FTRAN**: Löse $A_{\cdot B} w = A_{\cdot j}$.

$$\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} w = \begin{pmatrix} 0.4 \\ -7.8 \end{pmatrix} \qquad \Rightarrow w = \begin{pmatrix} 0.4 \\ -7.8 \end{pmatrix}$$

Es wird sich nachher als nützlich für die Darstellung herausstellen, wenn wir im **FTRAN**-Schritt nicht nur $w = A_{\cdot B}^{-1} A_{\cdot j}$ berechnen, sondern gleich die komplette Matrix $A_{\cdot B}^{-1} A_{\cdot N}$. Für diese gilt:

$$A_{\cdot B}^{-1} A_{\cdot N} = \begin{pmatrix} 0.4 & 0.2 & -1.4 & -0.2 \\ -7.8 & -1.4 & 7.8 & 0.4 \end{pmatrix} \tag{4.2}$$

Den Vektor w haben wir dabei in der Matrix hervorgehoben.

(4) **Ratio-Test**: Berechne $\gamma = \min\left\{ \frac{\bar{x}_{B_k}}{w_k} : w_k > 0 \text{ und } k \in \{1, \ldots, m\} \right\}$.
Wir haben $\bar{x}_B = (\bar{x}_5, \bar{x}_6)^T = (0,0)^T$. Da alle Basisvariablen den Wert 0 haben und nur $w_1 > 0$ gilt, haben wir $\gamma = 0$ und $x_{B_1} = x_5$ verlässt die Basis.

(5) **Update**:

$$\bar{x}_B = \begin{pmatrix} \bar{x}_5 \\ \bar{x}_6 \end{pmatrix} - \gamma w = \begin{pmatrix} \bar{x}_5 \\ \bar{x}_6 \end{pmatrix} - 0 \cdot w = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$$
$$N := N \setminus \{j\} \cup \{B_i\} = \{1,2,3,4\} \setminus \{1\} \cup \{5\} = \{2,3,4,5\}$$
$$B_1 := 1$$
$$\bar{x}_1 := \gamma = 0$$

Wir erhalten die neue Basislösung $\bar{x}$ zur Basis $B = (1,6)$:

$$\bar{x}_B = \begin{pmatrix} \bar{x}_1 \\ \bar{x}_6 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}, \quad \bar{x}_N = \begin{pmatrix} \bar{x}_2 \\ \bar{x}_3 \\ \bar{x}_4 \\ \bar{x}_5 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}.$$

Im Beispiel oben haben wir zwar eine neue Basis erhalten, die Basislösung stimmt aber mit derjenigen aus der letzten Iteration überein, da im **Ratio-Test** $\gamma = 0$ war.

Definition 4.1 (Degenerierter Pivot, degenerierte Iteration). *Eine Iteration des Simplex-Algoirthmus, bei der im **Ratio-Test** $\gamma = 0$ gilt, heißt* degenerierte Iteration *oder* degenerierter Pivot.

degenerierte Iteration

degenerierter Pivot

Bei einer degenerierten Iteration machen wir im Bezug auf die Zielfunktion keinen Fortschritt, da sich die eigentlichen Lösung nicht verändert. Wir führen nun unser Beispiel weiter fort:

(1) **BTRAN**: Löse $\bar{y}^T A_{\cdot B} = c_B^T$, d.h.

$$\bar{y}^T \begin{pmatrix} 0.4 & 0 \\ 0 & 1 \end{pmatrix} = (-2.3, 0) \quad \Rightarrow \bar{y}^T = (-5.75, 0)^T.$$

(2) **Pricing:** Berechne $\bar{z}_N = c_N - A_{\cdot N}^T \bar{y}$.

$$\bar{z}_N = \begin{pmatrix} -2.15 \\ 13.55 \\ 0.4 \\ 0 \end{pmatrix} - \begin{pmatrix} 0.2 & -1.4 \\ 1.4 & 7.8 \\ -0.2 & 0 \\ 1 & 0 \end{pmatrix} \begin{pmatrix} -5.75 & 0 \end{pmatrix} = \begin{pmatrix} -1 \\ 5.5 \\ -0.75 \\ 5.75 \end{pmatrix}$$

Gemäß unsere Festlegungen wählen wir $j = 1$ mit $\bar{z}_j = -1$ und die zugehörige Nichtbasisvariable x_2 wird in die Basis aufgenommen.

(3) **FTRAN**: Löse $A_{\cdot B} w = A_{\cdot j}$.

$$\begin{pmatrix} 0.4 & 0 \\ -7.8 & 1 \end{pmatrix} w = \begin{pmatrix} 0.2 \\ -1.4 \end{pmatrix} \qquad \Rightarrow w = \begin{pmatrix} 0.5 \\ 2.5 \end{pmatrix}$$

(4) **Ratio-Test**: Berechne $\gamma = \min \left\{ \frac{\bar{x}_{B_k}}{w_k} : w_k > 0 \text{ und } k \in \{1, \ldots, m\} \right\}$. Wir haben $\bar{x}_B = (\bar{x}_1, \bar{x}_6)^T = (0, 0)^T$. Da wieder alle Basisvariablen den Wert 0 haben und $w_2 > 0$ der größte Wert ist, entfernen wir gemäß unserer Vorgaben die Variable $x_{B_2} = x_6$ aus der Basis.

(5) **Update**: Wir erhalten die neue Basis $B = (1, 2)$ mit der zugehörigen Basislösung $\bar{x}_B = (\bar{x}_1, \bar{x}_2)^T = (0, 0)^T$, $\bar{x}_N = (\bar{x}_3, \bar{x}_4, \bar{x}_5, \bar{x}_6)^T = (0, 0, 0, 0)^T$.

Auch die zweite Iteration ist also degeneriert und wir haben keinen Fortschritt in Bezug auf die Zielfunktion. Die eigentliche Lösung $(0,0,0,0,0,0)^T$ hat sich bisher überhaupt nicht verändert, nur die Aufteilung in Basis-/Nichtbasisvariablen.

(1) **BTRAN**: Löse $\bar{y}^T A_{\cdot B} = c_B^T$, d.h.

$$\bar{y}^T \begin{pmatrix} 0.4 & 0.2 \\ -7.8 & -1.4 \end{pmatrix} = (-2.3, -2.15) \quad \Rightarrow \bar{y}^T = (-13.55, -0.4)^T.$$

(2) **Pricing:** Berechne $\bar{z}_N = c_N - A_{\cdot N}^T \bar{y}$.

$$\bar{z}_N = \begin{pmatrix} -13.55 \\ 0.4 \\ 0 \\ 0 \end{pmatrix} - \begin{pmatrix} -1.4 & 7.8 \\ -0.2 & 0.4 \\ 1 & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} -13.55 & -0.4 \end{pmatrix} = \begin{pmatrix} -2.3 \\ -2.15 \\ 13.55 \\ 0.4 \end{pmatrix}$$

Gemäß unsere Festlegungen wählen wir wieder $j = 1$ mit $\bar{z}_j = -1$ und die zugehörige Nichtbasisvariable x_3 wird in die Basis aufgenommen.

In diesem Moment halten wir kurz inne und vergleichen die aktuelle Situation mit der Situation in der ersten Iteration. Die beiden Vektoren $\bar{z}_N$ stimmen überein (lediglich die Namen der zugehörigen Variablen haben sich geändert). Anstelle im **FTRAN**-Schritt das Gleichungssystem $A_{\cdot B} w = A_{\cdot j}$ zu lösen (also $w = A_{\cdot B}^{-1} A_{\cdot j}$ zu berechnen), berechnen wir wie in der ersten Iteration diesmal $A_{\cdot B}^{-1} A_{\cdot N}$:

$$\begin{aligned} A_{\cdot B}^{-1} A_{\cdot N} &= \begin{pmatrix} 0.4 & 0.2 \\ -7.8 & -1.4 \end{pmatrix}^{-1} \begin{pmatrix} -1.4 & -0.2 & 1 & 0 \\ 7.8 & 0.4 & 0 & 1 \end{pmatrix} \\ &= \begin{pmatrix} 0.4 & 0.2 & -1.4 & -0.2 \\ -7.8 & -1.4 & 7.8 & 0.4 \end{pmatrix} \end{aligned} \tag{4.3}$$

Wieder haben wir den Vektor w der aktuellen Iteration hervorgehoben.

Wenn wir (**??**) mit der Situation in der ersten Iteration (**??**) vergleichen, so sehen wir, dass sich wie im Vektor $\bar{z}_N$ nur die Namen der zugehörigen Variablen geändert haben, genauer gesagt, wir haben eine zyklische Verschiebung der Variablen um zwei Stellen nach rechts: waren in der ersten Iteration von den sechs Variablen $(x_1, x_2, x_3, x_4, x_5, x_6)$ die letzten beiden Variablen Basisvariablen und der Rest Nichtbasisvariablen, so sind jetzt die x_1, x_2 die Basisvariablen. Die aktuelle Iteration verläuft

also genauso wie die erste, die nächste wie die zweite etc., und man sieht nun leicht, dass wir nach weiteren vier Iterationen wieder bei der Ausgangsbasis $B = (5, 6)$ landen (zwei Iterationen verschieben um zwei Stellen). Die Simplex-Methode terminiert also für unser Beispiel nicht.

4.1.1 Kreiseln

kreiselt

Definition 4.2 (Kreiseln). *Wir sagen, dass der Simplex-Algorithmus* kreiselt*, wenn er eine Basis wiederholt.*

Wie der folgende Satz zeigt, ist das Kreiseln die einzige Möglichkeit, dass die Simplex-Methode nicht terminiert.

Theorem 4.3. *Wenn der Simplex-Algorithmus nicht terminiert, so kreiselt er.*

Beweis. Eine Basis besteht aus m Basisvariablen. Es gibt nur endlich viele, genauer gesagt $\binom{n}{m}$, Möglichkeiten m Variablen aus n auszuwählen. Daher gibt es auch nur maximal $\binom{n}{m}$ mögliche Basen. Wenn der Simplex-Algorithmus unendlich viele Iterationen durchführt, muss daher eine Basis wiederholt werden, also der Algorithmus kreiseln. □

Theorem 4.4. *Das Lineare Programm* (1.17) *besitze zulässige Lösungen und die Eigenschaft, dass für alle primal zulässigen Basen B die entsprechende Basislösung $\binom{x_B}{x_N}$ nichtdeneneriert ist, d.h. $x_B := A_B^{-1} b > 0$ gilt. Dann terminiert der Simplex-Algorithmus 1.1 nach endlich vielen Schritten.*

Beweis. Wegen Satz **??** genügt es zu zeigen, dass der Simplex-Algorithmus unter den Voraussetzungen des Satzes nicht kreiselt. Wir betrachten wie in Lemma 1.6 zwei aufeinanderfolgende Basen B und B^+. Dann gilt für die zugehörigen Basislösungen $\bar{x}$ und $\bar{x}^+$:

$$\begin{aligned} c^T \begin{pmatrix} \bar{x}_{B^+} \\ \bar{x}_{N^+} \end{pmatrix} &= c_{B^+}^T \bar{x}_{B^+} = c_B^T \bar{x}_{B^+} + c_j \bar{x}_j - c_{B_i} \bar{x}_j \\ &= c_B^T (\bar{x}_B - \gamma w) + c_{B_i} \bar{x}_j + c_j \bar{x}_j - c_{B_i} \bar{x}_j \\ &= c_B^T \bar{x}_B + \underbrace{(c_j - c_B^T A_B^{-1} A_{\cdot j})}_{<0} \underbrace{\gamma}_{>0} \\ &< c_B^T \bar{x}_B \quad = \quad c^T \begin{pmatrix} \bar{x}_B \\ \bar{x}_N \end{pmatrix} \end{aligned}$$

Ist nun $B_1, B_2, \ldots$ eine Folge von Basen, die im Algorithmus 1.1 erzeugt werden, so folgt daraus, dass keine Basis innerhalb der Folge doppelt auftreten kann, da die Zielfunktionswerte streng monoton fallen. Da es nur endlich viele Basen gibt, folgt damit die Behauptung. □

4.1.2 Die Regel von Bland

Im Fall von degenerierten Basislösungen kann die Simplex-Methode kreiseln. Wir zeigen nun, dass wir dieses Kreiseln durch geeignete Wahl der eintretenden bzw. die Basis verlassenden Variablen verhindern können. Nach der *Regel von Bland* wählt man sowohl für die eintretende als auch für die verlassende Variable unter allen möglichen Variablen diejenige mit kleinstem Index:

Regel von Bland

Eintretende Variable: Falls beim **Pricing** es mehrere Nichtbasisvariable j mit $\bar{z}_j < 0$ gibt, so wählen wir diejenige mit kleinstem Index j aus, um in die Basis aufgenommen zu werden.

Verlassende Variable: Falls es beim **Ratio-Test** mehrere Basisvariablen gibt, für die das Minimum angenommen wird, so entfernen wir eine Basisvariable x_{B_k} mit kleinstem Index B_k.

Bevor wir zeigen, dass die Regel von Bland das Kreiseln verhindert, wiederholen wir kurz noch einmal wichtige Ergebnisse über das Simplex-Verfahren, die wir im Beweis verwenden werden: Ist B eine Basis von A mit zugehöriger Basislösungung $\bar{x} = \binom{A_{\cdot B}^{-1} b}{0}$, so gilt nach (1.15) für jede (nicht notwendigerweise zulässige) Lösung x von $Ax = b$:

$$c^T x = c^T \bar{x} + \bar{z}_N^T x_N = c^T \bar{x} + (c_N^T - c_B^T A_{\cdot B}^{-1} A_{\cdot N}) x_N. \quad (4.4)$$

Es stellt sich für den folgenden Beweis als nützlich heraus, den Vektor $\bar{z}(B) \in \mathbb{R}^n$ wie folgt zu definieren:

$$\bar{z}_i(B) := \begin{cases} \bar{z}_i & \text{falls } i \in N \\ 0 & \text{falls } i \in B. \end{cases} \quad (4.5)$$

Mit dieser Notation schreibt sich (**??**) wie folgt:

$$c^T x = c^T \bar{x} + \bar{z}(B)^T x \text{ für jede Lösung } x \text{ von } Ax = b. \quad (4.6)$$

Im Simplex-Verfahren nehmen wir eine Nichtbasisvariable $j \in N$ nur dann in die Basis auf, wenn

$$\bar{z}_j(B) < 0 \tag{4.7}$$

gilt. Durch Erhöhung des Wertes von x_j auf $t \geq 0$ bei gleichzeitigem Fixieren aller weiteren Nichtbasisvariablen auf 0 ändert sich der Wert der Basisvariable $x_i \in B$ auf:

$$\bar{x}_i - w_i(B)t, \tag{4.8}$$

wobei $w(B) := w = A_{.B}^{-1} A_{.j}$ gilt. Wir haben hier den Vektor w auch mit der Basis B indiziert, weil im folgenden Beweis verschiedene Basen auftauchen und wir Verwirrungen vermeiden wollen. Nach Konstruktion des Simplex-Verfahrens wird also nur dann eine Basisvariable $x_i \in B$ aus B entfernt, falls

$$w_i(B) > 0. \tag{4.9}$$

gilt.

Theorem 4.5. *Werden eintretende und verlassende Variable nach der Regel von Bland gewählt, so terminiert die Simplex-Methode nach endlich vielen Iterationen entweder mit einer Optimallösung oder der Information, dass das Problem (**??**) unbeschränkt ist.*

Beweis. Nach Satz 1.9 auf Seite 18 liefert das Simplex-Verfahren bei Abbruch entweder eine Optimallösung oder die korrekte Information über die Unbeschränktheit. Darüberhinaus zeigt Satz **??**, dass Kreiseln die einzige Möglichkeit ist, die Terminieren verhindert. Es genügt also zu beweisen, dass das Simplex-Verfahren mit der Regel von Bland nicht kreiseln kann.

Wir führen einen Widerspruchsbeweis, indem wir annehmen, dass das Vefahren kreiselt. Falls das Verfahren kreiselt, so durchläuft es eine Folge $B^{(p)}, B^{(p+1)}, \ldots, B^{(p+k)} = B^{(p)}$ von zulässigen Basen, wobei die Basis $B^{(p)} = B^{(p+k)}$ wiederholt wird. Wir nennen eine Variable x_i *unbeständig*, wenn sie in mindestens einer Basis der obigen Folge Basisvariable und in mindestens einer weiteren Basis der Folge Nichtbasisvariable ist. Wir setzen

$$f := \max \{\, i : x_i \text{ ist unbeständig} \,\}, \tag{4.10}$$

d.h. x_f ist die unbeständige Variable mit maximalem Index.

Da x_f unbeständig ist, gibt es eine Basis $\bar{B}$ mit $x_f \in \bar{B}$, so dass x_f im nächsten Schritt aus der Basis entfernt wird, und eine Basis B', bei der x_f im nächsten Schritt wieder aufgenommen wird:

$$\ldots \longrightarrow \bar{B} \xrightarrow[x_j \text{ kommt in die Basis}]{x_f \text{ verlässt die Basis}} \cdots \rightarrow B' \xrightarrow[x_f \text{ kommt in die Basis}]{} \ldots$$

Sei x_j die Variable, die beim Entfernen von x_f aus $\bar{B}$ in die Basis aufgenommen wird. Da $B^{(p)} = B^{(p+k)}$ wird jede Variable, die in unserer Folge aufgenommen wird, auch irgendwann einmal entfernt, daher ist auch x_j eine unbeständige Variable. Es gilt dann nach (**??**) und (**??**)

$$\bar{z}_j(\bar{B}) < 0, \quad \text{da } x_j \text{ nach } \bar{B} \text{ in die Basis kommt} \tag{4.11}$$
$$w_f(\bar{B}) > 0, \quad \text{da } x_f \text{ die Basis } \bar{B} \text{ verlässt} \tag{4.12}$$
$$\bar{z}_f(B') < 0, \quad \text{da } x_f \text{ nach } B' \text{ in die Basis kommt.} \tag{4.13}$$

Seien $\bar{x}$ und x' die zu den Basen $\bar{B}$ bzw. B' gehörenden Basislösungen. Wir werden zeigen, dass es einen Index $r \in \bar{B}$ mit $r < f$ und $\bar{x}_r = 0$ sowie $w_r(\bar{B}) > 0$ gibt. Nach der Regel von Bland hätte diese Variable wegen $r < f$ an Stelle von x_f aus der Basis $\bar{B}$ entfernt werden müssen. Dies ist dann der gewünschte Widerspruch, der den Beweis beendet.

Nach (**??**) gilt für jede (nicht notwendigerweise zulässige) Lösung von $Ax = b$:

$$c^T x = c^T \bar{x} + \bar{z}(\bar{B})^T x \tag{4.14a}$$
$$c^T x = c^T x' + \bar{z}(B')^T x. \tag{4.14b}$$

Da alle Iterationen in unserer Folge degenerierte Iterationen sind (andernfalls würde sich der Zielfunktionswert in einer Iteration strikt verbessern und Kreiseln wäre ausgeschlossen), folgt $c^T \bar{x} = c^T x' =: Z$. Daher ergibt sich für alle Lösungen von $Ax = b$ aus (**??**) die Gleichung:

$$\bar{z}(\bar{B})^T x = \bar{z}(B')^T x. \tag{4.15}$$

Wir betrachten für $t \in \mathbb{R}$ die spezielle Lösung

$$\begin{aligned} x_j(t) &:= t \\ x_i(t) &:= 0 \text{ für alle } i \in \bar{N} \setminus \{j\} \\ x_{\bar{B}}(t) &:= A_{\cdot \bar{B}}^{-1}(b - A_{\cdot N} x_{\bar{N}}(t)) = \bar{x} - w(\bar{B})t, \end{aligned}$$

von $Ax = b$, die wir bereits bei der Herleitung des Simplex-Verfahrens verwendet hatten. Da $\bar{z}_i(\bar{B}) = 0$ für $i \in \bar{B}$ und

$x_i(t) = 0$ für $i \in \bar{N} \setminus \{j\}$, ist die linke Seite von (**??**) gleich $\bar{z}_j(\bar{B})x_j(t) = \bar{z}_j(\bar{B})t$. Da wiederum $x_i(t) = 0$ für $i \notin \bar{B} \cup \{j\}$, ergibt sich durch Einsetzen in die rechte Seite von (**??**):

$$\begin{aligned}\bar{z}_j(\bar{B})t &= \bar{z}_j(B')x_j(t) + \sum_{i\in\bar{B}} \bar{z}_i(B')x_i(t)\\ &= \bar{z}_j(B')t + \sum_{i\in\bar{B}} \bar{z}_i(B')(\bar{x}_i - w_i(\bar{B})t).\end{aligned}$$

Durch Umordnen der Terme erhalten wir die Gleichung:

$$t\left(\bar{z}_j(\bar{B}) - \bar{z}_j(B') + \sum_{i\in\bar{B}} \bar{z}_i(B')w_i(\bar{B})\right) = \sum_{i\in\bar{B}} \bar{z}_i(B')\bar{x}_i, \tag{4.16}$$

die nach unserer Herleitung für *alle* $t \in \mathbb{R}$ gilt. Da die rechte Seite von (**??**) unabhängig von t ist, folgt, dass der Term in der Klammer auf der linken Seite von (**??**) gleich 0 sein muss, d.h.

$$\bar{z}_j(\bar{B}) - \bar{z}_j(B') = -\sum_{i\in\bar{B}} \bar{z}_i(B')w_i(\bar{B}). \tag{4.17}$$

Nach (**??**) gilt $\bar{z}_j(\bar{B}) < 0$. Da x_f die unbeständige Variable mit maximalem Index ist, haben wir $j < f$. Nun wird nach der Basis B' die Variable x_f in die Basis aufgenommen. Nach der Regel von Bland muss daher $\bar{z}_j(B') \geq 0$ gelten (wäre $\bar{z}_j(B') < 0$, so wäre $x_j \notin B'$ eine Variable mit kleinerem Index, die anstelle von x_f in die Basis käme). Somit ist die linke Seite von (**??**) strikt kleiner als 0. Daher ist auch die rechte Seite von (**??**) strikt kleiner als 0 und es existiert mindestens ein $r \in \bar{B}$ mit $\bar{z}_r(B')w_r(\bar{B}) > 0$.

Insbesondere ist $\bar{z}_r(B') \neq 0$ und $r \notin B'$ (für $r \in B'$ gilt $\bar{z}_r(\bar{B}) = 0$ nach Konstruktion in (**??**)). Wir haben also $r \in \bar{B}$, aber $r \notin B'$, womit auch x_r eine unbeständige Variable ist. Nach Wahl von f als maximaler Index einer unbeständigen Variablen folgt $r \leq f$. Es gilt sogar $r < f$, da nach (**??**) und (**??**) $\bar{z}_f(B') < 0$ und $w_f(\bar{B}) > 0$, d.h. $\bar{z}_f(B')w_f(\bar{B}) < 0$ gilt.

Wir haben gesehen, dass $\bar{z}_r(B') \neq 0$ gilt. Da $r < f$ und die Variable x_r mit $r \notin B'$ nicht in die Basis B' aufgenommen wird (sondern x_f), folgt $\bar{z}_r(B') > 0$. Wegen $\bar{z}_r(B')w_r(\bar{B}) > 0$ haben wir dann $w_r(\bar{B}) > 0$.

Wir haben bereits bemerkt, dass alle Iterationen in der Folge degeneriert sind. Insbesondere ändert sich für

keine der Variablen ihr Wert in der Folge. Da $r \notin B'$ gilt $x'_r = 0$ und somit auch $\bar{x}_r = 0$. Wegen $w_r(\bar{B}) > 0$ und $r < f$ hätten wir also nach der Regel von Bland x_r aus der Basis $\bar{B}$ entfernen müssen und nicht x_f. □

Als Korollar aus unseren Ergebnissen zur Regel von Bland können wir eine zweite Variante des Fundamentalsatzes der Linearen Programmierung beweisen:

Theorem 4.6 (Fundamentalsatz der Linearen Programmierung II). *Für das Lineare Programm in Standardform*

$$\begin{aligned} \min \quad & c^T x && \text{(4.18a)} \\ & Ax = b && \text{(4.18b)} \\ & x \geq 0 && \text{(4.18c)} \end{aligned}$$

mit $\operatorname{Rang} A = m$ *gilt genau eine der folgenden Aussagen:*

*(i) (**??**) hat keine zulässige Lösung.*
*(ii) (**??**) ist unbeschränkt, d.h. es gilt* $\min \{ c^T x : Ax = b, x \geq 0 \} = -\infty$
*(iii) (**??**) hat eine endliche Optimallösung die zugleich auch Basislösung ist.*

Beweis. Offenbar kann nur höchstens eine der drei Aussagen (i)–(iii) für ein Lineares Programm (**??**) gelten. Wir nehmen an, dass (i) und (ii) nicht gelten und zeigen, dass dann (iii) folgt.

Da (i) nicht gilt, hat (**??**) eine zuläsige Lösung. Nach dem Fundamentalsatz (Satz ??) hat das Problem dann aber auch eine zulässige Basis B mit zugehöriger Basislösung $\bar{x}$. Wir wenden die Simplex-Methode mit der Regel von Bland auf (**??**) an, wobei wir mit der Basis B starten. Es terminiert dann nach Satz **??** nach endlich vielen Schritten. Da (ii) nicht gilt, kann das Verfahren nicht mit der Information abbrechen, dass das Problem unbeschränkt ist. Also muss das Verfahren mit einer optimalen Lösung abbrechen, die gleichzeitig eine Basislösung ist, also folgt (iii).

4.2 Die Phase I der Simplex-Methode

Nachdem wir die Frage des Terminierens der Simplex-Methode erfolgreich beantwortet haben, wenden wir uns

nun der Frage zu, wie wir eine erste zulässige Basislösung für ein Lineares Programm in Standardform

$$\begin{aligned} \min \quad & c^T x && (??\text{a}) \\ & Ax = b && (??\text{b}) \\ & x \geq 0 && (??\text{c}) \end{aligned}$$

finden, bzw. feststellen, dass es keine zulässige Lösung gibt.

Wir verzichten in diesem Abschnitt auf die Annahme, dass die Matrix A vollen Zeilenrang hat, d.h. wir lassen nun $\text{Rang}(A) \leq m$ zu. Dafür nehmen wir aber an, dass der Vektor b auf der rechten Seite von (**??**) die Bedingung

$$b \geq 0 \tag{4.20}$$

erfüllt. Dies können wir einfach dadurch erreichen, dass wir eine Zeile $A_{i\cdot}x = b_i$ mit $b_i < 0$ mit -1 multiplizieren: $-A_{i\cdot}x = -b_i$.

Wir betrachten nun das folgende Hilfsproblem, das ebenfalls ein Lineares Programm in Standardform ist:

$$\begin{aligned} \min \quad & \sum_{i=1}^{m} y_i && (4.21\text{a}) \\ & Ax + y = b && (4.21\text{b}) \\ & x, y \geq 0 && (4.21\text{c}) \end{aligned}$$

Sei

$$e = (1, \ldots, 1)^T \in \mathbb{R}^n$$

der Vektor, der aus lauter Einsen besteht. Mit $u = \binom{x}{y}$, $d := \binom{0}{e}$ und $D := (A \quad I)$ schreibt sich (**??**) dann auch als

$$\begin{aligned} \min \quad & d^T u \\ & Du = b \\ & z \geq 0 \end{aligned}$$

Man beachte, dass die Matrix $D = (A \quad I)$ der Nebenbedingungen von (**??**) auf jeden Fall $\text{Rang}\, D = m$ hat, da D die $m \times m$-Einheitsmatrix als Teilmatrix besitzt. Sind insbesondere $\{1, 2, \ldots, n, n+1, \ldots, n+m\}$ die Spaltenindizes von D, dann ist ist $B := (n+1, \ldots, n+m)$ eine Basis von D mit $D_{\cdot B} = I$.

Die Variablen des Vektors y werden als *künstliche Variablen* bezeichnet, die eigentlichen Variablen x unseres

ursprünglichen Problems als *Strukturvariablen*. Die Basis $B = (n+1, \ldots, n+m)$ enthält nur Indizes von künstlichen Variablen. Die zu B gehörende Basislösung ist $\bar{u} = \binom{\bar{u}_B}{\bar{u}_N}$ mit

$$\begin{aligned} \bar{u}_B &:= D_{\cdot B}^{-1} b = Ib = b \geq 0 \\ \bar{u}_N &:= 0. \end{aligned}$$

Wegen unserer Annahme $b \geq 0$ aus (**??**) ist $B = (n+1, \ldots, n+m)$ also eine zulässige Basis für das Lineare Programm (**??**) und wir können das Simplex-Verfahren (Algorithmus 1.1 auf Seite 15) mit der Startbasis B auf (**??**) anwenden.

Wenn wir die Regel von Bland aus dem letzten Abschnitt verwenden, so terminiert die Simplex-Methode nach Satz **??** entwender mit einer optimalen Lösung des linearen Programms (**??**) oder der Information, dass (**??**) unbeschränkt ist. Da für jede zulässige Lösung $u = \binom{x}{y}$ von (**??**) wegen $y \geq 0$

$$d^T u = e^T y = \sum_{i=1}^{m} y_i \geq 0$$

gilt, ist (**??**) nicht unbeschränkt, es verbleibt also nur die Möglichkeit, dass das Simplex-Verfahren eine optimale Lösung findet, die auch Basislösung ist. Sei $u^* = \binom{x^*}{y^*}$ diese Lösung und B^* die zugehörige optimale Basis. Wir unterscheiden folgenden Fälle:

Fall 1: Es gilt $d^T u^* = e^T y^* > 0$

In diesem Fall kann das Ausgangsproblem (**??**) keine zulässige Lösung besitzen. Wäre nämlich x eine solche zulässige Lösung, so wäre $\binom{x}{0}$ eine zulässige Lösung von (**??**) mit Zielfuktionswert 0 im Widerspruch dazu, dass der optimale Wert $d^T u^*$ von (**??**) strikt größer als Null ist.

Fall 1: Es gilt $d^T u^* = e^T y^* = 0$

Diesen Fall untergliedern wir in zwei Unterfälle:

2a: $B^* \cap \{n+1, n+2, \ldots, n+m\} = \emptyset$, d.h. B^* enthält keine künstliche Variable.
In diesem Fall ist $D_{\cdot B^*} = A_{\cdot B^*}$. Insbesondere ist $A_{\cdot B^*}$ nichtsingulär und die zu B^* gehörende Basislösung $\bar{x} = \binom{\bar{x}_{B^*}}{\bar{x}_{N^*}}$ mit $N^* := \{1, \ldots, n\} \setminus B^*$ durch

$$\bar{x}_{B^*} = A^{-1}_{\cdot B^*} b = D^{-1}_{\cdot B^*} b = x^*_{B^*} \geq 0$$
$$\bar{x}_{N^*} = 0$$

zulässig. Also ist B^* eine zulässige Basis für das Ausgangsproblem (**??**) und wir können B^* als Startbasis verwenden, um mit dem Simplex-Verfahren das Problem (**??**) zu lösen.

2b: $B^* \cap \{n+1, n+2, \ldots, n+m\} \neq \emptyset$, d.h. B^* enthält mindestens eine künstliche Variable.
Wir versuchen nun, ähnlich wie beim Simplex-Verfahren die künstlichen Variablen in der Basis durch Strukturvariablen zu ersetzen.
Wir starten mit $B := B^*$ und $N := \{1, \ldots, n+m\} \setminus B$. Seien $B_y := B \cap \{n+1, n+2, \ldots, n+m\}$ die Indizes der künstlichen Variablen in der Basis B. Es gilt $y^* = 0$, insbesondere besitzen alle künstlichen Variablen in der Basis den Wert 0 und die Basis B ist degeneriert. Sei $j \in \{1, \ldots, n\} \cap N$ der Index einer Strukturvariablen, die nicht in der Basis ist und $w = D^{-1}_{\cdot B} A_{\cdot j}$. Falls $w_i \neq 0$ für ein $i \in B_y$, so führen wir eine (leicht modifizierte) Iteration des Simplex-Verfahrens mit x_j als eintretender und y_i als austretender Variable durch.
Da $y^*_i = 0$ und $w_i \neq 0$ gilt für den Wert γ' aus dem modifizierten Ratio-Test (1.25) die Bedingung $\gamma' := 0$. Nach Beobachtung 1.7 ist $B \setminus \{i\} \cup \{j\}$ eine zulässige Basis, wobei sich in dieser degenerierten Iteration die Basislösung (bis auf die Zuordnung der Basis- und Nichtbasisvariablen) nicht ändert. Wir setzen nun $B := B \setminus \{i\} \cup \{j\}$ und $N := N \setminus \{j\} \cup \{i\}$.
Solange wir also für einen Index $j \in \{1, \ldots, n\} \cap N$ und $i \in B_y$ finden, so dass für $w = D^{-1}_{\cdot B} A_{\cdot j}$ die Bedingung $w_i \neq 0$ gilt, können wir wie oben beschrieben eine künstliche Variable gegen eine Strukturvariable in der Basis austauschen.
Entweder können wir damit alle künstlichen Variablen aus der Basis entfernen und landen damit in Fall 2a, oder es gilt

$$(D^{-1}_{\cdot B} A_{\cdot j})_i = 0 \text{ für alle } j \in \{1, 2, \ldots, n\} \cap N \text{ und } i \in B \cap \{n+1, n+2, \ldots, n+m\} \tag{4.22}$$

Wir zeigen nun, dass wir die zu B_y gehörenden Zeilen aus $Ax = b$ streichen können, ohne die Lösungsmenge zu verändern. Dazu bemerken wir zunächst, dass das

System $Ax = b$ offenbar äquivalent zu $Ax + y = b$, $y = 0$, also zu $Du = 0$ und $y = 0$ mit $u = \binom{x}{y}$ ist.

Sei $\bar{u} = \binom{\bar{u}_B}{\bar{u}_N} = \binom{D_{\cdot B}^{-1} b}{0}$ die aktuelle optimale Basislösung von (**??**). Wir definieren

$$\begin{aligned} B_x &:= B \cap \{1, \ldots, n\} & N_x &:= N \cap \{1, \ldots, n\} \\ B_y &:= B \cap \{n+1, \ldots, n+m\} & N_y &:= N \cap \{n+1, \ldots, n+m\}. \end{aligned}$$

Wegen (**??**) ist die Matrix $(D_{\cdot B}^{-1} D_{\cdot N_x})_{B_y \cdot}$ die Nullmatrix:

$$(D_{\cdot B}^{-1} D_{\cdot N_x})_{B_y \cdot} = 0. \tag{4.23}$$

Für eine beliebige Lösung $u = \binom{u_B}{u_N}$ von $Du = b$ gilt dann

$$u_B = D_{\cdot B}^{-1} b - D_{\cdot B}^{-1} D_{\cdot N} u_N = \bar{u}_B - D_{\cdot B}^{-1} D_{\cdot N} u_N. \tag{4.24}$$

Wenn wir u_B in die Strukturvariablen und künstlichen Variablen aufschlüsseln, haben wir

$$\begin{aligned} x_{B_x} &= \bar{u}_{B_x} - \left(\begin{array}{c|c} (D_{\cdot B}^{-1} D_{\cdot N_x})_{B_x \cdot} & (D_{\cdot B}^{-1} D_{\cdot N_y})_{B_x \cdot} \end{array} \right) \begin{pmatrix} x_{N_x} \\ y_{N_y} \end{pmatrix} \\ &= \bar{u}_{B_x} - \left(\begin{array}{c|c} * & * \end{array} \right) \begin{pmatrix} x_{N_x} \\ y_{N_y} \end{pmatrix} \qquad (4.25) \\ y_{B_y} &= \underbrace{\bar{z}_{B_y}}_{=0} - \left(\begin{array}{c|c} \underbrace{(D_{\cdot B}^{-1} D_{\cdot N_x})_{B_y \cdot}}_{0} & (D_{\cdot B}^{-1} D_{\cdot N_y})_{B_y \cdot} \end{array} \right) \begin{pmatrix} x_{N_x} \\ y_{N_y} \end{pmatrix} \\ &= - \left(\begin{array}{c|c} 0 & * \end{array} \right) \begin{pmatrix} x_{N_x} \\ y_{N_y} \end{pmatrix} \qquad (4.26) \end{aligned}$$

Das Gleichungssystem $Du = b$ ist äquivalent zu $u_B = \bar{u}_B - D_{\cdot B}^{-1} D_{\cdot N} u_N$ aus (**??**) und wir haben gezeigt, dass (**??**) (bis auf Permutation der Zeilen) die folgende Form besitzt:

$$x_{B_x} = \bar{u}_{B_x} - \left(\begin{array}{c|c} * & * \end{array} \right) \begin{pmatrix} x_{N_x} \\ y_{N_y} \end{pmatrix} \tag{4.27a}$$

$$y_{B_y} = - \left(\begin{array}{c|c} 0 & * \end{array} \right) \begin{pmatrix} x_{N_x} \\ y_{N_y} \end{pmatrix} \tag{4.27b}$$

Ist also $Du = b$ mit $u = \binom{x}{y} = \binom{x}{0}$, so sind alle künstlichen Variablen gleich 0 und der untere Teil (**??**) des Gleichungssystems unabhängig von der Wahl von x_{N_x} stets erfüllt. Das bedeutet, dass die zu B_y gehörenden Zeilen redundant sind und eliminiert werden können. Damit reduziert sich die Matrix A des Gleichungssytems $Ax = b$ zu $A_{B_x\cdot}$ und $D_{\cdot B_x} = A_{B_x\cdot}$ ist regulär. Mit anderen Worten, B_x ist eine Basis für $A_{B_x\cdot}$ und es gilt darüberhinaus für die zugehörige Basislösung $\bar{x}$ nach (**??**) wegen $x_{N_x} = 0$:

$$\bar{x}_{B_x} = \bar{u}_{B_x} = D^{-1}_{\cdot B_x} b - D^{-1}_{\cdot B_x} D_{\cdot N} \bar{u}_N \geq 0.$$

Daher ist B_x eine zulässige Basis und damit das, was wir erreichen wollten.

4.3 Die Effizienz der Simplexmethode

Aus praktischer Sicht stellt sich natürlich nicht nur die Frage der Endlichkeit, sondern man benötigt auch Aussagen über das Worst-Case Verhalten des Simplex-Algorithmus. Weiterhin stellt sich die Frage, ob es eine polynomiale Auswahlregel für die Schritte zwei und drei gibt.
Bislang ist eine solche Regel nicht bekannt.

Beispiel 4.7. Betrachte für ein ϵ mit $0 < \epsilon < \frac{1}{2}$ das Problem

$$\begin{aligned} &\min -x_n \\ \epsilon \;\leq\; x_1 \;&\leq\; 1 \\ \epsilon x_{j-1} \;\leq\; x_j \;&\leq\; 1 - \epsilon x_{j-1} \qquad \text{für } j = 2, 3, \ldots, n \end{aligned}$$

Altes Kommando `df` hier verwendet

Dieses Beispiel ist in der Literatur unter dem Namen Klee-Minty-Würfel bekannt (benannt nach seinen Entdeckern). Man kann zeigen, dass der Simplex-Algorithmus 2^n Iterationen zur Lösung des obigen linearen Programms benötigt, demnach also alle Ecken abläuft (zum Beweis siehe beispielsweise Papadimitriou & Steiglitz [**?**]).

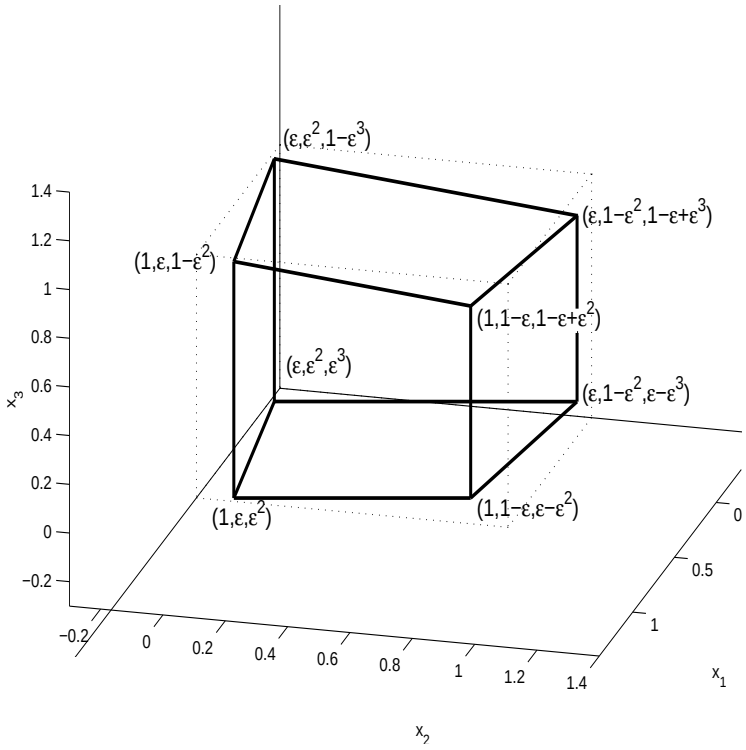


Abb. 4.1. Klee-Minty-Würfel (Beispiel **??**)

4.4 Implementierung des Simplex-Verfahrens

Im folgenden wollen wir noch einige Anmerkungen dazu machen, wie das Simplex-Verfahren in der Praxis implementiert wird.

Lösen der linearen Gleichungssystem

Das Lösen von linearen Gleichungssystemen kann sehr hohe Laufzeiten verursachen. Das Bilden einer Inversen benötigt mindestens $\mathcal{O}(n^{2+\epsilon})$ Schritte (hierbei gilt $\epsilon \in (0, 1]$). Hinzu kommt, dass die Inverse dicht besetzt sein kann (d.h. viele Nicht-Null-Einträge), obwohl die Ausgangsmatrix dünn besetzt ist. Hier ein paar Anmerkungen, wie man diese Probleme vermeiden kann.

Anstelle der Basisinversen A_B^{-1} kann man z.B. eine LU-Faktorisierung von A_B bestimmen, d.h. man erzeugt eine untere Dreiecksmatrix L und eine obere Dreiecksmatrix U, so dass $L \cdot U = A_B$ gilt. Das Gleichungssystem $A_B x = b$ kann dann in zwei Schritten gelöst werden:

$$\begin{aligned} b &= A_B x = L \underbrace{Ux}_{=:y} \\ Ly &= b \\ Ux &= y. \end{aligned}$$

Beachte, dass durch die Dreiecksgestalt der Matrizen L und U *beide* Gleichungssysteme durch einfaches Vorwärts-, bzw. Rückwärtseinsetzen gelöst werden können. Wichtig ist bei der LU-Zerlegung, L und U dünn besetzt zu halten. Dazu gibt es eine Reihe von Pivot-Strategien bei der Bestimmung von L und U. Letztendlich gilt

$$\begin{aligned} L &= L_m P_m L_{m-1} P_{m-1} \cdot \ldots \cdot L_1 P_1, \\ U &= U_m Q_m U_{m-1} Q_{m-1} \cdot \ldots \cdot U_1 Q_1, \end{aligned}$$

wobei die P_i und Q_i Permutationsmatrizen sind und L_i bzw. U_i von der Form

$$L_i = \begin{pmatrix} 1 & & & & \\ & \ddots & & 0 & \\ & & 1 & & \\ & & * & & \\ & & * & 1 & \\ & 0 & \vdots & & \ddots \\ & & * & & 1 \end{pmatrix} \qquad U_i = \begin{pmatrix} 1 & & * & & \\ & \ddots & \vdots & 0 & \\ & & 1 & * & \\ & & * & & \\ & & & 1 & \\ & 0 & & & \ddots \\ & & & & 1 \end{pmatrix}$$

(L_i: i-te Spalte; U_i: i-te Spalte)

sind. Eine sehr gute Implementierung einer LU-Zerlegung für dünn besetzte Matrizen ist z.B. in Suhl & Suhl [**?**] zu finden.

Altes Kommando `df` hier verwendet

Dennoch wäre es immer noch sehr teuer, in jeder Iteration des Simplex-Algorithmus eine neue LU-Faktorisierung zu bestimmen. Es gibt glücklicherweise Update-Formeln. Mit den Bezeichnungen aus (**??**) gilt:

Theorem 4.8. *Sei*

$$\eta_j = \begin{cases} \dfrac{1}{w_i}, & \textit{falls } j = i \\ -\dfrac{w_j}{w_i}, & \textit{sonst.} \end{cases}$$

sowie die Matrizen F und E wie folgt gegeben:

$$F = \begin{pmatrix} 1 & & w_1 & & \\ & \ddots & \vdots & & 0 \\ & & 1 & \vdots & \\ & & & w_i & \\ & & & \vdots & 1 \\ & 0 & & \vdots & \ddots \\ & & & w_m & & 1 \end{pmatrix} \qquad E = \begin{pmatrix} 1 & & \eta_1 & & \\ & \ddots & \vdots & & 0 \\ & & 1 & \vdots & \\ & & & \eta_i & \\ & & & \vdots & 1 \\ & 0 & & \vdots & \ddots \\ & & & \eta_m & & 1 \end{pmatrix}$$

$\uparrow$ i $\qquad\qquad$ $\uparrow$ i

Dann gelten die Beziehungen $A_{B^+} = A_B \cdot F$ und $A_{B^+}^{-1} = E \cdot A_B^{-1}$.

Beweis. Zunächst gilt $A_{B^+} = A_B \cdot F$ (vgl. FTRAN) und durch Nachrechnen verifiziert man $F \cdot E = I$, d.h. $F^{-1} = E$. Damit folgt nun

$$A_{B^+}^{-1} \;=\; (A_B \cdot F)^{-1} \;=\; F^{-1} \cdot A_B^{-1} \;=\; E \cdot A_B^{-1}.$$

Satz **??** kann nun verwendet werden, um Gleichungssysteme in Folgeiterationen zu lösen, ohne explizit eine neue Faktorisierung berechnen zu müssen. Sei B_0 die Basis, bei der zum letzten Mal eine LU-Faktorisierung berechnet wurde, d.h. $L \cdot U = A_{B_0}$ und wir betrachten die k-te Simplexiteration danach und wollen daher ein Gleichungssystem der Form

$$A_{B_k}\, \bar{x}_{B_k} \;=\; b \qquad (*)$$

lösen. Dann gilt

$$A_{B_k} \;=\; A_B \cdot F_1 \cdot F_2 \cdots F_k$$

und

$$\begin{aligned}\bar{x}_k &= F_k^{-1} \cdot F_{k-1}^{-1} \cdots F_1^{-1} x_0 \\ &= E_k \cdot E_{k-1} \cdots E_1 x_0,\end{aligned}$$

wobei x_0 die Lösung von $A_{B_0} x = b$ ist. $\bar{x}_k$ ist die Lösung von $(*)$, denn

$$\begin{aligned}A_{B_k} \bar{x}_k &= (A_{B_0} \cdot F_1 \cdot F_2 \cdots F_k) \cdot (E_k \cdot E_{k-1} \cdots E_1 x_0) \\ &= A_{B_0} \underbrace{(F_1 \cdot F_2 \cdots F_k \cdot E_k \cdot E_{k-1} \cdots E_1)}_{=I} x_0 \\ &= A_{B_0} x_0 = b.\end{aligned}$$

Beachte, dass diese Herleitung eine nochmalige Bestätigung der Korrektheit der Update-Formel im fünften Schritt (Update) des Algorithmus **??** darstellt. Mit der Beziehung $A_{B_k} = A_{B_0} \cdot F_1 \cdot F_2 \cdots F_k$ lassen sich entsprechend Update-Formeln für BTRAN (zur Berechnung von y) und FTRAN (zur Berechnung von w) herleiten. Mehr Informationen zur effizienten Implementierung dieser Berechnungsmöglichkeiten findet man z.B. in [**?**].

Abschließend sei bemerkt, dass die Update-Formeln die Gefahr bergen, dass die numerischen Fehler in den Lösungsvektoren verstärkt werden. Es empfiehlt sich daher, hin und wieder neu zu faktorisieren (aktuelle Codes tun dies etwa alle 100 Iterationen).

Pricing

In der Literatur wird eine Vielzahl von Auswahlregeln diskutiert, welcher Index $j \in N$ im Pricing gewählt werden soll. Die am häufigsten verwendeten Regeln in heutigen State-of-the-art Paketen sind:

Altes Kommando `df` hier verwendet

Altes Kommando `df` hier verwendet

(1) Kleinster Index.
Diese Auswahlregel ist Teil von Blands Regel zum Vermeiden des Kreiselns.

(2) Volles Pricing (Dantzigs Regel).
Berechne die reduzierten Kosten für alle Nichtbasisvariablen und wähle einen der Indizes mit den kleinsten reduzierten Kosten.
- Wird häufig benutzt in der Praxis.
- Kann sehr aufwendig sein, wenn die Anzahl der Variablen groß ist.

Kommando
er verwendet

(3) Partial und Multiple Pricing.
Partial Pricing versucht die Nachteile des vollen Pricings zu vermeiden und betrachtet nur eine Teilmenge der Nichtbasisvariablen. Nur wenn diese Teilmenge keine negativen reduzierten Kosten enthält, wird eine neue Teilmenge betrachtet. Dies wird fortgesetzt, bis alle reduzierten Kosten nicht-negativ sind.

Multiple Pricing nutzt die Tatsache, dass Variablen mit negativen reduzierten Kosten in Folgeiterationen häufig wiederum negative reduzierte Kosten haben. Deshalb werden Variablen, die in früheren Iterationen bereits negative reduzierte Kosten hatten, zuerst betrachtet.

Multiple Partial Pricing kombiniert diese beiden Ideen.

Altes Kommando
`df` hier verwendet

(4) Steepest Edge Pricing (Kuhn & Quandt [**?**], Wolfe & Cutler [**?**]).
Die Idee besteht hierbei darin, eine Variable zu wählen (geometrisch gesehen eine Kante), die am steilsten bzgl. der Zielfunktion ist, die also pro Einheit „Variablenerhöhung“ den größten Fortschritt in der Zielfunktion bewirkt. In Formeln sieht das folgendermaßen aus:
Der Update im fünften Schritt von Algorithmus **??** lautet

$$\begin{pmatrix}\bar{x}_B\\ \bar{x}_N\end{pmatrix} \;=\; \begin{pmatrix}\bar{x}_B\\ \bar{x}_N\end{pmatrix} + \gamma\,\eta_j \qquad \text{mit} \qquad \eta_j \;=\; \begin{pmatrix}-A_B^{-1}A_{\cdot j}\\ e_j\end{pmatrix}.$$

Mit dieser Definition von η gilt für die reduzierten Kosten

$$\bar{z}_j \;=\; c_j \;-\; c_B^T A_B^{-1} A_{\cdot j} \;=\; c^T \eta_j.$$

Im Gegensatz zum vollen Pricing, wo ein $j \in N$ gewählt wird mit

$$c^T \eta_j \;=\; \min_{l \in N} c^T \eta_l,$$

wählen wir nun ein $j \in N$ mit

$$\frac{c^T \eta_j}{||\eta_j||} \;=\; \min_{l \in N} \frac{c^T \eta_l}{||\eta_l||}.$$

Vorteil: Deutlich weniger Simplex-Iterationen nötig.

Nachteil: Rechenaufwendig, es muss in jedem Schritt ein Gleichungssystem zur Berechnung der η_j gelöst werden. Abhilfe schaffen die Update-Formeln von Goldfarb und Reid (siehe [**?**]).
Alternative: Approximation der Normen (siehe [**?**]). Bekannt als Devex Pricing.

Altes Kommando `df` hier verwendet

Ratio-Test

Ein ernsthaftes Problem im Ratio-Test ist es, dass das Minimum γ u.U. für einen Index i angenommen wird, für den der Nenner w_i sehr klein ist. Dies führt zu numerischen Instabilitäten. Eine Idee, um dieses Problem zu lösen, geht auf Harris (siehe [?]) zurück. Die Berechnung erfolgt in mehreren Schritten.

(1) Bestimme $r_k = \begin{cases} \dfrac{\bar{x}_{B_k}}{w_k}, & \text{falls } w_k > 0 \\ +\infty, & \text{sonst.} \end{cases}$ für $k = 1, 2, \ldots, m$.

(2) Berechne

$$t = \min\left\{ r_k + \frac{\epsilon}{w_k} \;\middle|\; k \in N \right\},$$

wobei $\epsilon > 0$ eine vorgegebene Toleranz bezeichnen soll (z.B. $\epsilon = 10^{-6}$).

(3) Der verlassende Index i ist nun

$$i = \operatorname{argmax}\{ w_k \,|\, r_k \leq t\,,\ k = 1, 2, \ldots, m \}.$$

Beachte, dass $\bar{x}_B$ negativ werden kann, da $\epsilon > 0$ gilt, und daraus eine unzulässige Basis resultiert. In diesem Fall wird die untere Schranke Null auf mindestens $-\epsilon$ gesetzt (engl.: shifting) und die Berechnungen der Schritte (1) – (3) wiederholt.

Wie man Variablen mit unteren Schranken, deren Wert ungleich Null ist, behandelt, werden wir im nächsten Abschnitt sehen.

Die unzulässigen Schranken werden erst am Ende des Algorithmus entfernt. Dies geschieht durch einen Aufruf der Phase I mit einem anschliessenden erneuten Durchlauf der Phase II des Simplex-Algorithmus. Die Hoffnung dabei ist, dass am Ende nur wenige Variablen echt kleiner als Null sind und damit die beiden Phasen schnell ablaufen. Mehr Details dazu findet man in Gill, Murray, Saunders & Wright [**?**].

4.5 Varianten des Simplex-Algorithmus

In diesem Abschnitt wollen wir uns noch mit zwei Varianten / Erweiterungen des Algorithmus **??** beschäftigen. Zum einen ist dies die Behandlung von unteren und oberen Schranken, zum anderen ist es der duale Simplex-Algorithmus. Wir beginnen mit letzterem.

4.5.1 Der duale Simplex-Algorithmus

Die Grundidee des dualen Simplex-Algorithmus ist es, den primalen Simplex-Algorithmus **??** auf das duale lineare Programm in Standardform (**??**) anzuwenden, ohne (**??**) explizit zu dualisieren. Historisch war die Entwicklung jedoch anders: Man dachte zunächst, man hätte ein neues Verfahren gefunden, ehe man den obigen Zusammenhang erkannte. Betrachten wir nochmals (**??**)

$$\begin{aligned} \min c^T x & \\ Ax &= b \\ x &\geq 0 \end{aligned}$$

und das dazu duale Programm in Standardform (**??**)

$$\begin{aligned} \max b^T y & \\ A^T y + Iz &= c \\ z &\geq 0. \end{aligned}$$

Wir gehen im Folgenden wieder davon aus, dass A vollen Zeilenrang hat, d.h. $\text{Rang}(A) = \text{Rang}(A^T) = m \leq n$. Mit $D = (A^T, I) \in \mathbb{K}^{n \times (m+n)}$ erhält (**??**) die Form

$$\begin{aligned} \max b^T y & \\ D\begin{pmatrix} y \\ z \end{pmatrix} &= c \\ z &\geq 0. \end{aligned}$$

Eine Basis H des dualen Programms in (**??**) hat also die Mächtigkeit n mit $H \subseteq \{1, 2, \ldots, m, m+1, \ldots, m+n\}$. Wir machen zunächst folgende Beobachtung:

Theorem 4.9. *Sei $A \in \mathbb{K}^{m \times n}$ mit vollem Zeilenrang und H eine zulässige Basis von (**??**). Dann ist (**??**) unbeschränkt oder es existiert eine optimale Basis H_{opt} mit $\{1, 2, \ldots, m\} \subseteq H_{opt}$.*

Beweis. Übung.

Altes Kommando `df` hier verwendet

Im Gegensatz zur Anwendung von Algorithmus **??** auf (**??**) haben wir hier die Besonderheit, dass nicht alle Variablen Vorzeichenbeschränkungen unterliegen. Die Variablen in y sind sogenannte freie Variablen. Satz **??** sagt aus, dass wir o.B.d.A. davon ausgehen können, dass alle Variablen in y in einer Basis H für (**??**) enthalten sind.

Da $|H| = n$ und y aus m Variablen besteht, muss H noch $(n-m)$ Variablen aus z enthalten. Wir bezeichnen mit $N \subseteq \{1, 2, \ldots, n\}$ diese $(n-m)$ Variablen und mit $B = \{1, 2, \ldots, n\} \setminus N$ die Indizes aus z, die nicht in der Basis sind. Da D_H regulär ist, muss also auch $(A_B)^T$ regulär sein, also ist A_B regulär. Zur Vereinfachung der Notation schreiben wir im Folgenden A_B^T anstelle von $(A_B)^T$ und für $(A_N)^T$ entsprechend $A_N f^T$. Nun gilt

$$
\begin{aligned}
D\begin{pmatrix} y \\ z \end{pmatrix} = c,\ z \geq 0 &\iff A^T y + Iz = c,\ z \geq 0 \\
&\iff \left\{ \begin{array}{l} A_B^T y + z_B = c_B\ ,\ z_B \geq 0 \\ A_N^T y + z_N = c_N\ ,\ z_N \geq 0 \end{array} \right\} \qquad (4.28) \\
&\iff \left\{ \begin{array}{r} y = A_B^{-T}(c_B - z_B) \\ z_N = c_N - A_N^T A_B^{-T}(c_B - z_B) \\ z_N,\, z_B \geq 0 \end{array} \right\}
\end{aligned}
$$

Setzen wir die Nichtbasisvariablen aus $z_B = 0$, so ist H (primal) zulässig für (**??**), falls

$$z_N = c_N - A_N^T A_B^{-T} c_B \geq 0,$$

was äquivalent dazu ist, dass B dual zulässig ist (vgl. Definition **??**(a)).

Man beachte weiterhin, dass H durch N und B eindeutig festgelegt ist. Daher ist es üblich, der Definition **??**(a) zu folgen und von einer dual zulässigen Basis B für (**??**) zu sprechen und weniger von einer (primal) zulässigen Basis H für (**??**).

Betrachten wir also eine dual zulässige Basis B (Beachte: B sind die Nichtbasisvariablen im Dualen und N sind die Basisvariablen im Dualen) mit

$$\begin{aligned} \bar{z}_N &= c_N - A_N^T A_B^{-T} c_B \geq 0\,, \\ \bar{z}_B &= 0 \end{aligned}$$

und wenden Algorithmus **??** auf (**??**) an.

(1) BTRAN liefert

$$\begin{aligned} \bar{x}^T D_H = \begin{pmatrix} b^T \\ 0 \end{pmatrix} \quad &\Longleftrightarrow \quad \bar{x}^T \begin{pmatrix} A_B^T & 0 \\ A_N^T & I_N \end{pmatrix} = \begin{pmatrix} b^T \\ 0 \end{pmatrix} \\ &\Longleftrightarrow \quad \begin{aligned} \bar{x}_B^T A_B^T + \bar{x}_N^T A_N^T &= b^T \\ \bar{x}_N^T &= 0 \end{aligned} \\ &\Longleftrightarrow \quad \bar{x}_N = 0\,,\ A_B \bar{x}_B = b \\ &\Longleftrightarrow \quad \bar{x}_N = 0\,,\ \bar{x}_B = A_B^{-1} b \end{aligned}$$

(2) Pricing berechnet die reduzierten Kosten (im Primalen $c_N - A_N^T \bar{y}$) zu

$$0_B - I_B \bar{x}_B = -\bar{x}_B$$

$\Big($ vgl. auch

$$b^T y \overset{(??)}{=} b^T (A_B^{-T}(c_B - z_B)) = b^T A_B^{-T} c_B - \underbrace{(A_B^{-1} b)^T}_{\geq 0 \Rightarrow \text{ optimal}} z_B \Big).$$

Da (**??**) ein Maximierungsproblem ist, können wir uns nur verbessern, falls $c_j - (A^T)_{j\cdot} y > 0$ gilt für ein $j \in N$. Gilt also $-\bar{x}_B \leq 0 \Leftrightarrow \bar{x}_B \geq 0$, so ist H (bzw. B) dual optimal (d.h. B ist primal zulässig). Andernfalls wähle einen Index B_j mit $\bar{x}_{B_j} < 0$.

(3) FTRAN berechnet

$$\begin{aligned} D_H \begin{pmatrix} w \\ \alpha_N \end{pmatrix} = \begin{pmatrix} e_j \\ 0 \end{pmatrix} \quad &\Longleftrightarrow \quad \begin{pmatrix} A_B^T & 0 \\ A_N^T & I_N \end{pmatrix} \begin{pmatrix} w \\ \alpha_N \end{pmatrix} = \begin{pmatrix} e_j \\ 0 \end{pmatrix} \\ &\Longleftrightarrow \quad A_B^T w = e_j \quad \text{und} \quad A_N^T w + \alpha_N = 0 \\ &\Longleftrightarrow \quad A_B^T w = e_j \quad \text{und} \quad \alpha_N = -A_N^T w. \end{aligned}$$

(4) Ratio-Test
Wir überprüfen, ob $\alpha_N \leq 0$ ist (Beachte, dass das Vorzeichen von w egal ist, da die Variablen in y freie Variablen sind). Ist dies der Fall, so ist (**??**) unbeschränkt, d.h. $P^=(A, b) = \emptyset$. Andernfalls setze

$$\gamma \;=\; \frac{\bar z_j}{\alpha_j} \;=\; \min\left\{ \frac{\bar z_k}{\alpha_k} \;\middle|\; \alpha_k > 0\,,\; k \in N \right\}$$

mit $j \in N\,,\ \alpha_j > 0$. Damit verlässt nun j die duale Basis H bzw. tritt in die Basis B ein.

Dies zusammen liefert

Der duale Simplex-Algorithmus

Input: Dual zulässige Basis $B\,,\ \bar z_N = c_N - A_N^T A_B^{-T} c_B \geq 0$.

Output:

(i) Eine Optimallösung $\bar x$ für (**??**) bzw.
eine Optimallösung $\bar y = A_B^{-T} c_B$ für (**??**) bzw.
eine Optimallösung $\bar y, \bar z_N, \bar z_B = 0$ für (**??**) oder

(ii) Die Meldung $P^=(A,b) = \emptyset$ bzw. (**??**) ist unbeschränkt.

Altes Kommando `df` hier verwendet

(1) BTRAN
Löse $A_B\,\bar x_B \;=\; b$.

Altes Kommando `df` hier verwendet

(2) Pricing
Falls $\bar x_B \geq 0$, so ist (**??**) bzw. (**??**) optimal, **Stop**.
Andernfalls wähle einen Index $i \in \{1, 2, \ldots, m\}$ mit $x_{B_i} < 0$, d.h. B_i verlässt die Basis B.

Altes Kommando `df` hier verwendet

(3) FTRAN
Löse $A_B^T\, w \;=\; e_i$ und berechne $\alpha_N \;=\; -A_N^T\, w$.

Altes Kommando `df` hier verwendet

(4) Ratio-Test
Falls $\alpha_N \leq 0$, so ist (**??**) unbeschränkt bzw. $P^=(A,b) = \emptyset$, **Stop**.
Andernfalls berechne

$$\gamma \;=\; \frac{\bar z_j}{\alpha_j} \;=\; \min\left\{ \frac{\bar z_k}{\alpha_k} \;\middle|\; \alpha_k > 0\,,\; k \in N \right\}$$

mit $j \in N\,,\ \alpha_j > 0$. Die Variable j tritt dann in die Basis ein.

Altes Kommando `df` hier verwendet

(5) Update

Setze $\quad \bar z_N \;=\; \bar z_N \,-\, \gamma\alpha_N\,, \quad \bar z_{B_i} \;=\; \gamma$
$\qquad\quad N \;=\; (N \setminus \{j\}) \cup B_i\,, \quad B_i \;=\; j.$
Gehe zu (1).

Die Korrektheit des Algorithmus **??** und alle weiteren Konsequenzen gelten dem Abschnitt **??** (bzw. Abschnitt **??**) entsprechend. Dabei wird die Tatsache genutzt, dass Algorithmus **??** nichts anderes ist als die Anwendung von Algorithmus **??** auf (**??**).

Beispiel 4.10. Betrachte

$$\begin{aligned} \min\ & 2x_1 + x_2 \\ & -x_1 - x_2 \leq -\tfrac{1}{2} \\ & -4x_1 - x_2 \leq -1 \\ & x_1\,,\ x_2 \geq 0. \end{aligned}$$

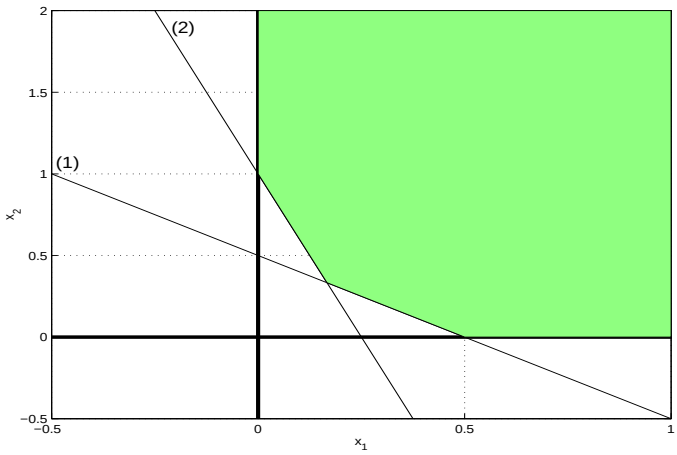


Abb. 4.2. Grafische Darstellung zu Beispiel **??**.

In Standardform lautet das LP

$$\begin{aligned} \min\ & 2x_1 + x_2 \\ & -x_1 - x_2 + x_3 = -\tfrac{1}{2} \\ & -4x_1 - x_2 + x_4 = -1 \\ & x_1\,,\ x_2\,,\ x_3\,,\ x_4 \geq 0. \end{aligned}$$

Wir starten mit $B = (3, 4)\,,\ N = \{1, 2\}$. Es gilt

$$\bar{z}_N = \begin{pmatrix}2\\1\end{pmatrix} - A_N^T A_B^{-T} \begin{pmatrix}0\\0\end{pmatrix} = \begin{pmatrix}2\\1\end{pmatrix} \geq 0.$$

Damit ist B dual zulässig.

(1.1) BTRAN

$$\begin{pmatrix}1 & 0\\ 0 & 1\end{pmatrix} \bar{x}_B = \begin{pmatrix}-\frac{1}{2}\\ -1\end{pmatrix}$$

$$\Longrightarrow \quad \bar{x}_B = \begin{pmatrix}\bar{x}_3\\ \bar{x}_4\end{pmatrix} = \begin{pmatrix}-\frac{1}{2}\\ -1\end{pmatrix}, \qquad \bar{x}_N = \begin{pmatrix}\bar{x}_1\\ \bar{x}_2\end{pmatrix} = \begin{pmatrix}0\\0\end{pmatrix}.$$

(1.2) Pricing

$\bar{x} \ngeq 0$. Wähle $i = 2$ mit $\bar{x}_{B_2} = \bar{x}_4 = -1 < 0$.

(1.3) FTRAN

$$A_B^T w = e_2 \qquad \Longleftrightarrow \qquad \begin{pmatrix}1 & 0\\ 0 & 1\end{pmatrix} w = \begin{pmatrix}0\\1\end{pmatrix} \qquad \Longrightarrow \qquad w = \begin{pmatrix}0\\1\end{pmatrix}$$

und

$$\alpha_N = -A_N^T w = \begin{pmatrix}1 & 4\\ 1 & 1\end{pmatrix} \begin{pmatrix}0\\1\end{pmatrix} = \begin{pmatrix}4\\1\end{pmatrix} = \begin{pmatrix}\alpha_1\\ \alpha_2\end{pmatrix}.$$

(1.4) Ratio-Test

$$\gamma = \min\left\{\frac{2}{4}, \frac{1}{1}\right\} = \frac{1}{2} \qquad \text{mit } j = 1.$$

(1.5) Update

$$\begin{pmatrix}\bar{z}_1\\ \bar{z}_2\end{pmatrix} = \bar{z}_N = \begin{pmatrix}2\\1\end{pmatrix} - \frac{1}{2}\begin{pmatrix}4\\1\end{pmatrix} = \begin{pmatrix}0\\ \frac{1}{2}\end{pmatrix}, \qquad \bar{z}_4 = \frac{1}{2}$$

$$N = \{2, 4\}, \qquad B = (3, 1).$$

(2.1) BTRAN

$$\begin{pmatrix}1 & -1\\ 0 & -4\end{pmatrix} \bar{x}_B = \begin{pmatrix}-1/2\\ -1\end{pmatrix}$$

$$\Longrightarrow \quad \bar{x}_B = \begin{pmatrix}\bar{x}_3\\ \bar{x}_1\end{pmatrix} = \begin{pmatrix}-1/4\\ 1/4\end{pmatrix}, \qquad \bar{x}_N = \begin{pmatrix}\bar{x}_2\\ \bar{x}_4\end{pmatrix} = \begin{pmatrix}0\\0\end{pmatrix}.$$

(2.2) Pricing

$$\bar{x}_B \ngeq 0. \quad \text{Wähle } i = 1 \text{ mit } \bar{x}_{B_1} = \bar{x}_3 = -\frac{1}{4} < 0.$$

(2.3) FTRAN

$$A_B^T w = e_1 \Leftrightarrow \begin{pmatrix} 1 & 0 \\ -1 & -4 \end{pmatrix} w = \begin{pmatrix} 1 \\ 0 \end{pmatrix} \qquad \Longrightarrow \qquad w = \begin{pmatrix} 1 \\ -1/4 \end{pmatrix}$$

und

$$\alpha_N = -A_N^T w = \begin{pmatrix} 1 & 1 \\ 0 & -1 \end{pmatrix} \begin{pmatrix} 1 \\ -1/4 \end{pmatrix} = \begin{pmatrix} 3/4 \\ 1/4 \end{pmatrix}.$$

(2.4) Ratio-Test

$$\gamma = \min \left\{ \frac{1/2}{3/4}, \frac{1/2}{1/4} \right\} = \frac{2}{3} \qquad \text{mit } j = 2.$$

(2.5) Update

$$\bar{z}_N = \begin{pmatrix} 1/2 \\ 1/2 \end{pmatrix} - \frac{2}{3} \begin{pmatrix} 3/4 \\ 1/4 \end{pmatrix} = \begin{pmatrix} 0 \\ 1/3 \end{pmatrix}, \qquad \bar{z}_{B_1} = \bar{z}_3 = \frac{2}{3},$$

$$N = \{3, 4\}, \qquad B = (2, 1).$$

(3.1) BTRAN

$$\begin{pmatrix} -1 & -1 \\ -1 & -4 \end{pmatrix} \bar{x}_B = \begin{pmatrix} -1/2 \\ -1 \end{pmatrix}$$

$$\Longrightarrow \quad \bar{x}_B = \begin{pmatrix} \bar{x}_2 \\ \bar{x}_1 \end{pmatrix} = \begin{pmatrix} 1/3 \\ 1/6 \end{pmatrix}, \qquad \bar{x}_N = \begin{pmatrix} \bar{x}_3 \\ \bar{x}_4 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}.$$

(3.2) Pricing

$$\bar{x}_B \geq 0 \quad \Longrightarrow \quad \bar{x}_1 = \frac{1}{3}, \; \bar{x}_2 = \frac{1}{6} \text{ ist optimal.}$$

4.5.2 Obere und untere Schranken

Häufig haben Variablen in linearen Programmen obere und untere Schranken, z.B. Relaxierungen von ganzzahligen oder 0-1 Programmen. D.h. wir haben ein Problem der Form

$$\begin{aligned} \min \; & c^T x \\ & Ax = b \\ & l \leq x \leq u, \end{aligned} \tag{4.29}$$

wobei $l \in \left(\mathbb{K} \cup \{-\infty\}\right)^n$ und $u \in \left(\mathbb{K} \cup \{+\infty\}\right)^n$ ist. Gilt für ein $i \in \{1, 2, \ldots, n\}$ $l_i = -\infty$ und $u_i = +\infty$, so heißt

Altes Komma
df hier verwen

die zugehörige Variable x_i freie Variable. Freie Variablen werden, insofern sie nicht zur Basis gehören, auf den Wert Null gesetzt. Sobald sie einmal in die Basis eintreten, werden sie diese nie wieder verlassen, vgl. auch Satz **??**.

Betrachten wir also den Fall, dass $l_i \neq -\infty$ oder $u_i \neq +\infty$ gilt für alle $i \in \{1, 2, \ldots, n\}$. Dann kann man durch eine Variablensubstitution (**??**) immer in die Form

$$\begin{aligned} \min\ & c^T x \\ & Ax = b \\ & 0 \leq x \leq u \end{aligned} \tag{4.30}$$

bringen mit $u \in \left(\mathbb{K}_+ \cup \{\infty\}\right)^n$.

Eine Möglichkeit (**??**) zu lösen besteht darin, die oberen Schranken explizit als Ungleichungen in die Nebenbedingungsmatrix aufzunehmen. Dies würde das System jedoch von der Größe $(m \times n)$ auf $(m+n) \times (2n)$ erweitern. Das ist sehr unpraktikabel, denn jede Basismatrix hat somit die Größe $(m+n) \times (m+n)$ anstatt $(m \times m)$ zuvor. Im Folgenden werden wir sehen, wie die oberen Schranken implizit im Algorithmus **??** berücksichtigt werden können, ohne die Größe des Problems und damit die Komplexität des Algorithmus zu erhöhen.

Wir unterteilen die Menge der Nichtbasisvariablen in zwei Mengen N_l und N_u. In N_l sind alle Nichtbasisvariablen, die in der gerade durchgeführten Iteration den Wert der unteren Schranke annehmen, d.h.

$$N_l = \{j \in N \mid \bar{x}_j = 0\}$$

und analog dazu sind in der Menge

$$N_u = \{j \in N \mid \bar{x}_j = u_j\}$$

alle Nichtbasisvariablen, die den Wert der oberen Schranke haben. Ist B eine Basis für (**??**), so wissen wir aus (1.14), dass

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

gilt. Setzen wir $x_j = u_j$ für $j \in N_u$ und $x_j = 0$ für $j \in N_l$, dann nennt man B primal zulässig, falls

$$u_B \geq x_B = A_B^{-1} b - A_B^{-1} A_{N_u} u_{N_u} \geq 0. \tag{4.31}$$

Für die Zielfunktion gilt mit (1.15)

$$\begin{aligned} c^T x &= c_B^T A_B^{-1} b + (c_N^T - c_B^T A_B^{-1} A_N) x_N \\ &= c_B^T A_B^{-1} b + (c_{N_l}^T - c_B^T A_B^{-1} A_{N_l}) \cdot 0 + (c_{N_u}^T - c_B^T A_B^{-1} A_{N_u}) \cdot u_{N_u}. \end{aligned}$$

Eine Verbesserung der Zielfunktion kann also nur erreicht werden, falls

$$\begin{aligned} c_j - c_B^T A_B^{-1} A_{\cdot j} &< 0 \qquad \text{für } j \in N_l \text{ oder} \\ c_j - c_B^T A_B^{-1} A_{\cdot j} &> 0 \qquad \text{für } j \in N_u \end{aligned} \tag{4.32}$$

gilt. In anderen Worten, B nennt man dual zulässig, falls (**??**) nicht gilt, d.h. falls

$$\begin{aligned} c_j - c_B^T A_B^{-1} A_{\cdot j} &\geq 0 \qquad \text{für } j \in N_l \text{ und} \\ c_j - c_B^T A_B^{-1} A_{\cdot j} &\leq 0 \qquad \text{für } j \in N_u. \end{aligned}$$

Entsprechend drehen sich die Vorzeichen im Ratio-Test um, wobei zusätzlich beachtet werden muss, dass die ausgewählte Variable j nicht um mehr als u_j erhöht bzw. erniedrigt werden kann. Gilt $j \in N_l$, so wirkt sich eine Erhöhung von x_j um den Wert γ auf die Basis wie folgt aus, vgl. (1.14):

$$\bar{x}_B \leftarrow \bar{x}_B - \gamma w,$$

d.h. γ muss so gewählt werden, dass

$$0 \leq \bar{x}_B - \gamma w \leq u_B$$

gilt. Damit ergibt sich für γ die Darstellung

$$\gamma = \min \left\{ u_j, \min \left\{ \frac{\bar{x}_{B_i}}{w_i} \,\middle|\, w_i > 0\,,\ i \in \{1, 2, \ldots, m\} \right\}, \right.$$
$$\left. \min \left\{ \frac{u_{B_i} - \bar{x}_{B_i}}{-w_i} \,\middle|\, w_i < 0\,,\ i \in \{1, 2, \ldots, m\} \right\} \right\}.$$

Gilt $j \in N_u$, so wirkt sich eine Verringerung von x_j um γ auf die Basis wie folgt aus, vgl. (**??**):

$$\begin{aligned} \bar{x}_B &\leftarrow A_B^{-1} b - A_B^{-1} A_j (u_j - \gamma) \\ &= (A_B^{-1} b - A_B^{-1} A_j u_j) + A_B^{-1} A_j \gamma \\ &= \bar{x}_B + \gamma w, \end{aligned}$$

d.h. γ muss so gewählt werden, dass

$$0 \leq \bar{x}_B + \gamma w \leq u_B$$

gilt. Damit ergibt sich für γ

$$\gamma = \min\left\{u_j, \min\left\{\frac{\bar{x}_{B_i}}{-w_i}\,\middle|\, w_i < 0\,,\ i \in \{1,2,\ldots,m\}\right\}\right.,$$
$$\left.\min\left\{\frac{u_{B_i} - \bar{x}_{B_i}}{w_i}\,\middle|\, w_i > 0\,,\ i \in \{1,2,\ldots,m\}\right\}\right\}.$$

Damit können wir den Simplex-Algorithmus mit allgemeinen oberen und unteren Schranken wie folgt angeben:

Simplex-Algorithmus mit oberen Schranken

Input:

- Ein lineares Programm der Form

$$\begin{aligned} \min\ & c^T x \\ & Ax = b \\ & 0 \le x \le u\,, \qquad u \in (\mathbb{K} \cup \{\infty\})^n. \end{aligned}$$

- Eine primal zulässige Basis B und Mengen N_l, N_u mit $\bar{x}_B = A_B^{-1}b - A_B^{-1}A_{N_u}u_{N_u}$, $\bar{x}_{N_l} = 0$.

Output:

- Eine Optimallösung $\bar{x}$ für (**??**)
  oder
- Die Meldung (**??**) ist unbeschränkt.

(1) BTRAN
Löse $\bar{y}^T A_B = c_B^T$.

(2) Pricing
Berechne $\bar{z}_N = c_N - A_N^T \bar{y}$.
Falls $\bar{z}_{N_l} \ge 0$ und $\bar{z}_{N_u} \le 0$, ist B optimal, **Stop**.
Andernfalls wähle $j \in N_l$ mit $\bar{z}_j < 0$ oder $j \in N_u$ mit $\bar{z}_j > 0$.

(3) FTRAN
Löse $A_B\, w = A_{\cdot j}$.

(4) Ratio-Test
I.Fall: $j \in N_l$.
Bestimme


Altes Kommando `df` hier verwendet


Altes Kommando `df` hier verwendet


Altes Kommando `df` hier verwendet

$$\begin{aligned}
\gamma_{fl} &= u_j, \\
\gamma_l &= \min\left\{ \frac{\bar{x}_{B_i}}{w_i} \;\middle|\; w_i > 0\,,\ i \in \{1,2,\ldots,m\} \right\}, \\
\gamma_u &= \min\left\{ \frac{u_{B_i} - \bar{x}_{B_i}}{-w_i} \;\middle|\; w_i < 0\,,\ i \in \{1,2,\ldots,m\} \right\}, \\
\gamma &= \min\{\gamma_{fl}\,,\ \gamma_l\,,\ \gamma_u\}\,.
\end{aligned}$$

Gilt $\gamma = \infty$, dann ist (**??**) unbeschränkt,
Stop.
Gilt $\gamma = \gamma_{fl}$, dann gehe zu (5).
Andernfalls sei $i \in \{1,2,\ldots,m\}$ der Index, der γ annimmt.

II.Fall: $j \in N_u$.
Bestimme

$$\begin{aligned}
\gamma_{fl} &= u_j, \\
\gamma_l &= \min\left\{ \frac{\bar{x}_{B_i}}{-w_i} \;\middle|\; w_i < 0\,,\ i \in \{1,2,\ldots,m\} \right\}, \\
\gamma_u &= \min\left\{ \frac{u_{B_i} - \bar{x}_{B_i}}{w_i} \;\middle|\; w_i > 0\,,\ i \in \{1,2,\ldots,m\} \right\}, \\
\gamma &= \min\{\gamma_{fl}\,,\ \gamma_l\,,\ \gamma_u\}\,.
\end{aligned}$$

Gilt $\gamma = \infty$, dann ist (**??**) unbeschränkt,
Stop.
Gilt $\gamma = \gamma_{fl}$, dann gehe zu (5).
Andernfalls sei $i \in \{1,2,\ldots,m\}$ der Index, der γ annimmt.

Altes Kommando
`df` hier verwendet

(5) Update

I.Fall: $j \in N_l$.

$$\begin{aligned}
\bar{x}_B &= \bar{x}_B - \gamma\, w \\
N_l &= N_l \setminus \{j\}
\end{aligned}$$

Falls $\gamma = \gamma_{fl} \rightarrow N_u = N_u \cup \{j\}\,,\ \bar{x}_j = u_j,$ gehe zu (2)
Falls $\gamma = \gamma_l \rightarrow N_l = N_l \cup \{B_i\}\,,\ B_i = j,\ \bar{x}_j = \gamma,$ gehe zu (1)
Falls $\gamma = \gamma_u \rightarrow N_u = N_u \cup \{B_i\}\,,\ B_i = j\,,\ \bar{x}_j = \gamma,$ gehe zu (1)

II.Fall: $j \in N_u$.

$$\begin{aligned}
\bar{x}_B &= \bar{x}_B + \gamma\, w \\
N_u &= N_u \setminus \{j\}
\end{aligned}$$

Falls $\gamma = \gamma_{fl} \rightarrow N_l = N_l \cup \{j\},\ \bar{x}_j = 0,$ gehe zu (2)
Falls $\gamma = \gamma_l \rightarrow N_l = N_l \cup \{B_i\},\ B_i = j,\ \bar{x}_j = u_j - \gamma,$ gehe zu (1)
Falls $\gamma = \gamma_u \rightarrow N_u = N_u \cup \{B_i\},\ B_i = j,\ \bar{x}_j = u_j - \gamma,$ gehe zu (1)

4.6 Sensitivitätsanalyse

Wir sind bisher davon ausgegangen, dass die Daten A, b und c eines linearen Programms der Form (**??**) fest vorgegeben waren. Häufig ist es jedoch der Fall, dass sich diese Daten im Laufe der Zeit ändern, wenn Bedingungen und/oder Variablen hinzukommen und damit nachoptimiert werden muss. Gehen wir einmal davon aus, dass wir ein lineares Programm der Form (**??**)

$$\begin{aligned} \min\ & c^T x \\ Ax &= b \\ x &\geq 0 \end{aligned}$$

bereits gelöst haben und eine optimale Basis B mit der Lösung $\bar{x}_B = A_B^{-1} b$ und $\bar{x}_N = 0$ kennen. Wir wollen nun folgende Änderungen des linearen Programms betrachten und uns überlegen, wie wir ausgehend von B möglichst schnell eine Optimallösung des veränderten Programms finden können:

(i) Änderung der Zielfunktion c.
(ii) Änderung der rechten Seite b.
(iii) Änderung eines Eintrags in der Matrix A.
(iv) Hinzufügen einer neuen Spalte.
(v) Hinzufügen einer neuen Zeile.

Die Änderungen (i) bis (iv) und deren Auswirkungen werden dem Leser als Übung empfohlen, hier wollen wir exemplarisch den Fall (v) betrachten. Dieser Fall wird insbesondere bei der Lösung ganzzahliger Optimierungsaufgaben von Bedeutung sein (siehe Diskrete Optimierung II).

(v) Hinzufügen einer neuen Zeile

Wir fügen dem gegebenen Problem

$$\begin{aligned} \min\ & c^T x \\ Ax &= b \\ x &\geq 0 \end{aligned}$$

eine neue Nebenbedingung

$$A_{m+1,\cdot}\, x = b_{m+1}$$

hinzu. Wir unterscheiden nun folgende Fälle:

1.Fall: Die Optimallösung $\bar{x}$ erfüllt auch die neue Nebenbedingung. Dann ist $\bar{x}$ auch für das erweiterte Problem eine Optimallösung. Ist $A_{m+1,\cdot}$ linear abhängig von den Zeilen von A, so ist die neue Zeile irrelevant und kann gelöscht werden.
Ist $A_{m+1,\cdot}$ linear unabhängig von den Zeilen von A, so ist $\bar{x}$ eine entartete Basislösung des erweiterten Systems. Eine der Nichtbasisvariablen aus dem Träger von $A_{m+1,\cdot}$ wird mit dem Wert Null in die Basis aufgenommen. Die neue Basis hat nun die Kardinalität $m+1$.

2.Fall: Im Allgemeinen jedoch wird die neue Nebenbedingung nicht erfüllt sein. Wir wollen hier annehmen, dass $A_{m+1,\cdot}\,\bar{x} > b_{m+1}$ gilt. Der zweite Fall wird analog behandelt. Wir führen eine neue Schlupfvariable $x_{m+1} \geq 0$ ein mit

$$A_{m+1,\cdot}\,x \;+\; x_{m+1} \;=\; b_{m+1}.$$

und erweitern die Basis um x_{m+1}. Die neue Basismatrix zur Basis B' hat die Form

$$\begin{pmatrix} A_B & 0 \\ A_{m+1,B} & 1 \end{pmatrix} \;\in\; \mathbb{K}^{(m+1)\times(m+1)}.$$

Die zugehörige inverse Basismatrix lautet dann

$$\begin{pmatrix} A_B^{-1} & 0 \\ -A_{m+1,B}\cdot A_B^{-1} & 1 \end{pmatrix} \;\in\; \mathbb{K}^{(m+1)\times(m+1)},$$

und es gilt

$$\begin{aligned} \bar{x}_{B'} \;&=\; \begin{pmatrix} A_B^{-1} & 0 \\ -A_{m+1,B}\cdot A_B^{-1} & 1 \end{pmatrix} \begin{pmatrix} b \\ b_{m+1} \end{pmatrix} \;=\; \begin{pmatrix} A_B^{-1}b \\ -A_{m+1,B}\cdot A_B^{-1}b + b_{m+1} \end{pmatrix} \\ &=\; \begin{pmatrix} \bar{x}_B \\ -A_{m+1,B}\,\bar{x}_B + b_{m+1} \end{pmatrix}, \end{aligned}$$

wobei $\bar{x}_B \geq 0$ und $-A_{m+1,B}\,\bar{x}_B + b_{m+1} < 0$ ist, d.h. B' ist nicht primal zulässig, jedoch ist sie dual zulässig, da

$$\begin{aligned} c_N - c_{B'}^T A_{B'}^{-1} \begin{pmatrix} A_N \\ A_{m+1,N} \end{pmatrix} &= c_N - (c_B^T,\,0) \begin{pmatrix} A_B^{-1} & 0 \\ -A_{m+1,B}\cdot A_B^{-1} & 1 \end{pmatrix} \begin{pmatrix} A_N \\ A_{m+1,N} \end{pmatrix} \\ &= c_N - (c_B^T,\,0) \begin{pmatrix} A_B^{-1}A_N \\ -A_{m+1,B}\cdot A_B^{-1}A_N + A_{m+1,N} \end{pmatrix} \\ &= c_N - c_B^T A_B^{-1} A_N \;\geq\; 0\,. \end{aligned}$$

Wir können daher mit dem dualen Simplex-Algorithmus starten. Die erste die Basis verlassende Variable ist x_{m+1}.
Für das neue Problem erhalten wir entweder eine Optimallösung oder aber die Meldung, dass

$$P^{=}\left(\begin{bmatrix} A \\ A_{m+1,\cdot} \end{bmatrix}, \begin{bmatrix} b \\ b_{m+1} \end{bmatrix}\right) = \emptyset$$

ist. Im letzteren Fall hat die Hyperebene $A_{m+1,\cdot}\, x = b_{m+1}$ einen leeren Schnitt mit $P^{=}(A, b)$.

5

Grundlagen der Polyedertheorie

5.1 Gültige Ungleichungen und Seitenflächen

Definition 5.1 (Gültige Ungleichung). *Seien $c \in \mathbb{K}^n$ ein Vektor und $\gamma \in \mathbb{K}$. Wir sagen, dass die Ungleichung*

$$c^T x \leq \gamma$$

gültig *für die Menge $S \subseteq \mathbb{K}^n$ ist, falls $c^T x \leq \gamma$ für alle $x \in S$ gilt, d.h. falls* gültig

$$S \subseteq \left\{ x \in \mathbb{K}^n : c^T x \leq \gamma \right\}.$$

Wir schreiben auch kurz $\binom{c}{\gamma}$ für die Ungleichung $c^T x \leq \gamma$. Die Menge

$$S^\gamma := \left\{ \binom{c}{\gamma} : c^T x \leq \gamma \text{ ist gültig für } S \right\}$$

nennt man auch γ-Polare *von S.* γ-Polare

Für jede Menge $S \subseteq \mathbb{K}^n$ sind die Ungleichungen $0^T x \leq 0$ und $0^T x \leq 1$ gültig. Ist $c^T x \leq \gamma$ eine gültige Ungleichung für $S \subseteq \mathbb{K}^n$, so ist auch für jedes $\lambda \in \mathbb{K}^n$ mit $\lambda \geq 0$ die Ungleichung $(\lambda c)^T x \leq \lambda\gamma$ gültig für S.

Definition 5.2 (Seitenfläche, induzierte Seitenfläche). *Sei $P \subseteq \mathbb{K}^n$ ein Polyeder. Eine Menge $F \subseteq P$ heißt* Seitenfläche von *P, wenn es eine für P gültige Ungleichung $c^T x \leq \gamma$ gibt, so dass* Seitenfläche von P

$$F = P \cap \left\{ x \in \mathbb{K}^n : c^T x = \gamma \right\}$$

gibt. Falls $F \neq \emptyset$, so sagen wir, dass die Ungleichung $c^T x \leq \gamma$ das Polyeder P stützt *bzw. dass $c^T x = \gamma$ eine* Stützhyperebene *für P ist.*

Eine Seitenfläche F von P heißt echte Seitenfläche*, falls $F \neq P$ ist. Wir bezeichnen die Seitenfläche F als* nichttrivial *oder* echt*, wenn $F \neq \emptyset$ und $F \neq P$ gilt.*

nichttrivial

Ist $c^T x \leq \gamma$ eine für das Polyeder P gültige Ungleichung, so heißt

$$P \cap \left\{ x \in \mathbb{K}^n : c^T x = \gamma \right\}$$

von $c^T x \leq \gamma$ induzierte Seitenfläche

die von $c^T x \leq \gamma$ induzierte Seitenfläche *von P.*

Da jede Seitenfläche F von $P(A, b)$ die Form

$$F = \left\{ x : Ax \leq b, c^T x \leq \gamma, -c^T x \leq -\gamma \right\}$$

besitzt, folgt, dass jede Seitenfläche selbst wieder ein Polyeder ist.

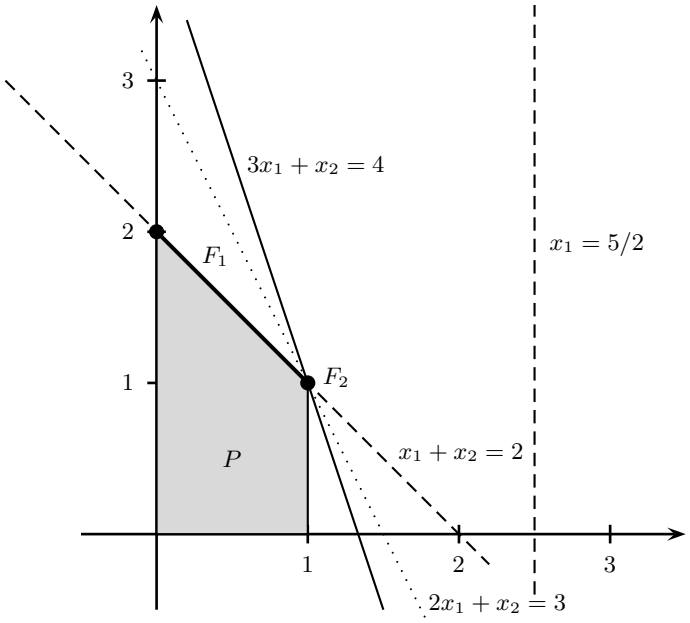


Abb. 5.1. Polyeder zu Beispiel **??**

Beispiel 5.3. Wir betrachten das Polyeder $P \subseteq \mathbb{R}^2$, das durch die Ungleichungen

$$x_1 + x_2 \leq 2 \tag{5.1a}$$
$$x_1 \leq 1 \tag{5.1b}$$
$$x_1, x_2 \geq 0 \tag{5.1c}$$

beschrieben wird. Es gilt dann $P = P(A, b)$ mit

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

Das Liniensegment F_1 von $\binom{0}{2}$ nach $\binom{1}{1}$ ist eine Seitenfläche von P, da $x_1 + x_2 \leq 2$ eine gültige Ungleichung für P ist und

$$F_1 = P \cap \left\{ x \in \mathbb{R}^2 : x_1 + x_2 = 2 \right\}$$

gilt. Die einelementige Menge $F_2 = \{\binom{1}{1}\}$ ist ebenfalls eine Seitenfläche von P, da

$$F_2 = P \cap \left\{ x \in \mathbb{R}^2 : 2x_1 + x_2 = 3 \right\}$$
$$F_2 = P \cap \left\{ x \in \mathbb{R}^2 : 3x_1 + x_2 = 4 \right\}.$$

Beide Ungleichungen $2x_1 + x_2 \leq 3$ und $3x_1 + x_2 \leq 4$ induzieren also die gleiche Seitenfläche von P. Insbesondere kann eine Seitenfläche im Allgemeinen also durch völlig verschiedene Ungleichungen (die nicht skalare Vielfache voneinander sind) induziert werden.

Die Ungleichungen $x_1 + x_2 \leq 2$, $2x_1 + x_2 \leq 3$ und $3x_1 + x_2 \leq 4$ induzieren nichtleere Seitenflächen von P. Sie stützen also das Polyeder P. Im Gegensatz dazu ist die Ungleichung $x_1 \leq 5/2$ zwar gültig für P, aber

$$F_3 = P \cap \left\{ x \in \mathbb{R}^2 : x_1 = 5/2 \right\} = \emptyset,$$

so dass $x_1 = 5/2$ keine Stützhyperebene für P bildet. ◁

Anmerkung 5.4. (i) Jedes Polyeder $P \subseteq \mathbb{K}^n$ ist eine Seitenfläche von sich selbst, da $P = P \cap \left\{ x \in \mathbb{K}^n : 0^T x = 0 \right\}$.

(ii) $\emptyset$ ist Seitenfläche jedes Polyeders $P \subseteq \mathbb{K}^n$, da $\emptyset = P \cap \left\{ x \in \mathbb{K}^n : 0^T x = 1 \right\}$.

(iii) Falls $F = P \cap \left\{ x \in \mathbb{K}^n : c^T x = \gamma \right\}$ eine nichttriviale Seitenfläche des Polyeders $P \subseteq \mathbb{K}^n$ ist, so gilt offenbar $c \neq 0$, da sonst einer der beiden Fälle (i) oder (ii) eintritt.

Betrachten wir noch einmal Beispiel **??**. Die Seitenfläche F_1 erhalten wir, indem wir eine der Ungleichungen, die P beschreiben (Ungleichung (**??**)), zu einer Gleichung machen:

$$F_1 = \left\{ x \in \mathbb{R}^2 : \begin{array}{rl} x_1 + x_2 & = 2 \\ x_1 & \leq 1 \\ x_1, x_2 & \geq 0 \end{array} \right\}$$

Die Seitenfläche F_2 lässt sich ebenso gewinnen, indem wir die Ungleichungen (**??**) und (**??**) zu Gleichungen umwandeln:

$$F_2 = \left\{ x \in \mathbb{R}^2 : \begin{array}{rl} x_1 + x_2 & = 2 \\ x_1 & = 1 \\ x_1, x_2 & \geq 0 \end{array} \right\}$$

Sei allgemein $P = P(A, b) \subseteq \mathbb{K}^n$ ein Polyeder und M die Zeilenindexmenge der Matrix A. Für eine Teilmenge $I \subseteq M$ der Zeilenindizes von A betrachten wir die Menge

$$\text{fa}(I) := \{ x \in P : A_{I\cdot} x = b_I \}. \tag{5.2}$$

Da jedes $x \in P$ das Ungleichungssystem $A_{I\cdot} x \leq b_I$ erfüllt, erhalten wir durch zeilenweise Summation mit

$$c^T := \sum_{i \in I} A_{I\cdot} \quad \text{und} \quad \gamma := \sum_{i \in I} b_i$$

eine für P gültige Ungleichung $c^T x \leq \gamma$. Für alle $x \in P \setminus \text{fa}(I)$ gibt es mindestens ein $i \in I$, so dass $A_{i\cdot} x < b_i$. Daher ist $c^T x < \gamma$ für alle $x \in P \setminus F$ und

$$\text{fa}(I) = \left\{ x \in P : c^T x = \gamma \right\}$$

ist eine Seitenfläche von P.

Definition 5.5 (Durch Indexmenge induzierte Seitenfläche). *Für ein Polyeder $P = P(A, b) \subseteq \mathbb{K}^n$ und eine Teilmenge I der Zeilenindizes der Matrix A heisst die Seitenfläche* $\text{fa}(I)$ *aus (**??**) die* von I induzierte Seitenfläche *von P.*

von I induzierte Seitenfläche

Im Beispiel **??** haben wir $F_1 = \text{fa}(\{1\})$ und $F_2 = \text{fa}(\{1, 2\})$. Der folgende Satz zeigt, dass wir für ein nichtleeres Polyeder $P \neq \emptyset$ alle Seitenflächen F in der Form $F = \text{fa}(I)$ schreiben können.

Theorem 5.6. *Sei $P = P(A, b) \subseteq \mathbb{K}^n$ ein nichtleeres Polyeder und M die Zeilenindexmenge von A. Die Menge $F \subseteq \mathbb{K}^n$ mit $F \neq \emptyset$ ist genau dann eine Seitenfläche von P, wenn $F = \text{fa}(I) = \{ x \in P : A_{I\cdot} x = b_I \}$ für eine Teilmenge $I \subseteq M$.*

Beweis. Wir haben bereits gesehen, dass $\mathrm{fa}(I)$ für jedes $I \subseteq M$ eine Seitenfläche von P ist. Sei daher umgekehrt $F = P \cap \left\{ x \in \mathbb{K}^n : c^T x = \gamma \right\}$ eine Seitenfläche von P. Dann ist F die Menge der Optimallösungen des Linearen Programms

$$\max \left\{ c^T x : Ax \leq b \right\}. \tag{5.3}$$

(wir benötigen an dieser Stelle die Annahme, dass $P = \{x : Ax \leq b\}$ nichtleer ist, da sonst das Lineare Programm (**??**) unzulässig sein könnte). Das duale Lineare Programm zu (**??**) ist

$$\min \left\{ b^T y : A^T y = c, y \geq 0 \right\}$$

hat nach dem Dualitätssatz der Linearen Programmierung ebenfalls eine optimale Lösung y^* mit $b^T y^* = \gamma$. Wir setzen $I := \{ i : y_i^* > 0 \}$. Nach dem Satz über den komplementären Schlupf sind die optimalen Lösungen von (**??**) genau diejenigen $x \in P$ mit $A_{i\cdot} x = b_i$ für $i \in I$. Daher gilt $F = \mathrm{fa}(I)$. □

Aus dem letzten Satz ergibt sich:

Korollar 5.7. *Jedes Polyeder hat nur endlich viele Seitenflächen.*

Beweis. Es gibt nur endlich viele Teilmengen $I \subseteq M = \{1, \ldots, m\}$. □

Definition 5.8 (Bindende Restriktionen). *Sei $P = P(A, b) \subseteq \mathbb{K}^n$ ein Polyeder und M die Zeilenindexmenge der Matrix A. Für $S \subseteq P$ nennen wir*

$$\mathrm{eq}(S) := \{ i \in M : A_{i\cdot} x = b_i \text{ für alle } x \in S \},$$

die Menge der für alle $x \in S$ bindenden Restriktionen *(engl.:* equality set*).*

bindenden Restriktionen

Falls $S, S' \subseteq P = P(A, b)$ mit $S \subseteq S'$ sind, dann folgt offenbar $\mathrm{eq}(S) \supseteq \mathrm{eq}(S')$. Ist daher $S \subseteq P$ nichtleer, so erfüllt jede Seitenfläche von P, die S enthält daher die Bedingung $\mathrm{eq}(F) \subseteq \mathrm{eq}(S)$. Andererseits ist $\mathrm{fa}(\mathrm{eq}(S))$ eine Seitenfläche von P, die S enthält. Damit haben wir:

Beobachtung 5.9. *Sei $P = P(A, b) \subseteq \mathbb{K}^n$ ein Polyeder und $S \subseteq P$ eine nichtleere Teilmenge von P. Die kleinste Seitenfläche von P, die S enthält, ist* $\mathrm{fa}(\mathrm{eq}(S))$.

Korollar 5.10. *(i) Das Polyeder $P = P(A, b)$ hat genau dann keine echte Seitenfläche, wenn* $\mathrm{eq}(F) = M$.
(ii) Falls $A\bar{x} < b$, dann ist $\bar{x}$ in keiner echten Seitenfläche von P enthalten.

Beweis. (i) Unmittelbar aus der Charakterisierung von Seitenflächen in Satz **??**.
(ii) Falls $A\bar{x} < b$, dann ist $\mathrm{eq}(\{\,\bar{x}\,\}) = \emptyset$ und $\mathrm{fa}(\emptyset) = P$. □

Bevor wir uns der Dimension von Polyedern widmen, betrachten wir noch einen speziellen Polyedertyp, der uns häufig begegnen wird.

Definition 5.11. *Ein Kegel $C \subseteq \mathbb{K}^n$ heißt genau dann polyedrisch, wenn C ein Polyeder ist.*

Anmerkung 5.12. Ein Kegel $C \subseteq \mathbb{K}^n$ ist genau dann polyedrisch, wenn es eine Matrix A gibt mit

$$C \;=\; P(A, 0).$$

Beweis. „$\Longleftarrow$“ Ist $C = P(A, 0)$, so ist C ein Polyeder und offensichtlich auch ein Kegel.
„$\Longrightarrow$“ Sei C ein polyedrischer Kegel. Dann existiert nach Definition **??**(c) eine Matrix A und ein Vektor b mit $C = P(A, b)$. Da $0 \in C$ gilt, folgt $b \geq 0$. Angenommen, es existiert ein $\bar{x} \in C$ mit $A\bar{x} \not\leq 0$, d.h. es existiert eine Zeile $A_{i\cdot}$ von A mit $t = A_{i\cdot}\,\bar{x} > 0$. Da C ein Kegel ist, gilt $\lambda\bar{x} \in C\ \forall\lambda \geq 0$. Jedoch für $\bar{\lambda} = \frac{b_i}{t} + 1 > 0$ gilt $\bar{\lambda}\bar{x} \in C$ und $(A_{i\cdot})(\bar{\lambda}\bar{x}) = \bar{\lambda}t = b_i + t > b_i$. Dies ist ein Widerspruch. Also erfüllen alle $x \in C$ auch $Ax \leq 0$. Daraus folgt nun $C = P(A, 0)$.

5.2 Dimension von Polyedern

Intuitiv ist der Begriff der „Dimension“ eines Objektes klar. Betrachten wir beispielsweise für die Polyeder in Abbildung **??** die „Freiheitsgrade“ bei der Bewegung innerhalb des jeweiligen Polyeders. Für P_0 und Q_0 („Punkte“) haben wir keinerlei Freiheitsgrad, so dass sich hier intuitiv die Dimension 0 ergibt. Für P_1 und Q_1 können wir uns entlang einer Strecke bewegen, wir haben hier Dimension 1. Die flächigen Polyeder P_2 und Q_2 haben Dimension 2, und der Würfel Q_3 besitzt Dimension 3.

Im Folgenden präzisieren wir den Begriff der Dimension von Polyedern. Wir erinnern daran, dass für Vektoren $v^1, \dots, v^k \in \mathbb{K}^n$ im Vektorraum $\mathbb{K}^n$, ein Vektor $x \in \mathbb{K}^n$ eine *Linearkombination* von $v^1, \dots, v^k$ ist, wenn es $\lambda_1, \dots, \lambda_k \in \mathbb{K}$ gibt, so dass

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

Definition 5.13 (Affine Kombination, affine Unabhängigkeit). *Eine* affine Kombination *der Vektoren $v^1, \dots, v^k \in \mathbb{K}^n$ ist eine Linearkombination $x = \sum_{i=1}^k \lambda_i v^i$, so dass $\sum_{i=1}^k \lambda_i = 1$.* affine Kombination

Die Vektoren $v^1, \dots, v^k \in \mathbb{K}^n$ heißen affin unabhängig, *falls aus $\sum_{i=1}^k \lambda_i v^i = 0$ und $\sum_{i=1}^k \lambda_i = 0$ folgt, dass $\lambda_1 = \lambda_2 = \cdots = \lambda_k = 0$ gilt.* affin unabhängig

Lemma 5.14. *Die folgenden Aussagen sind äquivalent:*

(i) Die Vektoren $v^1, \dots, v^k \in \mathbb{K}^n$ sind affin unabhängig.
(ii) Die Vektoren $v^2 - v^1, \dots, v^k - v^1 \in \mathbb{K}^n$ sind linear unabhängig.
(iii) Die Vektoren $\binom{v^1}{1}, \dots, \binom{v^k}{1} \in \mathbb{K}^{n+1}$ sind linear unabhängig.

Beweis. (i)⇔(ii) Falls $\sum_{i=2}^k \lambda_i (v^i - v^1) = 0$, so haben wir mit $\lambda_1 := -\sum_{i=2}^k \lambda_i$ auch $\sum_{i=1}^k \lambda_i v^i = 0$ und $\sum_{i=1}^k \lambda_i = 0$. Daher folgt aus der affinen Unabhängigkeit von $v^1, \dots, v^k$, dass $\lambda_1 = \cdots = \lambda_k = 0$, also sind $v^2 - v^1, \dots, v^k - v^1$ linear unabhängig.
Nehmen wir umgekehrt an, dass $v^2 - v^1, \dots, v^k - v^1$ linear unabhängig sind und $\sum_{i=1}^k \lambda_i v^i = 0$ mit $\sum_{i=1}^k \lambda_i = 0$, so folgt $\lambda_1 = -\sum_{i=2}^k \lambda_i$ und damit $\sum_{i=2}^k \lambda_i (v^i - v^1) = 0$. Aus der linearen Unabhängigkeit der Vektoren $v^2 - v^1, \dots, v^k - v^1$ ergibt sich $\lambda_2 = \cdots = \lambda_k = 0$ und damit dann auch $\lambda_1 = 0$.
(ii)⇔(iii) Diese Äquivalenz folgt sofort aus

$$\sum_{i=1}^{k} \lambda_i \begin{pmatrix} v^i \\ 1 \end{pmatrix} = 0 \Leftrightarrow \left\{ \begin{array}{c} \sum_{i=1}^k \lambda_i v^i = 0 \\ \sum_{i=1}^k \lambda_i = 0 \end{array} \right\}$$

Dies beendet den Beweis. □

Definition 5.15 (Dimension eines Polyeders, volldimensionales Polyeder). *Die* Dimension $\dim P$ *eines* Dimension

Polyeders $P \subseteq \mathbb{K}^n$ ist die Dimension von $\operatorname{aff} P$*, also die maximale Anzahl von affin unabhängigen Vektoren in P minus* 1*. Wir setzen* $\dim \emptyset := -1$*.*

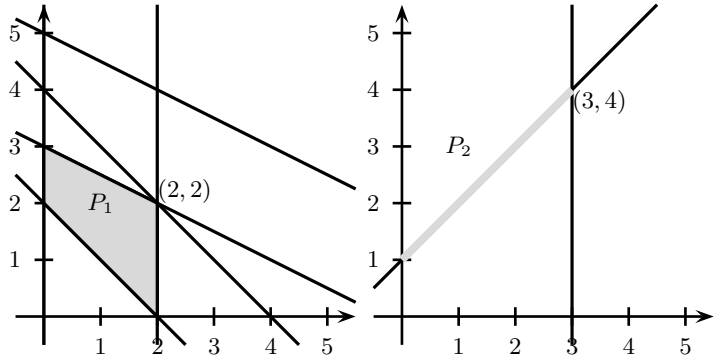
volldimensional

Ist $\dim P = n$ *gleich der Dimension des gesamten Raumes, so bezeichnen wir P als* volldimensional*.*

Beispiel 5.16. Wir betrachten das Polyeder $P_1 \subseteq \mathbb{R}^2$, das durch die folgenden Ungleichungen definiert wird:

$$x \leq 2 \tag{5.4a}$$
$$x + y \leq 4 \tag{5.4b}$$
$$x + 2y \leq 10 \tag{5.4c}$$
$$x + 2y \leq 6 \tag{5.4d}$$
$$x + y \geq 2 \tag{5.4e}$$
$$x, y \geq 0 \tag{5.4f}$$

Das Polyeder ist in Abbildung **??**(a) dargestellt. Die drei Vektoren $v^1 = (2, 0)^T$, $v^2 = (1, 1)^T$ und $v^3 = (2, 2)^T$ liegen alle in P_1 und sind darüberhinaus affin unabhängig. Somit ist die Dimension von P_1 mindestens $3 - 1 = 2$. Da im $\mathbb{R}^2$ auch nicht mehr als drei Vektoren affin unabhängig sein können, gilt $\dim P_1 = 2$ und P_1 ist volldimensional.



(a) Polyeder P_1 (b) Polyeder P_2

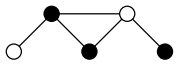
Abb. 5.2. Ein volldimensionales Polyeder P_1 und ein nicht-volldimensionales Polyeder P_2 im $\mathbb{R}^2$.

Im Gegensatz dazu ist das Polyeder $P_2 \subseteq \mathbb{R}^2$, welches durch die Ungleichungen:

$$-x + y \leq 1 \tag{5.5a}$$
$$x - y \leq -1 \tag{5.5b}$$
$$x \leq 3 \tag{5.5c}$$
$$x, y \geq 0 \tag{5.5d}$$

beschrieben ist, nicht volldimensional: Aus den ersten beiden Ungleichungen in (**??**) folgt, dass jeder Punkt $(x, y)^T \in P_2$ die Gleichung $-x+y = 1$ erfüllt. Wie das folgende Lemma zeigt, gibt es maximal $2-\text{Rang}\binom{-1}{1}+1 = 2$ affin unabhängige Vektoren x, so dass $\binom{-1}{1}^T x = 1$, womit die Dimension von P_2 maximal $2 - 1 = 1$ sein kann. ◁

Beispiel 5.17. Eine *stabile Menge* (oder *unabhängige Menge*) in einem ungerichteten Graphen $G = (V, E)$ ist eine Teilmenge $S \subseteq V$ der Eckenmenge, so dass keine zwei Ecken aus S durch eine Kante verbunden sind. Wir können das Problem, eine unabhängige Menge in G maximaler Kardinalität zu finden, als ganzzahliges Programm wie folgt formulieren:


eine stabile Menge

$$\max \sum_{v \in V} x_v \tag{5.6a}$$
$$x_u + x_v \leq 1 \qquad \text{for all edges } (u, v) \in E \tag{5.6b}$$
$$x_v \geq 0 \qquad \text{for all vertices } v \in V \tag{5.6c}$$
$$x_v \leq 1 \qquad \text{for all vertices } v \in V \tag{5.6d}$$
$$x_v \in \mathbb{Z} \qquad \text{for all vertices } v \in V \tag{5.6e}$$

Sei P das Polytop, das durch die Ungleichungen (**??**) bis (**??**) definiert wird. Wir zeigen, dass P volldimensional ist. Die n Einheitsvektoren $e_i = (0, \ldots, 1, 0, \ldots, 0)^T$, $i = 1, \ldots, n$ und der Nullvektor $e_0 := (0, \ldots, 0)^T$ liegen alle in P. Sie bilden eine Menge von $n+1$ affin unabhängigen Vektoren in P, also ist $\dim P = |V|$.

Definition 5.18 (Innerer Punkt). *Ein Punkt $\bar{x} \in P = P(A, b)$ eines Polyeders P heißt* innerer Punkt*, wenn $\bar{x}$ in keiner echten Seitenfläche von P enthalten ist. Der Punkt $\bar{x} \in P$ heißt* topologisch innerer Punkt*, wenn $A\bar{x} < b$ gilt.*

innerer Punkt

topologisch innerer Punkt

Nach Korollar **??**(ii) ist ein innerer Punkt $\bar{x}$ in keiner echten Seitenfläche enthalten.

Lemma 5.19. *Sei F eine Seitenfläche des Polyeders $P(A, b) \subseteq \mathbb{K}^n$ und $\bar{x} \in F$. Dann ist $\bar{x}$ genau dann ein innerer Punkt von F, wenn* $\text{eq}(\{\bar{x}\}) = \text{eq}(F)$.

Beweis. Sei G die inklusionsweise kleinste Seitenfläche von F, die $\bar{x}$ enthält. Dann ist $\bar{x}$ genau dann ein innerer Punkt von F, wenn $F = G$. Nach Beobachtung **??** haben wir $G = \text{fa}(\text{eq}(\{\bar{x}\}))$ und daher ist $\bar{x}$ genau dann ein innerer Punkt von F, wenn $\text{fa}(\text{eq}(\{\bar{x}\})) = F$ gilt.

Als Folge des letzten Lemmas können wir innere Punkte äquivalent auch als solche Punkte $\bar{x} \in P = P(A, b)$ definieren, die $\text{eq}(\{\bar{x}\}) = \text{eq}(P)$ erfüllen.

Lemma 5.20. *Sei $P = P(A, b)$ ein nichtleeres Polyeder. Dann besitzt P innere Punkte.*

Beweis. Sei $M = \{1, \ldots, m\}$ die Zeilenindexmenge der Matrix A, $I := \text{eq}(P)$ und $J := M \setminus I$. Falls $J = \emptyset$, also $I = M$, dann hat P nach Korollar **??**(i) keine echten Seitenflächen und jeder Punkt in P ist ein innerer Punkt.

Sei daher $J \neq \emptyset$. Dann finden wir für jedes $j \in J$ einen Punkt $x^j \in P$ mit $A_{j\cdot}x^j < b_j$. Da P konvex ist, ist der Vektor

$$\bar{x} := \frac{1}{|J|} \sum_{j \in J} x^j$$

also Konvexkombination der x^j, $j \in J$ ein Punkt in P. Es gilt dann $A_{J\cdot}\bar{x} < b_J$ und $A_{I\cdot}\bar{x} = b_I$. Daher ist $\text{eq}(\{\bar{x}\}) = \text{eq}(P)$ und die Behauptung folgt. □

Theorem 5.21 (Dimensionssatz für Seitenflächen von Polyedern). *Sei $F \neq \emptyset$ eine nichtleere Seitenfläche des Polyeders $P(A, b) \subseteq \mathbb{K}^n$. Dann gilt*

$$\dim F = n - \text{Rang}\, A_{\text{eq}(F)\cdot}.$$

Beweis. Nach dem Dimensionssatz für lineare Abbildungen (siehe z.B. [**?**]) gilt:

$$\dim \mathbb{R}^n = n = \text{Rang}\, A_{\text{eq}(F)\cdot} + \dim \text{Kern}\, A_{\text{eq}(F)\cdot}.$$

Wenn wir also zeigen können, dass $\dim \text{Kern}(A_{\text{eq}(F)\cdot}) = \dim F$, so folgt die Aussage. Wir setzen $I := \text{eq}(F)$, $r := \dim \text{Kern}\, A_{I\cdot}$ und $s := \dim F$.

„$r \geq s$“: Wähle $s+1$ affin unabhängige Vektoren $x^0, x^1, \ldots, x^s \in F$. Nach Lemma **??** sind dann $x^1 - x^0, \ldots, x^s - x^0$ linear unabhängige Vektoren und für $j = 1, \ldots, s$ gilt: $A_{I\cdot}(x^j - x^0) = b_I - b_I = 0$. Also ist die Dimension vn Kern $A_{I\cdot}$ mindestens s.

„$s \geq r$“: Nach Annahme ist $F \neq \emptyset$, also $s = \dim F \geq 0$. Daher können wir im Weiteren annehmen, dass $r \geq 0$ gilt, da für $r = 0$ nichts mehr zu zeigen ist.
Nach Lemma **??** enthält F einen inneren Punkt $\bar{x}$. Lemma **??** besagt, dass dieser Punkt die Bedingung $\mathrm{eq}(\{\bar{x}\}) = \mathrm{eq}(F) = I$ erfüllt. Mit $J := M \setminus I$ haben wir also

$$A_{I\cdot}\bar{x} = b_I \quad \text{und} \quad A_{J\cdot}\bar{x} < b_J.$$

Sei $\{x^1, \ldots, x^r\}$ eine Basis von $\operatorname{Kern} A_{I\cdot}$. Da $A_{J\cdot}\bar{x} < b_J$ können wir $\varepsilon > 0$ finden, so dass $A_{J\cdot}(\bar{x} + \varepsilon x^k) < b_J$ und $A_{I\cdot}(\bar{x} + \varepsilon x^k) = b_I$ für $k = 1, \ldots, r$. Damit haben wir $\bar{x} + \varepsilon x^k \in F$ für $k = 1, \ldots, r$.
Die Vektoren $\varepsilon x^1, \ldots, \varepsilon x^r$ sind nach Konstruktion linear unabhängig. Nach Lemma **??** ist dann $\{\bar{x}, \varepsilon x^1 + \bar{x}, \ldots, \varepsilon x^r + \bar{x}\}$ eine Menge von $r+1$ affin unabhängigen Vektoren in F. Also ist $s = \dim F \geq r$.

□

Korollar 5.22. *Sei $P = P(A, b) \subseteq \mathbb{K}^n$ ein nichtleeres Polyeder. Dann gilt:*

(i) $\dim P = n - \operatorname{Rang} A_{\mathrm{eq}(P)\cdot}$
(ii) Wenn $\mathrm{eq}(P) = \emptyset$*, dann ist P volldimensional. Gilt $A_{i\cdot} \neq 0$ für alle $i \in M$, so ist P genau dann volldimensional, wenn* $\mathrm{eq}(P) = \emptyset$.
(iii) Enthält P einen topologisch inneren Punkt, so ist P volldimensional. Gilt $A_{i\cdot} \neq 0$ für alle $i \in M$, so ist P genau dann volldimensional, wenn P einen topologisch inneren Punkt enthält.
(iv) Ist F eine echte Seitenfläche von P, dann gilt $\dim F \leq \dim P - 1$.

Beweis. (i) Benutze Satz **??** für $F = P$.
(ii) Die erste Behauptung folgt unmittelbar aus (i). Für die zweite Behauptung ist nur noch zu zeigen, dass $\dim P = n$ und $A_{i\cdot} \neq 0$ für alle $i \in M$ impliziert, dass $\mathrm{eq}(P) = \emptyset$. Wegen (i) gilt $\operatorname{Rang} A_{\mathrm{eq}(P)\cdot} = 0$ und wegen $A_{i\cdot} \neq 0$ für alle $i \in M$ haben wir dann $\mathrm{eq}(P) = \emptyset$.
(iii) P hat genau dann einen topologisch inneren Punkt, wenn $\operatorname{Rang} A_{\mathrm{eq}(P)\cdot} = \emptyset$. Die Behauptung ergibt sich nun direkt aus (ii).
(iv) Sei $I := \mathrm{eq}(P)$ und $j \in \mathrm{eq}(F) \setminus I$. Setze $J := I \cup \{j\}$. Wir zeigen, dass $A_{j\cdot}$ linear unabhängig von den Zeilen in $A_{I\cdot}$ ist. Daraus folgt dann $\operatorname{Rang} A_{\mathrm{eq}(F)\cdot} \geq$

Rang $A_{J\cdot} >$ Rang $A_{I\cdot}$ und mit dem Dimensionssatz dann $\dim F \leq \dim P - 1$.
Wir nehmen an, dass $A_{j\cdot} = \sum_{i \in I} \lambda_i A_{i\cdot}$. Wähle $\bar{x} \in F$ beliebig. Dann gilt:

$$b_j = A_{j\cdot}\bar{x} = \sum_{i \in I} \lambda_i A_{i\cdot} = \sum_{i \in I} \lambda_i b_i.$$

Da $j \notin \text{eq}(P)$, gibt es ein $x \in P$ mit $A_{j\cdot}x < b_j$. Dann haben wir aber den Widerspruch

$$b_j > A_{j\cdot}x = \sum_{i \in I} \lambda_i A_{i\cdot}x = \sum_{i \in I} \lambda_i b_i = b_j,$$

□

Beispiel 5.23. Sei $G = (V, R)$ ein gerichteter Graph und $s, t \in V$ zwei Ecken in G. Wir nennen eine Teilmenge $A \subseteq R$ einen *s-t-Connector*, falls der Teilgraph (V, A) einen s-t-Weg enthält. Man sieht leicht, dass A genau dann ein s-t-Connector ist, wenn

$$A \cap \delta^+(S) \neq \emptyset \text{ für alle } s\text{-}t\text{-Schnitte } (S, T)$$

gilt (vgl. auch [**?**, Satz 3.19]). Damit sind die s-t-Connektoren genau die Lösungen des folgenden Systems:

$$\sum_{r \in \delta^+(S)} x_r \geq 1 \qquad \text{für alle } s\text{-}t\text{-Schnitte } (S, T) \qquad (5.7a)$$

$$x_r \leq 1 \qquad \text{für alle } r \in R \qquad (5.7b)$$

$$x_r \geq 0 \qquad \text{for alle } r \in R \qquad (5.7c)$$

$$x_r \in \mathbb{Z} \qquad \text{für alle } r \in R. \qquad (5.7d)$$

Sei P das Polyeder, das durch die Ungleichungen (**??**)–(**??**) beschrieben wird. Wir werden später noch zeigen, dass P genau die konvexe Hülle der zu s-t-Connectoren gehörenden Vektoren im $\mathbb{R}^R$ ist. Momentan benötigen wir dieses Resultat aber nicht.

Sei $R' \subseteq R$ die Menge derjenigen Bögen $r \in R$, so dass es in $G - r$ noch einen s-t-Weg gibt. Mit anderen Worten, $r \in R'$, falls es mindestens einen s-t-Weg gibt, der nicht r benutzt. Wir behaupten, dass $\dim P = |R'|$ gilt. Nach dem Dimensionssatz ist dies äquivalent zu Rang $A_{\text{eq}(P)\cdot} = |R| - |R'|$.

Zunächst betrachten wir die Ungleichungen $x_r \geq 0$. Keine dieser Ungleichungen ist in eq(P), da jeder Obermenge eines s-t-Connectors wieder ein s-t-Connector ist.

Also gilt keinesfalls $x_r = 0$ für alle s-t-Connectoren x, also auch nicht für alle $x \in P$.

Jetzt untersuchen wir die Ungleichungen $x_r \leq 1$. Falls $r \notin R'$, dann benutzt jeder s-t-Weg den Bogen r. Wir definieren den s-t-Schnitt (S, T) durch

$$S := \{\, v \in V : v \text{ ist in } G - r \text{ von } s \text{ aus erreichbar} \,\}$$
$$T := V \setminus S.$$

Dann gilt $\delta^+(S) = \{r\}$. Nach (**??**) haben wir für $x \in P$:

$$1 \leq \sum_{r' \in \delta^+(S)} x_{r'} = x_r,$$

also gilt für alle $x \in P$ auch $x_r = 1$.

Falls $r \in R'$, so gibt es einen s-t-Weg W in $G - r$, der r nicht benutzt. Die Bogenmenge von W entspricht einem Vektor $x \in P$ mit $x_r = 0$. Also ist für $r \in R'$ die Ungleichung $x_r \leq 1$ nicht in $\mathrm{eq}(P)$.

Abschließend betrachten wir die Schnittungleichungen (**??**). Ist (S, T) ein s-t-Schnitt mit $\sum_{r' \in \delta^+(S)} x_{r'} = 1$ für alle $x \in P$, so muss diese Gleichung insbesondere auch für alle Inzidenzvektoren von s-t-Connectoren gelten. Da jede Obermenge eines Connectors wieder ein Connector ist, folgt $\delta^+(S) = \{r\}$ für ein $r \in R$ und es folgt $r \in R \setminus R'$. Die entsprechende Ungleichung $\sum_{r' \in \delta^+(S)} x_{r'} \geq 1$ reduziert sich also auf $x_r \geq 1$, von der wir bereits oben gesehen haben, dass sie in $\mathrm{eq}(P)$ liegt. Ist umgekehrt $\delta^+(S) = \{r\}$, so ist $r \in R \setminus R'$ und die entsprechende Ungleichung in (**??**) ist für alle $x \in P$ mit Gleichheit erfüllt.

Zusammenfassend bilden die Zeilen der Ungleichungen $x_r \leq 1$ für $r \in R \setminus R'$ eine maximale linear unabhängige Menge von Zeilen in $A_{\mathrm{eq}(P)\cdot}$. Also ist $\operatorname{Rang} A_{\mathrm{eq}(P)\cdot} = |R \setminus R'| = |R| - |R'|$.

Wir leiten nun eine weitere wichtige Konsequenz des Dimensionssatzes über die Seitenflächenstruktur von Polyedern her:

Theorem 5.24 (Hoffman and Kruskal). *Sei $P = P(A, b) \subseteq \mathbb{K}^n$ ein Polyeder. Eine nichtleere Menge $F \subseteq P$ ist genau dann eine inklusionsweise nichttriviale minimale Seitenfläche von P, wenn $F = \{\, x : A_I x = b_I \,\}$ für eine Teilmenge $I \subseteq M$ und* $\operatorname{Rang} A_{I\cdot} = \operatorname{Rang} A$.

Beweis. „⇒“: Sei F eine minimale nichttriviale Seitenfläche von P. Dann folgt nach Satz **??** und Beobachtung **??**, dass $F = \mathrm{fa}(I)$ mit $I = \mathrm{eq}(F)$. Also gilt mit $J := M \setminus I$

$$F = \{\, x : A_{I\cdot}x = b_I, A_{J\cdot}x \leq b_J \,\}\,. \tag{5.8}$$

Wir behaupten, dass $F = G$, wobei

$$G = \{\, x : A_{I\cdot}x = b_I \,\}\,. \tag{5.9}$$

Wegen (**??**) gilt offenbar $F \subseteq G$. Wir nehmen an, dass es ein $y \in G \setminus F$ gibt. Dann gibt es ein $j \in J$, so dass

$$A_{I\cdot}y = b_I, A_{j\cdot}y > b_j. \tag{5.10}$$

Sei $\bar{x} \in F$ ein innerer Punkt von F (so ein Punkt existiert nach Lemma **??**). Für $\tau \in \mathbb{R}$ betrachten wir

$$z(\tau) = \bar{x} + \tau(y - \bar{x}) = (1 - \tau)\bar{x} + \tau y.$$

Es gilt dann $A_{I\cdot}z(\tau) = (1-\tau)A_{I\cdot}\bar{x} + \tau A_{I\cdot}y = (1-\tau)b_I + \tau b_I = b_I$, da $\bar{x} \in F$ und y die Bedingung (**??**) erfüllt. Darüberhinaus gilt $A_{J\cdot}z(0) = A_{J\cdot}\bar{x} < b_J$, da $J \subseteq M \setminus I$.

Da $A_{j\cdot}y > b_j$, finden wir $\tau \in \mathbb{R}$ und $j_0 \in J$, so dass $A_{j_0\cdot}z(\tau) = b_{j_0}$ und $A_{J\cdot}z(\tau) \leq b_J$. Dann ist $\tau \neq 0$ und

$$F' := \{\, x \in P : A_{I\cdot}x = b_I, A_{j_0\cdot}x = b_{j_0} \,\}$$

eine echte Seitenfläche von F (es gilt $\bar{x} \in F \setminus F'$) im Widerspruch zur Minimalität von F.

Wir müssen noch $\operatorname{Rang} A_{I\cdot} = \operatorname{Rang} A$ zeigen. Falls $\operatorname{Rang} A_{I\cdot} < \operatorname{Rang} A$, gibt es ein $j \in J = M \setminus I$, so dass $A_{j\cdot}$ keine Linearkombination der Zeilen in $A_{I\cdot}$ ist. Dann gibt es ein $w \neq 0$ mit $A_{I\cdot}w = 0$ und $A_{j\cdot}w > 0$. Für geeignetes $\theta > 0$ erfüllt dann der Vektor $y := \bar{x} + \theta w$ die Bedingungen in (**??**) und wir können wieder eine echte Seitenfläche von F konstruieren.

„⇐“: Falls $F = \{\, x : A_{I\cdot} = b_I \,\}$, dann ist F ein affiner Teilraum und enthält nach Korollar **??**(i) keine echte Seitenfläche. Da $F \subseteq P$ haben wir $F = \{\, x : A_{I\cdot} = b_I, A_{J\cdot}x \leq b_J \,\}$ und F ist eine minimale nichttriviale Seitenfläche von P. □

Korollar 5.25. *Alle minimalen nichtleeren Seitenflächen eines Polyheders $P = P(A, b) \subseteq \mathbb{K}^n$ haben die Dimension $n - \operatorname{Rang} A$.* □

Korollar 5.26. *Sei $P = P(A,b) \subseteq \mathbb{K}^n$ ein nichtleeres Polyeder und* $\operatorname{Rang}(A) = n-k$*. Dann hat P eine Seitenfläche der Dimension k und keine nichttrivialer Seitenfläche geringerer Dimension.*

Beweis. Sei F eine nichtleere Seitenfläche von P. Dann gilt $\operatorname{Rang} A_{\mathrm{eq}(F)\cdot} \leq \operatorname{Rang} A = n-k$ und nach dem Dimensionssatz folgt $\dim(F) \geq n-(n-k) = k$. Andererseits hat nach Korollar **??** jede minimale nichtleere Seitenfläche von F Dimension $n - \operatorname{Rang} A = n-(n-k) = k$. □

In den nächsten Abschnitten behandeln wir besondere Seitenflächen, die sich im Hinblick auf die Optimierung als wichtig erweisen werden.

5.3 Ecken

Wir erinnern daran, dass ein Extremalpunkt $\bar{x}$ eines Polyeders P ein Punkt $\bar{x} \in P$ ist, der sich nicht als Konvexkombination zweier von $\bar{x}$ verschiedener Punkte in P darstellen lässt (Definition **??**). Wir haben in Lemma **??** bereits für das Polyeder $P^{=}(A,b) = \{\, x : Ax = b, x \geq 0 \,\}$ eine Charakterisierung von Extremalpunkten gefunden.

Definition 5.27 (Ecke, spitzes Polyeder). *Eine* Ecke *eines Polyeders $P = P(A,b) \subseteq \mathbb{K}^n$ ist ein Punkt $\bar{x} \in P$, so dass $\{\, \bar{x} \,\}$ eine nulldimensionale Seitenfläche von P ist. Ein Polyeder heisst* spitz*, falls P Ecken besitzt.*

Ecke

spitz

Der folgende Satz zeigt, dass für Polyeder die Begriffe der Ecke und des Extremalpunkts zusammenfallen.

Theorem 5.28 (Characterisierung von Extremalpunkten/Ecken). *Sei $P = P(A,b) \subseteq \mathbb{K}^n$ ein Polyeder und $\bar{x} \in P$. Dann sind folgende Aussagen äquivalent:*

(i) *$\bar{x}$ ist eine Ecke von P, d.h. $\{\bar{x}\}$ ist eine nulldimensionale Seitenfläche von P.*
(ii) *Es gibt einen Vektor $c \in \mathbb{R}^n$, so dass $\bar{x}$ die eindeutige Optimallösung des Linearen Programms* $\max \{\, c^T x : x \in P \,\}$ *ist.*
(iii) *$\bar{x}$ ist ein Extremalpunkt von P.*
(iv) $\operatorname{Rang} A_{\mathrm{eq}(\{\bar{x}\})\cdot} = n$.

Beweis. „(i)⇒(ii)“: Da $\{\bar{x}\}$ eine Seitenfläche von P ist, gibt es eine gültige Ungleichung $c^T x \leq \gamma$, so dass $\{x\} = \left\{ x \in P : c^T x = \gamma \right\}$. Also ist $\bar{x}$ die eindeutige Optimallösung des Linearen Programms $\max \left\{ c^T x : x \in P \right\}$.

„(ii)⇒(iii)“: Sei $\bar{x}$ die eindeutige Optimallösung von $\max \left\{ c^T x : x \in P \right\}$. Gilt $\bar{x} = \lambda x + (1-\lambda) y$ für $x, y \in P$ und $0 < \lambda < 1$, dann gilt

$$c^T \bar{x} = \lambda c^T x + (1-\lambda) c^T y \leq \lambda c^T \bar{x} + (1-\lambda) c^T \bar{x} = c^T \bar{x}.$$

Also folgt $c^T \bar{x} = c^T x = c^T y$ im Widerspruch dazu, dass $\bar{x}$ die eindeutige Optimallösung ist.

„(iii)⇒(iv)“: Sei $I := \text{eq}(\{x\})$. Falls $\text{Rang}\, A_{I\cdot} < n$, gibt es ein $y \in \mathbb{K}^n \setminus \{0\}$ mit $A_{I\cdot} y = 0$. Für hinreichend kleinesl $\varepsilon > 0$ gilt dann $x := \bar{x} + \varepsilon y \in P$ und $y := \bar{x} - \varepsilon y \in P$ (da $A_{j\cdot} \bar{x} < b_j$ für alle $j \notin I$). Dann haben wir aber $\bar{x} = \frac{1}{2} x + \frac{1}{2} y$ im Widerspruch zur Voraussetzung, dass $\bar{x}$ ein Extremalpunkt ist.

„(iv)⇒(i)“: Sei $I := \text{eq}(\{\bar{x}\})$. Nach (iv) hat das System $A_{I\cdot} x = b_I$ eine eindeutige Lösung, nämlich $\bar{x}$. Dann gilt

$$\{\bar{x}\} = \{ x : A_{I\cdot} x = b_I \} = \{ x \in P : A_{I\cdot} x = b_I \}$$

und nach Satz **??** ist $\{\bar{x}\}$ eine nulldimensionale Seitenfläche von P. □

Das obige Resultat hat interessante Folgen für die lineare Optimierung. Wir betrachten das Lineare Programm

$$\max \left\{ c^T x : x \in P \right\}, \tag{5.11}$$

wobei wir P als spitz und nichtleer annehmen. Da nach Korollar **??** alle minimalen Seitenflächen von P die gleiche Dimension haben, sind alle diese minimalen nichtleeren Seitenfläche von der Form $\{\bar{x}\}$, wobei $\bar{x}$ ein Extremalpunkt von P ist. Sei $c^T x$ beschränkt auf P. Dann gibt es nach dem Fundamentalsatz der Linearen Optimierung eine Optimallösung $x^* \in P$. Die Menge der Optimallösungen von (**??**) ist die Seitenfläche

$$F = \left\{ x \in P : c^T x = c^T x^* \right\},$$

die eine minimale nichtleere Seitenfläche $F' \subseteq F$, also eine Ecke von P, enthält. Damit haben wir

Korollar 5.29. *Falls das Polyeder $P \subseteq \mathbb{K}^n$ spitz ist und das Lineare Programm (**??**) eine Optimallösung hat, dann gibt es einen Extremalpunkt (eine Ecke) von P, die ebenfalls eine Optimallösung von P ist.*

Weiterhin haben wir:

Korollar 5.30. *Jedes Polyeder hat nur endlich viele Extremalpunkt.*

Beweis. Nach dem obigen Satz ist jeder Extremalpunkt eine Seitenfläche. Nach Korollar **??** gibt es nur endlich viele Seitenflächen.

Wir kehren kurz zum Linearen Programm (**??**) zurück, wobei wir wieder P als spitz annehmen und voraussetzen, dass (**??**) eine Optimallösung besitzt. Nach dem Satz von Hoffman and Kruskal (Satz **??**) ist jede Ecke $\bar{x}$ von P Lösung eines Systems

$$A_{I\cdot}x = b_I \text{ mit } \operatorname{Rang} A_{I\cdot} = n.$$

Im Prinzip können wir damit durch *Brute Force* eine Optimallösung von (**??**) bestimmen, indem wir alle Teilmengen $I \subseteq M$ mit $|I| = n$ enumerieren, testen, ob $\operatorname{Rang} A_{I\cdot} = n$ und dann $A_{I\cdot}x = b_I$ lösen. Wenn wir die beste zulässige Lösung, die wir auf diese Weise erhalten, wählen, so ist dies eine Optimallösung. Die Simplex-Methode aus Kapitel **??** ist ein effizienteres und systematisches Verfahren.

Korollar 5.31. *Ein nichtleeres Polyeder $P = P(A, b) \subseteq \mathbb{K}^n$ ist genau dann spitz, wenn* $\operatorname{Rang} A = n$.

Beweis. Nach Korollar ?? haben die minimalen Seitenflächen von P genau dann Dimension 0, wenn $\operatorname{Rang} A = n$. □

Korollar 5.32. *Jedes nichtleere Polytop ist spitz.*

Beweis. Sei $P = P(A, b)$ und $\bar{x} \in P$ beliebig. Nach Korollar **??** genügt es zu zeigen, dass $\operatorname{Rang} A = n$. Falls $\operatorname{Rang} A < n$, gibt es $y \in \mathbb{R}^n$ mit $y \neq 0$ und $Ay = 0$. Dann folgt aber $x + \theta y \in P$ für alle $\theta \in \mathbb{K}$ im Widerspruch zur Beschränktheit von P.

Korollar 5.33. *Jedes nichtleere Polyeder $P \subseteq \mathbb{K}^n_+$ ist spitz.*

Beweis. Gilt $P = P(A, b) \subseteq \mathbb{K}^n_+$, so können wir P äquivalent auch folgendermassen schreiben:

$$P = \left\{ x : \begin{pmatrix} A \\ -I \end{pmatrix} x \leq \begin{pmatrix} b \\ 0 \end{pmatrix} \right\} = P(\bar{A}, \bar{b}).$$

Da $\operatorname{Rang} \bar{A} = \operatorname{Rang} \begin{pmatrix} A \\ -I \end{pmatrix} = n$, folgt die Behauptung. □

Satz **??**(ii) ist eine Formalisierung der Intuation, dass wir durch Optimieren mit einem geeigneten Vektor jede Ecke „aussondern“ können. Wir zeigen nun eine stärkere Aussage für rationale Polyeder.

Theorem 5.34. *Sei $P = P(A, b)$ ein rationales Polyeder und $\bar{x} \in P$ eine Ecke von P. Dann gibt es einen ganzzahligen Vektor $c \in \mathbb{Z}^n$, so dass $\bar{x}$ die eindeutige Optimallösung von $\max\{ c^T x : x \in P \}$ ist.*

Beweis. Sei $I := \text{eq}(\{x\})$ und $M := \{1, \ldots, m\}$ die Indexmenge der Zeilen der Matrix A. Betrachte den Vektor $\bar{c} = \sum_{i \in M} A_{i\cdot}$. Da alle Zeilen $A_{i\cdot}$ rational sind, gibt es ein $\theta > 0$, so dass $c := \theta \bar{c} \in \mathbb{Z}^n$ ganzzahlig ist. Wegen $\text{fa}(I) = \{x\}$ (cf. Beobachtung **??**), gibt es für jedes $x \in P$ mit $x \neq \bar{x}$ mindestens ein $i \in I$, so dass $A_{i\cdot} x < b_i$. Daher gilt für alle $x \in P \setminus \{\bar{x}\}$

$$c^T x = \theta \sum_{i \in M} A_{i\cdot}^T x < \theta \sum_{i \in M} b_i = \theta c^T x^0.$$

Dies zeigt die Behauptung.

Im Rest dieses Abschnittes beschäftigen wir uns mit Polytopen und zeigen, dass diese genau die konvexen Hüllen endlich vieler Punkte sind. Dazu benötigen wir zunächst eine Hilfsaussage:

Lemma 5.35. *Sei $X \subset \mathbb{K}^n$ eine endliche Menge und $v \in \mathbb{R}^n \setminus \text{conv}(X)$. Dann gibt es eine Ungleichung, die $\text{conv}(X)$ und v trennt, d.h., es gibt $c \in \mathbb{R}^n$ und $\gamma \in \mathbb{R}$ so dass $c^T x \leq \gamma$ für alle $x \in \text{conv}(X)$ und $c^T v > t$.*

Beweis. Sei $X = \{x_1, \ldots, x_k\}$. Da $v \notin \text{conv}(X)$ ist das System

$$\begin{aligned} \sum_{i=1}^{k} \lambda_k x_k &= v \\ \sum_{i=1}^{k} \lambda_k &= 1 \\ \lambda_i &\geq 0 \quad \text{for } i = 1, \ldots, k \end{aligned}$$

unlösbar. Nach Farkas' Lemma gibt es einen Vektor $\binom{y}{z} \in \mathbb{R}^{n+1}$, so dass

$$y^T x_i + z \leq 0, \quad \text{for } i = 1, \ldots, k$$
$$y^T v + z > 0.$$

Wenn wir $c := -y$ und $\gamma := -z$ wählen, so gilt $c^T x_i \leq \gamma$ für $i = 1, \ldots, k$.

Sei $x = \sum_{i=1}^{k} \lambda_i x_i$ eine Konvexkombination der x_i. Dann gilt

$$c^T x = \sum_{i=1}^{k} \lambda_i c^T x_i \leq \max\{c^T x_i : i = 1, \ldots, k\} \leq \gamma.$$

Also ist $c^T x \leq \gamma$ für alle $x \in \text{conv}(X)$. □

Theorem 5.36. *Ein Polytop P ist die konvexe Hülle seiner Ecken.*

Beweis. Die Behauptung ist trivial, wenn $P =$. Sei also $P = P(A, b)$ ein nichtleeres Polyeder und $X = \{x_1, \ldots, x_k\}$ die Menge seiner Ecken (diese existieren nach Korollar **??** und sind endlich nach Korollar **??**). Da P konvex ist und $x_1, \ldots, x_k \in P$, folgt $\text{conv}(X) \subseteq P$. Wir müssen zeigen, dass $\text{conv}(X) = P$ gilt.

Angenommen, es gäbe $v \in P \setminus \text{conv}(X)$. Dann gibt es nach Lemma (**??**) eine für $\text{conv}(X)$ gültige Ungleichung $c^T x \leq \gamma$, so dass $c^T v > \gamma$. Da P beschränkt und nichtleer ist, hat das Linare programm $\max\left\{ c^T x : x \in P \right\}$ eine Optimallösung mit endlichem Optimalwert $\gamma^* \in \mathbb{K}$.

Da $v \in P$ folgt $\gamma^* \geq c^T v > \gamma$. Dann ist aber keine der Ecken aus X eine Optimallösung im Widerspruch zu Korollar **??**.

Theorem 5.37. *Eine Menge $P \subseteq \mathbb{K}^n$ ist genau dann ein Polytop, wenn $P = \text{conv}(X)$ für eine endliche Menge $X \subseteq \mathbb{K}^n$.*

Beweis. Nach Satz **??** gilt für jedes Polytop P, dass $P = \text{conv}(X)$, wobei X die endliche Menge seiner Ecken ist. Somit verbleibt nur, die andere Richtung des Satzes zu zeigen.

Sei $X = \{x_1, \ldots, x_k\} \subseteq \mathbb{K}^n$ eine endliche Menge und $P = \text{conv}(X)$. Wir definieren $Q \subseteq \mathbb{K}^{n+1}$ durch

$$Q := \left\{ \begin{pmatrix} a \\ t \end{pmatrix} : a \in [-1, 1]^n, t \in [-1, 1], a^T x \leq t \text{ für alle } x \in X. \right\}$$

Da Q nach Konstruktion beschränkt ist, ist Q ein Polytop. Seinen $A := \{\binom{a_1}{t_1}, \ldots, \binom{a_p}{t_p}\}$ die Ecken von Q. Nach Satz **??** gilt $Q = \text{conv}(A)$. Wir setzen

$$P' := \left\{ x \in \mathbb{K}^n : a_j^T x \leq t_j, j = 1, \ldots, p \right\}.$$

Wir zeigen $P = P'$, was den Beweis beendet.

Sei $\bar{x} \in P = \text{conv}(X)$ mit $\bar{x} = \sum_{i=1}^{k} \lambda_i x_i$ eine Konvexkombination der Vektoren in X. Sei $j \in \{1, \ldots, p\}$ fest. Da $\binom{a_j}{t_j} \in Q$ gilt $a_j^T x_i \leq t_j$ für alle i und daher

$$a_j^T \bar{x} = \sum_{i=1}^{k} \lambda_i \underbrace{a_j^T x_i}_{\leq t_j} = \sum_{i=1}^{k} \lambda_i t_j = t_j.$$

Somit ist $a_t^T \bar{x} \leq t_j$ für alle j und $\bar{x} \in P'$. Ergo ist $P \subseteq P'$.

Wir nehmen an, dass die andere Inklusion nicht gilt, dass es also $v \in P' \setminus P$ gibt. Nach Lemma **??** gibt es eine Ungleichung $c^T x \leq \gamma$, so dass $c^T x \leq \gamma$ für alle $x \in P$ und $c^T v > t$. Sei $\theta > 0$, so dass $\bar{c} := c/\theta \in [-1, 1]^n$ und $\bar{\gamma} := \gamma/\theta \in [-1, 1]$. Dann gilt immer noch $\bar{c}^T v > \gamma$ und $\bar{c}^T x \leq \bar{\gamma}$ für alle $x \in P$. Also ist $\binom{\bar{c}}{\bar{\gamma}} \in Q$.

Da $Q = \text{conv}(A)$ können wir $\binom{\bar{c}}{\bar{\gamma}}$ als Konvexkombination $\binom{\bar{c}}{\bar{\gamma}} = \sum_{j=1}^{p} \lambda_j \binom{a_j}{t_j}$ der Ecken von Q schreiben. Da $v \in P'$, gilt $a_j^T \leq t_j$ für alle j. Damit ergibt sich

$$\bar{c}^T v = \sum_{j=1}^{p} \lambda_j a_j^T v \leq \sum_{j=1}^{p} \lambda_j t_j = \gamma,$$

im Widerspruch zu $\bar{c}^T v > \gamma$. □

Beispiel 5.38. Als Anwendung von Satz **??** betrachten wir das sogenannte *stabile Mengen Polytop* $\text{STAB}(G)$, das als die konvexe Hülle der Inzidenzvektoren von stabilen Mengen des ungerichteten Graphen G definiert ist (vgl. Beispiel **??**):

$$\text{STAB}(G) = \text{conv}(\left\{ x \in \mathbb{B}^V : \text{ } x \text{ ist ein Inzidenzvektor einer stabilen Menge in } G \right\}). \tag{5.12}$$

Nach Satz **??** ist $\text{STAB}(G)$ ein Polytop, dessen Ecken alle Inzidenzvektoren von stabilen Mengen in G sind.

Die n Einheitsvektoren $e_i = (0, \ldots, 1, 0, \ldots, 0)^T$, $i = 1, \ldots, n$ und der Nullvektor $e_0 := (0, \ldots, 0)^T$ liegen alle in $\text{STAB}(G)$, womit wie in Beispiel **??** folgt, dass $\dim \text{STAB}(G) = n$ volldimensional ist.

5.4 Facetten

Im letzten Abschnitt haben wir gesehen, dass für jede endliche Menge $X \subset \mathbb{K}^n$ ihre konvexe Hülle $\text{conv}(X)$ ein Po-

lytop ist. Also gilt für endliches $X \subset \mathbb{K}^n$:

$$\operatorname{conv}(X) = P(A, b) = \{\, x : Ax \leq b \,\}$$

für geeignetes A und b. Es stellt sich die Frage, welche Ungleichungen notwendig und ausreichend sind, um ein Polytop oder Polyeder zu beschreiben.

Definition 5.39 (Facette). *Eine nichttriviale Seitenfläche F des Polyeders $P = P(A, b) \subseteq \mathbb{K}^n$ heißt* Facette*, wenn F nicht strikt in einer echten Seitenfläche von P enthalten ist.* Facette

Theorem 5.40 (Characterisierung von Facetten). *Sei $P = P(A, b) \subseteq \mathbb{K}^n$ ein Polyeder und F eine Seitenfläche von P. Dann sind folgende Aussagen äquivalent:*

(i) F *ist eine Facette von* P.
(ii) $\operatorname{Rang} A_{\operatorname{eq}(F)\cdot} = \operatorname{Rang} A_{\operatorname{eq}(P)\cdot} + 1$
(iii) $\dim F = \dim P - 1$.

Beweis. Die Äquivalenz von (ii) und (iii) ist eine unmittelbare Konsequenz aus dem Dimensionsatz (Satz **??**).

„(i)⇒(iii)“: Sei F eine Facette aber $k = \dim F < \dim P - 1$. Wegen der Äquivalent von (ii) und (iii) gilt $\operatorname{Rang} A_{I\cdot} > A_{\operatorname{eq}(P)\cdot} + 1$, wobei $I = \operatorname{eq}(F)$. Wähle $i \in I$ so dass für $J := I \setminus \{\, i \,\}$ gilt: $\operatorname{Rang} A_{J\cdot} = \operatorname{Rang} A_{I\cdot} - 1$. Dann ist $\operatorname{fa}(J)$ eine Seitenfläche, die F enthält und Dimension $k + 1 \leq \dim P - 1$ besitzt. Also ist $\operatorname{fa}(J)$ eine nichttriviale Seitenfläche, die F strikt enthält im Widerspruch zur Maximalität von F.

„(iii)⇒(i)“: Sei G eine nichttriviale Seitenfläche von P, die F strikt enthält. Dann ist F eine nichttriviale Seitenfläche von G und aus Korollar **??**(iv) (angewendet auf F und $P' = G$) erhalten wir $\dim F \leq \dim G - 1$. Zusammen mit $\dim F = \dim P - 1$ folgt $\dim G = \dim P$. Dann ist kann aber wieder nach Korolar **??**(iv) G keine nichttriviale Seitenfläche von P sein.

Beispiel 5.41. Wir betrachten noch einmal das stabile Mengen Polytop $\operatorname{STAB}(G)$ aus Beispiel **??**. Für $v \in V$ definiert die Ungleichung $x_v \geq 0$ eine Facette von $\operatorname{STAB}(G)$, da die $n-1$ Einheitsvektoren mit Einsen an anderen Stellen als v zusammen mit dem Nullvektor n affin unabhängige Vektoren in $\operatorname{STAB}(G)$ sind, die alle $x_v = 0$ erfüllen.

Als Folgerung des letzten Satzes leiten wir her, dass für jede Facette F eines Polyeders $P = P(A, b)$ mindestens

eine Ungleichung im System $Ax \leq b$ existieren muss, die F induziert.

Korollar 5.42. *Sei $P = P(A, b) \subseteq \mathbb{R}^n$ ein Polyeder und F eine Facette von P. Dann gibt es ein $j \in M \setminus \text{eq}(P)$, so dass*

$$F = \{\, x \in P : A_{j.}x = b_j \,\}. \tag{5.13}$$

Beweis. Sei $I = \text{eq}(P)$. Wir wählen $j \in \text{eq}(F) \setminus I$ beliebig und setzen $J := I \cup \{j\}$. Es gilt immer noch $J \subseteq \text{eq}(F)$, da $I \subseteq \text{eq}(F)$ und $j \in \text{eq}(F)$. Also ist $F \subseteq \text{fa}(J) \subset P$ (es gilt $\text{fa}(J) \neq P$, da jeder innere Punkt $\bar{x}$ von P die Bedingung $A_{j.}\bar{x} < b_j$ wegen $j \in \text{eq}(F) \setminus I$ erfüllt). Wegen der Maximalität von F folgt $F = \text{fa}(J)$. Also gilt,

$$\begin{aligned} F = \text{fa}(J) &= \{\, x \in P : A_{J.}x \leq b_J \,\} \\ &= \{\, x \in P : A_{I.}x = b_I, A_{j.}x = b_j \,\} \\ &= \{\, x \in P : A_{j.}x = b_j \,\}, \end{aligned}$$

wobei sich die letzte Gleichung aus $I = \text{eq}(P)$ ergibt. □

5.5 Projektion von Polyedern

Projektionen von Polyedern werden uns künftig häufiger begegnen. Sie sind ein wichtiges Hilfsmittel, um gewisse Aussagen über Polyeder abzuleiten oder beweisen zu können. Wir wollen in diesem Abschnitt ein Verfahren angeben, wie man die Projektion eines Polyeders auf eine andere Menge berechnen kann und daraus dann weitere Konsequenzen ableiten.

Definition 5.43. *Es seien S, $H \subseteq \mathbb{K}^n$ und $c \in \mathbb{K}^n \setminus \{0\}$. Die Menge*

$$\{x \in H \mid \exists \lambda \in \mathbb{K} \text{ mit } x + \lambda c \in S\}$$

heißt Projektion von S auf H entlang c. Gilt

$$H = \{x \in \mathbb{K}^n \mid c^T x = \gamma\}$$

für ein $\gamma \in \mathbb{K}$, so heißt die Projektion orthogonal.

Altes Kommando `df` hier verwendet

Altes Kommando `df` hier verwendet

Wir interessieren uns für den Fall, dass S ein Polyeder ist, d.h. wir müssen uns überlegen, wie die Projektion von Halbräumen aussieht. Dazu machen wir zunächst eine einfache Beobachtung.

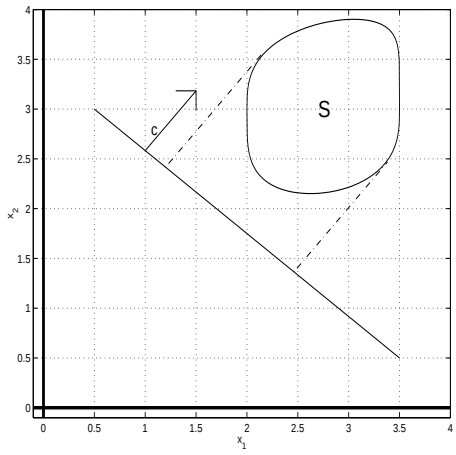


Abb. 5.3. Projektion der Menge S auf H (Definition **??**).

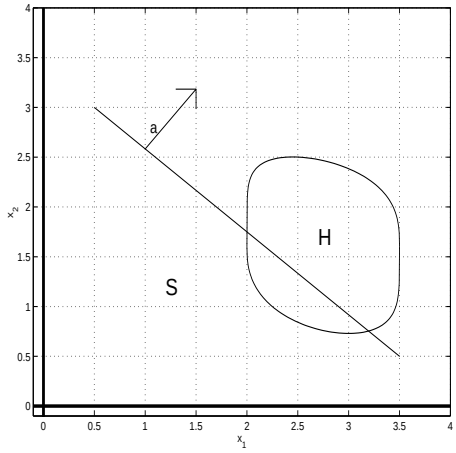


Abb. 5.4. Projektion der Menge S auf H (Bemerkung **??**).

Anmerkung 5.44. Sei $H \subseteq \mathbb{K}^n, c \in \mathbb{K}^n \setminus \{0\}, S = \{x \in \mathbb{K}^n \mid a^T x \leq \alpha\}$ ein Halbraum und P die Projektion von S auf H entlang c. Dann gilt:

(a) Ist a orthogonal zu c, dann gilt $P = H \cap S$.
(b) Ist a nicht orthogonal zu c, dann gilt $P = H$.

Beweis. Per Definition ist $P = \{x \in H \mid \exists \lambda \in \mathbb{K} \text{ mit } a^T(x + \lambda c) \leq \alpha\}$ und somit gilt in (a) die Hinrichtung „$\subseteq$“ per Definition; für die Rückrichtung wähle $\lambda = 0$. In (b) gilt

die Hinrichtung „$\subseteq$“ ebenfalls per Definition, und für die Rückrichtung wähle $\lambda = \frac{\alpha - a^T x}{a^T c}$.

Betrachten wir die Projektion des Durchschnitts zweier Halbräume

$$\begin{aligned} H_1 &= \{x \in \mathbb{K}^n \mid a_1^T x \leq \alpha_1\}, \\ H_2 &= \{x \in \mathbb{K}^n \mid a_2^T x \leq \alpha_2\}. \end{aligned}$$

Steht a_1 senkrecht zu c oder a_2 senkrecht zu c, so erhalten wir die Projektion aus der Beobachtung **??**. Es bleiben die beiden folgenden Fälle zu unterscheiden:

(1) Beide Winkel $\measuredangle(a_i, c)$ sind kleiner als 90° oder beide Winkel liegen zwischen 90° und 180°. In diesem

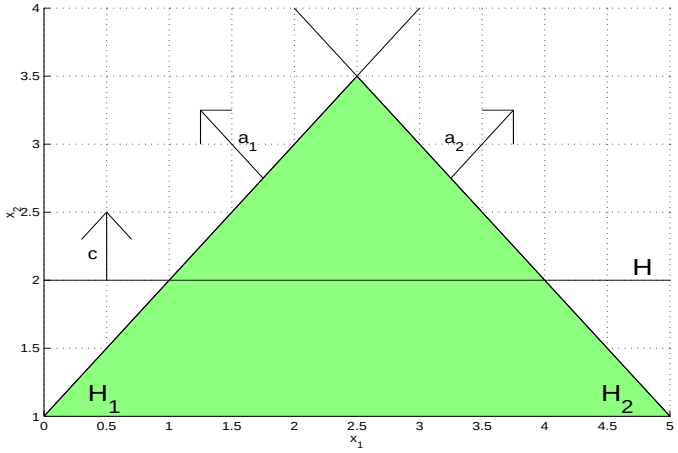


Abb. 5.5. Winkel im Fall (1)

Fall ist die Projektion des Durchschnitts der beiden Halbräume wieder H.

(2) Ein Winkel ist kleiner als 90° und einer liegt zwischen 90° und 180°. In diesem Fall ist die Projektion von $H_1 \cap H_2$ auf H eine echte Teilmenge von H.
Wir wollen diese Teilmenge zunächst *geometrisch* bestimmen. Man betrachte dazu den Durchschnitt der zu H_1 und H_2 gehörenden Hyperebenen

$$H'_{12} = \{x \in \mathbb{K}^n \mid a_1^T x = \alpha_1 \text{ und } a_2^T x = \alpha_2\},$$

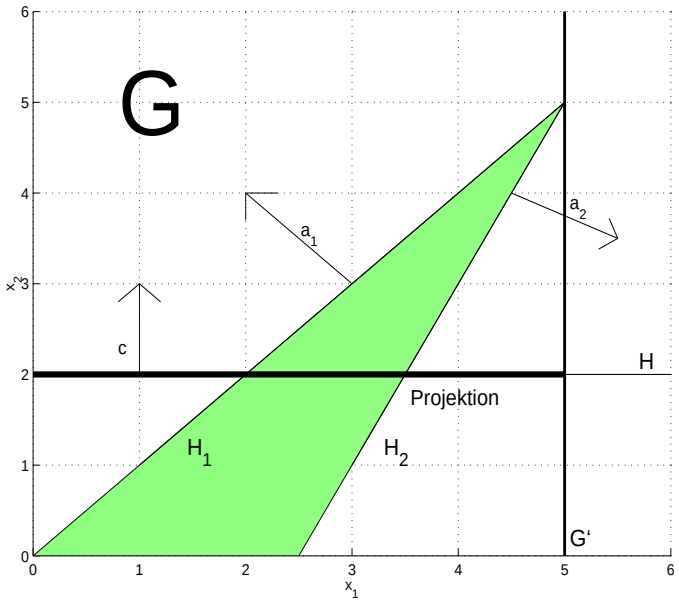


Abb. 5.6. Geometrische Bestimmung der Projektion im Fall (2).

dann ist

$$G' \;=\; \{x \in \mathbb{K}^n \;\big|\; \exists y \in H'_{12} \text{ und } \lambda \in \mathbb{K} \text{ mit } x \;=\; y{+}\lambda c\}$$

eine Hyperebene, deren Normalenvektor senkrecht auf c steht. Sei G der zu G' gehörende Halbraum, der $H_1 \cap H_2$ enthält. Dann ist $G \cap H$ die gesuchte Projektion (vgl. Beobachtung **??** (a)).

Algebraisch lässt sich G wie folgt bestimmen: Aufgrund der Formel

$$\cos \measuredangle(x,y) \;=\; \frac{x^T y}{||x|| \cdot ||y||}$$

lassen sich Aussagen über den Winkel $\measuredangle(x,y)$ zwischen den Vektoren x und y beschreiben, wobei $x, y \neq 0$ gelte. Sei nun

$$\begin{aligned} 90^\circ \;&<\; \measuredangle(c,a_1) \;<\; 180^\circ \quad \text{und} \\ 0^\circ \;&<\; \measuredangle(c,a_2) \;<\; 90^\circ. \end{aligned}$$

Dann ist $c^T a_1 < 0$ und $c^T a_2 > 0$. Wir bestimmen eine geeignete nicht-negative Linearkombination d aus a_1

und a_2 (genaue Formel siehe Satz **??**), so dass d auf c senkrecht steht, d.h. $d^T c = 0$ gilt. Dann ist

$$G \;=\; \{x \in \mathbb{K}^n \;\big|\; d^T x \;\leq\; \delta\},$$

wobei δ entsprechend aus a_1 und a_2 erzeugt wird, der oben beschriebene Halbraum.
Die genauen Formeln und deren Korrektheit werden dargestellt im

Theorem 5.45. *Sei $c \in \mathbb{K}^n \setminus \{0\}$, $H \subseteq \mathbb{K}^n$ eine beliebige Menge und*

$$H_i \;=\; \{x \in \mathbb{K}^n \;\big|\; a_i^T x \;\leq\; \alpha_i\}$$

für $i = 1, 2$. Wir bezeichnen mit P_H die Projektion von $H_1 \cap H_2$ auf H entlang c. Dann gelten die folgenden Aussagen:

(a) Gilt $a_i^T c = 0$ für $i = 1, 2$, so ist $P_H = H_1 \cap H_2 \cap H$.
(b) Gilt $a_1^T c = 0$ und $a_2^T c \neq 0$, so ist $P_H = H_1 \cap H$.
(c) Gilt $a_i^T c > 0$ für $i = 1, 2$ oder $a_i^T c < 0$ für $i = 1, 2$, so ist $P_H = H$.
(d) Gilt $a_1^T c < 0$ und $a_2^T c > 0$, dann setzen wir

$$\begin{aligned} d \;&=\; (a_2^T c)a_1 \;-\; (a_1^T c)a_2\,, \\ \delta \;&=\; (a_2^T c)\alpha_1 \;-\; (a_1^T c)\alpha_2\,, \end{aligned}$$

woraus dann die Darstellung

$$P_H \;=\; H \cap \{x \in \mathbb{K}^n \;\big|\; d^T x \;\leq\; \delta\}.$$

folgt.

Beachte in (d), dass $d^T x \leq \delta$ eine nicht-negative Linearkombination aus $a_i^T x \leq \alpha_i$ ist, und d auf c senkrecht steht. Zum Beweis des Satzes **??** benötigen wir folgendes Lemma.

Lemma 5.46. *Sei $\bar{x} \in \mathbb{K}^n$, $c \in \mathbb{K}^n \setminus \{0\}$ und $H = \{x \in \mathbb{K}^n \mid a^T x \leq \alpha\}$ mit $a^T c \neq 0$. Dann gilt:*

$$\begin{aligned} &\textit{(i)}\; a^T c > 0 \quad \Rightarrow \quad \bar{x} + \lambda c \in H \quad \forall\, \lambda \leq \frac{\alpha - a^T \bar{x}}{a^T c}, \\ &\textit{(ii)}\; a^T c < 0 \quad \Rightarrow \quad \bar{x} + \lambda c \in H \quad \forall\, \lambda \geq \frac{\alpha - a^T \bar{x}}{a^T c}. \end{aligned}$$

Beweis. Folgende Rechnung gilt sowohl für (i) als auch für (ii):

$$a^T(\bar{x}+\lambda c) \;=\; a^T\bar{x} + \lambda a^T c \;\leq\; a^T\bar{x} + \frac{\alpha - a^T\bar{x}}{a^T c}\cdot a^T c \;=\; \alpha.$$

Mit dieser Hilfe folgt nun der

Beweis von Satz ??.

(a),(b) beweist man analog zu Bemerkung **??**.

(c) Da $P_H \subseteq H$ per Definition gilt, bleibt $H \subseteq P_H$ zu zeigen. Sei also $\bar{x} \in H$ beliebig. Setze

$$\bar{\lambda} \;=\; \min\left\{ \frac{\alpha_i - a_i^T\bar{x}}{a_i^T c} \;\middle|\; i = 1,2 \right\}.$$

Dann gilt mit Lemma **??**: $\bar{x}+\lambda c \in H_1 \cap H_2$ für alle $\lambda \leq \bar{\lambda}$. Daraus folgt nun mit Definition **??** die verlangte Aussage $\bar{x} \in P_H$.

(d) Sei

$$Q \;=\; \{x \in \mathbb{K}^n \mid d^T x \leq \delta\}$$

und Q_H die Projektion von Q auf H entlang c. Da $d^T x \leq \delta$ eine konische Kombination aus $a_i^T x \leq \alpha_i$ für $i = 1, 2$ ist, gilt also $H_1 \cap H_2 \subseteq Q$ und damit $P_H \subseteq Q_H$. Da weiterhin $d^T c = 0$ ist, gilt nach Bemerkung **??** entsprechend $Q_H = Q \cap H$ und damit $P_H \subseteq Q \cap H$. Es bleibt noch $Q \cap H \subseteq P_H$ zu zeigen. Dazu sei $\bar{x} \in Q \cap H$ beliebig und

$$\lambda_i \;=\; \frac{\alpha_i - a_i^T\bar{x}}{a_i^T c}, \text{ für } i = 1, 2.$$

Nach Voraussetzung ist

$$(a_2^T c)a_1^T\bar{x} - (a_1^T c)a_2^T\bar{x} \;=\; d^T\bar{x} \;\leq\; \delta \;=\; (a_2^T c)\alpha_1 - (a_1^T c)\alpha_2$$

und damit

$$(a_1^T c)(\alpha_2 - a_2^T\bar{x}) \;\leq\; (a_2^T c)(\alpha_1 - a_1^T\bar{x}) \qquad \Longleftrightarrow \quad \lambda_2 \;\geq\; \lambda_1.$$

Für ein beliebiges $\lambda \in [\lambda_1, \lambda_2]$ gilt nun

$$a_i^T(\bar{x} + \lambda c) \;=\; a_i^T\bar{x} \;+\; \lambda a_i^T c \;\leq\; a_i^T\bar{x} \;+\; \lambda_i a_i^T c \;=\; \alpha_i.$$

Also ist $\bar{x} + \lambda c \in H_1 \cap H_2$ und damit $\bar{x} \in P_H$.

Damit können wir unseren Projektionsalgorithmus angeben.

Projektion eines Polyeders entlang c.

Input:
- $c \in \mathbb{K}^n$: die Projektionsrichtung.
- $P(A,b)$: ein Polyeder.

Output: $P(D,d)$, so dass für jede beliebige Menge $H \subseteq \mathbb{K}^n$ die Menge

$$H \cap P(D,d)$$

die Projektion von $P(A,b)$ auf H entlang c ist.

(1) Partitioniere die Zeilenindexmenge $M = \{1, 2, \ldots, m\}$ von A wie folgt:

$$\begin{aligned} N &= \{i \in M \mid A_{i\cdot}\, c < 0\}, \\ Z &= \{i \in M \mid A_{i\cdot}\, c = 0\}, \\ P &= \{i \in M \mid A_{i\cdot}\, c > 0\}. \end{aligned}$$

(2) Setze $r = |Z \cup (N \times P)|$ und sei $p : \{1, 2, \ldots, r\} \to Z \cup (N \times P)$ eine Bijektion.

(3) Für $i = 1, 2, \ldots, r$ führe aus:
   (a) Ist $p(i) \in Z$, dann setze $D_{i\cdot} = A_{p(i)\cdot}$, $d_i = b_{p(i)}$ (vgl. Satz **??** (a)).
   (b) Ist $p(i) = (s,t) \in N \times P$, dann setze

$$\begin{aligned} D_{i\cdot} &= (A_{t\cdot}\, c)A_{s\cdot} - (A_{s\cdot}\, c)A_{t\cdot}, \\ d_i &= (A_{t\cdot}\, c)b_s - (A_{s\cdot}\, c)b_t. \end{aligned}$$

   (vgl. Satz **??** (d)).

(4) **Stop** ($P(D,d)$ ist das gesuchte Objekt).

Theorem 5.47. *Seien A, b, c, sowie D, d wie im Algorithmus **??** gegeben. H sei eine beliebige Menge und P_H die Projektion von $P(A,b)$ auf H entlang c. Dann gilt:*

(a) Für alle $i \in \{1, 2, \ldots, r\}$ existiert ein $u_i \geq 0$ mit $D_{i\cdot} = u_i^T A$, $d_i = u_i^T b$.
(b) Für alle $i \in \{1, 2, \ldots, r\}$ gilt $D_{i\cdot}\, c = 0$.
(c) $P_H = H \cap P(D,d)$.
(d) Sei $\bar{x} \in H$ und

$$\lambda_i = \frac{b_i - A_{i\cdot}\, \bar{x}}{A_{i\cdot}\, c} \quad \forall\, i \in P \cup N,$$

$$L = \begin{cases} -\infty, & \text{falls } N = \emptyset \\ \max\{\lambda_i \mid i \in N\}, & \text{falls } N \neq \emptyset. \end{cases}$$

$$U = \begin{cases} \infty, & \text{falls } P = \emptyset \\ \min\{\lambda_i \mid i \in P\}, & \text{falls } P \neq \emptyset. \end{cases}$$

Dann gilt:

(d1) $\bar{x} \in P(D,d) \;\Rightarrow\; L \le U$ *und* $\bar{x}+\lambda c \in P(A,b)\ \forall\, \lambda \in [L,U]$.

(d2) $\bar{x} + \lambda c \in P(A,b) \;\Rightarrow\; \lambda \in [L,U]$ *und* $\bar{x} \in P(D,d)$.

Beweis. (a) und (b) folgen direkt aus der Konstruktion von D.

(c) Zum Beweis dieses Punktes benutzen wir (d):
Sei $\bar{x} \in P_H$. Daraus folgt $\bar{x} \in H$ und die Existenz eines $\lambda \in \mathbb{K}$ mit $\bar{x} + \lambda c \in P(A,b)$. Mit (d2) folgt daraus $\bar{x} \in P(D,d)$.
Umgekehrt sei $\bar{x} \in H \cap P(D,d)$. Daraus folgt mit (d1) $\bar{x} + \lambda c \in P(A,b)$ für ein $\lambda \in \mathbb{K}$, also ist $\bar{x} \in P_H$.

(d1) Sei $\bar{x} \in P(D,d)$. Wir zeigen zuerst $L \le U$.
Ist $P = \emptyset$ oder $N = \emptyset$, so ist dies offensichtlich richtig. Andernfalls sei $p(v) = (s,t)$ mit $\lambda_s = L$ und $\lambda_t = U$. Dann folgt analog zum Beweis von Satz **??** (d) die Behauptung $U \ge L$. Wir zeigen nun $A_{i\cdot}(\bar{x} + \lambda c) \le b_i$ für alle $\lambda \in [L,U]$, $i \in P \cup N \cup Z$.

Ist $i \in Z$, so gilt mit $p(j) = i$: $D_{j\cdot} = A_{i\cdot}$, $d_j = b_i$. Aus $A_{i\cdot}\, c = 0$ folgt

$$A_{i\cdot}(\bar{x} + \lambda c) \;=\; A_{i\cdot}\,\bar{x} \;=\; D_{j\cdot}\,\bar{x} \;\le\; d_j \;=\; b_i.$$

Ist $i \in P$, dann ist $U < +\infty$ und

$$\begin{aligned} A_{i\cdot}(\bar{x} + \lambda c) \;&=\; A_{i\cdot}\,\bar{x} \,+\, \lambda A_{i\cdot}\, c \;\le\; A_{i\cdot}\,\bar{x} \,+\, U A_{i\cdot}\, c \\ &\le\; A_{i\cdot}\,\bar{x} \,+\, \lambda_i A_{i\cdot}\, c \;=\; b_i \end{aligned}$$

Ist $i \in N$, so folgt die Behauptung entsprechend.

(d2) Sei $\bar{x} + \lambda c \in P(A,b)$. Angenommen, es gilt $\lambda \notin [L,U]$, wobei o.B.d.A. $\lambda < L$ angenommen werden kann. Dann gilt für $i \in N$ mit $\lambda_i > \lambda$:

$$A_{i\cdot}(\bar{x}+\lambda c) \;=\; A_{i\cdot}\,\bar{x} + \lambda A_{i\cdot}\, c \;>\; A_{i\cdot}\,\bar{x} + \lambda_i A_{i\cdot}\, c \;=\; b_i.$$

Dies ist ein Widerspruch. Es bleibt noch $\bar{x} \in P(D,d)$ zu zeigen.
Aus (a) und $A(\bar{x} + \lambda c) \le b$ folgt $D(\bar{x} + \lambda c) \le d$.
Nach (b) ist $Dc = 0$ und somit $D(\bar{x} + \lambda c) \;=\; D\bar{x} \;\le\; d$, also ist $\bar{x} \in P(D,d)$.

Ein wichtiger Spezialfall von Algorithmus **??** ist der

Fourier-Motzkin-Elimination (der j-ten Variable)

Input: Wie in Algorithmus **??**, jedoch mit folgenden Änderungen:
- $c = e_j$
- $H = \{x \in \mathbb{K}^n \mid x_j = 0\} = \{x \in \mathbb{K}^n \mid c^T x = 0\}$

Output: Wie in Algorithmus **??**, wobei gilt:

$$\{x \in \mathbb{K}^n \mid Dx \leq d,\, x_j = 0\}$$

ist die orthogonale Projektion von $P(A, b)$ auf H.

(1) Spezialfall von **??**:

$$\begin{aligned} N &= \{i \in M \mid A_{ij} < 0\}, \\ Z &= \{i \in M \mid A_{ij} = 0\}, \\ P &= \{i \in M \mid A_{ij} > 0\}. \end{aligned}$$

(2) Wie in **??**.

(3) Für $i \in \{1, 2, \ldots, r\}$ führe aus:
   (a) Wie in **??**,
   (b) wie in **??** mit

$$\begin{aligned} D_{i\cdot} &= a_{tj} A_{s\cdot} - a_{sj} A_{t\cdot}, \\ d_i &= a_{tj} b_s - a_{sj} b_t. \end{aligned}$$

(4) wie in **??**.

Mit Hilfe der Fourier-Motzkin-Elimination (FME) erhalten wir einen alternativen Beweis zum Farkas-Lemma (Satz **??**). Bevor wir diesen durchführen können, benötigen wir noch eine Folgerung aus Satz **??**.

Korollar 5.48. *Es gilt:* $P(A, b) \neq \emptyset \iff P(D, d) \neq \emptyset$.

Beweis. Die Projektion von $P(A, b)$ entlang c auf sich selbst ist $P(A, b)$. Mit Satz **??** (c) gilt damit $P(A, b) = P(A, b) \cap P(D, d)$. Gilt nun $P(D, d) = \emptyset$, so folgt $P(A, b) = \emptyset$. Ist dagegen $P(D, d) \neq \emptyset$, so ergibt Satz **??** (d1) $P(A, b) \neq \emptyset$.

Wenden wir nun die Fourier-Motzkin-Elimination auf die erste Variable (d.h. entlang e_1) an, so gilt mit Satz **??** (a), (b) und Folgerung **??**:

(i) $D^1 e_1 = 0$.

(ii) Es existiert eine Matrix $U^1 \geq 0$ mit $U^1 A = D^1$, $U^1 b = d^1$. (5.14)

(iii) $P(A,b) = \emptyset \iff P(D^1, d^1) = \emptyset$.

Entsprechend können wir die Fourier-Motzkin-Elimination fortsetzen und die zweite Variable eliminieren. Wir erhalten im zweiten Schritt eine Matrix D^2 mit:

(i) $D^2 e_2 = 0$.

(ii) Es gibt eine Matrix $\bar{U}^2 \geq 0$ mit $\bar{U}^2 D^1 = D^2$ und $\bar{U}^2 d^1 = d^2$. (5.15)

(iii) $P(D^1, d^1) = \emptyset \iff P(D^2, d^2) = \emptyset$.

Aus (**??**) (ii) und (**??**) (i) folgt

$$D^2 e_1 = \bar{U}^2 D^1 e_1 = \bar{U}^2 0 = 0,$$

und mit $U^2 = \bar{U}^2 U^1$ folgt aus (**??**) (ii) und (**??**) (ii)

$$U^2 A = \bar{U}^2 U^1 A = \bar{U}^2 D^1 = D^2,$$

d.h. es gilt:

(i) $D^2 e_1 = 0\,, \quad D^2 e_2 = 0$.

(ii) Es existiert eine Matrix $U^2 \geq 0$ mit $U^2 A = D^2$, $U^2 b = d^2$. (5.16)

(iii) $P(A,b) = \emptyset \iff P(D^2, d^2) = \emptyset$.

Dieser Prozess lässt sich fortsetzen und nach n-maliger Projektion erhalten wir:

(i) $D^n e_j = 0 \quad \forall\, j \in \{1, 2, \ldots, n\}$.

(ii) Es existiert eine Matrix $U^n \geq 0$ mit $U^n A = D^n$, $U^n b = d^n$. (5.17)

(iii) $P(A,b) = \emptyset \iff P(D^n, d^n) = \emptyset$.

Aus (**??**) (i) folgt $D^n = 0$. Damit gilt auch $P(D^n, d^n) \neq \emptyset \iff d^n \geq 0$. Gibt es ein i mit $d_i^n < 0$, so existiert nach (**??**) (ii) ein Vektor $u \geq 0$ mit $u^T A = D_{i\cdot}^n = 0$ und $u^T b = d_i^n < 0$. Mit (**??**) (iii) hat damit genau eines der beiden folgenden Gleichungssysteme eine Lösung (vgl. Folgerung **??**):

$$Ax \leq b \qquad \dot{\vee} \qquad \begin{aligned} u^T A &= 0 \\ u^T b &< 0 \\ u &\geq 0. \end{aligned}$$

Hier folgen noch ein paar Konsequenzen aus Satz **??** und Algorithmus **??**.

Korollar 5.49.

(a) Ist $H = P(A', b')$ ein Polyeder, dann ist die Projektion von $P(A, b)$ auf H entlang c ein Polyeder.
(b) Die Projektion eines Polyeders $P(A, b) \subseteq \mathbb{K}^n$ auf $\mathbb{K}^k$ mit $k \leq n$ ist ein Polyeder.

Beweis. (a) Nach Satz **??** (c) gilt

$$P_H = P(A', b') \cap P(D, d), \text{ d.h. } P_H = P\left(\begin{pmatrix} A' \\ D \end{pmatrix}, \begin{pmatrix} b' \\ d \end{pmatrix}\right).$$

(b) Folgt direkt aus (a), da $\mathbb{K}^k$ ein Polyeder ist.

Anmerkung 5.50. Die Projektion Q eines Polyeders $P(A, b) \subseteq \mathbb{K}^n$ auf $\mathbb{K}^k$ mit $k \leq n$ wird beschrieben durch

$$Q = \left\{ x \in \mathbb{K}^k \;\middle|\; \exists\, y \in \mathbb{K}^r \text{ mit } \begin{pmatrix} x \\ y \end{pmatrix} \in P(A, b) \right\},$$

wobei $k+r = n$ gilt und o.B.d.A. x zu den ersten k Spalten von A gehört.

Beweis. Vergleiche Definition **??** und die Fourier-Motzkin-Elimination.

Mit Hilfe der Projektion wollen wir noch einige Operationen nachweisen, die Polyeder erzeugen oder Polyeder in Polyeder überführen.

Altes Kommando `df` hier verwendet

Eine affine Abbildung $f : \mathbb{K}^n \to \mathbb{K}^k$ ist gegeben durch eine Matrix $D \in \mathbb{K}^{k \times n}$ und einen Vektor $d \in \mathbb{K}^k$, so dass $f(x) = Dx + d$ für alle $x \in \mathbb{K}^n$ gilt.

Theorem 5.51. *Affine Bilder von Polyedern sind Polyeder.*

Beweis. Sei $P = P(A,b) \subseteq \mathbb{K}^n$ ein Polyeder und $f : \mathbb{K}^n \to \mathbb{K}^k$ mit $f(x) = Dx + d$ eine affine Abbildung. Dann gilt

$$\begin{aligned} f(P) &= \{y \in \mathbb{K}^k \mid \exists\, x \in \mathbb{K}^n \text{ mit } Ax \leq b \text{ und } y = Dx + d\} \\ &= \left\{y \in \mathbb{K}^k \mid \exists\, x \in \mathbb{K}^n \text{ mit } B\begin{pmatrix} x \\ y \end{pmatrix} \leq \bar{b}\right\} \end{aligned}$$

mit

$$B = \begin{pmatrix} A & 0 \\ D & -I \\ -D & I \end{pmatrix} \qquad \text{und} \qquad \bar{b} = \begin{pmatrix} b \\ -d \\ d \end{pmatrix}.$$

Letzteres ist ein Polyeder nach Bemerkung **??** und Folgerung **??**.

Altes Kommando `df` hier verwendet

Korollar 5.52. *(Satz von Weyl)*
Für jede Matrix $A \in \mathbb{K}^{m\times n}$ gilt:

$$\left.\begin{matrix} \text{lin}(A) \\ \text{aff}(A) \\ \text{conv}(A) \\ \text{cone}(A) \end{matrix}\right\} \quad \textit{ist ein Polyeder.}$$

Beweis. Wir beweisen exemplarisch, dass cone(A) ein Polyeder ist, die anderen Fälle können analog dazu bewiesen werden. Es gilt:

$$\text{cone}(A) = \{x \in \mathbb{K}^m \mid \exists\, y \geq 0 \text{ mit } x = Ay\}.$$

Mit $P = P(-I, 0)$ und $f(x) = Ax$ gilt $f(P) = \text{cone}(A)$ und damit ist nach Satz **??** cone(A) ein Polyeder.

Korollar 5.53. *Die Summe $P_1 + P_2$ zweier Polyeder P_1 und P_2 ist wieder ein Polyeder.*

Beweis. Sei $P_i = P(A^i, b^i)\,,\ i = 1, 2$. Dann ist

$$\begin{aligned} P_1 + P_2 &= \{x_1 + x_2 \in \mathbb{K}^n \mid A^1 x_1 \leq b^1\,,\ A^2 x_2 \leq b^2\} \\ &= \{z \in \mathbb{K}^n \mid \exists\, x_1, x_2 \in \mathbb{K}^n :\ A^1 x_1 \leq b^1\,,\ A^2 x_2 \leq b^2\,,\ z = x_1 + x_2\}. \end{aligned}$$

Mit

$$P = P\left(\begin{pmatrix} A^1 & 0 \\ 0 & A^2 \end{pmatrix}\,,\ \begin{pmatrix} b^1 \\ b^2 \end{pmatrix}\right) \quad \text{und} \quad f\begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = Ix_1 + Ix_2$$

gilt $f(P) = P_1 + P_2$ und damit ist nach Satz **??** $f(P)$ wieder ein Polyeder.

Verbinden wir die beiden Folgerungen **??** und **??**, so erhalten wir

Korollar 5.54. *Es seien $A \in \mathbb{K}^{m\times n}$ und $B \in \mathbb{K}^{m\times n'}$. Dann gilt:*

$$P \;=\; \mathrm{conv}(A) + \mathrm{cone}(B) \qquad \textit{ist ein Polyeder.}$$

Folgerung **??** eröffnet uns eine andere Sichtweise auf Polyeder. Wir werden im folgenden Abschnitt zeigen, dass sich jedes Polyeder in der Form $\mathrm{conv}(A) + \mathrm{cone}(B)$ schreiben läßt. Dazu bedarf es jedoch einiger Vorbereitungen.

5.6 Darstellungssätze

Betrachten wir nochmals das Farkas-Lemma (Satz **??**) aus einem anderen Blickwinkel:

$$\exists\, x \geq 0 :\; Ax = b \qquad \dot{\bigvee} \qquad \exists\, y :\; y^T A \leq 0\,,\; y^T b > 0$$

oder anders ausgedrückt:

$$\exists\, x \geq 0 :\; Ax = b \qquad \Longleftrightarrow \qquad \forall\, y :\; A^T y \leq 0 \;\Rightarrow\; y^T b \leq 0.$$

Damit sind alle rechten Seiten b charakterisiert, für die das lineare Programm $Ax = b\,,\; x \geq 0$ eine Lösung hat. Nach Definition gilt

$$\mathrm{cone}(A) \;=\; \{b \in \mathbb{K}^m \;\big|\; \exists\, x \geq 0 \text{ mit } Ax = b\}.$$

Zusammen mit Satz **??** haben wir dann

Anmerkung 5.55. Für alle Matrizen $A \in \mathbb{K}^{m\times n}$ gilt

$$\mathrm{cone}(A) \;=\; \{b \in \mathbb{K}^m \;\big|\; y^T b \leq 0 \;\forall\, y \in P(A^T, 0)\}.$$

Die geometrische Interpretation von Bemerkung **??** könnte man in der folgenden Form schreiben:

Die zulässigen rechten Seiten b von $Ax = b,\, x \geq 0$ $\;\hat{=}\;$ Die Vektoren, die mit allen Vektoren aus $P(A^T, 0)$ einen stumpfen Winkel bilden.

Als Verallgemeinerung geben wir

Definition 5.56. *Sei $S \subseteq \mathbb{K}^n$ eine beliebige Menge.*

Kommando
er verwendet

(a) Die Menge aller Vektoren, die einen stumpfen Winkel mit allen Vektoren aus S bilden, heißt polarer Kegel. In Zeichen:

$$S^\circ \;=\; \{y \in \mathbb{K}^n \;\big|\; y^T x \le 0 \;\forall\, x \in S\}.$$

Für eine Matrix A bedeutet A° die Menge $\{y \in \mathbb{K}^n \,|\, y^T A \le 0\,\}$.

(b) Das orthogonale Komplement von S (vgl. die Lineare Algebra) ist die Menge aller Vektoren, die auf allen Vektoren aus S senkrecht stehen. In Zeichen:

$$S^\perp \;:=\; \{y \in \mathbb{K}^n \;\big|\; y^T x = 0 \;\forall\, x \in S\}.$$

Altes Kommando
`df` hier verwendet

Offensichtlich gilt $S^\perp \subseteq S^\circ$. Bemerkung **??** lässt sich nun in der folgenden Form schreiben.

Korollar 5.57. *Für alle Matrizen $A \in \mathbb{K}^{m\times n}$ gilt*

$$P(A,0)^\circ \;=\; \operatorname{cone}(A^T).$$

Beispiel 5.58. Sei

$$A \;=\; \begin{pmatrix} -3 & 2 \\ 1 & -2 \end{pmatrix}.$$

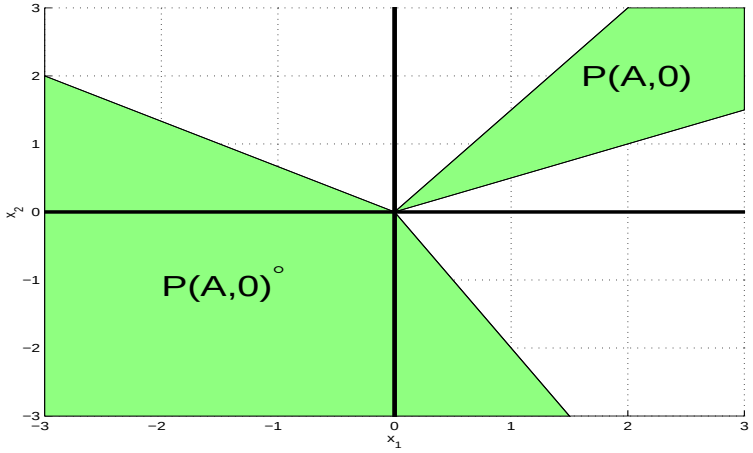


Abb. 5.7. Zu Beispiel **??**: $P(A,0)^\circ = \operatorname{cone}\left(\{\binom{-3}{2}, \binom{1}{-2}\}\right)$.

Es gilt: $Ax = b,\ x \ge 0$ ist lösbar $\iff$ $b \in P(A^T,0)^\circ$
$\iff$ $b \in \operatorname{cone}(A)$

Zum Beispiel ist $\begin{cases} Ax = \binom{0}{1},\ x \geq 0 & \text{nicht lösbar.} \\ Ax = \binom{-1}{0},\ x \geq 0 & \text{lösbar.} \end{cases}$

Wir schreiben im Weiteren kurz $S^{\circ\circ} = (S^\circ)^\circ$.

Lemma 5.59. *Für S, $S_i \subseteq \mathbb{K}^n$, $i \in \{1, 2, \ldots, k\}$ gilt:*

(a) $S_i \subseteq S_j \implies S_j^\circ \subseteq S_i^\circ$.
(b) $S \subseteq S^{\circ\circ}$.
(c) $\left(\bigcup_{i=1}^{k} S_i\right)^\circ = \bigcap_{i=1}^{k} S_i^\circ$.
(d) $S^\circ = \text{cone}(S^\circ) = (\text{cone}(S))^\circ$.
(e) $S = \text{lin}(S) \implies S^\circ = S^\perp$.

Beweis. Übung.

Korollar 5.60.

$$(\text{cone}(A^T))^\circ = \text{cone}(A^T)^\circ = P(A, 0).$$

Beweis. Es gilt:

$$(\text{cone}(A^T))^\circ \overset{??(d)}{=} \text{cone}(A^T)^\circ \overset{??(d)}{=} (A^T)^\circ \overset{??(a)}{=} \{x \mid Ax \leq 0\} = P(A, 0).$$

Altes Kommando `df` hier verwendet

Theorem 5.61. *(Polarensatz)*
Für jede Matrix A gilt

$$P(A, 0)^{\circ\circ} = P(A, 0),$$
$$\text{cone}(A)^{\circ\circ} = \text{cone}(A).$$

Beweis. Es gilt

$$P(A, 0) \overset{??}{=} \text{cone}(A^T)^\circ \overset{??}{=} P(A, 0)^{\circ\circ},$$

$$\text{cone}(A) \overset{??}{=} P(A^T, 0)^\circ \overset{??}{=} (\text{cone}(A)^\circ)^\circ \overset{??}{=} \text{cone}(A)^{\circ\circ}.$$

Damit haben wir alle Hilfsmittel zusammen, um zu zeigen, dass Polyeder nicht nur in der Form $P(A, b)$ dargestellt werden können. Genau dies besagt der

Altes Kommando `df` hier verwendet

Theorem 5.62. *(Satz von Minkowski)*
Eine Teilmenge $K \subseteq \mathbb{K}^n$ ist genau dann ein polyedrischer Kegel, wenn K die konische Hülle endlich vieler Vektoren ist, d.h. zu jeder Matrix $A \in \mathbb{K}^{m \times n}$ gibt es eine Matrix $B \in \mathbb{K}^{n \times d}$ mit

$$P(A, 0) = \text{cone}(B)$$

und umgekehrt.

Beweis. Es gilt:

$$P(A,0) \overset{??}{=} \text{cone}(A^T)^\circ \overset{??}{=} P(B^T,0)^\circ \overset{??}{=} \text{cone}(B).$$

Theorem 5.63. *Es seien $A \in \mathbb{K}^{m\times n}$, $b \in \mathbb{K}^m$. Dann existieren endliche Mengen $V, E \subseteq \mathbb{K}^n$ mit*

$$P(A,b) = \text{conv}(V) + \text{cone}(E).$$

Beweis. Setze

$$H = P\left(\begin{pmatrix} A & -b \\ 0^T & -1 \end{pmatrix}, \begin{pmatrix} 0 \\ 0 \end{pmatrix}\right).$$

Dann gilt: $x \in P(A,b) \Leftrightarrow \binom{x}{1} \in H$.

H ist ein polyedrischer Kegel, also gibt es nach Satz **??** eine Matrix $B \in \mathbb{K}^{(n+1)\times d}$ mit $H = \text{cone}(B)$. Aufgrund der Definition von H (vgl. die letzte Zeile) hat die letzte Zeile von B nur nicht-negative Einträge. Durch Skalierung und Vertauschung der Spalten von B können wir B in eine Matrix $\bar{B}$ überführen, für die dann gilt:

$$\bar{B} = \begin{pmatrix} V & E \\ \mathbb{1}^T & 0^T \end{pmatrix} \quad \text{mit } \text{cone}(\bar{B}) = H.$$

Damit gilt nun

$$\begin{aligned} x \in P(A,b) &\iff \begin{pmatrix} x \\ 1 \end{pmatrix} \in H \\ &\iff \begin{array}{l} x = V\cdot\lambda + E\cdot\mu \\ 1 = \mathbb{1}^T\cdot\lambda \end{array} \quad \lambda,\mu \geq 0 \\ &\iff x \in \text{conv}(V) + \text{cone}(E). \end{aligned}$$

Korollar 5.64. *Eine Teilmenge $P \subseteq \mathbb{K}^n$ ist genau dann ein Polytop, wenn P die konvexe Hülle endlich vieler Vektoren ist.*

Beweis. Sei $V \subseteq \mathbb{K}^n$ endlich und $P = \text{conv}(V)$, dann ist P nach Folgerung **??** ein Polyeder. Ist $x \in P$, so gilt

$$x = \sum_{i=1}^{k} \lambda_i v_i \text{ mit } v_i \in V,\ \lambda_i \geq 0,\ \sum_{i=1}^{k} \lambda_i = 1$$

und somit

$$||x|| \leq \sum_{i=1}^{k} ||v_i|| \quad \text{also} \quad P \subseteq \left\{ x \in \mathbb{K}^n \;\middle|\; ||x|| \leq \sum_{v\in V} ||v|| \right\}.$$

Also ist P beschränkt, d.h. P ist ein Polytop.

Umgekehrt, ist P ein Polytop, so gibt es nach Satz **??** endliche Mengen V, E mit $P = \text{conv}(V) + \text{cone}(E)$. Angenommen, es existiert ein $e \in E$ mit $e \neq 0$, so gilt $x + \nu e \in P$ für alle $\nu \in \mathbb{N}$ und alle $x \in \text{conv}(V)$. Demnach ist P unbeschränkt, falls $E \setminus \{0\} \neq \emptyset$. Daher muss $E \in \{\emptyset, \{0\}\}$ gelten und damit

$$\text{conv}(V) \;=\; \text{conv}(V) + \text{cone}(E) \;=\; P.$$

Altes Kommando `df` hier verwendet

Theorem 5.65. *(Darstellungssatz)*
Eine Teilmenge $P \subseteq \mathbb{K}^n$ ist genau dann ein Polyeder, wenn P die Summe eines Polytops und eines polyedrischen Kegels ist, d.h. wenn es endliche Mengen V, $E \subseteq \mathbb{K}^n$ gibt mit

$$P \;=\; \text{conv}(V) + \text{cone}(E).$$

Beweis. Kombiniere

Satz **??**: polyedrischer Kegel $= \text{cone}(E)$,
Satz **??**: $P(A, b) \;=\; \text{conv}(V) + \text{cone}(E)$,
Folgerung **??**: Polytop $= \text{conv}(V)$,
Folgerung **??**: $\text{conv}(V) + \text{cone}(E)$ ist ein Polyeder.

Damit kennen wir jetzt zwei Darstellungsformen für Polyeder

Altes Kommando `df` hier verwendet

(1) Die äußere Beschreibung: $P(A, b)$

$$P(A, b) \;=\; \bigcap_{i=1}^{m} \{x \in \mathbb{K}^n \;\big|\; A_{i\cdot}\, x \;\leq\; b_i\} \;\subseteq\; \{x \;\big|\; A_{i\cdot}\, x \;\leq\; b_i\},$$

d.h. $P(A, b)$ wird als Durchschnitt von größeren Mengen (Halbräumen) betrachtet.

Altes Kommando `df` hier verwendet

(2) Die innere Beschreibung: $\text{conv}(V) + \text{cone}(E)$
Ist $E = \emptyset$, so ist die Bezeichnung offensichtlich, denn es gilt $V \subseteq P$ und somit wird P durch eine konvexe Hüllenbildung aus Elementen von sich selbst erzeugt. Analoges gilt, wenn P ein polyedrischer Kegel ist. Gilt $V \neq \emptyset$ und $E \neq \emptyset$, so ist E nicht notwendigerweise Teilmenge von P, jedoch gelingt es immer, P durch Vektoren $v \in V$ und $e \in E$ „von innen heraus“ zu konstruieren, vgl. Abbildung **??**.

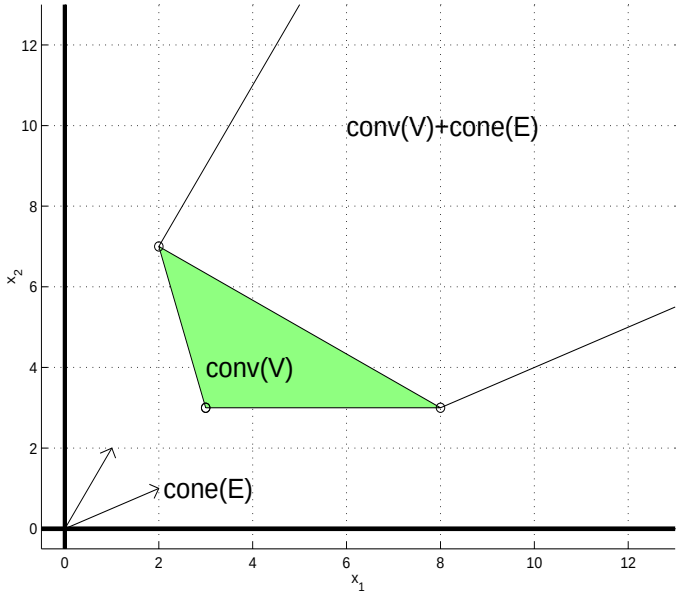


Abb. 5.8. innere Konstruktion eines Polyeders.

5.7 Minimale konvexe Erzeugendensysteme

Altes Kommando `df` hier verwendet

In Folgerung **??** haben wir gesehen, dass Polytope genau diejenigen Teilmengen des $\mathbb{K}^n$ sind, die durch Konvexkombinationen einer endlichen Menge erzeugt werden.

Vergleiche hierzu aus der Linearen Algebra die linearen Teilräume, die durch Linearkombinationen der Elemente einer endlichen Menge erzeugt werden. Ähnlich wie in der Linearen Algebra werden wir sehen, dass es für Polytope minimale Erzeugendensysteme (Basen) gibt und dass diese im Falle von Polytopen sogar eindeutig bestimmt sind, d.h. es existiert eine eindeutig bestimmte endliche Menge V mit $P = \text{conv}(V)$.

Im Falle von polyedrischen Kegeln wird diese Eindeutigkeit nur unter zusätzlichen Voraussetzungen gelten, wobei Eindeutigkeit natürlich nur bis auf eine Multiplikation mit einem positiven Skalar gelten kann.

Um diese Aussagen zu beweisen, benötigen wir noch ein wenig mehr Handwerkszeug. Zunächst suchen wir eine Charakterisierung der γ-Polare P^γ eines Polyeders P. Wir erinnern uns an Definition **??**:

$$P^\gamma \;=\; \left\{ \begin{pmatrix} a \\ \alpha \end{pmatrix} \in \mathbb{K}^{n+1} \;\middle|\; a^T x \;\le\; \alpha \;\forall\, x \in P \right\}.$$

Es lässt sich dann der folgende Satz beweisen.

Theorem 5.66. *Es sei $\emptyset \neq P \subseteq \mathbb{K}^n$ ein Polyeder mit den Darstellungen*

$$P \;=\; P(A,b) \;=\; \mathrm{conv}(V) \,+\, \mathrm{cone}(E).$$

Dann gilt:

(a)
$$P^\gamma \;=\; \left\{ \begin{pmatrix} a \\ \alpha \end{pmatrix} \in \mathbb{K}^{n+1} \;\middle|\; \exists\, u \ge 0 :\; u^T A = a^T \,,\; u^T b \le \alpha \right\}$$
$$\;=\; \mathrm{cone}\left(\begin{pmatrix} A^T & 0 \\ b^T & 1 \end{pmatrix} \right).$$

(b)
$$P^\gamma \;=\; \left\{ \begin{pmatrix} a \\ \alpha \end{pmatrix} \in \mathbb{K}^{n+1} \;\middle|\; \begin{pmatrix} V^T & -\mathbb{1} \\ E^T & 0 \end{pmatrix} \begin{pmatrix} a \\ \alpha \end{pmatrix} \;\le\; 0 \right\}$$
$$\;=\; P\left(\begin{pmatrix} V^T & -\mathbb{1} \\ E^T & 0 \end{pmatrix} , \begin{pmatrix} 0 \\ 0 \end{pmatrix} \right).$$

Beweis. (a)
$$\begin{pmatrix} a \\ \alpha \end{pmatrix} \in P^\gamma \quad\Longleftrightarrow\quad Ax \le b \,,\; a^T x > \alpha \text{ hat keine Lösung}$$
$$\overset{\substack{?? \\ P \neq \emptyset}}{\Longleftrightarrow} \quad \exists\, z \ge 0 \,,\; y > 0 \text{ mit } z^T A - y a^T = 0 \,,\; z^T b - y\alpha \le 0$$
$$\Longleftrightarrow \quad \exists\, z \ge 0 \text{ mit } z^T A = a^T \,,\; z^T b \le \alpha$$
$$\Longleftrightarrow \quad \exists\, z \ge 0 \,,\; \lambda \ge 0 \text{ mit } \begin{pmatrix} a \\ \alpha \end{pmatrix} = \begin{pmatrix} A^T & 0 \\ b^T & 1 \end{pmatrix} \begin{pmatrix} z \\ \lambda \end{pmatrix}$$
$$\Longleftrightarrow \quad \begin{pmatrix} a \\ \alpha \end{pmatrix} \;\in\; \mathrm{cone}\left(\begin{pmatrix} A^T & 0 \\ b^T & 1 \end{pmatrix} \right).$$

(b)
$$\begin{pmatrix} a \\ \alpha \end{pmatrix} \in P^\gamma \quad\Longrightarrow\quad a^T v \le \alpha \text{ für alle } v \in V \text{ und}$$
$$a^T(v + \lambda e) \le \alpha \text{ für alle } v \in V \,,\; e \in E \,,\; \lambda \ge 0$$
$$\Longrightarrow \quad a^T e \le 0 \,, \text{ denn andernfalls wäre } a^T(v + \lambda e) > \alpha$$
$$\text{für ein genügend großes } \lambda$$
$$\Longrightarrow \quad \begin{pmatrix} V^T & -\mathbb{1} \\ E^T & 0 \end{pmatrix} \begin{pmatrix} a \\ \alpha \end{pmatrix} \;\le\; \begin{pmatrix} 0 \\ 0 \end{pmatrix}.$$

Gilt umgekehrt

$$\begin{pmatrix} a \\ \alpha \end{pmatrix} \in P\left(\begin{pmatrix} V^T & -\mathbb{1} \\ E^T & 0 \end{pmatrix} , \begin{pmatrix} 0 \\ 0 \end{pmatrix} \right)$$

und ist $x \in P$, so existieren $v_1, v_2, \ldots, v_p \in V \,, \quad e_1, e_2, \ldots, e_q \in E$ und $\lambda_1, \lambda_2, \ldots, \lambda_p \ge 0$ mit $\sum_{i=1}^{p} \lambda_i = 1$ und $\mu_1, \mu_2, \ldots, \mu_q \ge 0$ mit

$$x \;=\; \sum_{i=1}^{p} \lambda_i v_i \;+\; \sum_{j=1}^{q} \mu_j e_j.$$

Damit gilt nun

$$a^T x \;=\; \sum_{i=1}^{p} \lambda_i a^T v_i + \sum_{j=1}^{q} \mu_j a^T e_j \;\leq\; \sum_{i=1}^{p} \lambda_i \alpha + \sum_{j=1}^{q} \mu_j \cdot 0 \;=\; \alpha.$$

Demnach ist $\begin{pmatrix} a \\ \alpha \end{pmatrix} \in P^\gamma$.

Korollar 5.67. *Die γ-Polare eines Polyeders $\emptyset \neq P \subseteq \mathbb{K}^n$ ist ein polyedrischer Kegel im $\mathbb{K}^{n+1}$.*

Korollar 5.68. *Ist $\emptyset \neq P = P(A,b) = \operatorname{conv}(V) + \operatorname{cone}(E)$ ein Polyeder und $a^T x \leq \alpha$ eine Ungleichung, dann sind folgende Aussagen äquivalent:*

(i) $a^T x \leq \alpha$ *ist gültig für* P.
(ii) $\exists\, u \geq 0$ *mit* $u^T A = a^T$, $u^T b \leq \alpha$.
(iii) $a^T v \leq \alpha \;\forall\, v \in V$ *und* $a^T e \leq 0 \;\forall\, e \in E$.
(iv) $\binom{a}{\alpha} \in P^\gamma$.

Im Folgenden benötigen wir noch einige weitere Begriffe, die als Hilfsmittel zur Vereinfachung der später zu führenden Beweise gebraucht werden.

Definition 5.69. *Sei $S \subseteq \mathbb{K}^n$ eine Menge. Wir definieren:*

(a) Die Menge $\operatorname{rec}(S) = \{y \in \mathbb{K}^n \mid \exists\, x \in S$ *mit* $x + \lambda y \in S \;\forall\, \lambda \geq 0\}$ *heißt Rezessionskegel von S.*
(b) Die Menge $\operatorname{lineal}(S) = \{y \in \mathbb{K}^n \mid \exists\, x \in S$ *mit* $x + \lambda y \in S \;\forall\, \lambda \in \mathbb{K}\}$ *heißt Linealitätsraum von S.*
(c) Die Menge $\operatorname{hog}(S) = \left\{\binom{x}{1} \in \mathbb{K}^{n+1} \mid x \in S\right\}^{\circ\circ}$ *heißt Homogenisierung von S.*

Altes Kommando `df` hier verwendet

Altes Kommando `df` hier verwendet

Altes Kommando `df` hier verwendet

Wir wollen nun kurz die in Definition **??** eingeführten Mengen für Polyeder charakterisieren.

Theorem 5.70. *Sei $P = P(A,b) = \operatorname{conv}(V) + \operatorname{cone}(E)$ ein nichtleeres Polyeder. Dann gilt*

$$\operatorname{rec}(P) \;=\; P(A,0) \;=\; \operatorname{cone}(E).$$

Beweis. Offensichtlich gilt $P(A,0) = \operatorname{cone}(E)$, was an folgenden Inklusionen zu sehen ist:

„$\supseteq$“: vgl. Folgerung **??** (iii).

„$\subseteq$“: $x \in P(A,0) \;\Rightarrow\; y + \lambda x \in P(A,b) \;\; \forall\, \lambda \geq 0, \; \forall\, y \in P(A,b)$

$\Rightarrow\; x \in \operatorname{cone}(E)$ nach Satz **??**.

Bleibt noch $\operatorname{rec}(P) = P(A,0)$ zu zeigen.

$\operatorname{rec}(P) \subseteq P(A,0)$:
Sei $y \in \operatorname{rec}(P)$. Dann existiert ein $x \in P$ mit $x + \lambda y \in P$ für alle $\lambda \geq 0$. Daraus folgt nun $b \geq A(x + \lambda y) = Ax + \lambda Ay$. Gäbe es eine Komponente von Ay, die größer als Null ist, z.B. $(Ay)_i$, so wäre

$$x + \bar{\lambda} y \quad \text{mit} \quad \bar{\lambda} = \frac{b_i - (Ax)_i}{(Ay)_i} + 1$$

nicht in $P(A,b)$, was ein Widerspruch ist. Also gilt $y \in P(A,0)$.

$P(A,0) \subseteq \operatorname{rec}(P)$:
Sei nun $y \in P(A,0)$, dann gilt für alle $x \in P(A,b)$ und $\lambda \geq 0$

$$A(x + \lambda y) \;=\; Ax \;+\; \lambda Ay \;\leq\; b \;+\; 0 \;=\; b,$$

also ist $y \in \operatorname{rec}(P)$.

Insbesondere folgt aus Satz **??**

$$\operatorname{rec}(P) \;=\; \{y \in \mathbb{K}^n \;\big|\; (\, x + \lambda y \in P \;\forall\, \lambda \geq 0) \;\forall\, x \in P\}.$$

Ist P ein Kegel, so gilt $P = \operatorname{rec}(P)$ und P ist genau dann ein Polytop, wenn $\operatorname{rec}(P) = \{0\}$ gilt. Aus Definition **??** folgt $\operatorname{lineal}(P) = \operatorname{rec}(P) \cap (-\operatorname{rec}(P))$. Offenbar ist $\operatorname{lineal}(P)$ ein linearer Teilraum des $\mathbb{K}^n$ und zwar ist es der größte Teilraum $L \subseteq \mathbb{K}^n$, so dass $x + L \subseteq P$ für alle $x \in P$ gilt. Weiterhin folgt

Theorem 5.71. *Sei* $\emptyset \neq P = P(A,b) = \operatorname{conv}(V) + \operatorname{cone}(E)$ *ein Polyeder, dann gilt*

$$\operatorname{lineal}(P) \;=\; \{x \in \mathbb{K}^n \;\big|\; Ax = 0\} \;=\; \operatorname{cone}(\{e \in E \;\big|\; -e \in \operatorname{cone}(E)\}).$$

Beweis. Wegen $\operatorname{lineal}(P) = \operatorname{rec}(P) \cap (-\operatorname{rec}(P))$ folgt die Behauptung direkt aus Satz **??**.

Die Definition von $\operatorname{hog}(S)$ erscheint etwas kompliziert, da man S zweimal polarisieren muss. Für Polyeder lässt sich $\operatorname{hog}(P)$ jedoch einfacher charakterisieren, wie der folgende Satz zeigt.

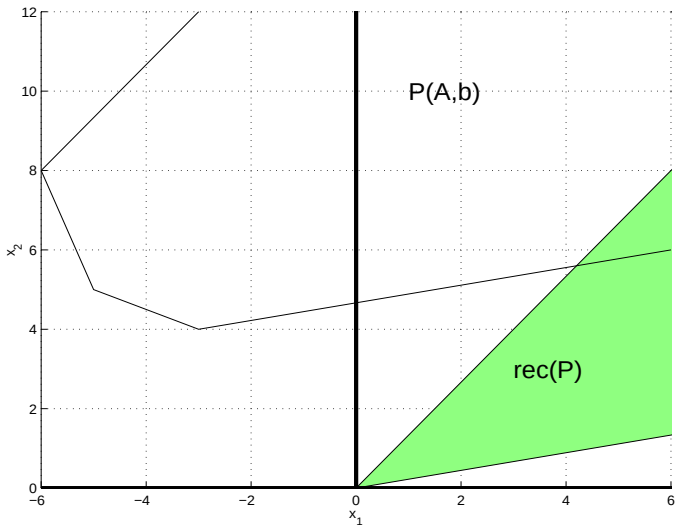


Abb. 5.9. Ein Polyeder P mit zugehörigem Rezessionskegel.

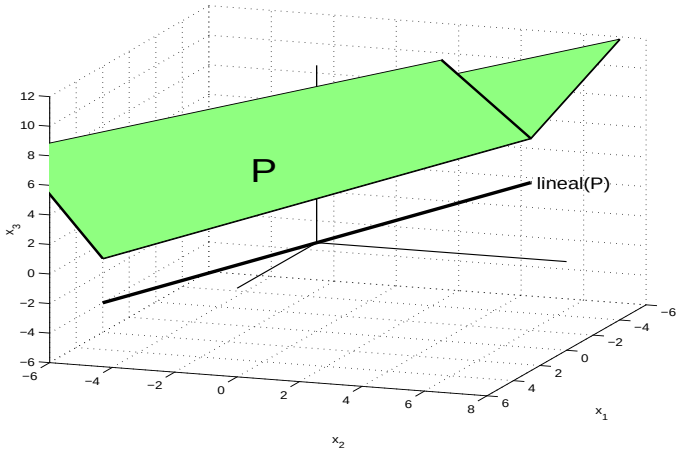


Abb. 5.10. Ein Polyeder P mit zugehörigem Linealitätsraum

Theorem 5.72. *Sei* $\emptyset \neq P = P(A,b) = \text{conv}(V) + \text{cone}(E)$ *ein Polyeder und*

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

Dann gilt

$$\operatorname{hog}(P) \;=\; P(B,0) \;=\; \operatorname{cone}\left(\left\{\begin{pmatrix} v \\ 1 \end{pmatrix} \;\middle|\; v \in V\right\}\right) + \operatorname{cone}\left(\left\{\begin{pmatrix} e \\ 0 \end{pmatrix} \;\middle|\; e \in E\right\}\right).$$

Beweis. Setzen wir

$$P_1 \;=\; \left\{\begin{pmatrix} x \\ 1 \end{pmatrix} \in \mathbb{K}^{n+1} \;\middle|\; x \in P\right\},$$

so gilt

$$P_1 \;=\; \operatorname{conv}\left(\left\{\begin{pmatrix} v \\ 1 \end{pmatrix} \;\middle|\; v \in V\right\}\right) \;+\; \operatorname{cone}\left(\left\{\begin{pmatrix} e \\ 0 \end{pmatrix} \;\middle|\; e \in E\right\}\right).$$

Aus Folgerung **??** (iii) ergibt sich dann

$$\begin{aligned} P_1^\circ &\overset{\text{Def}}{=} \{z \in \mathbb{K}^{n+1} \mid z^T u \le 0 \;\forall\, u \in P_1\} \\ &\overset{\text{??}\,(iii)}{=} \left\{z \in \mathbb{K}^{n+1} \;\middle|\; z^T \begin{pmatrix} v \\ 1 \end{pmatrix} \le 0 \;\forall\, v \in V \;\text{ und }\; z^T \begin{pmatrix} e \\ 0 \end{pmatrix} \le 0 \;\forall\, e \in E\right\} \\ &= \left\{z \in \mathbb{K}^{n+1} \;\middle|\; \begin{pmatrix} V^T & \mathbb{1} \\ E^T & 0 \end{pmatrix} z \;\le\; 0\right\} \\ &= P\left(\begin{pmatrix} V^T & \mathbb{1} \\ E^T & 0 \end{pmatrix},\; 0\right). \end{aligned}$$

Mit Folgerung **??** gilt weiter

$$\operatorname{hog}(P) \overset{\text{Def}}{=} P_1^{\circ\circ} \quad = \quad P\left(\begin{pmatrix} V^T & \mathbb{1} \\ E^T & 0 \end{pmatrix},\; 0\right)^\circ \quad \overset{\text{??}}{=} \quad \operatorname{cone}\left(\begin{pmatrix} V & E \\ \mathbb{1}^T & 0 \end{pmatrix}\right).$$

Die zweite Charakterisierung erhalten wir aus einer anderen Darstellung von P_1° unter Anwendung von Satz **??**. Demnach gilt

$$\begin{aligned} P_1^\circ &= \left\{\begin{pmatrix} a \\ \alpha \end{pmatrix} \in \mathbb{K}^{n+1} \;\middle|\; a^T x + \alpha \le 0 \;\forall\, x \in P\right\} \\ &= \left\{\begin{pmatrix} a \\ \alpha \end{pmatrix} \in \mathbb{K}^{n+1} \;\middle|\; a^T x \le -\alpha \;\forall\, x \in P\right\} \\ &= \left\{\begin{pmatrix} a \\ \alpha \end{pmatrix} \in \mathbb{K}^{n+1} \;\middle|\; \begin{pmatrix} a \\ -\alpha \end{pmatrix} \in P^\gamma\right\} \\ &\overset{\text{??}}{=} \left\{\begin{pmatrix} a \\ \alpha \end{pmatrix} \in \mathbb{K}^{n+1} \;\middle|\; \begin{pmatrix} a \\ -\alpha \end{pmatrix} \in \operatorname{cone}\left(\begin{pmatrix} A^T & 0 \\ b^T & 1 \end{pmatrix}\right)\right\} \\ &= \operatorname{cone}\left(\begin{pmatrix} A^T & 0 \\ -b^T & -1 \end{pmatrix}\right). \end{aligned}$$

Folgerung **??** impliziert schließlich

$$\operatorname{hog}(P) = P_1^{\circ\circ} \;=\; \left(\operatorname{cone}\left(\begin{pmatrix} A^T & 0 \\ -b^T & -1 \end{pmatrix}\right)\right)^{\circ} \quad \overset{??}{=} \quad P\left(\begin{pmatrix} A & -b \\ 0 & -1 \end{pmatrix},\, 0\right).$$

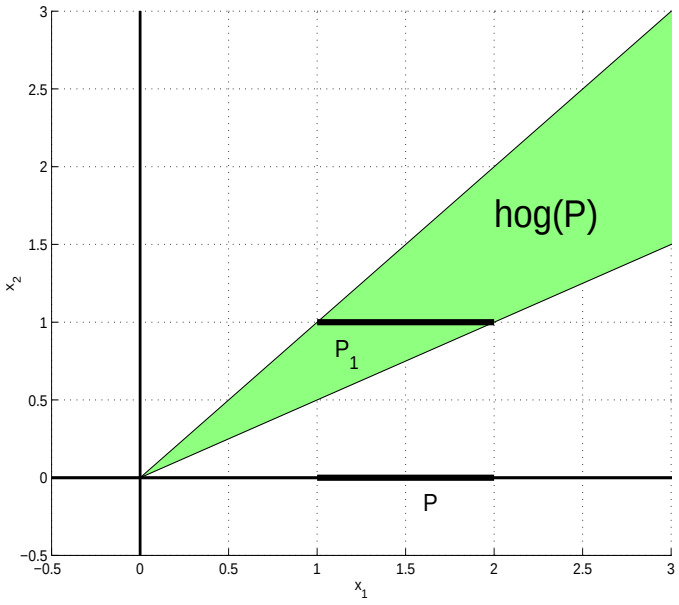


Abb. 5.11. Ein Polyeder P mit zugehöriger Homogenisierung.

Anmerkung 5.73. Sei $P \subseteq \mathbb{K}^n$ ein Polyeder. Dann gilt:

(a) $x \in P \qquad \Longleftrightarrow \begin{pmatrix} x \\ 1 \end{pmatrix} \in \operatorname{hog}(P).$

(b) $x \in \operatorname{rec}(P) \Longleftrightarrow \begin{pmatrix} x \\ 0 \end{pmatrix} \in \operatorname{hog}(P).$

Beweis. (a) „⇒“ folgt direkt aus der Definition **??** und Lemma **??** (b).
„⇐“ folgt direkt aus Satz **??**.

(b) Sei $P = P(A, b)$ eine Darstellung von P. Dann gilt nach Satz **??**

$$\operatorname{hog}(P) \;=\; P(B, 0) \quad \text{mit} \quad B \;=\; \begin{pmatrix} A & -b \\ 0 & -1 \end{pmatrix}.$$

Hieraus folgt zusammen mit Satz **??**

$$\begin{aligned}\begin{pmatrix} x \\ 0 \end{pmatrix} \in \hog(P) &\iff \begin{pmatrix} x \\ 0 \end{pmatrix} \in P(B,0) \\ &\iff Ax \leq 0 \\ &\iff x \in \rec(P).\end{aligned}$$

Im Folgenden konzentrieren wir uns auf Polyeder, die eine Ecke haben. Polyeder ohne Ecken brauchen nicht notwendigerweise eindeutige minimale Erzeugendensysteme zu haben. Zudem müssen minimale Erzeugendensysteme nicht einmal die gleiche Kardinalität besitzen.

Altes Kommando `df` hier verwendet

Definition 5.74. *Ein Polyeder heißt spitz, falls es eine Ecke besitzt.*

Theorem 5.75. *Sei $K \subseteq \mathbb{K}^n$ ein polyedrischer Kegel. Dann gilt:*

(a) Ist x eine Ecke von K, dann gilt $x = 0$.

(b) *F ist ein Extremalstrahl von K* $\iff$ *$\exists z \in \mathbb{K}^n \setminus \{0\}$, so dass $F = \cone(\{z\})$ eine Seitenfläche von K ist.*

Beweis. (a) Nach Bemerkung **??** existiert eine Matrix A mit $K = P(A,0)$. Die Menge F ist genau dann eine Seitenfläche von K, wenn eine Teilmenge $I \subseteq M$ existiert mit $F = \{x \in P \mid A_I.x \leq 0\}$. Jede nichtleere Seitenfläche von K ist daher ein Kegel, der den Nullvektor enthält. Damit gilt nun: Ist x eine Ecke, so ist $0 \in \{x\}$, also ist $x = 0$.

(b) „$\Leftarrow$“ Folgt aus Definition **??**.

„$\Rightarrow$“ Nach Definition **??** existiert ein $x \in \mathbb{K}^n$ und ein $z \in \mathbb{K}^n \setminus \{0\}$ mit $F = \{x\} + \cone(\{z\})$. Nach Bemerkung **??** gilt $0 \in F$ und damit existiert ein $\bar{\lambda} \geq 0$, $\bar{\lambda} \in \mathbb{K}$ mit $0 = x + \bar{\lambda}z$. Für $\bar{\lambda} = 0$ folgt sofort (b).
Sei also $\bar{\lambda} > 0$. Dann gilt $\{\lambda x \mid \lambda \geq 0\} \subseteq F$, denn $x \in F$ und K ist ein polyedrischer Kegel. Andererseits ist jedoch

$$\begin{aligned}\{\lambda x \mid \lambda \geq 0\} &= \{\lambda(-\bar{\lambda}z) \mid \lambda \geq 0\} \\ &\not\subseteq \{-\bar{\lambda}z\} + \{\lambda z \mid \lambda \geq 0\} = x + \cone(\{z\}) = F,\end{aligned}$$

also ein Widerspruch.

Theorem 5.76. *Sei $\emptyset \neq P = P(A,b) \subseteq \mathbb{K}^n$ ein Polyeder. Dann sind äquivalent:*

(1) P ist spitz.

(2) $\mathrm{Rang}(A) = n$.
(3) $\mathrm{rec}(P)$ *ist spitz, d.h.* 0 *ist eine Ecke von* $\mathrm{rec}(P)$.
(4) Jede nichtleere Seitenfläche von P *ist spitz.*
(5) $\mathrm{hog}(P)$ *ist spitz.*
(6) P *enthält keine Gerade.*
(7) $\mathrm{rec}(P)$ *enthält keine Gerade.*
(8) $\mathrm{lineal}(P) = \{0\}$.

Beweis. (1) $\Rightarrow$ (2): Ist x eine Ecke von P, so gilt nach Satz **??**

$$n \;=\; \mathrm{Rang}(A_{\mathrm{eq}(\{x\})\cdot}) \;\leq\; \mathrm{Rang}(A) \;\leq\; n$$

und demnach $\mathrm{Rang}(A) = n$.

(2) $\Rightarrow$ (1): Sei $x \in P$ so gewählt, dass $I = \mathrm{eq}(\{x\})$ maximal bezüglich der Mengeninklusion ist. Sei $F = \{y \in P \,\big|\, A_{I\cdot}\, y = b_I\}$, dann ist x ein innerer Punkt von F, vgl. Satz **??**. Würde $\mathrm{Rang}(A_{I\cdot}) < n$ gelten, dann enthielte der Kern von $A_{I\cdot}$ einen Vektor $y \neq 0$ und nach Lemma **??** gäbe es dann ein $\epsilon > 0$ mit $x \pm \epsilon y \in P$. Die Gerade $G = \{x + \lambda y \,\big|\, \lambda \in \mathbb{K}\}$ trifft mindestens eine der Hyperebenen $H_j = \{y \,\big|\, A_{j\cdot}\, y = b_j\}$, da $n = \mathrm{Rang}(A) > \mathrm{Rang}(A_{I\cdot})$ gilt (andernfalls wäre $A_{j\cdot}y = 0$ für alle $j \notin I$ und damit $n = \mathrm{Rang}(A) + \mathrm{Kern}(A) > \mathrm{Rang}(A) = n$). Also muss es ein $\delta \in \mathbb{K}$ geben, so dass $x + \delta y \in P$ und $I \subset \mathrm{eq}(\{x + \delta y\})$ ist. Dies ist jedoch ein Widerspruch zur Maximalität von I.
Aus der Äquivalenz von (1) und (2) folgt direkt die Äquivalenz mit (3), (4) und (5), denn es gilt:

$$\mathrm{rec}(P) \;=\; P(A, 0) \quad \text{nach Satz } \mathbf{??}.$$

$$\mathrm{hog}(P) \;=\; P\left(\begin{pmatrix} A & -b \\ 0 & -1 \end{pmatrix},\; \begin{pmatrix} 0 \\ 0 \end{pmatrix}\right) \quad \text{nach Satz } \mathbf{??}.$$

$$F \;=\; \{x \in \mathbb{K}^n \,\big|\, Ax \leq b\,,\; A_{\mathrm{eq}(F)\cdot}\, x \leq b_{\mathrm{eq}(F)}\,,\; -A_{\mathrm{eq}(F)\cdot}\, x \leq -b_{\mathrm{eq}(F)}\}$$
für alle Seitenflächen F von P.

(3) $\Rightarrow$ (6): Angenommen, P enthält eine Gerade $G = \{u + \lambda v \,\big|\, \lambda \in \mathbb{K}\; v \neq 0\}$, dann gilt

$$b \;\geq\; A(u + \lambda v) \;=\; Au \,+\, \lambda Av \quad \forall\, \lambda \in \mathbb{K}.$$

Daraus folgt $A(\lambda v) \leq 0$ für alle $\lambda \in \mathbb{K}$ und somit $v, -v \in \mathrm{rec}(P)$ nach Satz **??**, d.h. $0 = \frac{1}{2}v + \frac{1}{2}(-v)$ ist eine Konvexkombination der Null und damit ist Null keine Ecke von $\mathrm{rec}(P)$ nach Satz **??** (3).

(6) $\Rightarrow$ (3): Ist $\text{rec}(P)$ nicht spitz, so kann die Null nach Satz **??** (3)als echte Konvexkombination von Vektoren aus $\text{rec}(P)$ dargestellt werden:

$$0 \;=\; \lambda u \,+\, (1-\lambda)v\,,\; u \neq 0\,,\; v \neq 0\,,\; 0 < \lambda < 1.$$

Dann ist aber $G = \{\lambda u \,\big|\, \lambda \in \mathbb{K}\}$ eine Gerade in $\text{rec}(P)$, da $u, v \in \text{rec}(P)$ und $u \cong -v$, und für alle $x \in P$ ist $x + G$ eine Gerade in P, was ein Widerspruch ist.

Die Äquivalenz von (7) und (8) zu (1) – (6) zeigt man analog.

Aus Satz **??** können eine Reihe interessanter Konsequenzen gezogen werden, so hat $P^{=}(A, b)$ immer eine Ecke, sofern $P^{=}(A, b)$ nicht leer ist oder ein nichtleeres Polytop hat immer eine Ecke.

Unser Ziel ist es, minimale Erzeugendensysteme für Polyeder in der Darstellung $\text{conv}(V) + \text{cone}(E)$ zu gewinnen. Einen Bestandteil werden die Ecken bilden, die wir eben in Satz **??** charakterisiert haben. Den zweiten Bestandteil bilden die Extremalen, die wir im Folgenden näher betrachten wollen.

Altes Kommando `df` hier verwendet

Definition 5.77. *Sei $P = P(A, b)$ ein Polyeder. Ein Vektor $z \in \text{rec}(P)\backslash\{0\}$ heißt Extremale von P, falls $\text{cone}(\{z\})$ ein Extremalstrahl von $\text{rec}(P)$ ist.*

Nur spitze Polyeder haben Extremalen, denn wenn P nicht spitz ist, so ist nach Satz **??** $\text{rec}(P)$ nicht spitz, also existieren $0 \neq u, v \in \mathbb{K}^n$ mit

$$0 \;=\; \lambda u \,+\, (1-\lambda)v \qquad \text{mit } 0 < \lambda < 1.$$

Ist $F = \text{cone}(\{z\})$ ein Extremalstrahl von $\text{rec}(P)$, so gibt es eine gültige Ungleichung $c^T x \leq 0$ für $\text{rec}(P)$ mit $F = \{x \in \text{rec}(P) \,\big|\, c^T x = 0\}$ (vgl. Definition **??**).

Nun gilt

$$0 \;=\; c^T 0 \;=\; \lambda c^T u \,+\, (1-\lambda) c^T v \;\leq\; 0.$$

Da auch $c^T u \leq 0$ und $c^T v \leq 0$ gilt, folgt somit $c^T u = c^T v = 0$, also $u, v \in F$, was zum Widerspruch führt.

Aussagen über Extremalen haben also nur für spitze Polyeder einen Sinn.

Theorem 5.78. *Sei $P = P(A, b) \subseteq \mathbb{K}^n$ ein spitzes Polyeder und $z \in \text{rec}(P) \setminus \{0\}$. Dann sind äquivalent:*

(1) z ist eine Extremale von P.
(2) $\text{cone}(\{z\})$ *ist ein Extremalstrahl von* $\text{rec}(P)$.
(3) z lässt sich nicht als echte konische Kombination zweier linear unabhängiger Elemente von $\text{rec}(P)$ *darstellen.*
(4) $(\text{rec}(P) \setminus \text{cone}(\{z\})) \cup \{0\}$ *ist ein Kegel.*
(5) $\text{Rang}(A_{\text{eq}(\{z\})\cdot}) = n - 1$ (eq() *bzgl.* $Ax \leq 0$).

Beweis. (1)⇔(2): Definition **??**.
(3)⇔(4): Klar.
(2)⇒(3): Ist $F = \text{cone}(\{z\})$ ein Extremalstrahl von $\text{rec}(P)$, dann ist F eine eindimensionale Seitenfläche von $\text{rec}(P)$, d.h. F kann keine zwei linear unabhängige Vektoren enthalten. Außerdem gibt es ein $c \neq 0$ mit $F = \{x \in \text{rec}(P) \mid c^T x = 0\}$. Angenommen, es existieren $u, v \in \text{rec}(P)$, die linear unabhängig sind und $\lambda, \mu > 0$ mit $z = \lambda u + \mu v$, so gilt

$$0 \;=\; c^T z \;=\; \lambda c^T u + \mu c^T v \;\leq\; 0$$

und damit $c^T u = c^T v = 0$. Also gilt $u, v \in F$, was einen Widerspruch ergibt.
(3)⇒(5): Sei $I = \text{eq}(\{z\})$, dann ist z ein innerer Punkt von

$$F \;=\; \{x \in \text{rec}(P) \mid A_{I\cdot}\, x \;=\; 0\}.$$

Offenbar ist $\text{Rang}(A_{I\cdot}) \neq n$. Ist $\text{Rang}(A_{I\cdot}) < n - 1$, dann existiert ein $u \in \text{Kern}(A_{I\cdot})$ und u ist linear unabhängig von z. Nach Lemma **??** gibt es dann ein $\epsilon \in \mathbb{K}$, so dass $z \pm \epsilon u \in \text{rec}(P)$ gilt. Dann ist aber $z = \frac{1}{2}(z + \epsilon u) + \frac{1}{2}(z - \epsilon u)$ eine echte konische Kombination von zwei linear unabhängigen Elementen von $\text{rec}(P)$, woraus ein Widerspruch folgt.
(5)⇒(2): Sei $I = \text{eq}(\{z\})$ und $F = \{x \in \text{rec}(P) \mid A_{I\cdot}\, x = 0\}$. Nach Voraussetzung gilt $\dim(F) = 1$.
Da $\text{rec}(P)$ spitz ist, enthält $\text{rec}(P)$ nach Satz **??** keine Gerade, also muss die eindimensionale Seitenfläche F der Strahl $\text{cone}(\{z\})$ sein.

Theorem 5.79. *Sei $P \subseteq \mathbb{K}^n$ ein spitzes Polyeder. Dann gilt:*

(a) x *ist eine Ecke von* $P \iff \begin{pmatrix} x \\ 1 \end{pmatrix}$ *ist eine Extremale von* $\mathrm{hog}(P)$.

(b) z *ist eine Extremale von* $P \iff \begin{pmatrix} z \\ 0 \end{pmatrix}$ *ist eine Extremale von* $\mathrm{hog}(P)$.

Beweis. Sei $P = P(A, 0)$, dann gilt nach Satz **??**

$$\mathrm{hog}(P) \;=\; P(B, 0) \qquad \text{mit} \quad B \;=\; \begin{pmatrix} A & -b \\ 0 & -1 \end{pmatrix}.$$

(a) Sei $I = \mathrm{eq}(\{x\})$ bzgl. $P(A, b)$. Dann gilt

$$\begin{aligned} x \text{ ist eine Ecke von } P &\overset{??}{\iff} \mathrm{Rang}(A_{I\cdot}) \;=\; n \\ &\iff \mathrm{Rang}(B_{\mathrm{eq}(\{\binom{x}{1}\})\cdot}) \;=\; n \\ &\qquad \text{(denn die neu hinzukommende Ungleichung ist nicht mit Gleichheit erfüllt.)} \\ &\overset{??}{\iff} \begin{pmatrix} x \\ 1 \end{pmatrix} \text{ ist eine Extremale von } \mathrm{hog}(P). \end{aligned}$$

(b) Es gilt:

$$\begin{aligned} z \text{ ist Extremale von } P &\overset{??}{\iff} \mathrm{cone}(\{z\}) \text{ ist Extremalstrahl von } \mathrm{rec}(P) \\ &\overset{??}{\iff} \mathrm{Rang}(A_{\mathrm{eq}(\{z\})\cdot}) = n - 1 \\ &\iff \mathrm{Rang}(B_{\mathrm{eq}(\{\binom{z}{0}\})\cdot}) = n \\ &\overset{??}{\iff} \begin{pmatrix} z \\ 0 \end{pmatrix} \text{ ist eine Extremale von } \mathrm{hog}(P). \end{aligned}$$

Wir sind nun soweit, minimale Erzeugendensysteme für spitze Polyeder zu charakterisieren.

Altes Kommando `df` hier verwendet

Ist K ein polyedrischer Kegel und gilt $K = \mathrm{cone}(E)$, dann nennen wir E eine Kegelbasis von K, wenn es keine Menge $E' \subsetneq E$ gibt mit $K = \mathrm{cone}(E')$ und wenn für jede andere minimale Menge F mit $K = \mathrm{cone}(F)$ die Beziehung $|F| = |E|$ gilt.

Altes Kommando `df` hier verwendet

Ist P ein Polytop, dann heißt V mit $P = \mathrm{conv}(V)$ konvexe Basis von P, wenn es keine Menge $V' \subsetneq V$ gibt mit $P = \mathrm{conv}(V')$ und wenn für jede andere minimale Menge W mit $P = \mathrm{conv}(W)$ die Gleichung $|W| = |V|$ gilt.

Offensichtlich sind die Elemente einer Kegelbasis bzw. einer konvexen Basis konisch bzw. konvex unabhängig. Allerdings ist im Gegensatz zu Vektorraumbasen die konische (konvexe) Darstellung eines Elementes $x \in \text{cone}(E)$ ($y \in \text{conv}(V)$) durch Vektoren aus E (V) nicht immer eindeutig.

Theorem 5.80. *Sei $\{0\} \neq K \subseteq \mathbb{K}^n$ ein spitzer polyedrischer Kegel. Dann sind äquivalent:*

(1) E ist eine Kegelbasis von K.
(2) E ist eine Menge, die aus jedem Extremalstrahl von K genau einen von Null verschiedenen Vektor (also eine Extremale von K) auswählt.

Beweis. Ist z eine Extremale von K, so ist $K' = (K \setminus \text{cone}(\{z\})) \cup \{0\}$ nach Satz **??** ein Kegel. Folglich gilt $\text{cone}(E') \subseteq K'$ für alle Teilmengen E' von K'. Also muss jede Kegelbasis mindestens ein (aus der Basiseigenschaft folgt dann „genau ein“) Element von $\text{cone}(\{z\}) \setminus \{0\}$ enthalten.
Zum Beweis, dass (2) tatsächlich eine Kegelbasis ist, benutzen wir vollständige Induktion über $d = \dim(K)$.

$d = 1$: Ist klar.
$d \to d + 1$:
Sei K ein Kegel mit $\dim(K) = d+1$, $y \in K \setminus \{0\}$ beliebig und $c \in \mathbb{K}^n \setminus \{0\}$ ein Vektor, so dass die Ecke 0 von K die eindeutig bestimmte Lösung von $\max\{c^T x \mid x \in K\}$ ist (c existiert nach Satz **??**). Sei $z \in \{x \mid c^T x = 0\} \setminus \{0\}$. Dann ist für die Gerade $G = \{y + \lambda z \mid \lambda \in K\}$ die Menge $K \cap G$ ein endliches Streckenstück (andernfalls wäre $z \in \text{rec}(K) = K$ und wegen $c^T z = 0$ wäre 0 nicht die eindeutige Maximallösung). Folglich gibt es zwei Punkte z_1 und z_2, die auf echten Seitenflächen F_1 und F_2 von K liegen, so dass $K \cap G = \text{conv}(\{z_1, z_2\})$ ist. Es gilt $\dim(F_i) \leq d$ $(i = 1, 2)$, die F_i sind Kegel und die Extremalstrahlen von F_i sind Extremalstrahlen von K. Nach Induktionsvoraussetzung werden z_1 und z_2 durch die in Satz **??** (2) festgelegten Mengen bzgl. F_1 und F_2 konisch erzeugt. Daraus folgt, dass auch y durch die in Satz **??** (2) festgelegten Mengen konisch erzeugt wird.

Korollar 5.81. *Jeder spitze polyedrische Kegel besitzt eine (bis auf eine positive Skalierung der einzelnen Elemente) eindeutige Kegelbasis.*

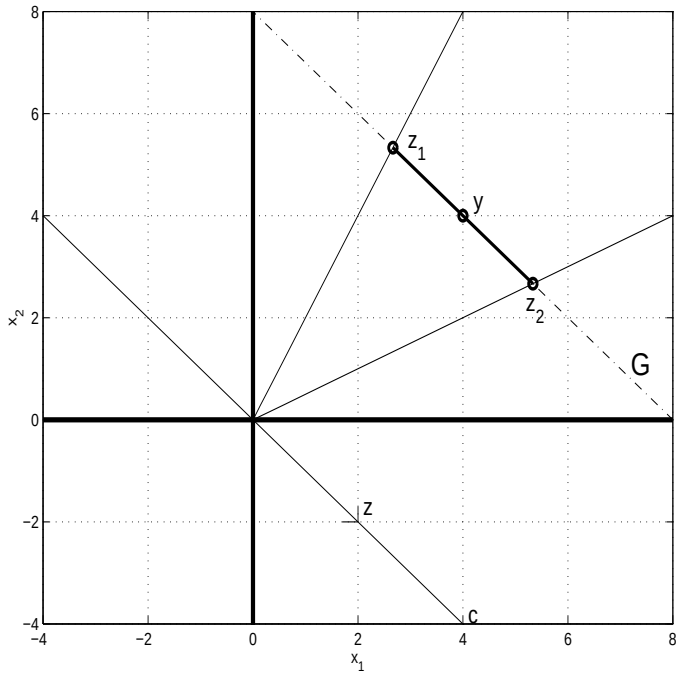


Abb. 5.12. Grafische Darstellung zu Satz **??**.

Korollar 5.82. *Jeder spitze polyedrische Kegel* $K \neq \{0\}$ *ist die Summe seiner Extremalstrahlen, d.h sind* $\text{cone}(\{e_i\})$, $i \in \{1, 2, \ldots, k\}$*, die Extremalstrahlen von* K*, so gilt*

$$K \;=\; \text{cone}(\{e_1, e_2, \ldots, e_k\}) \;=\; \sum_{i=1}^{k} \text{cone}(\{e_i\}).$$

Der folgende Satz verschärft Satz **??** für spitze Polyeder.

Theorem 5.83. *Jedes spitze Polyeder lässt sich als Summe der konvexen Hülle seiner Ecken und der konischen Hülle seiner Extremalen darstellen, d.h. ist* V *die Eckenmenge von* P *und* E *die Menge der Extremalstrahlen von* $\text{rec}(P)$ *(d.h.* E *ist eine Kegelbasis von* $\text{rec}(P)$ *), so gilt*

$$P \;=\; \text{conv}(V) \;+\; \text{cone}(E).$$

Beweis. Sei $\text{hog}(P)$ die Homogenisierung von P. Da P spitz ist, ist nach Satz **??** und Satz **??** $\text{hog}(P)$ ein spitzer

Kegel. Nach Folgerung **??** ist $\hog(P)$ die Summe seiner Extremalstrahlen $\cone(\{e_i\})$, $i \in \{1, 2, \ldots, k\}$. O.B.d.A. sei

$$e'_i = \begin{pmatrix} v_i \\ 1 \end{pmatrix},\ i \in \{1, 2, \ldots, p\} \quad \text{und} \quad e'_i = \begin{pmatrix} e_i \\ 0 \end{pmatrix},\ i \in \{p{+}1, p{+}2, \ldots, k\}$$

(vgl. Satz **??**). Aus Satz **??** folgt, dass $V = \{v_1, v_2, \ldots, v_p\}$ die Eckenmenge von P und $E = \{e_{p+1}, e_{p+2}, \ldots, e_k\}$ die Extremalenmenge von P ist. Nach Bemerkung **??** gilt

$$x \in P \iff \begin{pmatrix} x \\ 1 \end{pmatrix} \in \hog(P)$$

und damit

$$\begin{aligned} x \in P &\iff x = \sum_{i=1}^{p} \lambda_i v_i + \sum_{i=p+1}^{k} \mu_i e_i \quad \text{mit} \quad \lambda_i,\ \mu_i \geq 0,\ \sum_{i=1}^{p} \lambda_i = 1 \\ &\iff x \in \conv(V) + \cone(E). \end{aligned}$$

Korollar 5.84. *Polytope haben eine eindeutige konvexe Basis.*

Für lineare Programme gilt folgende wichtige Beobachtung:

Theorem 5.85. *Sei $P \subseteq \mathbb{K}^n$ ein spitzes Polyeder und $c \in \mathbb{K}^n$. Das lineare Programm $\max\{c^T x \,|\, x \in P\}$ ist genau dann unbeschränkt, wenn es eine Extremale e von P gibt mit $c^T e > 0$.*

Beweis. Übung.

Wir haben bereits gesehen, dass Elemente von spitzen Polyedern P keine eindeutige Darstellung als konvexe und konische Kombination von Ecken und Extremalen haben müssen. Man kann jedoch deren Anzahl beschränken.

Theorem 5.86. *Es sei $K \subseteq \mathbb{K}^n$ ein spitzer Kegel und $0 \neq x \in K$. Dann gibt es Extremalen $y_1, y_2, \ldots, y_d$ von K mit*

$$d \leq \dim(K) \leq n \qquad \textit{und} \qquad x = \sum_{i=1}^{d} y_i.$$

Beweis. Es seien $\text{cone}(\{e_i\})\,,\ i \in \{1, 2, \ldots, k\}$ die Extremalstrahlen von K. Dann gibt es nach Folgerung **??** Skalare $\lambda_i \geq 0\,,\ i \in \{1, 2, \ldots, k\}$ mit

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

Unter allen möglichen Darstellungen von x sei die obige so gewählt, dass $I = \{i \in \{1, 2, \ldots, k\} \mid \lambda_i > 0\}$ minimal ist. O.B.d.A. sei $I = \{1, 2, \ldots, d\}$. Angenommen, die Extremalen $e_1, e_2, \ldots, e_d$ seien linear abhängig, dann existieren $\mu_1, \mu_2, \ldots, \mu_d \in \mathbb{K}\,,\ \mu \neq 0$ mit

$$\sum_{i=1}^{d} \mu_i e_i \ = \ 0.$$

Angenommen, $\mu_1, \mu_2, \ldots, \mu_d \geq 0$ und o.B.d.A. $\mu_1 > 0$. Dann ist

$$-e_1 \ = \ \sum_{i=2}^{d} \frac{\mu_i}{\mu_1}\, e_i$$

eine konische Kombination von $e_2, e_3, \ldots, e_d$ und $e_1 \in \text{lineal}(K)$ nach Satz **??**. Dies ist jedoch ein Widerspruch zur Voraussetzung, dass es sich bei K um einen spitzen Kegel handelt (vgl. Satz **??**).

Daher können wir o.B.d.A. annehmen, dass $\mu_1 < 0$ und

$$\frac{\lambda_1}{\mu_1} \ = \ \max\left\{ \frac{\lambda_i}{\mu_i} \;\middle|\; \mu_i < 0 \right\}.$$

Damit gilt

$$\begin{aligned} e_1 \ &= \ -\sum_{i=2}^{d} \frac{\mu_i}{\mu_1} e_i, \\ x \ &= \ \sum_{i=2}^{d} \left(\lambda_i - \frac{\lambda_1}{\mu_1} \mu_i \right) e_i. \end{aligned}$$

Letzteres ist eine konische Kombination, denn es gilt für alle μ_i mit

$$\begin{aligned}
\mu_i > 0: &\quad \frac{\lambda_i}{\mu_i} \geq \frac{\lambda_1}{\mu_1} \quad \Longleftrightarrow \quad \left(\lambda_i - \frac{\lambda_1}{\mu_1}\mu_i\right) \geq 0\,,\\
\mu_i = 0: &\quad \lambda_i - \frac{\lambda_1}{\mu_1}\mu_i \geq 0 \Longleftrightarrow \quad \lambda_i \geq 0\,,\\
\mu_i < 0: &\quad \frac{\lambda_i}{\mu_i} \leq \frac{\lambda_1}{\mu_1} \quad \Longleftrightarrow \quad \left(\lambda_i - \frac{\lambda_1}{\mu_1}\mu_i\right) \geq 0.
\end{aligned}$$

Also kann x mit weniger als d Extremalen konisch kombiniert werden und dies führt zum Widerspruch zur Minimalität von I. Da ein Kegel höchstens $\dim(K)$ linear unabhängige Vektoren enthält, folgt $d \leq \dim(K)$. Setzen wir nun noch $y_i = \lambda_i e_i\,,\ i \in \{1, 2, \ldots, d\}$, so sind die Vektoren y_i Extremalen mit der gesuchten Eigenschaft.

Korollar 5.87. *Sei $P \subseteq \mathbb{K}^n$ ein spitzes Polyeder und $x \in P$, dann gibt es Ecken $v_0, v_1, \ldots, v_k$ und Extremalen $e_{k+1}, e_{k+1}, \ldots, e_d$ von P mit $d \leq \dim(P)$ und nichtnegative Skalare $\lambda_0, \lambda_1, \ldots, \lambda_k$ mit $\sum_{i=0}^{k} \lambda_i = 1$, so dass gilt*

$$x = \sum_{i=0}^{k} \lambda_i v_i + \sum_{i=k+1}^{d} e_i.$$

Beweis. Nach Satz **??** ist $\mathrm{hog}(P)$ ein Kegel, der nach Satz **??** spitz ist und nach Bemerkung **??** gilt

$$x \in P \quad \Longleftrightarrow \quad \begin{pmatrix} x \\ 1 \end{pmatrix} \in \mathrm{hog}(P).$$

Nach Satz **??** ist der Vektor $\binom{x}{1}$ eine konische Kombination von höchstens $d + 1 \leq \dim(P) + 1$ Extremalen von $\mathrm{hog}(P)$. O.B.d.A können wir annehmen, dass

$$\begin{pmatrix} x \\ 1 \end{pmatrix} = \sum_{i=0}^{k} \begin{pmatrix} y_i \\ \lambda_i \end{pmatrix} + \sum_{i=k+1}^{d} \begin{pmatrix} e_i \\ 0 \end{pmatrix}$$

gilt, wobei $\lambda_i > 0\,,\ i \in \{0, 1, \ldots, k\}$ ist (vgl. Satz **??**). Für $v_i = \dfrac{1}{\lambda_i} y_i$ gilt nun

$$x = \sum_{i=0}^{k} \lambda_i v_i + \sum_{i=k+1}^{d} e_i\,, \quad \sum_{i=0}^{k} \lambda_i = 1.$$

Ferner sind nach Satz **??** die Vektoren v_i Ecken von P und die Vektoren e_i Extremalen von P. Also haben wir die gewünschte Darstellung für x gefunden.

Altes Kommando `df` hier verwendet

Korollar 5.88. *(Satz von Carathéodory)*
Sei $P \subseteq \mathbb{K}^n$ ein Polytop. Dann ist jedes Element von P eine Konvexkombination von höchstens $\dim(P)+1$ Ecken von P.

5.8 Kodierung von Polyedern

Wenn wir von der „Äquivalenz“ der Probleme **??** bis **??** sprechen, müssen wir uns zuerst mit der Frage der Kodierung von Polyedern beschäftigen. Wir wissen, dass es zwei Darstellungsformen für Polyeder gibt:

$$P \;=\; P(A,b) \;=\; \mathrm{conv}(V) \;+\; \mathrm{cone}(E).$$

Je nach Darstellung ist entweder das Problem OPT oder das Problem SEP einfach:

$$\begin{aligned} P = P(A,b) \quad &\Longrightarrow \quad \text{SEP ist einfach.} \\ P = \mathrm{conv}(V) + \mathrm{cone}(E) \quad &\Longrightarrow \quad \text{OPT ist einfach.} \end{aligned}$$

Darüber hinaus kann die Transformation von einer Darstellung in die andere exponentiell sein.

Betrachte z.B. den Würfel $\{x \in \mathbb{R}^n \,\big|\, 0 \le x_i \le 1\,,\ i \in \{1,2,\ldots,n\}\}$ mit $2n$ Facetten und 2^n Ecken oder aber das Kreuzpolytop $\mathrm{conv}\{\pm e_i \,\big|\, i \in \{1,2,\ldots,n\}\}$ mit $2n$ Ecken und 2^n Facetten.

Eine Diskussion über polynomiale Äquivalenz macht also nur dann Sinn, wenn die Kodierung von Polyedern unabhängig von der Darstellung ist.

Definition 5.89. *Sei $P \subseteq \mathbb{K}^n$ ein Polyeder und φ und ν positive ganze Zahlen.*

Altes Kommando `df` hier verwendet

Altes Kommando `df` hier verwendet

(a) P hat eine Facettenkomplexität von höchstens φ, falls es ein System von Ungleichungen mit rationalen Koeffizienten $A \in \mathbb{Q}^{m\times n}$, $b \in \mathbb{Q}^m$ für ein $m \in \mathbb{N}$ gibt, so dass $P = P(A,b)$ und die Kodierungslänge jeder Ungleichung höchstens φ ist (für $P = \mathbb{R}^n$ verlangen wir $\varphi \ge n+1$).

(b) P hat eine Eckenkomplexität von höchstens ν, wenn es endliche Mengen $V, E \subseteq \mathbb{Q}^n$ gibt, so dass $P = \mathrm{conv}(V) + \mathrm{cone}(E)$ und die Kodierungslänge eines jeden Vektors in $V \cup E$ höchstens ν ist (für $P = \emptyset$ verlangen wir $\nu \ge n$).


Kommando
er verwendet


Kommando
er verwendet

(c) Ein wohlbeschriebenes Polyeder ist ein Tripel (P, n, φ), wobei $P \subseteq \mathbb{K}^n$ ein Polyeder mit einer Facettenkomplexität von höchstens φ ist. Die Kodierungslänge $\langle P \rangle$ eines wohlbeschriebenen Polyeders ist dann definiert als $\langle P \rangle = \varphi + n$.

In (c) hängt die Kodierungslänge von P nur von der Dimension und der Facettenkomplexität ab, nicht jedoch von der Eckenkomplexität. Das folgende Lemma zeigt, dass die Facettenkomplexität eines Polyeders polynomial durch seine Eckenkomplexität beschränkt ist und umgekehrt. Damit ist dann jede Aussage über polynomiale Lösbarkeit unabhängig von der Darstellung eines Polyeders.

Lemma 5.90. *Sei $P \subseteq \mathbb{K}^n$ ein Polyeder.*

(a) Falls P eine Facettenkomplexität von höchstens φ hat, dann hat es eine Eckenkomplexität von höchstens $4n^2\varphi$.
(b) Falls P eine Eckenkomplexität von höchstens ν hat, dann hat es eine Facettenkomplexität von höchstens $3n^2\nu$.

Zum Beweis von Lemma **??** benötigen wir folgende Beziehungen:

Lemma 5.91. *Es gelten die folgenden Abschätzungen:*

(a) Für jedes $x \in \mathbb{Q}^n$ gilt: $||x||_2 \leq 2^{\langle x \rangle - n} - 1$.
(b) Für jede Matrix $D \in \mathbb{Q}^{n \times n}$ gilt: $\langle \det(D) \rangle \leq 2 \langle D \rangle - n^2$.

Beweis. (a) Zunächst gilt für $x \in \mathbb{Q}^n$ die Ungleichung $||x||_2 \leq ||x||_1$, denn

$$||x||_2^2 = \sum_{i=1}^{n} x_i^2 \leq \left(\sum_{i=1}^{n} |x_i| \right)^2 = ||x||_1^2.$$

Für die L_1-Norm gilt dann

$$\begin{aligned} 1 + ||x||_1 &= 1 + \sum_{i=1}^{n} |x_i| \leq \prod_{i=1}^{n} (1 + |x_i|) \leq \prod_{i=1}^{n} 2^{\langle x_i \rangle - 1} \\ &= 2^{\sum_{i=1}^{n} (\langle x_i \rangle - 1)} = 2^{\langle x \rangle - n} \end{aligned}$$

und damit auch (a).

(b) Zum Beweis des Teils (b) benutzen wir die Hadamard-Ungleichung. Es gilt

$$|\det(A)| \;\leq\; \prod_{i=1}^{n} ||A_{\cdot i}||_2 \qquad \text{für eine Matrix } A \in \mathbb{Q}^{n\times n}.$$

Diese Beziehung liefert zusammen mit Teil (a)

$$\begin{aligned} 1 + |\det(D)| \;&\leq\; 1 + \prod_{i=1}^{n} ||D_{\cdot i}||_2 \;\leq\; \prod_{i=1}^{n} (1 + ||D_{\cdot i}||_2) \\ &\leq \prod_{i=1}^{n} 2^{\langle D_{\cdot i}\rangle - n} \;=\; 2^{\langle D\rangle - n^2} \end{aligned}$$

und damit

$$\begin{aligned} \langle \det(D)\rangle \;&=\; \lceil \log_2(|\det(D)| + 1)\rceil \;+\; 1 \\ &\leq\; \lceil \log_2 \left(2^{\langle D\rangle - n^2}\right)\rceil \;+\; 1 \;=\; \langle D\rangle \;-\; n^2 \;+\; 1 \\ &\leq\; 2\langle D\rangle \;-\; n^2. \end{aligned}$$

Nun zum
Beweis von Lemma ??.

(a) P habe eine Facettenkomplexität von höchstens φ, d.h. es ex. ein $m \in \mathbb{N}$, eine Matrix $A \in \mathbb{Q}^{m\times n}$ und ein Vektor $b \in \mathbb{Q}^m$ mit $P = P(A, b)$ und $\langle A_{i\cdot}\rangle + \langle b_i\rangle \leq \varphi\,,\;\; i \in \{1, 2, \ldots, m\}$. Weiterhin existieren endliche Mengen $V, E \subseteq \mathbb{Q}^n$ mit $P = \mathrm{conv}(V) + \mathrm{cone}(E)$ (vgl. Satz **??**). Mit Cramers Regel zur Lösung linearer Gleichungssysteme wissen wir nun, dass jede Komponente eines jeden Vektors aus $V \cup E$ der Quotient aus zwei Determinanten ist. Jede entspricht einer quadratischen Untermatrix (A, b) der Ordnung höchstens n. Für eine solche Unterdeterminante D folgt aus Lemma **??**

$$\langle \det(D)\rangle \;\leq\; 2\langle D\rangle - n^2 \;\leq\; 2\langle D\rangle \;\leq\; 2n\varphi.$$

Damit hat jeder Vektor in $V \cup E$ eine Kodierungslänge von höchstens $2n(2n\varphi) \;=\; 4n^2\varphi$.

(b) P habe eine Eckenkomplexität von höchstens ν. Falls $P = \emptyset$ oder $P = \{0\}$, so ist die Aussage offensichtlich richtig. In allen anderen Fällen gilt $\nu \geq n+1$ (Beachte, dass $\langle e_i\rangle = (n-1)\cdot 1 + 2 = n + 1$ gilt). Seien $V, E \subseteq \mathbb{Q}^n$ endliche Mengen mit $P = \mathrm{conv}(V) + \mathrm{cone}(E)$ und

jeder Vektor in $V \cup E$ habe eine Kodierungslänge von höchstens ν.
Wir betrachten zuerst den Fall, dass P volldimensional ist. Dann gilt, dass für jede Facette F von P Vektoren $v_1, v_2, \ldots, v_k \in V$, $r_1, r_2, \ldots, r_{n-k} \in E$ existieren, so dass

$$\text{aff}(F) \;=\; \left\{ x \in \mathbb{R}^n \;\middle|\; x \;=\; \sum_{i=1}^{k} \lambda_i v_i \;+\; \sum_{i=1}^{n-k} \mu_i r_i \,,\; \sum_{i=1}^{k} \lambda_i \;=\; 1 \right\}.$$

Mit anderen Worten, aff(F) ist definiert durch alle Vektoren x, die die folgende Gleichung erfüllen

$$\det \begin{pmatrix} 1 & 1 & \cdots & 1 & 0 & \cdots & 0 \\ x_1 & & & & & & \\ \vdots & \vdots & & \vdots & \vdots & & \vdots \\ \vdots & v_1 & \cdots & v_k & r_1 & \cdots & r_{n-k} \\ \vdots & \vdots & & \vdots & \vdots & & \vdots \\ x_n & & & & & & \end{pmatrix} \;=\; 0.$$

Entwickeln wir die Determinante nach der ersten Spalte, so erhalten wir

$$\sum_{i=1}^{n} (-1)^i \det(D_i)\, x_i \;=\; -\det(D_0),$$

wobei D_i die Matrix ist, die man durch Streichen der ersten Spalte und der $(i+1)$-ten Zeile erhält. Mit Lemma **??** erhalten wir nun für die Kodierungslänge dieser Gleichung

$$\begin{aligned} \sum_{i=0}^{n} \langle \det(D_i) \rangle \;&\leq\; \sum_{i=0}^{n} (2\langle D_i \rangle - n^2) \quad = \quad 2 \sum_{i=0}^{n} \langle D_i \rangle \;-\; (n+1)n^2 \\ &\leq\; 2n(n\nu \;+\; 2k \;+\; (n-k)) \;-\; (n+1)n^2 \\ &\leq\; 2n(n\nu + 2n) \;-\; (n+1)n^2 \\ &\leq\; 2n^2\nu \;-\; n^2(n-3) \quad \leq \quad 3n^2\nu. \end{aligned}$$

Damit gilt (b), falls P volldimensional ist. Der Fall, dass P nicht volldimensional ist, kann auf den gezeigten Fall zurückgeführt werden, siehe Übung.

Zur Entwicklung der Ellipsoid-Methode müssen wir uns nun zwei Fragen stellen: Wie finden wir am Anfang

ein geeignetes Ellipsoid, das P enthält und wann können wir die Iteration abbrechen. Die Antwort auf diese Fragen hängt in erster Linie von der Kodierungslänge ab, also von der Facetten-, bzw. Eckenkomplexität von P. Daher hängt auch die Iterationsanzahl und damit die Laufzeit von der Kodierungslänge von P ab. In unserer Analyse der Ellipsoid-Methode und der Transformation des Problems OPT in das Problem SEP benötigen wir einige Aussagen über das Volumen und die Rechengenauigkeit, um die Polynomialität des Algorithmus zu zeigen. Diese Aussagen hängen wiederum von der Kodierungslänge von P ab. Das folgende Beispiel soll diese Abhängigkeit demonstrieren.

Beispiel 5.92. Betrachten wir ein Polyeder mit einer Facettenkomplexität $\varphi \leq 10$, z.B.

$$P = \left\{ x \in \mathbb{R}^2 \;\middle|\; \begin{array}{rr} 2x_1 + 3x_2 & \leq 4 \\ -x_1 + x_2 & \geq 1 \\ x_1 & \geq 0 \end{array} \right\}.$$

Die Kodierungslänge der drei Facetten beträgt $10, 6$ und 4. Nach Lemma **??** lässt sich die Eckenkomplexität mit $4n^2\varphi = 160$ abschätzen. P hat drei Eckpunkte und kann in der Form

$$P = \operatorname{conv}\left(\left\{ \begin{pmatrix} 0 \\ 1 \end{pmatrix}, \begin{pmatrix} 0 \\ 4/3 \end{pmatrix}, \begin{pmatrix} 1/5 \\ 6/5 \end{pmatrix} \right\} \right) + \operatorname{cone}(\{0\}) \tag{5.18}$$

beschrieben werden. Die Kodierungslänge dieser vier Vektoren beträgt $3, 8, 13$ und 2. P hat damit eine Eckenkomplexität von höchstens 13.

Haben wir unser Polyeder in der Form (**??**) gegeben oder kennen wir eine Schranke für die Eckenkomplexität, so lässt sich ein Start-Ellipsoid leicht konstruieren. Die größtmögliche Zahl, die eine Kodierungslänge kleiner oder gleich ν hat, ist kleiner als 2^ν. Für unser Beispiel bedeutet dies, wir starten mit einer Kugel um den Punkt Null und einem Radius $r = 2^{13}$ (bzw. $r = 2^{160}$). Diese Kugel enthält nach Konstruktion alle Eckpunkte von P. Dies ist natürlich eine sehr grobe Abschätzung, aber für unser eigentliches Ziel (zu zeigen, dass alle vorkommenden Zahlen eine Kodierungslänge haben, die polynomial in n und φ ist) reicht dies aus.

Lemma 5.93. *Sei P ein Polyeder mit einer Eckenkomplexität von höchstens ν. Dann liegen alle Eckpunkte von P in einer Kugel B um den Nullpunkt mit Radius $r = 2^\nu$. Ist P zusätzlich beschränkt, dann gilt $P \subseteq B$.*

Beweis. Klar.

Ein Abbruchkriterium für die Ellipsoid-Methode erhält man aus einer unteren Schranke für das Volumen von P. Die Idee ist folgende: Konstruiere eine Kugel, die in P liegt und schätze deren Volumen ab. Für diese Kugel müssen wir einen Mittelpunkt und einen Radius bestimmen. Besitzen wir eine Beschreibung von P in der Form (**??**), so können wir den Schwerpunkt von P zum Mittelpunkt der Kugel wählen. In userem Beispiel wäre dies

$$c = \frac{1}{3}\left(\begin{pmatrix}0\\1\end{pmatrix} + \begin{pmatrix}0\\4/3\end{pmatrix} + \begin{pmatrix}1/5\\6/5\end{pmatrix}\right) = \begin{pmatrix}1/15\\53/45\end{pmatrix}.$$

Um einen geeigneten Radius zu finden, schätzen wir die Abstände der einzelnen Facetten zum Mittelpunkt ab. Sind alle Facetten gegeben (so wie in Beispiel **??**), ist dies recht einfach, aber wie gehen wir vor, wenn wir nicht alle Facetten kennen? Werfen wir einen Blick auf die verschiedenen Abstände, die vorkommen können. Alles was wir brauchen ist eine obere Grenze δ für den Nenner, der im Abstand einer Facette auftauchen kann. Die Kugel mit dem Mittelpunkt c und dem Radius $r = 1/\delta$ liegt dann vollständig in P. Da jede Ecke eine Kodierungslänge von höchstens ν und jede Facette eine Kodierungslänge von höchstens φ hat, können wir eine obere Schranke für δ bestimmen. Dies ist Gegenstand von

Lemma 5.94. *Sei $P \subseteq \mathbb{R}^n$ ein volldimensionales Polyeder mit einer Facettenkomplexität von höchstens φ. Dann gilt*

$$\mathrm{vol}(P) \geq 2^{-8n^4\varphi}.$$

Beweis. Wir konstruieren eine Kugel mit einem Radius r und berechnen dann das Volumen dieser Kugel. Nach Lemma **??**(a) existieren endliche Mengen $V, E \in \mathbb{Q}^n$ mit $P = \mathrm{conv}(V) + \mathrm{cone}(E)$ und jeder Vektor aus $V \cup E$ hat eine Kodierungslänge von höchstens $\nu = 4n^2\varphi$. Da P volldimensional ist, existieren Vektoren $v_1, v_2, \ldots, v_t \in V$, $r_1, r_2, \ldots, r_k \in E$ mit $t+k = n+1$ und $v_1, v_2, \ldots, v_t, v_1 + r_1, v_1 + r_2, \ldots, v_1 + r_k$ sind affin unabhängig. Setze nun

$$c = \frac{1}{n+1}\left(\sum_{i=1}^{t} v_i + \sum_{i=1}^{k}(v_1 + r_i)\right).$$

c ist eine Konvexkombination von Punkten in P und daher gilt $c \in P$. Die Behauptung ist nun, dass eine Kugel B mit dem Mittelpunkt c und dem Radius $2^{-7n^3\varphi}$ ganz in P liegt. Dazu zeigen wir, dass jede Facette mindestens den Abstand $2^{-7n^3\varphi}$ vom Punkt c besitzt.
Dazu sei $\bar{a}^T x \leq \bar{b}$ eine facettendefinierende Ungleichung von P mit einer Komplexität von höchstens φ. Multiplizieren wir diese Gleichung mit allen in $\bar{a}$ und $\bar{b}$ vorkommenden Nennern, so bekommen wir eine facettendefinierende Ungleichung $a^T x \leq b$ mit $a \in \mathbb{Z}^n$, $b \in \mathbb{Z}$. Mit $\langle a \rangle \leq n\varphi$ und $\langle b \rangle \leq \varphi$ lässt sich die Kodierungslänge der Ungleichung durch den Term $(n+1)\varphi$ abschätzen. Der Abstand dieser Facette zum Punkt c beträgt

$$d = \frac{b - a^T c}{||a||}.$$

Eine untere Schranke für d leiten wir aus dem Nenner von d her. Der Nenner von d setzt sich aus dem Nenner von $b - a^T c$ und dem Faktor $||a||$ zusammen. Da b und die Komponenten von a ganze Zahlen sind, hat der Term $b - a^T c$ den gleichen Nenner wie c. Dieser hat höchstens den Wert $(n+1)2^{(n+1)\nu}$. Da $c \in P$ liegt gilt $b - a^T c > 0$ und damit

$$b - a^T c \geq \frac{1}{(n+1)2^{(n+1)\nu}}.$$

Aus Lemma **??** wissen wir $||a|| \leq 2^{\langle a \rangle} \leq 2^{n\varphi}$. Zusammen bekommen wir nun

$$d \geq \frac{1}{(n+1)2^{(n+1)\nu} \cdot 2^{n\varphi}} = \frac{1}{(n+1)2^{(n+1)4n^2\varphi + n\varphi}} \geq 2^{-7n^3\varphi}.$$

Daher liegt B in P und wir erhalten für das Volumen von P die Abschätzung

$$\text{vol}(P) \geq \text{vol}(B) = 2^{-7n^4\varphi} \cdot \text{vol}(B_1) \geq \frac{2^{-7n^4\varphi}}{n^n} \geq \frac{2^{-7n^4\varphi}}{2^{n^2}} \geq 2^{-8n^4\varphi}.$$

Dabei bezeichnet B_1 die Einheitskugel mit einem Volumen von mindestens $1/n^n$, da $\{x \in \mathbb{R}^n \,|\, 0 \leq x_i \leq 1/n\} \subset B_1$.

6

Komplexität Linearer Programme

6.1 Komplexität von Polyedern

Die klassische Komplexitätstheorie misst die Laufzeit von Verfahren in Abhängigkeit der „Größe der Eingabe“. Unsere Objekte sind Polyeder, Vektoren und Matrizen, so dass wir uns mit der Frage beschäftigen müssen, wie wir deren „Größe“ messen.

Wir definieren zunächst die *Codierungslänge* einer ganzen Zahl $a \in \mathbb{Z}$ als: Codierungslänge

$$\langle a \rangle := 1 + \lceil \log_2(|a| + 1) \rceil.$$

Dies entspricht der Anzahl der Bits, die man in der Standard-Binärdarstellung benötigt, um a darzustellen. Jede rationale Zahl $r \in \mathbb{Q}$ besitzt eine eindeutige Darstellung $r = p/q$ mit teilerfremden $r \in \mathbb{Z}$ und $q \in \mathbb{N}$. Wir setzen dann

$$\langle q \rangle := \langle p \rangle + \langle q \rangle.$$

Wir setzen die Codierungslänge $\langle \cdot \rangle$ in naheliegender Weise auf Vektoren $x = (x_1, \ldots, x_n)^T \in \mathbb{Q}^n$ und Matrizen $A = (a_{ij}) \in \mathbb{Q}^{m \times n}$ fort:

$$\langle x \rangle := \sum_{i=1}^{n} \langle x_i \rangle$$
$$\langle A \rangle := \sum_{i=1}^{m} \sum_{j=1}^{n} \langle a_{ij} \rangle.$$

Wir zeigen zunächst zwei einfache Abschätzungen für die Codierungslänge von Vektoren und Matrizen. Für den Beweis benötigen wir zwei klassische Ungleichungen,

Hadamard Ungleichung

die *Cauchy-Schwarzsche Ungleichung* und die *Hadamard Ungleichung.*

Cauchy-Schwarzsche Ungleich

Lemma 6.1 (Cauchy-Schwarzsche Ungleichung). *Sind a_i, b_i, c_i für $i = 1, \ldots, n$ komplexe Zahlen, so gilt:*

$$\sum_{k=1}^{n} |a_k b_k| \leq \left(\sum_{k=1}^{n} |a_k|^2 \right)^{\frac{1}{2}} \cdot \left(\sum_{k=1}^{n} |b_k|^2 \right)^{\frac{1}{2}}. \tag{6.1}$$

Beweis. Wir setzen:

$$A := \left(\sum_{k=1}^{n} |a_k|^2 \right)^{\frac{1}{2}}$$
$$B := \left(\sum_{k=1}^{n} |b_k|^2 \right)^{\frac{1}{2}}.$$

Falls $A = 0$, so gilt $a_k = 0$ für alle k und die linke Seite von (**??**) ist Null, genauso wie die rechte Seite der Ungleichung. Analoges gilt im Fall $B = 0$. Wir können also $A > 0$ und $B > 0$ annehmen. In diesem Fall setzen wir für $k = 1, \ldots, n$:

$$\alpha_k := |a_k|/A \quad \text{ und } \beta_k := |b_k|/B.$$

Die Ungleichung (**??**) ist mit diesen Bezeichnungen dann äquivalent zu

$$\sum_{k=1}^{n} \alpha_k \beta_k \leq 1. \tag{6.2}$$

Da für $a, b \geq 0$ die Ungleichung $\sqrt{ab} \leq (a + b)/2$ gilt, haben wir:

$$\begin{aligned} \sum_{k=1}^{n} \alpha_k \beta_k &= \sum_{k=1}^{n} \sqrt{\alpha_k^2 \beta_k^2} \leq \sum_{k=1}^{n} \left(\frac{\alpha_k^2}{2} + \frac{\beta_k^2}{2} \right) \\ &= \frac{1}{2} \sum_{k=1}^{2} \alpha_k^2 + \frac{1}{2} \sum_{k=1}^{2} \beta_k^2 = \frac{1}{2} + \frac{1}{2} = 1. \end{aligned}$$

Dies zeigt (**??**). □

Lemma 6.2 (Hadamard Ungleichung). *Ist A eine $n \times m$-Matrix mit den Spalten $a_1, \ldots, a_m$, dann gilt:*

$$\sqrt{\det A^T A} \leq \sum_{i=1}^{m} \|a_i\|_2. \tag{6.3}$$

Ist insbesondere $m = n$, also A eine quadratische Matrix, so gilt:

$$|\det A| \leq \sum_{i=1}^{n} \|a_i\|_2. \tag{6.4}$$

Beweis. Siehe z.B. [**?**, ?] □

Lemma 6.3. *(a) Für jede rationale Zahl r gilt, $|r| \leq 2^{\langle r\rangle - 1} - 1$.*
(b) Für jeden Vektor $x \in \mathbb{Q}^n$, gilt $\|x\|_2 \leq \|x\|_1 \leq 2^{\langle x\rangle - n} - 1$.
(c) Für jede Matrix $D \in \mathbb{Q}^{m\times n}$ gilt: $|\det D| \leq 2^{\langle D\rangle - n^2} - 1$.
(d) Für jede Matrix $D \in \mathbb{Z}^{m\times n}$ gilt: $\langle \det D\rangle | \leq \langle D\rangle - n^2 + 1$.

Beweis. (a) Sei $r = p/q \in \mathbb{Q}$. Nach Definition von $\langle\cdot\rangle$ gilt für die ganze Zahl p dann $|p| \leq 2^{\langle p\rangle - 1} - 1$, so dass

$$|r| = |p/q| \leq |p| \leq 2^{\langle p\rangle - 1} - 1 \leq 2^{\langle r\rangle - 1} - 1.$$

(b) Sei $x = (x_1, \ldots, x_n)^T$. Wir haben nach der Cauchy-Schwarzschen-Ungleichung (**??**):

$$\|x\|_2^2 = \sum_{i=1}^{n} |x_i| \cdot |x_i| \leq \sum_{i=1}^{n} |x_i| \cdot \sum_{i=1}^{n} |x_i| = \|x\|_1^2,$$

also $\|x\|_2 \leq \|x\|_1$. Andererseits gilt für $i = 1, \ldots, n$ nach (a) die Ungleichung $|x_i| \leq 2^{\langle x_i\rangle - 1} - 1$. Daher folgt

$$\begin{aligned} 1 + \|x\|_1 = 1 + \sum_{i=1}^{n} |x_i| &\leq \prod_{i=1}^{n} (1 + |x_i|) \\ &\leq \sum_{i=1}^{n} 2^{\langle x_i\rangle - 1} = 2^{\langle x\rangle - n}. \end{aligned}$$

(c) Sind $d_1, \ldots, d_n$ die Zeilen von D, so gilt nach der Hadamard-Ungleichung (**??**):

$$\begin{aligned} 1 + |\det D| \leq 1 + \sum_{i=1}^{n} \|d_i\|_2 &\leq \prod_{i=1}^{n} (1 + \|d_i\|_2) \\ &\overset{\text{Teil (b)}}{\leq} \prod_{i=1}^{n} 2^{\langle d_i\rangle - 1} = 2^{\langle D\rangle - n^2}. \end{aligned}$$

(d) Folgt unmittelbar aus (c) und (a). □

Wir kommen nun zur Codierung von Polyedern. Wir wissen, dass es hier zwei Darstellungsformen für ein Polyeder $P \subseteq \mathbb{R}^n$ gibt:

$$P = P(A,b) = \{\, x \in \mathbb{R}^n : Ax \leq b \,\} \quad (H\text{-Darstellung})$$
$$P = \operatorname{conv}(V) + \operatorname{cone}(E) \quad (V\text{-Darstellung}).$$

Dabei ist es für ein bestimmtes Polyeder von entscheidender Bedeutung, welche Darstellung wir wählen. Beispielsweise hat der Einheitswürfel im $\mathbb{R}^n$ die H-Darstellung:

$$Q = \{\, x \in \mathbb{R}^n : 0 \leq x_i \leq 1, i = 1, \ldots, n \,\}$$

mit $2n$ Ungleichungen und die V-Darstellung

$$V = \operatorname{conv}\{\, \chi_S : S \subseteq \{1, \ldots, n\} \,\}$$

mit 2^n Ecken. Der Größensprung vn der H-Darstellung zur V-Darstellung kann also exponentiell sein. Anhand des Kreuzpolytops, das die V-Darstellung

$$K = \operatorname{conv}\{\, \pm e_i : i = 1, \ldots, n \,\}$$

mit $2n$ Ecken sowie die H-Darstellung

$$K = \{\ \}$$

besitzt, sieht man, dass auch die Umwandlung der V-Darstellung in eine H-Darstellung eine exponentielle Vergrößerung erfordern kann.

6.2 Innere-Punkte Verfahren

Wir betrachten das Lineare Programm

$$(P) \qquad \min \quad c^T x \tag{6.5a}$$
$$Ax = b \tag{6.5b}$$
$$x \geq 0, \tag{6.5c}$$

in Standardform, wobei wie üblich A eine $m \times n$-Matrix, $b \in \mathbb{R}^m$ und $c \in \mathbb{R}^n$ ist. Das duale Programm zu (**??**) lässt sich nach Einführen von Slackvariablen $s \in \mathbb{R}^n_+$ als

$$(D) \qquad \max \quad b^T y \tag{6.6a}$$
$$A^T y + s = c \tag{6.6b}$$
$$s \geq 0 \tag{6.6c}$$

schreiben. Unsere Ergebnisse über Innere-Punkte-Verfahren leiten wir zunächst einmal unter folgenden Voraussetzungen her:

Voraussetzung 6.4. *(i) Die Matrix A besitzt vollen Zeilenrang, d.h.* $\operatorname{Rang} A = m$.
(ii) Beide Probleme (??) und (??) besitzen strikt zulässige Lösungen in dem Sinne, dass die beiden folgenden Mengen nichtleer sind:

$$P^+ = \{\, x \in \mathbb{R}^n : Ax = b, x > 0 \,\}$$
$$D^+ = \left\{\, (y,s) \in \mathbb{R}^m \times \mathbb{R}^n : A^T y + s = c, s > 0 \,\right\}.$$

Wir werden später zeigen, dass wir die Voraussetzung **??** ohne Einschränkung der Allgemeinheit machen können. Im Moment vereinfacht sie aber die Darstellung.

Man beachte, dass aufgrund des Dualitätssatzes der Linearen Programmierung (Satz **??**) Voraussetzung **??** impliziert, dass sowohl (P) als auch (D) Optimallösungen haben, da beide Probleme zulässige Lösungen besitzen.

In diesem Abschnitt verwenden wir zudem die in der Literatur zu Innere-Punkte Verfahren übliche Notation, dass wir mit dem Großbuchstaben die zu einem Vektore entsprechende Diagonalmatrix bezeichnen: Ist also $x \in \mathbb{R}^n$, dann setzen wir

$$X = \operatorname{diag}(x) = \begin{pmatrix} x_1 & & & \\ & x_2 & & \\ & & \ddots & \\ & & & x_n \end{pmatrix}$$

Letztendlich schreibern wir e für den Vektor aus lauter Einsen:

$$e = (1, \ldots, 1)^T.$$

Wir leiten zunächst eine einfache aber hilfreiche Formel für die Dualitätslücke zweier zulässiger Vektoren her:

Lemma 6.5. *Sei $x \in \mathbb{R}^n$ zulässig für (P) und (y,s) zulässig für (D). Es gilt dann*

$$c^T x - b^T y = x^T s \geq 0. \tag{6.7}$$

Beweis. Wegen $x \geq 0$ und $s \geq 0$ ist auch das Skalarprodukt $x^T s$ nichtnegativ. Weiterhin gilt:

$$0 \leq x^T s = x^T (c - A^T y) = c^T x - (Ax)^T y = c^T x - b^T y.$$

Dies war zu zeigen. □

Der Wert $x^T s$ misst also genau die Dualitätslücke zwischen den Zielfunktionswerten $c^T x$ im primalen Problem (P) und $b^T y$ im dualen Problem (D). Somit sind die zulässigen Vektoren x und (y, s) genau dann optimal, wenn $x^T s = 0$ gilt (vgl. auch der Satz über den komplementären Schlupf (Korollar **??**). Für $x \geq 0$ und $s \geq 0$ ist die Bedingung $x^T s = 0$ offenbar äquivalent zu $x_i s_i = 0$ für $i = 1, \ldots, n$. Dies können wir mit der Notation $X = \operatorname{diag}(x)$ auch als $Xs = 0$ schreiben.

Wir erhalten also, dass $x \in \mathbb{R}^n$ and $(y, s) \in \mathbb{R}^m \times \mathbb{R}^n$ genau dann optimal für die Probleme (P) bzw. (D) sind, wenn sie folgendes (nichtlineares) System erfüllen:

$$Ax = b \tag{6.8a}$$

$$A^T y + s = c \tag{6.8b}$$

$$Xs = 0 \tag{6.8c}$$

$$x \geq 0, s \geq 0 \tag{6.8d}$$

Im obigen System mit $2n + m$ Gleichungen in $2n + m$ Variablen ist die einzige Nichtlinearität die Bedingung (**??**). Alle anderen Bedindungen sind lineare Gleichungen bzw. Vorzeichenrestriktionen.

Im Folgenden benötigen wir die zur Euklidischen Norm $\|.\|_2$ zugehörige *Matrixnorm* lub_2, die für eine (nicht notwendigerweise quadratische) Matrix M folgendermaßen definiert ist:

$$\mathrm{lub}_2(M) := \max_{x \neq 0} \frac{\|Mx\|_2}{\|x\|_2} = \max_{\|x\|_2 = 1} \|Mx\|_2. \tag{6.9}$$

Man sieht leicht, dass durch (**??**) tatsächlich eine Norm definiert wird. Insbesondere ist $\mathrm{lub}_2(M) > 0$ genau dann, wenn $M \neq 0$ gilt. Die lub_2-Norm ist mit der Euklidischen Norm kompatibel in dem Sinne, dass für alle $x \in \mathbb{R}^n$ gilt:

$$\|Mx\|_2 \leq \mathrm{lub}_2(M) \cdot \|x\|_2. \tag{6.10}$$

Man kann darüberhinaus zeigen (siehe z.B. [**?**, **?**], dass gilt:

$$\mathrm{lub}_2(M) = \max \left\{ \sqrt{\lambda} : \lambda \text{ ist Eigenwert von } M^T M \right\}. \tag{6.11}$$

Ist insbesondere M eine quadratische Matrix, so gilt $M^T M = M^2$ und

$$\mathrm{lub}_2(M) = \max \{ |\lambda| : \lambda \text{ ist Eigenwert von } M \}. \tag{6.12}$$

6.2.1 Exkurs: Das Newton-Verfahren

Das Newton-Verfahren ist ein Verfahren zur Lösung von Nullstellenproblemen der Form

$$F(z) = 0 \text{ mit } F\colon \mathbb{R}^n \to \mathbb{R}^n. \tag{6.13}$$

Bis auf die Vorzeichenrestriktionen (**??**) besitzt das System (**??**), das die Optimallösungen unserer Linearen Programme beschreibt, genau die Form (**??**).

Wir nehmen an, dass F eine (unbekannte) Nullstelle $\bar{z}$ besitzt, also $F(\bar{z}) = 0$ gilt und $F \in C^2(\mathbb{R}^n)$ gilt (also F zweimal stetig differenzierbar auf $\mathbb{R}^n$ ist).

Sei $z \in \mathbb{R}^n$ eine Näherung für $\bar{z}$ und $F(z) \neq 0$ (falls $F(z) = 0$, so haben wir bereits eine Nullstelle gefunden und müssen nicht mehr weitersuchen). Wir suchen einen „Korrekturschritt“ $\Delta z \in \mathbb{R}^n$, so dass $z + \Delta z$ eine bessere Näherung für $\bar{z}$ ist als z. Dazu nähern wir F durch seine Linearisierung in z an (vgl. den Satz von Taylor in Standardbüchern über Analysis [**?**, **?**, **?**]):

$$F(z + \Delta z) \approx F(z) + DF(z)\Delta z.$$

Wenn wir die rechte Seite Null setzen und $DF(z)^{-1}$ existiert, so ergibt sich der Newtonschritt als

$$\Delta z = -DF(z)^{-1}F(z).$$

Die nächste Näherung für $\bar{z}$ ist dann

$$z^+ := z - DF(z)^{-1}F(z).$$

Damit haben wir bereits die Grundform des *Newton-Verfahrens* hergeleitet. Es startet ausgehend von einem Vektor $z^{(0)} \in \mathbb{R}^n$ und berechnet dann iterativ eine Folge $(z^{(k)})_k$ durch

$$z^{(k+1)} := z^{(k)} - DF(z^{(k)})^{-1}F(z^{(k)}). \tag{6.14}$$

Das Ergebnis des folgenden Satzes wird für die weiteren Ergebnisse dieses Abschnittes nicht direkt benötigt. Andererseits ist die (lokal) quadratische Konvergenz des Newton-Verfahrens die Schlüsseleigenschaft für die Polynomialität des innere-Punkte Verfahrens.

Theorem 6.6. *Sei $F\colon \mathbb{R}^n \to \mathbb{R}$ dreimal stetig differenzierbar, $F \in C^3(\mathbb{R}^n)$ und $F(\bar{z}) = 0$ für ein (unbekanntes)*

$\bar{z} \in \mathbb{R}^n$, so dass die Inverse $DF(\bar{z})^{-1}$ der Ableitung $DF(\bar{z})$ in $\bar{z}$ existiert.

Für $z^{(0)}$ genügend nahe bei $\bar{z}$ konvergiert die Folge $(z^{(k)})_k$, welche das Newton-Verfahren (??) erzeugt gegen x^* und es gibt ein $c > 0$, so dass

$$\|z^{(k+1)} - \bar{z}\| \leq c\|z^{(k)} - \bar{z}\|^2, \text{ für } k = 0, 1, 2, \ldots. \quad (6.15)$$

Die Konvergenz ist also (lokal) mindestens quadratisch.

Beweis. Falls $DF(x^{(k)})^{-1}$ existiert, so gilt nach der Definition von $z^{(k+1)}$ in (??):

$$\begin{aligned}\|z^{(k+1)} - \bar{z}\|_2 &= \|z^{(k)} - DF(z^{(k)})^{-1}F(z^{(k)}) - \bar{z}\|_2 \\ &= \|DF(z^{(k)})^{-1}[DF(z^{(k)})(z^{(k)} - \bar{z}) - F(z^{(k)})]\|_2 \\ &= \|DF(z^{(k)})^{-1}[-F(z^{(k)}) + DF(z^{(k)})(z^{(k)} - \bar{z})]\|_2 \\ &\leq \mathrm{lub}_2(DF(z^{(k)})^{-1}) \cdot \| - F(z^{(k)}) + DF(z^{(k)})(z^{(k)} - \bar{z})\|_2\end{aligned}$$

Da $F(\bar{z}) = 0$ gilt, erhalten wir also

$$\|z^{(k+1)} - \bar{z}\|_2 \leq \mathrm{lub}_2(DF(z^{(k)})^{-1}) \cdot \|F(\bar{z}) - F(z^{(k)}) + DF(z^{(k)})(z^{(k)} - \bar{z})\|_2. \quad (6.16)$$

Da $F \in C^3(\mathbb{R}^n)$ und $DF(\bar{z})^{-1}$ existiert, ist die Matrix $DF(z)$ auch für alle z nahe bei $\bar{z}$ invertierbar. Es existieren also Konstanten $\alpha > 0$ und $\delta > 0$, so dass für $\|z - \bar{z}\|_2 < \delta$ die Ungleichung $\mathrm{lub}_2(DF(z)^{-1}) < \alpha$ gilt. Ist also $\|z^{(k)} - \bar{z}\| < \delta$, so ist die lub_2-Norm von $DF(z^{(k)})^{-1}$ in (??) nach oben durch α beschränkt. Wir betrachten nun den Term $F(\bar{z}) - F(z^{(k)}) + DF(z^{(k)})(z^{(k)} - \bar{z})$. Nach dem Satz von Taylor [?, ?, ?] gibt es $\beta > 0$ und $\gamma > 0$, so dass für $\|\bar{z} - z^{(k)}\| < \beta$ gilt:

$$F(\bar{z}) = F(z^{(k)}) + D(z^{(k)})(\bar{z} - z^{(k)}) + r(\|\bar{z} - z^{(k)}\|), \quad (6.17)$$

mit $r(h) \leq \gamma\|h\|_2^2$. Ist nun $\alpha\gamma\|z^{(k)} - \bar{z}\|_2 < 1$ und $\|z^{(k)} - \bar{z}\| < \delta$, so ergibt sich aus (??) und (??) dann:

$$\begin{aligned}\|z^{(k+1)} - \bar{z}\|_2 &\leq \alpha\|F(\bar{z}) - F(z^{(k)}) + DF(z^{(k)})(z^{(k)} - \bar{z})\|_2 \\ &\leq \alpha\gamma\|z^{(k)} - \bar{z}\|_2^2 \\ &< \|z^{(k)} - \bar{z}\|.\end{aligned}$$

Insbesondere liegt dann $z^{(k+1)}$ also näher an $\bar{z}$ als $z^{(k)}$ und $z^{(k+1)}$ erfüllt wieder alle notwendigen Voraussetzungen, um

$$\|z^{(k+2)} - \bar{z}\|_2 \leq \alpha\gamma\|z^{(k+1)} - \bar{z}\|_2^2 < \|z^{(k+1)} - \bar{z}\|_2$$

zu zeigen. Per Induktion folgt nun unmittelbar, dass das Verfahren durchführbar ist und die Ungleichung (**??**) mit $c := \alpha\gamma$ gilt. □

6.2.2 Der Newton-Schritt und ein relaxiertes nichtlineares System

Um die Linearen Programme (P) und (D) zu lösen, genügt es, das nichtlineare System (**??**) zu lösen. Wir setzen

$$P_0(x, y, z) := \begin{pmatrix} Ax - b \\ A^T y + s - c \\ Xs \end{pmatrix}. \tag{6.18}$$

Dann ist (**??**) äquivalent zu

$$P_0(x, y, s) = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}, \quad x \geq 0, s \geq 0. \tag{6.19}$$

Betrachten wir das System (**??**) und versuchen, es mit dem Newton-Verfahren zu lösen. Das Newton-Verfahren erzeugt Folgen $(x^{(k)})_k$ und $(y^{(k)}, s^{(k)})_k$ durch:

$$\begin{pmatrix} x^{(k+1)} \\ y^{(k+1)} \\ s^{(k+1)}) \end{pmatrix} = \begin{pmatrix} x^{(k)} \\ y^{(k)} \\ s^{(k)}) \end{pmatrix} - J^{-1} \cdot P_0(x^{(k)}, y^{(k)}, s^{(k)}),$$

wobei

$$\begin{aligned} J &:= J(x^{(k)}, y^{(k)}, s^{(k)}) \\ &:= DP_0(x^{(k)}, y^{(k)}, s^{(k)}) = \begin{pmatrix} A & 0 & 0 \\ 0 & A^T & I \\ S & 0 & X \end{pmatrix} \end{aligned}$$

die Jakobi-Matrix der Funktion P_0 ist. Damit das Newton-Verfahren durchführbar ist, müssen wir die Nichtsingularität der Matrix J sichern. Falls aber für unsere aktuellen Iterierten $x = x^{(k)}$ und $(y, s) = (y^{(k)}, s^{(k)})$ sowohl $x_j = 0$ als auch $s_j = 0$ für ein j sind, so enthält J eine Nullzeile! Wegen des komplementären Schlupfes gilt für die Optimallösungen $x_j^* s_j^* = 0$, so dass diese Situation sogar recht wahrscheinlich aussieht.

Daher betrachten wir jetzt für einen Parameter $\mu > 0$ ein „relaxiertes System“, das mittels der Funktion

$$P_\mu(x,y,z) := \begin{pmatrix} Ax-b \\ A^T y + s - c \\ Xs - \mu e \end{pmatrix} \tag{6.20}$$

definiert ist:

$$P_\mu(x,y,s) = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}, \quad x \geq 0, s \geq 0 \tag{6.21}$$

Die Jakob-Matrix der Funktion P_μ ist unabhängig von $\mu > 0$ und stimmt mit der von P_0 aus (**??**) überein. Jede Lösung von (**??**) (sofern sie existiert) besteht wegen der Bedingung $Xs - \mu e = 0$, also $x_i s_i = \mu > 0$ für $i = 1, \ldots, n$ und $x \geq 0$, $s \geq 0$ aus strikt positiven Vektoren $x(\mu) > 0$ und $s(\mu) > 0$. Somit tritt das Problem mit der Nullzeile in $J = DP_\mu$ nicht in der Lösung auf.

Theorem 6.7. *Die Jakobi-Matrix*

$$J := DP_\mu(x,y,s) = \begin{pmatrix} A & 0 & 0 \\ 0 & A^T & I \\ S & 0 & X \end{pmatrix} \tag{6.22}$$

*der Funktion P_μ aus (**??**) ist nichtsingulär, wenn $x > 0$ und $s > 0$.*

Beweis. Wir nehmen an, dass es $\begin{pmatrix} u \\ v \\ w \end{pmatrix}$ gibt, so dass

$$\begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix} = DP_\mu(x,y,s) \cdot \begin{pmatrix} u \\ v \\ w \end{pmatrix} = \begin{pmatrix} Au \\ A^T v + w \\ Su + Xw \end{pmatrix}.$$

Dann gilt

$$u^T w = u^T(-A^T v) = -(\underbrace{Au}_{=0})^T v = 0. \tag{6.23}$$

Wenn wir $Su + Xw = 0$ nach u auflösen, erhalten wir $u = -S^{-1}Xw$. Eingesetzt in (**??**) ergibt sich

$$0 = u^T w = w^T u = -w^T S^{-1} X w. \tag{6.24}$$

Da $x > 0$ und $s > 0$ ist die Diagonalmatrix $S^{-1}X$ positiv definit. Aus (**??**) folgt daher $w = 0$. Dann können wir aber aus $0 = Su + Xw = Su$ und $s > 0$ schließen, dass $u = 0$ gilt. Aus $0 = A^T v + w = A^T v$ ergibt sich $v = 0$ wegen Rang $A = m$ (siehe Voraussetzung **??**). □

Sofern wir im Newton-Verfahren strikt positives $x^{(k)}$ und $s^{(k)}$ haben, bleibt nach dem obigen Satz die Jakobi-Matrix $J := DP_\mu(x^{(k)}, y^{(k)}, s^{(k)})$ also nichtsingulär und der Newton-Schritt

$$\begin{pmatrix} x^{(k+1)} \\ y^{(k+1)} \\ s^{(k+1)}) \end{pmatrix} = \begin{pmatrix} x^{(k)} \\ y^{(k)} \\ s^{(k)}) \end{pmatrix} - J^{-1} \cdot P_\mu(x^{(k)}, y^{(k)}, s^{(k)})$$

ist durchführbar. Um das Newton-Verfahren jetzt aber weiterlaufen zu lassen, müssen wir zwei Dinge sicherstellen:

1. Es sollte auch $x^{(k+1)} > 0$ und $s^{(k+1)} > 0$ gelten, damit auch der nächste Newton-Schritt existiert.
2. Es ist bisher noch nicht klar, ob das System (**??**) überhaupt eine Lösung hat. Falls keine solche Lösung existiert, so kann das Newton-Verfahren offenbar auch nicht gegen eine solche konvergieren.

6.2.3 Der zentrale Pfad

Wir beschäftigen und zunächst mit der Frage nach der Existenz einer Lösung des Systems (**??**), das wir hier der Übersicht halber noch einmal explizit aufschreiben:

$$Ax = b \tag{??a}$$
$$A^T y + s = c \tag{??b}$$
$$Xs = \mu e \tag{??c}$$
$$x \geq 0, s \geq 0. \tag{??d}$$

Wir betrachten dazu die sogenannte *logarithmische Barrierefunktion* $f_\mu \colon P^+ \to \mathbb{R}$, definiert durch

logarithmische Barrierefunktion

$$f_\mu(x) = c^T x - \mu \sum_{i=1}^{n} \ln x_i. \tag{6.26}$$

Für $x \in P^+$ gilt $x > 0$, und somit ist f_μ wohldefiniert und sogar zweimal stetig differenzierbar. Wir haben für $x > 0$:

$$\nabla f_\mu(x) = c - \mu X^{-1} e$$
$$\nabla^2 f_\mu(x) = \mu X^{-2}.$$

Insbesondere ist die Hessematrix $\nabla^2 f_\mu$ für $x > 0$ positiv definit und f_μ daher nach bekannten Ergebnissen der Analysis strikt konvex in $\mathbb{R}^{++}_n = \{ x \in \mathbb{R}^n : x > 0 \}$ (siehe z.B. [?]). Wir benötigen ein etwas technisches Lemma über Minima von f_μ:

Lemma 6.8. *Der Vektor $x^* \in P^+ = \{ x : Ax = b, x > 0 \}$ ist genau dann ein Minimum von f_μ auf P^+, wenn*

$$\nabla f_\mu(x^*) \perp N(A),$$

d.h. $\nabla f_\mu(x^*)^T w = 0$ für alle $w \in N(A) = \{ x : Ax = 0 \}$, gilt.

Beweis. „⇒“: Sei x^* Minimum von f_μ auf P^+ und $w \in N(A)$ beliebig. Wir betrachten für einen skalaren Parameter λ den Vektor $x^* + \lambda w$. Es gilt dann für alle $t \in \mathbb{R}$:

$$A(x^* + \lambda w) = Ax^* + \lambda Aw \overset{w \in N(A)}{=} Ax^* + \lambda 0 = Ax^*.$$

Da $x^* > 0$ ist auch $x^* + \lambda w > 0$ für alle $\lambda \in \mathbb{R}$ mit $|\lambda|$ genügend klein. Somit ist für kleines $|\lambda|$ dann $x^* + \lambda w \in P^+$. Wir benutzen nun die Differenzierbarkeit von f_μ in x^*. Es gilt dann:

$$\begin{aligned} \nabla f_\mu(x^*)^T w &= \lim_{\lambda \downarrow 0} \frac{f_\mu(x^* + \lambda w) - f_\mu(x^*)}{\lambda} \\ &= \lim_{\lambda \uparrow 0} \frac{f_\mu(x^* + \lambda w) - f_\mu(x^*)}{\lambda}, \end{aligned}$$

da f_μ auf P^+ differenzierbar ist. Einerseits haben wir daher:

$$\nabla f_\mu(x^*)^T w = \lim_{\lambda \downarrow 0} \frac{f_\mu(x^* + \lambda w) - f_\mu(x^*)}{\lambda} \geq 0.$$

da aufgrund der Optimalität von x^* und $x^* + \lambda w \in P^+$ die Ungleichung $f_\mu(x^* + \lambda w) \geq f(x^*)$ gilt und im obigen Grenzwert stets $\lambda > 0$ ist. Andererseits gilt aber auch:

$$\nabla f_\mu(x^*)^T w = \lim_{\lambda \uparrow 0} \frac{f_\mu(x^* + \lambda w) - f_\mu(x^*)}{\lambda} \leq 0,$$

wie oben wegen der Optimalität von x^* und $\lambda < 0$ im obigen Grenzwert. Damit folgt insgesamt $\nabla f_\mu(x^*)^T w = 0$.
„⇐“: Wie wir bereits gesehen haben, ist f_μ konvex und differenzierbar auf der konvexen Menge P^+. Aus der Analysis ist bekannt [?], dass dann für $x, x^* \in P^+$ gilt:

$$f_\mu(x) \geq f_\mu(x^*) + \nabla f_\mu(x^*)^T (x - x^*). \tag{6.27}$$

Wegen $Ax = b$ und $Ax^* = b$ ist $A(x - x^*) = 0$, also $x - x^* \in N(A)$. Nach Voraussetzung ist also $\nabla f_\mu(x^*)^T (x - x^*) = 0$ und aus (**??**) folgt damit $f_\mu(x) \geq f_\mu(x^*)$. □

Lemma 6.9. *Sei B eine $m \times n$-Matrix mit* $\operatorname{Rang} B = m$ *und*

$$R(B^T) = \{\, B^T y : y \in \mathbb{R}^m \,\} \subseteq \mathbb{R}^n$$
$$N(B) = \{\, x : Bx = 0 \,\} \subseteq \mathbb{R}^n$$

der Bildraum von B^T bzw. der Nullraum von B. Dann sind $R(B^T)$ und $N(A)$ orthogonale Komplemente, d.h.,

$$R(B^T) \perp N(A) \quad \textit{und} \quad R(B^T) \oplus N(A) = \mathbb{R}^n.$$

Beweis. Sei $u \in R(B^T)$, etwa $u = B^T w$ und $v \in N(B)$. Dann gilt $u^T v = (B^T w)^T v = w^T B v = w^T 0 = 0$. Also haben wir $R(B^T) \perp N(B)$. Dies impliziert darüberhinaus, dass $R(B^T) \cap N(B) = \{\, 0 \,\}$, da jeder Vektor $x \in R(B^T) \cap N(A)$ die Bedingung $0 = x^T x = \|x\|^2$ erfüllt.

Für $x \in \mathbb{R}^n$ definieren wir $q := B^T(BB^T)^{-1}Bx \in R(B^T)$ und $s := x - q$. Man beachte, dass BB^T in der Tat invertierbar ist, da für $y \neq 0$ wegen $\operatorname{Rang} B = m$, also der linearen Unabhängigkeit der Spalten von B^T gilt: $B^T y \neq 0$. Daher ist dann für $y \neq 0$ auch $0 < \|B^T y\|^2 = (B^T y)^T (B^T y) = y^T B B^T y$ und die Matrix BB^T sogar positiv definit.

Es gilt nun für unsere oben definierten Vektoren q und s die Gleichung $Bs = Bx - BB^T(BB^T)^{-1}Bx = 0$. Also haben wir $x = q + s$ mit $q \in R(B^T)$ und $s \in N(A)$. □

Als Nebenprodukt des Beweises von Lemma **??** sehen wir, dass die Orthogonalprojektion auf den Unterraum $R := R(B^T)$ durch die Matrix

$$\Pi_R = B^T(BB^T)^{-1}B \tag{6.28}$$

beschrieben wird. Darüberhinaus ist die Orthogonalprojektionen auf das Komplement $N := N(B)$ gegeben durch die Matrix

$$\Pi_N = I - \Pi_R. \tag{6.29}$$

Diese Eigenschaften werden noch später nützlich sein.

Wir sind nun in der Lage, die Existenz einer Lösung $x(\mu)$, $y(\mu)$, $s(\mu)$ des Systems (**??**) zu zeigen.

Theorem 6.10. *Unter der Voraussetzung* **??** *hat das System (**??**) für jedes $\mu > 0$ eine eindeutige Lösung $x(\mu)$, $y(\mu)$, $s(\mu)$.*

Beweis. Zunächst zeigen wir, dass (**??**) höchstens eine Lösung hat. Seien (x, y, s) und (x', y', s') zwei Lösungen und $\Delta x := x - x'$, $\Delta y := y - y'$, $\Delta s := s - s'$. Da $Xs = \mu e = X's'$ haben wir:

$$X\Delta s + S'\Delta x = \underbrace{Xs}_{=\mu e} - Xs' + \underbrace{S'x}_{=Xs'} - \underbrace{S'x'}_{=\mu e} = 0.$$

Somit gilt für Δx, Δy und Δs:

$$\begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix} = \begin{pmatrix} A\Delta x \\ A^T\Delta y + \Delta s \\ S'\Delta x + X\Delta s \end{pmatrix} = \begin{pmatrix} A & 0 & 0 \\ 0 & A^T & I \\ S' & 0 & X \end{pmatrix} \begin{pmatrix} \Delta x \\ \Delta y \\ \Delta s \end{pmatrix} \quad (6.30)$$

Da $S' > 0$ und $X > 0$ ist die Matrix aus (**??**) nach Satz **??** nichtsingulär und es folgt $\Delta x = 0$, $\Delta y = 0$ und $\Delta s = 0$, also $x = x'$, $y = y'$ und $s = s'$.

Die Existenz einer Lösung ist etwas aufwändiger zu zeigen. Nach Voraussetzung **??** gibt es $\bar{x} > 0$ mit $Ax = b$ und $\bar{y}$, $\bar{s} > 0$ mit $A^T\bar{y} + \bar{s} = c$. Sei $\gamma := f_\mu(\bar{x})$. Wir betrachten die Menge

$$L := \left\{ x \in P^+ : f_\mu(x) \leq \gamma \right\}. \quad (6.31)$$

Da $\bar{x} \in L$, ist L eine nichtleere Teilmenge von P^+. Wir zeigen, dass L kompakt ist. Dazu genügt es, die Beschränktheit von L zu zeigen, da die Abgeschlossenheit sofort aus der Stetigkeit von f_μ auf P^+ folgt.

Bevor wir den technischen Beweis der Beschränktheit durchführen, zeigen wir, dass aus der Kompaktheit von L die Behauptung des Satzes folgt.

Wenn L kompakt (und nichtleer) ist, so hat die stetige Funktion f_μ auf L ein Minimum x. Dieses ist dann offenbar auch Minimum von f_μ auf P^+. Nach Lemma **??** gilt dann

$$0 = \nabla f_\mu(x)^T w = (c - \mu X^{-1}e)^T w$$

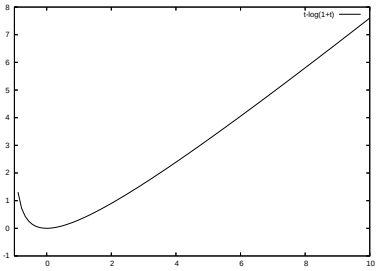
für alle $w \in N(A)$. Da $N(A)$ und $R(A^T) = \left\{ A^T y : y \in \mathbb{R}^m \right\}$ nach Lemma **??** orthogonale Komplemente sind, gibt es ein $y \in \mathbb{R}^m$ mit $c - \mu X^{-1}e = A^T y$. Wenn wir $s := \mu X^{-1}e > 0$ definieren, so haben wir damit eine Lösung von (**??**) konstruiert.

Um den Beweis zu vervollständigen, zeigen wir nun die Beschränktheit der Menge L aus (**??**). Für beliebiges $x \in P^+$ gilt:

$$
\begin{aligned}
f_\mu(x) &= c^T x - \mu \sum_{i=1}^{n} \ln x_i \\
&= b^T \bar{y} + x^T \bar{s} - \mu \sum_{i=1}^{n} \ln x_i \qquad \text{(nach Lemma ??)} \\
&= e^T (X\bar{s} - e) - \mu \sum_{i=1}^{n} \log \frac{x_j \bar{s}_j}{\mu} \\
&\quad + \underbrace{n - n\mu \ln \mu + b^T \bar{y} + \mu \textstyle\sum_{i=1}^{n} \ln \bar{s}_i}_{=:\alpha} \\
&= e^T (X\bar{s} - e) - \mu \sum_{i=1}^{n} \log \frac{x_j \bar{s}_j}{\mu} + \alpha.
\end{aligned}
$$

Falls $x \in L$, so gilt $f_\mu(x) \leq \gamma$, also

$$
e^T (X\bar{s} - e) - \mu \sum_{i=1}^{n} \log \frac{x_j \bar{s}_j}{\mu} \leq \gamma - \alpha =: \bar{\gamma}. \tag{6.32}
$$



Mit Hilfe der Funktion $\psi\colon (-1, \infty) \to \mathbb{R}$, die durch

$$
\psi(t) = t - \ln(1 + t)
$$

definiert ist, können wir die linke Seite von (**??**) umschreiben und erhalten:

$$
\mu \sum_{i=1}^{n} \psi\left(\frac{x_i \bar{s}_i}{\mu} - 1\right) \leq \bar{\gamma}. \tag{6.33}
$$

Hierbei haben wir benutzt, dass $e^T e = n$ gilt. Man sieht mit elementaren Methoden der Analysis leicht, dass für alle $t \in (-1, \infty)$ die Bedingung $\psi(t) \geq 0$ gilt. Somit ist jeder Term in der Summe in (**??**) nichtnegativ und es folgt für $x \in L$:

$$
\psi\left(\frac{x_i \bar{s}_i}{\mu} - 1\right) \leq \frac{\bar{\gamma}}{\mu}, \text{ für } i = 1, \ldots, n. \tag{6.34}
$$

Wegen $\psi'(t) = 1 - \frac{1}{1+t} = \frac{t}{1+t}$ ist die Funktion ψ für $t > 0$ strikt monoton steigend. Daher folgt aus (**??**) und $\bar{s}_i > 0$ für $i = 1, \ldots, n$, dass x_i von oben durch eine Konstante beschränkt ist. Daher ist L beschränkt. □

Definition 6.11 (Primal-dualer zentraler Pfad). *Für $\mu > 0$ sei $(x(\mu), y(\mu), s(\mu))$ die eindeutige Lösung von (**??**). Die Menge*

$$\{\, (x(\mu), y(\mu), s(\mu)) : \mu > 0 \,\}$$

nennt man dann den primal-dualen zentralen Pfad *für die Probleme (P) und (D).*

primal-dualen zentralen P

Falls $(x(\mu), y(\mu), s(\mu))$ auf dem zentralen Pfad liegen, dann gilt

$$\begin{aligned} n\mu &= \sum_{i=1}^{n} \underbrace{x_i(\mu)s_i(\mu)}_{=\mu} = x(\mu)^T s(\mu) \\ &= x(\mu)^T(c - A^T y(\mu)) = c^T x(\mu) - y^T \underbrace{Ax(\mu)}_{=b} \\ &= c^T x(\mu) - b^T y(\mu). \end{aligned}$$

Wir würden also erwarten, dass für $\mu \to 0$ die Lösungen auf den zentralen Pfad gegen Optimallösungen von (P) und (D) konvergieren, da die Dualitätslücke gegen 0 geht. Die Strategie für das primal-duale Innere-Punkte Verfahren sieht jetzt wie folgt aus:

1. Für ein (aktuell gegebenes) $\mu > 0$ und einen Startvektor (x, y, s) nähern wir den zugehörigigen Punkt $(x(\mu), y(\mu), s(\mu))$ auf dem zentralen Pfad mit Hilfe des Newton-Verfahrens an. (Es zeigt sich später, dass unter geeigneten Voraussetzungen dafür ein einziger Newton-Schritt ausreicht). Sei (x^+, y^+, s^+) die entsprechende Näherung.
2. Wir verringern μ, setzen $(x, y, s) := (x^+, y^+, s^+)$ und gehen wieder zu Schritt **??**.

6.2.4 Das Newton-Verfahren für das primal-duale System

Wir nehmen an, dass wir x, y, s gegeben haben, so dass

$$\begin{aligned} Ax - b &= 0 \\ A^T y + s - c &= 0 \\ x > 0, s &> 0. \end{aligned}$$

Mit anderen Worten, wir haben zulässige Vektoren x und (y, s) für (P) und (D), so dass $x > 0$ und $s > 0$ gilt. Solche Vektoren nennen wir *strikt primal zulässig* bzw. *strikt dual zulässig.*

strikt primal zulässig

strikt dual zulässig

Für diese Vektoren definieren wir das *Residuum* $r =$

Residuum

$r(x, y, s)$ durch:

$$r := r(x, y, s) := Xs - \mu e. \tag{6.35}$$

Der Newton-Schritt $(\Delta x, \Delta y, \Delta s)$ bei (x, y, s) ergibt sich als die Lösung von

$$DP_\mu(x, y, s) \cdot \begin{pmatrix} \Delta x \\ \Delta y \\ \Delta s \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ -r \end{pmatrix}.$$

Wenn wir die Formel aus (**??**) für $DP_\mu(x, y, s)$ verwenden, so ist dies äquivalent zu:

$$\begin{pmatrix} A & 0 & 0 \\ 0 & A^T & I \\ S & 0 & X \end{pmatrix} \cdot \begin{pmatrix} \Delta x \\ \Delta y \\ \Delta s \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ -r \end{pmatrix}. \tag{6.36}$$

Wie wir bereits aus Satz **??** wissen, ist $DP_\mu(x, y, s)$ nichtsingulär und das System (**??**) ist eindeutig lösbar. Sei

$$(x^+, y^+, s^+) := (x, y, s) + (\Delta x, \Delta y, \Delta s)$$

der Newton-Nachfolger von (x, y, s). Unser Ziel ist es nun, die Eigenschaften des Newton-Schritts zu analysieren. Nach Konstruktion ist $A\Delta x = 0$ und $A^T \Delta y + \Delta s = 0$, so dass $Ax^+ = b$ und $A^T y^+ + s^+ = c$, so dass wir uns hier auf das neue Residuum

$$r^+ := X^+ s^+ - \mu e$$

konzentrieren können.

Wir haben bereits weiter oben in (**??**) und (**??**) Eigenschaften von Orthogonalprojektionen auf bestimmte Unterräume hergeleitet. Für die Analyse des Newton-Schrittes betrachten wir diese Projektionen noch einmal:

Lemma 6.12. *Sei B eine m timesn-Matrix mit* $\operatorname{Rang} B = m$ *sowie Π_R und Π_N die Orthogonalprojektionen auf den Bildraum $R = R(B^T)$ bzw. den Nullraum $N(B)$. Für jeden Vektor $q \in \mathbb{R}^n$ gilt*

$$\|\Pi_N q\|_2 = \|q\|_2 \cdot \cos\theta \quad \text{und} \quad \|\Pi_R q\|_2 = \|q\|_2 \cdot \sin\theta,$$

wobei $\theta = \angle(q, \Pi_N q)$ der Winkel zwischen q und $\Pi_N q$ ist.

Beweis. Einserseits gilt

$$q^T \Pi_N q = \|q\|_2 \cdot \|\Pi_N q\|_2 \cdot \cos\angle(q, \Pi_N q). \tag{6.37}$$

Andererseits haben wir

$$q^T \Pi_N q = (\Pi_R q + \Pi_N q)^T \Pi_N q = \|\Pi_N q\|_2^2, \qquad (6.38)$$

da nach Lemma **??** R und N orthogonale Komplemente sind. Aus (**??**) und (**??**) folgt nun $\|\Pi_N q\|_2^2 = \|q\|_2 \|\Pi_N q\|_2 \cos \angle(q, \Pi_N)$. Dies ist die behauptete Gleichung für $\Pi_N q$.

Da $\|q\|_2^2 = \|\Pi_N q\|_2^2 + \|\Pi_R q\|_2^2$ erhalten wir aus der eben hergeleiteten Gleichung für $\Pi_N q$ nun

$$\|\Pi_R\|_2^2 = \|q\|_2^2 - \|q\|_2^2 \cos^2\theta = \|q\|_2^2(1-\cos^2\theta) = \|q\|_2^2 \sin^2\theta.$$

Dies war zu zeigen. □

Wir schreiben nun die Lösung $(\Delta x, \Delta y, \Delta s)$ von (**??**) mit Hilfe von Orthogonalprojektionen. Sei dazu die Diagonalmatrix D definiert durch:

$$D^2 := XS^{-1}. \qquad (6.39)$$

Dies ist möglich, da alle Einträge in X und S strikt positiv sind. Wir definieren außerdem den Vektor q durch

$$q := DX^{-1}r \qquad (6.40)$$

Lemma 6.13. *Die eindeutige Lösung von (**??**) ist gegeben durch:*

$$q = DX^{-1}r \qquad (6.41a)$$
$$\Delta x = D^2 A^T \Delta y - Dq \qquad (6.41b)$$
$$\Delta y = (AD^2A^T)^{-1} ADq \qquad (6.41c)$$
$$\Delta s = -D^{-1}q - D^{-2}\Delta x, \qquad (6.41d)$$

oder äquivalent als Lösung von

$$q = DX^{-1}r \qquad (6.42a)$$
$$DA^T \Delta y = \Pi_R q \qquad (6.42b)$$
$$\Delta x = -D\Pi_N q \qquad (6.42c)$$
$$\Delta s = -D^{-1}\Pi_R q. \qquad (6.42d)$$

Beweis. Wir wissen bereits aus Satz **??**, dass das System (**??**) eindeutig lösbar ist. Daher genügt es zu zeigen, dass die Vektoren aus den Systemen (**??**) bzw. (**??**) Lösungen von (**??**) sind.

Wir zeigen zunächst die Äquivalenz von (**??**) und (**??**). Da A^T wegen Rang $A = m$ vollen Spaltenrang besitzt, gibt

es höchstens seine Lösung von $DA^T \Delta y = \Pi_R q$. Es genügt für die Äquivalenz daher zu zeigen, dass jede Lösung von (**??**) auch Lösung von (**??**) ist.

Nach (**??**) und (**??**) werden die Orthogonalprojektionen auf den Nullraum von $B := AD$ und den Bildraum von $B^T = DA^T$ durch folgende Matrizen beschrieben:

$$\Pi_R = DA^T(AD^2A^T)^{-1}AD$$
$$\Pi_N = I - \Pi_R = I - DA^T(AD^2A^T)^{-1}AD.$$

Da A vollen Zeilenrang hat, können wir (**??**) mit DA^T multiplizieren, ohne die Lösungsmenge zu verändern. Damit ist (**??**) äquivalent zu:

$$DA^T \Delta y = DA^T(AD^2A^T)^{-1}ADq = \Pi_R q. \tag{6.43}$$

Da $\Pi_R + \Pi_N = I$, können wir (**??**) wie folgt äquivalent umschreiben:

$$\begin{aligned} \Delta x &= DDA^T \Delta y - Dq \\ &= D\Pi_R q - Dq \\ &= D(\Pi_R - I)q \\ &= -D\Pi_N q. \end{aligned} \tag{6.44}$$

Letztendlich formulieren wir (**??**) äquivalent als:

$$\begin{aligned} \Delta s &= -D^{-1}(q + D^{-1}\Delta x) \\ &= -D^{-1}(q + D^{-1}(-D\Pi_N q)) \\ &= -D^{-1}(q - \Pi_N q) \\ &= -D^{-1}\Pi_R q. \end{aligned} \tag{6.45}$$

Die Gleichungen (**??**), (**??**) und (**??**) entsprechen genau dem System (**??**), womit die gewünschte Äquivalenz gezeigt ist.

Im verbleibenden Beweis müssen wir nur noch zeigen, dass die Lösung von (**??**) auch das System (**??**) für den Newton-Schritt löst. Wir haben

$$A\Delta x = -AD\Pi_N q = 0 \qquad \text{(nach (??))}$$

da Π_N die Orthogonalprojektion auf den Nullraum von AD ist. Weiterhin gilt:

$$\begin{aligned} A^T \Delta y + \Delta s &= D^{-1}(DA^T \Delta y + D\Delta s) \\ &= D^{-1}(\Pi_R q + D(-D^{-1}\Pi_R q)) \quad \text{(nach (??) und (??))} \\ &= D^{-1}0 = 0. \end{aligned}$$

Letztendlich ist

$$\begin{aligned} X\Delta s + S\Delta x &= X(-D^{-1}\Pi_R q) + S(-D\Pi_N q) && \text{(by (??))} \\ &= -XD^{-1}(\Pi_R q + X^{-1}S \underbrace{D^2}_{=XS^{-1}} \Pi_N q) \\ &= -XD^{-1}(\Pi_R q + \Pi_N q) \\ &= -XD^{-1}q \\ &= -r && \text{(nach (??)).} \end{aligned}$$

Dies war zu zeigen.

Wenn wir ausgehend von (x, y, s) einen Newton-Schritt $(\Delta x, \Delta y, \Delta s)$ durchführen, so erhalten wir den Newton-Nachfolger $(x^+, y^+, s^+) = (x, y, s) + (\Delta x, \Delta y, \Delta s)$. Das neue Residuum $r^+ := X^+ s^+ - \mu e$ erfüllt dann

$$\begin{aligned} r^+ &= (X + \Delta X)(s + \Delta s) - \mu e \\ &= Xs + \underbrace{X\Delta s + S\Delta x}_{=-r \text{ nach (??)}} + \Delta X \Delta s - \mu e \\ &= \overbrace{Xs - \mu e}^{=r} - r + \Delta X \Delta s \\ &= \Delta X \Delta s. \end{aligned} \tag{6.46}$$

Wir setzen

$$\Delta\tilde{x} := -D^{-1}\Delta x = \Pi_N q \tag{6.47}$$

$$\Delta := -D\Delta s = \Pi_R q \tag{6.48}$$

In (??) folgt die letzte Gleichheit aus (??), in (??) ergibt sich die letzte Gleichheit aus (??). Es gilt dann $\Delta X \Delta s = \Delta\tilde{X}\Delta$ und nach (??) haben wir

$$r^+ = \Delta\tilde{X}\Delta\,. \tag{6.49}$$

Lemma 6.14. *Falls für unsere aktuellen Iterierten $x > 0$, y, $s > 0$ das Residuum $r := Xs - \mu e$ die Bedingung $\|r\|_2 \le \beta\mu$ für ein $\beta \in [0, 1/2]$ erfüllt, dann gilt für das Residuum r^+ des Newton-Nachfolgers*

$$\|r^+\|_2 \le \mu\beta^2.$$

Beweis. Nach Definition von q in (??) gilt $q = DX^{-1}r$, so dass $\|q\|_2 \le \mathrm{lub}_2(DX^{-1})\|r\|_2$ (vgl.(??). Damit ergibt sich

$$DX^{-1} = (\sqrt{XS^{-1}})X^{-1} = \sqrt{X^{-1}S^{-1}} = (\sqrt{R+\mu I})^{-1} = \begin{pmatrix} r_1+\mu & & & \\ & r_2+\mu & & \\ & & \ddots & \\ & & & r_n+\mu \end{pmatrix}^{-1/2},$$

wobei die letzte Gleichung aus $r_i = x_i s_i - \mu$ für $i = 1, \ldots, n$ folgt. Also können wir die Norm $\mathrm{lub}_2(DX^{-1})$ wie folgt nach oben abschätzen:

$$\begin{aligned} \mathrm{lub}_2(DX^{-1}) &= \mathrm{lub}_2((\sqrt{R+\mu I})^{-1}) \\ &= \max\left\{ |\lambda| : \ \lambda \text{ ist Eigenwert von } (\sqrt{R+\mu I})^{-1} \right\} \\ &= \max_{i=1,\ldots,n} \frac{1}{\sqrt{|r_i+\mu|}}. \end{aligned}$$

Für die letzte Gleichung haben wir ausgenutzt, dass die Einträge einer Diagonalmatrix genau ihre Eigenwerte sind. Mit Hilfe der Dreiecksungliechung und $|r_i| \leq \|r\|_2 \leq \beta\mu$ erhalten wir

$$|r_i + \mu| = |\mu - (-r_i)| \geq \mu - |r_i| \geq \mu - \beta\mu = (1-\beta)\mu.$$

Somit gilt

$$\mathrm{lub}_2(DX^{-1}) \leq 1/\sqrt{(1-\beta)\mu}. \tag{6.50}$$

Wir erhalten die folgende Abschätzung für die Norm $\|q\|_2$:

$$\|q\|_2 \leq \mathrm{lub}_2(DX^{-1})\|r\|_2 \leq \frac{\beta\mu}{\sqrt{(1-\beta)\mu}} = \beta\sqrt{\frac{\mu}{1-\beta}}. \tag{6.51}$$

Nach Definition von $\Delta\tilde{x}$ und Δ in (**??**) bzw.(**??**) folgt daraus

$$\|\Delta\tilde{x}\|_2 = \|\Pi_N q\|_2 \overset{\text{Lemma ??}}{=} \|q\|_2 \cdot |\cos\theta| \tag{6.52a}$$

$$\|\Delta\|_2 = \|\Pi_R q\|_2 \overset{\text{Lemma ??}}{=} \|q\|_2 \cdot |\sin\theta|. \tag{6.52b}$$

Die Gleichungen (**??**) liefern das entscheidende Hilfsmitte, um das Folgeresiduum r^+ abzuschätzen:

$$\begin{aligned}
\|r^+\|_2 &= \|\Delta\tilde{X}\Delta\,\|_2 \\
&\le \|\Delta\tilde{x}\|_2 \cdot \|\Delta\,\|_2 \\
&\le \|q\|_2^2 \cdot |\sin\theta\cos\theta| && \text{(nach (??))} \\
&= \|q\|_2^2 \cdot |\frac{1}{2}\sin 2\theta| \\
&\le \frac{1}{2}\|q\|_2^2 && \text{(da } \sin 2\theta \in [0,1]\text{)} \\
&\le \frac{1}{2}\beta^2\mu\frac{1}{1-\beta} && \text{(nach (??))} \\
&\le \beta^2\mu && \text{(da } \beta \in [0,1/2]\text{).}
\end{aligned}$$

Dies beendet den Beweis des Lemmas. □

Wir folgern nun aus dem letzten Lemma eine entscheidende Konsequenz: Falls die Voraussetzungen von Lemma **??** erfüllt sind, dann bleiben die Newton-Nachfolger x^+ und s^+ strikt positiv. Daher ist das Newton-Verfahren dann weiterhin durchführbar, weil die zugehörige Jakobi-Matrix aus (**??**) nichtsingulär bleibt (siehe Satz **??**):

Korollar 6.15. *Unter den Voraussetzungen von Lemma* **??** *gilt* $x^+ > 0$ *und* $s^+ > 0$.

Beweis. Zunächst zeigen wir $x^+ \ge 0$ und $x^* \ge 0$, also nur die Nichtnegativität der Newton-Nachfolger. Nach (**??**) gilt $\|\Delta\tilde{x}\|_2 \le \|q\|_2$. Daher folgt:

$$\begin{aligned}
\|X^{-1}\Delta x\|_2 &= \|X^{-1}D\Delta\tilde{x}\|_2 && \text{(nach (??))} \\
&\le \mathrm{lub}_2(DX^{-1})\|\Delta\tilde{x}\|_2 \\
&\le \frac{\|\Delta\tilde{x}\|_2}{\sqrt{(1-\beta)\mu}} && \text{(nach (??))} \\
&\le \frac{\beta\sqrt{\frac{\mu}{1-\beta}}}{\sqrt{(1-\beta)\mu}} && \text{(nach (??) und wegen } \|\Delta\tilde{x}\|_2 \le \|q\|_2\text{)} \\
&= \frac{\beta}{1-\beta} \\
&\le 1. && \text{(da } \beta \in [0,1/2]\text{)}
\end{aligned}$$

Nach der obigen Rechnung haben wir $\|X^{-1}x\|_2 \le 1$. Insbesondere folgt, dass die Norm jedes Eintrags des Vektors $X^{-1}x$ durch 1 beschränkt sein muss, also $|\frac{\Delta x_i}{x_i}| \le 1$ oder äquivalent $|\Delta x_i| \le |x_i|$ für $i = 1,\ldots,n$. Also ist $x_i + \Delta x_i \ge 0$ für $i = 1,\ldots,n$ und $x^+ = x + \Delta x \ge 0$.

Wir können eine analoge Rechnung für $s^+ = s + \Delta s$ durchführen. Wir haben $S^{-1}D^{-1} = S^{-1}\sqrt{X^{-1}S} = \sqrt{X^{-1}S^{-1}} = DX^{-1}$. Also ist

$$\begin{aligned}\|S^{-1}\Delta s\|_2 &= \|S^{-1}D^{-1}\Delta\|_2 && \text{(by (??))}\\ &\leq \mathrm{lub}_2(DX^{-1})\|\Delta\|_2 &&\\ &\leq \frac{\|\Delta\|_2}{\sqrt{(1-\beta)\mu}} && \text{(by (??))}\\ &\leq 1. && \text{(da } \beta \in [0,1/2]).\end{aligned}$$

Also ist auch $s^+ = s + \Delta s \geq 0$ nichtnegativ.

Wir zeigen nun, dass x^+ und s^+ sogar strikt positiv sind. Wir haben für $i = 1, \ldots, n$:

$$|x_i^+ s_i^+ - \mu| \leq \|X^+ s^+ - \mu e\|_2 = \|r^+\|_2 \overset{\text{Lemma ??}}{\leq} \beta^2\mu < \mu.$$

Daher müssen für $i = 1, \ldots, n$ sowohl $x_i^+ > 0$ als auch $s_i^+ > 0$ strikt positiv sein. □

6.2.5 Ein Primal-Duales Kurzschrittverfahren

Die Ergebnisse des letzten Abschnitts legen nun eine Idee für ein Verfahren nahe. Wir starten mit $(x^{(0)}, y^{(0)}, s^{(0)})$ und μ_0, so dass das Startresiduum $r^{(0)}$ die Ungleichung $\|r^{(0)}\|_2 \leq \frac{1}{2}\mu_0$ erfüllt (wir stellen die Frage, wie wir an solche Startvektoren gelangen, einen Moment zurück).

Für solche Startwerte sind die Voraussetzungen von Lemma **??** bzw. Korollar **??** erfüllt. Wir führen dann einen Newton-Schritt durch, wobei unser Ziel der Vektor $(x(\mu_0), y(\mu_0), s(\mu_0))$ auf dem zentralen Pfad ist. Der Newton-Schritt liefert uns wieder strikt zulässige Vektoren (Korollar **??**) und bringt uns deutlich dichter an den zentralen Pfad heran (Lemma **??**). Wir werden in der Analyse (Satz **??**) gleich sehen, dass wir sogar so nahe an den zentralen Pfad kommen, dass wir unser Ziel dort in Richtung Optimum verschieben können: wir reduzieren μ_0 auf $\mu_1 := (1 - \frac{1}{6\sqrt{n}})\mu_0$, peilen also mit dem nächsten Newton-Schritt $(x(\mu_1), y(\mu_1), s(\mu_1))$ auf dem zentralen Pfad an und iterieren. Algorithmus **??** stellt das Verfahren im Pseudo-Code dar.

Kurzschrittverfahren

Das Verfahren aus Algorithmus **??** ist als *Kurzschrittverfahren* bekannt, da „relativ kleine“ Newton-Schritte durchgeführt werden und sich der Parameter μ pro Iteration nur unwesentlich ändert. Wir beweisen nun das zentrale Konvergenzresultat.

Algorithmus 6.1 : Primal-duales Kurzschritt-Innere-Punkte Verfahren

Input : Zwei Lineare Programme (**??**) und (**??**), Startvektoren $x^{(0)} > 0$, $y^{(0)}$, $s^{(0)} > 0$ und ein Wert $\mu_0 > 0$, so dass

- $Ax^{(0)} = b$, $x^{(0)} > 0$ (strikte primale Zulässigkeit)
- $A^T y^{(0)} + s^{(0)} = c$, $s^{(0)} > 0$ (strikte duale Zulässigkeit)
- $r^{(0)} = X^{(0)} s^{(0)} - \mu_0 e$
- $\|r^{(0)}\|_2 \leq \mu_0/2$ (Nähe zum zentralen Pfad)

Gegeben sei auch ein Genauigkeitsprameter $\varepsilon > 0$.

Output : Strikt zulässige Vektoren $x^{(k)}$ für (**??**) und $(y^{(k)}, s^{(k)})$ für (**??**) mit $c^T x^{(k)} - b^T y^{(k)} \leq 2\varepsilon$

Setze $k := 0$; `/* Iterationszähler */`

while $\mu_k > \varepsilon/n$ **do**

Berechne den Newton-Schritt $(\Delta x, \Delta y, \Delta s)$ bei $(x^{(k)}, y^{(k)}, s^{(k)})$, der mit $r := r^{(k)} = X^{(k)} s^{(k)} - \mu_k e$ durch (**??**) definiert ist:

$$\begin{pmatrix} A & 0 & 0 \\ 0 & A^T & I \\ S & 0 & X \end{pmatrix} \cdot \begin{pmatrix} \Delta x \\ \Delta y \\ \Delta s \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ -r \end{pmatrix}. \qquad (\mathbf{??})$$

$(x^{(k+1)}, y^{(k+1)}, s^{(k+1)}) :=$
$(x^{(k)} + \Delta x, y^{(k)} + \Delta y, s^{(k)} + \Delta s)$;
$\mu_{k+1} := \left(1 - \frac{1}{6\sqrt{n}}\right) \mu_k$;
Setze $k := k + 1$;

end

return $x^{(k)}$, $y^{(k)}$ und $s^{(k)}$;

Theorem 6.16. *Das Kurzschrittverfahren aus Algorithmus* **??** *liefert eine Folge $(x^{(k)}, y^{(k)}, s^{(k)})_k$ von Vektoren, so dass $x^{(k)}$ strikt primal und $(y^{(k)}, s^{(k)})$ strikt dual zulässig sind.*

Das Verfahren terminiert nach höchstens $6\sqrt{n} \ln n\mu_0/\varepsilon$ Iterationen mit strikt zulässigen Vektoren x und (y, s), so dass $c^T x - b^T y \leq 2\varepsilon$.

Beweis. Um den ersten Teil des Satzes zu beweisen, genügt es nach Korollar **??** zu zeigen, dass

$$\|r^{(k)}\|_2 \leq \mu_k/2 \text{ für } k = 0, 1, \ldots \qquad (6.53)$$

gilt. Diese Aussage zeigen wir durch Induktion nach der Anzahl der Iterationen k. Für $k = 0$ ist die Aussage nach Voraussetzung erfüllt.

Im Induktionsschritt $k \to k+1$ müssen wir zeigen, dass

$$\|r^{(k+1)}\|_2 = \|X^{(k+1)}s^{(k+1)} - \mu_{k+1}e\| \leq \frac{1}{2}\mu_{k+1}$$

gilt. Wir haben

$$\begin{aligned} r^{(k+1)} &= X^{(k+1)}s^{(k+1)} - \mu_{k+1}e \\ &= \underbrace{X^{(k+1)}s^{(k+1)} - \mu_k e}_{=:r^+} + (\mu_k - \mu_{k+1})e \\ &= r^+ + \frac{\mu_k}{6\sqrt{n}}e. \end{aligned}$$

Für die letzte Gleichung haben wir die Definition von μ_{k+1} als $\mu_{k+1} = \left(1 - \frac{1}{6\sqrt{n}}\right)\mu_k$ benutzt. Nach Lemma **??** gilt $\|r^+\|_2 \leq \mu_k/4$. Daher folgt

$$\begin{aligned} \|r^{(k+1)}\|_2 &\leq \|r^+\|_2 + \frac{\mu_k}{6\sqrt{n}}\|e\|_2 \\ &\leq \frac{\mu_k}{4} + \frac{\mu_k}{6} \\ &= (\frac{1}{4} + \frac{1}{6})\mu_k \\ &= \frac{5}{12}\mu_k = \frac{1}{2}\cdot\frac{5}{6}\mu_k. \end{aligned}$$

Da $\mu_{k+1} \geq 5/6\mu_k$ folgt damit letztendlich $\|r^{(k+1)}\|_2 \leq 5/12 \cdot 6/5\mu_{k+1} = \mu_{k+1}/2$ wie behauptet und (**??**) ist bewiesen.

Wir schätzen jetzt die Anzahl der Iterationen bis zum Abbruch ab. Der Algorithmus stoppt, wenn $\mu_k \leq \varepsilon/n$ gilt. Da

$\mu_k = (1 - 1/(6\sqrt{n}))^k\mu_0$ ist dies äquivalent zu

$$\begin{aligned} & \left(1 - \frac{1}{6\sqrt{n}}\right)^k \mu_0 && \leq \frac{\varepsilon}{n} \\ \Leftrightarrow & \ln\frac{\varepsilon}{n\mu_0} && \geq k \cdot \ln\left(1 - \frac{1}{6\sqrt{n}}\right) \\ \Leftrightarrow & -\ln\frac{n\mu_0}{\varepsilon} && \geq k \cdot \ln\left(1 - \frac{1}{6\sqrt{n}}\right) \\ \Leftrightarrow & \frac{\ln\frac{n\mu_0}{\varepsilon}}{-\ln\left(1 - \frac{1}{6\sqrt{n}}\right)} && \leq k. \end{aligned} \tag{6.54}$$

Aus der Ungleichung $\ln(1-\tau) \leq -\tau$, die für $\tau < 1$ gilt, folgt $-\ln\left(1 - \frac{1}{6\sqrt{n}}\right) \geq \frac{1}{6\sqrt{n}}$. Damit ist (**??**) auf jeden Fall erfüllt, wenn $k \geq 6\sqrt{n}\ln\frac{n\mu_0}{\varepsilon}$. Also, bricht der Algorithmus nach höchstens $6\sqrt{n}\ln\frac{n\mu_0}{\varepsilon}$ Iterationen ab.

Seien $x = x^{(k)}$, $y = y^{(k)}$, $s = s^{(k)}$ und $\mu = \mu_k > 0$ die Werte bei Abbruch des Algorithmus. Nach Konstruktion des Algorithmus gilt dann $\mu \leq \varepsilon/n$. Weiterhin haben wir

$$\begin{aligned} c^T x - b^T y &= x^T s && \text{(nach Lemma ??)} \\ &= e^T (Xs) \\ &= e^T (\mu e + r) && \text{(nach Definition des Residuums in (??))} \\ &= \mu n + \sum_{i=1}^{n} r_i \\ &\leq \mu n + \sum_{i=1}^{n} |r_i| \\ &\leq \varepsilon + \sum_{i=1}^{n} |r_i|. && \text{(da } \mu \leq \varepsilon/n). \end{aligned} \tag{6.55}$$

Wir wenden die Cauchy-Schwarzsche-Ungleichung auf $\sum_{i=1}^{n} |r_i|$ an. Diese besagt, dass

$$\sum_{i=1}^{n} |r_i| = \sum_{i=1}^{n} |r_i| \cdot 1 \leq \|r\|_2 \|e\|_2 = \|r\|_2 \cdot \sqrt{n}. \tag{6.56}$$

Nach (**??**) gilt $\|r\|_2 \leq \mu/2$, also folgt aus (**??**) und (**??**)

$$c^T x - b^T y \leq \varepsilon + \sqrt{n}\frac{\mu}{2} \overset{\mu \leq \varepsilon/n}{\leq} \varepsilon + \frac{\varepsilon}{2\sqrt{n}} \leq 2\varepsilon.$$

Dies war zu zeigen. □

Für das Innere-Punkte Verfahren aus Algorithmus **??** müssen wir noch zwei zentrale Punkte klären, um es zu einem kompletten einsetzbaren Verfahren zu machen:

Initialisierung: Wie finden wir geeignete Startwerte?

Terminieren: Wenn wir geeignete Startwerte haben, können wir nach Satz ?? das Verfahren durchführen und durch Vorgabe eines geeigneten Genauigkeitsparameters $\varepsilon > 0$ Lösungen erzeugen, die „beliebig nahe“ an einer primalen bzw. dualen Optimallösung liegen (da die Dualitätslücke beliebig klein wird). Wie finden wir aber Optimallösungen von (P) und (D) *exakt*?

6.2.6 Terminieren

Unser Kurzschrittverfahren aus Algorithmus **??** liefert (bei geeigneter Initialisierung) primal und dual zulässige

Lösungen, die beliebig dicht an den entsprechenden Optimallösungen liegen. In diesem Abschnitt zeigen wir, wie man „fast optimale“ Lösungen zu einer optimalen Lösung „runden“ kann.

In diesem Abschnitt machen wir zusätzlich folgend Annahme:

Voraussetzung 6.17. *Alle Eingabedaten des Linearen Programms (P), d.h. A, b, c sind ganzzahlig.*

Diese Annahme kann man erzwingen, indem man bei rationalen Eingabedaten alle Werte mit dem kleinsten gemeinsamen Nenner multipliziert.

Die *Eingabegröße* (einer Instanz) des Linearens Programms (P) ist dann:

$$L := \sum_{j=1}^{n} \langle c_j \rangle + \sum_{i=1}^{m} \langle b_i \rangle + \sum_{i=1}^{m} \sum_{j=1}^{n} \langle a_{ij} \rangle.$$

Man beachte, dass L eine untere Schranke für die Anzahl der Bits ist, die zum Codieren der Eingabedaten notwendig ist.

Unser Ziel wird es sein, zu zeigen, dass wir in unserem Verfahren aus Algorithmus **??**

$$\varepsilon := 2^{-2L-2}$$

setzen können, und dann aus der Lösung, die das Verfahren bei Abbruch liefert, „ganz einfach“ eine optimale Basislösung bestimmen können. Dazu betrachten wir zunächst einmal die Darstellung von Basislösungen.

Lemma 6.18. *Sei x eine zulässige Basislösung von (P). Dann ist jeder Eintrag x_i von $x = (x_1, \ldots, x_n)^T$ eine rationale Zahl $x_i = p_i/q_i$ mit*

$$|x_i| \leq 2^L \quad \text{und } |q_i| \leq 2^L.$$

Beweis. Sei B eine zu x gehörende Basis, wobei wir ohne Beschränkung der Allgemeinheit $B = (1, \ldots, m)$ annehmen können. Dann gilt $A_{\cdot B} = (A_{\cdot 1}, \ldots, A_{\cdot m})$. Nach der Cramerschen Regel folgt:

$$x_i = \frac{\det(A_{\cdot 1}, \ldots, A_{\cdot i-1}, b, A_{\cdot i+1}, \ldots, A_{\cdot m})}{\det A_{\cdot B}}. \qquad (6.57)$$

Die Matrix $A_{\cdot B}$ ist eine ganzzahlige Matrix, deren Determinante wir nach der Hadamard-Ungleichung wie folgt beschränken können:

$$|\det A_{\cdot B}| \leq \prod_{i=1}^{m} \|A_{\cdot i}\|_2 \tag{6.58}$$

Wir haben für $a \in \mathbb{Z}$ die Ungleichung $|a| \leq 2^{\langle a \rangle - 1} - 1$ und für jeden ganzzahligen Vektor $d = (d_1, \ldots, d_m)^T \in \mathbb{R}^m$

$$1 + \|d\|_2 \leq 1 + \|d\|_1 = 1 + \sum_{i=1}^{m} (1 + |d_i|) \leq \prod_{i=1}^{m} (1 + |d_i|) \leq \prod_{i=1}^{m} 2^{\langle d_i \rangle - 1} = 2^{\langle d \rangle - n}. \tag{6.59}$$

Daher folgt aus (**??**), dass

$$1 + |\det A_{\cdot B}| \leq 1 + \prod_{i=1}^{m} \|A_{\cdot i}\|_2 \leq \prod_{i=1}^{m} (1 + \|A_{\cdot i}\|_2) \leq \prod_{i=1}^{m} 2^{\langle A_{\cdot i} \rangle - m} = 2^{\langle A_{\cdot B} \rangle - m^2} \leq 2^L.$$

Aus (**??**) ergibt sich dann: $|q_i| \leq 2^L$. Eine analoge Rechnung für die Matrix $(A_{\cdot 1}, \ldots, A_{\cdot i-1}, b, A_{\cdot i+1}, \ldots, A_{\cdot m})$ zeigt $|p_i| \leq 2^L$ und damit auch $|x_i| = |p_i/q_i| \leq 2^L$. □

Das obige Ergebnis über die Darstellung von Basislösungen können wir jetzt benutzen, um zu zeigen, dass sich Zielfunktionswerte von zwei Basislösungen entweder gleich sind oder sich um mehr als 2^{-2L} unterscheiden.

Korollar 6.19. *Seien x und x' zwei zulässige Basislösungen für (P) mit $|c^T x - c^T x'| < 2^{-2L}$. Dann gilt $c^T x = c^T x'$.*

Beweis. Wir nehmen an, dass $c^T x \neq c^T x'$ gilt. Der Vektor $c = (c_1, \ldots, c_n)^T$ der Zielfunktion von (P) ist ganzzahlig und hat daher Einträge vom Betrag höchstens 2^L. Nach Lemma **??** ist jeder Eintrag von x und x' eine rationale Zahl mit einem Nenner vom Betrag höchstens 2^L. Also gilt $c^T x = p/q$ und $c^T x' = p'/q'$ mit $|q| \leq 2^L$ und $|q'| \leq 2^L$. Daher folgt

$$|c^T x - c^T x'| = \frac{|pq' - p'q|}{|qq'|} \geq \frac{|pq' - p'q|}{2^L \cdot 2^L} = 2^{-2L} |pq' - p'q| \geq 2^{-2L},$$

wobei die letzte Ungleichung aus $c^T x \neq c^T x'$ folgt. Dies widerspricht der Voraussetzung, dass $|c^T x - c^T x'| < 2^{-2L}$. □

Damit haben wir alle Hilfsmittel zusammen, um das Terminieren des Kurzschritt-Verfahrens genauer zu analysieren. Wir setzen in Algorithmus ?? die Abbruchgenauigkeit auf $\varepsilon := 2^{-2L-2}$. Seien x und y die Lösungen, die das

Verfahren bei Abbruch liefert, und x^* eine Optimallösung von (P). Es gilt dann:

$$\begin{aligned} \cdot 2^{-2L-1} = 2\varepsilon &\geq c^T x - b^T y \quad \text{(nach Satz ??)} \\ &\geq c^T x - c^T x^* \quad \text{(schwache Dualität, Lemma \textbf{??})} \end{aligned}$$

In Lemma **??** hatten wir bereits gezeigt, wie wir aus x in $\mathcal{O}(n^3)$ Zeit eine Basislösung $\bar{x}$ konstruieren können, so dass $c^T\bar{x} \leq c^T x$. Für diese Basislösung gilt also

$$0 \leq c^T\bar{x} - c^T x^* \leq c^T x - c^T x^* \leq 2^{-2L-1} < 2^{-2L}.$$

Nach Korollar **??** folgt $c^T\bar{x} = c^T x^*$ und $\bar{x}$ ist eine Optimallösung von (P). Nach Satz ?? ist die Gesamtanzahl der Iterationen bis zum Abbruch (bei geeigneter Initialisierung)

$$6\sqrt{n}\ln n\mu_0/\varepsilon = 6\sqrt{n}(\ln\mu_0 - \ln 2^{-2L-2}) = \mathcal{O}(\sqrt{n}(\ln\mu_0 + L)). \tag{6.60}$$

6.2.7 Initialisierung

Zur Initialisierung des Innere-Punkte-Verfahrens benötigen wir Startvektoren $x^{(0)}$, $y^{(0)}$ und $s^{(0)}$ mit folgenden Eigenschaften:

- $Ax^{(0)} = b$, $x^{(0)} > 0$ (strikte primale Zulässigkeit)
- $A^T y^{(0)} + s^{(0)} = c$, $s^{(0)} > 0$ (strikte duale Zulässigkeit)
- $\|X^{(0)}s^{(0)} - \mu_0 e\|_2 \leq \mu_0/2$ (Nähe zum zentralen Pfad)

Der Trick ist es jetzt, das Problem (P) äquivalent zu einem neuen Problem aufzublähen, für das wir entsprechende Startlösungen einfach finden können.

Sei $L = L(A, b, c)$ die Codierungslänge von (P)mit n Variablen und m Ungleichungen. Wir definieren:

$$\begin{aligned} \tilde{n} &:= n + 2 \\ \tilde{m} &:= m + 1 \\ \alpha &:= 2^{4L} \\ \lambda &:= 2^{2L} \\ K_b &:= \alpha\lambda(n+1) - \lambda c^T e \\ K_c &:= \alpha\lambda = 2^{6L} \end{aligned}$$

Damit schreiben wir ein neues Lineares Programm

$$\begin{aligned}
\min \quad & c^T x + K_c x_{n+2} && \text{(6.61a)}\\
& Ax + (b - \lambda Ae)x_{n+2} = b && \text{(6.61b)}\\
& (\alpha e - c)^T x + \alpha x_{n+1} = K_b && \text{(6.61c)}\\
& x \geq 0, x_{n+1} \geq 0, x_{n+2} \geq 0. && \text{(6.61d)}
\end{aligned}$$

Mit

$$\tilde{b} := \begin{pmatrix} b \\ K_b \end{pmatrix}, \quad \tilde{c} = \begin{pmatrix} c \\ 0 \\ K_c \end{pmatrix}, \quad \tilde{A} = \begin{pmatrix} A & 0 & b - \lambda Ae \\ (\alpha e - c)^T & \alpha & 0 \end{pmatrix} \tag{6.62}$$

und $\tilde{x} = \begin{pmatrix} x \\ x_{n+1} \\ x_{n+2} \end{pmatrix}$, hat das Lineare Program (**??**) die Form von (P):

$$\begin{aligned}
\min \quad & \tilde{c}^T \tilde{x} \\
& \tilde{A}\tilde{x} = \tilde{b} \\
& \tilde{x} \geq 0.
\end{aligned}$$

Man beachte, dass Rang $\tilde{A} = \tilde{m} = m+1$, da wir Rang $A = m$ angenommen hatten und $\alpha \neq 0$ gilt. Das duale Programm zu (**??**) ist dann:

$$\begin{aligned}
\max \quad & b^T y + K_b y_{m+1} && \text{(6.63a)}\\
& A^T y + (\alpha e - c)y_{m+1} + s = c && \text{(6.63b)}\\
& \alpha y_{m+1} + s_{n+1} = 0 && \text{(6.63c)}\\
& (b - \lambda Ae)^T y + s_{n+2} = K_c && \text{(6.63d)}\\
& s \geq 0, s_{n+1} \geq 0, s_{n+2} \geq 0. && \text{(6.63e)}
\end{aligned}$$

Wir zeigen, dass wir leicht strikt zulässige Lösungen zu (**??**) und (**??**) angeben können, die sogar auf dem zentralen Pfad zu diesen Problemen liegen. Betrachte dazu

$$\tilde{x} := (x, x_{n+1}, x_{n+2})^T := (\lambda, \ldots, \lambda, 1)^T \in \mathbb{R}^{n+2} = \mathbb{R}^{\tilde{n}} \tag{6.64a}$$

$$\tilde{y} := (y, y_{m+1})^T = (0, \ldots, 0, -1)^T \in \mathbb{R}^{m+1} = \mathbb{R}^{\tilde{m}} \tag{6.64b}$$

$$:= (s, s_{n+1}, s_{n+2}) := (\alpha, \ldots, \alpha, \alpha\lambda)^T \in \mathbb{R}^{n+2} = \mathbb{R}^{\tilde{n}}. \tag{6.64c}$$

Dann gilt $\tilde{x} > 0$ und > 0. Darüberhinaus gilt:

$$Ax + (b - \lambda Ae)x_{n+2} = A\lambda e + (b - \lambda Ae) = b$$
$$(\alpha e - c)^T x + \alpha x_{n+1} = (\alpha e - c)^T \lambda e + \alpha\lambda = \alpha\lambda n - \lambda c^T e + \alpha\lambda = K_b.$$

Also ist $\tilde{x}$ zulässig für (**??**). Analog rechnen wir:

$$A^T y + (\alpha e - c)y_{m+1} + s = A^T 0 + (\alpha e - c)(-1) + (\alpha, \ldots, \alpha)^T = c$$
$$\alpha y_{m+1} + s_{n+1} = \alpha(-1) + \alpha = 0$$
$$(b - \lambda Ae)^T y + s_{n+2} = (b - \lambda Ae)^T 0 + \alpha\lambda = \alpha\lambda = K_c.$$

Somit haben wir strikt zulässige Lösungen von (**??**) und (**??**) gefunden.

Sei $\tilde{\mu} := \alpha\lambda$. Wegen

$$\tilde{X} = \begin{pmatrix} \lambda & & & & \\ & \lambda & & & \\ & & \ddots & & \\ & & & \lambda & \\ & & & & 1 \end{pmatrix} \cdot \begin{pmatrix} \alpha \\ \alpha \\ \vdots \\ \alpha \\ \alpha\lambda \end{pmatrix} = \begin{pmatrix} \alpha\lambda \\ \alpha\lambda \\ \vdots \\ \alpha\lambda \end{pmatrix} = \tilde{\mu}e,$$

liegen die konstruieren Vektoren auf den zentralen Pfad (für die Parameterwahl von $\tilde{\mu} = \alpha\lambda$). Wenn wir $\mu_0 := \tilde{\mu}$ setzen und mit den Vektoren aus (1.21) starten, so können wir nach Satz ?? strikt zulässige Lösungen für (**??**) und (**??**) finden, so dass $c^T\tilde{x} - b^T y \leq 2\varepsilon$. Die benötigte Anzahl der Iterationen ist höchstens:

$$6\sqrt{n}\ln n\tilde{\mu}/\varepsilon = 6\sqrt{n}\ln(\alpha\lambda) = 6\sqrt{n}\ln 2^{6L}/\varepsilon$$

Nach Korollar **??** und $\varepsilon := 1/4 \cdot 2^{2\tilde{L}}$ erhalten wir optimale Lösungen nach dem Runden. Die Gesamtzahl der Iterationen ist dann:

$$6\sqrt{n}\ln 2^{6L}/2^{-2\tilde{L}-1} = \mathcal{O}(\sqrt{n}(L + \tilde{L})).$$

Theorem 6.20. *Optimallösungen von (**??**) und (**??**) kann man in $\mathcal{O}(\sqrt{n}\tilde{L})$ Iterationen des Kurzschrittverfahrens finden. Die Gesamtzeit ist $\mathcal{O}(n^{2.5}\tilde{L})$, da wir in jeder Iteration ein lineares Gleichungssystem in $\mathcal{O}(n^2)$ Zeit lösen müssen.* □

Das gesamte Verfahren wird dann noch mit folgenden Ergebnissen (deren Beweise technisch sind) vervollständigt:

Lemma 6.21. *Sei $\tilde{L} = \tilde{L}(\tilde{A}, \tilde{b}, \tilde{c})$ die Eingabegröße von (**??**) und $L = L(A, b, c)$ die Größe von (P). Dann gilt: $L \leq \tilde{L} \leq 36L$.* □

Korollar 6.22. *Wir können Optimallösungen von (??) und (??) in $\mathcal{O}(n^{2.5}L)$ Zeit finden.* □

Theorem 6.23. *Se $\tilde{x}$ optimal für (??) und $(\tilde{y},)$ optimal für (??). Then hat (P) genau dann eine Optimallösung, wenn $\tilde{x}_{n+2} = 0$ und $_{n+1} = 0$. In diesem Fall sind $x = (\tilde{x}_1, \ldots, \tilde{x}_n)^T$ und $((\tilde{y}_1, \ldots, \tilde{y}_m)^T, (_1, \ldots, _n)^T)$ Optimallösungen von (P) und (D).* □

Theorem 6.24. *Wir können in Zeit $\mathcal{O}(n^{2.5}L)$ Optimallösungen von (P) und (D) bestimmen bzw. feststellen, dass keine existieren.* □

6.3 Die Ellipsoid-Methode

Die Ellipsoid-Methode wurde ursprünglich mehr für die konvexe als für die lineare Programmierung entwickelt. Einen kleinen historischen Überblick findet man z.B. in [**?**]. Khachiyan [**?**] passte diese Methode 1979 der Linearen Programmierung an und zeigte, wie man das Zulässigkeitsproblem für ein System linearer Ungleichungen in polynomialer Zeit lösen kann. Die Ellipsoid-Methode liefert das Gerüst, um die Äquivalenz des Separierungs- und des Optimierungsproblems zu zeigen. Bis heute ist dies auch der einzige bekannte Weg diese Äquivalenz zu zeigen. Viele schöne Sätze über diese Äquivalenz basieren auf der Ellipsoid-Methode (siehe z.B. [**?**]) und viele von ihnen können bis heute nicht auf anderen Wegen bewiesen werden. In diesem Abschnitt beschreiben wir die Grundform des Algorithmus und liefern einen Beweis für seine Korrektheit, falls Rundungsfehler ausgeschlossen werden. Zuerst einige grundlegene Eigenschaften von Ellipsoiden.

Definition 6.25. Für eine symmetrische, positiv definite Matrix $A \in \mathbb{R}^{n\times n}$ und einen Vektor $a \in \mathbb{R}^n$ heißt die Menge

$$E(A,a) \;=\; \left\{ x \in \mathbb{R}^n \;\middle|\; (x-a)^T A^{-1} (x-a) \;\leq\; 1 \right\}$$

Ellipsoid. Der Vektor a heißt Zentrum des Ellipsoids.

Altes Kommando `df` hier verwendet

Altes Kommando `df` hier verwendet

Beachte, dass $E(A,a)$ durch A und a eindeutig bestimmt ist, d.h. $E(A,a) \neq E(A',a')$ falls $A \neq A'$ oder $a \neq a'$. Ein Spezialfall eines Ellipsoids ist die Einheitskugel $E(I,0)$.

Bemerkung 6.26.

(a) Das Volumen eines Ellipsoids ist gegeben durch

$$\mathrm{vol}(E(A,a)) \;=\; \sqrt{\det(A)} \;\cdot\; V_n$$

wobei V_n das Volumen der Einheitskugel im $\mathbb{R}^n$ bezeichnet.

(b) Zu jeder positiv definiten Matrix A existiert eine eindeutig bestimmte positiv definite Matrix $A^{1/2}$ mit $A = A^{1/2}\,A^{1/2}$.

(c) Für jedes Ellipsoid $E(A,a)$ und jeden Vektor $c \in \mathbb{R}^n$ gilt

$$-\sqrt{c^T A c} \;\le\; c^T(x-a) \;\le\; \sqrt{c^T A c} \qquad \forall\, x \in E(A,a).$$

Die Grundidee der Ellipsoid-Methode ist es, einen Ellipsoid in zwei Teile zu zerlegen und dann eines der beiden Teile mit einem neuen, kleineren Ellipsoid zu umschließen.

Satz 6.27. *Sei $A \in \mathbb{R}^{n\times n}$ eine beliebige symmetrische, positiv definite Matrix und $a \in \mathbb{R}^n$ ein beliebiger Vektor. Betrachte einen weiteren Vektor $c \in \mathbb{R}^n \setminus \{0\}$ und die Menge $S = E(A,a) \cap \{x \in \mathbb{R}^n \,\big|\, c^T x \le c^T a\}$. Definiere nun*

$$\bar{a} \;=\; a \;-\; \frac{1}{n+1}\,\frac{Ac}{\sqrt{c^T A c}}\,, \tag{6.65}$$

$$\bar{A} \;=\; \frac{n^2}{n^2-1}\left(A \;-\; \frac{2}{n+1}\,\frac{Ac(Ac)^T}{c^T A c}\right). \tag{6.66}$$

Dann ist $\bar{A}$ ebenfalls symmetrisch und positiv definit und es gilt $S \subseteq E(\bar{A},\bar{a})$.

Beweis. Zuerst wollen wir festhalten, dass $\bar{A}$ eine explizit darstellbare Inverse besitzt:

$$\bar{A}^{-1} \;=\; \frac{n^2-1}{n^2}\left(A^{-1} \;+\; \frac{2}{n-1}\,\frac{cc^T}{c^T A c}\right). \tag{6.67}$$

Dies kann man durch eine direkte Multiplikation überprüfen. Da eine Matrix B genau dann positiv definit ist, wenn die Inverse B^{-1} positiv definit ist, genügt es zu zeigen, dass $\bar{A}^{-1}$ positiv definit ist. Dies ist jedoch einfach zu sehen, da A^{-1} positiv definit und die Matrix cc^T positiv semidefinit ist. Nach Konstruktion ist $\bar{A}$ auch symmetrisch und daher ist die Definition von $E(\bar{A},\bar{a})$ sinnvoll. Damit bleibt nur noch $S \subseteq E(\bar{A},\bar{a})$ zu zeigen. Für ein beliebiges $x \in S$ gilt:

$$
\begin{aligned}
&(x-\bar{a})^T \bar{A}^{-1} (x-\bar{a}) \\
&= \left((x-a) + \frac{1}{n+1}\frac{Ac}{\sqrt{c^T Ac}}\right)^T \left(\frac{n^2-1}{n^2}\left(A^{-1} + \frac{2}{n-1}\frac{cc^T}{c^T Ac}\right)\right) \\
&\qquad\qquad \left((x-a) + \frac{1}{n+1}\frac{Ac}{\sqrt{c^T Ac}}\right) \\
&= \frac{n^2-1}{n^2}\left((x-a)^T A^{-1}(x-a) + \frac{2}{n-1}\frac{c^T(x-a)}{\sqrt{c^T Ac}}\left(1 + \frac{c^T(x-a)}{\sqrt{c^T Ac}}\right) + \frac{1}{n^2-1}\right) \\
&\leq \frac{n^2-1}{n^2}\left((x-a)^T A^{-1}(x-a) \; + \; \frac{1}{n^2-1}\right) \\
&\leq \frac{n^2-1}{n^2}\left(1 \; + \; \frac{1}{n^2-1}\right) = 1.
\end{aligned}
$$

Dabei stammt die erste Ungleichung aus der Bemerkung **??** und die zweite benutzt die Tatsache, dass $x \in E(A,a)$ gilt. Damit gilt nun $x \in E(\bar{A},\bar{a})$ und der Beweis ist vollständig.

Wie wir gleich sehen werden, benutzt die Ellipsoid-Methode hauptsächlich die beiden Gleichungen (**??**) und (**??**). Natürlich ist es nicht schwer, ein Ellipsoid zu finden, das S enthält, aber das in Satz **??** definierte Ellipsoid $E(\bar{A},\bar{a})$ ist minimal bezüglich der Eigenschaft S zu enthalten und $E(\bar{A},\bar{a})$ ist bezüglich dieser Eigenschaft auch eindeutig (siehe z.B. [?]). Wir brauchen diese beiden Eigenschaften jedoch nicht zur Durchführung der Ellipsoid-Methode. Betrachte den folgenden Algorithmus (ξ und p werden später definiert):

Ellipsoid-Methode

Input: $\epsilon \in \mathbb{Q}_+$, ein durch ein Separierungsorakel gegebenes, wohlbeschriebenes, beschränktes Polyeder (P,n,φ), das SEP für P löst.

Output: Entweder

(i) ein zulässiger Vektor $y \in P$ oder

(ii) eine symmetrische, positiv definite Matrix $A \in \mathbb{Q}^{n\times n}$ und ein Vektor $a \in \mathbb{Q}^n$ mit $P \subseteq E(A,a)$ und $\text{vol}(E(A,a)) < \epsilon$.

(1) Initialisierung:

$$
\begin{aligned}
R &= 2^{4n^2\varphi},\\
N &= \lceil (2n+1)\,(|\log\epsilon| + n|\log(2R)|)\rceil,\\
a_0 &= 0,\\
A_0 &= R^2\, I.
\end{aligned}
$$

(2) For $k = 0$ To N Do
(3) Rufe das Orakel (SEP) für $y = a_k$.
(4) Falls $y \in P$, **Stop** (mit Antwort (i)).
(5) Falls $y \notin P$:
(SEP) liefert einen Vektor $c \in \mathbb{Q}^n$ mit $c^T y > \max\{c^T x \,\big|\, x \in P\}$.
(6) Berechne

$$
\begin{aligned}
a_{k+1} &=_p a_k - \frac{1}{n+1}\,\frac{A_k c}{\sqrt{c^T A_k c}},\\
A_{k+1} &=_p \xi\,\frac{n^2}{n^2-1}\left(A_K - \frac{2}{n+1}\,\frac{A_k c c^T A_k}{c^T A_k c}\right),
\end{aligned}
$$

dabei bezeichne $=_p$ eine Rundung der Ergebnisse auf die p-te
Nachkommastelle.
(7) End For
(8) Gib A_N und a_N aus.

Altes Kommando `df` hier verwendet

ξ heißt Blow-Up Parameter. Wird mit unendlicher Genauigkeit ($p = \infty$) und ohne Ausdehnung des Ellipsoids in Schritt (6) gearbeitet ($\xi = 1$), so bekommen wir genau die Update-Formeln aus (**??**) und (**??**). Der folgende Satz garantiert uns die Korrektheit der Ellipsoid-Methode unter diesen Voraussetzungen.

Satz 6.28. *Gilt $p = \infty$ und $\xi = 1$, so arbeitet der Algorithmus* **??** *korrekt und terminiert nach einer polynomialen Anzahl von Iterationen.*

Beweis. Die polynomiale Beschränktheit der Anzahl der Iterationen sieht man leicht ein, da N polynomial in φ, n und $\langle\epsilon\rangle$ ist.

Aus Satz **??** folgt, dass sich die Symmetrie und die positive Definitheit von A_k auf A_{k+1} überträgt. Da A_0 symmetrisch und positiv definit ist, gilt dies auch für alle A_k, $k \in \{0, 1, \ldots, N\}$.

Nach Lemma **??** hat P eine Eckenkomplexität von höchstens $4n^2\varphi$. Weiterhin ist P beschränkt und mit Lemma **??** gilt dann $P \subseteq E(A_0, a_0)$. Liefert (SEP) in Schritt

(5) einen Vektor c mit $c^T y > \max\{c^T x \,|\, x \in P\}$, so haben wir $P \subseteq E(A_k, a_k) \cap \{x \in \mathbb{R}^n \,|\, c^T x \leq c^T a_k\}$. Satz **??** liefert dann sofort $P \subseteq E(A_{k+1}, a_{k+1})$ und damit $P \subseteq E(A_k, a_k)\ \forall\, k \in \{0, 1, \ldots, N\}$.

Bleibt noch zu zeigen, dass $\mathrm{vol}(E(A_N, a_N)) < \epsilon$ gilt, falls der Algorithmus **??** nicht im vierten Schritt abbricht. Angenommen, es gilt

$$\frac{\mathrm{vol}(E(A_{k+1}, a_{k+1}))}{\mathrm{vol}(E(A_k, a_k))} \;<\; e^{-\frac{1}{2(n+1)}} \quad \forall\, k \in \{0, 1, \ldots, N\}. \tag{6.68}$$

Dann erhält man

$$\begin{aligned}
\mathrm{vol}(E(A_N, a_N)) \;&<\; \left(e^{-\frac{1}{2(n+1)}}\right)^N \mathrm{vol}(E(A_0, a_0)) \\
&\leq\; e^{-(|\log \epsilon| + n|\log(2R)|)}\,(2R)^n \\
&\leq\; 2^{-(|\log \epsilon| + n|\log(2R)|)}\,(2R)^n \\
&\leq\; 2^{-|\log \epsilon|} \\
&\leq\; \epsilon,
\end{aligned}$$

wobei wir die Tatsache benutzt haben, dass

$$E(A_0, a_0) \subseteq \{x \in \mathbb{R}^n \,|\, -R \leq x_i \leq R\ \ \forall\, i \in \{1, 2, \ldots, n\}\}$$

gilt.

Bleibt noch die Gültigkeit der Gleichung (**??**) zu zeigen. Dazu benutzen wir das Verhältnis der beiden Determinanten von A_k bzw. A_{k+1}.

$$\begin{aligned}
\det(A_{k+1}) \;&=\; \det\left(\frac{n^2}{n^2-1}\left(A_k \,-\, \frac{2}{n+1}\,\frac{A_k c c^T A_k}{c^T A_k c}\right)\right) \\
&=\; \left(\frac{n^2}{n^2-1}\right)^n \det\left(A_k^{1/2}\left(I \,-\, \frac{2}{n+1}\,\frac{A_k^{1/2} c c^T A_k^{1/2}}{c^T A_k c}\right) A_k^{1/2}\right) \\
&=\; \left(\frac{n^2}{n^2-1}\right)^n \det\left(I \,-\, \frac{2}{n+1} b b^T\right) \det(A_k) \\
\text{mit } b \;&=\; \frac{A_k^{1/2} c}{\sqrt{c^T A_k c}}.
\end{aligned}$$

Untersuchen wir nun die Determinante von $I - \frac{2}{n+1} b b^T$. Dabei benutzen wir die Eigenschaft von Determinanten,

dass man ein Vielfaches einer Zeile zu einer anderen addieren kann, ohne die Determinante zu verändern. Für jede Zeile $i\,(i \in \{2,3,\ldots,n\})$ multiplizieren wir die erste Zeile mit $-\frac{b_i}{b_1}$ und addieren dies zur i-ten Zeile. Es entsteht die Matrix

$$\begin{bmatrix} 1-\frac{2}{n+1}b_1^2 & -\frac{2}{n+1}b_1 d^T \\ -\frac{1}{b_1}d & I_{n-1} \end{bmatrix} \qquad \text{mit } d=(b_2,b_3,\ldots,b_n)^T.$$

Für den oben definierten Vektor b gilt $||b||_2 = 1$. Damit lässt sich die Matrix weiter zerlegen. Aus der Darstellung

$$\begin{bmatrix} 1-\frac{2}{n+1}b_1^2 & -\frac{2}{n+1}b_1 d^T \\ -\frac{1}{b_1}d & I_{n-1} \end{bmatrix} = \begin{bmatrix} 1 & -\frac{2}{n+1}b_1 d^T \\ 0 & I_{n-1} \end{bmatrix} \cdot \begin{bmatrix} 1-\frac{2}{n+1} & 0 \\ -\frac{1}{b_1}d & I_{n-1} \end{bmatrix}$$

lässt sich die Determinante sofort berechnen:

$$\det\left(I-\frac{2}{n+1}bb^T\right) = 1-\frac{2}{n+1}$$

und daraus folgt nun

$$\det(A_{k+1}) = \left(\frac{n^2}{n^2-1}\right)^n \left(1-\frac{2}{n+1}\right)\det(A_k).$$

Mit der Bemerkung **??**(a) und der Abschätzung $1+x < e^x \;\forall\, x \in \mathbb{R}\setminus\{0\}$ bekommen wir nun

$$\begin{aligned} \frac{\mathrm{vol}(E(A_{k+1},a_{k+1}))}{\mathrm{vol}(E(A_k,a_k))} &= \frac{\sqrt{\det(A_{k+1})}\,V_n}{\sqrt{\det(A_k)}\,V_n} \\ &= \sqrt{\left(\frac{n^2}{n^2-1}\right)^n \left(1-\frac{2}{n+1}\right)} \\ &= \left(\frac{n}{n+1}\right)\left(\frac{n^2}{n^2-1}\right)^{\frac{n-1}{2}} \\ &< e^{-\frac{1}{n+1}}\, e^{\frac{n-1}{2(n^2-1)}} \quad = \quad e^{-\frac{1}{2(n+1)}}. \end{aligned}$$

Dies entspricht genau (**??**) und damit ist der Beweis beendet.

Im Satz **??** machten wir die Annahme, dass alle Berechnungen exakt seien. Um einen polynomialen Algorithmus zu erhalten, müssen auch die Zahlen bzw. deren Kodierungslänge polynomial bleiben. Um dies zu erreichen,

müssen wir im sechsten Schritt die Anzahl der Nachkommastellen polynomial begrenzen, d.h. p muss polynomial in n, φ und $\langle\epsilon\rangle$ sein. Dieser Rundungsprozess bewirkt jedoch, dass sich das Zentrum a_{k+1} ein wenig bewegt und damit verliert das neue Ellipsoid $E(A_{k+1}, a_{k+1})$ die Eigenschaft P zu enthalten, zumindest dann, wenn wir weiterhin $\xi = 1$ wählen. Um diese Eigenschaft wieder herzustellen, müssen wir das neue Ellipsoid ein wenig vergrößern und wählen deshalb ein $\xi > 1$. Dabei kann es jedoch passieren, dass (**??**) keine Gültigkeit mehr hat und dass das Verhältnis zweier aufeinanderfolgender Ellipsoide zu nahe an den Wert eins rückt [?]. Der folgende Satz zeigt, dass uns die Wahl

$$\begin{aligned} N &= \lceil 5n(|\log\epsilon| + n|\log(2R)|)\rceil\,, \\ p &= 8N\,, \\ \xi &= 1 + \frac{n^2-3}{2n^4} \end{aligned} \tag{6.69}$$

die Polynomialität garantiert. Auch die Abschätzungen $P \subseteq E(A_k, a_k)$ für alle k und $\operatorname{vol}(E(A_{k+1}, a_{k+1}))/\operatorname{vol}(E(A_k, a_k)) \leq e^{-\frac{1}{5n}}$ bleiben erhalten, so dass A_N und a_N weiterhin die Eigenschaften des Outputs (ii) haben.

Theorem 6.29. *Für den Parametersatz aus (**??**) hat die Ellipsoid-Methode **??** polynomiale Laufzeit.*

Der Beweis ist länglich und technisch, der interessierte Leser findet Details in [?]. Ähnliche Beweise mit leicht geänderten Parametern findet man auch in [**?**] und [**?**]. Es sollte noch einmal erwähnt werden, dass der zurückgegebene Vektor c des (SEP)-Orakels ebenfalls polynomial ist. Dies ist Gegenstand der allgemeinen Annahmen über Orakel.

6.4 Separieren und Optimieren

Die Äquivalenz von Separieren und Optimieren ist eines der bedeutendsten Resultate der polyedrischen Kombinatorik (vgl. [**?**]). Es hat sowohl die Theorie der kombinatorischen und ganzzahligen Optimierung entscheidend beeinflusst, als auch die theoretische Rechtfertigung für Branch-and-Cut Verfahren (die derzeit erfolgreichste Methode zum Lösen $\mathcal{NP}$-schwerer praktischer Probleme der Diskreten Optimierung) ermöglicht.

In diesem Kapitel geben wir eine genaue Formulierung des Ergebnisses an und stellen die wesentlichen Beweisideen dar. Dies beinhaltet die Kodierung von Polyedern, die Ellipsoidmethode und die Theorie zur Approximation rationaler Zahlen. Wir beschränken uns in unseren Ausführungen auf volldimensionale und beschränkte Polyeder (das Originalresultat gilt für konvexe Körper). Literatur zu diesem Kapitel findet man z.B. in [?], [?] oder [?]. Wir beginnen mit der Definition der Probleme, die wir in diesem Kapitel untersuchen wollen.

Altes Kommando `df` hier verwendet

Problem 6.30. (Optimierungsproblem — OPT) Gegeben sei ein Polyeder $P \subseteq \mathbb{R}^n$ und ein Vektor $c \in \mathbb{Q}^n$.

(i) Bestätige, dass $P = \emptyset$ oder
(ii) Finde $y \in P$ mit $c^T y = \max\{c^T x \,|\, x \in P\}$ oder
(iii) bestätige, dass $\max\{c^T x \,|\, x \in P\}$ unbeschränkt ist, d.h. finde eine Extremale z von P mit $c^T z \geq 1$.

Unter unserer Annahme, dass P volldimensional und beschränkt sein soll, kommt natürlich nur Punkt (ii) in Frage, d.h. Problem **??** reduziert sich auf das Finden einer Optimallösung.

Altes Kommando `df` hier verwendet

Problem 6.31. (Separierungsproblem — SEP) Gegeben sei ein Polyeder $P \subseteq \mathbb{R}^n$ und ein Vektor $y \in \mathbb{Q}^n$. Entscheide, ob $y \in P$. Falls nicht, finde einen Vektor $c \in \mathbb{Q}^n$ mit $c^T y > \max\{c^T x \,|\, x \in P\}$.

Wir betrachten noch ein drittes Problem, das – wie wir noch sehen werden – dual zum Problem **??** ist.

Altes Kommando `df` hier verwendet

Problem 6.32. (Verletztheitsproblem — VIOL) Gegeben sei ein Polyeder $P \subseteq \mathbb{R}^n$, ein Vektor $c \in \mathbb{Q}^n$ und ein Skalar $\gamma \in \mathbb{Q}$. Entscheide, ob $c^T x \leq \gamma$ für alle $x \in P$. Falls nicht, finde einen Vektor $y \in P$ mit $c^T y > \gamma$.

Die Fragen, die uns in diesem Kapitel interessieren, kann man folgerndermaßen beschreiben: Angenommen, wir kennen eine Methode (im Folgenden werden wir von einem Orakel sprechen), die eines der drei Probleme löst. Ist es dann auch möglich, die beiden anderen Probleme in orakel-polynomialer Zeit zu lösen?

Altes Kommando `df` hier verwendet

Auf den ersten Blick scheint das Optimierungsproblem das schwerste zu sein. Könnten wir dies lösen, so könnten wir sicher auch das Verletztheitsproblem lösen. Aber könnten wir damit auch das Separierungsproblem lösen?

Altes Kommando `df` hier verwendet

Und umgekehrt, wenn wir separieren können, können wir dann auch optimieren?

In diesem Kapitel werden wir zeigen, dass alle drei Probleme in der Tat polynomial äquivalent sind. Wir werden im Folgenden die Probleme **??** bis **??** immer mit OPT, SEP und VIOL abkürzen. Schreiben wir Klammern um die Abkürzungen, so bezeichne dies ein Orakel für das jeweilige Problem, d.h. (OPT), (SEP) und (VIOL) sind Orakel für OPT, SEP und VIOL. Wir nehmen in diesem Kapitel $n \geq 2$ an.

6.4.1 Kettenbrüche

Das Problem, reelle Zahlen durch rationale Zahlen zu approximieren, ist ein altes und bekanntes Problem aus der Zahlentheorie. In diesem Abschnitt wollen wir das zweidimensionale Approximationsproblem angehen. Dieses Resultat benötigen wir im nächsten Abschnitt, wenn wir die Äquivalenz des Separierungs- und des Optimierungsproblems zeigen. Dazu betrachten wir folgendes

Problem 6.33. Gegeben sei eine Zahl $\alpha \in \mathbb{R}$ und ein $\epsilon \in \mathbb{Q}$, $0 < \epsilon < 1$.

Aufgabe: Finde ganze Zahlen $p, q \in \mathbb{Z}$ mit

$$1 \leq q \leq \frac{1}{\epsilon} \quad \text{und} \quad \left|\alpha - \frac{p}{q}\right| < \frac{\epsilon}{q}.$$

Auf den ersten Blick ist nicht einzusehen, dass solch eine rationale Zahl immer existiert, aber genau dies ist der Fall. Mehr noch, eine solche Zahl kann sogar in polynomialer Zeit bestimmt werden. Dazu dient der folgende

Kettenbruch-Algorithmus

Input: $\alpha \in \mathbb{Q}$, $\epsilon \in \mathbb{Q} \cap (0,1)$.

Output: p und q mit $1 \leq q \leq \frac{1}{\epsilon}$ und $\left|\alpha - \frac{p}{q}\right| < \frac{\epsilon}{q}$.

(1) Initialisierung:

$$\begin{aligned} \alpha_0 &= \alpha\,, & a_0 &= \lfloor \alpha \rfloor \\ g_{-2} &= 0\,, & g_{-1} &= 1\,, \\ h_{-2} &= 1\,, & h_{-1} &= 0\,, \\ i &= -1. \end{aligned}$$

(2) Führe die folgenden Schritte durch:

(3) $i = i + 1$
(4) $g_i = a_i g_{i-1} + g_{i-2}$
(5) $h_i = a_i h_{i-1} + h_{i-2}$
(6) Falls $h_i > \frac{1}{\epsilon}$ **Stop** (gib $p = g_{i-1}$ und $q = h_{i-1}$ aus).
(7) Falls $\alpha_i = a_i$ **Stop** (gib $p = g_i$ und $q = h_i$ aus).
(8) $\alpha_{i+1} = \dfrac{1}{\alpha_i - a_i}$
(9) $a_{i+1} = \lfloor \alpha_{i+1} \rfloor$
(10) Gehe zu (3).

Bevor wir die Korrektheit dieses Algorithmus beweisen, wollen wir uns ein kleines Beispiel ansehen. Dazu wollen wir den Wert $\sqrt{2} = 1,4142135...$ mit einer Genauigkeit von $\epsilon = 0,01$ durch eine rationale Zahl approximieren, d.h. gesucht sind

$$p, q \in \mathbb{N} \text{ mit } \left|\sqrt{2} - \frac{p}{q}\right| < \frac{0,01}{q}, \quad 1 \leq q \leq 100.$$

Ein erster Versuch mit $p = 141$ und $q = 100$ scheitert, da die Fehlerschranke nicht eingehalten wird, denn $\left|\sqrt{2} - \frac{141}{100}\right| = 0,0042135 \not< 0.0001$. Schauen wir uns das Ergebnis des Algorithmus **??** an.

(1) $\alpha_0 = 1,4142135$, $a_0 = 1$, $g_{-2} = 0$, $g_{-1} = 1$, $h_{-2} = 1$, $h_{-1} = 0$, $i = -1$
(3) $i = 0$
(4) $g_0 = 1 + 0 = 1$
(5) $h_0 = 0 + 1 = 1$
(8) $\alpha_1 = \frac{1}{1,4142135-1} = 2,4142139$
(9) $a_1 = 2$
(3) $i = 1$
(4) $g_1 = 2 + 1 = 3$
(5) $h_1 = 2 + 0 = 2$
(8) $\alpha_2 = \frac{1}{2,4142139-2} = 2,4142116$
(9) $a_2 = 2$
(3) $i = 2$
(4) $g_2 = 6 + 1 = 7$
(5) $h_2 = 4 + 1 = 5$
(8) $\alpha_3 = \frac{1}{2,4142116-2} = 2,4142250$
(9) $a_3 = 2$
(3) $i = 3$
(4) $g_3 = 14 + 3 = 17$
(5) $h_3 = 10 + 2 = 12$

(8) $\alpha_4 = \frac{1}{2{,}4142250-2} = 2{,}4141469$
(9) $a_4 = 2$
(3) $i = 4$
(4) $g_4 = 34 + 7 = 41$
(5) $h_4 = 24 + 5 = 29$
(8) $\alpha_5 = \frac{1}{2{,}4141469-2} = 2{,}4146022$
(9) $a_5 = 2$
(3) $i = 5$
(4) $g_5 = 82 + 17 = 99$
(5) $h_5 = 58 + 12 = 70$
(8) $\alpha_6 = \frac{1}{2{,}4146022-2} = 2{,}4119505$
(9) $a_6 = 2$
(3) $i = 6$
(4) $g_6 = 198 + 41 = 239$
(5) $h_6 = 140 + 29 = 169$
(6) $h_6 > 100$, **Stop** $\rightarrow$ $p = 99$, $q = 70$

Ein Test zeigt:

$$\left|\sqrt{2} - \frac{99}{70}\right| = 0{,}000072 < \frac{\epsilon}{q}.$$

Der nächste Satz befasst sich mit der Korrektheit des Algorithmus **??**.

Theorem 6.34. *Der Algorithmus* **??** *löst das Problem* **??**. *Die Laufzeit beträgt* $O(\log \frac{1}{\epsilon})$ *und ist daher polynomial in der Input-Länge.*

Zum Beweis dieses Satzes benötigen wir die folgenden zwei Lemmata.

Lemma 6.35.

(a) $\alpha = \dfrac{\alpha_i g_{i-1} + g_{i-2}}{\alpha_i h_{i-1} + h_{i-2}}$ *für alle* $i \geq 0$.
(b) $g_i h_{i-1} - g_{i-1} h_i = (-1)^{i-1}$ *für alle* $i \geq -1$.

Beweis. Wir zeigen beide Teile durch vollständige Induktion über i.

zu (a): Der Fall $i = 0$ folgt direkt aus der Initialisierungphase des Algorithmus **??**. Angenommen, die Aussage ist richtig für alle $j < i$ und sei $i \geq 1$. Wir haben dann

$$\frac{\alpha_{i+1}g_i + g_{i-1}}{\alpha_{i+1}h_i + h_{i-1}} = \frac{\frac{1}{\alpha_i - a_i}(a_i g_{i-1} + g_{i-2}) + g_{i-1}}{\frac{1}{\alpha_i - a_i}(a_i h_{i-1} + h_{i-2}) + h_{i-1}} = \frac{\alpha_i g_{i-1} + g_{i-2}}{\alpha_i h_{i-1} + h_{i-2}} = \alpha.$$

Dabei stammt die erste Gleichheit aus der Definition von α_{i+1}, g_i und h_i und die letzte Gleichheit aus der Induktionsannahme.

zu (b): Der Fall $i = -1$ folgt wieder direkt aus der Initialisierung. Für $i \geq 1$ bekommen wir

$$\begin{aligned} g_{i+1}h_i - g_i h_{i+1} &= (a_{i+1}g_i + g_{i-1})h_i - g_i(a_{i+1}h_i + h_{i-1}) \\ &= h_i g_{i-1} - g_i h_{i-1} = (-1)(g_i h_{i-1} - h_i g_{i-1}) = (-1)^i. \end{aligned}$$

Wiederum stammt die erste Gleichheit aus der Definition von g_{i+1} und h_{i+1} und die letzte Gleichheit aus der Induktionsannahme.

Lemma 6.36. *Es gilt:*

$$\frac{g_i}{h_i} > \alpha, \textit{ falls } i \textit{ ungerade}$$

$$\frac{g_i}{h_i} < \alpha, \textit{ falls } i \textit{ gerade.}$$

Beweis. Betrachte die Funktion

$$f(x) = \frac{xg_{i-1} + g_{i-2}}{xh_{i-1} + h_{i-2}}, \quad \text{für} \quad xh_{i-1} + h_{i-2} \neq 0.$$

Solange der Algorithmus **??** nicht im siebten Schritt abbricht gilt $a_i < \alpha_i$. Aus den Schritten (4) und (5) bekommen wir die Aussage $f(a_i) = \frac{g_i}{h_i}$ und nach Lemma **??** gilt $f(\alpha_i) = \alpha$ (beachte, dass sowohl a_i als auch α_i ungleich dem Ausdruck $-\frac{h_{i-2}}{h_{i-1}} < 0$ sind). Betrachten wir nun die Ableitung von f:

$$\begin{aligned} f'(x) &= \frac{g_{i-1}(xh_{i-1} + h_{i-2}) - h_{i-1}(xg_{i-1} + g_{i-2})}{(xh_{i-1} + h_{i-2})^2} \\ &= \frac{g_{i-1}h_{i-2} - h_{i-1}g_{i-2}}{(xh_{i-1} + h_{i-2})^2} = \frac{(-1)^{i-2}}{(xh_{i-1} + h_{i-2})^2}. \end{aligned} \tag{6.70}$$

Dabei wurde am Schluss Lemma **??** (b) benutzt. Daher gilt nun:

$$\begin{aligned} i \text{ ist gerade} \quad &\Rightarrow \quad f'(x) > 0 \quad \Rightarrow \quad \frac{g_i}{h_i} = f(a_i) < f(\alpha_i) = \alpha, \\ i \text{ ist ungerade} \quad &\Rightarrow \quad f'(x) < 0 \quad \Rightarrow \quad \frac{g_i}{h_i} = f(a_i) > f(\alpha_i) = \alpha. \end{aligned}$$

Jetzt haben wir alles zusammen für den

Beweis von Satz ??.

Es gilt $a_i \leq \alpha_i < a_i + 1$. Demnach gilt $\alpha_i > 1$ und $a_i \geq 1$ für alle $i \geq 1$. Daraus folgt dann $h_i < h_{i+1}$ für alle $i \geq 1$. Da zusätzlich $1 = h_0 < h_1$ gilt, stoppt der Algorithmus immer, sagen wir nach m Schritten. Weiterhin haben wir $h_i \geq h_{i-1} + h_{i-2} > 2h_{i-2}$. Dies impliziert ein exponentielles Wachsen der h_i, also gilt $m = O(\log \frac{1}{\epsilon})$.

Bleibt noch zu zeigen, dass der Algorithmus **??** ein korrektes Ergebnis liefert. Angenommen, der Algorithmus stoppt im sechsten Schritt. Zusammen mit Lemma **??** (b) und Lemma **??** haben wir dann

$$\begin{aligned}
\left|\alpha - \frac{g_{m-1}}{h_{m-1}}\right| &\leq \left|\alpha - \frac{g_m}{h_m}\right| + \left|\frac{g_m}{h_m} - \frac{g_{m-1}}{h_{m-1}}\right| \\
&< \left|\frac{g_m}{h_m} - \frac{g_{m-1}}{h_{m-1}}\right| \\
&= \frac{|g_m h_{m-1} - g_{m-1} h_m|}{h_m\, h_{m-1}} \\
&= \frac{1}{h_m\, h_{m-1}} \\
&< \frac{\epsilon}{h_{m-1}}.
\end{aligned}$$

Stoppt der Algorithmus dagegen im siebten Schritt, so liefert uns Lemma **??**(a) die Aussage

$$\begin{aligned}
\alpha &= \frac{\alpha_m g_{m-1} + g_{m-2}}{\alpha_m h_{m-1} + h_{m-2}} \\
&= \frac{a_m g_{m-1} + g_{m-2}}{a_m h_{m-1} + h_{m-2}} \\
&= \frac{g_m}{h_m}.
\end{aligned}$$

Da der Algorithmus in diesem Fall nicht im sechsten Schritt gestoppt hat, gilt $h_m \leq \frac{1}{\epsilon}$ und daraus folgt die Behauptung.

Am Ende dieses Abschnitts wollen wir noch darauf hinweisen, dass sich der Quotient $\frac{g_i}{h_i}$ auch in folgender Form schreiben läßt:

$$\frac{g_i}{h_i} = a_0 + \cfrac{1}{a_1 + \cfrac{1}{a_2 + \cfrac{1}{\cdots a_{i-1} + \cfrac{1}{a_i}}}}.$$

Der Name des Algorithmus **??** „Kettenbruch-Algorithmus“ stammt von dieser Darstellung.

6.4.2 Die Äquivalenz von SEP, OPT und VIOL

Jetzt sind wir in der Lage, die zeitpolynomiale Äquivalenz von SEP, OPT und VIOL zu beweisen.

Theorem 6.37. *Gegeben sei ein wohlbeschriebenes, beschränktes und volldimensionales Polyeder (P, n, φ). Angenommen wir hätten ein Orakel, dass eines der drei Probleme SEP, OPT oder VIOL für (P, n, φ) löst. Dann können die zwei verbliebenen Probleme in orakel-polynomialer Zeit gelöst werden.*

Beweis. Wir zeigen den Satz **??** in drei Schritten.

- (SEP) → (OPT)
- (OPT) → (VIOL)
- (VIOL) → (SEP)

Zu zeigen ist, dass die Transformationen jeweils in orakelpolynomialer Zeit erfolgen können.

(SEP) → (OPT).

Angenommen, wir haben ein Separationsorakel (SEP), das SEP für (P, n, φ) löst, wobei P volldimensional und beschränkt ist. Weiterhin haben wir einen Zielfunktionsvektor $c \in \mathbb{Q}^n$ und damit möchten wir das Problem $\max\{c^T x \,|\, x \in P\}$ lösen. O.B.d.A können wir annehmen, dass $c \in \mathbb{Z}^n$ gilt, andernfalls können wir dies durch Skalierung der Zielfunktion immer erreichen. Die skalierte Zielfunktion hat dann höchstens die Kodierungslänge $n\langle c\rangle$.

Wir werden das Problem $\max\{c^T x \,|\, x \in P\}$ nicht direkt lösen, sondern verwenden eine leicht abgeänderte Zielfunktion. Für diese gestörte Zielfunktion existiert eine eindeutige Lösung. Diese Optimallösung ist gleichzeitig optimal für das Ausgangsproblem. Nach Lemma **??** wissen wir, dass jede Ecke von P eine Komplexität von höchstens $\nu = 4n^2\varphi$ besitzt. Wir definieren $r = 2^{2\nu}$ und

$$d = r^n c + (1, r, \ldots, r^{n-1})^T.$$

Betrachten wir nun das Optimierungsproblem $\max\{d^T x \,|\, x \in P\}$. Als erstes wollen wir folgende Eigenschaften nachweisen:

(i) $\max\{d^T x \,|\, x \in P\}$ hat eine eindeutige Lösung x^*.
(ii) Für jede Ecke $u \neq x^*$ gilt $d^T(x^* - u) \geq 2^{-2\nu}$.
(iii) x^* löst $\max\{c^T x \,|\, x \in P\}$.

Beweis. Sei x^* ein optimale Ecke von $\max\{d^T x \,|\, x \in P\}$ und sei $u \in P$ ein von x^* verschiedene Ecke von P. Dann gilt $x^* - u = \frac{1}{\alpha} z$ mit $\alpha \in \mathbb{Z}$, $0 < \alpha < r$, $z \in \mathbb{Z}^n \setminus \{0\}$ und $0 \leq |z_i| < r$, $i \in \{1, 2, \ldots, n\}$. Beachte dabei, dass x^* und u rationale Vektoren sind, deren Kodierungslänge höchstens ν ist. Wir erhalten

$$0 \leq d^T(x^* - u) = \frac{1}{\alpha}\Big(r^n c^T z + (1, r, \ldots, r^{n-1})^T z\Big). \tag{6.71}$$

Dies impliziert $c^T z \geq 0$, da $c^T z \in \mathbb{Z}$ und der Term r^n dominiert den Term $\sum_{i=1}^{n} r^{i-1} z_i$. Daraus folgt nun $c^T(x^* - u) = \frac{1}{\alpha} c^T z \geq 0$ und damit Punkt (iii).
Weiterhin gilt $\sum_{i=1}^{n} r^{i-1} z_i \neq 0$, da $z \neq 0$ ist. Aus Gleichung (**??**) folgern wir nun

$$d^T(x^* - u) \geq \frac{1}{\alpha} > \frac{1}{r}. \tag{6.72}$$

Daraus folgen jetzt die Punkte (i) und (ii).

Unter Berücksichtigung der Aussagen (i) und (iii) reicht es aus, das Problem $\max\{d^T x \,|\, x \in P\}$ zu lösen, um die Ecke x^* zu finden. Setzen wir nun $N = 2^{\langle d \rangle + 4n^2 \varphi}$, so wissen wir mit Lemma **??**, dass der Optimalwert des Problems $\max\{d^T x \,|\, x \in P\}$ im Intervall $[-N, N]$ liegt. Mit Hilfe des Parameters

$$\epsilon = \frac{1}{8r^2\, 2^{\nu}}$$

definieren wir eine Familie von Polytopen durch

$$P_k = \{x \in P \,|\, -N + k\epsilon \leq d^T x \leq N\}, \quad k \in \left\{0, 1, \ldots, \frac{2N}{\epsilon}\right\}.$$

Das Orakel (SEP) kann auf einfache Weise auf ein Seperationsorakel für P_k ausgedehnt werden. P_k ist ein wohlbeschriebenes Polyeder, dessen Facettenkomplexität höchstens $\hat{\varphi} = \max\{\varphi, \langle d\rangle + \langle N\rangle + \langle \epsilon k\rangle\}$ ist. Verwenden wir nun die Ellipsoid-Methode für ein k und $\hat{\epsilon} = 2^{-8n^4\hat{\varphi}}$, so finden wir in polynomialer Zeit einen Vektor $y^k \in P_k$ oder wir wissen, dass $\text{vol}(P_k) < \hat{\epsilon}$ gilt. Mit einer binären Suche finden wir daher ein $k \in \{0, 1, \ldots, \frac{2N}{\epsilon}\}$ und einen Vektor $y^k \in P_k$ mit $\text{vol}(P_{k+1}) < \hat{\epsilon}$. Wir werden zeigen, dass dieses y^k so nahe am Optimum liegt, dass eine Rundung mit dem Kettenbruch-Algorithmus **??** die Optimallösung liefert.

Da $\text{vol}(P_{k+1}) < \hat{\epsilon}$ gilt, wissen wir mit Lemma **??**, dass P_{k+1} nicht volldimensional sein kann. Daraus folgt

$$d^T x^* \;\leq\; d^T y^k \;+\; \epsilon. \tag{6.73}$$

Aus $y^k \in P$ folgt die Existenz von $n+1$ Ecken von P, so dass $y^k = \sum_{i=0}^n \lambda_i x^i$ gilt mit $\sum_{i=0}^n \lambda_1 = 1$, $\lambda_i \geq 0 \;\forall\, i \in \{0, 1, \ldots, n\}$. Einer dieser $n+1$ Vektoren muss überdies hinaus x^* sein, denn ansonsten gilt nach (**??**)

$$d^T x^* - d^T y^k \;=\; \sum_{i=0}^{n} \lambda_i d^T (x^* - x^i) \;>\; \frac{1}{r} \;>\; \epsilon.$$

O.B.d.A gelte $x^0 = x^*$. Dann bekommen wir

$$\begin{aligned} d^T y^k \;&=\; d^T x^* \;+\; \sum_{i=1}^{n} \lambda_i (d^T x^i - d^T x^*) \\ &\leq\; d^T x^* \;-\; \frac{1-\lambda_0}{r} \end{aligned}$$

und zusammen mit (**??**)

$$\frac{1-\lambda_0}{r} \;\leq\; d^T x^* - d^T y^k \;\leq\; \epsilon \qquad \Longrightarrow \qquad 1 - \lambda_0 \;\leq\; r\epsilon.$$

Dies führt nun auf die Abschätzung

$$\begin{aligned}
||y^k - x^*|| &= (1-\lambda_0) \left|\left| \sum_{i=1}^{n} \frac{\lambda_i}{1-\lambda_0} x^i - x^* \right|\right| \\
&\leq (1-\lambda_0) \left(\sum_{i=1}^{n} \left(\frac{\lambda_i}{1-\lambda_0} ||x^i|| \right) + ||x^*|| \right) \\
&\leq (1-\lambda_0) \left(\sum_{i=1}^{n} \left(\frac{\lambda_i}{1-\lambda_0} 2^\nu \right) + 2^\nu \right) \\
&= (1-\lambda_0)\, 2^{\nu+1} \\
&\leq \epsilon r 2^{\nu+1} \quad = \quad \frac{1}{4r}.
\end{aligned}$$

Für jede Komponente von y^k wird nun der Kettenbruch-Algorithmus **??** mit $\epsilon = \frac{1}{2^{\nu+1}}$ aufgerufen. Nach Satz **??** erhalten wir damit in polynomialer Zeit ganze Zahlen p_i, q_i mit $1 \leq q_i \leq 2^{\nu+1}$ und

$$\left| y_i^k - \frac{p_i}{q_i} \right| < \frac{1}{2^{\nu+1}\, q_i}.$$

Da x^* eine Ecke von P ist, hat jede Komponente $\frac{s_i}{t_i}$ einen Nenner, der kleiner oder höchstens gleich 2^ν ist, d.h. $1 \leq t_i \leq 2^\nu$. Damit folgt nun

$$\begin{aligned}
\left| \frac{p_i}{q_i} - \frac{s_i}{t_i} \right| &\leq \left| \frac{p_i}{q_i} - y_i^k \right| + \left| y_i^k - \frac{s_i}{t_i} \right| \\
&< \frac{1}{2^{\nu+1}\, q_i} + \frac{1}{4r} \\
&\leq \frac{1}{2\, t_i\, q_i} + \frac{1}{2\, t_i\, q_i} \quad = \quad \frac{1}{t_i\, q_i}.
\end{aligned}$$

Aus dieser Ungleichung folgern wir

$$\left| \frac{p_i}{q_i} - \frac{s_i}{t_i} \right| = \frac{|p_i t_i - s_i q_i|}{q_i\, t_t} < \frac{1}{t_i\, q_i} \qquad \Longrightarrow \qquad |p_i t_i - s_i q_i| < 1.$$

Da alle vorkommenden Zahlen ganze Zahlen sind, folgt sofort

$$p_i t_i = s_i q_i \qquad \Longrightarrow \qquad \frac{s_i}{t_i} = \frac{p_i}{q_i} \qquad \Longrightarrow \qquad x^* = \left(\frac{p_1}{q_1}, \cdots, \frac{p_n}{q_n} \right)^T.$$

Dies ist die Optimallösung für das Problem $\max\{d^T x \,|\, x \in P\}$. Nach Punkt (iii) ist dies gleichzeitig die Optimallösung für $\max\{c^T x \,|\, x \in P\}$.

(OPT) → (VIOL).

Gegeben sei ein wohlbeschriebenes Polyeder (P, n, φ), wobei P beschränkt und volldimensional sei. Weiterhin seien ein Vektor $c \in \mathbb{Q}^n$ und eine Zahl $\gamma \in \mathbb{Q}$ gegeben. Wir haben nun zu entscheiden, ob die Ungleichung $c^T x \leq \gamma$ für $x \in P$ erfüllt ist. Ist dies nicht der Fall, so brauchen wir einen Vektor $y \in P$ mit $c^T y > \gamma$. Als erstes fragen wir das Orakel (OPT) nach einer Lösung für das Problem $\max\{c^T x \,|\, x \in P\}$. Das Orakel wird uns eine Optimallösung y zurückgeben, da P beschränkt und volldimensional ist. Gilt nun $c^T y \leq \gamma$, so haben wir einen Beweis dafür, dass die Ungleichung $c^T x \leq \gamma$ für alle $x \in P$ gültig ist. Andernfalls haben wir einen Vektor y gefunden mit $c^T y > \gamma$.

(VIOL) → (SEP).

Angenommen, wir haben ein Verletztheitsorakel (VIOL), dass das Problem VIOL für ein wohlbeschriebenes, beschränktes und volldimensionales Polyeder (P, n, φ) löst. Betrachte die γ-Polare von P

$$Q \;=\; \left\{ (a^T, \alpha)^T \;\middle|\; a^T x \;\leq\; \alpha \;\forall\, x \in P \right\},$$

d.h. Q beschreibt die Menge aller gültigen Ungleichungen für P. Q ist ein polyedrischer Kegel. Beachte, dass jede gültige Ungleichung von Q die Form $(x^T, -1)y \leq 0$, $x \in P$ hat, und dass die Facetten von Q in einer bijektiven Beziehung zu den Ecken von P stehen. Umgekehrt sind die Extremalstrahlen von Q die Facetten von P. Daher ist $(Q, n+1, 4n^2\varphi + 3)$ ein wohlbeschriebenes Polyeder. Q ist auch volldimensional, da P beschränkt ist (Beachte, dass $((e^i)^T, -2^{4n^2\varphi})^T \in Q \;\forall\, i \in \{1, 2, \ldots, n\}$ und $-e^{n+1} \in Q$ gilt). Betrachte nun die folgende beschränkte Version von Q:

$$\bar{Q} \;=\; \left\{ y \in Q \;\middle|\; -2^{4n^2\varphi} \;\leq\; y_i \;\leq\; 2^{4n^2\varphi} \right\}.$$

$(\bar{Q}, n+1, 4n^2\varphi + n + 3)$ ist ein wohlbeschriebenes, beschränktes und volldimensionales Polyeder. Das Verletztheitsorakel (VIOL) für P ist offensichtlich ein Separationsalgorithmus für Q. Dies kann leicht auf ein Seperationsorakel ($\overline{\text{SEP}}$) für $\bar{Q}$ erweitert werden. Aus den ersten beiden Teilen dieses Beweises folgt daraus die Existenz eines orakelpolynomialen Verletztheitsorakels ($\overline{\text{VIOL}}$) für

$\bar{Q}$. Für ein gegebenes $c \in \mathbb{Q}^{n+1}$ und $\gamma \in \mathbb{Q}$ existiert nun genau dann ein Vektor $\binom{a}{\alpha} \in Q$ mit $c^T\binom{a}{\alpha} > \gamma$, wenn so ein Vektor auch für $\bar{Q}$ existiert, da P eine Facettenkomplexität von höchstens φ besitzt. Daher ist $(\overline{\text{VIOL}})$ ein Verletztheitsorakel, das auch das VIOL für Q löst.

Schließlich kann $(\overline{\text{VIOL}})$ auch benutzt werden, um das Problem SEP für (P, n, φ) zu lösen. Dies geschieht folgendermaßen:

Gegeben sei ein Vektor y, bei dem wir entscheiden müssen, ob $y \in P$ liegt. Ist dies nicht der Fall, so müssen wir ein $a \in \mathbb{Q}^n$ und ein $\alpha \in \mathbb{Q}$ finden mit $a^T x \leq \alpha \ \forall\, x \in P$ und $a^T y > \alpha$. Anders ausgedrückt brauchen wir ein $\binom{a}{\alpha} \in Q$ mit $a^T y > \alpha$. Dazu fragen wir unser Orakel $(\overline{\text{VIOL}})$ für das Polyeder Q, den Vektor $c = \binom{y}{-1}$ und $\gamma = 0$. Meldet $(\overline{\text{VIOL}})$ nun, dass $(y, -1)\binom{a}{\alpha} \leq \gamma = 0$ für alle $\binom{a}{\alpha} \in Q$, d.h. $a^T y \leq \alpha$ für alle gültigen Ungleichungen $a^T x \leq \alpha$, so gilt $y \in P$. Andernfalls gibt $(\overline{\text{VIOL}})$ einen Vektor $\binom{a}{\alpha} \in Q$ zurück mit $(y, -1)\binom{a}{\alpha} > \gamma = 0$. Dann ist aber $a^T x \leq \alpha$ eine gültige Ungleichung für P, die von y verletzt wird.

Damit ist der Beweis beendet.

Bevor wir dieses Kapitel beenden, wollen wir uns noch einmal dem allgemeinen Fall zuwenden, denn Polyeder sind nicht notwendigerweise beschränkt und volldimensional.

Ein Weg, diese Polyeder zu behandeln ist es, durch geeignete Störungen des Polyeders ein volldimensionales Polyeder zu bekommen. Dieser Vorschlag kam von Khachiyan [?]. Die Störung hängt von der Kodierungslänge des Inputs ab und muss mit größter Sorgfalt erfolgen, damit der Rundungsschritt am Ende des Algorithmus auch wirklich auf die Optimallösung des Ausgangsproblems führt.

Eine andere Möglichkeit stammt aus [?]. Dort konzentriert man sich auf die Ausgabe des Ellipsoid-Algorithmus. Angenommen, das Polyeder wäre $(n-1)$-dimensional, d.h. es liegt in einer Hyperebene $H \subset \mathbb{R}^n$. Da die Volumina der berechneten Ellipsoiden gegen Null gehen, werden sie immer flacher um H herum. Der Eigenvektor der zum kleinsten Eigenwert gehört muss nun ziemlich nahe am Normalenvektor von H liegen. Um diesen Vektor zu finden konstruiert man einen approximativen Vektor u zum Eigenvektor des kleinsten Eigenwertes der Matrix, die das

Ellipsoid definiert. Der Vektor u hat die zusätzliche Eigenschaft, dass alle seine Komponenten $u_1, \ldots, u_n$ einen gemeinsamen Nenner q haben, dessen Kodierungslänge polynomial beschränkt ist. Die Methode, mit der gleichzeitig n Zahlen durch rationale Zahlen angenähert werden, die alle den gleichen, möglichst kleinen Nenner haben, heißt Simultane Diophantische Approximation und ist eine Verallgemeinerung des Kettenbruch-Algorithmus für höhere Dimensionen. Da der H beschreibende Vektor eine Kodierungslänge von höchstens φ hat, erzeugt die richtige Wahl des Nenners q in der Tat einen Vektor u, der H beschreibt. Durch Iteration kann die affine Hülle von P bestimmt werden und von dort aus auf den volldimensionalen Fall angewendet werden.

Altes Kommando `df` hier verwendet

Weiterhin kann man diese Methode benutzen, um in polynomialer Zeit zu klären, ob ein Polyeder leer ist oder nicht. Falls nicht, soll der Algorithmus einen Vektor $y \in P$ liefern (Bemerkung: dies ist ein spezieller Fall von VIOL mit $c = 0$ und $\gamma = -1$). Weiterhin kann ein Separierungsorakel für P in ein Separierungsorakel für $\text{rec}(P)$ umgewandelt werden.

Mit diesen Werkzeugen ist es nun möglich, auch unbeschränkte Polyeder zu behandeln. Wie oben schon gesagt kann ein Separierungsorakel für P ein Separierungsorakel $(\text{SEP})_r$ für $\text{rec}(P)$ erzeugen und damit auch das Problem lösen, ob $\text{rec}(P)$ leer ist. Ist nun ein beliebiges Optimierungsproblem $\max\{c^T x \,|\, x \in P\}$ für ein wohlbeschriebenes Polyeder (P, n, φ) und ein Separierungsorakel für (P, n, φ) gegeben, so können wir sofort prüfen, ob die Menge $\text{rec}(P) \cap \{x \in \mathbb{R}^n \,|\, c^T x = 1\}$ leer ist oder nicht. Wir können weiterhin sofort überprüfen, ob $\max\{c^T x \,|\, x \in P\}$ unbeschränkt ist.

Korollar 6.38. *Lineare Programme können in polynomialer Zeit gelöst werden.*

Die Möglichkeiten sind noch lange nicht ausgeschöpft. Mit den hier angeführten Methoden ist es unter anderem möglich, die Ecken, duale Lösungen, verletzte Facetten und vieles andere zu bestimmen. Vieles davon findet sich in [?].

Teil II

Ganzzahlige Optimierung

7

Ein erste Blick auf ganzzahlige Optimierung

7.1 Der Weg nach Hause (oder doch nicht?)

Für unseren Austauschstudenten *Simple Ex* ist das Ende seines geplanten Aufenthalts gekommen. Er will nach Hause fliegen und hat dafür bereits seine seine sechs Koffer gepackt. Alle sind randvoll gepackt, ihre Gewichte addieren sich genau auf die 20 kg, die er bei der Fluggesellschaft *Knapsack Airlines* aufgeben darf. Handgepäck ist aufgrund von Sicherheitsbestimmungen mal wieder verboten.

Kurz vor dem Abflug klingelt die beste Freundin *Cutty Plain* seiner Freundin und bringt noch einen weiteren Koffer vorbei, den er ihr mitbringen soll. Dieser Koffer wiegt erschreckende 8 kg, was *Simple Ex* sofort über das erlaubte Gewichtslimit bringt. Eine kurze Diskussion mit *Cutty Plain* macht ihm klar, dass er ohne diesen Koffer gar nicht zu Hause antreten darf. Also bleiben ihm noch 12 kg für sein restliches Gepäck, von dem er damit einiges zurücklassen werden muss. Aber was?

Die Zeit drängt und Aus- oder Umpacken ist hoffnungslos. *Simple Ex* erinnert sich an seine erfolgreich bestandene Klausur in Lineare Optimierung und macht sich daran, sein Kofferproblem mit Hilfe seines neu erworbenen Wissens zu lösen. Ist das Ganze nicht einfach ein LP? Eine Nebenbedingung ist wohl, dass maximal noch 12 kg Gewicht eingepackt werden kann, aber wie sieht die Zielfunktion aus?

Simple Ex macht sich klar, dass unterschiedliche Koffer für ihn unterschiedlichen Wert besitzen. Er bewertet sie

daher mit Punkten, je wichtiger der Koffer, um so mehr Punkte erhält dieser. Damit steht er jetzt vor den Daten aus Tabelle **??** und sein Wunsch ist es, so auszuwählen, dass die Summe der Punkte von mitgenommenen Koffern möglichst groß ist.

	Koffer 1	Koffer 2	Koffer 3	Koffer 4	Koffer 5	Koffer 6
Gewicht	4 kg	3 kg	6 kg	3 kg	3 kg	2 kg
Punkte	7	5	9	4	2	1

Tabelle 7.1. Die Koffer von *Simple Ex* mit Gewichten und Bewertungen in Punkten

Schnell formuliert *Simple Ex* sein Lineares Programm

$$\begin{aligned} \max \quad & 7x_1 + 5x_2 + 9x_3 + 4x_4 + 2x_5 + 1x_6 && \text{(7.1a)} \\ & 4x_1 + 3x_2 + 6x_3 + 3x_4 + 3x_5 + 2x_6 \leq 12 && \text{(7.1b)} \\ & 0 \leq x_i \leq 1, i = 1, \ldots, 6. && \text{(7.1c)} \end{aligned}$$

und erhält als Optimallösung

$$x_1 = 1, x_2 = 1, x_3 = 5/6, x_4 = x_5 = x_6 = 0$$

mit Gesamtbewertung 19.5. Es zeigt sich dabei aber eine Schwierigkeit: *Simple Ex* kann schlecht den dritten Koffer zersägen, er muss entweder ganz oder gar nicht mitgenommen werden (genauso wie jeder andere Koffer auch). Wie *Cutty Plain* richtig betont, ist eigentlich nicht (**??**) zu lösen, sondern ein Problem, in dem die Variablen nur die Werte 0 oder eins annehmen:

$$\begin{aligned} \max \quad & 7x_1 + 5x_2 + 9x_3 + 4x_4 + 2x_5 + 1x_6 && \text{(7.2a)} \\ & 4x_1 + 3x_2 + 6x_3 + 3x_4 + 3x_5 + 2x_6 \leq 12 && \text{(7.2b)} \\ & x_i \in \{0, 1\}, i = 1, \ldots, 6. && \text{(7.2c)} \end{aligned}$$

Simple Ex ist nicht davon überzeugt, dass hier außer dem Koffer von *Cutty* eine weitere Schwierigkeit vorliegt. Wenn man die ersten beiden Koffer einpackt, wird der dritte wohl dabeibleiben müssen (seine Variable wird einfach auf 0 gerundet) so dass zuänchst 12 Punkte auf dem Konto stehen. Nach Mitnehmen der ersten beiden Koffer sind noch 5 kg frei, die man am besten mit den Koffern 4 und 6 auffüllen kann. Das ergibt insgesamt genau 12 kg

und 17 Punkte. *Simple Ex* ist glücklich und zufrieden mit seiner Lösung. Da das Gewichtslimit genau ausgeschöpft ist, muss das doch die beste Lösung sein.

Cutty Plain schaut noch nicht ganz überzeugt. Sie hat nämlich bereits eine Vorlesung über ganzzahlige Optimierung gehört. Klar, besser als 19.5 Punkte kann man sicher nicht erreichen und, da das Problem (**??**) höchstens einen besseren Optimalwert hat als „die ganzzahlige Variante“ (**??**). Da man Koffer immer ganz oder gar nicht mitnehmen muss, sind sogar 19 Punkte eine obere Schranke für das Optimum. *Cutty Plain* weist *Simple Ex* darauf hin, dass er sein Lineares Programm (**??**) noch verbessern kann, indem er seine Beobachtung von vorhin, dass die ersten drei Koffer nicht alle drei mitgenommen werden können, als Nebenbedingung hinzufügt.

$$x_1 + x_2 + x_3 \leq 2. \tag{7.3}$$

Simple Ex reoptimiert das Lineare Programm (**??**) mit der zusätzlichen Bedingung (**??**) und erhält als neue Lösung

$$x_1 = 1, x_2 = 0, x_3 = 1, x_4 = 2/3, x_5 = x_6 = 0$$

mit Zielfunktionswert $56/3 \approx 18.666$. Damit ist jetzt klar, dass 19 als optimaler Wert ausscheidet. „Cool“, sagt *Simple Ex*: „Das ist ja einfach!“. Er sieht sofort, dass man auch nicht gleichzeitig den ersten, dritten und vierten Koffer mitnehmen kann, da diese zusammen 13 kg wiegen. Auch dies lässt sich wieder als Nebenbedingung formulieren. Damit erhält *Simple Ex* jetzt das folgende Lineare Programm:

$$\begin{aligned}
\max \quad & 7x_1 + 5x_2 + 9x_3 + 4x_4 + 2x_5 + 1x_6 && \text{(7.4a)}\\
& 4x_1 + 3x_2 + 6x_3 + 3x_4 + 3x_5 + 2x_6 \leq 12 && \text{(7.4b)}\\
& x_1 + x_2 + x_3 \leq 2 && \text{(7.4c)}\\
& x_1 + x_3 + x_4 \leq 2 && \text{(7.4d)}\\
& 0 \leq x_i \leq 1, i = 1, \ldots, 6. && \text{(7.4e)}
\end{aligned}$$

Ein erneuter Optimierungslauf liefert jetzt die Lösung

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

mit Zielfunktionswert 18. „Das ist ja besser als ich vorhin“, denkt *Simple Ex*: „Und jetzt weiß ich außerdem, dass man nicht mehr als 18 Punkte erzielen kann. Geht das immer so?“.

Cutty Plain antwortet: „Nicht immer, aber immer öfter. Und wann und wie lernst Du in der ganzzahligen Optimierung“. *Simple Ex* beschliesst, dass ihn das ganz sehr interessiert und seine Freundin wohl noch ein weiteres Semester auf ihn warten muss.

7.2 Ganzzahlige Lineare Programme

Definition 7.1 (Gemischt-Ganzzahliges Lineares Programm, Mixed Integer Programm (MIP)). *Sei* $c \in \mathbb{R}^n$, $A \in \mathbb{R}^{m\times n}$, $b \in \mathbb{R}^m$, $p \in \{0, 1, \ldots, n\}$. *Dann heißt*

$$\begin{aligned} \max\, & c^\top x \\ & Ax \leq b \\ & x \in \mathbb{Z}^p \times \mathbb{R}^{n-p} \end{aligned}$$

gemischt-ganzzahliges lineares Programm (Optimierungsproblem).

Die folgenden Spezialfälle haben eigene Bezeichnungen:

- $p = 0$: Lineares Programm (vgl. DO I),
- $p = n$: (rein) ganzzahliges (lineares) Programm,
- $p = n$ und $x \in \{0, 1\}^n$: binäres (lineares) Programm oder 0/1 Programm.

Referenz auf nicht vorhandenen Satz

In engem Zusammenhang zu MIPs stehen lineare kombinatorische Optimierungsprobleme, die uns auch in dieser Vorlesung interessieren werden (vgl. Def. 1.8, DO I).

Definition 7.2 ((Lineares) Kombinatorisches Optimierungsproblem). *Gegeben sei eine endliche Grundmenge E, eine Teilmenge $\mathcal{I}$ der Potenzmenge 2^E von E (die Elemente in $\mathcal{I}$ heißen* zulässige Mengen *oder* Lösungen*) und eine Funktion $c\colon E \to \mathbb{R}$. Für jede Menge $F \subseteq E$ sei $c(F) := \sum_{e\in F} c_e$. Das Problem*

$$\max_{I\in\mathcal{I}} c(I) \quad \textit{bzw.} \quad \min_{I\in\mathcal{I}} c(I)$$

heißt (Lineares) kombinatorisches Optimierungsproblem.

7.3 Komplexität ganzzahliger Optimierung

P und NP etc.

7.4 Fast alle IPs sind doch LPs

Die zulässigen Mengen von (gemischt-) ganzzahligen Programmen sind spezielle Teilmengen von Polyedern.

Intuitiv könnte man meinen, dass durch Bildung der konvexen Hülle aller zulässigen Punkte wieder ein Polyeder entsteht. Dem ist aber nicht so.

Beispiel 7.3. *Betrachte*

$$\begin{aligned} z^* = \max -\sqrt{2}x &+ y \\ -\sqrt{2}x &+ y \leq 0 \\ x &\geq 1 \\ y &\geq 0 \\ x, y &\text{ ganzzahlig} \end{aligned} \tag{7.5}$$

Dann gilt
(a) (**??**) hat zulässige Lösungen (bezeichne diese mit S).
(b) (**??**) ist nach oben beschränkt.
(c) (**??**) hat keine Optimallösung (sonst wäre $\sqrt{2}$ rational).
Wäre nun $\text{conv}(S)$ ein Polyeder, so wäre

$$z^* = \max\left\{\sqrt{2}x + y \mid x \in S\right\} = \max\left\{\sqrt{2}x + y \mid x \in \text{conv}\,(S)\right\}$$

und besäße eine optimale Ecklösung, Widerspruch zu (3).

Der Grund für obige Tatsache liegt daran, dass wir irrationale Daten erlauben.

Altes Kommando `df` hier verwendet

Definition 7.4. Ein Polyeder $P \subseteq \mathbb{K}^n$ heißt rational, falls es eine Matrix $A \in \mathbb{Q}^{m\times n}$ und einen Vektor $b \in \mathbb{Q}^m$ gibt mit $P = P(A, b)$.

Darüber hinaus bezeichnen wir mit $P_{I,p} := \text{conv}\,\{x \in P \mid x \in \mathbb{Z}^p \times \mathbb{R}^{n-p}\}$ für $p \in \{0, 1, \ldots, n\}$, wobei wir auch P_I als Abkürzung für $P_{I,n}$ schreiben.

Zunächst zwei einfache Beobachtungen

Beobachtung 7.5.

(a) Ist P ein beschränktes Polyeder, so ist P_I ein Polyeder.
(b) Ist P ein rationaler polyedrischer Kegel, so gilt $P = P_I$.

Beweis. (a) Da P beschränkt, ist $X = \{x \in \mathbb{Z}^n \mid x \in P\}$ endlich. Nach Folgerung 6.14, DO I, ist $\text{conv}\,(X) = P_I$ ein Polyeder.

(b) Klar.

Bemerkung 7.6.

(a) **??**(a) gilt nicht nur für rationale Polyeder.
(b) Entsprechende Aussagen gelten auch für $P_{I,p}$ für alle $p = 0, 1, \ldots, n$.

Satz 7.7. *Ist P ein rationales Polyeder, dann ist P_I ebenfalls ein (rationales) Polyeder und es gilt* $\operatorname{rec}(P) = \operatorname{rec}(P_I)$.

Beweis. Aus Satz 5.25, DO I, wissen wir, P hat eine Darstellung der Form $P = Q + C$, wobei Q ein Polytop ist und C ein Kegel. Nach Beobachtung **??** (b) gibt es $y_i \in \mathbb{Z}^n, i = 1, \ldots, s$, mit $C = \operatorname{cone}(\{y_i, \ldots, y_s\})$. Wir setzen

$$B := \left\{ \sum_{i=1}^{s} \mu_i y_i \mid 0 \leq \mu_i \leq 1, i = 1, \ldots, s \right\} .$$

Wir behaupten

$$P_I = (Q + B)_I + C .$$

Daraus folgt die Behauptung, denn $(Q + B)$ ist beschränkt, also nach Beobachtung **??** (a) ist $(Q + B)_I$ ein Polyeder und falls $P_I \neq \emptyset$, dann ist C der Rezessionskegel von P_I.

$\subseteq$: Sei $p \in P_I$ ein ganzzahliger Punkt in P.
Dann ex. $q \in Q$ und $c \in C$ mit $p = q + c$. Für c gilt $c = \sum_{i=1}^{s} \mu_i y_i$ mit $\mu_i \geq 0$. Setze $c' = \sum_{i=1}^{s} \lfloor \mu_i \rfloor y_i$ und $b = c - c'$, so ist c' ganzzahlig und $q + b = p - c'$ ebenfalls, da p und c' ganzzahlig.
Außerdem ist $b \in B$ nach Definition und damit $q + b \in (Q + B)_I$. Also ist $p = q + b + c' \in (Q + B)_I + C$.
$\supseteq$: Hier gilt:

$$(Q+B)_I+C \subseteq P_I+C \underbrace{=}_{\text{Beob.?? } (b)} P_I+C_I = (P+C)_I = P_I.$$

Entsprechend zeigt man

Satz 7.8. *Ist P ein rationales Polyeder, so ist $P_{I,p}$ ebenfalls ein rationales Polyeder für alle $p \in \{0, 1, \ldots, n\}$.*

Wir wissen nun also, dass sich jedes MIP auf ein lineares Programm zurückführen lässt, d.h. es ex. $D \in \mathbb{Q}^{m \times n}$, $d \in \mathbb{Q}^m$ mit $P_I = P(D, d)$. Wie sieht D, d aus?

Zunächst eine Beobachtung, die wir jedoch nicht beweisen wollen.

Lemma 7.9. *Sei $P = P(A, b)$ ein rationales Polyeder, so dass die Kodierungslänge jeder Ungleichung höchstens φ ist.*

(a) Dann gibt es ein Ungleichungssystem $Dx \leq d$ mit $P_I = P(D, d)$, so dass die Kodierungslänge jeder Ungleichung höchstens $15n^6\varphi$ ist.
(b) Falls $P_I \neq \emptyset$, so gibt es einen ganzzahligen Punkt, dessen Kodierungslänge höchstens $5n^4\varphi$ ist.

Zum Beweis dieses Lemmas siehe beispielsweise [?], Kapitel 17.

8

Modellierung ganzzahliger Probleme

Beispiel 8.1 (Das Zuordnungsproblem).
In einer Firma gibt es n Mitarbeiter, die n Jobs ausführen können. Jede Person kann genau einen Job durchführen. Die Kosten, die der Firma entstehen, wenn Mitarbeiter i Job j ausführt, betragen c_{ij}. Wie soll die Firma ihre Mitarbeiter kostengünstig einsetzen: Variablen:

$$x_{ij} = \begin{cases} 1 & \text{falls Mitarbeiter } i \text{ Job } j \text{ bearbeitet,} \\ 0 & \text{sonst .} \end{cases}$$

Binäres Programm:

$$\begin{aligned} \min \sum_{i,j} c_{ij} x_{ij} & \\ \sum_{j=1}^{n} x_{ij} = 1 & \qquad \text{für alle } i = 1, \ldots, n \\ \sum_{i=1}^{n} x_{ij} = 1 & \qquad \text{für alle } j = 1, \ldots, n \\ x_{ij} \in \{0,1\} & \qquad \text{für alle } i, j = 1, \ldots, n. \end{aligned}$$

Beispiel 8.2 (Das Rucksack-Problem (knapsack)).
Eine Firma möchte ein Budget von b Geldeinheiten in n Projekte investieren. Jedes dieser Projekte j kostet a_j Geldeinheiten und hat einen erwarteten Gewinn von c_j Geldeinheiten. In welche Projekte soll die Firma investieren? Variablen:

$$x_j = \begin{cases} 1 & \text{falls Projekt } j \text{ gewählt wird,} \\ 0 & \text{sonst.} \end{cases}$$

0/1 Programm:

$$\max \sum_{j=1}^{n} c_j x_j$$
$$\sum_{j=1}^{n} a_j x_j \leq b$$
$$x_j \in \{0,1\} \qquad \text{für } j = 1, \ldots, n\,.$$

Beispiel 8.3 (Set-Packing | Partitioning | Covering).
Gegeben eine endliche Grundmenge $E = \{1, \ldots, m\}$ und eine Liste $F_j \subseteq E, j = 1, \ldots, n$ von Teilmengen von E. Mit jeder Teilmenge F_j sind Kosten / Erträge c_j assoziiert. Das Problem eine bzgl. c minimale bzw. maximale Auswahl von Teilmengen F_j zu finden, so dass jedes Element $e \in E$ in höchstens, genau, mindestens einer Teilmenge vorkommt, heißt *Set-Packing, -partitioning* bzw. *-covering Problem.* Variablen:

$$x_j = \begin{cases} 1 & \text{falls } F_j \text{ gewählt wird,} \\ 0 & \text{sonst.} \end{cases}$$

Definiere zusätzlich eine 0/1 Matrix $A \in \{0,1\}^{m \times n}$, wobei $a_{ij} = 1$ genau dann, wenn Teilmenge F_j Element $i \in E$ enthält.

0/1 Programm:

$$\min c^\top x$$
$$Ax \begin{cases} \leq 1 & \text{für das Set-Packing Problem} \\ = 1 & \text{für das Partitioning Problem} \\ \geq 1 & \text{für das Covering Problem} \end{cases}$$
$$x \in \{0,1\}^n\,.$$

Beispiel 8.4. Optimale Steuerung von Gasnetzwerken

Beispiel 8.5. Frequenzplanung in der Telekommunikation

9

Ganzzahlige Polyeder

In diesem Kapitel werden wir uns mit speziellen Polyedern beschäftigen, die aus ganzzahligen Programmen resultieren. Wir werden insbesondere solche Polyeder untersuchen, deren Ecken alle ganzzahlig sind. Diese können als "Glücksfälle" betrachtet werden, da es in diesen Fällen ausreicht, nur das zugrundeliegende lineare Programm zu lösen.

9.1 Charakterisierungen von Ganzzahligkeit

Definition 9.1 (Integral Polyhedron). *A polyhedron P is called* integral *if every face of P contains an integral point.*

integral

We are now going to give some equivalent definitions of integrality which will turn out to be quite useful later.

Theorem 9.2. *Let $P = P(A, b)$ be a pointed rational polyhedron. Then, the following statements are equivalent:*

(i) P *is an integral polyhedron.*
(ii) The LP $\max\{ c^T x : x \in P \}$ *has an optimal integral solution for all* $c \in \mathbb{R}^n$ *where the value is finite.*
(iii) The LP $\max\{ c^T x : x \in P \}$ *has an optimal integral solution for all* $c \in \mathbb{Z}^n$ *where the value is finite.*
(iv) The value $z^{(P)} = \max\{ c^T x : x \in P \}$ *is integral for all* $c \in \mathbb{Z}^n$ *where the value is finite.*
(v) $P = \text{conv}(P \cap \mathbb{Z}^n)$.

Beweis. We first show the equivalence of statements (i)-(iv):

(i)⇒(ii) The set of optimal solutions of the LP is a face of P. Since every face contains an integral point, there is an integral optimal solution.
(ii)⇒(iii) trivial.
(iii)⇒(iv) trivial.
(iv)⇒(i) Suppose that (i) is false and let x^0 be an extreme point which by assumption is not integral, say component x^0_j is fractional. By Theorem **??** there exists a vector $c \in \mathbb{Z}^n$ such that x^0 is the unique solution of $\max\{ c^T x : x \in P \}$. Since x^0 is the unique solution, we can find a large $\omega \in \mathbb{N}$ such that x^0 is also optimal for the objective vector $\bar{c} := c + \frac{1}{\omega} e_j$, where e_j is the jth unit vector. Clearly, x^0 must then also be optimal for the objective vector $\tilde{c} := \omega\bar{c} = \omega c + e_j$. Now we have

$$\tilde{c}^T x^0 - \omega c^T x^0 = (\omega c^T x^0 + e_j^T x^0) - \omega c^T x^0 = e_j^T x^0 = x^0_j.$$

Hence, at least one of the two values $\tilde{c}^T x^0$ and $c^T x^0$ must be fractional, which contradicts (iv).

We complete the proof of the theorem by showing two implications:

(i)⇒(v) Since P is convex, we have $\text{conv}(P \cap \mathbb{Z}^n) \subseteq P$. Thus, the claim follows if we can show that $P \subseteq \text{conv}(P \cap \mathbb{Z}^n)$. Let $v \in P$, then $v = \sum_{k\in K} \lambda_k x^k + \sum_{j\in J} \mu_j r^j$, where the x^k are the extreme points of P and the r^j are the extreme rays of P. By (i) every x^k is integral, thus $\sum_{k\in K} \lambda_k x^k \in \text{conv}(P \cap \mathbb{Z}^n)$. Since by Observation **??** the extreme rays of P and $\text{conv}(P \cap \mathbb{Z}^n)$ are the same, we get that $v \in \text{conv}(P \cap \mathbb{Z}^n)$.
(v)⇒(iv) Let $c \in \mathbb{Z}^n$ be an integral vector. Since by assumption $\text{conv}(P \cap \mathbb{Z}^n) = P$, the (P) $\max\{ c^T x : x \in P \}$ has an optimal solution in $P \cap \mathbb{Z}^n$ (If $x = \sum_i x^i \in \text{conv}(P \cap \mathbb{Z}^n)$ is a convex combination of points in $P \cap \mathbb{Z}^n$, then $c^T x \le \max_i c^T x^i$ (cf. Observation **??**)). Thus, the LP has an integral value for every integral $c \in \mathbb{Z}^n$ where the value is finite.

This shows the theorem.

Recall that each minimal nonempty face of $P(A, b)$ is an extreme point if and only if $\text{Rang}(A) = n$ (Corollary ??). Thus, we have the following result:

Beobachtung 9.3. *A nonempty polyhedron $P = P(A, b)$ with $\text{Rang}(A) = n$ is integral if and only if all of its extreme points are integral.* □

Moreover, if $P(A, b) \subseteq \mathbb{R}^n_+$ is nonempty, then $\text{Rang}(A) = n$. Hence, we also have the following corollary:

Korollar 9.4. *A nonempty polyhedron $P \subseteq \mathbb{R}^n_+$ is integral if and only if all of its extreme points are integral.* □

9.2 Totale Unimodularität

In diesem Kapitel untersuchen wir Eigenschaften der Nebenbedingungsmatrix A, so dass für jede rechte Seite b das Polyeder $\{x : Ax \leq b,\, x \geq 0\}$ ganzzahlig ist. Zur Motivation der Definition von (totaler) Unimodularität betrachten wir das Polyeder $\{x \geq 0 : Ax = b\}$, wobei A vollen Zeilenrang habe sowie A und b ganzzahlig seien. Zu jeder Ecke x gibt es eine Basis B, so dass $x_B = A_B^{-1}b$ und $x_N = 0$ gilt, vgl. Kapitel **??**. Für eine Matrix A mit $\det A_B = \pm 1$ stellt die Cramersche Regel sicher, dass A_B^{-1} ganzzahlig ist. Deshalb kann die Ganzzahligkeit von x_B und damit der Ecke x, garantiert werden, indem wir fordern, dass $\det A_B$ gleich ± 1 ist.

Definition 9.5.

(a) *Sei A eine $m \times n$ Matrix mit vollem Zeilenrang. A heißt* unimodular*, falls alle Einträge von A ganzzahlig sind und jede invertierbare $m \times m$-Untermatrix von A Determinante ± 1 hat.* unimodular

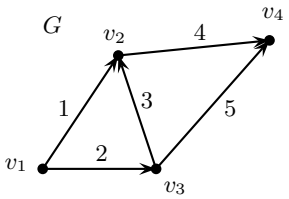
(b) *Eine Matrix A heißt* total unimodular*, falls jede quadratische Untermatrix Determinante ± 1 oder 0 hat.* total unimodular

Beachte, dass alle Einträge (die quadratischen Untermatrizen der Größe 1) einer total unimodularen Matrix entweder 0 oder ± 1 sind. Die folgenden Beobachtungen sind leicht einzusehen:

Beobachtung 9.6. *Die folgenden Aussagen sind äquivalent.*

(a) *Die Matrix A ist total unimodular.*
(b) *Die Matrix $[A\ I]$ ist unimodular.*
(c) *Die Matrix* $\begin{bmatrix} A \\ -A \\ I \\ -I \end{bmatrix}$ *ist total unimodular.*

incidence



$$A = \begin{pmatrix} 1 & 1 & 0 & 0 & 0 \\ -1 & 0 & 1 & 1 & 0 \\ 0 & -1 & -1 & 0 & 1 \\ 0 & 0 & 0 & -1 & -1 \end{pmatrix}$$

(d) Die Matrix A^T ist total unimodular.

Beweis. Übung.

Beispiel 9.7. Sei A die Knoten-Kanten-Inzidenzmatrix eines gerichteten Graphen $G = (V, E)$. Diese ist definiert als

$$A = (a_{ij})_{i \in V, j \in E} \text{ mit } a_{ij} := \begin{cases} 1 & \text{falls } j \in \delta^+(i) \\ -1 & \text{falls } j \in \delta^-(i) \\ 0 & \text{sonst} \end{cases}$$

Dann ist A total unimodular.

Beweis. Übung.

Die folgenden drei Sätze präzisieren, dass ein lineares Programm mit einer unimodularen bzw. total unimodularen Nebenbedingungsmatrix immer eine ganzzahlige Optimallösung besitzt, sofern das Optimum endlich ist.

Theorem 9.8. *Sei A eine ganzzahlige $m \times n$ Matrix mit vollem Zeilenrang. Dann ist das Polyeder $\{x \in \mathbb{R}^n : Ax = b, x \geq 0\}$ genau dann ganzzahlig für alle ganzzahligen Vektoren $b \in \mathbb{Z}^m$, wenn A unimodular ist.*

Beweis. Angenommen, A sei unimodular und $b \in \mathbb{Z}^m$ ein beliebiger ganzzahliger Vektor. Sei $\bar{x}$ eine Ecke von $\{x \in \mathbb{R}^n : Ax = b, x \geq 0\}$. Wir haben zu zeigen, dass $\bar{x}$ ganzzahlig ist. Da A vollen Zeilenrang hat, gibt es eine Basis $B \subseteq \{1, \ldots, n\}, |B| = m$, so dass $\bar{x}_B = A_B^{-1} b$ und $\bar{x}_N = 0$ ist, wobei $N = \{1, \ldots, n\} \setminus B$ gilt. Da A unimodular ist, folgern wir aus der Cramerschen Regel die Integralität von A_B^{-1} und somit auch von $\bar{x}$.

Umgekehrt sei $\{x \in \mathbb{R}^n : Ax = b, x \geq 0\}$ ganzzahlig für alle ganzzahligen Vektoren b, ebenso sei $B \subseteq \{1, \ldots, n\}$ mit A_B regulär. Wir haben zu zeigen, dass $\det A_B = \pm 1$ gilt. Sei $\bar{x}$ die zu B gehörige Ecke, d. h. es gilt $\bar{x}_B = A_B^{-1} b$ und $\bar{x}_N = 0$ mit $N = \{1, \ldots, n\} \setminus B$. Nach unserer Voraussetzung ist $\bar{x}_B = A_B^{-1} b$ ganzzahlig für alle ganzzahligen Vektoren b, also insbesondere für die Einheitsvektoren, also ist A_B^{-1} ganzzahlig. Also sind sowohl $\det A_B$ und $\det A_B^{-1}$ ganze Zahlen, woraus $\det A_B = \pm 1$ folgt. □

Korollar 9.9. *(Satz von Hoffmann und Kruskal [?])*
Sei A eine ganzzahlige Matrix. Dann ist A genau dann total unimodular, wenn $\{x : Ax \leq b, x \geq 0\}$ für alle ganzzahligen Vektoren b ganzzahlig ist.

Beweis. Aus Beobachtung **??** wissen wir, dass A genau dann total unimodular ist, wenn $[A\ I]$ unimodular ist. Weiterhin gilt für einen ganzzahligen Vektor b, dass $\{x : Ax \leq b, x \geq 0\}$ genau dann ganzzahlig ist, wenn $\{z : [A\ I]z = b, z \geq 0\}$ ganzzahlig ist (siehe Übung). Die Anwendung von Satz **??** auf die Matrix $[A\ I]$ zeigt nun die Behauptung. □

Beobachtung **??** in Verbindung mit Korollar **??** liefert weitere Charakterisierungen von totaler Unimodularität.

Korollar 9.10. *Ist A eine ganzzahlige Matrix, so sind die folgenden Aussagen sind äquivalent.*

(a) Die Matrix A ist total unimodular.
(b) für alle ganzzahligen Vektoren a, b, l, u gilt

$$\{ x : a \leq Ax \leq b,\ l \leq x \leq u \} \subset \mathbb{Z}\,.$$

(c) für alle ganzzahligen Vektoren b und u gilt

$$\{ x : Ax = b,\ 0 \leq x \leq u \} \subset \mathbb{Z}\,.$$

Zwei Anwendungsbeispiele aus der Kombinatorischen Optimierung

Eine interessante Anwendung von total unimodularen Matrizen ist das Max-Flow-Min-Cut Theorem von [**?**]. Aus Korollar **??** wissen wir, dass eine Matrix A genau dann total unimodular ist, wenn für alle ganzzahligen Vektoren c, b, u das Optimum des linearen Programms

$$\max \left\{ c^T x : Ax = b,\ 0 \leq x \leq u \right\}$$

von einem ganzzahligen Punkt angenommen wird. Da A genau dann total unimodular ist, wenn A^T total unimodular ist, gilt darüber hinaus dieselbe Bedingung für das duale lineare Programm, d. h.

$$\min \left\{ b^T z + u^T y : A^T z + y \geq c,\ y \geq 0 \right\}$$

hat eine ganzzahlige Optimallösung.

Korollar 9.11. *Eine ganzzahlige Matrix A ist genau dann total unimodular, wenn für alle ganzzahligen Vektoren u, b, c beide Seiten der Dualitätsgleichung*

$$\begin{aligned} &\max \left\{ c^T x :\ Ax = b,\ 0 \leq x \leq u \right\} \\ &\qquad = \min \left\{ b^T z + u^T y : A^T z + y \geq c,\ y \geq 0 \right\} \end{aligned}$$

ganzzahlige Lösungen x, y und z haben.

Aus Korollar **??** leiten wir das Max-Flow-Min-Cut Theorem ab.

Theorem 9.12. *Sei $G = (V, E)$ ein gerichteter Graph mit ganzzahligen Bogenkapazitäten u. Seien $s \neq t$ zwei Knoten in G. Dann ist der Wert eines maximalen Flusses von s nach t gleich dem Wert eines minimalen Schnittes, der s und t trennt.*

Beweis. Sei A die Knoten-Kanten Inzidenzmatrix des gerichteten Graphen $G = (V, E \cup \{(t, s)\})$. Offensichtlich ist ein maximaler (s,t)-Fluss die Lösung des linearen Programms $\max\{x_{ts} : Ax = 0,\, 0 \leq x \leq u\}$, wobei $u_{ts} := \infty$ gilt. Aus Korollar **??** folgt, dass es ganzzahlige optimale Lösungen $\bar{x}, \bar{y}$ und $\bar{z}$ für die Optimierungsprobleme

$$\begin{aligned} &\max\{\, x_{ts} : Ax = 0,\, 0 \leq x \leq u \,\} \\ &\qquad = \min\left\{\, u^T y : A^T z + y \geq e_{ts},\, y \geq 0 \,\right\} \end{aligned}$$

gibt. In Worten ausgedrückt: Der Wert $\bar{x}_{ts}$ eines maximalen (s,t)-Flusses ist gleich dem Wert $u^T\bar{y}$, wobei $\bar{y}$ die folgenden Bedingungen erfüllt:

$$\begin{aligned} \bar{z}_u - \bar{z}_v + \bar{y}_{uv} &\geq 0\,, \text{ falls } uv \neq ts \\ \bar{z}_t - \bar{z}_s + \bar{y}_{ts} &\geq 1\,. \end{aligned}$$

Referenz auf nicht vorhandenen Satz

Aus dem Satz vom schwachen komplementären Schlupf, vgl. 3.13, und $u_{ts} = \infty$ folgern wir, das $\bar{y}_{ts} = 0$ und damit $\bar{z}_t \geq 1 + \bar{z}_s$ gilt. Sei $W := \{w \in V : \bar{z}_w \geq \bar{z}_t\}$. Dann gilt $\bar{y}_{vw} \geq 1$ für alle $vw \in \delta^-(W)$ aufgrund von $z_w \geq z_t > z_v$. Daraus folgern wir

$$u^T\bar{y} \geq \sum_{vw \in \delta^-(W)} u_{vw}\,.$$

Anders ausgedrückt, der maximale (s,t)-Fluss ist größer oder gleich dem Wert des (s,t)-Schnittes $\delta^-(W)$. Da der maximale Fluss nicht größer als der Wert eines (s,t)-Schnittes sein kann, folgt die Behauptung. □

Ein weiterer Anwendungsbereich total unimodularer Matrizen sind ungerichtete Graphen. Sei G ein ungerichteter Graph und A die Kanten-Knoten Inzidenzmatrix von G. Im Folgenden untersuchen wir Bedingungen, unter denen das Polyeder $P = \{x : Ax \leq b,\, x \geq 0\}$ ganzzahlig ist. Dies ist ein interessantes Problem, denn für den Fall $b = 1$

entsprechen die ganzzahligen Lösungen in P stabilen Mengen in G (d. h. einer Teilmenge S der Knoten, so dass jedes Paar von Knoten aus S nicht benachbart ist, in Formeln $E(S) = \emptyset$) bzw. zulässigen Lösungen des Set Packing Problems aus Beispiel **??**. Das folgende Beispiel zeigt, dass A nicht total unimodular ist, falls G einen ungeraden Kreis enthält, also nicht bipartit ist.

Beispiel 9.13. Sei G ein Graph und C ein ungerader Kreis in G. Betrachte das lineare Programm

$$\max \left\{ c^T x : 0 \leq x \leq 1,\, x_i + x_j \leq 1,\, (i,j) \in E \right\}$$

mit $c = \chi^{V(C)}$. Es gilt:

(a) $x_i^* = \frac{1}{2}$ für $i \in V(C)$, $x_i^* = 0$ für $i \notin V(C)$, löst das lineare Programm.
(b) x^* ist keine Konvexkombination von Inzidenzvektoren von stabilen Mengen.

Beweis. Übung.

Theorem 9.14. *A ist genau dann total unimodular, wenn G bipartit ist.*

Beweis. Um die Notwendigkeit der Bedingung zu erkennen, beachte man, dass $G = (V, E)$ genau dann bipartit ist, wenn E keinen Kreis ungerader Länge enthält. Angenommen, G ist nicht bipartit, dann enthält G einen Kreis $C \subseteq E$ ungerader Länge. Sei $c = \chi^{V(C)}$. Dann wird $\max\{c^T x : Ax \leq 1,\, x \geq 0\}$ von einem Vektor x^* mit $x_i^* = \frac{1}{2}$ angenommen, falls Knoten $i \in V(C)$ und $x_i^* = 0$ ist, für den anderen Fall siehe Beispiel **??** (a). Überdies kann x^* nicht eine Konvexkombination von Inzidenzvektoren stabiler Mengen in G sein, betrachte dazu Beispiel **??** (b). Also gibt es eine Ecke des Polyeders $\{x \in \mathbb{R}^n : Ax \leq 1,\, x \geq 0\}$, die nicht als Konvexkombination von Inzidenzvektoren stabiler Mengen darstellbar ist und folglich nicht ganzzahlig ist. Aus der Betrachtung von Korollar **??** folgt, dass A nicht total unimodular ist.

Ist umgekehrt G bipartit, so bezeichnen wir mit V_1 und V_2 die beiden Partitionen der Knotenmengen. Sei $b \in \mathbb{Z}^m$ und $c \in \mathbb{R}^n$ eine Zielfunktion. Wir werden im Verlauf des Beweises zeigen, dass das lineare Programm $\max\{c^T x : Ax \leq b,\, x \geq 0\}$ eine ganzzahlige Lösung hat. Betrachte die folgende Heuristik:

Berechne eine optimale Lösung x^* des linearen Problems $c^* = \max\{c^T x : Ax \leq b,\, x \geq 0\}$. Sei U eine

gleich verteilte Zufallszahl auf dem Intervall $[0,1]$. Für $i = 1, \ldots, n$ definiere

$$z_i := \begin{cases} \lceil x_i^* \rceil & \text{falls } i \in V_1 \text{ und } x_i^* - \lfloor x_i^* \rfloor \geq U, \\ \lceil x_i^* \rceil & \text{falls } i \in V_2 \text{ und } x_i^* - \lfloor x_i^* \rfloor > 1 - U, \\ \lfloor x_i^* \rfloor & \text{sonst.} \end{cases}$$

Dann ist z ist ein ganzzahliger Punkt mit $z \geq 0$. Betrachte die i-te Zeile von A, die sich zu $e^k + e^l$ mit $(k,l) \in E$ und $k \in V_1$, $l \in V_2$ ergibt.

Gilt $\lfloor x_k^* \rfloor + \lfloor x_l^* \rfloor = b_i$, dann bestimmt die Heuristik den Wert $\lfloor x_k^* \rfloor$ zu z_k und den Wert $\lfloor x_l^* \rfloor$ zu z_l. Wir folgern $z_k + z_l = b_i$.

Gilt $\lfloor x_k^* \rfloor + \lfloor x_l^* \rfloor \leq b_i - 2$, dann folgt $\lceil x_k^* \rceil + \lceil x_l^* \rceil \leq b_i$. In diesem Fall folgern wir $z_k + z_l \leq b_i$.

Andernfalls ist $\lfloor x_k^* \rfloor + \lfloor x_l^* \rfloor = b_i - 1$. Dann ist $x_k^* - \lfloor x_k^* \rfloor + x_l^* - \lfloor x_l^* \rfloor \leq 1$. Ist $x_k^* - \lfloor x_k^* \rfloor \geq U$, dann gilt $x_l^* - \lfloor x_l^* \rfloor < 1 - U$ und umgekehrt. Also werden nicht beide Werte x_k^* und x_l^* aufgerundet, woraus $z_k + z_l \leq b_i$ folgt.

Dies zeigt $Az \leq b$. Da U eine gleichverteilte Zufallsvariable im Intervall $[0,1]$ ist, gilt dies für $1 - U$ ebenso. Für $k \in V_1$ erhalten wir

$$\begin{aligned} \text{Prob}(z_k = \lceil x_k^* \rceil) &= \text{Prob}(U \leq x_k^* - \lfloor x_k^* \rfloor) \\ &= x_k^* - \lfloor x_k^* \rfloor \,, \end{aligned}$$

und für $l \in V_2$ ergibt sich

$$\begin{aligned} \text{Prob}(z_l = \lceil x_l^* \rceil) &= \text{Prob}(1 - U < x_l^* - \lfloor x_l^* \rfloor) \\ &= 1 - \text{Prob}(1 - U \geq x_l^* - \lfloor x_l^* \rfloor) \\ &= 1 - \text{Prob}(U \leq 1 - (x_l^* - \lfloor x_l^* \rfloor)) \\ &= 1 - (1 - (x_l^* - \lfloor x_l^* \rfloor)) \\ &= x_l^* - \lfloor x_l^* \rfloor \,. \end{aligned}$$

Insgesamt haben wir gezeigt:

$$\text{Prob}(z_k = \lceil x_k^* \rceil) = x_k^* - \lfloor x_k^* \rfloor \text{ für alle } k \in V \,.$$

Bezeichnen wir mit c^{IP} den Zielfunktionswert einer maximalen stabilen Menge in G bzgl. c mit $c^H = c^T z$ den Zielfunktsionswert der heuristischen Lösung und mit $E(\cdot)$ den Erwartungswertoperator, so erhalten wir

$$\begin{aligned} c^* \geq c^{\mathrm{IP}} \geq E(c^H) &= \sum_{i \in V} c_i \lfloor x_i^* \rfloor + \sum_{i \in V} c_i \operatorname{Prob}(z_i = \lceil x_i^* \rceil) \\ &= \sum_{i \in V} c_i \lfloor x_i^* \rfloor + \sum_{i \in V} c_i (x_i^* - \lfloor x_i^* \rfloor) \\ &= c^* \,. \end{aligned}$$

Es folgt $c^* = c^{\mathrm{IP}}$. Also hat das Polyeder $\{x \in \mathbb{R}^n : Ax \leq b, x \geq 0\}$ nur ganzzahlige Werte. Korollar **??** impliziert, dass A total unimodular ist. □

Korollar 9.15. *Betrachte das Zuordnungsproblem aus Beispiel (fixme Skript 1.4). Die LP-Relaxierung des binären Programms (die man bekommt, indem man $x_{ij} \in \{0,1\}$ durch $x_{ij} \in [0,1]$ ersetzt) hat immer eine ganzzahlige Optimallösung.*

Referenz auf nicht vorhandenen Satz

9.3 Ein ganzahliges Farkas Lemma

In diesem Abschnitt zeigen wir, dass jede rationale Matrix in eine bestimmte Form, die sogenannte Hermitesche Normalform, gebracht werden kann. Diese Form wird uns helfen, ein ganzzahliges Analogon zum Farkas-Lemma zu beweisen. Dieses Lemma werden wir noch häufiger benötigen, um die Ganzzahligkeit bestimmter Polyeder zu zeigen.

Referenz auf nicht vorhandenen Satz

Definition 9.16.

(a) *Die folgenden Operationen an einer Matrix heißen* unimodulare (elementare) Spalten- (Zeilen-) Operationen*:*
   1. *Vertauschen zweier Spalten (Zeilen).*
   2. *Multiplikation einer Spalte (Zeile) mit -1.*
   3. *Addition eines ganzzahligen Vielfachen einer Spalte (Zeile) zu einer anderen Spalte (Zeile).*

unimodulare (elementare) Spalten- (Zeilen-) Operationen

(b) *Eine Matrix A mit vollem Zeilenrang ist in* Hermitescher Normalform*, wenn sie die Form $[B\ 0]$ hat, wobei B eine reguläre, nicht-negative, untere Dreiecksmatrix ist und jede Zeile genau einen maximalen Eintrag auf der Hauptdiagonalen hat.*

Hermitescher Normalform

Theorem 9.17. *Jede rationale Matrix mit vollem Zeilenrang kann mit elementaren Spaltenoperationen in Hermitesche Normalform gebracht werden.*

Beweis. Sei A eine rationale Matrix mit vollem Zeilenrang. Wir können annehmen, A sei ganzzahlig (andernfalls skaliere die Matrix mit einem geeigneten Faktor, der am Ende der durchzuführenden Operationen wieder dividiert wird). Wir zeigen zunächst, dass A in die Matrix $[B\ 0]$ transformiert werden kann, wobei B eine untere Dreiecksmatrix und $B_{ii} > 0$ für alle i ist. Angenommen, wir haben A bereits in die Form

$$\begin{bmatrix} B & 0 \\ C & D \end{bmatrix}$$

gebracht, wobei B eine untere Dreiecksmatrix mit positiver Diagonale ist. Mittels elementarer Spaltenoperationen bringen wir die erste Zeile von $D \in \mathbb{R}^k$ in eine Form, so dass $D_{11} \geq D_{12} \geq \ldots \geq D_{1k} \geq 0$ gilt und $\sum_{i=1}^{k} D_{1i}$ minimal ist. Da A vollen Zeilenrang hat, folgt $D_{11} > 0$. Wir behaupten, dass $D_{1i} = 0$ für $i = 2, \ldots, k$ gilt. Falls $D_{12} > 0$ ist, subtrahieren wir die zweite Spalte von der ersten und nach eventueller Umordnung der ersten und zweiten Spalte erhalten wir wieder die Elemente der ersten Zeile in nicht steigender Reihenfolge, wobei deren Summe streng streng monoton abnimmt, was ein Widerspruch zur Annahme ist, dass $\sum_{i=1}^{k} D_{1i}$ minimal ist. Nachdem wir diese Schritte n mal durchgeführt haben, haben wir A nach $[B\ 0]$ transformiert, wobei B eine untere Dreiecksmatrix mit positiver Hauptdiagonale ist.

Um Hermitesche Normalform zu erreichen, müssen wir $0 \leq B_{ij} < B_{ii}$ für jedes $i > j$ garantieren können. Dies kann erreicht werden, indem wir ein entsprechendes ganzzahliges Vielfaches von Spalte i zu Spalte j für jedes $i > j$ addieren. Diese Operationen müssen in der Reihenfolge $(2,1), (3,1), (3,2), (4,1), (4,2), \ldots$ durchgeführt werden. □

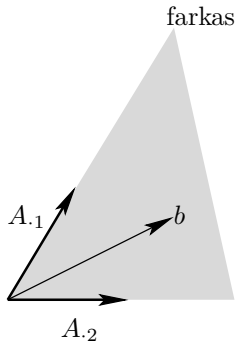
Anmerkung 9.18.

(a) Die Operationen, die im Beweis von Satz **??** ausgeführt wurden, können durch eine unimodulare Matrix ausgedrückt werden, d. h. es gibt eine unimodulare Matrix U, so dass
$$[B\ 0] = AU$$
und $[B\ 0]$ ist in Hermitescher Normalform.

(b) Für jede rationale Matrix mit vollem Zeilenrang gibt es eine *eindeutige* Hermitesche Normalform und damit eine eindeutige unimodulare Matrix U, siehe [?].

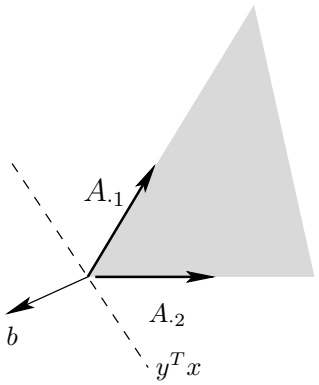
Theorem 9.19 (Ganzzahliges Analogon des Farkas-Lemma). *Sei $A \in \mathbb{Q}^{m\times n}$ eine rationale Matrix und $b \in \mathbb{Q}^m$ ein rationaler Vektor. Dann hat genau eines der beiden folgenden Systeme eine Lösung.*

$$\begin{aligned} Ax &= b & y^T A &\in \mathbb{Z}^m \\ x &\in \mathbb{Z}^n & y^T b &\notin \mathbb{Z} \\ & & y &\in \mathbb{Q}^m \end{aligned}$$



Beweis. Beide Gleichungssysteme können nicht gleichzeitig eine Lösung haben, da aus der Integralität von x und $y^T A$ auch diejenige von $y^T b = y^T Ax$ folgt.

Angenommen, $y^T A \in \mathbb{Z}^n, y^T b \notin \mathbb{Z}$, $y \in \mathbb{Q}^m$ habe keine Lösung, d. h. für alle $y \in \mathbb{Q}^m$, für die $y^T A$ ganzzahlig ist, ist $y^T b$ auch ganzzahlig. Wir müssen zeigen, dass $Ax = b$, $x \in \mathbb{Z}^n$ eine Lösung hat.



Zunächst hat $Ax = b$ eine (möglicherweise gebrochene) Lösung, da es anderweitig rationale Vektoren y mit $y^T A = 0$ und $y^T b \neq 0$ gäbe (vgl. das Eliminationsverfahren nach Gauss). Nach entsprechender Reskalierung gibt es ein y mit $y^T A = 0$ und $y^T b = \frac{1}{2}$, Widerspruch!. Wir können also annehmen, dass A vollen Zeilenrang hat. Der Satz ist invariant bzgl. elementarer Spaltenoperationen. Also können wir nach Satz **??** annehmen, dass A in Hermitescher Normalform $A = [B, 0]$ ist. Da $B^{-1}[B\ 0] = [I\ 0]$ eine ganzzahlige Matrix ist, folgt aus unserer Annahme, dass $B^{-1}b$ ebenso ganzzahlig ist (wähle $y = B^{-1}_{i,\cdot}$ für $i = 1, \ldots, m$). Wegen

$$[B\ 0]\begin{pmatrix} B^{-1}b \\ 0 \end{pmatrix} = b$$

ist der Vektor $x := \binom{B^{-1}b}{0}$ eine ganzzahlige Lösung von $Ax = b$. □

Beispiel 9.20. Sei $a \in \mathbb{Z}^n$, $\alpha \in \mathbb{Z}$ und $P_I := \operatorname{conv}(\{x \in \mathbb{Z}^n : a^T x \leq \alpha\})$. Weiterhin sei $k = \gcd(a_1, \ldots, a_n)$ der größte gemeinsame Teiler der Komponenten von a. Dann folgt aus Satz **??**, dass

$$P_I = \left\{ x \in \mathbb{R}^n : \sum_{i=1}^{n} \tfrac{a_i}{k} x_i \leq \lfloor \tfrac{\alpha}{k} \rfloor \right\}.$$

Beweis. Übung.

9.4 Totale Duale Integralität

In Abschnitt **??** haben wir gesehen, dass totale Unimodularität einer ganzzahligen Matrix A sicherstellt, dass für jede ganzzahlige rechte Seite b das Polyeder $P_I = \text{conv}\{x \in \mathbb{Z}^n : Ax \leq b\}$ durch die Originalungleichungen $Ax \leq b$ vollständig beschrieben wird. Legen wir uns dagegen auf eine bestimmte rechte Seite b fest, dann ist TDI (total dual integrality) das richtige Konzept um Ganzzahligkeit eines Polyeders zu garantieren.

total dual integral (TDI)

Definition 9.21. *Sei $A \in \mathbb{Q}^{m\times n}$, $b \in \mathbb{Q}^m$. Das System von Ungleichungen $Ax \leq b$ heißt* total dual integral (TDI)*, falls es für jeden ganzzahligen Vektor $c \in \mathbb{Z}^n$, für den $z := \min\{b^T y : A^T y = c, y \geq 0\}$ endlich ist, einen ganzzahligen Vektor $y^* \in \mathbb{Z}^m$ mit $A^T y^* = c$, $y^* \geq 0$ gibt, so dass $b^T y^* = z$ gilt.*

Die TDI-Eigenschaft eines Systems $Ax \leq b$ hat eine *geometrische* Bedeutung: Sei c ein ganzzahliger Vektor, der in dem von den Zeilen von A aufgespannten Kegel liegt, also ein $y \geq 0$ mit $A^T y = c$ existiert. Unter allen Möglichkeiten c als konische Kombination der Zeilen von A zu schreiben sei S die Menge der kürzesten (bzgl. b) konischen Kombination, d. h.

$$S = \left\{ y \geq 0 : A^T y = c, \text{ so dass } b^T y \text{ minimal ist} \right\} .$$

Dann ist $Ax \leq b$ TDI, falls es einen ganzzahligen Vektor in S gibt. Dies bedeutet mit anderen Worten, dass unter allen kürzesten Möglichkeiten, c als konische Kombination der Zeilen von A zu schreiben, eine dabei ist, die ganzzahlig ist. Wir werden sehen, dass dies in engem Zusammenhang mit Hilbert-Basen steht, die in der ganzzahligen Programmierung eine wichtige Rolle spielen.

Aus der Definition der TDI-Eigenschaft folgt direkt, dass $Ax \leq b$ TDI ist, falls die Matrix A total unimodular ist.

Beachte auch, dass TDI eine Eigenschaft eines Ungleichungssystems ist und nicht eine Eigenschaft des Polyeders $P = \{x \in \mathbb{R}^n : Ax \leq b\}$. Wir werden sehen, dass es viele Möglichkeiten geben kann P zu beschreiben, aber es gibt nur eine, die auch die TDI-Eigenschaft besitzt. Dazu kann es notwendig sein, viele redundante Ungleichungen zur Ausgangsformulierung hinzufügen zu müssen.

Der folgende Satz gibt uns den Zusammenhang zwischen der Eigenschaft eines Ungleichungssystems $Ax \leq b$,

TDI zu sein, und der Ganzzahligkeit des zugehörigen Polyeders $P = \{x \in \mathbb{R}^n : Ax \leq b\}$.

Theorem 9.22. *Ist $Ax \leq b$ TDI und b ganzzahlig, dann ist $\{x \in \mathbb{R}^n : Ax \leq b\}$ ganzzahlig.*

Beweis. Sei $P = \{x \in \mathbb{R}^n : Ax \leq b\}$ und $\emptyset \neq F \neq P$ eine minimale Seitenfläche von P. Wir setzen $k := |\text{eq}(F)|$. Da F minimal ist, folgt $F = \{x \in P : A_{\text{eq}(F)}x = b_{\text{eq}(F)}\} = \{x \in \mathbb{R}^n : A_{\text{eq}(F)}x = b_{\text{eq}(F)}\}$. Wir müssen zeigen, dass F ganzzahlige Punkte enthält.

Enthält F keine ganzzahligen Punkte, so existiert ein $y \in \mathbb{Q}^k$ mit $c := y^T A_{\text{eq}(F)} \in \mathbb{Z}^n$ und $\gamma := y^T b_{\text{eq}(F)} \notin \mathbb{Z}$, siehe Satz **??**. Wir können annehmen, dass y nicht-negativ ist (andernfalls wähle $s \in \mathbb{Z}_+^k$, groß genug, so dass $y + s \geq 0$, $(y+s)^T A \in \mathbb{Z}^n$ und $(y+s)^T b_{\text{eq}(F)} \notin \mathbb{Z}$. Wir behaupten, dass $\max\{c^T x : Ax \leq b\}$ existiert und von jedem $\hat{x} \in F$ angenommen wird. Die Behauptung ist korrekt, da für alle $x \in P$ und $\hat{x} \in F$ gilt

$$c^T x = y^T A_{\text{eq}(F)} x \leq y^T b_{\text{eq}(F)} = y^T A_{\text{eq}(F)} \hat{x} = c^T \hat{x}.$$

Referenz auf nicht vorhandenen Satz

Aufrund des Dualitätssatzes der Linearen Optimierung (vgl.) und der Tatsache, dass $Ax \leq b$ TDI ist, folgern wir, dass der Wert $\min\{b^T y : A^T y = c, y \geq 0\}$ existiert und für einen ganzzahligen Vektor angenommen wird. Da b ganzzahlig ist, ist der Wert $\min\{b^T y : A^T y = z, y \geq 0\}$ eine ganze Zahl. Diese Zahl muss mit dem Wert $\max\{c^T x : Ax \leq b\} = \gamma \notin \mathbb{Z}$ übereinstimmen, ein Widerspruch. Also enthält F ganzzahlige Punkte. □

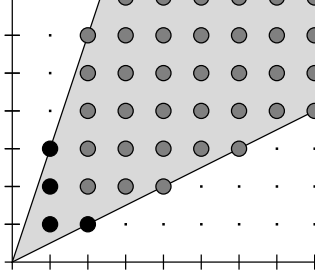
Wir haben bereits die geometrische Interpretation von TDI betrachtet und den Zusammenhang zu Hilbert-Basen angedeutet. Dies werden wir nun präzisieren.

Definition 9.23. *Sei $C \subseteq \mathbb{R}^n$ ein rationaler polyedrischer Kegel. Eine endliche Teilmenge $H = \{h^1, \ldots, h^t\} \subset C$ heißt* Hilbert-Basis *von C, falls sich jeder ganzzahlige Punkt $z \in C \cap \mathbb{Z}^n$ als nicht-negative ganzzahlige Linearkombination von Elementen aus H darstellen lässt, d. h.*

Hilbert-Basis

$$z = \sum_{i=1}^{t} \lambda_i h^i$$

mit $\lambda_1, \ldots, \lambda_t \in \mathbb{N}_0$. Eine Hilbert-Basis heißt ganzzahlig, *falls $H \subseteq \mathbb{Z}^n$ ist.*



Beispiel 9.24. Betrachte $C = \text{cone}(\{\binom{1}{3}, \binom{2}{1}\})$. Dann ist $H = \{\binom{1}{3}, \binom{1}{2}, \binom{1}{1}, \binom{2}{1}\}$ eine Hilbert-Basis von C, vgl. Abbildung **??**.

Anmerkung 9.25.

(a) Für jeden rationalen polyedrischen Kegel existiert eine ganzzahlige Hilbert-Basis.
(b) Falls C spitz ist, ist die ganzzahlige Hilbert-Basis eindeutig bestimmt.

Beweis. Zum Beweis siehe z. B. Schrijver [?], Kapitel 16.4.

Theorem 9.26. *Sei $A \in \mathbb{Q}^{m\times n}$ und $b \in \mathbb{Q}^m$. Das Ungleichungssystem*

$$Ax \leq b$$

hat genau dann die TDI-Eigenschaft, wenn für jede Seitenfläche F von $P = \{x \in \mathbb{R}^n : Ax \leq b\}$ die Zeilen von $A_{\text{eq}(F)\cdot}$ eine Hilbert-Basis des Kegels $C(A_{\text{eq}(F)\cdot})$ bilden.

Um den Beweis von Satz **??** nachzuvollziehen, ist folgende geometrische Interpretation hilfreich: Sei c ein ganzzahliger Vektor, der in dem von den Zeilen von A aufgespannten Kegel liegt, so dass der Wert $\min\{b^T y : A^T y = c,\, y \geq 0\}$ existiert. Dann wissen wir, dass c das Optimum an einer Seitenfläche von P annimmt. Aus der Theorie der Linearen Programmierung den Satz vom komplementären Schlupf, Satz 3.13, DO I wissen wir, dass c in dem von den Zeilen von $A_{\text{eq}(F)\cdot}$ aufgespannten Kegel liegt, wenn F eine Seitenfläche ist, an der c das Optimum annimmt. Damit existiert der Wert $\min\{b^T y : A^T y = c,\, y \geq 0\}$. Die Bedingung, dass die von F induzierten Zeilen eine Hilbert-Basis bilden, ist äquivalent dazu, dass c als nicht-negative ganzzahlige Kombination von F induzierter Zeilen dargestellt werden kann. In Formeln bedeutet dies, es gibt einen ganzzahligen Vektor $y^* \in \mathbb{Z}^m$, $A^T y^* = c$, $y^* \geq 0$ mit $b^T y^* = \min\{b^T y : A^T y = c, y \geq 0\}$. Dies ist die Definition der TDI-Eigenschaft eines Ungleichungssystems $Ax \leq b$.

Referenz auf nicht vorhandenen Satz

Beweis. Sei $Ax \leq b$ TDI. Weiter sei $F \neq \emptyset$ eine Seitenfläche von P und $c \in C(A_{\text{eq}(F)}) \cap \mathbb{Z}^n$. Wir müssen zeigen, dass c als eine nichtnegative ganzzahlige Kombination der Zeilenvektoren von $A_{\text{eq}(F)}$ dargestellt werden kann. Wir wissen, dass c eine nichtnegative Linearkombination der Zeilenvektoren von $A_{\text{eq}(F)}$ ist, d. h. es gibt ein $\hat{y} \geq 0$ mit $\hat{y}_i = 0$ für alle $i \notin \text{eq}(F)$ und $c = \hat{y}^T A$.

Wir behaupten nun, dass $\max\{c^T x : Ax \leq b\}$ existiert und für jedes $\hat{x} \in F$ angenommen wird. Dies folgt, da für alle $x \in P$ und $\hat{x} \in F$ die Aussage

$$c^T x = \hat{y}^T Ax \leq \hat{y}^T b = \hat{y}^T A\hat{x} = c^T \hat{x}$$

gilt. Aus dem Dualitätssatz der Linearen Optimierung, 3.7, DO I, schließen wir, dass der Wert $\min\{b^T y : A^T y = c,\, y \geq 0\}$ existiert. Verwenden wir, dass $Ax \leq b$ TDI ist, so erhalten wir

Referenz auf nicht vorhandenen Satz

$$\min\{b^T y : A^T y = c,\, y \geq 0\} = \max\{c^T x : Ax \leq b\}\,,$$

und dass es einen ganzzahligen Vektor y^* gibt, so dass $b^T y^* = \min\{b^T y : A^T y = c,\, y \geq 0\}$ gilt. Da jedes $\hat{x} \in F$ eine optimale Lösung des Programms $\max\{c^T x : Ax \leq b\}$ ist, gibt es $\hat{x}$, so dass $A^T_{i,\cdot}\hat{x} = b_i$ für alle $i \in \text{eq}(F)$ und $A^T_{i,\cdot}\hat{x} < b_i$ für alle $i \notin \text{eq}(F)$ gilt. Aus dem Satz vom schwachem komplementären Schlupf (vgl. Satz 3.13) folgt $y^*_i = 0$ für alle $i \notin \text{eq}(F)$. Damit ist der erste Teil des Beweises vollendet, da y^* ganzzahlig, $y^*_i = 0$ für alle $i \notin \text{eq}(F)$, $y^*_i \geq 0$ für alle $i \in \text{eq}(F)$ und c eine nichtnegative ganzzahlige Kombination der Vektoren von $A_{\text{eq}(F)}$ ist.

Referenz auf nicht vorhandenen Satz

Für die Umkehrung sei $c \in \mathbb{Z}^n$, so dass

$$\gamma := \min\left\{\, b^T y : A^T y = c,\, y \geq 0 \,\right\}$$

existiert. Aus dem Dualitätssatz der Linearen Optimierung (vgl. Satz 3.13, DO I) wissen wir, dass der Wert $\gamma := \max\{c^T x : Ax \leq b\}$ existiert. Sei $F = \{x \in \mathbb{R}^n : Ax \leq b, c^T x = \gamma\}$ eine Seitenfläche von P, in der die Funktion ihren Optimalwert annimmt. Wir bemerken, dass $c \in C(A_{\text{eq}(F)})$ aufgrund des Satzes vom schwachem komplementären Schlupf , vgl. Satz 3.13, DO I, gilt und es gibt einen Vektor $y^* \geq 0$ mit $y^*_i = 0$ für alle $i \notin \text{eq}(F)$ und es ist $c = \sum_{i \in \text{eq}(F)} A_{i\cdot} y^*_i$. Da die Menge $A_{\text{eq}(F)}$ eine Hilbert-Basis von $C(A^T_{\text{eq}(F)})$ ist, gibt es einen nichtnegativen ganzzahligen Vektor $\tilde{y}$, so dass $c = \sum_{i \in \text{eq}(F)} A_{i\cdot} \tilde{y}_i$. Setzen wir $y_i := 0$ für alle $i \notin \text{eq}(F)$ und $y_i := \tilde{y}_i$ für alle $i \in \text{eq}(F)$, so haben wir den ganzzahligen Vektor y gefunden, so dass $A^T y = c$ und $y \geq 0$ ist. Weiter gilt für $x \in F$

Referenz auf nicht vorhandenen Satz

Referenz auf nicht vorhandenen Satz

$$b^T y = \sum_{i \in \text{eq}(F)} y_i b_i = \sum_{i \in \text{eq}(F)} y_i A^T_{i\cdot} x = c^T x = \gamma\,.$$

Also ist $Ax \leq b$ total dual integral. □

Als Anwendung von Satz **??** zeigen wir, dass TDI eines Ungleichungssystems erhalten bleibt, wenn wir uns auf Teilsysteme, die Seitenflächen des mit dem Originalsystem assoziierten Polyeders induzieren, beschränken.

Korollar 9.27. *Sei $A \in \mathbb{Q}^{m\times,n}, b \in \mathbb{Q}^m, c \in \mathbb{Q}^n, d \in \mathbb{Q}$. Ist das Ungleichungssystem $Ax \leq b$, $c^T x \leq d$ TDI, so ist auch das System $Ax \leq b$, $c^T x \leq d$, $-c^T x \leq -d$ TDI.*

Beweis. Sei F eine Seitenfläche des Polyeders $\{x \in \mathbb{R}^n : Ax \leq b,\, c^T x \leq d,\, -c^T x \leq -d\}$. Dann ist F ebenso eine Seitenfläche des Polyeders $\{x \in \mathbb{R}^n : Ax \leq b,\, c^T x \leq d\}$. Da $Ax \leq b, c^T x \leq d$ TDI ist, folgern wir mit Satz **??**, dass $A^T_{\mathrm{eq}(F)}, c$ eine Hilbert-Basis des Kegels $C(A^T_{\mathrm{eq}(F)}, c)$ ist. Sei $z \in C(A^T_{\mathrm{eq}(F)}, c, -c) \cap \mathbb{Z}^n$. Dann ist z von der Form

$$z = \sum_{i\in \mathrm{eq}(F)} \lambda_i A_{i\cdot} + \mu c - \sigma c$$

mit $\lambda_i \geq 0$, $i \in \mathrm{eq}(F)$, $\mu \geq 0$, $\sigma \geq 0$. Dies ist äquivalent zu $z + \sigma c \in C(A^T_{\mathrm{eq}(F)}, c)$. Sei $\sigma' \in \mathbb{N}$, $\sigma' \geq \sigma$ so gewählt, dass $\sigma' c \in \mathbb{Z}$ ist. Es folgt $z + \sigma' c \in C(A^T_{\mathrm{eq}(F)}, c)$. Da $A^T_{\mathrm{eq}(F)}, c$ eine Hilbert-Basis von $C(A^T_{\mathrm{eq}(F)}, c)$ ist, können wir $z + \sigma' c$ auch schreiben als

$$z + \sigma' c = \sum_{i\in \mathrm{eq}(F)} \lambda'_i A_{i\cdot} + \mu' c\,,$$

wobei $\lambda'_i \geq 0$ ganzzahlig ist für alle $i \in \mathrm{eq}(F)$ und ganzzahliges $\mu' \geq 0$. Es folgt, dass

$$z = \sum_{i\in \mathrm{eq}(F)} \lambda'_i A_{i\cdot} + \mu' c - \sigma' c\,,$$

d. h. z kann als nichtnegative ganzzahlige Kombination der Erzeugenden des Kegels $C(A^T_{\mathrm{eq}(F)}, c, -c)$ ausgedrückt werden. Daraus folgt, dass $A^T_{\mathrm{eq}(F)}, c, -c$ eine Hilbert-Basis von $C(A^T_{\mathrm{eq}(F)}, c, -c)$ ist. Aus Satz **??** folgt nun die Behauptung. □

Das folgende Theorem gibt eine weitere Charakterisierung von TDI über Hilbert-Basen. Diese ist hilfreich, wenn man algorithmisch überprüfen möchte, ob ein Ungleichungssystem TDI ist.

Theorem 9.28. *Sei $A \in \mathbb{Q}^{m\times n}$ und $b \in \mathbb{Q}^m$ rational, so dass $P = \{x \in \mathbb{R}^n : Ax \leq b\} \neq \emptyset$ gilt. Das System $Ax \leq b$ ist genau dann TDI, wenn die beiden folgenden Bedingungen erfüllt sind.*

(1) Die Zeilen von A bilden eine Hilbert-Basis des Kegels, der von den Zeilen von A aufgespannt wird.
(2) Für jede Teilmenge $S \subseteq \{1, \ldots, m\}$ hat das lineare Programm

$$\min \left\{ b^T y : A^T y = \sum_{i \in S} A_{i\cdot},\ y \geq 0 \right\}$$

eine ganzzahlige Optimallösung.

Beweis. Bedingung (1) ist notwendig: Ist $Ax \leq b$ TDI, dann bilden die Zeilen von A eine Hilbert-Basis. Wäre dies nicht der Fall, gäbe es ein $z \in C(A_{1\cdot}, \ldots, A_{m\cdot}) \cap \mathbb{Z}^n$, das nicht als nichtnegative ganzzahlige Kombination von $A_{1\cdot}, \ldots, A_{m\cdot}$ geschrieben werden kann. Da $P \neq \emptyset$, existierte dann aber der Wert $\min\{b^T y : A^T y = z,\ y \geq 0\}$ und die optimale Lösung würde nicht durch einen ganzzahligen Vektor angenommen, ein Widerspruch zur Definition eines TDI-Systems.

Die Notwendigkeit der Bedingung (2) folgt direkt aus der Definition eines TDI-Systems. Man beachte, dass sich, falls $c := \sum_{i \in S} A_{i\cdot}$ nicht ganzzahlig ist, ein geeignetes Skalar λ finden lässt, so dass $A^T y = c$ genau dann eine ganzzahlige Lösung hat, wenn $\lambda A^T y = \lambda c$ eine ganzzahlige Lösung hat.

Es bleibt zu zeigen, dass (1) und (2) zusammen hinreichend sind, um die TDI-Eigenschaft zu garantieren. Wir nehmen an, dass die Zeilen von A eine Hilbert-Basis bilden und dass für jede Menge $S \subseteq \{1, \ldots, m\}$ das lineare Optimierungsproblem

$$\min \left\{ b^T y : A^T y = \sum_{i \in S} A_{i\cdot},\ y \geq 0 \right\}$$

eine ganzzahlige optimale Lösung hat. Wir müssen zeigen, dass für jedes ganzzahlige $c \in \mathbb{Z}^n$, für das der Wert $\gamma = \min\{b^T y : A^T y = c,\ y \geq 0\}$ existiert, dieser von einer ganzzahligen Lösung angenommen wird. Da die Zeilen von A eine Hilbert-Basis bilden, wissen wir, dass es einen ganzzahligen Vektor y gibt, so dass $A^T y = c$, $y \geq 0$ ist. Sei y^* eine ganzzahlige Lösung von $A^T y^* = c$, $y^* \geq 0$, wobei $b^T y^*$ so klein als möglich ist. Definiere S als den Support von y^*, d. h. $S = \{i \in \{1, \ldots, m\} : y_i^* > 0\}$.

Wir zeigen als Zwischenschritt, dass $\sum_{i \in S} b_i$ der optimale Wert des Hilfsproblems

$$\min \left\{ b^T y : A^T y = \sum_{i \in S} A_{i\cdot},\ y \geq 0 \right\} \qquad (*)$$

ist. Nach unserer Annahme gibt es eine ganzzahlige optimale Lösung u^* von $(*)$. Ist

$$(u^*)^T b < \sum_{i \in S} b_i = (\sum_{i \in S} e^i)^T b\,,$$

dann ist v definiert durch

$$v_i := \begin{cases} u_i^* & \text{für alle } i \notin S \\ u_i^* - 1 & \text{für alle } i \in S \text{ ganzzahlig} \end{cases}$$

und erfüllt $v \geq -1$. Damit ist $y^* + v$ größer oder gleich 0 und ganzzahlig, ebenso

$$A^T(y^* + v) = A^T(y^*) + A^T u^* - A^T \sum_{i \in S} e^i = c$$

und $b^T(y^* + v) < b^T(y^*)$, was ein Widerspruch zur Wahl von y^* ist.

Wir folgern, dass $\sum_{i \in S} b_i$ das Optimum des Hilfsproblems $(*)$ ist und dass $\sum_{i \in S} e^i$ eine optimale Lösung dieses Problems ist, die den Zielfunktionswert $\sum_{i \in S} b_i$ annimmt. Aus dem Satz vom komplementären Schlupf folgt die Existenz eines Vektors $x^* \in P$, so dass $A_{i\cdot}^T x^* = b_i$ für alle $i \in S$, $A_{i\cdot}^T x^* \leq b_i$ für alle $i \notin S$ ist und dass x^* die Zielfunktion $\sum_{i \in S} A_{i\cdot}$ maximiert. Damit haben wir ein Paar x^*, y^* von primal, dual zulässigen Lösungen mit

$$c^T x^* = (y^*)^T A x^* = \sum_{i \in S} y_i^* A_{i\cdot}^T x^* = \sum_{i \in S} y_i^* b_i = b^T y^*\,.$$

Es folgt, dass der ganzzahlige Vektor y^* das Problem $\min\{b^T y : A^T y = c,\ y \geq 0\}$ löst. □

Der folgende Satz zeigt, dass es zu jedem rationalen Polyeder P ein Ungleichungssystem mit der TDI-Eigenschaft gibt. Man erhält es, indem man zu jeder minimalen Seitenfläche F von P eine Hilbert-Basis des Kegels berechnet, der durch die F induzierenden Ungleichungen aufgespannt wird.

Theorem 9.29. *Jedes rationale Polyeder P kann durch ein TDI-System der Form $Ax \leq b$, wobei A ganzzahlig ist, beschrieben werden.*

Beweis. Sei $P = P(D,d)$ mit $D \in \mathbb{Q}^{m\times n}$, $d \in \mathbb{Q}^m$ und $Dx \leq d$ nicht redundant. Desweiteren seien $F_1, \ldots, F_t$ alle (minimalen) Seitenflächen von P. Für $i \in \{1,\ldots,t\}$ sei $\mathcal{H}_i \subseteq \mathbb{Z}^n$ eine minimale ganzzahlige Hilbert-Basis des Kegels $C(D^T_{\mathrm{eq}(F_i)})$.

Wir definieren A als eine Matrix, die als Zeilen gerade die Vektoren in $\bigcup_{i=1}^t \mathcal{H}_i$ besitzt. Betrachte den k-ten Zeilenvektor $A_{k\cdot}$ von A. Dann ist $A^T_{k\cdot}$ ein Element einer ganzzahligen Hilbert-Basis, wir nennen diese $\mathcal{H}_i$. Die Matrix A ist offensichtlich ganzzahlig. Wir definieren desweiteren den Vektor b komponentenweise durch

$$b_k := \max\{\, A_{k\cdot}x : x \in P \,\} = A_{k\cdot}\hat{x} \text{ für alle } \hat{x} \in F_i\,. \quad (9.1)$$

Dieses Maximum existiert und hat die angegebene Form, da wegen $A_{k\cdot} \in \mathcal{H}_i$ auch $A_{k\cdot} \in C(D^T_{\mathrm{eq}(F_i)})$ gilt.

Wir zeigen nun, dass P durch das System $Ax \leq b$ beschrieben wird, i. e.

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

Die Richtung "$\subseteq$" folgt daraus, dass für jedes $y \in P$ und jeden Zeilenindex k der Matrix A gilt

$$A^T_{k\cdot}y \leq \max\left\{\, A^T_{k\cdot}x : x \in P \,\right\} = b_k\,.$$

Zur Richtung "$\supseteq$": Ist $y \notin P$, dann gibt es einen Zeilenindex $l \in \{1,\ldots,m\}$, so dass $D_{l\cdot}y > d_l$ ist. Sei $i \in \{1,\ldots,t\}$, so dass $l \in \mathrm{eq}(F_i)$. Der Index i ist wohldefiniert, da D nicht redundant ist.

Sei $\delta \geq 0$ ein Skalar, so dass $\delta D_{l\cdot} \in \mathbb{Z}^n$. Es gibt nichtnegative ganzzahlige Vielfache $\delta_1, \ldots, \delta_s$ für die Elemente in $\{a^1, \ldots, a^s\} = \mathcal{H}_i$, so dass

$$\delta D_{l\cdot} = \sum_{w=1}^{s} \delta_w a^w\,.$$

Unter Berücksichtigung von (**??**) erhalten wir für $\hat{x} \in F_i$

$$\sum_{w=1}^{s} \delta_w (a^w)^T y = \delta D_{l\cdot}y \quad >$$

$$\delta d_l = \delta D_{l\cdot}\hat{x} = \sum_{w=1}^{s} \delta_w (a^w)^T\hat{x} = \sum_{w=1}^{s} \delta_w b_w\,,$$

wobei b_w diejenige Komponente der rechten Seite b ist, die zur Zeile a^w von A gehört. Also gibt es ein $\hat{w} \in \{1,\ldots,s\}$,

so dass $(a^{\hat{w}})^T y > b_{\hat{w}}$. Das vollendet den Beweis, dass $P = \{x \in \mathbb{R}^n : Ax \leq b\}$ ist.

Das System $Ax \leq b$ ist TDI, denn ist F_i eine Seitenfläche von P und $\{a^1, \ldots, a^s\} = \mathcal{H}_i$, so ist $(a^w)^T x = b_w$ für alle $x \in F_i$ und $w \in \{1, \ldots, s\}$, siehe (**??**). Zusätzlich ist $\{a^1, \ldots, a^s\}$ eine ganzzahlige Hilbert-Basis von $C(A^T_{\text{eq}(F_i)}) = C(D^T_{\text{eq}(F_i)})$. Unter Anwendung von Satz **??** folgt die Behauptung. □

In der Tat gilt sogar, dass P genau dann ganzzahlig ist, wenn b ganzzahlig gewählt werden kann. Die eine Richtung erhält man aus dem Beweis von Satz **??**, denn P ist genau dann ganzzahlig, wenn jede minimale Seitenfläche einen ganzzahligen Punkt enthält. Konstruieren wir unser TDI-System wie im Beweis von Satz **??**, so erfüllt jeder ganzzahlige Punkt z einer Seitenfläche einige der Ungleichungen von $Ax \leq b$ mit Gleichheit. Aufgrund von (**??**) müssen die zugehörigen Komponenten von b ganzzahlig sein. Umgekehrt impliziert Satz **??** im Falle der Ganzzahligkeit von b die Ganzzahligkeit von P. Damit gilt

Korollar 9.30. *Ein rationales Polyeder P ist genau dann ganzzahlig, wenn es ein TDI-System der Form $Ax \leq b$ (mit A und b ganzzahlig) gibt, das P beschreibt.*

Anwendungsbeispiele

Es gibt viele Beispiele für TDI-Systeme, von denen hier exemplarisch zwei genannt seien.

balanciert

Eine 0/1 Matrix A heißt *balanciert*, falls sie keine ungerade quadratische Untermatrix mit zwei Einsen pro Zeile und Spalte enthält. [**?**] haben gezeigt, dass das System $Ax \leq 1$, $x \geq 0$ die TDI-Eigenschaft besitzt, falls A balanciert ist.

Häufig werden TDI Systeme auch dazu verwendet, diskrete Min-Max Resultate zu beweisen, vgl. das Max-Flow-Min-Cut Theorem **??**. Dazu betrachten wir folgendes Beispiel:

r-Arboreszenz

Sei $G = (V, E)$ ein gerichteter Graph. Sei $|E| = m$ und wähle einen bestimmten Knoten $r \in V$. Eine *r-Arboreszenz* ist eine Teilmenge E' von E mit $|E'| = |V| - 1$ Bögen, so dass es für jeden Knoten $v \neq r$ genau einen Bogen in E' gibt, dessen Endknoten v ist. In Formeln,

$$|\delta^-(v) \cap E'|| = 1 \text{ für alle } v \in V \setminus \{ r \} \ .$$

nitt

Ein *r-Schnitt* ist eine Teilmenge der Bögen der Form $\delta^-(U)$ mit $\emptyset \neq U \subset V \setminus \{r\}$. Sei C die Menge aller r-Schnitte von G und bezeichne A die $\pm 1/0$ Matrix, deren Zeilen die Inzidenzvektoren aller r-Schnitte in C repräsentieren. [**?**] haben gezeigt, dass das System

$$\{\, x \in \mathbb{R}^m : Ax \geq 1,\, x \geq 0 \,\} \tag{9.2}$$

TDI ist.

Mit diesem Resultat kann man einfach Fulkersons Satz von der optimalen Arboreszenz [**?**] beweisen. Sei $c \in \mathbb{N}_0^m$ ein nicht-negativer ganzzahliger Vektor. Aufgrund von (**??**) und der LP-Dualität erhalten wir, dass

$$\min\{\, c^T x : Ax \geq 1,\, x \geq 0 \,\} = \max\{\, 1^T y : A^T y \leq c,\, y \geq 0 \,\}$$

und beide Optima von ganzzahligen Punkten x^* und y^* angenommen werden, siehe Satz **??**. Das heißt, die minimale (bzgl. c) r-Arboreszenz ist gleich der maximalen Anzahl von r-Schnitten, wobei kein Bogen $uv \in E$ in mehr als c_{uv} r-Schnitten enthalten ist.

10

Schranken

Betrachten wir ein allgemeines gemischt-ganzzahliges Programm

$$\min c^T x \tag{10.1}$$
$$Ax \leq b \tag{10.2}$$
$$x \in \mathbb{Z}^{n-p} \times \mathbb{R}^p \,, \tag{10.3}$$

wobei $A \in \mathbb{R}^{m \times n}, c \in \mathbb{R}^n, b \in \mathbb{R}^m$ und $p = \{0, \ldots, n\}$.

Im vorigen Kapitel haben wir untersucht, unter welchen Bedingungen wir ganzzahlige Lösungen bekommen, ohne dass wir dies explizit fordern müssen und $x \in \mathbb{Z}^{n-p}$ durch $x \in \mathbb{R}^{n-p}$ ersetzen können. In diesem Kapitel behandeln wir Methoden, die versuchen, mit dem allgemeinen (NP-schweren) Fall umzugehen. Die Grundidee aller Methoden ist, einen Teil, der das Problem schwierig macht, los zu werden. Sie unterscheiden sich darin, welche Schwierigkeit getrennt behandelt wird und in der Art und Weise, wie dieser herausgelöste Teil in das Problem zurückgeführt wird. In Kapitel **??** betrachten wir LP-Relaxierungen. Hier vernachlässigen wir zunächst die Ganzzahligkeitsbedingungen und versuchen sie später durch Hinzufügen von Schnittebenen zu erfüllen. In Kapitel **??** werden Lagrange-Relaxierungen behandelt. Hier löschen wir einen Teil der Nebenbedingungen und transferieren sie stattdessen durch Einführung eines Strafparameters in die Zielfunktion. In Kapitel **??** diskutieren wir Dekompositionsmethoden, u.a. Dantzig-Wolfe und Benders' Dekomposition. Diese Methoden löschen ebenfalls einen Teil der Nebenbedingungsmatrix, reformulieren den gelöschten Teil und fügen den reformulierten Teil wieder in das Gesamtproblem ein.

10.1 LP-Relaxierungen

Relaxieren wir die Ganzzahligkeitsbedingungen in (**??**) so erhalten wir die sog. *LP-Relaxierung* von (**??**):

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

Zur Lösung linearer Programme sind polynomiale und effiziente Methoden bekannt, wie sie ausführlich in Teil 1 dieses Buches diskutiert werden. Falls die optimale Lösung x^* der LP-Relaxierung ganzzahlig ist, haben wir (**??**) gelöst. Andernfalls muss es eine Ungleichung geben (auch *Schnittebene* genannt) die x^* von $P_{I,p} = \text{conv}(\{x \in \mathbb{Z}^{n-p} \times \mathbb{R}^p \mid Ax \leq b\})$ trennt. Das Problem, eine solche Ungleichung zu finden, nennt man das *Separierungsproblem*, siehe (TODO Problem 6.2, DO I). Finden wir eine solche Ungleichung, so verbessern wir die LP-Relaxierung, indem wir die Ungleichung dem LP hinzufügen und fahren fort. Dieser Gesamtprozess wird auch als *Schnittebenenverfahren* bezeichnet. Da wir wissen, dass SEP und OPT äquivalent sind, vgl. TODO DO I, können wir nicht erwarten, auf diese Weise immer eine Optimallösung zu finden. Falls das nicht gelingt, d. h. wenn die optimale Lösung des LP gebrochen ist und wir keine weiteren Ungleichungen finden, betten wir das Verfahren in ein Enumerationsschema ein, siehe dazu Kapitel **??**. Wir hoffen aber, durch das Schnittebenenverfahren die Ausgangsformulierung so zu verbessern, dass der anschließende Enumerationsaufwand gering bleibt.

Referenz auf nicht vorhandenen Satz

Referenz auf nicht vorhandenen Satz

Der Schlüssel zum Erfolg dieser Methode ist also, gute Ungleichungen für das zugrundeliegende Polyeder zu kennen bzw. zu finden. Wir diskutieren zunächst Wege, Ungleichungen zu generieren, die unabhängig von der zugrundeliegenden Problemstruktur sind und schauen dann auf MIPs mit spezieller lokaler Struktur.

10.1.1 Allgemeine Schnittebenen

In diesem Kapitel behandeln wir eine Klasse von Ungleichungen, die gültig für P_I ist und die unabhängig von jeder Problemstruktur angewandt werden kann ([**?**], [**?**]). Wir werden sehen, dass diese Klasse das Potential hat, P_I vollständig zu beschreiben. Es gibt zwei Zugänge für diese

Ungleichungen, einen geometrischen und einen algorithmischen. Wir beginnen mit dem geometrischen Ansatz und beschränken uns zunächst auf rein ganzzahlige Probleme.

Chvátals geometrischer Zugang

Betrachten wir ein rationales Polyeder

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

mit $A \in \mathbb{Q}^{m \times n}$, $b \in \mathbb{Q}^m$. Wir suchen eine lineare Beschreibung von

$$P_I := \operatorname{conv}\{ x \in \mathbb{Z}^n : Ax \leq b \} \ .$$

Nach Beobachtung **??** ist $P = P_I$ genau dann, wenn jede (minimale) Seitenfläche von P ganzzahlige Punkte enthält. Dies wiederum ist äquivalent dazu, dass jede Stützhyperebene ganzzahlige Punkte enthält.

Lemma 10.1. *Sei P ein Polyeder. Dann enthält jede minimale Seitenfläche von P ganzzahlige Punkte genau dann, wenn jede Stützhyperebene ganzzahlige Punkte enthält.*

Beweis. Übung.

Die Idee der Methode, die wir nun diskutieren, ist, sich jede Stützhyperebene von P anzuschauen und sie solange näher an P_I zu schieben, bis sie einen ganzzahligen Punkt enthält.

Sei $\{x \in \mathbb{R}^n : c^T x = \delta\}$ eine Stützhyperebene von P mit $P \subseteq \{x \in \mathbb{R}^n : c^T x \leq \delta\}$ und c ganzzahlig. Offensichtlich gilt

$$P_I \subseteq \left\{ x \in \mathbb{R}^n : c^T x \leq \lfloor \delta \rfloor \right\} \ .$$

Diese Beobachtung legt nahe, alle Stützhyperebenen mit ganzzahliger linker Seite zu nehmen und die rechte Seite zu runden, um eine schärfere Formulierung zu erhalten. Definiere dazu

$$P^1 := \Big\{ x \in \mathbb{R}^n : c^T x \leq \lfloor \delta \rfloor \text{ für alle Stützhyperebenen } \{x \in \mathbb{R}^n : c^T x = \delta\} \text{ von } P \text{ mit } c \text{ ganzzahlig} \Big\} \ . \quad (10.4)$$

Auf den ersten Blick ist nicht klar, ob P^1 wieder ein Polyeder ist, weil es unendlich viele Stützhyperebenen gibt. Wir werden jedoch beweisen, dass P^1 tatsächlich wieder

ein Polyeder ist. Damit können wir das Verfahren fortsetzen und dasselbe Spiel mit P^1 machen. Definiere

$$P^0 := P \text{ und } P^{t+1} := (P^t)^1 \,. \tag{10.5}$$

Die folgenden Beziehungen sind offensichtlich.

$$P = P^0 \supseteq P^1 \supseteq \ldots \supseteq P_I \,. \tag{10.6}$$

Damit stellt sich die Frage, ob dieses Verfahren endlich ist. In der Tat, es ist es, und wir werden zeigen, dass $P^t = P_I$ für ein $t \in \mathbb{N}$. Das heißt, dass P^t unser Ziel liefert, nämlich eine Beschreibung P_I in Form von linearen Ungleichungen.

Wir beginnen mit dem Beweis, dass P^1 ein Polyeder ist. Wir erinnern uns aus Satz **??**, dass jedes rationale Polyeder durch ein TDI System der Form $Ax \leq b$ mit A ganzzahlig beschrieben werden kann. Und in der Tat, runden wir die rechte Seite dieses Systems, so erhalten wir P^1.

Theorem 10.2. *Sei $P = \{x \in \mathbb{R}^n : Ax \leq b\}$ mit $Ax \leq b$ TDI und A ganzzahlig. Dann gilt $P^1 = \{x \in \mathbb{R}^n : Ax \leq \lfloor b \rfloor\}$, d. h. P^1 ist ein Polyeder.*

Beweis. Im Falle $P = \emptyset$ ist nichts zu beweisen. Sei $P \neq \emptyset$. Offensichtlich gilt $P^1 \subseteq \{x \in \mathbb{R}^n : Ax \leq \lfloor b \rfloor\}$ (man wähle als Stützhyperebene $\{x \in \mathbb{R}^n : A_{i,}.x \leq b_i\}$).

Um die Umkehrung zu beweisen, sei $\{x \in \mathbb{R}^n : c^T x = \delta\}$ eine Stützhyperebene von P mit $P \subseteq \{x \in \mathbb{R}^n : c^T x \leq \delta\}$ und c ganzzahlig, $\delta \in \mathbb{Q}$. Aus dem Dualitätssatz der Linearen Optimierung wissen wir

$$\delta = \max \left\{ c^T x : Ax \leq b,\, x \in \mathbb{R}^n \right\} = \min \left\{ y^T b : A^T y = c,\, y \geq 0 \right\} \,.$$

Da $Ax \leq b$ TDI ist, gibt es eine ganzzahlige optimale Lösung y^* von $\min\{y^T b : A^T y = c,\, y \geq 0\}$. Da x die Ungleichung $Ax \leq \lfloor b \rfloor$ erfüllt, gilt

$$c^T x = (A^T y^*)^T x = (y^*)^T (Ax) \quad \leq (y^*)^T \lfloor b \rfloor = \lfloor (y^*)^T \lfloor b \rfloor \rfloor \leq \lfloor (y^*)^T b \rfloor = \lfloor \delta \rfloor \,.$$

Es folgt

$$\left\{ x \in \mathbb{R}^n : Ax \leq \lfloor b \rfloor \right\} \subseteq \left\{ x \in \mathbb{R}^n : c^T x \leq \lfloor \delta \rfloor \right\} \,.$$

Also gilt

$$P^1 = \bigcap_{c,\delta} \{\, x \in \mathbb{R}^n : c^T x \leq \lfloor \delta \rfloor \,\} \supseteq \{\, x \in \mathbb{R}^n : Ax \leq \lfloor b \rfloor \,\} \;,$$

wobei wir die Schnittmenge über allen Stützhyperebenen $\{x \in \mathbb{R}^n : c^T x = \delta\}$ mit ganzzahligem c und $P \subseteq \{x \in \mathbb{R}^n : c^T x \leq \delta\}$ nehmen. □

Korollar 10.3. *Sei F eine Seitenfläche von P. Dann gilt*

$$F^1 = P^1 \cap F \,.$$

Beweis. Sei $P = \{x \in \mathbb{R}^n : Ax \leq b\}$ mit $Ax \leq b$ TDI und A ganzzahlig. Sei desweiteren $F = \{x \in P : c^T x = \delta\}$ eine Seitenfläche von P mit c, δ ganzzahlig und $P \subseteq \{x \in \mathbb{R}^n : c^T x \leq \delta\}$. Satz **??** impliziert, dass das System $Ax \leq b$, $c^T x \leq \delta$ ebenfalls TDI ist. Also ist nach Korollar **??** auch das System $Ax \leq b$, $c^T x = \delta$ TDI. Wir erhalten mit Satz **??**

$$\begin{aligned} P^1 \cap F &= P^1 \cap P \cap \{\, x \in \mathbb{R}^n : c^T x = \delta \,\} \\ &= \{\, x \in \mathbb{R}^n : Ax \leq \lfloor b \rfloor,\, c^T x = \delta \,\} \\ &= \{\, x \in \mathbb{R}^n : Ax \leq \lfloor b \rfloor,\, c^T x \leq \lfloor \delta \rfloor,\, -c^T x \leq \lfloor -\delta \rfloor \,\} \\ &= F^1 \,. \end{aligned}$$

□

Anmerkung 10.4.

(a) Die Seite F^1 ist eine (möglicherweise leere) Seitenfläche von P^1, da

$$F^1 = P^1 \cap F = P^1 \cap P \cap \{\, x \in \mathbb{R}^n : c^T x = \delta \,\} = P^1 \cap \{\, x \in \mathbb{R}^n : c^T x = \delta \,\} \;.$$

(beachte, dass $P^1 \subseteq P \subseteq \{x \in \mathbb{R}^n : c^T x \leq \delta\}$)

(b) Die wiederholte Anwendung von Korollar **??** liefert

$$F^t = P^t \cap F, \text{ für } t = 1, 2, \ldots$$

Nun haben wir alles zusammen, um die Endlichkeit der Rundeverfahrens zu zeigen.

Theorem 10.5. *Sei P ein rationales Polyeder. Dann gibt es eine natürliche Zahl $t \in \mathbb{N}$, so dass $P^t = P_I$.*

Beweis. Ist $P = \emptyset$, dann folgt $P^0 = P = P_I$. Wir können also $P \neq \emptyset$ annehmen. Wir beweisen die Behauptung mittels Induktion nach der Dimension d von P.

Ist $d = 0$, dann gilt $P = \{\bar{x}\}$ für ein $\bar{x} \in \mathbb{R}^n$. Ist $\bar{x} \in \mathbb{Z}^n$, so erhalten wir $P^0 = P = P_I$. Im Fall $\bar{x} \notin \mathbb{Z}^n$ ist $P^1 = \emptyset = P_I$. Es sei $d > 0$ und die Behauptung sei wahr für alle Polyeder von kleinerer Dimension. Wir zeigen zunächst, dass wir o.B.d.A. annehmen können, dass P volldimensional ist.

Sei $\{x \in \mathbb{R}^n : Ax = b\}$ die affine Hülle von P, d. h. $P \subseteq \{x \in \mathbb{R}^n : Ax = b\}$. O.B.d.A. ist A eine ganzzahlige Matrix mit vollem Zeilenrang, d. h. der Zeilenrang von A ist $n - d$.

Falls $\{x \in \mathbb{R}^n : Ax = b\}$ keine ganzzahlige Lösung hat, dann gibt es nach Satz **??** ein $y \in \mathbb{Q}^{n-d}$ mit $c := A^T y \in \mathbb{Z}^n$ und $\delta := b^T y \notin \mathbb{Z}$. Jedes $x \in P$ erfüllt $Ax = b$ und damit gilt $c^T x = (A^T y)^T x = y^T Ax = y^T b = \delta$. Also ist $\{x \in \mathbb{R}^n : c^T x = \delta\}$ eine Stützhyperebene von P. Wir folgern

$$P^1 \subseteq \left\{ x \in \mathbb{R}^n : c^T x \leq \lfloor \delta \rfloor, c^T x \geq \lceil \delta \rceil \right\} = \emptyset$$

woraus wiederum $P_I \subseteq P^1 = \emptyset$ folgt.

Nun sei $\hat{x}$ eine ganzzahlige Lösung von $\{x \in \mathbb{R}^n : Ax = b\}$. Satz **??** ist invariant unter Verschiebung um den Vektor $\hat{x}$ und somit können wir $\text{aff}(P) = \{ x \in \mathbb{R}^n : Ax = 0 \}$ annehmen. Aus Satz **??** und Bemerkung **??** (a) sowie der Tatsache, dass der Zeilenrang von A gleich $n - d$ ist, folgt, dass es eine reguläre unimodulare Matrix U gibt, so dass $AU = [B\ 0]$ gilt, wobei $B \in \mathbb{Z}^{(n-d)\times(n-d)}$ regulär ist. Da U regulär und unimodular ist, ist auch U^{-1} unimodular und die Variablentransformation $x = Uz$ verletzt nicht die Gültigkeit des Satzes, insbesondere gilt $x \in \mathbb{Z}^n \Leftrightarrow z \in \mathbb{Z}^n$. Wir können annehmen, dass

$$\text{aff}(P) = \{ z \in \mathbb{R}^n : [B\ 0]z = 0 \} = \{ 0 \}^{n-d} \times \mathbb{R}^d$$

gilt. Jede Stützhyperebene $H = \{x \in \mathbb{R}^n : c^T x = \delta\}$ von P wird in die Standardform

$$H' = \left\{ x \in \mathbb{R}^n : \sum_{i=1}^{n-d} 0x_i + \sum_{i=n-d+1}^{n} c^T x_i = \delta \right\}$$

überführt, indem geeignete Vielfache der Zeilen von $[B\ 0]$ zu c addiert werden. Bei der Konstruktion von P^1 können

wir uns auf die Stützhyperebenen der Form H' beschränken. Wir können also $n - d = 0$ annehmen, d. h. P ist volldimensional.

Wir behaupten, es gibt eine ganzzahlige Matrix W, einen rationalen Vektor w und einen ganzzahligen Vektor w', so dass $P = \{x \in \mathbb{R}^n : Wx \leq w\}$ und $P_I = \{x \in \mathbb{R}^n : Wx \leq w'\}$ ist. Um dies nachzuvollziehen, sei $P = \{x \in \mathbb{R}^n : Ax \leq b\}$ und $P_I = \{x \in \mathbb{R}^n : Cx \leq b'\}$ mit ganzzahligen Matrizen A und C. Definiere

$$W := \begin{bmatrix} A \\ C \end{bmatrix}.$$

Für jeden Zeilenindex i von W liefert die Wahl

$$\begin{aligned} w_i &:= \max\{\, W_i.x : x \in P \,\} \text{ und} \\ w'_i &:= \max\{\, W_i.x : x \in P_I \,\} \end{aligned}$$

das Gewünschte. Beachte, dass wir annehmen können, dass P bereits zu Beginn in dieser Form gegeben ist, da in der Konstruktion von P^1 *alle* Stützhyperebenen von P betrachtet werden, insbesondere alle Zeilen der Matrix W.

Betrachte eine Ungleichung $a^T x \leq \beta'$ des Systems $Wx \leq w'$. Wir behaupten, dass es eine natürliche Zahl $s \in \mathbb{N}$ gibt mit $P^s \subseteq \{x \in \mathbb{R}^n : a^T x \leq \beta'\}$.

Nehmen wir das Gegenteil an. Sei $a^T x \leq \beta$ die zugehörige Ungleichung von $Wx \leq w$. Wegen

$$P^1 \subseteq \left\{\, x \in \mathbb{R}^n : a^T x \leq \lfloor \beta \rfloor \,\right\}$$

gibt es ein $\beta'' \in \mathbb{Z}$ und $r \in \mathbb{N}$, so dass

$$\begin{aligned} &\beta' < \beta'' \leq \lfloor \beta \rfloor \text{ und} \\ &P^u \subseteq \left\{\, x \in \mathbb{R}^n : a^T x \leq \beta'' \,\right\} \\ &P^u \not\subseteq \left\{\, x \in \mathbb{R}^n : a^T x \leq \beta'' - 1 \,\right\} \text{ für alle } u \geq r \end{aligned}$$

gilt. Auf Grund der Wahl von r ist $\{x \in \mathbb{R}^n : a^T x = \beta''\}$ eine Stützhyperebene von P^r (beachte, dass $P \not\subseteq \{x \in \mathbb{R}^n : a^T x \leq \beta''\}$, da P volldimensional) und damit hat

$$F := P^r \cap \left\{\, x \in \mathbb{R}^n : a^T x = \beta'' \,\right\}$$

eine geringere Dimension als n. Weiter folgern wir aus $P_I \subseteq \{a^T x \leq \beta'\}$, dass $F \cap \mathbb{Z}^n = \emptyset$ gilt. Also existiert nach der Induktionsannahme ein $q \in \mathbb{N}$ mit $F^q = \emptyset$. Aus Bemerkung **??** (b) folgt

$$\begin{aligned}\emptyset = F^q &= \left(P^r \cap \left\{\, x \in \mathbb{R}^n : a^T x = \beta'' \,\right\}\right)^q \\ &= P^{r+q} \cap \left\{\, x \in \mathbb{R}^n : a^T x = \beta'' \,\right\} .\end{aligned}$$

Es gilt also $P^{r+q} \subseteq \{x \in \mathbb{R}^n : a^T x < \beta''\}$, was heißt $P^{r+q+1} \subseteq \{x \in \mathbb{R}^n : a^T x \leq \beta'' - 1\}$ und dies ist ein Widerspruch zur Wahl von β'' und r.

Da $a^T x \leq \beta'$ beliebig gewählt war und das System $Wx \leq w'$ endlich ist, haben wir Satz **??** bewiesen. □

Betrachten wir uns noch einmal, was wir bisher gezeigt haben. Die Vorgehensweise, um eine lineare Beschreibung des ganzzahligen Polyeders $P_I = \text{conv}\{x \in \mathbb{Z}^n : Ax \leq b\}$ zu erreichen, ist die folgende. Wir beginnen mit der linearen Relaxierung $P^0 = P = \{x \in \mathbb{R}^n : Ax \leq b\}$ von P_I. Als nächstes betrachten wir jede Stützhyperebene von P, deren linke Seite ganzzahlig ist und runden die rechte Seite nach unten zur nächsten ganzen Zahl ab. Diese Prozedur, für alle solche Stützhyperebenen durchgeführt, ergibt das Polyeder P^1. Satz **??** zeigt, dass keine Notwendigkeit besteht, alle Stützhyperebenen zu testen. Alles, was wir benötigen, ist ein TDI-System $Dx \leq d$, das P beschreibt. Für dieses TDI-System müssen wir den Vektor der rechten Seite d nach unten abrunden. Im Beweis von Satz **??** haben wir das TDI-System für ein rationales Polyeder P explizit konstruiert, indem wir für jede Seitenfläche F von P eine Hilbert-Basis des Kegels $\text{cone}(A^T_{\text{eq}(F)})$ erzeugen. Letztendlich erhalten wir mit einem Verfahren zur Konstruktion einer Hilbert-Basis für einen polyedrischen Kegel insgesamt einen Algorithmus zur Berechnung von P^1. Nach Satz **??** müssen wir diesen Algorithmus nur eine endliche Zahl von Schritten durchführen, um eine lineare Beschreibung von P_I zu erzielen.

Dennoch ist die gesamte Prozedur kaum praktikabel. Zunächst ist die Zahl $t \in \mathbb{N}$ in Satz **??** möglicherweise exponentiell in der Kodierungslänge der Eingabegrößen A, b (Übung). Zweitens müssen wir in jeder Iteration i Hilbert-Basen für alle Kegel bestimmen, die von den Seitenflächen von P^{i-1} erzeugt werden. Im Allgemeinen ist nicht nur die Zahl der Seitenflächen exponentiell, wir können auch die Hilbert-Basen nicht in polynomialer Zeit berechnen.

Andererseits ist zum Lösen von (**??**) eine vollständige Beschreibung von $P_I = \text{conv}\{x \in \mathbb{Z}^n : Ax \leq b\}$ nicht notwendig. Alles, was wir benötigen, ist das Finden einer optimalen Lösung. Anders ausgedrückt, wir sind nur an einer Seitenfläche interessiert, nämlich an der, die

die optimalen Lösungen enthält. Ja sogar für diese Seitenfläche ist es nicht unbedingt notwendig, eine Hilbert-Basis für den zugehörigen Kegel zu bestimmen. Wir stellen uns also die Frage, ob es möglich ist, ganzzahlige Vektoren in diesem Kegel nach Bedarf zu bestimmen. Wir nehmen an, wir beginnen mit der Lösung der LP-Relaxierung $\max\{c^T x : Ax \leq b\}$. Sei x^* eine optimale Lösung, die nicht ganzzahlig ist. Die zentrale Frage besteht darin, einen ganzzahligen Vektor in $C(A^T_{\text{eq}(x^*)})$ zu finden, der die momentane optimale Lösung abschneidet, d. h. finde $d \in \mathbb{Z}^n \cap C(A^T_{\text{eq}(x^*)})$ mit $d^T x^* \notin \mathbb{Z}$.

Gomorys algorithmischer Ansatz

Gomory schlug in [?] einen systematischen Weg vor, um einen solchen Vektor d durch Auswertung der Information aus dem Simplexalgorithmus zu erhalten. Wir nehmen an, A und b seien ganzzahlig. Wir nehmen weiterhin an, dass (**??**) in Standardform

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

mit $P = \{x \in \mathbb{R}^n : Ax = b,\, x \geq 0\}$ und $P_I = \text{conv}\{x \in \mathbb{Z}^n : Ax = b,\, x \geq 0\}$ ist. Dies kann durch Aufspaltung von x in x^+ und x^- mit $x = x^+ - x^-$, $x^+, x^- \geq 0$ erreicht werden, wenn die Nichtnegativitäts-Bedingungen nicht in das System $Ax \leq b$ aufgenommen werden, wobei Schlupfvariablen für die kleiner oder gleich Bedingungen eingeführt werden. Beachte, dass für die Schlupfvariablen ebenfalls Ganzzahligkeit gefordert werden kann, da alle Werte ganzzahlig sind. Lösen wir die LP-Relaxierung

$$\begin{aligned} \max\, & c^T x \\ & Ax = b \\ & x \geq 0 \end{aligned}$$

mittels der Simplexmethode, so erhalten wir eine optimale Lösung x^* und eine optimale Basis B mit $B \subseteq \{1, \ldots, n\}$, $|B| = m$ und A_B regulär. Über die Basis B drücken sich die Werte von x^* aus als

$$\begin{aligned} x^*_B &= A_B^{-1} b - A_B^{-1} A_N x^*_N \\ x^*_N &= 0\,. \end{aligned} \tag{10.7}$$

Ist x^* ganzzahlig, haben wir eine optimale Lösung für (**??**). Im anderen Fall ist einer der Werte x_B^* gebrochen. Sei $i \in B$ ein Index mit $x_i^* \notin \mathbb{Z}$. Da jede zulässige ganzzahlige Lösung (**??**) erfüllt, gilt

$$A_{i\cdot}^{-1}b - \sum_{j\in N} A_{i\cdot}^{-1}A_{\cdot j}x_j \in \mathbb{Z} \tag{10.8}$$

für alle ganzzahligen Lösungen x. (**??**) gilt auch, wenn wir ganzzahlige Vielfache von x_j, $j \in N$, oder eine ganze Zahl zu $\bar{A}_{i,\cdot}^{-1}b$ addieren. Wir erhalten

$$f(A_{i\cdot}^{-1}b) - \sum_{j\in N} f(A_{i\cdot}^{-1}A_{\cdot j})x_j \in \mathbb{Z} \tag{10.9}$$

für alle ganzzahligen Lösungen x, wobei $f(\alpha) = \alpha - \lfloor\alpha\rfloor$ für $\alpha \in \mathbb{R}$ ist. Da $0 \le f(\cdot) < 1$ und $x \ge 0$ ist, folgt

$$f(A_{i\cdot}^{-1}b) - \sum_{j\in N} f(A_{i\cdot}^{-1}A_{\cdot j})x_j \le 0\,.$$

Also ist die Ungleichung

$$\sum_{j\in N} f(A_{i\cdot}^{-1}A_{\cdot j})x_j \ge f(A_{i\cdot}^{-1}b) \tag{10.10}$$

gültig für alle zulässigen ganzzahligen Lösungen x. Weiterhin ist sie von der momentanen LP-Lösung x^* verletzt, da $x_N^* = 0$ und $f(A_{i\cdot}^{-1}b) = f(x_i^*) > 0$ ist.

Es stellt sich heraus, dass (**??**) in der Tat eine Stützhyperebene von P mit ganzzahliger linker Seite ist. Um dies zu sehen, subtrahieren wir

$$\begin{aligned} A_{i\cdot}^{-1}Ax &= A_{i\cdot}^{-1}b \\ \Leftrightarrow x_i + \sum_{j\in N} A_{i\cdot}^{-1}A_{\cdot j}x_j &= A_{i\cdot}^{-1}b \end{aligned}$$

von (**??**). Wir erhalten

$$x_i + \sum_{j\in N} \lfloor A_{i\cdot}^{-1}A_{\cdot j}\rfloor x_j \le \lfloor A_{i\cdot}^{-1}b\rfloor$$

und damit, wenn die rechte Seite nicht gerundet ist, eine Stützhyperebene von P mit ganzzahliger linker Seite.

Fügt man überdies diese Ungleichung zu dem System $Ax = b$, $x \ge 0$ hinzu, so erhält sich die Eigenschaft, dass

alle Werte ganzzahlig sind. Also kann die Schlupfvariable, die für die neue Ungleichung eingeführt werden muss, ebenso als ganzzahlig angesehen werden und so das Verfahren iteriert werden. Gomory zeigt in [**?**], dass mit einer bestimmten Wahl der erzeugenden Zeile für die Schnitte ein endlicher Algorithmus vorliegt, d. h. nachdem eine endliche Zahl an Ungleichungen hinzugefügt wurde, wird eine ganzzahlige Lösung gefunden. Dies stellt einen alternativen Beweis für Satz **??** dar.

Beispiel 10.6. Betrachte das folgende IP

$$\begin{aligned} \max x_2 & \\ 4x_1 + x_2 &\leq 4 \\ -4x_1 + x_2 &\leq 0 \\ x_1, x_2 &\geq 0 \\ x_1, x_2 &\in \mathbb{Z}\,. \end{aligned}$$

Hinzufügen von Schlupfvariablen ergibt

$$\begin{aligned} \max x_2 & \\ 4x_1 + x_2 + x_3 &= 4 \\ -4x_1 + x_2 + x_4 &= 0 \\ x_1, x_2, x_3, x_4 &\geq 0 \\ x_1, x_2, x_3, x_4 &\in \mathbb{Z}\,. \end{aligned}$$

Die optimale Basis für die LP-Relaxierung ist $B = \{1, 2\}$ mit

$$A_B^{-1} = \begin{bmatrix} \frac{1}{8} & -\frac{1}{8} \\ \frac{1}{2} & \frac{1}{2} \end{bmatrix}\,.$$

Also ist $x_B^* = \binom{1/2}{2}$. Die einzige gebrochene Komponente ist x_1 und für $i = 1$ wird (**??**) zu

$$\tfrac{1}{2} - \tfrac{1}{8}x_3 + \tfrac{1}{8}x_4 \in \mathbb{Z}\,,$$

was für alle ganzzahligen Lösungen x gilt. Wir fügen x_4 zur obigen Gleichheit hinzu, um (**??**) zu erhalten:

$$\tfrac{1}{8}x_3 + \tfrac{7}{8}x_4 \geq \tfrac{1}{2}.$$

Um zu zeigen, dass die Ungleichung zu einer Stützhyperebene gehört, subtrahieren wir $x_1 + 0x_2 + \frac{1}{8}x_3 - \frac{1}{8}x_4 = \frac{1}{2}$ und erhalten

$$-x_1 + x_4 \geq 0 \;=\; \lfloor \tfrac{1}{2} \rfloor$$

Drücken wir diese Ungleichung in den Originalvariablen x_1, x_2 aus, erhalten wir durch Ersetzen der Schlupfvariable x_4

$$-3x_1 + x_2 \leq 0.$$

Dies ist eine gültige Ungleichung für

$$P_I = \text{conv}\{x \in \mathbb{Z}^2_+ : 4x_1 + x_2 \leq 4,\ -4x_1 + x_2 \leq 0\},$$

siehe (**??**).

Gomorys gemischt-ganzzahlige Schnitte

Wir wollen uns schließlich noch mit dem Fall beschäftigen, dass wir kein rein ganzzahliges Programm haben, sondern ein gemischt-ganzzahliges Programm. In diesem Fall funktionieren die Ideen von Gomory und Chvátal nicht. Chvátal's Ansatz funktioniert nicht, da die rechte Seite in (**??**) nicht abgerundet werden kann. Gomory's Ansatz versagt, da es nicht mehr länger möglich ist, ganzzahlige Vielfache auf kontinuierliche Variablen zu addieren, um (**??**) aus (**??**) zu erhalten. So hat beispielsweise $\frac{1}{3} + \frac{1}{3}x_1 - 2x_2 \in \mathbb{Z}$ mit $x_1 \in \mathbb{Z}_+, x_2 \in \mathbb{R}_+$ eine größere Lösungsmenge (z. B. $\binom{1}{1/3}$) als $\frac{1}{3} + \frac{1}{3}x_1 \in \mathbb{Z}$. Wir können also nicht mehr garantieren, dass die Koeffizienten der kontinuierlichen Variablen nichtnegativ sind und damit die Gültigkeit von (**??**) nachweisen. Dennoch kann man mittels des folgenden *disjunktiven Arguments* gültige Ungleichungen erzeugen.

Beobachtung 10.7. *Sei $(a^k)^T x \leq \alpha^k$ eine gültige Ungleichung für ein Polyeder $P^k \subseteq \mathbb{R}^n_+$ für $k = 1, 2$. Dann ist*

$$\sum_{i=1}^{n} \min(a^1_i, a^2_i) x_i \leq \max(\alpha^1, \alpha^2)$$

gültig für $P^1 \cup P^2$ und $\text{conv}(P^1 \cup P^2)$.

Diese Beobachtung kann auf verschiedene Weise gültige Ungleichungen für den gemischt-ganzzahligen Fall erzeugen. Wir erläutern einen dieser Wege, nämlich Gomory's gemischt ganzzahlige Schnitte.

Wir betrachten wieder die Situation in (**??**), wobei $x_i, i \in B$, eine ganze Zahl sein soll. Wir benutzen die folgenden Abkürzungen $\bar{a}_j = A^{-1}_{i\cdot} A_{\cdot j}$, $\bar{b} = A^{-1}_{i\cdot} b, f_j = f(\bar{a}_j), f_0 = f(\bar{b})$, und $N^+ = \{j \in N : \bar{a}_j \geq 0\}$ sowie $N^- = N \setminus N^+$. Der Ausdruck (**??**) ist äquivalent

zu $\sum_{j\in N} \bar{a}_j x_j = f_0 + k$ für ein $k \in \mathbb{Z}$. Wir unterscheiden nun die zwei Fälle $\sum_{j\in N} \bar{a}_j x_j \geq f_0 \geq 0$ und $\sum_{j\in N} \bar{a}_j x_j \leq f_0 - 1 < 0$. Im ersten Fall muss

$$\sum_{j\in N^+} \bar{a}_j x_j \geq f_0$$

gelten. Im zweiten Fall erhalten wir $\sum_{j\in N^-} \bar{a}_j x_j \leq f_0 - 1$, was zu

$$-\frac{f_0}{1-f_0} \sum_{j\in N^-} \bar{a}_j x_j \geq f_0$$

äquivalent ist. Wir wenden nun Beobachtung **??** auf die Zerlegung $P^1 = P \cap \{x : \sum_{j\in N} \bar{a}_j x_j \geq 0\}$ und $P^2 = P \cap \{x : \sum_{j\in N} \bar{a}_j x_j \leq 0\}$ an und erhalten die gültige Ungleichung

$$\sum_{j\in N^+} \bar{a}_j x_j - \frac{f_0}{1-f_0} \sum_{j\in N^-} \bar{a}_j x_j \geq f_0. \tag{10.11}$$

Diese Ungleichung kann auf folgende Weise verschärft werden. Man beachte, dass die Herleitung von (**??**) immer noch gültig ist, wenn wir ganzzahlige Vielfache zu ganzzahligen Variablen addieren. Auf diese Weise können wir jede ganzzahlige Variable entweder in die Menge N^+ oder in die Menge N^- einordnen. Ist eine Variable in N^+, so ist der Koeffizient in (**??**) gleich $\bar{a}_j$ und somit ist der beste mögliche Koeffizient nach der Addition von ganzzahligen Vielfachen $f_j = f(\bar{a}_j)$. In N^- ist der Koeffizient in (**??**) gleich $-\frac{f_0}{1-f_0}\bar{a}_j$ und somit ist $\frac{f_0(1-f_j)}{1-f_0}$ die beste Wahl. Insgesamt erhalten wir den bestmöglichen Koeffizienten durch die Wahl von $\min(f_j, \frac{f_0(1-f_j)}{1-f_0})$. Dies führt auf Gomory's gemischt-ganzzahligen Schnitt

$$\begin{aligned} &\sum_{\substack{j: f_j \leq f_0 \\ x_j \in \mathbb{Z}}} f_j x_j + \sum_{\substack{j: f_j > f_0 \\ x_j \in \mathbb{Z}}} \frac{f_0(1-f_j)}{1-f_0} x_j + \\ &\sum_{\substack{j\in N^+ \\ x_j \notin \mathbb{Z}}} \bar{a}_j x_j - \sum_{\substack{j\in N^- \\ x_j \notin \mathbb{Z}}} \frac{f_0}{1-f_0} \bar{a}_j x_j \geq f_0 \,. \end{aligned} \tag{10.12}$$

Gomory zeigt in [**?**], dass ein Algorithmus, der auf iterativer Hinzufügung dieser Ungleichungen beruht, das MIP $\min\{c^T x : x \in X\}$ mit $X = \{x \in \mathbb{Z}^p_+ \times \mathbb{R}^{n-p}_+ : Ax = b\}$ in einer endlichen Anzahl von Schritten löst, falls $c^T x \in \mathbb{Z}$ für alle $x \in X$ gilt.

Man kennt in der Literatur eine ganze Reihe weitere Ungleichungen, die unabhängig von einer bestimmten Problemstruktur sind. Unter diesen findet man beispielsweise Mixed Integer Rounding Schnitte (MIR) [**?**] und sogenannte Lift-and-Project Schnitte [**?**].

10.1.2 Ungleichungen mit Struktur

Wir haben uns soeben mit gültigen Ungleichungen für allgemeine IP's und MIP's beschäftigt. Richten wir unser Augenmerk auf eine einzelne Nebenbedingung oder eine kleine Teilmenge von solchen Nebenbedingungen, so kann ein allgemeines Problem eine gewisse "lokale" Struktur enthalten. So können beispielsweise alle Variablen in einer Nebenbedingung 0/1-Variablen sein, oder ein kleiner Teil eines MIP kann ein Flussproblem in einem Netzwerk sein. Wir suchen hier nach Wegen, um strengere Ungleichungen unter Ausnutzung solcher lokaler Strukturen zu erhalten.

Rucksack und Cover Ungleichungen

Das Konzept eines Covers wurde in der Literatur häufig benutzt, um gültige Ungleichungen für (gemischt) ganzzahlige Mengen zu erzeugen. In diesem Abschnitt zeigen wir zunächst, wie wir dieses Konzept nutzen können, um Cover Ungleichungen für die 0/1-Rucksackmenge zu erzeugen. Wir überlegen dann, wie wir diese Ungleichungen auf komplexere gemischt ganzzahlige Mengen ausdehnen können.

Definition 10.8. *Wir betrachten das* 0/1*-Rucksackpolytop*

$$P_K(N,a,b) = \left\{ x \in \{0,1\}^N : \sum_{j\in N} a_j x_j \leq b \right\}$$

mit nichtnegativen Koeffizienten, d. h. es ist $a_j \geq 0$ *für* $j \in N$ *und* $b \geq 0$*. Eine Menge* $C \subseteq N$ *heißt* Cover*, wenn gilt:*

$$\lambda = \sum_{j\in C} a_j - b > 0. \tag{10.13}$$

Zusätzlich nennt man den Cover C minimal*, wenn* $a_j \geq \lambda$ *für alle* $j \in C$*.*

Mit jedem Cover C können wir eine einfache gültige Ungleichung identifizieren, die aussagt, dass nicht alle Variablen x_j für $j \in C$ gleichzeitig auf eins gesetzt werden können.

Theorem 10.9 ([?, ?, ?, ?]). *Sei $C \subseteq N$ ein Cover. Die* Coverungleichung

$$\sum_{j \in C} x_j \leq |C| - 1 \tag{10.14}$$

ist gültig für $P_K(N, a, b)$. Ist überdies C minimal, dann definiert die Ungleichung (??) eine Facette von $P_K(C, a, b)$.

Beweis. Die Gültigkeit ist offensichtlich. Um zu zeigen, dass diese eine Facette induziert, nehmen wir an, es gibt eine facettendefinierende Ungleichung $b^T x \leq \beta$ mit

$$\left\{ x \in P : \sum_{j \in C} x_j = |C| - 1 \right\} \subseteq F_b := \left\{ x \in P : b^T x = \beta \right\}.$$

Mit $C_i = C \setminus \{i\}$ folgt $\chi^{C_i} \in F_b$ für alle $i \in C$. Damit gilt

$$0 = b^T \chi^i - b^T \chi^j = b_i - b_j$$

und damit $b_i = b_j$ für alle $i, j \in C$.

Also ist $b^T x \leq \beta$ bis auf ein positives Vielfaches die Coverungleichung. □

Ist ein Cover C nicht minimal, dann kann man leicht erkennen, dass die zugehörige Coverungleichung redundant ist, d. h. sie ist die Summe einer minimalen Coverungleichung und einigen Bedingungen, die sich aus oberen Schranken ergeben.

Beispiel 10.10. Betrachte die 0/1-Rucksackmenge

$$P_K = \left\{ x \in \{0, 1\}^6 : 5x_1 + 5x_2 + 5x_3 + 5x_4 + 3x_5 + 8x_6 \leq 17 \right\}.$$

Dann ist $C = \{1, 2, 3, 4\}$ ein minimales Cover für P_K und die zugehörige Coverungleichung

$$x_1 + x_2 + x_3 + x_4 \leq 3$$

definiert eine Facette von

$$\operatorname{conv}\left\{ x \in \{0, 1\}^4 : 5x_1 + 5x_2 + 5x_3 + 5x_4 \leq 17 \right\}.$$

Es gibt viele weitere Klassen von Ungleichungen für das Rucksack-Polytop, u.a. sog. $(1,k)$-Konfigurations- oder "Extended Weight"-Ungleichungen. Darüber hinaus kann die sogenannte *Liftingmethode* verwendet werden, um solche Coverungleichungen, die keine Facetten definieren, zu verschärfen. Man erhält die Klasse der "Lifted Cover"-Ungleichungen, auf die wir hier nicht näher eingehen wollen. Weiterführende Literatur hierzu ist zu finden in [**?**, **?**, **?**].

Das Separieren der Coverungleichungen ist NP-schwer ([**?**, **?**]), so dass hier im allgemeinen auf Heuristiken zurückgegriffen wird.

Das Konzept eines Covers ist auch sinnvoll bei den Studien der polyedrischen Struktur von Problemen, die sowohl 0/1 als auch stetige Variablen enthalten. Betrachte die gemischte 0/1-Rucksackmenge

$$S = \left\{ \binom{x}{s} \in \{0,1\}^N \times \mathbb{R}_+ : \sum_{j \in N} a_j x_j \leq b + s \right\}$$

mit nichtnegativen Koeffizienten, d. h. $a_j \geq 0$ für $j \in N$ und $b \geq 0$.

Theorem 10.11. *[?] Sei $C \subseteq N$ ein Cover, d. h. C ist eine Teilmenge von N, die* (**??**) *erfüllt. Die Ungleichung*

$$\sum_{j \in C} \min(a_j, \lambda)\, x_j \leq \sum_{j \in C} \min(a_j, \lambda) - \lambda + s \qquad (10.15)$$

ist gültig für S. Überdies definiert die Ungleichung (**??**) *eine Facette von* $\operatorname{conv}(S_C)$*, wobei $S_C = S \cap \{x : x_j = 0,\, j \in N \setminus C\}$ gilt.*

Beweis. Wir zeigen die Gültigkeit der Ungleichung. Zum Beweis der Facetten-Eigenschaft verweisen wir auf [?].

Sei $\binom{x}{s} \in S$ beliebig. Sei $C_\lambda = \{j \in C : \lambda \leq a_j\}$ und $C_a = C \setminus C_\lambda$. Wir unterscheiden folgende Fälle:

1. $C_\lambda = \emptyset$. Dann gilt
$$\sum_{j \in C} a_j x_j \leq b + s = \sum_{j \in C} a_j - \lambda + s$$
2. $C_\lambda \neq \emptyset$.
   a) $x_j = 0$ für ein $j \in C_\lambda$. Dann gilt

$$\sum_{j\in C_\lambda} \lambda x_j + \sum_{j\in C_a} a_j x_j \le (|C_\lambda|-1)\lambda + \sum_{j\in C_a} a_j x_j$$
$$\le (|C_\lambda|-1)\lambda + \sum_{j\in C_a} a_j + s.$$

b) $x_j = 1$ für alle $j \in C_\lambda$.

$$\begin{aligned}\sum_{j\in C} \min(a_j,\lambda)\, x_j &= |C_\lambda|\lambda + \sum_{j\in C_a} a_j x_j \\ &\le |C_\lambda|\lambda + s - \sum_{j\in C_\lambda} a_j + b \\ &= |C_\lambda|\lambda + s - \sum_{j\in C_\lambda} a_j + \sum_{j\in C} a_j - \lambda \\ &= |C_\lambda|\lambda + \sum_{j\in C_a} a_j - \lambda + s \\ &= \sum_{j\in C} \min(a_j,\lambda) - \lambda + s\,.\end{aligned}$$

□

Beachte, dass jedes Cover C eine Coverungleichung impliziert, die eine Facette von $\mathrm{conv}(S_C)$ definiert. Dies ist anders als im rein ganzzahligen Fall, wo nur minimale Cover Facetten induzieren.

Beispiel 10.12. Wir betrachten die gemischte 0/1-Rucksackmenge

$$S = \Big\{ (x,s) \in \{0,1\}^6 \times \mathbb{R}_+ : \\ 5x_1 + 5x_2 + 5x_3 + 5x_4 + 3x_5 + 8x_6 \le 17 + s \Big\}\,. \quad (10.16)$$

Wählen wir $C' = \{1,2,3,6\}$ als (nichtminimalen) Cover für S, so definiert die zugehörige Coverungleichung

$$5x_1 + 5x_2 + 5x_3 + 6x_6 \le 15 + s$$

eine Facette von

$$\mathrm{conv}\Big\{ (x,s) \in \{0,1\}^4 \times \mathbb{R}_+ : \\ 5x_1 + 5x_2 + 5x_3 + 8x_6 \le 17 + s \Big\}\,.$$

Das Set-Packing Polytop

Ganzzahlige und gemischt ganzzahlige Probleme enthalten oft einige Nebenbedingungen, die nur 0/1-Koeffizienten

enthalten. Viele der Preprocessingroutinen für ganzzahlige Probleme erzeugen automatisch logische Ungleichungen der Form $x_i + x_j \leq 1, x_i \leq x_j$, Coverungleichungen, usw. Dies führt auf das Studium von ganzzahligen Programmen mit 0/1-Matrizen.

Das Studium solcher Probleme und im Besonderen Set-Packing und Covering-Probleme spielt eine bedeutende Rolle in der kombinatorischen Optimierung. Diese Probleme gehören zu den am häufigsten studierten mit einer weit entwickelten Theorie, die sich mit Begriffen wie perfekten, idealen, oder balancierten Matrizen, perfekten Graphen, der Theorie von blockierenden und anti-blockierenden Polyedern, Unabhängigkeitssystemen und semidefiniter Optimierung beschäftigt.

Der Fokus dieses Abschnitts liegt in der (teilweisen) Beschreibung der zugehörigen Polyeder mittels Ungleichungen. Unter der Annahme, dass Relaxierungen von verschiedenen ganzzahligen Problemen auf Set-Packing oder Covering Probleme führen, kann das Wissen über diese Polyeder genutzt werden, um die Formulierung des Ausgangsproblems zu verschärfen.

Definition 10.13. *Sei* $A \in \{0,1\}^{m \times n}$ *eine* 0/1*-Matrix und* $c \in \mathbb{R}^n$*. Die ganzzahligen* 0/1 *Probleme*

$$\max \left\{ c^T x : Ax \leq 1,\, x \in \{0,1\}^n \right\}$$
$$\min \left\{ c^T x : Ax \geq 1,\, x \in \{0,1\}^n \right\}$$

nennen wir das Set-Packing *und das* Set-Covering Problem*, vgl. Beispiel* **??**.

Jede Spalte j von A kann als der Inzidenzvektor einer Teilmenge F_j der Grundmenge $\{1, \ldots, m\}$ gesehen werden, d. h. es gilt $F_j := \{i \in \{1, \ldots, m\} : A_{ij} = 1\}$. Mit dieser Interpretation, besteht das Set-Packing Problem darin, eine Auswahl von Mengen aus $F_1, \ldots, F_n$ zu finden, die paarweise disjunkt und maximal bzgl. der Zielfunktion c sind. Analogerweise zielt das Covering-Problem auf das Auffinden einer Auswahl von Teilmengen, deren Vereinigung die Grundmenge überdeckt und die minimal bzgl. c ist.

Zulässige Mengen des Set-Packing Problems haben eine schöne graphentheoretische Interpretation. Wir führen einen Knoten für jeden Spaltenindex von A und eine Kante (i, j) zwischen zwei Knoten i und j ein, falls deren zugehörige Spalten einen gemeinsamen nichtverschwindenden Eintrag in derselben Zeile haben. Der resultierende

Graph, den wir mit $G(A)$ bezeichnen, heißt *(Spalten-) Konfliktgraph* (engl. *column intersection graph*). Offensichtlich ist jeder zulässige 0/1-Vektor x, der die Ungleichung $Ax \leq 1$ erfüllt, der Inzidenzvektor einer *stabilen Menge* ($U \subseteq V$ ist eine stabile Menge, falls aus $i, j \in U$ die Aussage $(i, j) \notin E$ folgt) im Graphen $G(A)$. Umgekehrt ist der Inzidenzvektor einer stabilen Menge in $G(A)$ eine zulässige Lösung des Set-Packing Problems $Ax \leq 1$. Also ist das Studium der stabilen Mengen in Graphen dem Studium des Set-Packing Problems gleichwertig.

Wir betrachten nun eine 0/1-Matrix A und bezeichnen mit

$$P(A) = \text{conv}\left\{ x \in \{ 0, 1 \}^N : Ax \leq 1 \right\}$$

das *Set-Packing Polytop*. Sei $G(A) = (V, E)$ der Konfliktgraph von A. Aus unseren bisherigen Überlegungen folgt, dass $P(A) = \text{conv}\{x \in \{0,1\}^n : x_i + x_j \leq 1, (i, j) \in E\}$ ist, wobei Letzteres eine Formulierung als ganzzahliges Problem des Problems der stabilen Mengen in G ist. Anders ausgedrückt, man erhält mit zwei Matrizen A und A' genau dasselbe Set-Packing Polytop, wenn deren zugehörige Konfliktgraphen übereinstimmen. Wir können also $P(A)$ betrachten mittels des Graphen G und bezeichnen nun das Set-Packing Polytop und das Stabile Mengen Polytop mit $P(G)$.

Anmerkung 10.14. Einige einfache Anmerkungen zu $P(G)$.

(i) $P(G)$ ist volldimensional.
(ii) $P(G)$ ist absteigend monoton, d. h. $x \in P(G)$ impliziert $y \in P(G)$ für alle $0 \leq y \leq x$. Alle nichttrivialen Facetten von $P(G)$ haben nichtnegative Koeffizienten.
(iii) Die Nichtnegativitätsbedingungen $x_j \geq 0$ induzieren Facetten von $P(G)$, siehe Übung.

Aus Satz **??** wissen wir, dass die Kanten und die Nichtnegativitätsbedingungen genau dann genügen, um $P(G)$ zu beschreiben, wenn G bipartit ist. Nicht-bipartite Graphen enthalten ungerade Kreise. Ungerade Kreise erzeugen neue gültige Ungleichungen, die nicht als Linearkombinationen von Kantenungleichungen erzeugt werden können.

Theorem 10.15 ([?]). *Sei $C \subseteq E$ ein Kreis ungerader Kardinalität in G. Die* Ungerade-Kreis Ungleichung

$$\sum_{i \in V(C)} x_i \leq \frac{|V(C)| - 1}{2}$$

ist gültig für $P(G)$. Die Ungleichung definiert genau dann eine Facette von $P(\,(V(C), E(V(C))\,)$, wenn C ein ungerades Loch ist, d. h. ein Kreis ohne Sehne.

Beweis. Die Gültigkeit der Ungleichung ist klar.

Zur Charakterisierung der Facetten-Eigenschaft betrachte ein ungerades Loch C mit $V(C) = \{0, \ldots, k-1\}, k \in \mathbb{N}$, ungerade. Sei $b^T x \leq \beta$ eine facettendefinierende Ungleichung mit

$$F_a := \left\{ x \in P : \sum_{i \in V(C)} x_i = \frac{|V(C)|-1}{2} \right\} \subseteq F_b := \left\{ x \in P : b^T x = \beta \right\} . \tag{10.17}$$

Betrachte für ein $i \in V(C)$ die stabilen Mengen $S_1 = \{j : j = i+2, i+4, \ldots, i-3, i-1\}$ und $S_2 = \{j : j = i+2, i+4, \ldots, i-3, i\}$, wobei alle Indizes modulo k gerechnet sind. Dann gilt $\chi^{S_1}, \chi^{S_2} \in F_b$ und damit $0 = b^T\chi^{S_1} - b^T\chi^{S_2} = b_i - b_{i-1}$. Da i beliebig gewählt war, folgt $b_i = b_j$ für alle $i, j \in V(C)$. Damit ist $b^T x \leq \beta$ bis auf ein positives Vielfaches die Kreisungleichung.

Betrachte umgekehrt einen Kreis C mit Diagonalen. Dann enthält C ein ungerades Loch H unter Hinzunahme einer Kante $(i, j) \in E(V(C)) \setminus C$. Betrachte paarweise verschiedene Knoten $i_l, j_l \in V(C) \setminus V(H)$ für $l = 1, \ldots, \frac{|V(C)|-|V(H)|}{2}$
mit $(i_l, j_l) \in C$. Dann ist

$$\sum_{i \in V(C)} x_i \leq \frac{|V(C)|-1}{2}$$

die Summe aus den gültigen Ungleichungen

$$\sum_{i \in V(H)} x_i \leq \frac{|V(H)|-1}{2} \text{ und}$$
$$x_{i_l} + x_{j_l} \leq 1$$

für $l = 1, \ldots, \frac{|V(C)|-|V(H)|}{2}$. Also kann es keine Facette induzieren. □

Ungerade-Kreis Ungleichungen können in polynomialer Zeit mittels des Algorithmus aus Lemma 9.1.11 in [?] separiert werden. Graphen $G = (V, E)$, für die $P(G)$ völlig durch die *Kantenungleichungen* $x_i + x_j \leq 1$ für

$(i,j) \in E$ und die Ungerader-Kreis Ungleichungen beschrieben wird, heißen *t-perfekt*, vgl. Chvátal [**?**]. Die Klasse der t-perfekten Graphen enthält eine Reihe von Graphen, wie z. B. serien-parallele und bipartite Graphen.

Eine weitere wichtige Klasse von gültigen Ungleichungen für das Stabile Mengen Polytop sind die Cliquenungleichungen.

Theorem 10.16. *[?, ?] Sei $(C, E(C))$ eine Clique in G. Die Ungleichung*

$$\sum_{i \in C} x_i \leq 1$$

ist gültig für $P(G)$. Sie definiert genau dann eine Facette von $P(G)$, wenn $(C, E(C))$ maximal bzgl. Knoteninklusion ist.

Beweis. Übung.

Graphen $G = (V, E)$, für die $P(G)$ vollständig durch die Cliquengleichungen beschrieben werden, heißen *perfekt*, eine Bezeichnungsweise, die auf Berge [**?**] zurückgeht.

Leider ist das Separierungsproblem für die Klasse der Cliquenungleichungen NP-schwer, siehe [?], Satz 9.2.9. Überraschenderweise gibt es eine größere Klasse von Ungleichungen, die sogenannten *orthonormalen Repräsentations-Ungleichungen*, die die Cliquenungleichungen umfassen und die in polynomialer Zeit separiert werden können. Neben den Kreis-, Cliquen und OR-Ungleichungen gibt es eine Menge anderer Ungleichungen, die für das Stabile-Mengen-Polytop bekannt sind. Unter diesen findet man z. B. die Blüten-, ungerade Antiloch-, Räder-, Antinetz- und Netz-, Keil-Ungleichungen (engl. blossom, odd antihole, wheel, antiweb and web, wedge inequalities) und viele weitere. Auskunft darüber gibt [**?**] inklusive einer Diskussion deren Separierbarkeit.

10.2 Lagrange-Relaxierungen

Im vorigen Abschnitt haben wir das gemischt ganzzahlige Problem mittels Relaxierung der Ganzzahligkeitsbedingungen zu lösen versucht und durch den Versuch, die Ganzzahligkeit der Lösung durch Hinzufügung von Schnittebenen zu erreichen. In der Methode, die wir nun betrachten wollen, behalten wir die Ganzzahligkeitsbedingungen

bei, relaxieren aber die Teile der Nebenbedingungsmatrix, die Schwierigkeiten erzeugt. Der gelöschte Teil wird wieder in das Problem eingeführt, indem er in die Zielfunktion mit Straftermen versehen wird. Betrachte wieder (**??**). Wir zerlegen A und b in zwei Teile $A = \binom{A_1}{A_2}$ und $b = \binom{b_1}{b_2}$, wobei $A_1 \in \mathbb{R}^{m_1 \times n}, A_2 \in \mathbb{R}^{m_2 \times n}, b_1 \in \mathbb{R}^{m_1}, b_2 \in \mathbb{R}^{m_2}$ mit $m_1 + m_2 = m$ ist. Dann wird (**??**) zu

$$\begin{aligned} \min\, & c^T x \\ & A_1 x \leq b_1 \\ & A_2 x \leq b_2 \\ & x \in \mathbb{Z}^{n-p} \times \mathbb{R}^p \,. \end{aligned} \tag{10.18}$$

Betrachte für ein festes $\lambda \in \mathbb{R}^{m_1}, \lambda \geq 0$ die folgende Funktion

$$L(\lambda) = \min c^T x - \lambda^T (b_1 - A_1 x) \tag{10.19}$$

wobei

$$x \in P^2 \text{ mit } P^2 = \left\{ x \in \mathbb{Z}^{n-p} \times \mathbb{R}^p : A_2 x \leq b_2 \right\}$$

gelte. Wir nennen (**??**) die *Lagrange-Funktion*. Offensichtlich ist (**??**) eine untere Schranke für (**??**), da für jede zulässige Lösung $\bar{x}$ von (**??**) gilt

$$\begin{aligned} c^T \bar{x} &\geq c^T \bar{x} - \lambda^T (b_1 - A_1 \bar{x}) \\ &\geq \min_{x \in P^2} c^T x - \lambda^T (b_1 - A_1 x) \\ &= L(\lambda) \,. \end{aligned}$$

Da dies für jedes $\lambda \geq 0$ gilt, erhalten wir mit

$$\max_{\lambda \geq 0} L(\lambda) \tag{10.20}$$

eine untere Schranke für (**??**). (**??**) heißt *Lagrange-Relaxierung*. Häufig wird **??** das "Lagrange Dual" bezeichnet und der gesamte Ansatz Lagrange-Relaxierung. Sei λ^* eine optimale Lösung von (**??**). Es bleiben die Fragen, wie gut ist $L(\lambda^*)$ und wie kann λ^* berechnet werden. Eine Antwort auf die erste Frage erhalten wir durch folgenden Satz.

Theorem 10.17. *Es gilt*

$$L(\lambda^*) = \min \left\{ c^T x : A_1 x \leq b_1, x \in \text{conv}(P^2) \right\} \,.$$

Beweis. Es gilt

$$\begin{aligned} L(\lambda) &= \min_{x \in P^2} c^T x - \lambda^T (b_1 - A_1 x) \\ &\min_{x \in \mathrm{conv}(P^2)} c^T x - \lambda^T (b_1 - A_1 x) \end{aligned}$$

und damit

$$\begin{aligned} L(\lambda^*) &= \max_{\lambda \geq 0} L(\lambda) \\ &= \max_{\lambda \geq 0} \min_{x \in \mathrm{conv}(P^2)} c^T x - \lambda^T (b_1 - A_1 x). \end{aligned}$$

Falls $P^2 = \emptyset$ ist das innere Minimum $+\infty$ für alle λ und damit auch $L(\lambda^*)$. Andernfalls existieren Vektoren $v_1, \ldots, v_k$ und $e_1, \ldots, e_l$, so dass

$$\mathrm{conv}(P^2) = \mathrm{conv}(\{ v_1, \ldots, v_k \}) + \mathrm{cone}(\{ e_1, \ldots, e_l \}) .$$

Falls nun $(c^T + \lambda^T A_1) e_i < 0$ für ein $i \in \{1, \ldots, l\}$ ist, gilt daher

$$\min_{x \in \mathrm{conv}(P^2)} c^T x - \lambda^T (b_1 - A_1 x) = -\infty ,$$

andernfalls gilt

$$\min_{x \in \mathrm{conv}(P^2)} c^T x - \lambda^T (b_1 - A_1 x) = c^T v_j - \lambda^T (b_1 - A_1 v_j)$$

für ein $j \in \{1, \ldots, k\}$. Also gilt auch

$$\begin{aligned} L(\lambda^*) = \max_{\lambda \geq 0} \min_{j \in \{ 1, \ldots, k \}} & c^T v_j - \lambda^T (b_1 - A_1 v_j) \\ & (c^T + \lambda^T A_1) e_i \geq 0 \text{ für } i = 1, \ldots, l . \end{aligned} \tag{10.21}$$

Letzteres ist äquivalent zu

$$\begin{aligned} L(\lambda^*) = & \max_{\lambda \geq 0, \eta \in \mathbb{R}} \eta \\ \eta + \lambda^T (b_1 - A_1 v_j) \leq & c^T v_j \qquad \text{für } j = 1, \ldots, k \\ -\lambda^T A_1 e_i \leq & c^T e_i \qquad \text{für } i = 1, \ldots, l . \end{aligned}$$

Mit dem Dualitätssatz der Linearen Programmierung erhalten wir schließlich

$$
\begin{aligned}
L(\lambda^*) = \min c^T (\sum_{j=1}^{k} \alpha_j v_j + \sum_{i=1}^{l} \beta_i e_i) & \\
\sum_{j=1}^{k} \alpha_j = 1 & \\
-A_1(\sum_{j=1}^{k} \alpha_j v_j + \sum_{i=1}^{l} \beta_i e_i) \geq -b_1(\sum_{j=1}^{k} \alpha_j) & \\
\alpha_j \geq 0 \quad \text{für } j = 1, \ldots, k & \\
\beta_i \geq 0 \quad \text{für } i = 1, \ldots, l & \\
= \min \left\{ c^T x : A_1 x \leq b_1, x \in \operatorname{conv}(P^2) \right\} . &
\end{aligned}
$$

□

Wegen

$$
\begin{aligned}
\{ x \in \mathbb{R}^n : Ax \leq b \} &\supseteq \left\{ x \in \mathbb{R}^n : A_1 x \leq b_1,\, x \in \operatorname{conv}(P^2) \right\} \\
&\supseteq \operatorname{conv} \left\{ x \in \mathbb{Z}^{n-p} \times \mathbb{R}^p : Ax \leq b \right\}
\end{aligned}
$$

erhalten wir aus Satz **??**

Korollar 10.18. *Es gilt*

$$z_{LP} \leq L(\lambda^*) \leq z_{IP} ,$$

wobei z_{LP} der optimale Lösungswert der LP-Relaxierung und z_{IP} der optimale ganzzahlige Lösungswert ist.

Es bleibt zu überlegen, auf welche Art $L(\lambda^*)$ berechnet werden kann. Aus theoretischer Sicht kann gezeigt werden, dass unter der Ausnutzung der Äquivalenz von Separieren und Optimieren $L(\lambda^*)$ in polynomialer Zeit berechnet werden kann, wenn $\min\{\tilde{c}^T x : x \in \operatorname{conv}(P^2)\}$ in polynomialer Zeit für jede Zielfunktion $\tilde{c}$ berechnet werden kann, siehe [?]. Wir werden darauf später noch einmal zurückkommen.

Zur Berechnung von $L(\lambda^*)$ werden häufig Subgradientenverfahren verwendet. Dazu folgende Überlegungen:

Theorem 10.19. *Die Funktion $L(\lambda)$ ist stückweise linear und konkav auf dem Definitionsbereich, über dem sie beschränkt ist.*

Beweis. Zunächst beobachten wir, dass $L(\lambda)$ genau dann endlich ist, wenn λ in dem Polyeder

$$\left\{ \hat{\lambda} \in \mathbb{R}_+^{m_1} : -\hat{\lambda}^T A_1 e_i \leq c^T e_i,\, i = 1, \ldots, l \right\}$$

liegt, siehe (**??**). In diesem Polyeder gilt

$$L(\lambda) = \min_{j \in \{1,\ldots,k\}} c^T v_j - \lambda^T (b_1 - A_1 v_j)\,,$$

d. h. $L(\lambda)$ ist das Minimum über eine endliche Anzahl affiner Funktionen.

Bleibt zu zeigen, dass die Funktion konkav ist. Betrachte dazu $\lambda^3 = \alpha\lambda^1 + (1-\alpha)\lambda^2$ für $\lambda^1, \lambda^2, \lambda^3 \geq 0$. Zu zeigen ist, dass $L(\lambda^3) \geq \alpha L(\lambda^1) + (1-\alpha)L(\lambda^2)$ gilt. Seien v_{j_t} die zugehörigen Optimallösungen mit $L(\lambda^t) = c^T v_{j_t} + (\lambda^t)^T(b_1 - A_1 v_{j_t})$ für $t = 1, 2, 3$. Dann gilt

$$\begin{aligned}
L(\lambda^3) &= c^T v_{j_3} - (\lambda^3)^T (b_1 - A_1 v_{j_3}) \\
&= c^T v_{j_3} - (\alpha\lambda^1 + (1-\alpha)\lambda^2)^T (b_1 - A_1 v_{j_3}) \\
&= \alpha(c^T v_{j_3} - (\lambda^1)^T (b_1 - A_1 v_{j_3})) \\
&\quad + (1-\alpha)(c^T v_{j_3} - (\lambda^2)^T (b_1 - A_1 v_{j_3})) \\
&\geq \alpha(c^T v_{j_1} - (\lambda^1)^T (b_1 - A_1 v_{j_1})) \\
&\quad + (1-\alpha)(c^T v_{j_2} - (\lambda^2)^T (b_1 - A_1 v_{j_2})) \\
&= \alpha L(\lambda^1) + (1-\alpha) L(\lambda^2)\,.
\end{aligned}$$

□

Definition 10.20. *Sei $L\colon \mathbb{R}^{m_1} \mapsto \mathbb{R}$ eine konkave Funktion. Dann heißt ein Vektor $u \in \mathbb{R}^{m_1}$* Subgradient *an L im Punkt λ_0, falls*

$$L(\lambda) - L(\lambda_0) \leq u^T(\lambda - \lambda_0)$$

für alle $\lambda \in \mathbb{R}^{m_1}$.

Theorem 10.21.

*(a) Betrachte $\lambda^0 \in \mathbb{R}_+^{m_1}$ und eine Optimallösung x^0 für (**??**). Dann ist $g^0 = A_1 x^0 - b_1$ ein Subgradient an L in λ^0.*

(b) λ^* maximiert L genau dann, wenn 0 ein Subgradient an L in λ^* ist.

Beweis. Zu (a):

$$\begin{aligned}
L(\lambda) - L(\lambda^0) &= c^T x^\lambda - \lambda^T (b_1 - A_1 x^\lambda) \\
&\quad - (c^T x^0 - (\lambda^0)^T (b_1 - A_1 x^0)) \\
&\leq c^T x^0 - \lambda^T (b_1 - A_1 x^0) \\
&\quad - (c^T x^0 - (\lambda^0)^T (b_1 - A_1 x^0)) \\
&= (g^0)^T (\lambda - \lambda^0)\,.
\end{aligned}$$

Zu (b): Ist 0 ein Subgradient in λ^*, so gilt

$$L(\lambda) - L(\lambda^*) \leq 0^T(\lambda - \lambda^*) = 0$$

für alle $\lambda \in \mathbb{R}^{m_1}$. Also maximiert λ^* die Funktion L. Umgekehrt, falls $L(\lambda^*)$ maximal ist, gilt $L(\lambda^*) \geq L(\lambda)$ für alle $\lambda \in \mathbb{R}^{m_1}$, und damit

$$L(\lambda) - L(\lambda^*) \leq 0 = 0^T(\lambda - \lambda^*)\,.$$

Folglich ist 0 ein Subgradient. □

Also gilt für λ^* die Aussage $(g^0)^T(\lambda^* - \lambda^0) \geq L(\lambda^*) - L(\lambda^0) \geq 0$. Daher können wir, um λ^* zu finden, mit einem λ^0 beginnen, dann berechnen wir

$$x^0 = \operatorname{argmin}\left\{ c^T x - (\lambda^0)^T(b_1 - A_1 x) : x \in P^2 \right\}$$

und bestimmen iterativ $\lambda^0, \lambda^1, \lambda^2, \ldots$, indem wir $\lambda^{k+1} = \lambda^k + \mu^k g^k$ setzen, wobei μ^k eine noch zu bestimmende Schrittlänge ist. Diese iterative Methode ist die Essenz der *Subgradientenmethode*.

Algorithmus 10.1 : Subgradientenmethode

Input : Eine konkave Funktion $L\colon \mathbb{R}^n \mapsto \mathbb{R}$
Output : $\max\{L(\lambda) : \lambda \in \mathbb{R}^n_+\}$

1 Wähle $\lambda^0 \in \mathbb{R}^n$ beliebig;
2 Setze $k = 0$;
3 Berechne $L(\lambda^k)$. Sei x^k zugehörige Optimallösung;
4 **if** *Stopp Kriterium ist erfüllt* **then**
  stop gib λ^k und x^k aus;
end
5 Wähle ein neues λ^{k+1} durch

$$\lambda^{k+1} = \lambda^k + \mu^k g^k$$

wobei μ^k eine zu spezifizierende Schrittlänge ist;
6 Setze $k = k + 1$ und gehe zu Schritt **??**.

Theorem 10.22. *Falls die konkave Funktion $L\colon \mathbb{R}^{m_1} \mapsto \mathbb{R}$ nach oben beschränkt ist und die Folge (μ^k) von Schrittlängen*

$$\lim_{k\to\infty} \mu^k = 0 \text{ und } \sum_{k=0}^{\infty} \mu^k = \infty\,,$$

erfüllt, so gilt

$$\lim_{k\to\infty} L(\lambda^k) = L(\lambda^*)\,.$$

Beweis. siehe z. B. [**?**].

Natürlich hängt die Qualität der Lagrange-Relaxierung sehr davon ab, welche Menge an Bedingungen relaxiert wird. Einerseits müssen wir (**??**) für verschiedene Werte von λ bestimmen und damit ist es notwendig, einen Wert von $L(\lambda)$ schnell zu berechnen. Es ist einerseits also wünschenswert, so viele (komplizierte) Bedingungen wie möglich zu relaxieren. Andererseits, je mehr Bedingungen relaxiert werden, desto schlechter wird die Schranke $L(\lambda^*)$ werden. Also muss man einen Kompromiss zwischen diesen beiden sich widersprechenden Zielen finden. Im Folgenden geben wir einige Anwendungen, bei denen diese Methode erfolgreich angewendet werden konnte und eine gute Balance zwischen den beiden gegensätzlichen Zielen gefundenen werden kann.

Anwendungen

Eine Anwendung ist das Problem des Handlungsreisenden. Durch Relaxierung der Gradbedingungen in der IP-Formulierung für dieses Problem bleibt das Problem des Auffindens eines aufspannenden Baumes erhalten, welches leicht mit dem Greedy-Algorithmus gelöst werden kann. Ein Hauptvorteil dieser TSP-Relaxierung ist, dass für die Berechnung von (**??**) kombinatorische Algorithmen vorhanden sind und kein allgemeines LP oder IP Lösungsverfahren angewendet werden muss.

Lagrange-Relaxierungen werden sehr oft benutzt, wenn die zugrundeliegenden LP's von (**??**) einfach zu groß sind, um direkt gelöst werden zu können und sogar die relaxierten Probleme in (**??**) sind noch sehr groß. Oft kann die Relaxierung in der Weise getan werden, dass die Berechnung von (**??**) kombinatorisch geschehen kann. Andere Beispiele sind Multicommodity Flow-Probleme, die z. B. in der Fahrzeugumlaufplanung oder in Zerlegungen von stochastischen gemischt ganzzahligen Problemen auftreten. In der Tat gehören die letzten beiden Anwendungen zu einer Klasse von Problemen, wobei die zugrundeliegende Matrix (nahezu) Blockdiagonalform hat, siehe Bild **??**. Relaxieren wir die Kopplungsbedingungen in einer Lagrange-Relaxierung, so zerfällt die verbleibende Matrix in k unabhängige Blöcke. Also ist ein Wert $L(\lambda)$ die Summe von k unabhängigen Termen, die getrennt bestimmt werden können. Oft stellt jeder einzelne Block A_i ein Netzwerk

Flussproblem, ein Rucksackproblem oder dergleichen dar und kann daher unter Benutzung spezieller kombinatorischer Algorithmen gelöst werden.

Abb. 10.1. Matrix in begrenzter Blockdiagonalform

10.3 Dekompositionsmethoden

10.3.1 Dantzig-Wolfe Dekomposition

Die Idee von Dekompositionsmethoden besteht darin, eine Menge von Bedingungen (Variablen) aus dem Problem herauszunehmen und diese auf einer übergeordneten Ebene (oft *Masterproblem* genannt) zu betrachten. Das resultierende untergeordnete Problem kann oft effizienter gelöst werden. Dekompositionsmethoden arbeiten nun abwechselnd auf dem Master- und dem Subproblem und tauschen iterativ Informationen aus, um das Ausgangsproblem optimal zu lösen. In diesem Abschnitt betrachten wir zwei bekannte Beispiele dieses Ansatzes, die Dantzig-Wolfe Dekomposition und Benders' Dekomposition. Wir werden erkennen, dass wie im Fall der Lagrange-Relaxierung diese Methoden ebenso einen Teil der Nebenbedingungsmatrix weglassen. Aber anstelle diesen Teil wieder in die Zielfunktion einzuführen, wird dieser nun umformuliert und wieder in das System der Nebenbedingungen eingeführt.

Wir beginnen mit der Dantzig-Wolfe Dekomposition [**?**] und betrachten wieder (**??**), wobei wir zunächst annehmen, dass $p = 0$, d. h. wir haben ein LP. Betrachten wir das Polyeder $P^2 = \{x \in \mathbb{R}^n : A_2 x \leq b_2\}$. Wir wissen aus Teil I, dass es Vektoren $v_1, \ldots, v_k$ und $e_1, \ldots, e_l$ gibt, so dass

$$P^2 = \text{conv}(\{ v_1, \ldots, v_k \}) + \text{cone}(\{ e_1, \ldots, e_l \}) .$$

In anderen Worten, $x \in P^2$ kann in der Form

$$x = \sum_{i=1}^{k} \lambda_i v_i + \sum_{j=1}^{l} \mu_j e_j \tag{10.22}$$

mit $\lambda_1, \ldots, \lambda_k \geq 0, \sum_{i=1}^{k} \lambda_i = 1$ und $\mu_1, \ldots, \mu_l \geq 0$ geschrieben werden. Ersetzen wir x aus (**??**), so können wir (**??**) schreiben als

$$\begin{aligned} \min\, & c^T \Big(\sum_{i=1}^{k} \lambda_i v_i + \sum_{j=1}^{l} \mu_j e_j\Big) \\ & A_1 \Big(\sum_{i=1}^{k} \lambda_i v_i + \sum_{j=1}^{l} \mu_j e_j\Big) \leq b_1 \\ & \sum_{i=1}^{k} \lambda_i = 1 \\ & \lambda \in \mathbb{R}^k_+, \mu \in \mathbb{R}^l_+ \,, \end{aligned} \tag{10.23}$$

was gleichwertig ist zu

$$\begin{aligned} \min\, & \sum_{i=1}^{k} (c^T v_i)\, \lambda_i + \sum_{j=1}^{l} (c^T e_j)\, \mu_j \\ & \sum_{i=1}^{k} (A_1 v_i)\, \lambda_i + \sum_{j=1}^{l} (A_1 e_j)\, \mu_j \leq b_1 \\ & \sum_{i=1}^{k} \lambda_i = 1 \\ & \lambda \in \mathbb{R}^k_+, \mu \in \mathbb{R}^l_+ \,. \end{aligned} \tag{10.24}$$

(**??**) nennt man das *Masterproblem* von (**??**). Beim Vergleich der Formulierungen (**??**) und (**??**) erkennen wir, dass wir die Zahl der Nebenbedingungen von m auf m_1 reduziert haben, aber wir haben nun $k+l$ Variablen anstelle von n. Der Wert $k + l$ kann im Vergleich zu n groß sein, ja sogar exponentiell (betrachte z. B. den Einheitswürfel im $\mathbb{R}^n$ mit $2n$ Nebenbedingungen und 2^n Ecken), so dass auf den ersten Blick kein Vorteil in der Anwendung von (**??**) erkennbar ist. Dennoch können wir die Simplexmethode für die Lösung von (**??**) verwenden. Wir kürzen (**??**) durch

$$\min \left\{\, w^T \eta : D\eta = d, \eta \geq 0 \,\right\}$$

mit $D \in \mathbb{R}^{(m_1+1)\times(k+l)}$, $d \in \mathbb{R}^{m_1+1}$ ab. Man erinnere sich, dass die Simplexmethode mit einer (zulässigen) Basis $B \subseteq \{1, \ldots, k+l\}$, $|B| = m_1 + 1$, mit D_B regulär und der zugehörigen (zulässigen) Lösung $\eta^*_B = D_B^{-1} d$ und $\eta^*_N = 0$ mit $N = \{1, \ldots, k+l\} \setminus B$ startet. Beachte,

dass $D_B \in \mathbb{R}^{(m_1+1)\times(m_1+1)}$ (viel) kleiner als eine Basis für das ursprüngliche System (**??**) ist und dass nur ein Teil der Variablen ($m_1 + 1$ von $k + l$) nichtverschwinden können. Zusätzlich gilt, dass auf dem Weg zu einer optimalen Lösung die einzige Operation in der Simplexmethode, die alle Spalten benutzt, der Schritt Pricing ist, bei dem geprüft wird, ob die reduzierten Kosten $w_N - \tilde{y}^T D_N$ nichtnegativ mit $\tilde{y} \leq 0$ als Lösung von $y^T D_B = w_B$ ist. Die Nichtnegativität der reduzierten Kosten kann mittels des folgenden linearen Programms geprüft werden:

$$\begin{aligned} \min(c^T - \bar{y}^T A_1)x & \\ A_2 x &\leq b_2 \\ x &\in \mathbb{R}^n \,, \end{aligned} \tag{10.25}$$

Dabei sind $\bar{y}$ die ersten m_1 Komponenten der Lösung von $\tilde{y}$. Die folgenden Fälle können auftreten:

(i) Das LP (**??**) hat eine optimale Lösung $\tilde{x}$ mit

$$(c^T - \bar{y}^T A_1)\tilde{x} < \tilde{y}_{m_1+1} \,.$$

In diesem Fall ist $\tilde{x}$ einer der Vektoren $v_i, i \in \{1, \ldots, k\}$ mit zugehörigen reduzierten Kosten

$$\begin{aligned} w_i - \tilde{y}^T D_{\cdot i} &= c^T v_i - \tilde{y}^T \tbinom{A_1 v_i}{1} \\ &= c^T v_i - \bar{y}^T A_1 v_i - \tilde{y}_{m_1+1} \\ &< 0 \,. \end{aligned}$$

In anderen Worten: $\binom{A_1 v_i}{1}$ ist die eintretende Spalte im Simplexalgorithmus.

(ii) Das LP (**??**) ist unbeschränkt. Wir erhalten einen zulässigen Extremstrahl e^* mit

$$(c^T - \bar{y}^T A_1)e^* < 0 \,,$$

und e^* ist einer der Vektoren e_j, $j \in \{1, \ldots, l\}$. Wir erhalten eine Spalte $\binom{A_1 e_j}{0}$ mit reduzierten Kosten

$$\begin{aligned} w_{k+j} - D_{\cdot(k+j)} &= c^T e_j - \tilde{y}^T \tbinom{A_1 e_j}{0} \\ &= c^T e_j - \bar{y}^T (A_1 e_j) \\ &< 0 \,. \end{aligned}$$

Also ist $\binom{A_1 e_j}{0}$ die eintretende Spalte.

(iii)Das LP (**??**) hat eine optimale Lösung $\tilde{x}$ mit

$$(c^T - \bar{y}^T A_1)^T \tilde{x} \geq \tilde{y}_{m_1+1} .$$

In diesem Fall erhalten wir mit denselben Argumenten wie in (i) und (ii), dass $w_i - \tilde{y}^T D_{\cdot i} \geq 0$ für alle $i = 1, \ldots, k+l$ gilt, was zeigt, dass x^* eine optimale Lösung des Masterproblems (**??**) ist.

Man beachte, dass das ganze Problem (**??**) in zwei Teilprobleme zerlegt wird, d. h. in (**??**) und (**??**) und dieser Ansatz arbeitet iterativ auf der höheren Ebene (**??**) und auf der niedrigeren Ebene (**??**). Das Verfahren beginnt mit einer zulässigen Lösung für (**??**) und erzeugt neue erfolgversprechende Spalten nach Bedarf durch Lösung von (**??**). Solche Verfahren werden gewöhnlich *spaltenerzeugende* oder *verzögerte spaltenerzeugende Algorithmen* (engl. delayed column generation algorithms) genannt.

Dieser Ansatz kann ebenso auf allgemeine IP's mit einiger Vorsicht übertragen werden. In diesem Fall verändert sich das Problem (**??**) von einem linearen zu einem ganzzahligen linearen Problem. Zusätzlich müssen wir sicherstellen, dass in (**??**) alle zulässigen ganzzahligen Lösungen x von (**??**) erzeugt werden können durch (ganzzahlige) Linearkombinationen der Vektoren $v_1, \ldots, v_k$ und $e_1, \ldots, e_l$ mit

$$\operatorname{conv}\{\, x \in \mathbb{Z}^n : Ax \leq b \,\} = \operatorname{conv}(v_1, \ldots, v_k) + \operatorname{cone}(e_1, \ldots, e_l) .$$

Es genügt nicht zu verlangen, dass λ und μ ganzzahlig sein müssen. Betrachte als Gegenbeispiel

$$A_1 = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix},\ b_1 = \begin{pmatrix} 1.5 \\ 1.5 \end{pmatrix} \text{ und } A_2 = \begin{pmatrix} 1 \\ 1 \end{pmatrix},\ b_2 = 2$$

sowie das Problem

$$\max \left\{ x_1 + x_2 : A_1 x \leq b_1, A_2 x \leq b_2, x \in \{\, 0, 1, 2 \,\}^2 \right\} .$$

Dann ist $P^2 = \operatorname{conv}(\{\binom{0}{0}, \binom{2}{0}, \binom{0}{2}\})$, siehe Abbildung **??**, aber die optimale Lösung $\binom{1}{1}$ des ganzzahligen Problems ist keine ganzzahlige Linearkombination der Ecken von P^2. Sind jedoch alle Variablen 0/1, so tritt diese Schwierigkeit nicht auf, da jede 0/1-Lösung der LP-Relaxierung eines binären MIP immer eine Ecke dieses Polyeders ist

(Übung). In der Tat werden spaltenerzeugende Algorithmen nicht nur für die Lösung von großen linearen Problemen benutzt, sondern insbesondere für große binäre ganzzahlige Probleme.

Abb. 10.2. Erweiterung der Dantzig-Wolfe Dekomposition auf IP's

Natürlich ist die Dantzig-Wolfe Zerlegung für lineare oder binäre ganzzahlige Probleme nur eine Art von spaltenerzeugenden Algorithmen. Andere Autoren lösen das untergeordnete Problem nicht mittels allgemeiner Techniken für LP oder IP Probleme sondern durch kombinatorische oder explizite enumerative Algorithmen. Weiterhin werden die Probleme oft nicht über (**??**) modelliert, sondern direkt wie in (**??**). Dies ist z. B. der Fall, wenn die Menge der zulässigen Lösungen eine komplexe Beschreibung durch lineare Ungleichungen hat, aber diese Bedingungen leicht in ein Enumerationsschema eingebaut werden können.

10.3.2 Benders' Dekomposition

Abschließend wollen wir *Benders' Dekomposition* [**?**] betrachten. Benders' Dekomposition eliminiert auch einen Teil der Nebenbedingungsmatrix, aber anders als bei der Dantzig-Wolfe Dekomposition, bei der wir einen Teil der Nebenbedingungen löschen und diese wieder mittels Spaltenerzeugung erneut einfügen, löschen wir nun einen Teil der Variablen und führen diese wieder mittels Schnittebenen ein. Aus dieser Sicht ist Benders' Dekomposition gleich der Dantzig-Wolfe Dekomposition angewendet auf das duale Problem, was wir in Abschnitt **??** sehen werden. Betrachte wieder (**??**) und schreibe es in der Form

$$\begin{aligned} \min\, & c_1^T x_1 + c_2^T x_2 \\ & A_1 x_1 + A_2 x_2 \leq b \\ & x_1 \in \mathbb{R}^{n_1}, x_2 \in \mathbb{R}^{n_2}\,, \end{aligned} \tag{10.26}$$

wobei $A = [A_1\ A_2] \in \mathbb{R}^{m \times n}$ und dazu verträglich

$A_1 \in \mathbb{R}^{m \times n_1}$, $A_2 \in \mathbb{R}^{m \times n_2}$, $c_1, x_1 \in \mathbb{R}^{n_1}$ und $c_2, x_2 \in \mathbb{R}^{n_2}$

mit $n_1 + n_2 = n$ gilt. Beachte, dass wir zur Einfachheit der Darstellung den kontinuierlichen Fall angenommen haben. Wir werden jedoch sehen, dass alle Ergebnisse auch für den Fall $x_1 \in \mathbb{Z}^{n_1}$ gültig sind. Wir wollen die Variablen x_2 entfernen. Diese Variablen halten (**??**) davon ab, ein reines ganzzahliges Programm zu sein, falls $x_1 \in \mathbb{Z}^{n_1}$. Auch im Fall eines LP's können sie der Grund für einige Schwierigkeiten sein, siehe beispielsweise die Anwendungen aus Stochastischen Programmierung. Ein bekannter Ansatz zur Entfernung von Variablen ist die Projektion.

Um Projektion anwenden zu können, müssen wir (**??**) umformulieren zu

$$\begin{aligned} &\min z \\ -z + c_1^T x_1 + c_2^T x_2 &\le 0 \\ A_1 x_1 + A_2 x_2 &\le b \\ z \in \mathbb{R},\, x_1 \in \mathbb{R}^{n_1},\, x_2 &\in \mathbb{R}^{n_2}\,, \end{aligned} \tag{10.27}$$

Referenz auf nicht vorhandenen Satz

Nun ist (**??**) äquivalent zu (vgl. Fourier-Motzkin-Elimination)

$$\begin{aligned} &\min z \\ -uz + uc_1^T x_1 + v^T A_1 x_1 &\le v^T b \\ z \in \mathbb{R},\, x_1 &\in \mathbb{R}^{n_1} \\ \tbinom{u}{v} &\in C\,, \text{ wobei} \end{aligned} \tag{10.28}$$

$$C = \left\{ \tbinom{u}{v} \in \mathbb{R}^{m+1} : v^T A_2 + uc_2^T = 0,\, u \ge 0, v \ge 0 \right\}\,.$$

C ist ein spitzer polyedrischer Kegel, es gibt also Vektoren $\binom{\bar{u}_1}{\bar{v}_1}, \ldots, \binom{\bar{u}_s}{\bar{v}_s}$, so dass gilt $C = \text{cone}(\{\binom{\bar{u}_1}{\bar{v}_1}, \ldots, \binom{\bar{u}_s}{\bar{v}_s}\})$. Diese Extremalstrahlen können so reskaliert werden, dass $\bar{u}_i$ zu null oder eins wird. Es folgt

$$C = \text{cone}(\left\{ \begin{pmatrix} 0 \\ v_k \end{pmatrix} : k \in K \right\}) + \text{cone}(\left\{ \begin{pmatrix} 1 \\ v_j \end{pmatrix} : j \in J \right\})$$

mit $K \cup J = \{1, \ldots, s\}$ und $K \cap J = \emptyset$, vgl. Teil I. Mit dieser Beschreibung von C kann (**??**) dargestellt werden als

$$\begin{aligned} &\min z \\ -z \le -c_1^T x_1 + v_j^T (b - A_1 x_1) &\quad \text{für alle } j \in J \\ 0 \le \qquad\quad v_k^T (b - A_1 x_1) &\quad \text{für alle } k \in K \\ z \in \mathbb{R},\, x_1 \in \mathbb{R}^{n_1}\,. & \end{aligned} \tag{10.29}$$

(**??**) nennt man *Benders' Masterproblem*. Benders' Masterproblem hat nur $n_1 + 1$ Variablen anstelle von $n_1 + n_2$

Variablen in (**??**) oder im Fall $x_1 \in \mathbb{Z}^{n_1}$ haben wir das gemischt ganzzahlige Programm (**??**) in ein i.W. rein ganzzahliges Programm (**??**) mit einer zusätzlichen kontinuierlichen Variablen z umgeschrieben. Dennoch enthält (**??**) eine große Anzahl an Bedingungen, i.A. exponentiell viele in n. Um dieses Problem zu umgehen, lösen wir Benders' Masterproblem mittels Schnittebenenverfahren, siehe Abschnitt **??**. Wir beginnen mit einer kleinen Teilmenge von Extremstrahlen von C (möglicherweise mit der leeren Menge) und optimieren (**??**) einfach über dieser Teilmenge. Wir erhalten eine optimale Lösung x^*, z^* des relaxierten Problems und wir müssen prüfen, ob diese Lösung alle anderen Ungleichungen in (**??**) erfüllt. Dies kann mittels des folgenden linearen Programms getan werden, vgl. (**??**):

$$\begin{aligned} \min\, & v^T(b - A_1 x_1^*) + u(z^* - c_1^T x_1^*) \\ & \tbinom{u}{v} \in C\,. \end{aligned} \tag{10.30}$$

(**??**) heißt *Benders' Teilproblem*. Es ist zulässig, da $\binom{0}{0} \in C$ und (**??**) hat eine optimale Lösung mit Wert Null oder ist unbeschränkt. Im ersten Fall erfüllt x_1^*, z^* alle Ungleichungen in (**??**) und wir haben (**??**) gelöst und damit (**??**). Im anderen Fall erhalten wir einen Extremstrahl $\binom{u^*}{v^*}$ aus (**??**) mit $(v^*)^T(b - A_1 x_1^*) + u^*(z^* - c_1^T x_1^*) < 0$, der nach Reskalierung einen Schnitt für (**??**) erzeugt, der von x_1^*, z^* verletzt wird. Wir fügen diesen Schnitt zu Benders' Masterproblem (**??**) hinzu und iterieren.

10.4 Verbindungen zwischen diesen Ansätzen

Auf den ersten Blick scheinen die Lagrange-Relaxierung, Dantzig-Wolfe- und Benders Dekomposition vollständig verschiedene Relaxierungsansätze zu sein. Sie sind allerdings stark miteinander verbunden, wie wir im Folgenden kurz zeigen wollen. Betrachte noch einmal (**??**), das für ein festes $\bar{y} \leq 0$ geschrieben werden kann als

$$\begin{aligned} & \min_{x \in P^2} (c^T - \bar{y}^T A_1) x \\ = & \min_{x \in P^2} c^T x + \bar{y}^T (b_1 - A_1 x) - \bar{y}^T b_1 \\ = & L(-\bar{y}) - \bar{y}^T b \qquad , \end{aligned}$$

d. h. (**??**) und (**??**) stellen dieselben Probleme bis auf die Konstante $-\bar{y}^T b$ dar. Weiterhin kann man durch Ersetzen von P^2 durch $\text{conv}(\{v_1, \ldots, v_k\}) + \text{cone}(\{e_1, \ldots, e_l\})$ zeigen, dass (**??**) mit der rechten Seite in Satz **??** übereinstimmt und somit mit $L(\lambda^*)$. In anderen Worten, sowohl Dantzig-Wolfe und Lagrange-Relaxierung berechnen dieselbe Schranke. Die einzigen Unterschiede sind, dass für den Update die dualen Variablen, d. h. λ in der Lagrange-Relaxierung und $\bar{y}$ in Dantzig-Wolfe, im ersten Fall Subgradientmethoden wohingegen im zweiten Fall LP-Techniken angewendet werden. Andere Möglichkeiten zur Berechnung von λ^* ergeben sich aus der Bündelmethode, die auf quadratischer Optimierung [**?**] beruht und aus der Analytic-Center Schnittebenenmethode, die auf einem Innere Punkte Algorithmus beruht [**?**].

Analogerweise ist Benders' Dekomposition nichts anderes als Dantzig-Wolfe angewendet auf das Duale von (**??**). Um dies zu sehen, betrachte dessen duales lineare Programm

$$\begin{aligned} \max & -y^T b \\ & -y^T A_1 = c_1^T \\ & -y^T A_2 = c_2^T \\ & y \geq 0 \\ & : . \end{aligned} \tag{10.31}$$

Nun schreibe $\bar{P}^2 = \{y \in \mathbb{R}^{n_2} : y^T A_2 = -c_2^T, y \geq 0\}$ als $\bar{P}^2 = \text{conv}(\{v_j : j \in J\}) + \text{cone}(\{v_k : k \in K\})$, wobei K, J und v_l, $l \in K \cup J$ genau die Werte aus (**??**) sind (beachte, $\text{hog}(\bar{P}^2) = C$, vgl. DO I, d. h. C ist die Homogenisierung von $\bar{P}^2$) und schreibe (**??**) als

Referenz auf nicht vorhandenen Satz

$$\begin{aligned} \max & \sum_{j \in J} (-v_j^T b)\, \lambda_j + \sum_{k \in K} (-v_k^T b)\, \mu_k \\ & \sum_{j \in J} (-v_j^T A_1)\, \lambda_j + \sum_{k \in K} (-v_k^T A_1)\, \mu_k = c_1^T \\ & \sum_{j \in J} \lambda_j = 1 \\ & \lambda \in \mathbb{R}_+^J,\ \mu \in \mathbb{R}_+^K . \end{aligned} \tag{10.32}$$

Wir folgern nun mit den Ergebnissen aus Abschnitt **??**, dass (**??**) das Masterproblem von (**??**) ist. Dualisierung von (**??**) ergibt nun

$$
\begin{aligned}
&\min c_1^T x_1 + z \\
&v_j^T (b - A_1 x_1) \geq -z && \text{für alle } j \in J \\
&v_k^T (b - A_1 x_1) \geq 0 && \text{für alle } k \in K \,,
\end{aligned}
$$

was gleichwertig ist zu (**??**), also zu Benders' Masterproblem von (**??**). In anderen Worten: Benders' und Dantzig-Wolfe Dekomposition ergeben im Falle von linearen Programmen dieselbe Schranke, die nach den eingangs gemachten Überlegungen gleich dem Wert der Lagrange-Relaxierung (**??**) ist.

11

Enumerative Verfahren

In diesem Kapitel beschäftigen wir uns mit exakten Verfahren für diskrete Optimierungsprobleme, d. h. mit Verfahren, die den gesamten Lösungsraum absuchen und somit garantieren, eine Optimallösung zu finden, falls eine existiert. Diese Verfahren haben im schlimmsten Fall (worst case) exponentielles Laufzeitverhalten, sofern $\mathsf{P} \neq \mathsf{NP}$. Im Folgenden behandeln wir zwei klassische exakte Algorithmen, Branch-and-Bound Verfahren und Dynamische Programmierung.

11.1 Branch-and-Bound Verfahren

Der Kern von Branch-and-Bound Verfahren ist durch geschickten Einsatz von primalen und dualen Heuristiken den gesamten Suchraum so weit wie möglich einzuschränken und somit den enumerativen Teil so klein wie möglich zu halten. Die Idee ist für ein Minimierungsproblem die folgende:

Zunächst wird mit einer beliebigen Methode, z. B. aus Kapitel **??**, eine zulässige Lösung bestimmt. Außerdem wird eine untere Schranke für die Lösung bestimmt, z. B. durch LP-Relaxierung, Lagrange-Relaxierung oder Schnittebenenverfahren, vgl. Kapitel **??**. Diesen Schritt bezeichnen wir als *Bounding*.

Sei z_{Best} die kleinste bereits gefundene Lösung und d_{Best} die größte gefundene untere Schranke (bounding). Falls keine primalen Verfahren anwendbar sind, setzen wir $z_{\text{Best}} := +\infty$. Gilt $d_{\text{Best}} \geq z_{\text{Best}}$, ist nichts weiter zu tun, da z_{Best} bereits eine Optimallösung ist. Andernfalls teilen

wir das Problem in zwei Teilprobleme auf, was wir *Branching* nennen.

Die Teilprobleme entstehen aus dem urspünglichen Problem durch Einschränkung des Zulässigkeitsbereichts einer ganzzahligen Variable: Wir wählen eine solche ganzzahlige Variable $x_j \in \mathbb{Z}$ mit unterer und oberer Schranke l_j bzw. u_j, also

$$l_j \leq x_j \leq u_j \,,$$

sowie einen Parameter

$$\gamma \in \{l_j, l_{j+1}, \ldots, u_j - 1\} \,.$$

Im ersten Teilproblem beschränken wir x_j durch l_j und γ, im zweiten Problem wählen wir $\gamma+1$ und u_j als Schranken für x_j. Das Minimum der beiden besten Lösungen aus den Teilproblemen ist offensichtlich eine Optimallösung für das Gesamtproblem.

Mit den beiden Teilproblemen fährt man rekursiv fort, bis alle Teilprobleme abgearbeitet sind. Die beste so gefundene Lösung ist eine Optimallösung. Im Detail ist der Algorithmus in **??** aufgeführt.

Anmerkung 11.1. (a) Das Verfahren läuft auch ohne primale und duale Heuristiken. In diesem Fall ist es ein reines Enumerationsverfahren.

(b) Aus Lemma **??** wissen wir, dass es, falls $P_{I,p} \neq \emptyset$, einen ganzzahligen Punkt der Kodierungslänge höchstens $5n^4\varphi$ gibt (wobei φ die max. Kodierungslänge einer Ungleichung ist). Das bedeutet, dass Algorithmus **??** in endlicher Zeit einen ganzzahligen Punkt findet, falls ein solcher existiert, oder in endlicher Zeit entscheiden kann, ob es einen ganzzahligen Punkt gibt. Ist das Optimum endlich, z. B. falls $l, u \in \mathbb{Q}^n$ gilt, so wird mit Algorithmus **??** auch eine Optimallösung gefunden.

(c) Das Verfahren lässt sich gut in dem sog. *Branch-and-Bound Baum* darstellen. Die Knoten entsprechen den Teilproblemen aus L, die Kanten geben Auskunft über die Verzweigungen.
Man kann sich auch andere als binäre Verzweigungen vorstellen (z. B. auch Branchen auf Ungleichungen). Wichtig ist nur, dass die Vereinigung der Lösungsräume der Teilprobleme den Lösungsraum des Gesamtproblems enthält.

(d) Beachte, dass z_{IP} *global* für den gesamten Baum gültig ist, d_{Best} dagegen nur *lokal* für den Teilbaum.

(e) Wird in Schritt (5) zur Bestimmung einer unteren Schranke ein Schnittebenenverfahren verwendet, so spricht man von *Branch-and-Cut Verfahren*.
In diesem Fall wählt man i.d.R. in Schritt (7) eine Variable x_j, deren aktueller LP-Wert $x_j^* \notin \mathbb{Z}$ ist und setzt $\gamma = \lfloor x_j^* \rfloor$.
Branch-and-Cut Verfahren zählen derzeit zu den erfolgreichsten Methoden zur Lösung von MIPs. Nahezu jede Standardsoftware beruht auf dieser Methode.

(f) Implementierungshinweise
- Preprocessing
- Auswahl der Variablen / Knoten ([**?**])
- Fixieren durch reduzierte Kosten
- Schnittebenen - Management
- Verwenden des dualen Simplex-Algorithmus

11.2 Dynamische Programmierung

Optimallösungen für diskrete Optimierungsprobleme lassen sich häufig rekursiv aus optimalen Teillösungen zusammensetzen. Dies bezeichnen wir als *Prinzip der Optimalität* – Teillösungen von Optimallösungen sind selbst optimal. Zum Beispiel muss für einen kürzesten Weg von s nach t, der über den Knoten r geht, auch der Teilweg von s nach r und von r nach t ein kürzester Weg sein. Die dynamische Programmierung macht sich dieses Prinzip zu Nutze, indem Optimallösungen sukzessive aus Teillösungen zusammengesetzt werden.

Wir betrachten ein *dynamisches System* in seinem *zeitlichen* Verlauf von T Perioden. In jeder Periode $t \in \{0, 1, \ldots, T\}$ kann sich das System in einer Menge von Zuständen X_t befinden. Die *Zustandsvariable* x_t beschreibt den Zustand des Systems zum Zeitpunkt t und fasst alle relevanten Informationen der Vergangenheit für die künftige Optimierung zusammen.

Der Zustand x_{t+1} zum Zeitpunkt $t+1$ hängt funktional vom Zustand x_t und einer *Steuervariable* y_t ab, d. h.

$$x_{t+1} = f_t(x_t, y_t) \qquad t = 0, \ldots, T-1\,. \tag{11.1}$$

Man beachte, dass der Zustand x_{t+1} nicht von früheren Zuständen $x_l, l < t$, abhängt. Die Steuervariablen seien Elemente einer Menge S_t.

Ebenso gibt es eine zu minimierende Zielfunktion,

$$g_t(x_t, y_t) \qquad t = 0, \ldots, T\,, \tag{11.2}$$

die additiv über die Zeit ist, d. h. der Gesamtwert G ergibt sich aus

$$G = \sum_{t=0}^{T-1} g_t(x_t, y_t) + g_T(x_T)\,, \tag{11.3}$$

wobei $g_T(x_T)$ Terminalkosten am Ende des Prozesses bezeichnen.

Definition 11.2. *(Dynamisches Programm) Das folgende Problem heißt* Dynamisches Programm

$$\begin{aligned} \min \sum_{t=0}^{T-1} g_t(x_t, y_t) + g_T(x_T) & \\ x_{t+1} &= f_t(x_t, y_t) \\ x_0 &\in X_0 \\ x_t &\in X_t \qquad t = 1, \ldots, T \\ y_t &\in S_t \qquad t = 0, \ldots, T-1 \end{aligned} \tag{DP}$$

Der Wert x_0 ist der Anfangszustand *und meist vorgegeben. Eine Entscheidungsfolge $\Pi = (y_0, \ldots, y_{T-1})$ heißt* Strategie*. Eine Strategie heißt* zulässig*, falls $y_t \in S_t$ und x_t, definiert durch (**??**), aus X_t ist.*

Ein Dynamisches Programm ist also nichts anderes als ein Optimierungsproblem über einem dynamischen System. Das Prinzip der Optimalität lässt sich an einem Dynamischen Programm sehr einfach formulieren.

Prinzip der Optimalität für Dynamische Programme

Sei $\Pi^* = (y_0^*, \ldots, y_{T-1}^*)$ eine optimale Strategie für **??**. Betrachte nun das Teilproblem, in dem wir am Zeitpunkt t beginnend (d. h. x_t ist der Anfangszustand) die Kosten für die verbleibenden Perioden $l = t+1, \ldots, T$ minimieren möchten. Dann ist die Teilstrategie

$$(y_t^*, \ldots, y_{T-1}^*)$$

optimal für dieses Teilproblem. In anderen Worten, die Entscheidung y_t zum Zeitpunkt t hängt nicht von den vorhergehenden Entscheidungen $y_1, \ldots, y_{t-1}$ ab. Dieses Prinzip kann man sich nun algorithmisch zunutze machen, in

dem man von hinten beginnend ($t = T$) sich optimale Teilstrategien konstruiert und diese zu optimalen Strategien für frühere Perioden zusammensetzt:

Theorem 11.3. *Algorithmus* **??** *löst (DP).*

Beweis. Seien

$$J_t^*(x_t) = \min_{\Pi}(g_T(x_T) + \sum_{k=t}^{T-1} g_k(x_k, y_k))$$

die Werte der optimalen Teilstrategien. Wir zeigen per Induktion nach t, dass

$$J_t(x_t) = J_t^*(x_t)$$

für alle $x_t \in X_t$. Der Fall $t = T$ ist offensichtlich, nehmen wir also an, dass für ein $t < T$ und alle $x_{t+1} \in X_{t+1}$ gilt

$$J_{t+1}^*(x_{t+1}) = J_{t+1}(x_{t+1})\,.$$

Dann gilt mit $\Pi^t = (y_t, \Pi^{t+1})$ für alle x_t

$$\begin{aligned}
J_t^*(x_t) &= \min_{(y_t, \Pi^{t+1})} \left(g_t(x_t, y_t) + g_T(x_T) + \sum_{k=t+1}^{T-1} g_k(x_k, y_k) \right) \\
&= \min_{y_t \in S_t} \left(g_t(x_t, y_t) + \min_{\Pi^{t+1}} \left(g_T(x_T) + \sum_{k=t+1}^{T-1} g_k(x_k, y_k) \right) \right) \\
&= \min_{y_t \in S_t} g_t(x_t, y_t) + J_{t+1}^*(x_{t+1}) \\
&= \min_{y_t \in S_t} g_t(x_t, y_t) + J_{t+1}(x_{t+1}) \\
&= J_t(x_t)\,.
\end{aligned}$$

Anwendungen

Beispiel 11.4 (Kürzeste Wege). Betrachte einen gerichteten Graphen $D = (V, A)$ mit nichtnegativen Gewichten $c_{ij} \geq 0$. Gesucht sind für einen Knoten s die kürzesten Wege von s zu allen anderen Knoten $i \in V$.

Zur Berechnung dieser Wege betrachte die Funktion

$$\begin{aligned}
D_k(i) := &\text{ Länge eines kürzesten Weges von } s \text{ nach } i, \\
&\text{der über maximal } k \text{ Knoten läuft.}
\end{aligned}$$

Dann gilt die Rekursion

$$D_k(j) = \min\left\{D_{k-1}(j), \min_{i \in V(j)} (D_{k-1}(i) + c_{ij})\right\} \quad (11.4)$$

Berechnen wir diese Werte für k von 1 bis $n-1$ für alle $j \in V$, so enthält am Ende $D_{n-1}(j)$ den Wert eines kürzesten Weges von s nach j. Die Laufzeit des Algorithmus beträgt $O(m \cdot n)$.

Dieser Algorithmus ist ein DP Algorithmus im Sinne von Algorithmus **??**. Betrachte dazu das folgende Dynamische Programm

$$\begin{aligned} T &= n \\ X_t &= \{P_j \colon P_j \text{ ist ein kürzester Weg von } s \text{ nach } j \\ &\qquad \text{über max. } t \text{ Knoten}\} \cup \{\emptyset\} \\ S_t &= V \end{aligned}$$

$$f_t(P_j, i) = \begin{cases} P_i & \text{falls } ij \in A \\ P_j & \text{falls } i = j \\ \emptyset & \text{sonst} \end{cases}$$

$$g_t(P_j, i) = \begin{cases} c_{ij} & \text{falls } ij \in A \\ 0 & \text{falls } i = j \\ \infty & \text{sonst} \end{cases}$$

Damit liest sich Algorithmus **??** wie folgt

$$J_0(P_i) = \begin{cases} c_{si} & \text{falls } si \in A, \\ +\infty & \text{sonst} \end{cases}$$
$$J_0(\emptyset) = +\infty$$

(Setze $P_i = \{si\}$, falls $J_0(P_i) = c_{si}, P_i = \emptyset$, sonst.)

$$\begin{aligned} J_t(P_j) &= \min_{i \in V} \left(g_t(P_j, i) + J_{t-1}(f_t(P_j, i))\right) \quad (11.5) \\ &= \min\left(\min_{i \in V(\delta(j))} c_{ij} + J_{t-1}(P_i), 0 + J_{t-1}(P_j), \infty\right). \end{aligned}$$

Setze

$$P_j = \begin{cases} P_i \cup \{ij\} & \text{falls } J_t(P_i) = c_{ij} + J_{t-1}(P_i) \\ P_j, & \text{falls } J_t(P_j) = J_{t-1}(P_j) \end{cases},$$

was genau (**??**) entspricht. Beachte auch, dass $P_i \cup \{ij\} = P_j$ wieder einen Weg ergibt, da $c_{ij} \geq 0$.

Beispiel 11.5 (Das 0/1 *Rucksackproblem).*
Betrachte folgendes 0/1 Rucksackproblem:

$$\max \sum_{i=1}^{n} c_i y_i$$
$$\sum_{i=1}^{n} a_i y_i \leq b$$
$$y_i \in \{0,1\} \qquad i = 1, \ldots, n$$

Das 0/1 Rucksackproblem kann auch als Dynamisches Programm aufgefasst werden. Sei dazu

$T = n + 1$
(für jede Variable eine Periode plus Slackvariable)
$x_j =$ Restkapazität in Periode j (Zustandsvariable)
$y_j =$ Steuervariablen

Übergangsfunktion f aus (**??**) ist

$$x_{j+1} = f_j(x_j, y_j) = x_j - a_j y_j$$

Für die Zielfunktion (**??**) gilt

$$g_j(x_j, y_j) = c_j y_j$$

Darüber hinaus haben wir

$$X_j = \{0, 1, \ldots, b\} \qquad j = 1, \ldots, n+1$$
$$X_1 = \{b\},$$

sowie

$$S_j = \{0,1\} \qquad \text{falls } x_j \geq a_j$$
$$S_j = \{0\} \qquad \text{sonst.}$$

Beachte, dass S_j vom Zustand x_{j-1} der Vorperiode abhängt, was aber der Gültigkeit der entwickelnden Theorie und Verfahren keinen Abbruch tut.

Das 0/1 Rucksackproblem liest sich nun als dynamisches Programm wie folgt:

$$\max \sum_{j=1}^{T} g_j(x_j, y_j) = \sum_{j=1}^{n} c_j y_j$$
$$x_{j+1} = x_j - a_j y_j \qquad j = 1, \ldots, n$$
$$x_1 = b$$
$$x_j \in X_j$$
$$y_j \in S_j$$

Algorithmus **??** liest sich entsprechend

$$J_{n+1}(d) = 0 \quad \text{für alle } d = 0, 1, \ldots, b \tag{11.6}$$

$$\begin{aligned} J_i(d) &= \max_{y_i \in S_i} (g_i(d, y_j) + J_{i+1}(f_i(d, y_i))) \qquad (11.7) \\ &= \begin{cases} J_{i+1}(d) & \text{falls } d < a_i \\ \max\{J_{i+1}(d), c_i + J_{i+1}(d - a_i)\} & \text{falls } d \geq a_i, \end{cases} \end{aligned}$$

für $i = n, n-1, \ldots, 1$. Die Optimallösung ist dann $\max_d J_1(d)$.

Beachte Algorithmus **??** hat Laufzeit $O(n \cdot b)$, ist also pseudo-polynomial und für kleine rechte Seiten durchaus verwendbar.

12

Approximations-Algorithmen

In diesem Kapitel beschäftigen wir uns mit approximativen Algorithmen zur Bestimmung von Lösungen mit beweisbarer Qualität für NP-harte Optimierungsprobleme.

Definition 12.1. *[Approximations-Algorithmen]*
Sei Π ein diskretes Optimierungsproblem und A ein Algorithmus zur Lösung von Π. Bezeichne $c_{opt}(I)$ den Optimalwert für $I \in \Pi$ und $c_A(I)$ den Wert, den Algorithmus A liefert. Der Algorithmus A heißt

(a) ϵ-Approximations-Algorithmus,
falls A polynomiale Laufzeit hat und $c_A(I)$ höchstens ϵ-mal schlechter als $c_{opt}(I)$ ist für alle $I \in \Pi$. Das bedeutet für Minimierungsprobleme:

$$c_A \leq \epsilon \, c_{opt} \text{ mit } \epsilon \geq 1$$

und für Maximierungsprobleme:

$$c_A \geq \epsilon \, c_{opt} \text{ mit } \epsilon \leq 1 \, .$$

ϵ heißt die Gütegarantie *von A.*

(b) polynomiales Approximationsschema (PAS),
falls für jedes feste $\epsilon > 0$ A in polynomialer Laufzeit (polynomial in $\langle I \rangle$ für $I \in \Pi$) eine Gütegarantie von $1 + \epsilon$ (bzw. $1 - \epsilon$) hat.

(c) vollpolynomiales Approximationsschema (FPAS),
falls die Laufzeit polynomial in $\langle I \rangle$ und $\frac{1}{\epsilon}$ ist.

Der folgende Satz zeigt, dass ein FPAS das Beste ist, was man für ein NP-schweres Problem erreichen kann.

Theorem 12.2. *Sei Π ein diskretes Optimierungsproblem. Gibt es für Π ein FPAS, das auch polynomial in $\langle \epsilon \rangle$ ist, so gibt es einen polynomialen Algorithmus für Π.*

Beweis. Wir zeigen die Aussage für Minimierungsprobleme, für Maximierungsprobleme geht es analog.

Sei A ein FPAS für Π, das auch polynomial in $\langle\epsilon\rangle$ ist. O.B.d.A. sei die Zielfunktion c ganzzahlig. Wir konstruieren einen polynomialen Algorithmus wie folgt:

1. Setze $\epsilon = \frac{1}{2}$ und wende A auf $I \in \Pi$ an. Wir erhalten einen Wert $c_{A_\epsilon}(I)$.
2. Setze $\delta = \frac{1}{c_{A_\epsilon}(I)+1}$.
3. Wende A auf I mit Approximationsgüte δ an. Wir erhalten einen Lösungswert $c_{A_\delta}(I)$.

Obiger Algorithmus zur Berechnung von $c_{A_\delta}(I)$ ist polynomial. Wir zeigen nun noch, dass $c_{opt}(I) = c_{A_\delta}(I)$.

$$\begin{aligned} c_{opt}(I) \leq c_{A_\delta}(I) &\leq (1+\delta)c_{opt}(I) \\ &= (1 + \frac{1}{c_{A_\epsilon}(I)+1})c_{opt}(I) \\ &< (1 + \frac{1}{c_{opt}(I)})c_{opt}(I) = c_{opt}(I) + 1\ . \end{aligned}$$

Im vorletzten Schritt wurde $c_{opt}(I) < c_{A_\epsilon}(I) + 1$ benutzt. Da c ganzzahlig ist, folgt die Behauptung.

Im Folgenden werden wir uns Approximations-Algorithmen bzw. FPAS für verschiedene diskrete Optimierungsprobleme anschauen. Wir werden dabei einige interessante Algorithmenideen kennenlernen, die sich auf andere Probleme übertragen lassen. Wir beginnen mit einem Beispiel aus einem großen Anwendungsfeld für Approximations-Algorithmen, den Scheduling-Problemen.

12.1 Randomisierte Rundeverfahren

Die Idee vieler Approximations-Algorithmen ist das Potential der LP-Relaxierungen zu nutzen und diese geschickt zu runden. Der Erfolg dieser Verfahren hängt von dem Geschick ab, die gerundete Lösung geeignet abschätzen zu können. Hier helfen häufig randomisierte Techniken. Wir erläutern die Ideen an einem Beispiel aus dem Scheduling-Bereich, einem der klassischen Bereiche für Approximationsalgorithmen.

Beispiel 12.3. Seien $M_1, \ldots, M_m$ mit $m \geq 2$ identische Maschinen und $J_1, \ldots, J_n$ Jobs. Jeder Job J_i hat Strafkosten e_i bei Ablehnung und Produktionszeiten p_{ij} auf

Maschine M_j. Jeder Job kann beliebig unterbrochen und weitergeführt werden. Für jeden Job soll entschieden werden, ob er angenommen wird oder nicht und die angenommenen Jobs sollen so auf die Maschinen verteilt werden, dass der Makespan (Produktionszeit der langsamsten Maschine) minimiert wird.

Formulierung als MIP:

$$\begin{aligned} x_{ij} &= \text{Anteil von Job } j \text{ auf Maschine } i \\ y_j &= \begin{cases} 1 & \text{falls Job } i \text{ angenommen wird} \\ 0 & \text{sonst.} \end{cases} \\ T &= \text{Makespan} \end{aligned}.$$

$$\begin{aligned} \min T + \sum_{j=1}^{n} (1 - y_j) e_j & \\ \sum_{j=1}^{n} p_{ij} x_{ij} &\leq T \qquad i = 1, \ldots, m \\ \sum_{i=1}^{m} p_{ij} x_{ij} &\leq T \qquad j = 1, \ldots, n \\ \sum_{i=1}^{m} x_{ij} &= y_j \qquad j = 1, \ldots, n \\ x_{ij} &\geq 0 \qquad i = 1, \ldots, m,\ j = 1, \ldots, n \\ y_j &\in \{0, 1\} \quad j = 1, \ldots, n\,. \end{aligned}$$

Beachte, dass obiges Modell das Scheduling-Problem tatsächlich korrekt beschreibt, da die Jobs beliebig unterbrechbar sind. Man kann zeigen, dass obiges Schedulingproblem NP-schwer ist.

Theorem 12.4. *Das Scheduling-Problem aus Beispiel* **??** *ist* NP*-schwer.*

Beweis. Zum Beweis siehe [**?**].

Betrachte folgenden Algorithmus zur Lösung des Scheduling-Problems:

Theorem 12.5. *Algorithmus* **??** *ist ein* 2*-Approximations-Algorithmus.*

Beweis. Es gilt (vgl. Schritt (4) und (5))

Algorithmus 12.1 : Deterministischer Rundealgorithmus

Bestimme Lösung x^*, y^* der LP-Relaxierung des Scheduling-Problems;
Wähle α mit $0 < \alpha < 1$;
foreach *Job* j **do**
  if $y_j^* \leq \alpha$ **then**
    setze $y_j = x_{ij} = 0$ für alle i;
  else
    setze $y_j = 1$ und $x_{ij} = \frac{x_{ij}^*}{y_j^*}$ für alle i;
  end
end
Gib y, x aus;

$$(1 - y_j) \leq \frac{1}{1-\alpha}\,(1 - y_j^*)$$

und

$$x_{ij} \leq \frac{1}{\alpha}\, x_{ij}^*$$

und damit

$$T + \sum_{j=1}^{n}(1 - y_j)e_j \leq \frac{1}{\alpha}\, T^* + \frac{1}{1-\alpha}\sum_{j=1}^{n}(1 - y_j^*)e_j \quad (12.1)$$

$$\leq \max\{\frac{1}{\alpha}, \frac{1}{1-\alpha}\}(T^* + \sum_{j=1}^{n}(1 - y_j^*)e_j) \quad (12.2)$$

$$\leq \max\{\frac{1}{\alpha}, \frac{1}{1-\alpha}\}c_{opt}\,. \quad (12.3)$$

Für $\alpha = \frac{1}{2}$ erhält man die Behauptung.

Durch Randomisieren kann man die Gütegarantie sogar noch auf $\frac{e}{e-1} \approx 1.58$ verbessern:

Algorithmus 12.2 : Randomisierter Rundealgorithmus

Wie in **??**;
foreach $\alpha \in [\frac{1}{e}, 1] \cap \{y_1^*, \ldots, y_n^*\}$ **do**
  führe Schritte (3)-(5) in **??** aus.
end
Gib die beste der n gefundenen Lösungen aus.;

Theorem 12.6. *Algorithmus* **??** *ist* 1.58 *approximativ.*

Beweis. Betrachte zunächst eine gleichverteilte Zufallsvariable α auf dem Intervall $[\frac{1}{e}, 1]$. Dann gilt

$$E(T) = \frac{e}{e-1}\int_{\frac{1}{e}}^{1} T d\alpha \leq \frac{e}{e-1}\int_{\frac{1}{e}}^{1} \frac{1}{\alpha} T^* d\alpha = \frac{e}{e-1} T^*.$$

Ferner haben wir

$$\begin{aligned} E(\sum_{j=1}^{n}(1-y_j)e_j) &= \sum_{j=1}^{n} e_j \ \text{Prob}(y_j^* \leq \alpha) \\ &= \sum_{j=1}^{n} e_j \int_{\max\{\frac{1}{e}, y_j^*\}}^{1} \frac{e}{e-1} d\alpha \\ &\leq \sum_{j=1}^{n} e_j \int_{y_j^*}^{1} \frac{e}{e-1} d\alpha \\ &= \sum_{j=1}^{n} \frac{e}{e-1} \cdot e_j(1-y_j^*) \\ &= \frac{e}{e-1}\sum_{j=1}^{n} e_j(1-y_j^*). \end{aligned}$$

D. h.

$$E(T + \sum_{j=1}^{n} e_j(1-y_j)) \leq \frac{e}{e-1} c_{opt}$$

und damit haben wir einen randomisierten $\frac{e}{e-1}$ Approximations-Algorithmus.

Zur Derandomisierung beachte, dass verschiedene Lösungen nur dann auftreten, falls $\alpha \in \{y_1^*, \ldots, y_n^*\}$ und damit genügt es für $\alpha \in [\frac{1}{e}, 1] \cap \{y_1^*, \ldots, y_n^*\}$ gerundete Lösungen zu berechnen.

12.2 Ein FPAS für das Rucksackproblem

In diesem Abschnitt wollen wir uns ein FPAS für das Rucksackproblem betrachten. Die Idee des Algorithmus ist es, sich auf die Gegenstände zu konzentrieren, die den maximalen Gewinn versprechen und dieses Teilproblem annähernd optimal zu lösen. Bei geeigneter Definition von "groß" und "annähernd" kann dieses Problem mittels Dynamischer Programmierung optimal gelöst werden. Der Rest wird dann per Greedy-Algorithmus aufgefüllt:

Algorithmus 12.3 : (FPAS für das 0/1 Rucksackproblem)

Input : $a_j, c_j \in \mathbb{Z}_+, j = 1, \ldots, n$. O.B.d.A $\frac{c_1}{a_1} \geq \frac{c_2}{a_2} \geq \ldots \geq \frac{c_n}{a_n}$, $b \in \mathbb{Z}_+$ und zwei Parameter s, t

Output : Approx. Lösung des 0/1 Rucksackproblems (geeignete Wahl von s und t liefern Gütegarantie ϵ, siehe Satz **??**).

`/* Abschätzung der Optimallösung */`

Sei k maximaler Index mit $\sum_{j=1}^{k} a_j \leq b$.;

Setze $c_{est} = \sum_{j=1}^{k+1} c_j$;

`/* Zerlegung der Indexmenge */`

$L := \{j \in \{1, \ldots, n\} \mid c_j \geq t\}$(large indices);

$S := \{j \in \{1, \ldots, n\} \mid c_j < t\}$(small indices);

`/* Löse folgende spezielle Rucksack-Probleme für` $d = 0, 1, \ldots, \lfloor \frac{c_{est}}{s} \rfloor$: `*/`

`/*`

$$\min \sum_{j \in L} a_j x_j$$
$$\sum_{j \in L} \lfloor \frac{c_j}{s} \rfloor x_j = d \qquad (GKP_d)$$
$$x_j \in \{0,1\}, \qquad j \in L$$

`*/`

for $d = 0, \ldots, \lfloor \frac{c_{est}}{s} \rfloor$ **do**

Sei $x_j^d, j \in L$ die Optimallösung von (GKP_d);

if $\sum_{j \in L} a_j x_j^d \leq b$ **then**

Wende Gewichtsdichten-Greedy an auf

$$\max \sum_{j \in S} c_j x_j$$
$$\sum_{j \in S} a_j x_j \leq b - \sum_{j \in L} a_j x_j^d \ (=: b_d) \qquad (KP_d)$$
$$x_j \in \{0,1\}, \qquad j \in S$$

end

Sei $x_j^d, j \in S$, die Greedy-Lösung. Dann ist $x_j^d, j = 1, \ldots, n$ eine Lösung des 0/1 Rucksackproblems.

end

Gib die beste der maximal $(\lfloor \frac{c_{est}}{s} \rfloor + 1)$ gefundenen Lösungen aus.;

Theorem 12.7. *Sei c_{IK} der durch Algorithmus* **??** *gefundene Lösungswert. Dann gilt*

$$c_{IK} \geq c_{opt} - (\frac{s}{t}c_{opt} + t).$$

Beweis. Sei $x_1^*, \ldots, x_n^*$ eine Optimallösung des 0/1 Rucksackproblems. Setze $d = \sum_{j \in L} \lfloor \frac{c_j}{s} \rfloor x_j^*$. Dann gilt

$$d \leq \lfloor \frac{1}{s} \sum_{j \in L} c_j x_j^* \rfloor \leq \lfloor \frac{1}{s} c_{opt} \rfloor \leq \lfloor \frac{1}{s} c_{est} \rfloor,$$

wobei zu beachten ist, dass c_{est} größer oder gleich als der LP-Wert ist. Also ist in Algorithmus **??** eine Kandidatenlösung $\bar{x}_1, \ldots, \bar{x}_n$ mit

$$\sum_{j \in L} \lfloor \frac{c_j}{s} \rfloor \bar{x}_j = d$$

betrachtet worden, da in diesem Fall die Lösungsmenge nicht leer ist.

Dann gilt

$$c_{IK} \geq \sum_{j=1}^{n} c_j \bar{x}_j = c_{opt} - (\sum_{j \in L} c_j x_j^* - \sum_{j \in L} c_j \bar{x}_j) - (\sum_{j \in S} c_j x_j^* - \sum_{j \in S} c_j \bar{x}_j). \tag{A}$$

Wir erhalten

$$\sum_{j \in L} c_j x_j^* - \sum_{j \in L} c_j \bar{x}_j \leq s \underbrace{\sum_{j \in L} \lfloor \frac{c_j}{s} \rfloor x_j^*}_{=d} + \sum_{j \in L} (c_j - s \lfloor \frac{c_j}{s} \rfloor) x_j^* - s \underbrace{\sum_{j \in L} \lfloor \frac{c_j}{s} \rfloor \bar{x}_j}_{=d} \tag{B}$$

$$= \sum_{j \in L} \underbrace{(c_j - s \lfloor \frac{c_j}{s} \rfloor)}_{\leq s} x_j^* \leq s \sum_{j \in L} x_j^* \leq \frac{s}{t} \sum_{j \in L} c_j x_j^* \leq \frac{s}{t}\, c_{opt}.$$

Um den zweiten Term in (A) abzuschätzen, benötigen wir folgende Beziehung

$$c_{Greedy} \geq c_{opt} - \max\{c_j \mid j = 1, \ldots, n\}. \qquad (*)$$

Gilt $\sum_{j=1}^{n} a_j \leq b$, so ist dies offensichtlich richtig. Andernfalls gilt

$$0 \leq b - \sum_{j=1}^{k} a_j < a_{k+1}$$

und damit

$$c_{opt} \leq \sum_{j=1}^{k} c_j + c_{k+1} \frac{b - \sum_{j=1}^{n} a_j}{a_{k+1}} < c_{Greedy} + c_{k+1}.$$

Mit $(*)$ folgt

$$\underbrace{c_{Greedy}}_{(KP_d)} = \sum_{j \in S} c_j \bar{x}_j > \underbrace{c_{opt}}_{(KP_d)} - \max\{c_j \mid j \in S\} \geq \underbrace{c_{opt}}_{(KP_d)} - t.$$

Es folgt daraus

$$t > \underbrace{c_{opt}}_{(KP_d)} - \underbrace{c_{Greedy}}_{(KP_d)} \geq \sum_{j \in S} c_j x_j^* - \sum_{j \in S} c_j \bar{x}_j. \qquad \text{(C)}$$

(A), (B) und (C) zusammen ergeben

$$c_{IK} \geq c_{opt} - (\frac{s}{t} c_{opt} + t).$$

Theorem 12.8. *Sei $\epsilon > 0$. Setze $s = (\frac{\epsilon}{3})^2 c_{est}$ und $t = \frac{\epsilon}{3} c_{est}$. Dann gilt*

(a) Die Laufzeit von Algorithmus **??** *ist $O(n \log n) + O(n(\frac{1}{\epsilon})^2)$.*
(b) Für den Wert der Optimallösung gilt

$$c_{IK} \geq (1 - \epsilon) c_{opt}.$$

Also ist Algorithmus **??** *ein FPAS für das* 0/1 *Rucksackproblem.*

Beweis. Ad (a): Schritt (1) und (6) (einmalig) benötigen zur Sortierung $O(n \log n)$. Berechnung von s und t ist $O(\log \epsilon + n)$. Lösung von Schritt (3) benötigt $O(n \lfloor \frac{c_{est}}{s} \rfloor) = O(\frac{n}{\epsilon^2})$, siehe Kapitel **??**. In Schritt (4) wird $O(\frac{1}{\epsilon^2})$ mal der Greedy-Algorithmus mit Laufzeit $O(n)$ angewendet, macht $O(\frac{n}{\epsilon^2})$. Insgesamt beläuft sich die Laufzeit also auf $O(n \log n + \frac{n}{\epsilon^2})$.

Ad (b): Zunächst gilt $\sum_{j=1}^{k} c_j \leq c_{opt}$ und $c_{k+1} \leq c_{opt}$ und damit $c_{est} \leq 2c_{opt}$ sowie $t \leq \frac{2\epsilon}{3} c_{opt}$. Mit Satz **??** erhalten wir

$$c_{IK} \geq c_{opt} - (\frac{s}{t} c_{opt} + t) \geq c_{opt} - (\frac{\epsilon}{3} c_{opt} + \frac{2\epsilon}{3} c_{opt}) = (1 - \epsilon) c_{opt}.$$

12.3 Primal-Dual Verfahren

Die Idee dieser Verfahren ist, ausgehend von einer zulässigen Lösung des Dualen der LP-Relaxierung eines Ganzzahligen Programms eine zulässige primale Lösung zu generieren. Wir wissen aus DO I, ein Paar von Lösungen

x und y ist optimal für das primale bzw. duale lineare Programm genau dann, wenn die Bedingungen des Satzes vom schwachen komplementären Schlupfes erfüllt sind. Wir wissen, dass für lineare Programme, welche bekanntlich polynomial lösbar sind, diese Bedingungen auch tatsächlich erfüllt werden können. Für ganzzahlige Programme ist dies nicht zu erwarten, sofern $\mathsf{P} \neq \mathsf{NP}$. Die Idee ist nun die komplementären Schlupfbedingungen zu relaxieren und sich auf nur eine der beiden Bedingungen zu konzentrieren. Wir werden das Verfahren an einer sehr allgemeinen Klasse von Problemen, dem sog. Hitting Set Problem, erklären. Wir werden gleich sehen, dass dieses Problem viele bekannte polynomiale als auch NP-schwere Probleme beinhaltet.

Definition 12.9. Hitting Set Problem

Gegeben ist eine Grundmenge E und eine Liste von Teilmengen $T_1, \ldots, T_p \subseteq E$ sowie nicht-negative Gewichte c_e, $e \in E$. Finde eine Teilmenge $A \subseteq E$ minimalen Gewichtes, so dass $A \cap T_i \neq \emptyset$ für $i = 1, \ldots, p$ (d. h. A „hits“ (trifft) jedes T_i).

Das Hitting Set Problem beinhaltet unter anderem folgende Probleme

- Kürzeste-Wege-Problem in ungerichteten Graphen
- Knotenüberdeckungs-Problem
- Minimal-Aufspannende-Baum Problem
- Minimales Arboreszenz-Problem
- Steinerbaum-Problem
- Set Covering Problem (ist sogar äquivalent dazu)
- Survivable Network Design Problem
- Perfektes Matching Problem

Das Hitting Set Problem kann als ganzzahliges Programm wie folgt modelliert werden:

$$\begin{aligned} \min \sum_{e \in E} c_e x_e & \\ \sum_{e \in T_i} x_e \geq 1 & \qquad \text{für } i = 1, \ldots, p \\ x_e \in \{0,1\} & \qquad \text{für } e \in E\,. \end{aligned} \tag{12.4}$$

Die LP-Relaxierung von (**??**) lautet

$$\begin{aligned} \min & \sum_{e \in E} c_e x_e \\ & \sum_{e \in T_i} x_e \geq 1 \qquad \text{für } i = 1, \ldots, p \\ & x_e \geq 0 \text{für } e \in E, \end{aligned} \tag{12.5}$$

und das duale lineare Programm zu (**??**) ist

$$\begin{aligned} \min & \sum_{i=1}^{p} y_i \\ & \sum_{i:\ e \in T_i} y_i \leq c_e \qquad \text{für } e \in E, \\ & y_i \geq 0 \qquad \text{für } i = 1, \ldots, n\,. \end{aligned}$$

Sei x der Inzidenzvektor einer Menge $A \subseteq E$ und y eine dual zulässige Lösung, so lauten die komplementären Schlupf-Bedingungen:

$$e \in A \Longrightarrow \sum_{i:\ e \in T_i} y_i = c_e \tag{12.6}$$

$$y_i > 0 \Longrightarrow |A \cap T_i| = 1 \tag{12.7}$$

(**??**) werden auch die *primalen* und (**??**) die *dualen komplementäre Schlupf-Bedingungen* genannt.

Die Idee ist nun, sich auf die primalen Bedingungen zu konzentrieren und die dualen zu relaxieren.

Betrachte eine zulässige duale Lösung y (diese ist einfach zu finden, z. B. $y = 0$, da $c \geq 0$) und setze

$$A := \{e \in E \mid \sum_{i:\ e \in T_i} y_i = c_e\}.$$

Ist A zulässig, so haben wir ein A gefunden, das (**??**) erfüllt. Ist A unzulässig, so kann es keine zulässige Lösung geben, die (**??**) erfüllt. In diesem Fall kann jedoch y erhöht werden.

Da A unzulässig ist, existiert ein T_k mit $A \cap T_k = \emptyset$. Wir nennen T_k *verletzt*. Erhöhen wir y_k, so verbessert sich die duale Zielfunktion. Um duale Zulässigkeit beizubehalten, kann y_k maximal auf

$$y_k = \min_{e \in T_k} \left\{ c_e - \sum_{i \neq k:\ e \in T_i} y_i \right\} \tag{12.8}$$

erhöht werden. Beachte $y_k > 0$, da $A \cap T_k = \emptyset$ und somit $c_e - \sum_{i \neq k:\ e \in T_i} y_i \geq c_e - \sum_{i:\ e \in T_i} y_i > 0$ für alle $e \in T_k$.

Für diesen Wert y_k gibt es mindestens ein $e \in E \setminus A$, nämlich argmin in (**??**), das zu A hinzugefügt werden kann. Dieses Verfahren wird nun wiederholt bis A zulässig ist.

Algorithmus 12.4 : Primal-Dual Algorithmus (Grundversion)

Setze $y := 0$;
Setze $A := \emptyset$;
while $\exists k:\ A \cap T_k = \emptyset$ **do**
    Erhöhe y_k bis $\exists e \in T_k : \sum_{i:\ e \in T_i} y_i = c_e$;
    Setze $A \leftarrow A \cup \{e\}$;
end
Gib A (und y) aus;

Um die Laufzeit von Algorithmus **??** abzuschätzen, beachte, dass die Schritte (3) bis (6) höchstens $|E|$-mal durchlaufen werden, da $A \subseteq E$, und dass Schritt (4) maximal $O(|E|)$ Zeit benötigt. Die Gesamtlaufzeit hängt also von Schritt (3) ab. Nehmen wir an, wir haben ein Orakel, das uns, gegeben eine Menge $A \subseteq E$, entscheidet, ob A zulässig ist, und falls nicht, ein T_k mit $A \cap T_k = \emptyset$ zurückgibt, so haben wir einen orakel-polynomialen Algorithmus.

Im Folgenden nehmen wir an, dass dieses Orakel uns immer eine (bzgl. Inklusion) minimale verletzte Menge zurückgibt. Im Falle von Netzwerkentwurfsproblemen, in denen die $T_i = \delta(S_i)$ für ein $S_i \subset V$ sind, beziehen wir die Minimalität auf die zugehörige Knotenmenge S_i. Wir nennen diese Regel die *Minimale Verletztheitsregel*. Wir werden sehen, dass diese Mengen in den nun folgenden Anwendungen einfach zu bestimmen sind und wir somit einen polynomialen Algorithmus **??** erhalten.

Um die Güte der Lösung A in Schritt (7) von Algorithmus **??** abschätzen zu können, beobachten wir, dass aufgrund von (**??**) gilt:

$$c(A) = \sum_{e \in A} c_e = \sum_{e \in A} \sum_{i:\ e \in T_i} y_i \qquad (12.9)$$
$$= \sum_{i=1}^{p} |A \cap T_i|\, y_i.$$

Gelingt es uns nun ein α zu finden, so dass

$$|A \cap T_i| \leq \alpha \text{ für alle } i = 1, \ldots, p \text{ mit } y_i > 0, \qquad (12.10)$$

so ist Algorithmus **??** ein α-Approximationsalgorithmus.

Ein triviales α ist

$$\alpha = \max_{i=1,\ldots,p} |T_i|. \qquad (12.11)$$

Diese Beobachtung liefert uns zum Beispiel für das Knotenüberdeckungs-Problem einen 2-Approximationsalgorithmus, da $|T_i| = 2$ für alle $i = 1, \ldots, p$.

Theorem 12.10. *Algorithmus* **??** *ist ein 2-Approximationsalgorithmus für das Knotenüberdeckungs-Problem.*

Im Allgemeinen ist jedoch (**??**) keine Konstante und (**??**) auch nicht. Betrachte zum Beispiel Algorithmus **??** für das Kürzeste-Wege-Problem auf einem Stern, vgl. Abbildung **??**.

Abb. 12.1. Beispiel für nicht konstantes α

Hier ist es möglich, dass A am Ende von Algorithmus **??** alle Kanten beinhaltet, jedoch die Kante st genügen würde.

Wir verfeinern daher unseren Algorithmus und eliminieren am Ende alle unnötigen Kanten:

Schritte (9) bis (11) nennen wir *Rückwärtslöschung.*

Durch diese Erweiterung erhalten wir eine neue interessante Abschätzung für $|A \cap T_i|$ für alle $i = 1, \ldots, p$.

Definition 12.11. *Sei $A \subseteq E$ unzulässig. Dann heißt $B \subseteq E$* minimale Erweiterung von *A, falls gilt:*

(a) B ist zulässig.
(b) $A \subseteq B$.

Algorithmus 12.5 : Verbesserter Primal-Dual Algorithmus

Setze $y = 0$;
Setze $A = \emptyset$;
Setze $l = 0$;
while $\exists k : A \cap T_k = \emptyset$ **do**
  Setze $l = l + 1$;
  Erhöhe y_k bis $\exists e_l \in T_k : \sum_{i:\, e \in T_i} y_i = c_{e_l}$;
  Setze $A = A \cup \{e_l\}$;
end
for $j = l, l-1, \ldots, 1$ **do**
  if $A \setminus \{e_j\}$ *zulässig,* **then** setze $A = A \setminus \{e_j\}$
end
Gib A (und y) aus;

(c) Für alle $e \in B \setminus A$ gilt: $B \setminus \{e\}$ ist unzulässig.

In Abbildung **??** ist eine minimale Augmentierung für das Kürzeste-Wege-Problem angedeutet, die gestrichelten Kanten sind die Kanten in A, B enthält zusätzliche alle durchgezogenen Kanten.

Abb. 12.2. Beispiel einer minimalen Augmentierung

Betrachte nun noch einmal Algorithmus **??** und sei A_f die Lösung, die am Schluss in Schritt (12) ausgegeben wird. Betrachte in Schritt (10) die Situation, dass e_j nicht gelöscht werden kann. Dann ist $A = \{e_1, \ldots, e_{j-1}\}$ unzulässig und $B = A_f \cup \{e_1, \ldots, e_{j-1}\}$ eine minimale Augmentierung von $\{e_1, \ldots, e_{j-1}\}$. Ferner gilt:

$$|A_f \cap T_i| \leq |B \cap T_i| \quad \text{für alle } i = 1, \ldots, p.$$

Letzteres gilt sicher auch, wenn wir das Maximum über alle möglichen minimalen Augmentierungen nehmen. Sei

$$\beta = \max_{A \subset E \text{unzulässig}} \; \max_{B:\, \text{min.Augm. von } A} |B \cap T(A)|, \tag{12.12}$$

wobei $T(A)$ die von Algorithmus **??** in Schritt (4) gewählte Menge T_k bezeichne, wenn dem Orakel als Input die Menge A gegeben wird.

Damit erhalten wir

Theorem 12.12. *Algorithmus* **??** *ist ein β-Approximationsalgorithmus für das Hitting Set Problem.*

Korollar 12.13. *Für das Kürzeste-Wege-Problem gilt $\beta = 1$. D. h. Algorithmus* **??** *löst das Kürzeste-Wege-Problem in polynomialer Zeit.*

Beweis. Sei A eine beliebige unzulässige Menge. D. h. A zerfällt in mindestens 2 Komponenten, eine davon, sagen wir T_s, enthält den Startknoten s, eine andere den Endknoten t, die wir mit T_t bezeichnen. Aufgrund unserer minimalen Verletztheitsregel wird in Schritt (4) T_s vom Orakel zurückgegeben. Offensichtlich gilt für jede minimale Augmentierung B von A, dass $|B \cap \delta(T_s)| = 1$ gilt. Daraus folgt $\beta = 1$.

Korollar 12.14. *Für das Minimalkosten-Arboreszenz-Problem gilt $\beta = 1$. D. h. Algorithmus* **??** *löst das Minimalkosten-Arboreszenz-Problem in polynomialer Zeit.*

Beweis. Zur Wiederholung, das *Minimalkosten-Arboreszenz-Problem* ist das Problem, gegeben ein gerichteter Graph $G = (V, E)$ mit nicht-negativen Bogengewichten und ein bestimmter Knoten $r \in V$, *Wurzel* genannt, finde einen aufspannenden Baum mit minimalen Kosten, so dass es einen gerichteten Weg von r zu allen anderen Knoten gibt.

Sei nun A eine beliebige Teilmenge von E. Das Verletztheitsorakel mit der minimalen Verletztheitsregel in Schritt (4) von Algorithmus **??** kann so implementiert werden, dass zunächst alle stark zusammenhängenden Komponenten bestimmt werden und dann überprüft wird, ob es eine Komponente S gibt mit $r \notin S$ und $\delta^-(S) \cap A = \emptyset$. Gibt es kein solches S, so enthält A offensichtlich eine r-Arboreszenz.

Andernfalls erhalten wir in Schritt (4) eine stark zusammenhängende Komponente S mit $r \notin S$ und $\delta^-(S) \cap A = \emptyset$. Offensichtlich enthält jede minimale Augmentierung B von A genau einen Bogen aus $\delta^-(S)$, da dadurch alle Knoten in S erreichbar sind. Damit folgt $\beta = 1$.

Wir wollen uns zum Abschluss dieses Kapitels noch eine weitere Verfeinerung von Algorithmus **??** bzw. Algorithmus **??** ansehen. Für Graphenprobleme, die mehrere Ziel- und/oder Quellknoten haben (wie zum Beispiel für das Matching-Problem oder das Steinerbaum-Problem) reichen die bisherigen Argumente nicht aus, da eine minimale Augmentierung immer eine nicht-konstante Anzahl an Kanten in den jeweiligen Schnitten enthalten kann.

Die Idee ist nun, sich nicht nur auf eine verletzte Menge in Schritt (4) zu konzentrieren, sondern auf alle gleichzeitig und gleichzeitig alle zugehörigen dualen Variablen zu erhöhen. Die konstante Approximation erhält man, wie wir später sehen werden, schließlich durch Durchschnittsbildung. Zunächst der Algorithmus.

Algorithmus 12.6 : Simultaner Primal-Dual Algorithmus

```
y = 0;
Setze A = ∅;
Setze l = 0;
while A nicht zulässig do
    Setze l = l + 1;
    V(A) = Menge aller minimalen verletzten Mengen S von A;
    Erhöhe y_S gleichmäßig für alle S ∈ V(A) bis
    ∃e_l ∉ A : Σ_{i: e∈T_i} y_i = c_{e_l};
    Setze A = A ∪ {e_l};
end
for j = l, l − 1, ..., 1 do
    if A \ {e_j} zulässig then setze A = A \ {e_j}
end
Gib A (und y) aus;
```

Zur Abschätzung der Güte von Algorithmus **??** bezeichne in Iteration j mit ϵ_j die Erhöhung aller dualen Variablen y_i mit $T_i \in \mathcal{V}_j$, wobei $\mathcal{V}_j$ die Menge der minimal verletzten Mengen in Iteration j sei. Dann gilt

$$y_i = \sum_{j:\, T_i \in \mathcal{V}_j} \epsilon_j,$$

und damit (vgl. Schritt (7))

$$\sum_{i=1}^{p} y_i = \sum_{j=1}^{l} |\mathcal{V}_j|\, \epsilon_j.$$

Für die Kosten von A_f ergibt sich damit, vgl. (**??**):

$$\begin{aligned} c(A_f) &= \sum_{i=1}^{p} |A_f \cap T_i| y_i \\ &= \sum_{i=1}^{p} |A_f \cap T_i| \sum_{j:\, T_i \in \mathcal{V}_j} \epsilon_j \\ &= \sum_{j=1}^{l} \Big(\sum_{T_i \in \mathcal{V}_j} |A_f \cap T_i| \Big) \epsilon_j. \end{aligned}$$

Das heißt wir erhalten einen γ-Approximationsalgorithmus, falls

$$\sum_{T_i \in \mathcal{V}_j} |A_f \cap T_i| \leq \gamma |\mathcal{V}_j|. \tag{12.13}$$

Wenden wir nun die gleiche Argumentation wie in (**??**) und Satz **??** an, so erhalten wir folgenden Satz.

Theorem 12.15. *Algorithmus* **??** *ist ein γ-Approximationsalgorithmus, falls*

$$\sum_{T_i \in \mathcal{V}(A)} |B \cap T_i| \leq \gamma |\mathcal{V}(A)| \tag{12.14}$$

für alle unzulässigen Mengen A und alle minimalen Augmentierungen B von A.

Wir wollen nun diesen Algorithmus verwenden, um Approximationsresultate (genauer gesagt 2-Approximationen) für Netzwerkentwurfsprobleme zu zeigen, die folgende Struktur haben:

$$\begin{aligned} \min & \sum_{e \in E} c_e x_e \\ & \sum_{e \in \delta(S)} x_e \geq f(S) \text{ für alle } S \subseteq V, \\ & x_e \in \{0,1\} \qquad \text{für } e \in E. \end{aligned} \tag{12.15}$$

$f : 2^V \mapsto \{0,1\}$ sei dabei eine 0/1 Funktion mit folgenden Eigenschaften:

(a) $f(V) = 0$.
(b) f ist *maximal*, das heißt

$$\forall A, B \subseteq V, A \cap B = \emptyset \text{ gilt} : f(A \cup B) \leq \max\{f(A), f(B)\}.$$

(c) f ist symmetrisch, das heißt: $f(S) = f(V \setminus S)$.

Wir nennen eine Funktion mit diesen Eigenschaften *echt*.

Beachte, dass die Mengen T_i in Algorithmus **??** assoziiert werden können mit Knotenmengen $S_i \subseteq V$, wobei $T_i = \delta(S_i)$. Unser Ziel (**??**) mit $\gamma = 2$ zu zeigen, liest sich damit wie folgt:

$$\sum_{S \in \mathcal{V}(A)} |\delta_B(S)| \leq 2|\mathcal{V}(A)|. \tag{12.16}$$

Bevor wir (**??**) für Probleme der Form (**??**) zeigen, geben wir zunächst zwei Beispiele:

Beispiel 12.16. Das Steinerbaum-Problem Gegeben sei ein ungerichteter Graph $G = (V, E)$ und eine Teilmenge der Knoten $T \subseteq V$ (*Terminalmenge* genannt). Finde eine Kantenmenge A, die T aufspannt, das heißt eine Kantenmenge A, so dass alle Paare $s, t \in T$ durch einen Weg in (V, A) verbunden sind.

Setze $f(S) = 1$, falls $S \cap T \neq \emptyset$ und $(V \setminus S) \cap T \neq \emptyset$, sowie $f(S) = 0$ andernfalls. Dann ist f echt und (**??**) ist eine korrekte Modellierung des Steinerbaum-Problems (siehe Übung).

Beispiel 12.17. Das T-join Problem Sei $T \subseteq V$ eines Graphen $G = (V, E)$ und $|T|$ gerade. Finde eine Kantenmenge $A \subseteq E$ minimalen Gewichtes (minimal bzgl. einer Funktion $c : E \mapsto \mathbb{R}_+$), so dass jeder Knoten in T ungeraden Grad und jeder Knoten nicht in T geraden Grad hat.

Setze $f(S) = 1$, falls $|S \cap T|$ ungerade, und $f(S) = 0$, falls $|S \cap T|$ gerade. Dann ist f echt und (**??**) ist eine korrekte Modellierung des T-join Problems (siehe Übung).

Um (**??**) zeigen zu können benötigen wir zwei Lemmata.

Lemma 12.18. *Sei f echt und $A \subseteq E$ beliebig. Dann gilt:*

(a) A ist zulässig genau dann, wenn für alle zusammenhängenden Komponenten C von (V, A) gilt $f(C) = 0$.
(b) Die minimalen verletzten Mengen von A sind die zusammenhängenden Komponenten C von (V, A) mit $f(C) = 1$.

Beweis. (a) ist klar, vgl. (**??**).

Um (b) zu zeigen, betrachte eine verletzte Menge S mit $f(S) = 1$ und $\delta_A(S) = \emptyset$. Offensichtlich besteht S aus der Vereinigung von zusammenhängenden Komponenten

von S. Da f maximal ist, muss mindestens für eine dieser Komponenten C gelten $f(C) = 1$. Diese ist damit ebenfalls verletzt. Das heißt nur zusammenhängende Komponenten können minimal verletzte Mengen sein und diese sind dies natürlich auch.

Lemma 12.19. *Sei $f : 2^V \mapsto \{0,1\}$ symmetrisch. Dann gilt: f ist maximal genau dann, wenn f* komplementär *ist, das heißt:*

$\forall S \subseteq V$ *und* $A \subseteq S$ *mit* $f(A) = f(S) = 0$ *folgt:* $f(S \backslash A) = 0$.

Beweis. „⇒“ Sei $S \subseteq V$, $A \subseteq S$ und $f(S) = f(A) = 0$. Zu zeigen ist $f(S \setminus A) = 0$.
Da f symmetrisch, folgt $f(V \setminus S) = 0$. Da f maximal und $f(A) = 0$ folgt damit $f(A \cup (V \setminus S)) = 0$. Wieder mit dem Argument, dass f symmetrisch ist, erhalten wir somit $f(V \setminus (A \cup (V \setminus S)) = 0$. Die letztere Menge ist jedoch gerade $S \setminus A$, woraus die Behauptung folgt.
„⇐“ Seien $A, B \subseteq V$ mit $A \cap B = \emptyset$ beliebig. Zu zeigen ist $f(A \cup B) \leq \max\{f(A), f(B)\}$.
Gilt $f(A) = 1$ oder $f(B) = 1$ so ist nichts zu zeigen. Sei also $f(A) = f(B) = 0$. Zu zeigen ist $f(A \cup B) = 0$. Da f symmetrisch, folgt $f(V \setminus A) = 0$. Wenden wir nun die Voraussetzung auf $S = V \setminus A$ und $A = B$ an, so gilt $f((V \setminus A) \setminus B) = 0$. Letztere Menge ist gleich $V \setminus (A \cup B)$, woraus mit der Symmetrie von f die Behauptung folgt.

Nun sind wir soweit, um unseren Hauptsatz zu zeigen.

Theorem 12.20. *Algorithmus* **??** *ist ein 2-Approximationsalgorithmus für das ganzzahlige Programm (***??***), falls $f : 2^V \mapsto \{0,1\}$ echt ist.*

Beweis. Betrachte eine beliebige unzulässige Menge A. Sei $\mathcal{V}(A)$ die Menge aller minimal verletzten Mengen, siehe Schritt (6) von Algorithmus **??**. Aus Lemma **??** (b) wissen wir, dass $f(S) = 1$ für alle $S \in \mathcal{V}(A)$. Betrachte weiter eine beliebige minimale Augmentierung B von A und den Graphen (V, B).

Wir bestimmen aus (V, B) einen neuen Graphen, indem wir jede zusammenhängende Komponente in (V, A) zu einem Knoten kontrahieren. Sei (H, F) der so entstehende Graph. Da B (kanten-)minimal ist, ist (H, F) ein Wald und es besteht eine 1-1 Beziehung zwischen

den Kanten $B \setminus A$ und F. Ferner gehört jeder Knoten $v \in H$ zu einer zusammenhängenden Komponente S_v von (V, A). Sei $d_F(v)$ der Grad des Knoten v in (H, F), d. h. $d_F(v) = |\delta_B(S_v)|$.

Sei weiter W die Menge aller Knoten, deren zugehörige Zusammenhangskomponente verletzt ist, das heißt $\mathcal{V}(A) = \{S_w \mid w \in W\}$. Um (**??**) zu zeigen, genügt es also

$$\sum_{w \in W} d_F(v) \leq 2|W| \tag{12.17}$$

zu zeigen.

Für den Beweis von **??** zeigen wir zunächst, dass für alle Blätter v von (H, F), das heißt für alle Knoten $v \in H$ mit $d_F(v) = 1$, gilt $f(S_v) = 1$.

Angenommen dem wäre nicht so. Sei v ein solcher Knoten und e die zu v inzidente Kante, vgl. Abbildung **??**, und sei ferner C die Zusammenhangskomponente in (V, B), die S_v enthält (beachte, S_v ist Zusammenhangskomponente in (V, A)).

Abb. 12.3. Illustration des Grad-Arguments im Beweis von Satz **??**

Da B zulässig, gilt $f(C) = 0$. Da nach Annahme $f(S_v) = 0$, folgt mit Lemma **??**, dass $f(C \setminus S_v) = 0$ gilt. Da jedoch B minimal ist, ist $B \setminus \{e\}$ unzulässig, was bedeutet, dass $f(S_v) = 1$ oder $f(C \setminus S_v) = 1$ gelten muss (vgl. Lemma **??**), ein Widerspruch.

Also wissen wir, jedes Blatt in (H, F) gehört zu W. Nun gilt:

$$\begin{aligned}
\sum_{w \in W} d_F(v) &= \sum_{w \in H} d_F(v) - \sum_{w \notin W} d_F(v) \\
&\leq (2|H| - 2) - \sum_{w \notin W} d_F(v) \\
&\leq (2|H| - 2) - 2(|H| - |W|) \\
&= 2|W| - 2,
\end{aligned}$$

was zu zeigen war.

Korollar 12.21. *Algorithmus* **??** *ist ein 2-Approximationsalgorithmus für das Steinerbaum-Problem.*

Korollar 12.22. *Algorithmus* **??** *ist ein 2-Approximationsalgorithmus für das T-join Problem.*

Man kann sich leicht überlegen, dass die relevanten Schritte (6) und (7) in Algorithmus **??** tatsächlich in polynomialer Zeit (in der Tat in linearer Zeit) für die angegebenen Probleme implementiert werden können.

Derselbe Algorithmus oder zumindest dieselbe Idee kann für weitere kombinatorische Optimierungsprobleme angewendet werden. Dazu zählen Verallgemeinerungen des Steinerbaum-Problems, das Perfekte Matching Problem, das Aufspannende Baum Problem und viele mehr. Mehr Informationen hierzu sind in [**?**] zu finden.

Es gibt viele weitere Approximationsresultate mit vielen interessanten Ideen, wie schon erwähnt sind ein großes Feld Scheduling-Probleme. Weitere bekannte Beispiele beinhalten das euklidischer TSP, das MAX-CUT Problem oder Standortplanungsprobleme um nur einige Beispiele zu nennen. Weiterführende Literatur hierzu ist zum Beispiel das Buch von Dorit Hochbaum mit dem Titel „Approximation Algorithms“.

13

Heuristiken

In diesem Kapitel beschäftigen wir uns mit dem Problem, wie man (heuristisch) zulässige Lösungen für ein gemischt ganzzahliges Programm oder kombinatorisches Optimierungsproblem finden kann. Wir werden dabei insbesondere einige Verfahren und Methoden vorstellen, die in der Praxis häufig zum Einsatz kommen. Heuristische Verfahren nützen häufig die sehr spezielle Struktur von Problemen aus, damit sie zu effizienten Verfahren werden. Wir werden hier exemplarisch auf einige grundlegende Vorgehensweise eingehen, die immer und immer wieder verwendet werden.

13.1 Der Greedy-Algorithmus

Die Idee des Greedy-Algorithmus ist, eine Lösung von Null (der leeren Menge) beginnend aufzubauen und dabei in jedem Schritt denjenigen Gegenstand (Variable) zu nehmen, der (die) den meisten Profit verspricht (engl. greedy = gefräßig). Die Vorgehensweise des Greedy-Algorithmus läßt sich gut an Optimierungsproblemen über Unabhängigkeitssystemen erklären. Wir werden danach sehen, wie man ihn auch auf andere Probleme und Situationen anpassen kann.

Definition 13.1. *Sei E eine endliche Grundmenge. Eine Menge $\mathcal{I} \subseteq \mathcal{P}(E)$ heißt* Unabhängigkeitssystem*, falls mit $G \subseteq F \in \mathcal{I}$ auch $G \in \mathcal{I}$ folgt. Jede Menge $F \in \mathcal{I}$ heißt* unabhängig*, alle anderen Mengen* abhängig*. Ist $F \subseteq E$, so heißt eine unabhängige Teilmenge von F* Basis von F*, falls sie in keiner anderen unabhängige Teilmenge von F*

Unabhängigkeitssystem

Basis von F

enthalten ist. Sei $c\colon E \mapsto \mathbb{R}$ *eine Gewichtsfunktion auf* E*. Das Problem*

$$\max_{I \in \mathcal{I}} c(I) \tag{13.1}$$

heißt Maximierungsproblem über einem Unabhängigkeitssystem*. Das Problem*

$$\min_{B\colon Basis} c(B) \tag{13.2}$$

heißt Minimierungsproblem über einem Basissystem*.*

Beispiele für (**??**) sind das Rucksack-Problem, das Stabile-Mengen-Problem, das maximale Cliquenproblem usw., Beispiele für (**??**) sind das Minimal-Aufspannende-Baum Problem und das Traveling Salesman Problem.

Der Greedy-Algorithmus betrachtet nun eine Sortierung der Grundelemente von E. Die Sortierung erfolgt nach einer bestimmten Gewichtung / Präferenz der Grundelemente. Diese muss nicht mit $c\colon E \mapsto \mathbb{R}$ übereinstimmen. Nun werden beginnend mit der leeren Menge gemäß dieser Sortierung Elemente hinzugenommen, solange die Lösung zulässig bleibt bzw. noch zu einer zulässigen Lösung ausgebaut werden kann.

Anwendungen des Greedy-Algorithmus

Minimal aufspannende Bäume

Betrachte das Problem, in einem ungerichteten Graphen $G = (V, E)$ einen bzgl. der Gewichtung $c\colon E \mapsto \mathbb{R}$ minimal aufspannenden Baum zu bestimmen (d. h. eine kreisfreie Menge $B \subseteq E$ mit $|B| = |V| - 1$). Die Menge aller kreisfreien Teilmengen von E (auch *Wälder* genannt) ist offensichtlich ein Unabhängigkeitssystem. Wähle in Algorithmus **??** für die Gewichtung $w = c$. In Schritt (3) liest sich die Bedingung wie folgt: „Falls $B_{\text{Greedy}} \cup \{e_k\}$ kreisfrei“. Dann liefert Algorithmus **??** tatsächlich eine Optimallösung.

Wälder

Theorem 13.2. *Greedy-Min liefert einen minimal aufspannenden Baum in Zeit* $\mathcal{O}(m \cdot n)$*.*

Beweis. Wir beweisen induktiv, dass es nach dem k-ten Schritt einen minimal aufspannenden Baum $\bar{B}$ mit $\bar{B} \cap \{e_1, \ldots, e_k\} = B_{\text{Greedy}} \cap \{e_1, \ldots, e_k\}$ gibt. Der Fall $k = 0$ ist offensichtlich, für den Induktionsschritt $k - 1 \mapsto k$ unterscheiden wir zwei Fälle:

(a) B_{Greedy} enthält einen Kreis, d. h. $e_k \notin B_{\text{Greedy}}$.
Da $\bar{B} \cap \{e_1, \ldots, e_{k-1}\} = B_{\text{Greedy}} \cap \{e_1, \ldots, e_{k-1}\}$ enthält auch $\bar{B} \cup \{e_k\}$ einen Kreis, also $e_k \notin \bar{B}$.
(b) B_{Greedy} enthält keinen Kreis, d. h. $e_k \in B_{\text{Greedy}}$.
Gilt $e_k \in \bar{B}$ so sind wir fertig. Falls $e_k \notin \bar{B}$ so enthält $\bar{B} \cup \{e_k\}$ einen Kreis C. Da $\bar{B} \cap \{e_1, \ldots, e_{k-1}\} = B_{\text{Greedy}} \cap \{e_1, \ldots, e_{k-1}\}$ und $e_k \in B_{\text{Greedy}}$, existiert ein Index $l > k$ mit $e_l \in C$. D. h. $B' = \bar{B} \cup \{e_k\} \setminus \{e_l\}$ ist ebenfalls ein aufspannender Baum mit $c(B') \leq c(\bar{B})$ und erfüllt die Behauptung.

Zur Laufzeit: Sortieren in Schritt (1) kostet $\mathcal{O}(m \log m)$, z. B. unter Verwendung von Quicksort oder Heapsort. Schritt (3) wird maximal m-mal ausgeführt. Bleibt abzuschätzen, wie lange Schritt (3) selbst benötigt. Das Überprüfen, ob die Kantenmenge in Schritt (3) kreisfrei ist, kann mit Tiefensuche (engl. depth first search) in $\mathcal{O}(n)$ durchgeführt werden; beachte, dass B_{Greedy} maximal n Kanten enthält. Damit erhalten wir insgesamt für Algorithmus **??** eine Laufzeit von $\mathcal{O}(m \cdot \log m + m \cdot n) = \mathcal{O}(m \cdot n)$. □

Das Rucksack-Problem

Betrachte das 0/1 Rucksack-Problem (vgl. Beispiel **??**)

$$\begin{aligned} \max & \sum_{j=1}^{n} c_j x_j \\ & \sum_{j=1}^{n} a_j x_j \leq b \\ & x_j \in \{0,1\} \qquad \text{für } j = 1, \ldots, n\,. \end{aligned}$$

Offensichtlich ist die Menge aller zulässigen Lösungen $\mathcal{F}$ ein Unabhängigkeitssystem. Wenden wir Algorithmus **??** auf das 0/1 Rucksack-Problem an, so kann der Algorithmus beliebig schlechte Lösungen liefern.

Beispiel 13.3 (Zielfunktions-Greedy). Wähle $w = c$ in Algorithmus **??** und betrachte

$$\begin{aligned} \max\ & n x_1 + (n-1) x_2 + \ldots + (n-1) x_{n+1} \\ & n x_1 + \quad x_2 + \ldots + \quad x_{n+1} \leq n\,. \end{aligned}$$

Algorithmus **??** liefert die Lösung $I_{\text{Greedy}} = \{1\}$ mit Zielfunktionswert $c_{\text{Greedy}} = n$, während die Optimallösung $c_{\text{OPT}} = n(n-1)$ ist.

Beispiel 13.4 (Gewichtsdichten-Greedy). Wähle $w_i = \frac{c_i}{a_i}$ für $i = 1, \ldots, n$ in Algorithmus **??** und betrachte

$$\begin{aligned} \max\, & x_1 + \alpha x_2 \\ & x_1 + \alpha x_2 \leq \alpha \\ & x_1, x_2 \in \{0, 1\}\,. \end{aligned}$$

Dann liefert $c_{\text{Greedy}} = 1$, während $c_{\text{OPT}} = \alpha$ ist. Man kann jedoch zeigen, wie wir später noch sehen werden, dass der Gewichtsdichten-Greedy höchstens zweimal so schlecht ist wie die Optimallösung, wenn $x_i \in \{0, 1\}$ durch $x_i \in \mathbb{Z}_+$ für alle $i = 1, \ldots, n$ ersetzt wird.

Wir sehen also, dass der Greedy-Algorithmus auch beliebig schlecht abschneiden kann. Wir wollen im Folgenden zeigen, dass der Greedy-Algorithmus dann und nur dann eine Optimallösung liefert, falls das Unabhängigkeitssystem zusätzlich die Eigenschaft erfüllt, dass für alle $F \subseteq E$ gilt:

$$B, B' \text{ Basen von F } \implies |B| = |B'|\,.$$

Matroid

In diesem Fall spricht man von einem *Matroid*. Um diesen Satz zeigen zu können, benötigen wir folgende Definition:

Definition 13.5. *Für $F \subseteq E$ bezeichne*

$$r(F) := \max\{|B| \colon B \textit{ ist Basis von } F\}$$

Rang

den Rang *von F sowie*

$$r_u(F) := \min\{|B| \colon B \textit{ ist Basis von } F\}$$

unteren Rang

den unteren Rang *von F.*

Anmerkung 13.6. Für Matroide gilt offensichtlich:

$$r_{\text{u}}(F) = r(F)\,.$$

Theorem 13.7 (Jenkyns, 1976 [?]). *Es gilt*

$$\min_{F \subseteq E} \frac{r_u(F)}{r(F)} \leq \frac{c(I_{Greedy})}{c(I_{OPT})} \leq 1\,,$$

und für jedes Unabhängigkeitssystem gibt es Gewichte $c_i \in \{0, 1\}$ für $i \in E$, so dass die erste Ungleichung mit Gleichheit angenommen wird.

Beweis. O.B.d.A. gelte $c_i > 0$ für $i = 1, \ldots, n$ und $c_1 \geq c_2 \geq \ldots \geq c_n$. Wir führen ein weiteres Element $c_{n+1} := 0$ ein und setzen

$$E_i = \{1, \ldots, i\} \qquad \text{sowie} \qquad q = \min_{F \subseteq E} \frac{r_{\mathrm{u}}(F)}{r(F)}$$

Für $F \subseteq E$ gilt:

$$\begin{aligned} c(F) &= \sum_{i \in F} c_i \\ &= \sum_{i \in F} \left(\sum_{j=i}^{n} (c_j - c_{j+1}) \right) \\ &= \sum_{i=1}^{n} |F \cap E_i| \cdot (c_i - c_{i+1}) \end{aligned}$$

Wegen $I_{\mathrm{OPT}} \cap E_i \subseteq I_{\mathrm{OPT}}$ gilt $I_{\mathrm{OPT}} \cap E_i \in \mathcal{I}$, und somit

$$|I_{\mathrm{OPT}} \cap E_i| \leq r(E_i) \,.$$

Die Vorgehensweise des Greedy-Algorithmus impliziert, dass $I_{\mathrm{Greedy}} \cap E_i$ eine Basis von E_i ist, also

$$|I_{\mathrm{Greedy}} \cap E_i| \geq r_{\mathrm{u}}(E_i) \,.$$

Damit gilt:

$$|I_{\mathrm{Greedy}} \cap E_i| \geq |I_{\mathrm{OPT}} \cap E_i| \cdot \frac{r_{\mathrm{u}}(E_i)}{r(E_i)} \geq |I_{\mathrm{OPT}} \cap E_i| \cdot q$$

und

$$\begin{aligned} c(I_{\mathrm{Greedy}}) &= \sum_{i=1}^{n} |I_{\mathrm{Greedy}} \cap E_i| \cdot (c_i - c_{i+1}) \\ &\geq \sum_{i=1}^{n} |I_{\mathrm{OPT}} \cap E_i| \cdot q(c_i - c_{i+1}) \\ &= q \sum_{i=1}^{n} |I_{\mathrm{OPT}} \cap E_i| \cdot (c_i - c_{i+1}) \\ &= q \cdot c(I_{\mathrm{OPT}}) \,. \end{aligned}$$

Die erste Ungleichung der Behauptung wäre damit bewiesen. Die zweite Ungleichung ist trivial, da jede Lösung höchstens schlechter als die Optimallösung sein kann.

Es bleibt zu zeigen, dass das Minimum auch angenommen wird. Sei nun

$$F \subseteq E \text{ mit } q = \frac{r_{\mathrm{u}}(F)}{r(F)} .$$

O.B.d.A. sei $F = \{1, \ldots, k\}$ und $B = \{1, \ldots, p\} \subseteq F$ eine Basis von F mit $|B| = r_{\mathrm{u}}(F)$. Setze

$$c_i = \begin{cases} 1 & \text{für } i = 1, \ldots, k \\ 0 & \text{sonst.} \end{cases}$$

Der Greedy-Algorithmus liefert

$$I_{\mathrm{Greedy}} = B$$

mit $c(I_{\mathrm{Greedy}}) = r_{\mathrm{u}}(F) = p$, während für jede Optimallösung I_{OPT} gilt

$$c(I_{\mathrm{OPT}}) = r(F) .$$

□

Korollar 13.8. *Sei $\mathcal{I}$ ein Unabhängigkeitssystem auf E. Dann sind äquivalent:*

(i) $\mathcal{I}$ ist ein Matroid.
(ii) Für alle Gewichte $c \in \mathbb{R}^E$ liefert der Greedy-Algorithmus **??** *eine optimale Lösung für*

$$\min\{c(I) \colon I \in \mathcal{I}\} .$$

(iii) Für alle „0/1“-Gewichte $c \in \{0,1\}^E$ liefert der Greedy-Algorithmus eine optimale Lösung für

$$\max\{c(I) \colon I \in \mathcal{I}\} .$$

Beweis. Die Folgerung ergibt sich mit Bemerkung **??** direkt aus Satz **??**. □

Wir haben damit durch **??** die Problemklasse der Matroide über einen Algorithmus **??** charakterisiert. Ist also kein Matroid gegeben, was z. B. für das Rucksack-Problem, das Traveling Salesman Problem und für die meisten praxisrelevanten Probleme gilt, liefert der Greedy-Algorithmus keine Optimallösung, meist sogar beliebig schlechte Lösungen.

Dennoch erfreut sich der Greedy-Algorithmus in der Praxis großer Beliebtheit, denn er basiert auf einem einfach zu überschauenden Zielkriterium in Form der Funktion w und er ist lokal nicht zu entkräften.

13.2 Lokale Suche

Sobald man einmal eine zulässige Lösung gefunden hat, versucht man diese noch zu verbessern. In diesem Abschnitt behandeln wir einige Methoden und Ideen, die versuchen, dies zu leisten. Wir werden sehen, das diese Methoden auch dazu verwendet werden können, zulässige Lösungen zu finden, indem man die Unzulässigkeiten einer bereits bestehenden Lösung in die Zielfunktion mitaufnimmt und versucht diese Unzulässigkeiten zu minimieren.

Die grundlegende Vorgehensweise aller Verfahren ist gleich. Ausgehend von einer Lösung (zulässig oder unzulässig) definiert man sich eine Nachbarschaft der Lösung. In dieser Nachbarschaft sucht man nach einer besseren (oder bestmöglichen) Lösung. Hat man eine gefunden, tauscht man die alte gegen die neue Lösung und iteriert.

Sei dazu $\mathcal{F}$ die Menge von (zulässigen) Lösungen für ein Problem Π (z. B. für ein gemischt-ganzzahliges Problem, für ein Traveling Salesman Problem, ...). Für jede Lösung $S \in \mathcal{F}$ definiere eine (zulässige) Nachbarschaft $N(S) \subseteq \mathcal{F}$. Darüber hinaus sei ein Zielfunktional $w\colon \mathcal{F} \mapsto \mathbb{R}$ bzgl. dem wir messen, ob eine Lösung besser ist als eine andere (vgl. Gewichtung w beim Greedy-Algorithmus). Dann kann man die Grundversion der lokalen Suche wie in **??** beschreiben:

Der Erfolg dieser Algorithmus hängt natürlich stark von der Definition der Nachbarschaft $N(\cdot)$ ab. Hier zwei Beispiele.

Beispiel 13.9.

- 2-Austausch beim TSP: vgl. Abbildung **??**.

Abb. 13.1. 2-Austausch beim TSP

- 1-Austausch / 2-Austausch beim Equipartitions-Problem: vgl. Abbildung **??**. Als mögliches Zielfunktional kann hier beispielsweise

$$w(S) = \sum_{e \in \delta(S)} c_e + \alpha(|S| - |V \setminus S|)^2$$

gewählt werden.

Abb. 13.2. 1-Austausch bzw. 2-Austausch beim Equipartitions-Problem

Zielfunktional und Nachbarschaft sind natürlich stark problemabhängig. Ein Problem dieses Verfahrens ist es, dass nur Verbesserungen akzeptiert werden und damit in lokalen Minima hängen geblieben werden kann, vgl. Abbildung **??**.

Abb. 13.3. Lokales versus globales Minimum

Die folgenden Verfahren versuchen dies zu vermeiden unter Beibehaltung der Grundstruktur von Algorithmus **??**.

13.2.1 Tabu Search

Die Grundidee ist, sobald man in einem lokalen Optimum angekommen ist, die bestmögliche Lösung in der Nachbarschaft zu akzeptieren, auch wenn sie schlechter ist. Ein Problem hierbei ist das Auftreten von Kreiseln, da die Lösung der Voriteration häufig in der Nachbarschaft der Folgelösung enthalten ist:

$$S^0 \to S^1 \to S^0 \to S^1 \to \cdots$$

tabu

Tabuliste

Um ein solches Kreiseln zu vermeiden, werden gewisse Austäusche (Nachbarschaften) für *tabu* erklärt. Um Kreiseln vollkommen auszuschließen, müssten alle Nachbarschaften verboten werden, die zu irgendeiner vorher erzeugten Lösung führen. Dies ist allerdings sehr zeit- und speicheraufwendig. Deshalb beschränkt man sich meist auf eine *Tabuliste* beschränkter Größe, die die letzten Lösungen oder Austäusche beinhaltet.

Die Parameter, die hier zusätzlich zu denen aus Algorithmus **??** zu spezifizieren sind, sind die Größe der Tabu-Liste und die Stop-Bedingung. Je kleiner die Liste, desto größer die Gefahr des Kreiselns. Je größer die Liste, desto größer die Laufzeit. Als Stop-Bedingung wir häufig eine

feste Anzahl von Iterationen oder eine feste Anzahl von Iterationen ohne Verbesserung gewählt.

Auch hier gilt, dass die einzelnen Parameter stark problemabhängig sind und intensiven Tunings bedarf, bis vernünftige Werte gefunden werden.

13.2.2 Simulated Annealing

Auch Simulated Annealing erlaubt zwischendurch schlechtere Lösungen, jedoch nicht wie eben durch Steuerung einer festen Tabuliste, sondern zufallsgesteuert. Die Idee dabei ist, bessere Lösungen immer zu erlauben, schlechtere jedoch nur mit einer gewissen Wahrscheinlichkeit. Dabei ist die Wahrscheinlichkeit um so geringer je schlechter die Lösung ist. Darüber hinaus wird die Wahrscheinlichkeit, schlechtere Lösungen zu akzeptieren, im Laufe des Verfahrens immer weiter verringert, so dass der Algorithmus mit einer guten Lösung enden sollte.

Die Idee zu diesem Verfahren kommt aus der theoretischen Physik zur Simulation von Phänomenen der statistischen Mechanik, insbesondere von Phasenübergängen wie Kristallisation von Flüssigkeiten. Daher kommen auch die Begriffe wie Temperatur und Gefrierpunkt.

Algorithmus **??** hat in der Regel sehr hohe Laufzeiten; es gibt keine Garantie, wann das Optimum gefunden wird. Dazu das Bild aus der Physik: Zu schnelles Abkühlen würde keinen Zustand minimaler Energie erzielen, es muss langsam abgekühlt werden, damit sich „ungewollte Strukturen (Klumpen)“ reorganisieren können.

Man kann zeigen, dass bei geeigneter Wahl von l und T mit Simulated Annealing tatsächlich das Optimum gefunden werden kann, allerdings unter der Voraussetzung, das man Algorithmus **??** beliebig lange laufen lässt. Es sind im Allgemeinen viele Experimente notwendig, um Schritte (**??**) und (**??**) geeignet zu wählen.

13.2.3 Genetische Algorithmen

Die grundlegende Idee genetischer Algorithmen ist, nicht nur eine Lösung zu betrachten und versuchen, diese zu verbessern, sondern eine Menge von Lösungen (Population genannt) zu betrachten und durch Kombinationen dieser Lösungen zu neuen besseren Lösungen zu gelangen.

Die Grundmuster und Begriffswelt sind dabei aus der Genetik entnommen. Aus einer Population werden eine Reihe von Elternpaare (Paare von Lösungen) bestimmt, die sich kreuzen und damit einige neue Lösungen erzeugen. Diese Lösungen (evtl. zusammen mit den Eltern) werden teilweise noch mutiert (zufällig geändert). Basierend auf ihrer Fitness (Bewertungsfunktion w wie in den Abschnitten oben) wird eine neue Generation ausgewählt und der Prozess wiederholt. Im Detail wird ein genetischer Algorithmus in **??** beschrieben. Auch hier gilt wie bei den Verfahren vorher, dass die meisten Schritte sehr problemabhängig sind und nur durch viele Tests geeignet bestimmt werden können.

Einige allgemeine Regeln zu den einzelnen Schritten sind:

- *Elternwahl:* Idee, wähle „fittere“ Lösungen mit höherer Wahrscheinlichkeit, z. B. S_i mit Wahrscheinlichkeit

$$\frac{w(S_i)}{\sum_{j=1}^{k} w(S_j)} .$$

- *Kreuzungen:* Hier werden Teillösungen, von denen man glaubt, dass sie gut sind, von den Eltern übernommen (z. B. Teilweg beim TSP) und zu neuen Lösungen verkettet.
- *Mutation:* Meist zufällig gesteuerte Austäusche (z. B. 2-Austausch beim TSP).
- *Selektion:* Es werden in der Regel nicht nur die bzgl. w besten, sondern auch mit gewissen Wahrscheinlichkeiten schlechtere Lösungen in der Population behalten.

Allen Verfahren, die wir in diesem Kapitel kennengelernt haben, ist gemein, dass sie keine Gütegarantie geben können. Sie mögen zwar oft gute Lösungen finden, aber wie gut sie tatsächlich sind oder ob sie ggf. eine Optimallösung gefunden haben, können sie nicht feststellen. Wir werden in einem späteren Kapitel noch primale Verfahren kennenlernen, die in polynomialer Zeit eine gewisse Gütegarantie geben können (sogenannte Approximationsalgorithmen). Meist sind die Garantien, die man dort erhält, für die Praxis nicht tauglich. Um wirklich (auch für die Praxis) verwertbare Garantien geben zu können, muss man sich mit der Struktur der Probleme, insbesondere mit der Struktur der Lösungsmengen der zugrundeliegenden Probleme befassen.

A

Notation

A.1 Grundlegende Notationen

Mit $\mathbb{R}$, $\mathbb{Q}$, $\mathbb{Z}$ und $\mathbb{N}$ bezeichnen wir die Menge der reellen, rationalen Zahlen.

Literaturverzeichnis

1. E. Balas, *Facets of the knapsack polytope*, Mathematical Programming **8** (1975), 146 – 164.
2. E. Balas, S. Ceria, and G. Cornuéjols, *A lift-and-project cutting plane algorithm for mixed* $0-1$ *programs*, Mathematical Programming **58** (1993), 295–324.
3. J.F. Benders, *Partitioning procedures for solving mixed variables programming*, Numerische Mathematik **4** (1962), 238–252.
4. C. Berge, *Färbung von Graphen, deren sämtliche bzw. deren ungerade Kreise starr sind (Zusammenfassung)*, Wissenschaftliche Zeitschrift, Mathematisch-Naturwissenschaftliche Reihe, Martin Luther Universität Halle-Wittenberg, 1961.
5. A. Beutelspacher, *Lineare Algebra. Eine Einführung in die Wissenschaft der Vektoren, Abbildungen und Matrizen*, Vieweg, 2004 (ngerman).
6. R.G. Bland, D. Goldfarb, and M.J. Todd, *The ellipsoid method: A survey*, Operations Research **29** (1981), 1039 – 1091.
7. R. Borndörfer, *Aspects of set packing, partitioning, and covering*, Ph.D. thesis, Technische Universität Berlin, 1998.
8. V. Chvátal, *Edmonds polytopes and a hierarchy of combinatorial problems*, Discrete Mathematics **4** (1973), 305 – 337.
9. ———, *On certain polytopes associated with graphs*, Journal on Combinatorial Theory B **18** (1975), 305 – 337.
10. G.B. Dantzig and P. Wolfe, *Decomposition principle for linear programs*, Operations Research **8** (1960), 101–111.
11. J. Edmonds and R. Giles, *A min-max relation for submodular functions on graphs*, Studies in Integer Programming (P.L. Hammer, ed.), vol. 1, North-Holland, Amsterdam, 1977, pp. 185 – 204.

12. C.E. Ferreira, *On combinatorial optimization problems arising in computer system design*, Ph.D. thesis, Technische Universität Berlin, 1994.
13. Jr. L.R. Ford and D.R. Fulkerson, *Maximal flow through a network*, Canadian Journal of Mathematics **8** (1956), 399–404.
14. J.J.H. Forrest and J.A. Tomlin, *Updating triangular factors of the basis to maintain sparsity in the product-form simplex method*, Mathematical Programming **2** (1972), 263 – 278.
15. D. Fulkerson, A.J. Hoffman, and R. Oppenheim, *On balanced matrices*, Mathematical Programming Study **1** (1974), 120–132.
16. D.R. Fulkerson, *Blocking and Anti-Blocking Pairs of Polyhedra*, Mathematical Programming **1** (1971), 168–194.
17. ———, *Packing rooted directed cuts in a weighted directed graph*, Mathematical Programming **6** (1974), 1 – 13.
18. P.E. Gill, W. Murray, M.A. Saunders, and M.H. Wright, *A practical anti-cycling procedure for linearly constrained optimization*, Mathematical Programming **45** (1989), 437 – 474.
19. M.X. Goemans and D.P. Williamson, *The primal-dual method for approximation algorithms and its application to network design problems*, Approximation Algorithms (D. Hochbaum, ed.), 1997.
20. J.L. Goffin and J.P. Vial, *Convex nondifferentiable optimization: A survey focussed on the analytic center cutting plane method*, Tech. Report 99.02, Logilab, Universite de Geneve, 1999.
21. R.E. Gomory, *An algorithm for the mixed integer problem*, Technical Report RM-2597, The RAND Cooperation, 1960.
22. ———, *Solving linear programming problems in integers*, Combinatorial analysis, Proceedings of Symposia in Applied Mathematics (Providence RI) (R. Bellman and M. Hall, eds.), vol. 10, 1960.
23. ———, *An algorithm for integer solutions to linear programming*, Recent Advances in Mathematical Programming (New York) (R.L. Graves and P. Wolfe, eds.), McGraw-Hill, 1969, pp. 269 – 302.
24. M. Grötschel, L. Lovász, and A. Schrijver, *The ellipsoid method and its consequences in combinatorial optimization*, Combinatorica **1** (1981), 169 – 197.
25. ———, *Geometric algorithms and combinatorial optimization*, Springer, 1988.
26. P.L. Hammer, E.L. Johnson, and U.N. Peled, *Facets of regular 0-1 polytopes*, Mathematical Programming **8** (1975), 179 – 206.
27. P.M.J. Harris, *Pivot selection methods of the devex LP code*, Mathematical Programming **5** (1973), 1 – 28.

28. H. Heuser, *Lehrbuch der Analysis*, vol. 1, B.G. Teubner, 1988 (ngerman).
29. ———, *Lehrbuch der Analysis*, vol. 2, B.G. Teubner, 1988 (ngerman).
30. J.B. Hiriart-Urruty and C. Lemarechal, *Convex analysis and minimization algorithms, part 2: Advanced theory and bundle methods*, Grundlehren der mathematischen Wissenschaften, vol. 306, Springer-Verlag, 1993.
31. A.J. Hoffmann and J.B. Kruskal, *Integral boundary points of convex polyhedra*, Linear inequalities and related systems (H. W. Kuhn and A. W. Tucker, eds.), Princeton University Press, Princeton, NJ, 1956, pp. 223–246.
32. S. Hoogeveen, M. Skutella, and G. Woeginger, 2001, Personal Communication with M. Skutella.
33. T.A. Jenkins, *The efficacy of the "greedy" algorithm*, Proc. 7th southeast. Conf. Comb., Graph Theory, Comput. (1976), 341 – 350.
34. L.G. Khachiyan, *A polynomial algorithm in linear programming*, Doklady Akademiia Nauk SSSR **244** (1979), 1093 – 1096, (translated in Soviet Mathematics Doklady **20**, 191 – 194, 1979.
35. D. Klabjan, G.L. Nemhauser, and C. Tovey, *The complexity of cover inequality separation*, Operations Research Letters **23** (1998), 35 – 40.
36. H.-J. Kowalsky, *Lineare Algebra*, de Gruyter, 1979 (ngerman).
37. S. O. Krumke and H. Noltemeier, *Graphentheorische Konzepte und Algorithmen*, B.G. Teubner, 2005 (ngerman).
38. H. W. Kuhn and R.E. Quandt, *An experimental study of the simplex method*, Proceedings of Symposia in Applied Mathematics XV (Providence, RI) (N.C. Metropolis, ed.), American Mathematical Society, 1963.
39. H. Marchand, A. Martin, R. Weismantel, and L.A. Wolsey, *Cutting planes in integer and mixed integer programming*, Tech. Report CORE DP9953, Université Catholique de Louvain, Louvain-la-Neuve, Belgium, 1999.
40. H. Marchand and L.A. Wolsey, *The* $0-1$ *knapsack problem with a single continuous variable*, Mathematical Programming **85** (1999), 15 – 33.
41. A. Martin, *Integer programs with block structure*, Habilitations-Schrift, Technische Universität Berlin, 1998.
42. M.W. Padberg, *On the facial structure of set packing polyhedra*, Mathematical Programming **5** (1973), 199–215.
43. ———, *A note on zero-one programming*, Operations Research **23** (1975), 833–837.
44. ———, $(1, k)$*-configurations and facets for packing problems*, Mathematical Programming **18** (1980), 94–99.
45. ———, *Linear optimization and extensions*, Springer, 1995.

46. B. Polyak, *A general method of solving extremum problems (in russian)*, Doklady Akademii Nauk SSR **174** (1967), 33 – 36, [English translation in USSR Computational Mathematics and Mechanical Physics 9 (1969), 14 – 29].
47. W. Rudin, *Principles of mathematical analysis*, 3 ed., McGraw Hill, 1976.
48. A. Schrijver, *Theory of linear and integer programming*, Wiley, Chichester, 1986.
49. J. Stoer and R. Bulirsch, *Numerische Mathematik*, vol. 1, Springer, 1991 (ngerman).
50. _____, *Numerische Mathematik*, vol. 2, Springer, 1991 (ngerman).
51. U.H. Suhl and L.M. Suhl, *Computing sparse lu factorizations for large-scale linear programming bases*, ORSA Journal on Computing **2** (1990), 325–335.
52. R. Weismantel, *On the 0/1 knapsack polytope*, Mathematical Programming **77** (1997), 49–68.
53. P. Wolfe and L. Cutler, *Experiments in linear programming*, Recent Advances in Mathematical Programming (New York) (R.L. Graves and P. Wolfe, eds.), McGraw-Hill, 1963, pp. 177 – 200.
54. L.A. Wolsey, *Faces of linear inequalities in 0-1 variables*, Mathematical Programming **8** (1975), 165 – 178.