Lineare Optimierung

(Algorithmische Diskrete Mathematik II)

Skriptum zur Vorlesung im WS 2003/2004

Prof. Dr. Martin Grötschel
Institut für Mathematik
Technische Universität Berlin

Version vom 27. Februar 2004

Vorwort

Die Vorlesung "Lineare Optimierung" ist die zweite Vorlesung (ADM II) im Zyklus des Studienschwerpunktes "Algorithmische Diskrete Mathematik" an der TU Berlin in den Studiengängen Mathematik, Techno- und Wirtschaftsmathematik. Detaillierte Kenntnis der vorausgegangenen Vorlesung "Graphen- und Netzwerkalgorithmen" wird nicht vorausgesetzt. Graphen- und Netzwerke und damit zusammenhängende Probleme werden lediglich als Beispielmaterial verwendet.

In dieser Vorlesung werden hauptsächlich die Grundlagen der linearen Optimierung (auch lineare Programmierung genannt) behandelt, also einige Aspekte der linearen Algebra und der Polyedertheorie. Natürlich steht die Entwicklung von Algorithmen zur Lösung von Optimierungsproblemen im Vordergrund. Sehr ausführlich werden der Simplexalgorithmus und seine Varianten behandelt, aber ebenso werden wir Innere-Punkte-Verfahren, die Ellipsoidmethode und Schnittebenenverfahren der ganzzahligen Optimierung diskutieren.

Oktober 2003 M. Grötschel

Vorbemerkungen

Literatur

Die Literatur zum Thema „Lineare Programmierung“ ist überwältigend umfangreich. Hierzu rufe man einfach einmal die Datenbank MATH des Zentralblatts für Mathematik auf und starte eine Suche nach Büchern, die die Wörter *linear* und *programming* im Titel enthalten. Man erhält derzeit beinahe 200 Referenzen. Über 30 Bücher enthalten die beiden Wörter *lineare Optimierung* und fast genauso viele die Wörter *lineare Programmierung* in ihrem Titel.

Eine Google-Suche nach „linear programming“ liefert derzeit fast 240.000 Ergebnisse. Der (bei meiner Suche) am höchsten bewertete Eintrag war „Linear Programming Frequently Asked Questions“. Diese Webseite, gepflegt von Bob Fourer, hat die URL: `http://www-unix.mcs.anl.gov/otc/Guide/faq/linear-programming-faq.html` und ist eine wirklich gute Quelle mit vielfältigen und ausgezeichneten Informationen zur linearen Optimierung. Unter der Rubrik „Textbooks“ ist hier eine Reihe guter Bücher zum Thema zusammengestellt worden. Ich empfehle aus dieser Auswahl:

- Dantzig, George B., *Linear Programming and Extensions*, Princeton University Press, 1963. Dies ist der „Klassiker“ des Gebiets, auch heute noch eine interessante Quelle, und 1998 im Paperback-Format erneut erschienen.
- Chvátal, Vasek, *Linear Programming*, Freeman, 1983. Dies ist ein ausgezeichnetes Einführungsbuch, das sich insbesondere an Leser mit geringer mathematischer Vorbildung richtet.
- Padberg, Manfred, *Linear Optimization and Extensions*, Springer, 2001. Wenn sie wissen möchten, was die Berliner Luftbrücke mit linearer Optimierung zu tun hat, dann sollten Sie einmal in dieses Buch schauen.
- Schrijver, Alexander, *Theory of Linear and Integer Programming*, Wiley, 1986. Dieses Buch fasst den Kenntnisstand bis zum Jahre 1985 so gut wie vollständig zusammen, ein herausragendes Buch, das sich an den fortgeschrittenen Leser richtet.
- Vanderbei, Robert J., *Linear Programming: Foundations and Extensions.* Kluwer Academic Publishers, 1996. Eine schöne, Praxis bezogene Darstellung von Simplex- und Innere-Punkte-Methoden, Source-Codes zu den im Buch präsentierten Verfahren sind online verfügbar, siehe `http://www.princeton.edu/~rvdb/LPbook/`

- Wright, Stephen J., *Primal-Dual Interior-Point Methods.* SIAM Publications, 1997. Innere-Punkte-Methoden sind erst Mitte der 80er Jahre entdeckt und populär geworden. Wrights Buch behandelt die wichtigsten Verfahrensklassen dieses Ansatzes.

Polyedertheorie ist ein wichtiger Aspekt der Vorlesung. Sehr gute Bücher hierzu sind:

- Grünbaum, Branko, *Convex Polytopes*, Springer-Verlag, Second Edition, 2003
- Ziegler, Günter M., *Lectures on Polytopes*, Springer-Verlag, Revised Edition, 1998

Ein Artikel, der die enormen Fortschritte bei der praktischen Lösung linearer Programme seit Anfang der 90er Jahre beschreibt, ist:

- Robert E. Bixby, *Solving Real-World Linear Programs: A Decade and More of Progress*, Operations Research 50 (2002) 3-15.

Fortschritte in den letzten Jahren im Bereich der ganzzahligen und gemischt-ganzzahligen Optimierung sind dokumentiert in

- Robert E. Bixby. Mary Fenelon, Zonghao Gu, Ed Rothberg, und Roland Wunderling, *Mixed-Integer Programming: A Progress Report*, erscheint 2004 in: Grötschel, M. (ed.), *The Sharpest Cut*, MPS/SIAM Series on Optimization.

Lineare Optimierung im Internet: Vermischtes

Umfangreiche Literaturverweise, Preprint-Sammlungen, Hinweise auf Personen, die sich mit Optimierung und Spezialgebieten davon beschäftigen, verfügbare Computercodes, Beispiele zur Modellierung von praktischen Problemen, Testbeispiele mit Daten linearer Programme, etc, findet man u. a. auf den folgenden Internet-Seiten:

- `http://www.optimization-online.org/`
  (Optimization Online is a repository of eprints about optimization and related topics.)

- `http://www-unix.mcs.anl.gov/otc/InteriorPoint/`
  (Interior Point Methods Online)

- `http://www-fp.mcs.anl.gov/otc/Guide/CaseStudies/index.html` (NEOS Guide: Case Studies)

- `http://miplib.zib.de`
  (Testbeispiele für gemischt-ganzzahlige Optimierungsprobleme)

- `http://elib.zib.de/pub/Packages/mp-testdata/` (Verweise auf Sammlungen von Test-Beispielen wie die Netlib-LP-Testprobleme.)

Im Internet gibt es auch verschiedene Einführungen in die lineare Optimierung. Solche Webseiten, gerichtet an Anfänger, sind z.B.:

- `http://carbon.cudenver.edu/˜hgreenbe/courseware/LPshort/intro.html`

- `http://fisher.osu.edu/˜croxton_4/tutorial/`

Wie oben erwähnt, wird in dieser Vorlesung besonders viel Wert auf die geometrische Fundierung der linearen Optimierung gelegt. Die Lösungsmengen von linearen Programmen sind Polyeder. Im Internet findet man viele Seiten, die Polyeder bildlich darstellen. Hier sind zwei Webseiten mit animierten Visualisierungen von Polyedern:

- `http://www.eg-models.de/` (gehe z. B. zu Classical Models, Regular Solids oder Discrete Mathematics, Polytopes)

- `http://mathworld.wolfram.com/topics/Polyhedra.html`

Software

Häufig möchte man Polyeder von einer Darstellungsform (z.B. Lösungsmenge von Ungleichungssystemen) in eine andere Darstellungsform (z.B. konvexe Hülle von endlich vielen Punkten) verwandeln, um gewisse Eigenschaften besser zu verstehen. Auch hierzu gibt es eine Reihe von Codes (die meisten stehen kostenlos zum Download bereit). Eine Sammlung solcher Verfahren ist zu finden unter:

`http://elib.zib.de/pub/Packages/mathprog/polyth/index.html`

Die Programmsammlung PORTA stammt aus einer Diplomarbeit, die Thomas

Christof bei mir geschrieben und dann mit anderen Kollegen (u. a. A. Löbel (ZIB)) weiterentwickelt hat. Das deutlich umfangreichere Programmpaket Polymake von E. Gawrilow und M. Joswig wurde im mathematischen Institut der TU Berlin entwickelt und wird hier weiter gepflegt und ausgebaut.

Es gibt eine Vielzahl von Codes zur Lösung linearer Programme und Modellierungssprachen, mit Hilfe derer man praktische Probleme (einigermaßen) bequem in Datensätze verwandeln kann, die LP-Löser akzeptieren. Eine kommentierte Übersicht hierzu ist zu finden unter:

- `http://www-unix.mcs.anl.gov/otc/Guide/faq/linear-programming-faq.html#Q2`

Eine weitere Liste findet man unter:

- `http://elib.zib.de/pub/Packages/mathprog/linprog/index.html`

Die Zeitschrift OR/MS Today veröffentlicht regelmäßig Übersichten über kommerzielle und frei verfügbare Software zur Lösung linearer Programme. Die letzte Übersicht datiert aus dem Jahre 2001:

- `http://lionhrtpub.com/orms/surveys/LP/LP-survey.html`

In dieser Vorlesung haben wir Zugang zu dem kommerziellen Code CPLEX der Firma ILOG. Nach Ansicht vieler ist dies derzeit der weltweit beste Code zur Lösung linearer und ganzzahliger Programme (Ihnen steht allerdings nicht die allerneueste Version zur Verfügung). Vom Konrad-Zuse-Zentrum (ZIB) wird ein für akademische Zwecke kostenlos verfügbarer Code namens SoPlex zur Lösung von LPs angeboten, siehe:

- `http://www.zib.de/Optimization/Software/Soplex`

Dieses Programm stammt aus der Dissertation von Roland Wunderling aus dem Jahre 1997 an der TU Berlin und wird am ZIB weiterentwickelt.

Als Modellierungssprache (z. B. zur Erzeugung von Daten im lp- oder mps-Format, die von den meisten LP-Lösern gelesen werden können) wird in der Vorlesung Zimpl angeboten, die von Thorsten Koch (ZIB) entwickelt und gepflegt wird, siehe:

- `http://www.zib.de/koch/zimpl/`

Zum Schluss noch ein Hinweis auf allgemeine Optimierungsprobleme. Hans Mittelmann unterhält eine Webseite, siehe

- `http://plato.la.asu.edu/guide.html`

die dabei hilft, für ein vorliegendes Optimierungsproblem geeignete Software ausfindig zu machen.

Inhaltsverzeichnis

Kapitel 1

Einführung

Dieses erste Kapitel dient der Einführung in das Gebiet "Optimierung". Optimierungsaufgaben werden zunächst ganz allgemein definiert und dann immer mehr spezialisiert, um zu Aufgaben zu kommen, über die mathematisch interessante Aussagen gemacht werden können. Anwendungen der mathematischen Modelle werden nur andeutungsweise behandelt. Wir beschäftigen uns zunächst hauptsächlich mit der Theorie, denn ohne theoretische Vorkenntnisse kann man keine Anwendungen betreiben.

Eine für die Praxis wichtige Tätigkeit ist die Modellbildung oder -formulierung, also die Aufgabe ein reales Problem der Praxis in eine mathematische Form zu "kleiden". Diese Aufgabe wird von Mathematikern häufig ignoriert oder als trivial betrachtet. Aber es ist gar nicht so einfach, für ein wichtiges Praxisproblem ein "gutes" mathematisches Modell zu finden. Dies muss anhand vieler Beispiele geübt werden, um „Modellbildungserfahrung" zu gewinnen. Wir werden dies vornehmlich in den Übungen zur Vorlesung abhandeln.

1.1 Ein Beispiel

Was ist ein Optimierungsproblem? Bevor wir diese Frage durch eine allgemeine Definition beantworten, wollen wir ein Beispiel ausführlich besprechen, um zu zeigen, wie Optimierungsprobleme entstehen, und wie man sie mathematisch behandeln kann. Wir beginnen mit einem (natürlich für Vorlesungszwecke aufbereiteten) Beispiel aus der Ölindustrie.

(1.1) Beispiel (Ölraffinierung). In Ölraffinerien wird angeliefertes Rohöl durch

Anwendung von chemischen und (oder) physikalischen Verfahren in gewisse gewünschte Komponenten zerlegt. Die Ausbeute an verschiedenen Komponenten hängt von dem eingesetzten Verfahren (Crackprozess) ab. Wir nehmen an, dass eine Raffinerie aus Rohöl drei Komponenten (schweres Öl S, mittelschweres Öl M, leichtes Öl L) herstellen will. Sie hat zwei Crackverfahren zur Verfügung, die die folgende Ausbeute liefern und die die angegebenen Kosten (Energie, Maschinenabschreibung, Arbeit) verursachen. (Zur Verkürzung der Schreibweise kürzen wir das Wort "Mengeneinheit" durch ME und das Wort "Geldeinheit" durch GE ab. 10 ME Rohöl ergeben in:

$$\begin{array}{llll}
\text{Crackprozess 1 :} & 2\,\mathrm{ME} \quad S & & \\
 & 2\,\mathrm{ME} \quad M & & \text{Kosten: } 3\,\mathrm{GE} \\
 & 1\,\mathrm{ME} \quad L & & \\
 & & & \\
\text{Crackproze\ss{} 2 :} & 1\,\mathrm{ME} \quad S & & \\
 & 2\,\mathrm{ME} \quad M & & \text{Kosten: } 5\,\mathrm{GE} \\
 & 4\,\mathrm{ME} \quad L & &
\end{array}$$

Aufgrund von Lieferverpflichtungen muss die Raffinerie folgende Mindestproduktion herstellen:

$$\begin{array}{ll}
3\,\mathrm{ME} & S \\
5\,\mathrm{ME} & M \\
4\,\mathrm{ME} & L
\end{array}$$

Diese Mengen sollen so kostengünstig wie möglich produziert werden.

$\square$

Wir wollen nun die obige Aufgabe (1.1) mathematisch formulieren. Unser Ergebnis wird eine mathematische Aufgabe sein, die wir später als lineares Programm oder lineares Optimierungsproblem bezeichnen werden.

Zunächst stellen wir fest, dass die Crackprozesse unabhängig voneinander arbeiten und (im Prinzip) mit jeder beliebigen Rohölmenge beschickt werden können. Wir führen daher zwei Variable x_1 und x_2 ein, die das Produktionsniveau der beiden Prozesse beschreiben. Wird z. B. $x_1 = 1.5$ gewählt, so bedeutet dies, dass der Crackprozess 1 mit 15 ME Rohöl beschickt wird, während bei $x_2 = 0.5$ der Crackprozess 2 nur mit 5 ME Rohöl gefahren wird. Natürlich macht es keinen Sinn, einen Crackprozess mit negativen Rohölmengen zu beschicken, aber jeder Vektor $x = (x_1, x_2) \in \mathbb{R}^2$, $x \geq 0$, beschreibt ein mögliches Produktionsniveau der beiden Prozesse.

Angenommen durch $(x_1, x_2) \geq 0$ sei ein Produktionsniveau beschrieben, dann folgt aus den technologischen Bedingungen der beiden Crackprozesse, dass sich der Ausstoß an schwerem Öl S auf

$$2x_1 + x_2 \quad \text{ME} \quad S$$

beläuft, und analog erhalten wir

$$\begin{array}{llll} 2x_1 & +2x_2 & \text{ME} & M \\ x_1 & +4x_2 & \text{ME} & L. \end{array}$$

Die Kosten betragen offenbar

$$z = 3x_1 + 5x_2 \quad \text{GE}.$$

Die eingegangenen Lieferverpflichtungen besagen, dass gewisse Mindestmengen an S, M und L produziert werden müssen, das heißt, ein Vektor (x_1, x_2) beschreibt nur dann ein Produktionsniveau, bei dem auch alle Lieferverpflichtungen erfüllt werden können, wenn gilt:

$$(1.2) \qquad \begin{array}{rl} 2x_1 + x_2 & \geq 3 \\ 2x_1 + x_2 & \geq 5 \\ x_1 + 4x_2 & \geq 4 \\ x_1 & \geq 0 \\ x_2 & \geq 0 \end{array}$$

Unsere Aufgabe besteht nun darin, unter allen Vektoren $(x_1, x_2) \in \mathbb{R}^2$, die die 5 Ungleichungen in (1.2) erfüllen, einen Vektor zu finden, der minimale Kosten verursacht, d. h. bei dem

$$z = 3x_1 + 5x_2$$

einen möglichst kleinen Wert annimmt.

Es hat sich eingebürgert, eine derartige Aufgabe ein **lineares Programm** oder ein **lineares Optimierungsproblem** zu nennen, und es wie folgt zu schreiben:

$$\min\ 3x_1 + 5x_2$$

unter den Nebenbedingungen:

$$(1.3) \qquad \begin{array}{lrl} (1) & 2x_1 + x_2 \geq 3 & (S\text{-Ungleichung}) \\ (2) & 2x_1 + 2x_2 \geq 5 & (M\text{-Ungleichung}) \\ (3) & x_1 + 4x_2 \geq 4 & (L\text{-Ungleichung}) \\ (4) & x_1 \geq 0 & \\ (5) & x_2 \geq 0 & \end{array}$$

Dabei nennt man die zu minimierende Funktion $3x_1 + 5x_2$ die **Zielfunktion** des Problems. Jeder Vektor, der $(1), \ldots, (5)$ erfüllt, heißt **zulässige Lösung** des Problems.

Da wir nur 2 Variablen haben, können wir unsere Aufgabe graphisch analysieren. Ersetzen wir das "$\geq$"-Zeichen in $(1), \ldots, (5)$ durch ein Gleichheitszeichen, so erhalten wir 5 Gleichungen. Die Lösungsmenge einer Gleichung im $\mathbb{R}^2$ ist bekanntlich eine Gerade. Diese 5 Geraden "beranden" die Lösungsmenge des Systems der Ungleichungen $(1), \ldots, (5)$. Diese Lösungsmenge ist in Abbildung 1.1 graphisch dargestellt. (Der Leser möge sich unbedingt mit der geometrischen Interpretation von Ungleichungen vertraut machen.)

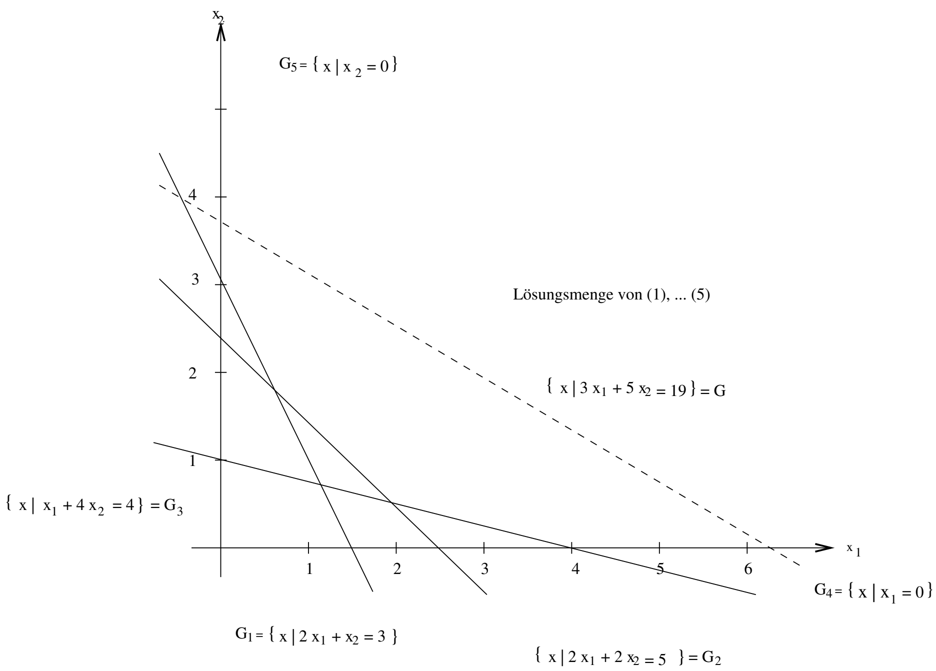


Abb. 1.1

Die Zielfunktion ist natürlich keine Gerade, sie repräsentiert eine Schar paralleler Geraden. Nehmen wir z. B. den Vektor $x = (3, 2)$, der alle Ungleichungen $(1), \ldots, (5)$ erfüllt. Sein Zielfunktionswert ist 19, d. h. bei diesem Produktionsniveau treten Gesamtkosten in Höhe von 19 GE auf. In Abbildung 1.1 ist die Gerade $G = \{x \mid 3x_1 + 5x_2 = 19\}$ gestrichelt gezeichnet. Die Menge aller derjenigen Punkte, die auf dieser Geraden liegen und $(1), \ldots, (5)$ erfüllen, stellen Produktionsniveaus mit Gesamtkosten 19 GE dar.

Geometrisch ist nun klar, wie wir einen Punkt finden können, der $(1), \ldots, (5)$ erfüllt und die Zielfunktion minimiert. Wir verschieben die Gerade G so lange parallel in Richtung auf den Punkt $(0, 0)$, bis die verschobene Gerade die Lösungsmenge von $(1), \ldots, (5)$ nur noch tangiert. Führen wir dies graphisch durch, so sehen wir, dass wir die Tangentialstellung im Punkt $x^* = (2, \frac{1}{2})$ erreichen. Die zu G parallele Gerade

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

berührt die Lösungsmenge von $(1), \ldots, (5)$ in nur einem Punkt, nämlich x^*, jede weitere Parallelverschiebung würde zu einem leeren Durchschnitt mit dieser Lösungsmenge führen. Wir schließen daraus, dass

$$x^* = (2, \tfrac{1}{2})$$

die Optimallösung unseres Problems ist, d. h. alle Lieferverpflichtungen können bei diesem Produktionsniveau erfüllt werden, und alle anderen Produktionsniveaus führen zu Gesamtkosten, die höher sind als die Kosten von 8.5 GE, die beim Produktionsniveau x^* anfallen.

Die vorgeführte Lösung des Beispiels (1.1) ist anschaulich und sieht recht einfach aus. Aber es ist klar, dass sich diese graphische Argumentation nicht für mehr als zwei Variable durchführen lässt. Die graphische Methode ist zwar nützlich, um die geometrische Intuition anzuregen, aber zur Lösung von linearen Programmen taugt sie nicht. Und, können unsere Schluss folgerungen als ein korrekter Beweis für die Optimalität von x^* angesehen werden?

Wir wollen daher einen zweiten Versuch zur Lösung von (1.1) unternehmen und betrachten das Problem etwas allgemeiner. Wählen wir statt der speziellen Zielfunktion $3x_1 + 5x_2$ eine beliebige, sagen wir

$$ax_1 + bx_2,$$

so sehen wir, dass unser Problem gar keine Optimallösung haben muss. Ist nämlich einer der Parameter negativ, sagen wir $a < 0$, so können wir den Zielfunktionswert so klein machen, wie wir wollen. Z. B. gilt für die Folge der Punkte $x^{(n)} := (n, 0)$, $n \in \mathbb{N}$,

$$\begin{aligned} x^{(n)} &\quad \text{erfüllt} \quad (1), \ldots, (5) \quad \text{für alle } n \geq 4, \\ ax_1^{(n)} + bx_2^{(n)} &= na. \end{aligned}$$

Also ist der Wert der Zielfunktion über der Lösungsmenge $(1), \ldots, (5)$ nicht nach unten beschränkt. Von unserem Beispiel ausgehend ist eine derartige Zielfunktion natürlich unsinnig, aber mathematisch müssen derartige Fälle behandelt und klassifiziert werden. Kommt so etwas in der Praxis vor? Natürlich, und zwar öfter als man denkt! Gründe hierfür liegen in der Regel bei fehlerhafter Dateneingabe, falschem Zugriff auf Datenbanken oder irgendwelchen Übertragungsfehlern. Auch Lösungsmengen, die aufgrund der eigentlichen Daten nicht leer sein sollten, können auf diese Weise plötzlich leer sein.

Sind a und b nichtnegativ, so ist klar, dass es eine Optimallösung gibt. Führen wir unser Parallelverschiebungsexperiment mehrmals durch, so werden wir feststellen, dass wir immer eine Optimallösung finden können, die den Durchschnitt von zwei der fünf Geraden $G_1, \ldots, G_5$ bildet. Man könnte also vermuten, dass dies immer so ist, d. h. dass immer Optimallösungen gefunden werden können, die (bei n Variablen) den Durchschnitt von n Hyperebenen bilden.

Wir werden später sehen, dass dies (bis auf Pathologien) in der Tat der Fall ist, und dass dies eine der fruchtbarsten Beobachtungen in Bezug auf die Lösungstheorie von Problemen des Typs (1.1) ist.

Die Lösungsmenge von (1.3) $(1), \ldots, (5)$ enthält unendlich viele Punkte. Unsere obige Überlegung hat die Suche auf endlich viele Punkte reduziert, und diese Punkte können wir sogar effektiv angeben. Wir schreiben die Ungleichungen $(1), \ldots, (5)$ als Gleichungen und betrachten alle möglichen 10 Systeme von 2 Gleichungen (in 2 Variablen) dieses Systems. Jedes dieser 10 Gleichungssysteme können wir mit Hilfe der Gauß-Elimination lösen. Z. B. ist der Durchschnitt von G_1 und G_3 der Punkt $y_1 = (\frac{8}{7}, \frac{5}{7})$, und der Durchschnitt von G_1 und G_2 der Punkt $y_2 = (\frac{1}{2}, 2)$. Der Punkt y_1 erfüllt jedoch die Ungleichung (2) nicht. Wir müssen also zunächst alle Durchschnitte ausrechnen, dann überprüfen, ob der jeweils errechnete Punkt tatsächlich alle Ungleichungen erfüllt, und unter den so gefundenen Punkten den besten auswählen.

Dieses Verfahren ist zwar endlich, aber bei m Ungleichungen und n Variablen erfordert es $\binom{m}{n}$ Aufrufe der Gauß-Elimination und weitere Überprüfungen. Es ist also praktisch nicht in vernünftiger Zeit ausführbar. Dennoch steckt hierin der Kern des heutzutage wichtigsten Verfahrens zur Lösung von linearen Programmen, des Simplexverfahrens. Man berechnet zunächst eine Lösung, die im Durchschnitt von n Hyperebenen liegt, und dann versucht man "systematisch" weitere derartige Punkte zu produzieren und zwar dadurch, dass man jeweils nur eine Hyperebene durch eine andere austauscht und so von einem Punkt auf "zielstrebige" Weise über "Nachbarpunkte" zum Optimalpunkt gelangt. Überlegen Sie sich einmal selbst, wie Sie diese Idee verwirklichen würden!

Eine wichtige Komponente des Simplexverfahrens beruht auf der Tatsache, dass man effizient erkennen kann, ob ein gegebener Punkt tatsächlich der gesuchte Optimalpunkt ist oder nicht. Dahinter steckt die sogenannte Dualitätstheorie, die wir anhand unseres Beispiels (1.1) erläutern wollen. Wir wollen also zeigen, dass der Punkt $x^* = (2, \frac{1}{2})$ minimal ist.

Ein Weg, dies zu beweisen, besteht darin, untere Schranken für das Minimum zu finden. Wenn man sogar in der Lage ist, eine untere Schranke zu bestimmen, die mit dem Zielfunktionswert einer Lösung des Ungleichungssystems übereinstimmt, ist man fertig (denn der Zielfunktionswert irgendeiner Lösung ist immer eine obere Schranke, und wenn untere und obere Schranke gleich sind, ist der optimale Wert gefunden).

Betrachten wir also unser lineares Program (1.3). Wir bemerken zunächst, dass man alle Ungleichungen $(1), \dots, (5)$ skalieren kann, d. h. wir können z. B. die Ungleichung

(1) $2x_1 + x_2 \geq 3$

mit 2 multiplizieren und erhalten:

$(1')$ $4x_1 + 2x_2 \geq 6.$

Die Menge der Punkte des $\mathbb{R}^2$, die (1) erfüllt, stimmt offenbar überein mit der Menge der Punkte, die $(1')$ erfüllt. Wir können auch mit negativen Faktoren skalieren, aber dann dreht sich das Ungleichungszeichen um!

Erfüllt ein Vektor zwei Ungleichungen, so erfüllt er auch die Summe der beiden Ungleichungen. Das gilt natürlich nur, wenn die Ungleichungszeichen gleichgerichtet sind. Addieren wir z. B. zur Ungleichung (1) die Ungleichung (3), so erhalten wir

$$3x_1 + 5x_2 \geq 7.$$

Die linke Seite dieser Ungleichung ist (natürlich nicht rein zufällig) gerade unsere Zielfunktion. Folglich können wir aus der einfachen Überlegung, die wir gerade gemacht haben, schließen, dass jeder Vektor, der (1.3) $(1), \dots, (5)$ erfüllt, einen Zielfunktionswert hat, der mindestens 7 ist. Der Wert 7 ist somit eine untere Schranke für das Minimum in (1.3). Eine tolle Idee, oder?

Das Prinzip ist damit dargestellt. Wir skalieren die Ungleichungen unseres Systems und addieren sie, um untere Schranken zu erhalten. Da unsere Ungleichungen alle in der Form "$\geq$" geschrieben sind, dürfen wir nur positiv skalieren, denn zwei Ungleichungen $a_1x_1 + a_2x_2 \leq \alpha$ und $b_1x_1 + b_2x_2 \geq \beta$ kann man nicht addieren. Gibt jede beliebige Skalierung und Addition eine untere Schranke? Machen wir noch zwei Versuche. Wir addieren (1) und (2) und erhalten

$$4x_1 + 3x_2 \geq 8.$$

Das hilft uns nicht weiter, denn die linke Seite der Ungleichung kann nicht zum Abschätzen der Zielfunktion benutzt werden. Betrachten wir nun, das 1.5-fache der Ungleichung (2), so ergibt sich

$$3x_1 + 3x_2 \geq 7.5.$$

Hieraus schließen wir: Ist $3x_1 + 3x_2 \geq 7.5$, dann ist erst recht $3x_1 + 5x_2 \geq 7.5$, denn nach (5) gilt ja $x_2 \geq 0$.

Daraus können wir eine Regel für die Bestimmung unterer Schranken ableiten. Haben wir eine neue Ungleichung als Summe von positiv skalierten Ausgangsungleichungen erhalten und hat die neue Ungleichung die Eigenschaft, dass jeder ihrer Koeffizienten höchstens so groß ist wie der entsprechende Koeffizient der Zielfunktion, so liefert die rechte Seite der neuen Ungleichung eine untere Schranke für den Minimalwert von (1.3).

Diese Erkenntnis liefert uns ein neues mathematisches Problem. Wir suchen nichtnegative Multiplikatoren (Skalierungsfaktoren) der Ungleichungen (1), (2), (3) mit gewissen Eigenschaften. Multiplizieren wir (1) mit y_1, (2) mit y_2 und (3) mit y_3, so darf die Summe $2y_1 + 2y_2 + y_3$ nicht den Wert des ersten Koeffizienten der Zielfunktion (also 3) überschreiten. Analog darf die Summe $y_1 + 2y_2 + 4y_3$ nicht den Wert 5 überschreiten. Und die y_i sollen nicht negativ sein. Ferner soll die rechte Seite der Summenungleichung so groß wie möglich werden. Die rechte Seite kann man durch $3y_1 + 5y_2 + 4y_3$ berechnen. Daraus folgt, dass wir die folgende Aufgabe lösen müssen. Bestimme

$$\max\ 3y_1 + 5y_2 + 4y_3$$

unter den Nebenbedingungen:

$$\begin{aligned} 2y_1 + 2y_2 + \ \ y_3 &\leq 3 \\ y_1 + 2y_2 + 4y_3 &\leq 5 \\ y_1, y_2, y_3 &\geq 0 \end{aligned} \tag{1.4}$$

Auch (1.4) ist ein lineares Programm. Aus unseren Überlegungen folgt, dass der Maximalwert von (1.4) höchstens so groß ist wie der Minimalwert von (1.3), da jede Lösung von (1.4) eine untere Schranke für (1.3) liefert. Betrachten wir z. B. den Punkt

$$y^* = (y_1^*, y_2^*, y_3^*) = (0, \tfrac{7}{6}, \tfrac{2}{3}).$$

Durch Einsetzen in die Ungleichungen von (1.4) sieht man dass y^* alle Ungleichungen erfüllt. Addieren wir also zum $\frac{7}{6}$-fachen der Ungleichung (2) das $\frac{2}{3}$-fache der Ungleichung (3), so erhalten wir

$$3x_1 + 5x_2 \geq 8.5\,.$$

Damit wissen wir, dass der Minimalwert von (1.3) mindestens 8.5 ist. Der Punkt x^* liefert gerade diesen Wert, er ist also optimal. Und außerdem ist y^* eine Optimallösung von (1.4).

Die hier dargestellten Ideen bilden die Grundgedanken der Dualitätstheorie der linearen Programmierung, und sie sind außerordentlich nützlich bei der Entwicklung von Algorithmen und in Beweisen. In der Tat haben wir bereits ein kleines Resultat erzielt, das wir wie folgt formal zusammenfassen wollen.

(1.5) Satz. *Es seien $c \in \mathbb{R}^n$, $b \in \mathbb{R}^m$, und A sei eine reelle (m, n)-Matrix. Betrachten wir die Aufgaben*

(P) $\min \quad c^T x \quad$ *unter* $\quad Ax \geq b, \quad x \geq 0,$

(lies: bestimme einen Vektor x^*, der $Ax^* \geq b$, $x^* \geq 0$ erfüllt und dessen Wert $c^T x^*$ so klein wie möglich ist),

(D) $\max \quad y^T b \quad$ *unter* $\quad y^T A \leq c^T, \quad y \geq 0$

(lies: bestimme einen Vektor y^*, der $y^{*T} A \leq c^T$, $y^* \geq 0$ erfüllt und dessen Wert $y^{*T} b$ so groß wie möglich ist),

dann gilt Folgendes:

Seien $x_0 \in \mathbb{R}^n$ und $y_0 \in \mathbb{R}^m$ Punkte mit $Ax_0 \geq b$, $x_0 \geq 0$ und $y_0^T A \leq c^T$, $y_0 \geq 0$, dann gilt

$$y_0^T b \leq c^T x_0.$$

Beweis : Durch einfaches Einsetzen:

$$y_0^T b \leq y_0^T (Ax_0) = (y_0^T A) x_0 \leq c^T x_0.$$

□

Satz (1.5) wird **schwacher Dualitätssatz** genannt. Für Optimallösungen x^* und y^* von (P) bzw. (D), gilt nach (1.5) $y^{*T} b \leq c^T x^*$. Wir werden später zeigen, dass in diesem Falle sogar immer $y^{*T} b = c^T x^*$ gilt. Dies ist allerdings nicht ganz so einfach zu beweisen wie der obige Satz (1.5).

Unser Beispiel (1.1) stammt aus einer ökonomischen Anwendung. Das lineare Programm (1.3) haben wir daraus durch mathematische Formalisierung abgeleitet und daraus durch mathematische Überlegungen ein neues lineares Programm (1.4) gewonnen. Aber auch dieses Programm hat einen ökonomischen Sinn.

(1.6) Ökonomische Interpretation von (1.4). Als Manager eines Ölkonzerns sollte man darüber nachdenken, ob es überhaupt sinnvoll ist, die drei Ölsorten S,

M, L selbst herzustellen. Vielleicht ist es gar besser, die benötigten Ölmengen auf dem Markt zu kaufen. Die Manager müssen sich also überlegen, welche Preise sie auf dem Markt für S, M, L gerade noch bezahlen würden, ohne die Produktion selbst aufzunehmen. Oder anders ausgedrückt, bei welchen Marktpreisen für S, M, L lohnt sich die Eigenproduktion gerade noch.

Bezeichnen wir die Preise für eine Mengeneinheit von S, M, L mit y_1, y_2, y_3, so sind beim Ankauf der benötigten Mengen für S, M, L genau $3y_1 + 5y_2 + 4y_3$ GE zu bezahlen. Der erste Crackprozess liefert bei 10 ME Rohöleinsatz 2 ME S, 2 ME M und 1 ME L und verursacht 3 GE Kosten. Würden wir den Ausstoss des ersten Crackprozesses auf dem Markt kaufen, so müssten wir dafür $2y_1 + 2y_2 + y_3$ GE bezahlen. Ist also $2y_1 + 2y_2 + y_3 > 3$, so ist es auf jeden Fall besser, den Crackprozess 1 durchzuführen als auf dem Markt zu kaufen. Analog gilt für den zweiten Prozess: Erfüllen die Marktpreise die Bedingung $y_1 + 2y_2 + 4y_3 > 5$, so ist die Durchführung des Crackprozesses 2 profitabel.

Nur solche Preise y_1, y_2, y_3, die die Bedingungen

$$\begin{aligned} 2y_1 + 2y_2 + \ \ y_3 &\leq 3 \\ y_1 + 2y_2 + 4y_3 &\leq 5 \\ y_1, y_2, y_3 \qquad &\geq 0 \end{aligned}$$

erfüllen, können also das Management veranlassen, auf dem Markt zu kaufen, statt selbst zu produzieren. Der beim Kauf der durch Lieferverträge benötigten Mengen zu bezahlende Preis beläuft sich auf

$$3y_1 + 5y_2 + 4y_3.$$

Wenn wir diese Funktion unter den obigen Nebenbedingungen maximieren, erhalten wir Preise (in unserem Falle $y_1^* = 0$, $y_2^* = \frac{7}{6}$, $y_3^* = \frac{2}{3}$), die die Eigenproduktion gerade noch profitabel erscheinen lassen. Diese Preise werden **Schattenpreise** genannt. Sie sagen nichts über den tatsächlichen Marktpreis aus, sondern geben in gewisser Weise den Wert des Produktes für den Hersteller an.

Das Resultat $y_1^* = 0$ besagt z. B., dass das Schweröl S für den Hersteller wertlos ist. Warum? Das liegt daran, dass bei der optimalen Kombination der Crackprozesse mehr Schweröl anfällt (nämlich 4.5 ME) als benötigt wird (3 ME). Das heißt, bei jeder Lieferverpflichtung für Schweröl, die zwischen 0 und 4.5 ME liegt, würde sich die optimale Prozess kombination nicht ändern, und die gesamten Kosten wären in jedem Falle die gleichen. Ein gutes Management wird in einer solchen Situation die Verkaufsmitarbeiter dazu anregen, Schweröl zu verkaufen. Bei jedem beliebigen Verkaufspreise winkt (innerhalb eines begrenzten Lieferumfangs) Gewinn! Und es gibt weniger Entsorgungsprobleme. Müsste der Hersteller dagegen eine ME mehr an leichtem Öl L liefern, so müsste die Kombination der Crackprozesse geändert werden, und zwar würden dadurch insgesamt Mehrkosten in Höhe von $\frac{2}{3}$ GE anfallen. Analog würde bei einer Erhöhung der Produktion von 5 ME mittleren Öls auf 6 ME eine Kostensteigerung um $\frac{7}{6}$ GE erfolgen.

Diesen Sachverhalt können wir dann auch so interpretieren. Angenommen, die Ölfirma hat die Prozess niveaus $x_1^* = 2$, $x_2^* = 0.5$ gewählt. Ein Mineralölhändler möchte 1 ME leichten Öls L zusätzlich kaufen. Ist der Marktpreis mindestens $\frac{2}{3}$ GE pro ME L wird die Firma das Geschäft machen, andernfalls wäre es besser, die zusätzliche Einheit nicht selbst zu produzieren sondern von einem anderen Hersteller zu kaufen. Würde der Mineralölhändler 1 ME schweres Öl nachfragen, würde die Firma bei jedem Preis verkaufen, denn sie hat davon zu viel.

$\square$

1.2 Optimierungsprobleme

Wir wollen zunächst ganz informell, ohne auf technische Spitzfindigkeiten einzugehen, Optimierungsprobleme einführen. Sehr viele Probleme lassen sich wie folgt formulieren.

Gegeben seien eine Menge S und eine geordnete Menge $(T, \leq)$, d. h. zwischen je zwei Elementen $s, t \in T$ gilt genau eine der folgenden Beziehungen $s < t$, $s > t$ oder $s = t$. Ferner sei eine Abbildung $f : S \to T$ gegeben. Gesucht ist ein Element $x^* \in S$ mit der Eigenschaft $f(x^*) \geq f(x)$ für alle $x \in S$ (Maximierungsproblem) oder $f(x^*) \leq f(x)$ für alle $x \in S$ (Minimierungsproblem). Es ist üblich hierfür eine der folgenden Schreibweisen zu benutzen:

$$(1.7) \qquad \begin{array}{lll} \max_{x \in S} f(x) & \text{oder} & \max\{f(x) \mid x \in S\}, \\ \min_{x \in S} f(x) & \text{oder} & \min\{f(x) \mid x \in S\}. \end{array}$$

In der Praxis treten als geordnete Mengen $(T, \leq)$ meistens die reellen Zahlen $\mathbb{R}$, die rationalen Zahlen $\mathbb{Q}$ oder die ganzen Zahlen $\mathbb{Z}$ auf, alle mit der natürlichen Ordnung versehen (die wir deswegen auch gar nicht erst notieren). Die Aufgabe (1.7) ist viel zu allgemein, um darüber etwas Interessantes sagen zu können. Wenn S durch die Auflistung aller Elemente gegeben ist, ist das Problem entweder sinnlos oder trivial (man rechnet ganz einfach $f(x)$ für alle $x \in S$ aus). Das heißt, S muss irgendwie (explizit oder implizit) strukturiert sein, so dass vernünftige Aussagen über S möglich sind, ohne dass man alle Elemente in S einzeln kennt. Das gleiche gilt für die Funktion $f : S \to T$. Ist sie nur punktweise durch $x \mapsto f(x)$

gegeben, lohnt sich das Studium von (1.7) nicht. Erst wenn f durch hinreichend strukturierte “Formeln” bzw. “Eigenschaften” bestimmt ist, weden tieferliegende mathematische Einsichten möglich.

Die Optimierungsprobleme, die in der Praxis auftreten, haben fast alle irgendeine “vernünftige” Struktur. Das muss nicht unbedingt heißen, dass die Probleme dadurch auf einfache Weise lösbar sind, aber immerhin ist es meistens möglich, sie in das zur Zeit bekannte und untersuchte Universum der verschiedenen Typen von Optimierungsproblemen einzureihen und zu klassifizieren.

Im Laufe des Studiums werden Ihnen noch sehr unterschiedliche Optimierungsaufgaben begegnen. Viele werden von einem der folgenden Typen sein.

(1.8) Ein Kontrollproblem. Gegeben sei ein Steuerungsprozess (z. B. die Bewegungsgleichung eines Autos), etwa der Form

$$\dot{x}(t) = f(t, x(t), u(t)),$$

wobei u eine Steuerung ist (Benzinzufuhr). Ferner seien eine Anfangsbedingung

$$x(0) = x_0,$$

(z. B.: das Auto steht) sowie eine Endbedingung

$$x(T) = x_1$$

(z.B. das Auto hat eine Geschwindigkeit von 50 km/h) gegeben. Gesucht ist eine Steuerung u für den Zeitraum $[0, T]$, so dass z. B.

$$\int_0^T |u|^2 \,\mathrm{dt}$$

minimal ist (etwa minimaler Benzinverbrauch).

$\square$

(1.9) Ein Approximationsproblem. Gegeben sei eine (numerisch schwierig auszuwertende) Funktion f, finde eine Polynom p vom Grad n, so dass

$$\|f - p\| \text{ oder } \|f - p\|_\infty$$

minimal ist.

$\square$

(1.10) Nichtlineares Optimierungsproblem. Es seien f, g_i $(i = 1, \dots, m)$, h_j $(j = 1, \dots, p)$ differenzierbare Funktionen von $\mathbb{R}^n \to \mathbb{R}$, dann heißt

$$\begin{aligned} &\min f(x) \\ &g_i(x) \leq 0 \quad i = 1, \dots, m \\ &h_i(x) = 0 \quad j = 1, \dots, p \\ &\quad x \in \mathbb{R}^n \end{aligned}$$

ein nichtlineares Optimierungsproblem. Ist eine der Funktionen nicht differenzierbar, so spricht man von einem nichtdifferenzierbaren Optimierungsproblem. Im Allgemeinen wird davon ausgegangen, daß alle betrachteten Funktionen zumindest stetig sind.

$\square$

(1.11) Konvexes Optimierungsproblem. Eine Menge $S \subseteq \mathbb{R}^n$ heißt **konvex**, falls gilt: Sind $x, y \in S$ und ist $\lambda \in \mathbb{R}$, $0 \leq \lambda \leq 1$, dann gilt $\lambda x + (1 - \lambda)y \in S$. Eine Funktion $f : \mathbb{R}^n \to \mathbb{R}$ heißt **konvex**, falls für alle $\lambda \in \mathbb{R}$, $0 \leq \lambda \leq 1$ und alle $x, y \in \mathbb{R}^n$ gilt

$$\lambda f(x) + (1 - \lambda)f(y) \geq f(\lambda x + (1 - \lambda)y).$$

Ist $S \subseteq \mathbb{R}^n$ konvex (z. B. kann S wie folgt gegeben sein $S = \{x \in \mathbb{R}^n \mid g_i(x) \leq 0,\ i = 1, \dots, m\}$ wobei die g_i konvexe Funktionen sind), und ist $f : \mathbb{R}^n \to \mathbb{R}$ eine konvexe Funktion, dann ist

$$\min_{x \in S} f(x)$$

ein konvexes Minimierungsproblem.

$\square$

(1.12) Lineares Optimierungsproblem (Lineares Programm).

Gegeben seien $c \in \mathbb{R}^n$, $A \in \mathbb{R}^{(m,n)}$, $b \in \mathbb{R}^m$, dann heißt

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

lineares Optimierungsproblem.

$\square$

(1.13) Lineares ganzzahliges Optimierungsproblem.

Gegeben seien $c \in \mathbb{R}^n$, $A \in \mathbb{R}^{(m,n)}$, $b \in \mathbb{R}^m$, dann heißt

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

lineares ganzzahliges (oder kurz: ganzzahliges) Optimierungsproblem.

□

Selbst bei Optimierungsproblemen wie $(1.8), \ldots, (1.13)$, die nicht sonderlich allgemein erscheinen mögen, kann es sein, dass (bei spezieller Wahl der Zielfunktion f und der Nebenbedingungen) die Aufgabenstellung (finde ein $x^* \in S$, so dass $f(x^*)$ so groß (oder klein) wie möglich ist) keine vernünftige Antwort besitzt. Es mag sein, dass f über S unbeschränkt ist; f kann beschränkt sein über S, aber ein Maximum kann innerhalb S nicht erreicht werden, d.h., das "max" müßte eigentlich durch "sup" ersetzt werden. S kann leer sein, ohne dass dies a priori klar ist, etc.. Der Leser möge sich Beispiele mit derartigen Eigenschaften überlegen! Bei der Formulierung von Problemen dieser Art muss man sich also Gedanken darüber machen, ob die betrachtete Fragestellung überhaupt eine sinnvolle Antwort erlaubt.

In unserer Vorlesung werden wir uns lediglich mit den Problemen (1.12) und (1.13) beschäftigen. Das lineare Optimierungsproblem (1.12) ist sicherlich das derzeit für die Praxis bedeutendste Problem, da sich außerordentlich viele und sehr unterschiedliche reale Probleme als lineare Programme formulieren lassen. Außerdem liegt eine sehr ausgefeilte Theorie vor, und mit den modernen Verfahren der linearen Optimierung können derartige Probleme mit Hunderttausenden (und manchmal sogar mehr) von Variablen und Ungleichungen gelöst werden. Dagegen ist Problem (1.13), viel schwieriger. Die Einschränkung der Lösungsmenge auf die zulässigen ganzzahligen Lösungen führt direkt zu einem Sprung im Schwierigkeitsgrad des Problems. Verschiedene spezielle lineare ganzzahlige Programme können in beliebiger Größenordnung gelöst werden. Bei wenig strukturierten allgemeinen Problemen des Typs (1.13) versagen dagegen die Lösungsverfahren manchmal bereits bei weniger als 100 Variablen und Nebenbedingungen.

Über Kontrolltheorie (Probleme des Typs (1.8)), Approximationstheorie (Probleme des Typs (1.9)), Nichtlineare Optimierung (Probleme des Typs (1.10) und (1.11) werden an der TU Berlin Spezialvorlesungen angeboten. Es ist anzumerken, dass sich sowohl die Theorie als auch die Algorithmen zur Lösung von Problemen des Typs (1.8) bis (1.11) ganz erheblich von denen zur Lösung von Problemen des Typs (1.12) und (1.13) unterscheiden.

1.3 Notation

Wir gehen in der Vorlesung davon aus, dass die Hörer die Grundvorlesungen Lineare Algebra und Analysis kennen. Wir werden hauptsächlich Methoden der linearen Algebra benutzen und setzen eine gewisse Vertrautheit mit der Vektor- und Matrizenrechnung voraus.

1.3.1 Grundmengen

Wir benutzen folgende Bezeichnungen:

$\mathbb{N} = \{1, 2, 3, \ldots\} =$ Menge der **natürlichen Zahlen,**
$\mathbb{Z} =$ Menge der **ganzen Zahlen,**
$\mathbb{Q} =$ Menge der **rationalen Zahlen,**
$\mathbb{R} =$ Menge der **reellen Zahlen,**

und mit M_+ bezeichnen wir die Menge der nichtnegativen Zahlen in M für $M \in \{\mathbb{Z}, \mathbb{Q}, \mathbb{R}\}$. Wir betrachten $\mathbb{Q}$ und $\mathbb{R}$ als Körper mit der üblichen Addition und Multiplikation und der kanonischen Ordnung "$\leq$". Desgleichen betrachten wir $\mathbb{N}$ und $\mathbb{Z}$ als mit den üblichen Rechenarten versehen. Wir werden uns fast immer in diesen Zahlenuniversen bewegen, da diese die für die Praxis relevanten sind. Manche Sätze gelten jedoch nur, wenn wir uns auf $\mathbb{Q}$ oder $\mathbb{R}$ beschränken. Um hier eine saubere Trennung zu haben, treffen wir die folgende Konvention. Wenn wir das Symbol

$$\mathbb{K}$$

benutzen, so heißt dies immer das $\mathbb{K}$ einer der angeordneten Körper $\mathbb{R}$ oder $\mathbb{Q}$ ist. Sollte ein Satz nur für $\mathbb{R}$ oder nur für $\mathbb{Q}$ gelten, so treffen wir die jeweils notwendige Einschränkung.

Für diejenigen, die sich für möglichst allgemeine Sätze interessieren, sei an dieser Stelle folgendes vermerkt. Jeder der nachfolgend angegebenen Sätze bleibt ein wahrer Satz, wenn wir als Grundkörper $\mathbb{K}$ einen archimedisch angeordneten

Körper wählen. Ein bekannter Satz besagt, dass jeder archimedisch angeordnete Körper isomorph zu einem Unterkörper von $\mathbb{R}$ ist, der $\mathbb{Q}$ enthält. Unsere Sätze bleiben also richtig, wenn wir statt $\mathbb{K} \in \{\mathbb{Q}, \mathbb{R}\}$ irgendeinen archimedisch angeordneten Körper $\mathbb{K}$ mit $\mathbb{Q} \subseteq \mathbb{K} \subseteq \mathbb{R}$ wählen.

Wir können in fast allen Sätzen (insbesondere bei denen, die keine Ganzzahligkeitsbedingungen haben) auch die Voraussetzung "archimedisch" fallen lassen, d. h. fast alle Sätze gelten auch für angeordnete Körper. Vieles, was wir im $\mathbb{K}^n$ beweisen, ist auch in beliebigen metrischen Räumen oder Räumen mit anderen als euklidischen Skalarprodukten richtig. Diejenigen, die Spaß an derartigen Verallgemeinerungen haben, sind eingeladen, die entsprechenden Beweise in die allgemeinere Sprache zu übertragen.

In dieser Vorlesung interessieren wir uns für so allgemeine Strukturen nicht. Wir verbleiben in den (für die Praxis besonders wichtigen) Räumen, die über den reellen oder rationalen Zahlen errichtet werden. Also, nochmals, wenn immer wir das Symbol $\mathbb{K}$ im weiteren gebrauchen, gilt

$$\mathbb{K} \in \{\mathbb{R}, \mathbb{Q}\},$$

und $\mathbb{K}$ ist ein Körper mit den üblichen Rechenoperationen und Strukturen. Natürlich ist $\mathbb{K}_+ = \{x \in \mathbb{K} \mid x \geq 0\}$

Die Teilmengenbeziehung zwischen zwei Mengen M und N bezeichnen wir wie üblich mit $M \subseteq N$. Gilt $M \subseteq N$ und $M \neq N$, so schreiben wir $M \subset N$. $M \setminus N$ bezeichnet die mengentheoretische Differenz $\{x \in M \mid x \notin N\}$.

1.3.2 Vektoren und Matrizen

Ist R eine beliebige Menge, $n \in \mathbb{N}$, so bezeichnen wir mit

$$R^n$$

die Menge aller **n-Tupel** oder Vektoren der Länge n mit Komponenten aus R. (Aus technischen Gründen ist es gelegentlich nützlich, Vektoren $x \in R^0$, also Vektoren ohne Komponenten, zu benutzen. Wenn wir dies tun, werden wir es explizit erwähnen, andernfalls setzen wir immer $n \geq 1$ voraus.) Wir betrachten Vektoren $x = (x_i)_{i=1,\dots,n} \in R^n$ immer als **Spaltenvektoren** d. h.

$$x = \begin{pmatrix} x_1 \\ \vdots \\ x_n \end{pmatrix},$$

wollen wir mit Zeilenvektoren rechnen, so schreiben wir x^T (lies: x **transponiert**). Die Menge $\mathbb{K}^n$ ist bekanntlich ein n-dimensionaler Vektorraum über $\mathbb{K}$. Mit

$$y^T x := \sum_{i=1}^{n} x_i y_i$$

bezeichnen wir das **innere Produkt** zweier Vektoren $x, y \in \mathbb{K}^n$. Wir nennen x und y **senkrecht (orthogonal)**, falls $x^T y = 0$ gilt. Der $\mathbb{K}^n$ ist für uns immer (wenn nichts anderes gesagt wird) mit der **euklidischen Norm**

$$\|x\| := \sqrt{x^T x}$$

ausgestattet.

Für Mengen $S, T \subseteq \mathbb{K}^n$ und $\alpha \in \mathbb{K}$ benutzen wir die folgenden Standardbezeichnungen für Mengenoperationen

$$\begin{array}{ll} S + T & := \{x + y \in \mathbb{K}^n \mid x \in S, y \in T\}, \\ S - T & := \{x - y \in \mathbb{K}^n \mid x \in S, y \in T\}, \\ \alpha S & := \{\alpha x \in \mathbb{K}^n \mid x \in S, \}. \end{array}$$

Einige Vektoren aus $\mathbb{K}^n$ werden häufig auftreten, weswegen wir sie mit besonderen Symbolen bezeichnen. Mit e_j bezeichnen wir den Vektor aus $\mathbb{K}^n$, dessen j-te Komponente 1 und dessen übrige Komponenten 0

sind. Mit 0 bezeichnen wir den Nullvektor, mit $\mathbb{1}$ den Vektor, dessen Komponenten alle 1 sind. Also

$$e_j = \begin{pmatrix} 0 \\ \vdots \\ 0 \\ 1 \\ 0 \\ \vdots \\ 0 \end{pmatrix}, \quad 0 = \begin{pmatrix} 0 \\ \vdots \\ \vdots \\ \vdots \\ 0 \end{pmatrix}, \quad \mathbb{1} = \begin{pmatrix} 1 \\ \vdots \\ \vdots \\ \vdots \\ 1 \end{pmatrix}.$$

Welche Dimension die Vektoren e_j, 0, $\mathbb{1}$ haben, ergibt sich jeweils aus dem Zusammenhang.

Für eine Menge R und $m, n \in \mathbb{N}$ bezeichnet

$$R^{(m,n)} \text{ oder } R^{m \times n}$$

die Menge der (m, n)-**Matrizen** (m Zeilen, n Spalten) mit Einträgen aus R. (Aus technischen Gründen werden wir gelegentlich auch $n = 0$ oder $m = 0$ zulassen, d. h. wir werden auch Matrizen mit m Zeilen und ohne Spalten bzw. n Spalten und ohne Zeilen betrachten. Dieser Fall wird jedoch immer explizit erwähnt, somit ist in der Regel $n \geq 1$ und $m \geq 1$ vorausgesetzt.) Ist $A \in R^{(m,n)}$, so schreiben wir

$$A = (a_{ij})_{\substack{i=1,\dots,m \\ j=1,\dots,n}}$$

und meinen damit, dass A die folgende Form hat

$$A = \begin{pmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \dots & a_{mn} \end{pmatrix},$$

Wenn nichts anderes gesagt wird, hat A die Zeilenindexmenge $M = \{1, \dots, m\}$ und die Spaltenindexmenge $N = \{1, \dots, n\}$. Die j**-te Spalte** von A ist ein m-Vektor, den wir mit $A_{.j}$ bezeichnen,

$$A_{.j} = \begin{pmatrix} a_{1j} \\ \vdots \\ a_{mj} \end{pmatrix}.$$

Die i**-te Zeile** von A ist ein Zeilenvektor der Länge n, den wir mit $A_{i.}$ bezeichnen, d. h.

$$A_{i.} = (a_{i1}, a_{i2}, \dots, a_{in}).$$

Wir werden in dieser Vorlesung sehr häufig Untermatrizen von Matrizen konstruieren, sie umsortieren und mit ihnen rechnen müssen. Um dies ohne Zweideutigkeiten durchführen zu können, führen wir zusätzlich eine etwas exaktere als die oben eingeführte Standardbezeichnungsweise ein.

Wir werden — wann immer es aus technischen Gründen notwendig erscheint — die Zeilen- und Spaltenindexmengen einer (m, n)-Matrix A nicht als Mengen sondern als Vektoren auffassen. Wir sprechen dann vom

$$\begin{array}{rrcl} \text{vollen Zeilenindexvektor} & M & = & (1, 2, \dots, m) \\ \text{vollen Spaltenindexvektor} & N & = & (1, 2, \dots, n) \end{array}$$

von A. Ein **Zeilenindexvektor** von A ist ein Vektor mit höchstens m Komponenten, der aus M durch Weglassen einiger Komponenten von M und Permutation der übrigen Komponenten entsteht. Analog entsteht ein **Spaltenindexvektor** durch Weglassen von Komponenten von N und Permutation der übrigen Komponenten. Ist also

$$\begin{array}{ll} I = (i_1, i_2, \ldots, i_p) & \text{ein Zeilenindexvektor von } A \text{ und} \\ J = (j_1, j_2, \ldots, j_q) & \text{ein Spaltenindexvektor von } A, \end{array}$$

so gilt immer $i_s, i_t \in \{1, \ldots, m\}$ und $i_s \neq i_t$ für $1 \leq s < t \leq m$, und analog gilt $j_s, j_t \in \{1, \ldots, n\}$ und $j_s \neq j_t$ für $1 \leq s < t \leq n$. Wir setzen

$$A_{IJ} := \begin{pmatrix} a_{i_1 j_1} & a_{i_1 j_2} & \ldots & a_{i_1 j_q} \\ a_{i_2 j_1} & a_{i_2 j_2} & \ldots & \\ \vdots & \vdots & \ddots & \vdots \\ a_{i_p j_1} & a_{i_p j_2} & \ldots & a_{i_p j_q} \end{pmatrix}$$

und nennen A_{IJ} **Untermatrix** von A.

A_{IJ} ist also eine (p, q)-Matrix, die aus A dadurch entsteht, dass man die Zeilen, die zu Indizes gehören, die nicht in I enthalten sind, und die Spalten, die zu Indizes gehören, die nicht in J enthalten sind, streicht und dann die so entstehende Matrix umsortiert.

Ist $I = (i)$ und $J = (j)$, so erhalten wir zwei Darstellungsweisen für Zeilen bzw. Spalten von A:

$$\begin{array}{ll} A_{IN} & = A_{i.}\ , \\ A_{MJ} & = A_{.j}\ . \end{array}$$

Aus Gründen der Notationsvereinfachung werden wir auch die folgende (etwas unsaubere) Schreibweise benutzen. Ist M der volle Zeilenindexvektor von A und I ein Zeilenindexvektor, so schreiben wir auch

$$I \subseteq M,$$

obwohl I und M keine Mengen sind, und für $i \in \{1, \ldots, m\}$ benutzen wir

$$i \in I \quad \text{oder } i \notin I,$$

um festzustellen, dass i als Komponente von I auftritt oder nicht. Analog verfahren wir bezüglich der Spaltenindizes.

Gelegentlich spielt die tatsächliche Anordnung der Zeilen und Spalten keine Rolle. Wenn also z. B. $I \subseteq \{1, \ldots, n\}$ und $J \subseteq \{1, \ldots, m\}$ gilt, dann werden wir auch einfach schreiben

$$A_{IJ},$$

obwohl diese Matrix dann nur bis auf Zeilen- und Spaltenpermutationen definiert ist. Wir werden versuchen, diese Bezeichnungen immer so zu verwenden, dass keine Zweideutigkeiten auftreten. Deshalb treffen wir ab jetzt die Verabredung, dass wir — falls wir Mengen (und nicht Indexvektoren) I und J benutzen — die Elemente von $I = \{i_1, \dots, i_p\}$ und von $J = \{j_1, \dots, j_q\}$ kanonisch anordnen, d. h. die Indizierung der Elemente von I und J sei so gewählt, dass $i_1 < i_2 < \dots < i_p$ und $j_1 < j_2 < \dots < j_q$ gilt. Bei dieser Verabredung ist dann A_{IJ} die Matrix die aus A durch Streichen der Zeilen i, $i \notin I$, und der Spalten j, $j \notin J$, entsteht.

Ist $I \subseteq \{1, \dots, m\}$ und $J \subseteq \{1, \dots, n\}$ oder sind I und J Zeilen- bzw. Splaltenindexvektoren, dann schreiben wir auch

$$\begin{array}{ccc} A_{I.} & \text{statt} & A_{IN} \\ A_{.J} & \text{statt} & A_{MJ} \end{array}$$

$A_{I.}$ entsteht also aus A durch Streichen der Zeilen i, $i \notin I$, $A_{.J}$ durch Streichen der Spalten j, $j \notin J$.

Sind $A, B \in \mathbb{K}^{(m,n)}$, $C \in \mathbb{K}^{(n,s)}$, $\alpha \in \mathbb{K}$, so sind

die **Summe**	$A + B$,
das **Produkt**	αA,
das **Matrixprodukt**	AC

wie in der linearen Algebra üblich definiert.

Für einige häufig auftretende Matrizen haben wir spezielle Symbole reserviert. Mit 0 bezeichnen wir die Nullmatrix (alle Matrixelemente sind Null), wobei sich die Dimension der Nullmatrix jeweils aus dem Zusammenhang ergibt. (Das Symbol 0 kann also sowohl eine Zahl, einen Vektor als auch eine Matrix bezeichnen). Mit I bezeichnen wir die Einheitsmatrix. Diese Matrix ist quadratisch, die Hauptdiagonalelemente von I sind Eins, alle übrigen Null. Wollen wir die Dimension von I betonen, so schreiben wir auch I_n und meinen damit die (n, n)-Einheitsmatrix. Diejenige (m, n)-Matrix, bei der alle Elemente Eins sind, bezeichnen wir mit E. Wir schreiben auch $E_{m,n}$ bzw. E_n, um die Dimension zu spezifizieren (E_n ist eine (n, n)-Matrix). Ist x ein n-Vektor, so bezeichnet $\operatorname{diag}(x)$ diejenige (n, n)-Matrix $A = (a_{ij})$ mit $a_{ii} = x_i$ $(i = 1, \dots, n)$ und $a_{ij} = 0$ $(i \neq j)$. Wir halten an dieser Stelle noch einmal Folgendes fest: Wenn wir von einer Matrix A sprechen, ohne anzugeben, welche Dimension sie hat und aus welchem Bereich sie ist, dann nehmen wir implizit an, dass $A \in \mathbb{K}^{(m,n)}$ gilt. Analog gilt immer $x \in \mathbb{K}^n$, wenn sich nicht aus dem Zusammenhang anderes ergibt.

1.3.3 Kombinationen von Vektoren, Hüllen, Unabhängigkeit

Ein Vektor $x \in \mathbb{K}^n$ heißt **Linearkombination** der Vektoren $x_1, \ldots, x_k \in \mathbb{K}^n$, falls es einen Vektor $\lambda = (\lambda_1, \ldots, \lambda_k)^T \in \mathbb{K}^k$ gibt mit

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

Gilt zusätzlich

$$\left.\begin{array}{l} \lambda \geq 0 \\ \lambda^T \mathbb{1} = 1 \\ \lambda \geq 1 \quad \text{und} \quad \lambda^T \mathbb{1} = 1 \end{array}\right\} \quad \text{so heißt } x \quad \left\{\begin{array}{l} \textbf{konische} \\ \textbf{affine} \\ \textbf{konvexe} \end{array}\right\} \textbf{Kombination}$$

der Vektoren $x_1, \ldots, x_k$. Diese Kombinationen heißen **echt**, falls weder $\lambda = 0$ noch $\lambda = e_j$ für ein $j \in \{1, \ldots, k\}$ gilt.

Für eine nichtleere Teilmenge $S \subseteq \mathbb{K}^n$ heißt

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

die Menge aller Vektoren, die als lineare (konische, affine oder konvexe) Kombination von endlich vielen Vektoren aus S dargestellt werden können. Wir setzen außerdem

$$\begin{array}{ll} \operatorname{lin}(\emptyset) & := \operatorname{cone}(\emptyset) := \{0\}, \\ \operatorname{aff}(\emptyset) & := \operatorname{conv}(\emptyset) := \emptyset. \end{array}$$

Ist A eine (m, n)-Matrix, so schreiben wir auch

$$\operatorname{lin}(A), \ \operatorname{cone}(A), \ \operatorname{aff}(A), \ \operatorname{conv}(A)$$

und meinen damit die lineare, konische, affine bzw. konvexe Hülle der Spaltenvektoren $A_{.1}, A_{.2}, \ldots, A_{.n}$ von A. Eine Teilmenge $S \subseteq \mathbb{K}^n$ heißt

$$\left\{\begin{array}{l} \textbf{linearer Raum} \\ \textbf{Kegel} \\ \textbf{affiner Raum} \\ \textbf{konvexe Menge} \end{array}\right\} \quad \text{falls} \quad \left\{\begin{array}{l} S = \operatorname{lin}(S) \\ S = \operatorname{cone}(S) \\ S = \operatorname{aff}(S) \\ S = \operatorname{conv}(S) \end{array}\right\}.$$

Eine nichtleere endliche Teilmenge $S \subseteq \mathbb{K}^n$ heißt **linear** (bzw. **affin**) **unabhängig**, falls kein Element von S als echte Linearkombination (bzw. Affinkombination) von Elementen von S dargestellt werden kann. Die leere Menge ist affin, jedoch nicht linear unabhängig. Jede Menge $S \subseteq \mathbb{K}^n$, die nicht linear bzw. affin unabhängig ist, heißt **linear** bzw. **affin abhängig**. Aus der linearen Algebra wissen wir, dass eine linear (bzw. affin) unabhängige Teilmenge des $\mathbb{K}^n$ höchstens n (bzw. $n + 1$) Elemente enthält. Für eine Teilmenge $S \subseteq \mathbb{K}^n$ heißt die Kardinalität einer größten linear (bzw. affin) unabhängigen Teilmenge von S der **Rang** (bzw. **affine Rang**) von S. Wir schreiben dafür rang(S) bzw. arang(S). Die **Dimension** einer Teilmenge $S \subseteq \mathbb{K}^n$, Bezeichnung: $\dim(S)$, ist die Kardinalität einer größten affin unabhängigen Teilmenge von S minus 1, d. h. $\dim(S) = \text{arang}(S) - 1$.

Der **Rang einer Matrix** A, bezeichnet mit rang(A), ist der Rang ihrer Spaltenvektoren. Aus der linearen Algebra wissen wir, dass rang(A) mit dem Rang der Zeilenvektoren von A übereinstimmt. Gilt für eine (m, n)-Matrix A, rang$(A) = \min\{m, n\}$, so sagen wir, dass A **vollen Rang** hat. Eine (n, n)-Matrix mit vollem Rang ist **regulär**, d. h. sie besitzt eine (eindeutig bestimmte) inverse Matrix (geschrieben A^{-1}) mit der Eigenschaft $AA^{-1} = I$.

Kapitel 2

Polyeder

Das Hauptanliegen dieser Vorlesung ist die Behandlung von Problemen der linearen Programmierung. Man kann die Verfahren zur Lösung derartiger Probleme rein algebraisch darstellen als Manipulation von linearen Gleichungs- und Ungleichungssystemen. Die meisten Schritte dieser Algorithmen haben jedoch auch eine geometrische Bedeutung; und wenn man erst einmal die geometrischen Prinzipien und Regeln verstanden hat, die hinter diesen Schritten liegen, ist es viel einfacher, die Algorithmen und ihre Korrektheitsbeweise zu verstehen. Wir wollen daher in dieser Vorlesung einen geometrischen Zugang wählen und zunächst die Lösungsmengen linearer Programme studieren. Speziell wollen wir unser Augenmerk auf geometrische Sachverhalte und ihre algebraische Charakterisierung legen. Dies wird uns helfen, aus geometrischen Einsichten algebraische Verfahren abzuleiten.

(2.1) Definition.

(a) *Eine Teilmenge* $G \subseteq \mathbb{K}^n$ *heißt* **Hyperebene**, *falls es einen Vektor* $a \in \mathbb{K}^n \setminus \{0\}$ *und* $\alpha \in \mathbb{K}$ *gibt mit*

$$G = \{x \in \mathbb{K}^n \mid a^T x = \alpha\}.$$

Der Vektor a *heißt* **Normalenvektor** *zu* G.

(b) *Eine Teilmenge* $H \subseteq \mathbb{K}^n$ *heißt* **Halbraum**, *falls es einen Vektor* $a \in \mathbb{K}^n \setminus \{0\}$ *und* $\alpha \in \mathbb{K}$ *gibt mit*

$$H = \{x \in \mathbb{K}^n \mid a^T x \leq \alpha\}.$$

Wir nennen a den **Normalenvektor** *zu H. Die Hyperebene $G = \{x \in \mathbb{K}^n \mid a^T x = \alpha\}$ heißt die zum Halbraum H gehörende Hyperebene oder die H* ***berandende Hyperebene****, und H heißt der zu G gehörende Halbraum.*

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

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

Um zu betonen, dass P durch A und b definiert ist, schreiben wir auch

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

(d) *Ein Polyeder P heißt* **Polytop**, *wenn es beschränkt ist, d. h. wenn es ein $B \in \mathbb{K}$, $B > 0$ gibt mit $P \subseteq \{x \in \mathbb{K}^n \mid \|x\| \leq B\}$.*

□

Polyeder können wir natürlich auch in folgender Form schreiben

$$P(A, b) = \bigcap_{i=1}^{m} \{x \in \mathbb{K}^n \mid A_{i}.x \leq b_i\}.$$

Halbräume sind offensichtlich Polyeder. Aber auch die leere Menge ist ein Polyeder, denn $\emptyset = \{x \mid 0^T x \leq -1\}$, und der gesamte Raum ist ein Polyeder, denn $\mathbb{K}^n = \{x \mid 0^T x \leq 0\}$. Sind alle Zeilenvektoren $A_{i}.$ von A vom Nullvektor verschieden, so sind die bei der obigen Durchschnittsbildung beteiligten Mengen Halbräume. Ist ein Zeilenvektor von A der Nullvektor, sagen wir $A_{1}. = 0^T$, so ist $\{x \in \mathbb{K}^n \mid A_{1}.x \leq b_1\}$ entweder leer (falls $b_1 < 0$) oder der gesamte Raum $\mathbb{K}^n$ (falls $b_1 \geq 0$). Das heißt, entweder ist $P(A, b)$ leer oder die Mengen $\{x \mid A_{i}. \leq b_i\}$ mit $A_{i}. = 0$ können bei der obigen Durchschnittsbildung weggelassen werden. Daraus folgt

Jedes Polyeder $P \neq \mathbb{K}^n$ ist Durchschnitt von endlich vielen Halbräumen.

Gilt $P = P(A, b)$, so nennen wir das Ungleichungssystem $Ax \leq b$ ein **P definierendes System** (von linearen Ungleichungen). Sind $\alpha > 0$ und $1 \leq i < j \leq m$, so gilt offensichtlich

$$P(A, b) = P(A, b) \cap \{x \mid \alpha A_{i}.x \leq \alpha b_i\} \cap \{x \mid (A_{i}. + A_{j}.)x \leq b_i + b_j\}.$$

Daraus folgt, dass A und b zwar $P = P(A, b)$ eindeutig bestimmen, dass aber P unendlich viele Darstellungen der Form $P(D, d)$ hat.

(2.2) Beispiel. Wir betrachten das Ungleichungssystem

$$
\begin{array}{lrcrcr}
(1) & 2x_1 & & & \leq & 5 \\
(2) & & - & 2x_2 & \leq & 1 \\
(3) & -x_1 & - & x_2 & \leq & -1 \\
(4) & 2x_1 & + & 9x_2 & \leq & 23 \\
(5) & 6x_1 & - & 2x_2 & \leq & 13
\end{array}
$$

Hieraus erhalten wir die folgende Matrix A und den Vektor b:

$$
A = \begin{pmatrix} 2 & 0 \\ 0 & -2 \\ -1 & -1 \\ 2 & 9 \\ 6 & -2 \end{pmatrix} \qquad b = \begin{pmatrix} 5 \\ 1 \\ -1 \\ 23 \\ 13 \end{pmatrix}
$$

Das Polyeder $P = P(A, b)$ ist die Lösungsmenge des obigen Ungleichungssystems (1),. . .,(5) und ist in Abbildung 2.1 graphisch dargestellt.

□

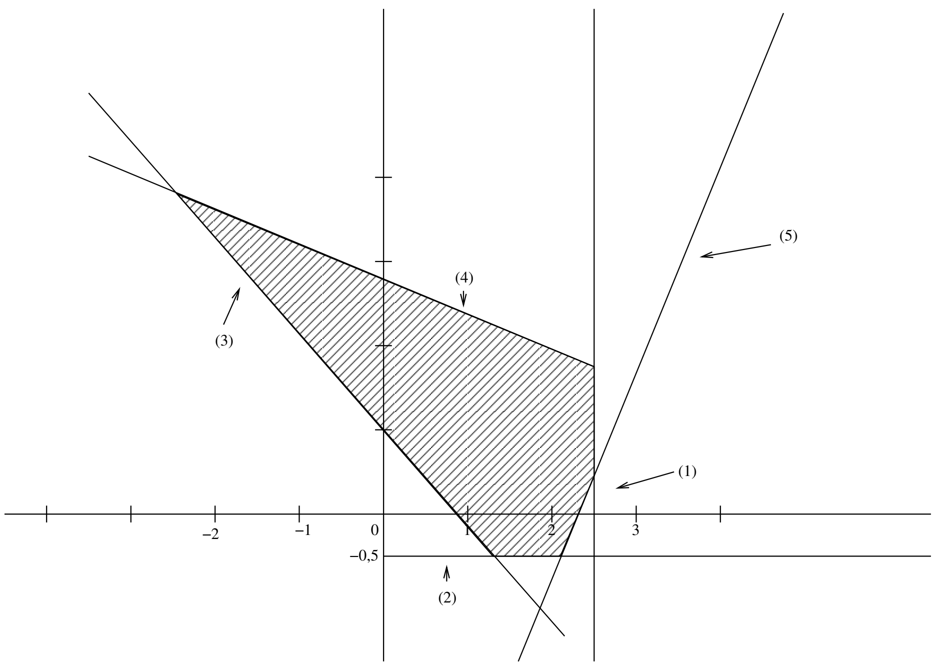


Abb. 2.1

Die Mengen zulässiger Lösungen linearer Programme treten nicht immer in der Form $Ax \leq b$ auf. Häufig gibt es auch Gleichungen und vorzeichenbeschränkte Variable. Vorzeichenbeschränkungen sind natürlich auch lineare Ungleichungssysteme, und ein Gleichungssytem $Dx = d$ kann in der Form von zwei Ungleichungssystemen $Dx \leq d$ und $-Dx \leq -d$ geschrieben werden. Allgemeiner gilt

(2.3) Bemerkung. *Die Lösungsmenge des Systems*

$$\begin{array}{ll} Bx + Cy & = c \\ Dx + Ey & \leq d \\ x & \geq 0 \\ x \in \mathbb{K}^p, & y \in \mathbb{K}^q \end{array}$$

ist ein Polyeder.

Beweis : Setze $n := p + q$ und

$$A := \begin{pmatrix} B & C \\ -B & -C \\ D & E \\ -I & 0 \end{pmatrix}, \quad b := \begin{pmatrix} c \\ -c \\ d \\ 0 \end{pmatrix}$$

dann ist $P(A, b)$ die Lösungsmenge des vorgegebenen Gleichungs- und Ungleichungssystems. □

Ein spezieller Polyedertyp wird uns häufig begegnen, weswegen wir für ihn eine besondere Bezeichnung wählen wollen. Für $A \in \mathbb{K}^{(m,n)}$, $b \in \mathbb{K}^m$ setzen wir

$$\mathbf{P^=(A, b)} := \{x \in \mathbb{K}^n \mid Ax = b, x \geq 0\}.$$

Nicht alle Polyeder können in der Form $P^=(A, b)$ dargestellt werden, z. B. nicht $P = \{x \in \mathbb{K} \mid x \leq 1\}$.

Wir werden später viele Sätze über Polyeder P beweisen, deren Aussagen **darstellungsabhängig** sind, d. h. die Art und Weise, wie P gegeben ist, geht explizit in die Satzaussage ein. So werden sich z. B. die Charakterisierungen gewisser Polyedereigenschaften von $P(A, b)$ (zumindest formal) von den entsprechenden Charakterisierungen von $P^=(A, b)$ unterscheiden. Darstellungsabhängige Sätze wollen wir jedoch nur einmal beweisen (normalerweise für Darstellungen, bei denen die Resultate besonders einprägsam oder einfach sind), deshalb werden wir uns nun Transformationsregeln überlegen, die angeben, wie man von einer Darstellungsweise zu einer anderen und wieder zurück kommt.

(2.4) Transformationen.

Regel I: Einführung von Schlupfvariablen

Gegeben seien $a \in \mathbb{K}^n$, $\alpha \in \mathbb{K}$. Wir schreiben die Ungleichung

(i) $a^T x \leq \alpha$

in der Form einer Gleichung und einer Vorzeichenbeschränkung

(ii) $a^T x + y = \alpha, \quad y \geq 0.$

y ist eine neue Variable, genannt **Schlupfvariable**.

Es gilt:

$$\begin{aligned} x \quad \text{erfüllt (i)} &\implies \begin{pmatrix} x \\ y \end{pmatrix} \text{erfüllt (ii) für} \quad y = \alpha - a^T x. \\ \begin{pmatrix} x \\ y \end{pmatrix} \text{erfüllt (ii)} &\implies x \text{ erfüllt (i).} \end{aligned}$$

Allgemein: Ein Ungleichungssystem $Ax \leq b$ kann durch Einführung eines Schlupfvariablenvektors y transformiert werden in ein Gleichungssystem mit Vorzeichenbedingung $Ax + y = b$, $y \geq 0$. Zwischen den Lösungsmengen dieser beiden Systeme bestehen die oben angegebenen Beziehungen. $P(A, b)$ und $\{\binom{x}{y} \in \mathbb{K}^{n+m} \mid Ax + Iy = b, y \geq 0\}$ sind jedoch zwei durchaus verschiedene Polyeder in verschiedenen Vektorräumen.

Regel II: Einführung von vorzeichenbeschränkten Variablen

Ist x eine (eindimensionale) nicht vorzeichenbeschränkte Variable, so können wir zwei vorzeichenbeschränkte Variablen x^+ und x^- einführen, um x darzustellen. Wir setzen

$$x := x^+ - x^- \quad \text{mit} \quad x^+ \geq 0,\ x^- \geq 0.$$

$\square$

Mit den Regeln I und II aus (2.4) können wir z. B. jedem Polyeder $P(A, b) \subseteq \mathbb{K}^n$ ein Polyeder $P^=(D, d) \subseteq \mathbb{K}^{2n+m}$ wie folgt zuordnen. Wir setzen

$$D := (A, -A, I_m),\ d := b,$$

d. h. es gilt

$$P^{=}(D,d) = \left\{ \begin{pmatrix} x \\ y \\ z \end{pmatrix} \in \mathbb{K}^{2n+m} \mid Ax - Ay + z = b,\ x, y, z \geq 0 \right\}.$$

Es ist üblich, die oben definierten Polyeder $P(A,b)$ und $P^{=}(D,d)$ **äquivalent** zu nennen. Hierbei hat "Äquivalenz" folgende Bedeutung.

Für $x \in \mathbb{K}^n$ sei

$$\begin{aligned} x^+ &:= (x_1^+, \ldots, x_n^+)^T \quad \text{mit} \quad x_i^+ = \max\{0, x_i\}, \\ x^- &:= (x_1^-, \ldots, x_n^-)^T \quad \text{mit} \quad x_i^- = \max\{0, -x_i\}, \end{aligned}$$

dann gilt:

$$x \in P(A,b) \quad \Longrightarrow \quad \begin{pmatrix} x^+ \\ x^- \\ z \end{pmatrix} \in P^{=}(D,d) \ \text{ für } z = b - Ax$$

$$\begin{pmatrix} u \\ v \\ w \end{pmatrix} \in P^{=}(D,d) \quad \Longrightarrow \quad x := u - v \in P(A,b).$$

Besonders einfache Polyeder, auf die sich jedoch fast alle Aussagen der Polyedertheorie zurückführen lassen, sind polyedrische Kegel.

(2.5) Definition. *Ein Kegel $C \subseteq \mathbb{K}^n$ heißt* **polyedrisch** *genau dann, wenn C ein Polyeder ist.*

□

(2.6) Bemerkung. *Ein Kegel $C \subseteq \mathbb{K}^n$ ist genau dann polyedrisch, wenn es eine Matrix $A \in \mathbb{K}^{(m,n)}$ gibt mit*

$$C = P(A, 0).$$

Beweis : Gilt $C = P(A,0)$, so ist C ein Polyeder und offensichtlich ein Kegel.

Sei C ein polyedrischer Kegel, dann existieren nach Definition (2.1) (c) eine Matrix A und ein Vektor b mit $C = P(A,b)$. Da jeder Kegel den Nullvektor enthält, gilt $0 = A0 \leq b$. Angenommen, es existiert ein $\overline{x} \in C$ mit $A\overline{x} \not\leq 0$, d. h. es existiert eine Zeile von A, sagen wir $A_{i.}$, mit $t := A_{i.}\overline{x} > 0$. Da C ein Kegel ist, gilt $\lambda\overline{x} \in C$ für alle $\lambda \in \mathbb{K}_+$. Jedoch für $\overline{\lambda} := \frac{b_i}{t} + 1$ gilt einerseits $\overline{\lambda}\overline{x} \in C$ und andererseits $A_{i.}(\overline{\lambda}\overline{x}) = \overline{\lambda}t > b_i$, ein Widerspruch. Daraus folgt, für alle $x \in C$ gilt $Ax \leq 0$. Hieraus ergibt sich $C = P(A, 0)$. $\square$

In der folgenden Tabelle 2.1 haben wir alle möglichen Transformationen aufgelistet. Sie soll als "Nachschlagewerk" dienen.

Transformation nach / von	$By \le d$	$By = d$, $y \ge 0$	$By \le d$, $y \ge 0$
$Ax = b$ hierbei ist $x_i^+ = \max\{0, x_i\}$ $x_i^- = \max\{0, -x_i\}$	$\begin{pmatrix} A \\ -A \end{pmatrix} y \le \begin{pmatrix} b \\ -b \end{pmatrix}$ $y = x$	$(A, -A)\ y = b$ $y \ge 0$ $y = \begin{pmatrix} x^+ \\ x^- \end{pmatrix}$	$\begin{pmatrix} A & -A \\ -A & A \end{pmatrix} y \le \begin{pmatrix} b \\ -b \end{pmatrix}$ $y \ge 0$ $y = \begin{pmatrix} x^+ \\ x^- \end{pmatrix}$
$Ax \le b$	$Ay \le b$ $y = x$	$(A, -A, I)\ y = b$ $y \ge 0$ $y = \begin{pmatrix} x^+ \\ x^- \\ b - Ax \end{pmatrix}$	$(A, -A)\ y \le b$ $y \ge 0$ $y = \begin{pmatrix} x^+ \\ x^- \end{pmatrix}$
$Ax = b$ $x \ge 0$	$\begin{pmatrix} A \\ -A \\ -I \end{pmatrix} y \le \begin{pmatrix} b \\ -b \\ 0 \end{pmatrix}$ $y = x$	$Ay = b$ $y \ge 0$ $y = x$	$\begin{pmatrix} A \\ -A \end{pmatrix} y \le \begin{pmatrix} b \\ -b \end{pmatrix}$ $y \ge 0$ $y = x$
$Ax \le b$ $x \ge 0$	$\begin{pmatrix} A \\ -I \end{pmatrix} y \le \begin{pmatrix} b \\ 0 \end{pmatrix}$ $y = x$	$(A, I)\ y = b$ $y \ge 0$ $y = \begin{pmatrix} x \\ b - Ax \end{pmatrix}$	$Ay \le b$ $y \ge 0$ $y = x$

Kapitel 3

Fourier-Motzkin-Elimination und Projektion

In diesem Kapitel untersuchen wir die folgende Frage: Wie kann man herausfinden, ob ein lineares Ungleichungssystem $Ax \leq b$ eine Lösung hat oder nicht? Nehmen wir an, dass A und b rational sind (wenn man Computer benutzen will, muss man diese Annahmen sowieso machen), dann können wir die Elemente von $\mathbb{Q}^n$ durchnummerieren und iterativ jedes Element dieser Folge in das System $Ax \leq b$ einsetzen. Falls $Ax \leq b$ lösbar ist, werden wir nach endlich vielen Schritten einen zulässigen Vektor finden. Hat $Ax \leq b$ keine zulässige Lösung, so läuft diese Enumeration unendlich lange. Kann man die Unlösbarkeit vielleicht auch mit einem endlichen Verfahren prüfen? Wir werden hierauf in diesem Kapitel eine positive Antwort geben.

Wir beginnen mit der Beschreibung eines Algorithmus, der aus einer (m, n)-Matrix A und einem Vektor $b \in \mathbb{K}^m$ eine (r, n)-Matrix D und einen Vektor $d \in \mathbb{K}^r$ macht, so dass eine Spalte von D aus lauter Nullen besteht und dass gilt:
$Ax \leq b$ hat eine Lösung genau dann, wenn $Dx \leq d$ eine Lösung hat.

3.1 Fourier-Motzkin-Elimination

(3.1) Algorithmus. Fourier-Motzkin-Elimination (der j-ten Variablen).

Input: *Eine (m, n)-Matrix $A = (a_{ij})$, ein Vektor $b \in \mathbb{K}^m$ und ein Spaltenindex $j \in \{1, \ldots, n\}$ der Matrix A.*
Output: *Eine (r, n)-Matrix $D = (d_{ij})$ (r wird im Algorithmus berechnet) und ein*

Vektor $d \in \mathbb{K}^r$, so dass die j-te Spalte von D, bezeichnet mit $D_{\cdot j}$, der Nullvektor ist.
Schritt 1. *Partitioniere die Menge der Zeilenindizes $M = \{1, \ldots, m\}$ von A wie folgt:*

$$\begin{aligned} N &:= \{i \in M \mid a_{ij} < 0\}, \\ Z &:= \{i \in M \mid a_{ij} = 0\}, \\ P &:= \{i \in M \mid a_{ij} > 0\}. \end{aligned}$$

(Die Menge $Z \cup (N \times P)$ wird die Zeilenindexmenge von D. Diese kann übrigens leer und $r = 0$ sein.)
Schritt 2. *Setze $r := |Z \cup (N \times P)|$, $R := \{1, \ldots, r\}$, und sei $P : R \to Z \cup (N \times P)$ eine Bijektion (d.h. eine kanonische Indizierung der Elemente von $Z \cup (N \times P)$).*
Schritt 3. *Führe für $i = 1, 2, \ldots, r$ aus:*

(a) *Falls $p(i) \in Z$, dann setze $D_{i\cdot} := A_{p(i)\cdot}$, $d_i := b_{p(i)}$*

(d.h., die i-te Zeile von D ist gleich der $p(i)$-ten Zeile von A).

(b) *Falls $p(i) = (s, t) \in N \times P$, dann setze*

$$\begin{aligned} D_{i\cdot} &:= a_{tj} A_{s\cdot} - a_{sj} A_{t\cdot}, \\ d_i &:= a_{tj} b_s - a_{sj} b_t \end{aligned}$$

(d.h., das a_{sj}-fache der t-ten Zeile von A wird vom a_{tj}-fachen der s-ten Zeile von A abgezogen und als i-te Zeile von D betrachtet).

Ende.

Bevor wir den Algorithmus analysieren, betrachten wir ein Beispiel:

$$\begin{array}{lrrl} (1) & -7\,x_1 & +\,6x_2 & \le 25 \\ (2) & +\;x_1 & -\,5x_2 & \le \;\;1 \\ (3) & +\;x_1 & & \le \;\;7 \\ (4) & -\;x_1 & +\,2x_2 & \le 12 \\ (5) & -\;x_1 & -\,3x_2 & \le \;\;1 \\ (6) & +2\,x_1 & -\;\;x_2 & \le 10 \end{array}$$

Wir wollen die zweite Variable x_2 eliminieren und erhalten in Schritt 1 von (3.1):

$$N = \{2, 5, 6\}, \quad Z = \{3\}, \quad P = \{1, 4\}.$$

Daraus ergibt sich $|Z \cup (N \times P)| = 7$. Die Nummerierung der Zeilen von D ist dann $R = \{1, \ldots, 7\}$, und wir setzen:

$p(1) = 3,\ p(2) = (2,1),\ p(3) = (2,4),\ p(4) = (5,1),\ p(5) = (5,4),\ p(6) = (6,1),\ p(7) = (6,7)$.

Die zweite Zeile von D ergibt sich dann als die Summe vom 6-fachen der zweiten Zeile von A und dem 5-fachen der ersten Zeile von A. Die zweite Ungleichung des Systems $Dx \leq d$ hat somit die Form $-29x_1 + 0x_2 \leq 131$.

Insgesamt ergibt sich (wenn wir die aus lauter Nullen bestehende zweite Spalte weglassen), das folgende System $Dx \leq d$:

$$\begin{array}{rcrcr}
(1) & + & x_1 & \leq & 7 \\
(2) & - & 29x_1 & \leq & 131 \\
(3) & - & 3x_1 & \leq & 62 \\
(4) & - & 27x_1 & \leq & 81 \\
(5) & - & 5x_1 & \leq & 38 \\
(6) & + & 5x_1 & \leq & 85 \\
(7) & + & 3x_1 & \leq & 32
\end{array}$$

Ganz offensichtlich können einige der Koeffizienten gekürzt werden. Wir sehen, dass die Zeilen $(1), (6), (7)$ obere und die Ungleichungen $(2), (3), (4), (5)$ untere Schranken auf den für x_1 zulässigen Wert liefern. Wir sehen aber auch, dass die Eliminationsmethode viele überflüssige (wir werden das später *redundante* nennen) Ungleichungen liefert. Wir wollen nun das Ergebnis der Elimination der j-ten Variablen analysieren. Wir schauen zunächst einige Trivialfälle an.

Wenn alle Elemente der Spalte $A_{.j}$ Null sind, so ist $A = D$, und wir haben (noch) nichts erreicht. Gilt $M = P$ oder $M = N$, sind also alle Elemente der Spalte $A_{.j}$ von Null verschieden und haben dasselbe Vorzeichen, dann ist $r = 0$ und somit D die leere Matrix mit 0 Zeilen und n Spalten. $P(D, d)$ ist dann der gesamte Raum $\mathbb{K}^n$.

(3.2) Satz. *Sei $A \in \mathbb{K}^{(m,n)}$, $b \in \mathbb{K}^m$, j ein Spaltenindex von A, und seien $D \in \mathbb{K}^{(r,n)}$, $d \in \mathbb{K}^r$ die in (3.1) konstruierte Matrix bzw. Vektor. Es gilt:*

(a) Die j-te Spalte von D ist der Nullvektor.

(b) Jede Ungleichung $D_{i.}x \leq d_i$, $i = 1, \ldots, r$, ist eine konische Kombination der Ungleichungen $A_{k.}x \leq b_k$, $k = 1, \ldots, m$, d. h. es existiert ein Vektor $u \in \mathbb{R}^m$, $u \geq 0$ mit $u^T A = D_{i.}$ und $u^T b = d_i$.

(c) *Daraus folgt, es existiert eine* (r, m)*-Matrix* U, $U \geq 0$, *mit* $UA = D$ *und* $Ub = d$.

(d) *Sei* $\overline{x} \in \mathbb{K}^n$ *mit* $\overline{x_j} = 0$, *sei* e_j *der* j*-te Einheitsvektor und seien* N, Z, P *die in Schritt 1 von (3.1) definierten Mengen.*

Setzen wir:

$$\lambda_i := \frac{1}{a_{ij}}(b_i - A_i.\overline{x}) \qquad \text{für alle } i \in P \cup N$$

$$L := \begin{cases} -\infty & \text{falls } N = \emptyset \\ \max\{\lambda_i \mid i \in N\} & \text{falls } N \neq \emptyset \end{cases}$$

$$U := \begin{cases} +\infty & \text{falls } P = \emptyset. \\ \min\{\lambda_i \mid i \in P\} & \text{falls } P \neq \emptyset. \end{cases}$$

Dann gilt

$$\begin{array}{lll} (d_1) & \overline{x} \in P(D, d) & \Rightarrow L \leq U \text{ und } \overline{x} + \lambda e_j \in P(A, b) \, \forall \, \lambda \in [L, U] \\ (d_2) & \overline{x} + \lambda e_j \in P(A, b) & \Rightarrow \lambda \in [L, U] \text{ und } \overline{x} \in P(D, d). \end{array}$$

Beweis :

(a) Falls D die leere Matrix ist, so ist die Behauptung natürlich richtig. Wurde eine Zeile i von D in Schritt 3 (a) definiert, so ist $d_{ij} = 0$, da $a_{p(i)j} = 0$. Andernfalls gilt $d_{ij} = a_{tj}a_{sj} - a_{sj}a_{tj} = 0$ nach Konstruktion. Das heißt, die j-te Spalte $D._j$ von D ist der Nullvektor. Man beachte auch, dass im Falle $N = \emptyset$ oder $P = \emptyset$, D nur aus den Zeilen $A_i.$ von A besteht mit $i \in Z$.

(b) ist offensichtlich nach Definition und impliziert (c).

(d) Sei $\overline{x} \in \mathbb{K}^n$ mit $\overline{x_j} = 0$ ein beliebiger Vektor.

(d$_1$) Angenommen $\overline{x} \in P(D, d)$. Wir zeigen zunächst, dass $L \leq U$ gilt. Ist $P = \emptyset$ oder $N = \emptyset$, so gilt offensichtlich $L \leq U$ nach Definition. Wir können also annehmen, dass $P \neq \emptyset$, $N \neq \emptyset$ gilt. Es seien $s \in N, t \in P$ und $q \in R$ so gewählt, dass $p(q) = (s, t)$ und $\lambda_s = L$, $\lambda_t = U$ gelten.

Dann gilt

$$a_{tj}A_s.\overline{x} - a_{sj}A_t.\overline{x} = D_q.\overline{x} \leq d_q = a_{tj}b_s - a_{sj}b_t,$$

woraus folgt

$$a_{sj}(b_t - A_t.\overline{x}) \leq a_{tj}(b_s - A_s.\overline{x})$$

und somit (wegen $a_{Aj} > 0$ und $a_{sj} < 0$)

$$U = \lambda_t = \frac{1}{a_{tj}}(b_t - A_t.\overline{x}) \geq \frac{1}{a_{sj}}(b_s - A_s.\overline{x}) = \lambda_s = L.$$

Wir zeigen nun $A_i.(\overline{x}+\lambda_{e_j}) \leq b_i$ für alle $\lambda \in [L, U]$ und alle $i \in M = P \cup N \cup Z$, wobei e_j den j-ten Einheitsvektor bezeichnet.

Ist $i \in Z$, so gibt es ein $j \in R$ mit $p(j) = i$ und $D_j. = A_i.$, $d_j = b_i$. Aus $D_j.\overline{x} \leq d_j$ folgt daher die Behauptung.

Ist $i \in P$, dann gilt $U < +\infty$ und somit

$$A_i.(\overline{x} + \lambda ej) = A_i.\overline{x} + a_{ij}\lambda \leq A_i.\overline{x} + a_{ij}U \leq A_i.\overline{x} + a_{ij}\lambda_i = b_i.$$

Analog folgt die Behauptung für $i \in N$.
Der Beweis (d_2) sei dem Leser überlassen. □

(3.3) Folgerung. $P(A, b) \neq \emptyset \Longleftrightarrow P(D, d) \neq \emptyset$. □

Sprechweise für (3.3):

$Ax \leq b$ ist konsistent (lösbar) $\Longleftrightarrow$ $Dx \leq d$ ist konsistent (lösbar).

Die Fourier-Motzkin-Elimination der j-ten Variablen kann man natürlich auch auf ein System von normalen und strikten Ungleichungen anwenden. Eine Nachvollziehung des obigen Beweises liefert:

(3.4) Satz. *Seien A eine (m, n)-Matrix, $b \in \mathbb{K}^m$, $j \in \{1, \ldots, n\}$ und I, J eine Partition der Zeilenindexmenge $M = \{1, \ldots, m\}$. Seien $D \in \mathbb{K}^{r \times n}$, $d \in \mathbb{K}^r$ die durch Fourier-Motzkin-Elimination der Variablen j gewonnene Matrix bzw. Vektor.*

Setze $E := p^{-1}((Z \cap I) \cup ((N \times P) \cap (I \times I)))$, $F := R \setminus E$, (wobei p die in (3.1) Schritt 2 definierte Abbildung ist),

dann gilt:

Das System

$A_I.x \leq b_I$, $A_J.x < b_J$ hat eine Lösung genau dann, wenn

das System

$D_E.x \leq d_E, D_F.x < d_F$ *eine Lösung hat.*

Beweis : Hausaufgabe.

Wir können nun die Fourier-Motzkin-Elimination sukzessive auf alle Spaltenindizes anwenden. Ist in einem Schritt die neu konstruierte Matrix leer, so ist das System konsistent. Wir wollen herausfinden, was passiert, wenn $Ax \leq b$ nicht konsistent ist. Wir nehmen daher an, dass keine der sukzessiv konstruierten Matrizen leer ist.

Eliminieren wir die 1. Spalte von A, so erhalten wir nach Satz (3.2) eine (r_1, n)-Matrix D_1 und einen Vektor $d_1 \in \mathbb{K}^{r_1}$ mit

$$Ax \leq b \text{ konsistent } \Longleftrightarrow D_1x \leq d_1 \text{ konsistent.}$$

Die erste Spalte von D_1 ist eine Nullspalte. Nun eliminieren wir die zweite Spalte von D_1 und erhalten mit Satz (3.2) eine (r_2, n)-Matrix D_2 und einen Vektor $d_2 \in \mathbb{K}^{r_2}$ mit

$$D_1x \leq d_1 \text{ konsistent } \Longleftrightarrow D_2x \leq d_2 \text{ konsistent}$$

Die erste und die zweite Spalte von D_2 bestehen aus lauter Nullen. Wir eliminieren nun die dritte Spalte von D_2 und fahren auf diese Weise fort, bis wir die n-te Spalte eliminiert haben. Insgesamt erhalten wir:

$$Ax \leq b \text{ konsistent } \Longleftrightarrow D_1x \leq d_1 \text{ konsistent. } \Longleftrightarrow \ldots \Longleftrightarrow D_nx \leq d_n \text{ konsistent.}$$

Was haben wir durch diese äquivalenten Umformungen gewonnen? Nach Konstruktion hat die n-te Matrix D_n dieser Folge von Matrizen nur noch Nullspalten, ist also eine Nullmatrix. Das System $D_nx \leq d_n$ ist also nichts anderes als $0x \leq d_n$, und die Konsistenz dieses letzten Ungleichungssystems ist äquivalent zur Konsistenz von $Ax \leq b$. Die Konsistenz von $0x \leq d_n$ ist trivial überprüfbar, denn $0x \leq d_n$ hat genau dann keine Lösung, wenn der Vektor $d_n \in \mathbb{K}^{r_n}$ eine negative Komponente hat; das heißt, $d_n \geq 0$ ist äquivalent zur Konsistenz von $Ax \leq b$.

Aus Satz (3.3) folgt außerdem Folgendes: Es existiert eine Matrix $U_1 \in \mathbb{K}^{(r_1,m)}$ mit $U_1 \geq 0$ und

$$D_1 = U_1A \quad , \quad d_1 = U_1b$$

Erneute Anwendung von (3.3) liefert die Existenz einer Matrix $U_2 \in \mathbb{K}^{(r_2,r_1)}$ mit $U_2 \geq 0$ und

$$D_2 = U_2 D_1 \quad , \quad d_2 = U_2 d_1$$

Setzen wir die Schlussweise fort, so erhalten wir eine Matrix $U_n \in \mathbb{K}^{(r_n,r_{n-1})}$ mit $U_n \geq 0$ und

$$0 = D_n = U_n D_{n-1} \quad , \quad d_n = U_n d_{n-1}$$

Für die Matrix

$$U := U_n \cdot \ldots \cdot U_1 \in \mathbb{K}^{(r_n,m)}$$

gilt dann

$$U \geq 0 \quad , \quad 0 = UA \quad , \quad d_n = Ub$$

Daraus folgt, dass jede Zeile von $0 = D_n$ eine konische Kombination von Zeilen von A ist.

Wir haben uns oben überlegt, dass $Ax \leq b$ genau dann nicht konsistent ist, wenn $D_n x \leq d_n$ nicht konsistent ist. Dies ist genau dann der Fall, wenn es eine Komponente von d_n gibt, die negativ ist, anders ausgedrückt, wenn es einen Vektor u (eine Zeile der Matrix U) gibt mit $0^T = u^T A$ und $u^T b < 0$. Fassen wir diese Beobachtung zusammen.

(3.5) Folgerung. *Es seien $A \in \mathbb{K}^{(m,n)}$ und $b \in \mathbb{K}^m$, dann gilt: Das Ungleichungssystem $Ax \leq b$ hat genau dann keine Lösung, wenn es einen Vektor $u \in \mathbb{K}^m$, $u \geq 0$ gibt mit $u^T A = 0^T$ und $u^T b < 0$.*

□

Diese (einfache) Konsequenz aus unserem Eliminationsverfahren (3.1) ist eines der nützlichsten Resultate der Polyedertheorie.

Ergänzende Abschweifung

Das Fourier-Motzkin-Eliminationsverfahren ist ein Spezialfall eines allgemeinen Projektionsverfahrens auf lineare Teilräume des $\mathbb{K}^n$.

(3.6) Definition. *Sei L ein linearer Unterraum von $\mathbb{K}^n$. Ein Vektor $\overline{x} \in \mathbb{K}^n$ heißt* **orthogonale Projektion** *eines Vektors $x \in \mathbb{K}^n$* **auf** *L, falls $\overline{x} \in L$ und $(x - \overline{x})^T \overline{x} = 0$ gilt.*

(3.7) Bemerkung. *Sei $B \in K^{(m,n)}$ eine Matrix mit vollem Zeilenrang, $L = \{x \in \mathbb{K}^n \mid Bx = 0\}$, dann ist für jeden Vektor $x \in \mathbb{K}^n$, der Vektor*

$$\overline{x} := (I - B^T(BB^T)^{-1}B)x$$

die orthogonale Projektion auf L.

Die Fourier-Motzkin-Elimination der j-ten Variablen kann man, wie wir jetzt zeigen, als orthogonale Projektion von $P(A,b)$ auf den Vektorraum $L = \{x | x_j = 0\}$ deuten.

Wir zeigen nun, wie man ein Polyeder $P(A,b)$ auf $L = \{y | c^T y = 0\}, c \neq 0$, projiziert.

(3.8) Algorithmus. *Orthogonale Projektion eines Polyeders auf* $L = \{y | c^T y = 0\}, c \neq 0$ *(kurz: Projektion entlang c).*

Input: *Ein Vektor $c \in \mathbb{K}^n$, $c \neq 0$ (die Projektionsrichtung),*
eine Matrix $A \in \mathbb{K}^{(m,n)}$,
ein Vektor $b \in \mathbb{K}^m$.

Output: *Eine Matrix $D \in \mathbb{K}^{(r,n)}$* (r wird im Algorithmus berechnet),
ein Vektor $d \in \mathbb{K}^r$

(1) *Partitioniere die Menge $M = \{1, \ldots, m\}$ der Zeilenindizes von A wie folgt*

$$\begin{aligned} N &:= \{i \in M \mid A_{i.}c < 0\}, \\ Z &:= \{i \in M \mid A_{i.}c = 0\}, \\ P &:= \{i \in M \mid A_{i.}c > 0\}. \end{aligned}$$

(2) *Setze*

$$\begin{aligned} &r := |Z \cup N \times P|, \\ &R := \{1, 2, \ldots, r\}, \quad \textit{und} \\ &p : R \to Z \cup N \times P \quad \textit{sei eine Bijektion.} \end{aligned}$$

(3) *Für $i = 1, 2, \ldots, r$ führe aus:*

(a) *Ist $p(i) \in Z$, dann setze*

$$D_{i.} := A_{p(i).}\,, d_i = b_{p(i)}\,.$$

(D. h. die i-te Zeile von D ist gleich der $p(i)$-ten Zeile von A.

(b) *Ist $p(i) = (s,t) \in N \times P$, dann setze*

$$\begin{aligned} D_{i.} &:= (A_{t.}c)A_{s.} - (A_{s.}c)A_{t.}, \\ d_i &:= (A_{t.}c)b_s - (A_{s.}c)b_t. \end{aligned}$$

(4) *Gib D und d aus.*

Ende. □

(3.9) Satz. *Seien $D \in \mathbb{K}^{(r,n)}$ und $d \in \mathbb{K}^r$ die in (3.1) konstruierte Matrix bzw. Vektor. Dann gilt*

(a) *Die Zeilen von D sind orthogonal zu c, d.h., die Zeilenvektoren von D sind in $L = \{y | c^T y = 0\}$ enthalten und konische Kombinationen der Zeilen von A.*

(b) *$P(D, d) \cap L$ ist die orthogonale Projektion von $P(A, b)$ auf L.*

(3.10) Satz. *Die Projektion eines Polyeders ist ein Polyeder!*

Durch sukzessive Projektion kann man ein Polyeder auf einen beliebigen Unterraum $\{y | By = 0\}$ projizieren.

3.2 Lineare Optimierung und Fourier-Motzkin-Elimination

Die Fourier-Motzkin-Elimination ist nicht nur eine Projektionsmethode, man kann sie auch zur Lösung linearer Programme verwenden. Dies geht wie folgt. Wir beginnen mit einem linearen Programm der Form

$$\begin{aligned} &\max c^T x \\ &Ax \leq a \end{aligned}$$

mit $A \in \mathbb{K}^{m \times n}$, $a \in \mathbb{K}^m$ und setzen $P = P(A, b)$. Wir führen nun eine zusätzliche Variable x_{n+1} ein und fügen zur Matrix A eine zusätzliche Spalte und Zeile hinzu. Diese Matrix nennen wir B. Die $(m + 1)$-te Zeile von B enthält in den ersten n Spalten den Vektor c; wir setzen $b_{i,n+1} = 0$ für $i = 1, \ldots, m$ und $b_{m+1,n+1} = -1$. Wir verlängern den Vektor $a \in \mathbb{K}^m$ um eine Komponente, die den Wert 0 enthält, und nennen diesen $(m + 1)$-Vektor b:

$$B = \begin{pmatrix} A & 0 \\ -c^T & 1 \end{pmatrix}, \quad b = \begin{pmatrix} a \\ 0 \end{pmatrix}$$

Wenden wir die Fourier-Motzkin-Elimination auf $Ax \leq a$ an, so können wir feststellen, ob die Lösungsmenge P des linearen Programms leer ist oder nicht. Führen wir die Fourier-Motzkin-Elimination mit $Bx \leq b$ durch und eliminieren wir nur die ersten n Variablen, so erhalten wir am Ende ein System $D_n x \leq d_n$, welches nichts anderes ist als ein Ungleichungssystem der Form $\alpha_i \leq x_{n+1} \leq \beta_j$, wobei die β_j die positiven und die α_i die negativen Einträge des Vektors d_n sind.

Das Minimum über die Werte β_j liefert den maximalen Wert der Zielfunktion $c^T x$ des gegebenen linearen Programms.

Setzen wir $\overline{x}_i := 0,\ i = 1, \dots, n$ und $\overline{x}_{n+1} := \min\{\beta_j\}$, so können wir mit diesem Vektor $\overline{x} = (\overline{x}_i)$ beginnend durch Rücksubstitution – wie in Satz (3.2) (d) beschrieben – eine Optimallösung von $\max c^T x,\ Ax \leq b$ berechnen.

Wenn wir statt des Maximums von $c^T x$ über P das Minimum finden wollen, so fügen wir bei der Konstruktion der Matrix B statt der Zeile $(c^T, -1)$ die Zeile $(-c^T, 1)$ ein. Wollen wir das Maximum und Minimum gleichzeitig bestimmen, so führen wir die Fourier-Motzkin-Elimination der ersten n Variablen auf der Matrix durch, die um die beiden Zeilen $(c^T, -1)$ und $(-c^T, 1)$ erweitert wurde.

Dieses Verfahren zur Lösung linearer Programme ist konzeptionell extrem einfach und klar. Wenn es in der Praxis funktionieren würde, könnten wir das Thema „lineare Optimierung“ beenden. Als Hausaufgabe haben Sie die Fourier-Motzkin-Elimination implementiert. Versuchen Sie einmal, damit lineare Programme „anständiger Größenordnung“ zu lösen. Sie werden sich wundern und wissen, warum die Vorlesung weitergeht.

Wir beschließen die Diskussion der Fourier-Motzkin-Elimination als Methode zur Lösung linearer Programme durch numerische Berechnung eines Beispiels.

(3.11) Beispiel. *Wir beginnen mit folgendem System von 5 Ungleichungen mit 2 Variablen*

$$
\begin{array}{lrrcr}
(1) & & -x_2 & \leq & 0 \\
(2) & -x_1 & -x_2 & \leq & -1 \\
(3) & -x_1 & +x_2 & \leq & 3 \\
(4) & +x_1 & & \leq & 3 \\
(5) & +x_1 & +2x_2 & \leq & 9
\end{array}
$$

Wir wollen die Zielfunktion

$$x_1 + 3x_2$$

über der Menge der zulässigen Lösungen sowohl maximieren als auch minimieren. Dazu schreiben wir (wie oben erläutert) das folgende Ungleichungssystem

$$
\begin{array}{lrrrcr}
(1) & & -x_2 & & \leq & 0 \\
(2) & -x_1 & -x_2 & & \leq & -1 \\
(3) & -x_1 & +x_2 & & \leq & 3 \\
(4) & +x_1 & & & \leq & 3 \\
(5) & +x_1 & +2x_2 & & \leq & 9 \\
(6) & +x_1 & +3x_2 & -x_3 & \leq & 0 \\
(7) & -x_1 & -3x_2 & +x_3 & \leq & 0
\end{array}
$$

auf und eliminieren die Variable x_1. Das Ergebnis ist das folgende Ungleichungssystem:

$$\begin{array}{rrrrcr}
(1) & (1) & -x_2 & & \leq & 0 \\
(2,4) & (2) & -x_2 & & \leq & 2 \\
(2,5) & (3) & +x_2 & & \leq & 8 \\
(2,6) & (4) & +2x_2 & -x_3 & \leq & -1 \\
(3,4) & (5) & +x_2 & & \leq & 6 \\
(3,5) & (6) & +x_2 & & \leq & 4 \\
(3,6) & (7) & +4x_2 & -x_3 & \leq & 3 \\
(7,4) & (8) & -3x_2 & +x_3 & \leq & 3 \\
(7,5) & (9) & -x_2 & +x_3 & \leq & 9
\end{array}$$

Wir haben oben die Variable x_1 weggelassen. Die erste Spalte zeigt an, woher die neue Ungleichung kommt. So ist z. B. die 5. Ungleichung aus der Kombination der 3. und 4. Ungleichung des vorhergehenden Systems entstanden. Ungleichungen der Form $0x_2 + 0x_3 \leq \alpha$ haben wir weggelassen. Eine solche ist z. B. die Ungleichung , die aus der Kombination der 7. und 6. Ungleichung entsteht.

Die Elimination der Variablen x_2 liefert nun analog

$$\begin{array}{rrrcr}
(1,4) & (\ 1) & -x_3 & \leq & -1 \\
(1,7) & (\ 2) & -x_3 & \leq & 3 \\
(2,4) & (\ 3) & -x_3 & \leq & 3 \\
(2,7) & (\ 4) & -x_3 & \leq & 11 \\
(8,3) & (\ 5) & +x_3 & \leq & 27 \\
(8,4) & (\ 6) & -x_3 & \leq & 3 \\
(8,5) & (\ 7) & +x_3 & \leq & 21 \\
(8,6) & (\ 8) & +x_3 & \leq & 15 \\
(8,7) & (\ 9) & +x_3 & \leq & 21 \\
(9,3) & (10) & +x_3 & \leq & 17 \\
(9,4) & (11) & +x_3 & \leq & 17 \\
(9,5) & (12) & +x_3 & \leq & 15 \\
(9,6) & (13) & +x_3 & \leq & 13 \\
(9,7) & (14) & +3x_3 & \leq & 39
\end{array}$$

Die größte untere Schranke für x_3 ist 1. Sie wird von der ersten Ungleichung geliefert. Die kleinste obere Schranke hat den Wert 13, dieser folgt aus den Ungleichungen 13 und 14. Damit ist 13 der Maximalwert der Zielfunktion, 1 ist der Minimalwert. Setzen wir $x_3 = 13$ in das vorhergehende Ungleichungssystem ein, so ergibt sich der Wert $x_2 = 4$. Durch Substitution dieser beiden Werte in das

Ausgangssystem erhalten wir $x_1 = 1$. *Der maximale Zielfunktionswert wird also im zulässigen Punkt* $x_1 = 1, x_2 = 4$ *angenommen.*

$\square$

Kapitel 4

Das Farkas-Lemma und seine Konsequenzen

Wir werden zunächst Folgerung (3.5) auf verschiedene Weisen formulieren, um sie für den späteren Gebrauch "aufzubereiten". Dann werden wir einige interessante Folgerungen daraus ableiten.

4.1 Verschiedene Versionen des Farkas-Lemmas

(4.1) (allgemeines) Farkas-Lemma. *Für dimensionsverträgliche Matrizen A, B, C, D und Vektoren a, b gilt:*

$$\left.\begin{array}{rl} \text{Es existieren } x, y \text{ mit} & \\ Ax + By & \leq a \\ Cx + Dy & = b \\ x & \geq 0 \end{array}\right\} \quad \dot{\vee} \quad \left\{\begin{array}{rl} \text{Es existieren } u, v \text{ mit} & \\ u^T A + v^T C & \geq 0^T \\ u^T B + v^T D & = 0^T \\ & u \geq 0 \\ u^T a + v^T b & < 0. \end{array}\right.$$

(Hierbei bezeichnet "$\dot{\vee}$" das "entweder oder", d. h. eine der beiden Aussagen gilt, aber niemals beide gleichzeitig, also eine Alternative.)

Beweis : Durch die Transformationsregeln (2.4) führen wir die obige Aussage auf Folgerung (3.5) zurück. Die linke Aussage der Alternative lässt sich schreiben als

"$\exists\, z$ mit $\overline{A}z \leq \overline{a}$" wobei

$$\overline{A} := \begin{pmatrix} A & B \\ C & D \\ -C & -D \\ -I & 0 \end{pmatrix} \qquad \text{und} \qquad \overline{a} = \begin{pmatrix} a \\ b \\ -b \\ 0 \end{pmatrix}.$$

Nach (3.5) hat dieses System genau dann keine Lösung, wenn gilt:

$$\exists\, y^T = (u^T, \overline{v}^T, \overline{\overline{v}}^T, w^T) \geq 0$$

mit $y^T\overline{A} = 0$ und $y^T\overline{a} < 0$. Ausführlich geschrieben heißt dies:

$$\exists \begin{pmatrix} u \\ \overline{v} \\ \overline{\overline{v}} \\ w \end{pmatrix} \geq 0 \quad \text{mit} \quad \begin{array}{rcl} u^TA + \overline{v}^TC - \overline{\overline{v}}^TC - w^T & = & 0^T \\ u^TB + \overline{v}^TD - \overline{\overline{v}}^TD & = & 0^T \\ u^Ta + \overline{v}^Tb - \overline{\overline{v}}^Tb & < & 0. \end{array}$$

Setzen wir $v := \overline{v} - \overline{\overline{v}}$ und betrachten wir w als einen Vektor von Schlupfvariablen, so lässt sich dieses System äquivalent schreiben als:

$$\exists\, u, v \quad \text{mit} \quad \begin{array}{rcl} u^TA + v^TC & \geq & 0^T \\ u^TB + v^TD & = & 0^T \\ u & \geq & 0 \\ u^Ta + v^Tb & < & 0, \end{array}$$

und dies ist das gewünschte Resultat. □

Durch Spezialisierung von (4.1) erhalten wir:

(4.2) (spezielles) Farkas-Lemma. *Es seien $A \in \mathbb{K}^{(m,n)}$ und $b \in \mathbb{K}^m$, dann gilt:*

$$\begin{array}{lllllll} (a) \exists\, x & \textit{mit} & Ax \leq b & \dot\vee & \exists\, u \geq 0 & \textit{mit} & u^TA = 0^T \ \textit{und} \ u^Tb < 0. \\ (b) \exists\, x \geq 0 & \textit{mit} & Ax \leq b & \dot\vee & \exists\, u \geq 0 & \textit{mit} & u^TA \geq 0^T \ \textit{und} \ u^Tb < 0. \\ (c) \exists\, x \geq 0 & \textit{mit} & Ax = b & \dot\vee & \exists\, u & \textit{mit} & u^TA \geq 0^T \ \textit{und} \ u^Tb < 0. \\ (d) \exists\, x & \textit{mit} & Ax = b & \dot\vee & \exists\, u & \textit{mit} & u^TA = 0^T \ \textit{und} \ u^Tb < 0. \end{array}$$

□

Das Farkas-Lemma (4.2) in seiner Version (a) charakterisiert also

(a) *Die Lösbarkeit eines linearen Ungleichungssystems,*
in seiner Version (4.2) (b)

(b) *die nichtnegative Lösbarkeit eines linearen Ungleichungssystems,* in seiner Version (4.2) (c)

(c) *die nichtnegative Lösbarkeit eines linearen Gleichungssystems,* in seiner Version (4.2) (d)

(d) *die Lösbarkeit eines linearen Gleichungssystems,*

wobei (d) natürlich bereits aus der linearen Algebra bekannt ist. Die zweite Alternative in (d) ist nichts anderes als eine Umformulierung von $\text{rang}(A) \neq \text{rang}(A, b)$.

Die linken Alternativen in (4.1) und (4.2) (a), (b), (c), (d) sagen aus, dass gewisse Polyeder nicht leer sind. Die Lösungsmengen der rechten Alternativen sind dagegen keine Polyeder, weil jeweils eine strikte Ungleichung in den Gleichungs- und Ungleichungssystemen vorkommt. Da in allen Fällen die rechten Seiten der rechten Alternativen Nullvektoren sind, sind die Lösungsmengen (ohne die strikte Ungleichung) nach (2.6) Kegel. Folglich können die Lösungsvektoren skaliert werden. Aus dieser Beobachtung folgt:

(4.3) Farkas-Lemma (polyedrische Version). *Es seien $A \in \mathbb{K}^{(m,n)}$ und $b \in \mathbb{K}^m$, dann gilt:*

$$
\begin{aligned}
&(a) P(A,b) && \neq \emptyset \quad \dot{\vee} \quad P^{=}\left(\begin{pmatrix} A^T \\ b^T \end{pmatrix}, \begin{pmatrix} 0 \\ -1 \end{pmatrix}\right) && \neq \emptyset. \\
&(b) P^{+}(A,b) && \neq \emptyset \quad \dot{\vee} \quad P^{+}\left(\begin{pmatrix} -A^T \\ b^T \end{pmatrix}, \begin{pmatrix} 0 \\ -1 \end{pmatrix}\right) && \neq \emptyset. \\
&(c) P^{=}(A,b) && \neq \emptyset \quad \dot{\vee} \quad P\left(\begin{pmatrix} -A^T \\ b^T \end{pmatrix}, \begin{pmatrix} 0 \\ -1 \end{pmatrix}\right) && \neq \emptyset.
\end{aligned}
$$

$\square$

Polyedrische Formulierungen von (4.2) (b), (d) und (4.1) seien dem Leser überlassen.

4.2 Alternativ- und Transpositionssätze

Wir haben das Farkas-Lemma nun in verschiedenen Versionen formuliert. Folgerung (3.5) beschreibt ein zur Unlösbarkeit eines Ungleichungssystems äquivalentes Kriterium. Die Sätze (4.1) und (4.2) haben die Form von Alternativen, während in (4.3) die Lösbarkeit eines Systems mit der Unlösbarkeit eines durch Transposition gewonnenes System festgestellt wird. Man nennt deshalb Sätze des Typs (4.1), (4.2) bzw. (4.3) Alternativ- bzw. Transpositionssätze. Die Literatur ist außerordentlich reich an derartigen Sätzen, da sie — wie wir später noch sehen werden — auf vielerlei Weise nützlich sind. Wir wollen hier noch einige weitere Sätze dieser Art angeben.

(4.4) Satz. *Es seien $A \in \mathbb{K}^{(p,n)}$, $B \in \mathbb{K}^{(q,n)}$, $a \in \mathbb{K}^p$, $b \in \mathbb{K}^q$, dann gilt genau eine der beiden folgenden Alternativen:*

(a) $\exists\, x \in \mathbb{K}^n$ *mit* $Ax \leq a$, $Bx < b$

(b) (b_1) $\exists\, u \in \mathbb{K}^p_+$, $v \in \mathbb{K}^q_+ \setminus \{0\}$ *mit* $u^T A + v^T B = 0^T$, $u^T a + v^T b \leq 0$, *oder*
(b_2) $\exists\, u \in \mathbb{K}^p_+$, *mit* $u^T A = 0^T$, $u^T a < 0$.

Beweis : Angenommen (a) und (b_2) gelten gleichzeitig, dann gibt es also einen Vektor x mit $Ax \leq a$ und einen Vektor $u \in \mathbb{K}^p_+$ mit $u^T A = 0^T$, $u^T a < 0$, was (4.2) (a) widerspricht. Angenommen (a) und (b_1) gelten gleichzeitig, dann gilt: $0^T x = (u^T A + v^T B)x = u^T(Ax) + v^T(Bx) < u^T a + v^T b \leq 0$, ein Widerspruch.

Wir nehmen nun an, dass (a) nicht gilt und wollen zeigen, dass dann (b) gilt. Wir wenden die Fourier-Motzkin-Elimination n-mal iterativ auf $\binom{A}{B}$ und $\binom{a}{b}$ an. Nach (3.4) und erhalten wir nach n Schritten Matrizen C, D und Vektoren c, d, so dass $Ax \leq a$, $Bx < b$ genau dann lösbar ist, wenn $Cx \leq c$, $Dx < d$ lösbar ist, wobei $C = 0$, $D = 0$ gilt. Nach Annahme gilt (a) nicht, also muss es einen Zeilenindex i von C geben mit $c_i < 0$ oder einen Zeilenindex j von D mit $d_j \leq 0$.

Wendet man die Fourier-Motzkin-Elimination nur auf A, a und n-mal sukzessiv an, so erhält man nach (3.4) das System C, c. Das heißt, die Lösbarkeit von $Ax \leq a$ ist äquivalent zur Lösbarkeit von $Cx \leq c$. Ist also $Ax \leq a$ nicht lösbar, so gibt es nach (3.5) einen Vektor $u \geq 0$ mit $u^T A = 0^T$, $u^T a < 0$, das heißt (b_2) gilt. Ist $Ax \leq a$ lösbar, so gibt es keinen Vektor $u \geq 0$ mit $u^T A = 0^T$, $u^T a < 0$, d. h. $c_i \geq 0$ für alle i. Folglich muss $d_j \leq 0$ für ein j gelten. $D_{j.} = 0^T$ ist eine konische Kombination von Zeilen von A und Zeilen von B. Nach Definition, siehe (3.4), muss dabei mindestens eine Zeile von B mit einem positiven Multiplikator beteiligt sein, also gibt es $u \in K^p_+$, $v \in \mathbb{K}^q_+ \setminus \{0\}$ mit $u^T A + v^T B = D_{j.} = 0^T$ und $u^T a + v^T b = d_j \leq 0$, das heißt (b_1) gilt. □

(4.5) Folgerung. Ist $P(A, a) \neq \emptyset$, dann gilt genau eine der beiden folgenden Alternativen:

(a) $\exists\, x \in \mathbb{K}^n$ mit $Ax \leq a$, $Bx < b$,

(b) $\exists \binom{u}{v} \in \mathbb{K}_+^{p+q}$, $v \neq 0$ mit $u^T A + v^T B = 0^T$ und $u^T a + v^T b \leq 0$.

□

(4.6) Satz (Gordan (1873)). *Es gilt genau eine der beiden folgenden Alternativen:*

(a) $\exists\, x$ *mit* $Ax < 0$,

(b) $\exists\, u \geq 0$, $u \neq 0$ *mit* $u^T A = 0^T$.

Beweis : Setze in (4.5) $B := A$ (wobei A die obige Matrix ist), $b := 0$, $A := 0$, $a := 0$. □

(4.7) Satz (Stiemke (1915)). *Es gilt genau eine der folgenden Alternativen:*

(a) $\exists\, x > 0$ *mit* $Ax = 0$,

(b) $\exists\, u$ *mit* $u^T A \geq 0^T$, $u^T A \neq 0$.

Beweis : Setze in (4.5) $B := -I$, $b = 0$, $A := \begin{pmatrix} A \\ -A \end{pmatrix}$, $a := 0$. □

Der Satz von Stiemke charakterisiert also die strikt positive Lösbarkeit eines homogenen Gleichungssystems, während der Satz von Gordan die semipositive Lösbarkeit des homogenen Gleichungssytems $u^T A = 0$ kennzeichnet. Eine Sammlung weiterer Alternativsätze kann man in

O. L. Mangasarian, “Nonlinear Programming”, McGraw-Hill, New York, 1969 (80 QH 400 M 277)

finden.

4.3 Trennsätze

So genannte Trennsätze spielen in der Konvexitätstheorie und — in noch allgemeinerer Form — in der Funktionalanalysis eine wichtige Rolle. Diese Sätze sind

i. a. vom folgenden Typ: Seien S und T zwei Mengen, dann gibt es unter gewissen Voraussetzungen eine Hyperebene H, die S und T trennt. Geometrisch heißt dies, dass S auf der einen, T auf der anderen Seite von H liegt. Genauer: Sei $H = \{x | c^T x = \gamma\}, c \neq 0$, eine Hyperebene, dann sagen wir, dass H zwei Mengen P und Q **trennt**, falls $P \cup Q \not\subseteq H$ und $P \subseteq \{x | c^T x \leq \gamma\}, Q \not\subseteq \{x | c^T x \geq \gamma\}$ gilt.Wir sagen, dass H die Mengen P und Q **strikt trennt**, falls $P \subseteq \{x | c^T x < \gamma\}$ und $Q \not\subseteq \{x | c^T x > \gamma\}$.

Das Farkas-Lemma impliziert einige interessante Trennsätze für Polyeder $P \subseteq \mathbb{K}^n$. Wir wollen hier einen Satz dieser Art angeben, der als Prototyp für die allgemeineren Trennsätze angesehen werden kann.

(4.8) Trennsatz. *Es seien $P = P(A, a)$ und $Q = P(B, b)$ zwei Polyeder im $\mathbb{K}^n$. Es gibt eine P und Q strikt trennende Hyperebene genau dann, wenn $P \cap Q = \emptyset$ gilt, es sei denn, eines der Polyeder ist leer und das andere der $\mathbb{K}^n$.*

Beweis : Gibt es eine P und Q strikt trennende Hyperebene, dann ist $P \cap Q$ offensichtlich leer.

Sei $P \cap Q = \emptyset$. Zu zeigen ist die Existenz eines Vektors $c \in \mathbb{K}^n$, $c \neq 0$, und einer Zahl $\gamma \in \mathbb{K}$, so dass $P \subseteq \{x | c^T x < \gamma\}$ und $P \subseteq \{x | c^T x > \gamma\}$ gilt. Sind P und Q leer, so leistet jede beliebige Hyperebene das Gewünschte. Ist eines der Polyeder leer, sagen wir $Q = \emptyset$, $P \neq \emptyset$, dann ist wegen $P \neq \mathbb{K}^n$ eine der Zeilen von A von Null verschieden, sagen wir $A_{i.} \neq 0$. Dann haben $c^T := A_{i.}$, $\gamma := a_i + 1$ die geforderten Eigenschaften. Sind P und Q nicht leer, dann gilt nach Voraussetzung

$$P\left(\begin{pmatrix} A \\ B \end{pmatrix}, \begin{pmatrix} a \\ b \end{pmatrix}\right) = \emptyset.$$

Folglich existiert nach (4.2) (a) ein Vektor $\binom{u}{v} \geq 0$ mit $u^T A + v^T B = 0^T$ und $u^T a + v^T b < 0$. Wir setzen

$$c^T := u^T A (= -v^T B) \text{ und } \gamma := \tfrac{1}{2}(u^T a - v^T b).$$

Daraus folgt:

$$\begin{aligned} x \in P \Rightarrow \quad c^T x &= u^T A x \leq u^T a < u^T a - \tfrac{1}{2}(u^T a + v^T b) &&= \tfrac{1}{2}(u^T a - v^T b) = \gamma. \\ x \in Q \Rightarrow \quad c^T x &= -v^T B x \geq -v^T b > -v^T b + \tfrac{1}{2}(u^T a + v^T b) &&= \tfrac{1}{2}(u^T a - v^T b) = \gamma. \end{aligned}$$

Der Vektor c ist offenbar von Null verschieden. □

Der folgende Satz, den wir nicht mit den bisher entwickelten Methoden beweisen können, soll als Beispiel für die oben angesprochene allgemeine Art von Trennsätzen dienen.

(4.9) Trennsatz für konvexe Mengen. *Es seien P und Q zwei nichtleere konvexe Teilmengen des $\mathbb{R}^n$. Es gibt eine P und Q trennende Hyperebene $H = \{x \in \mathbb{R}^n \mid c^T x = \gamma\}$ (d. h. $P \cup Q \not\subseteq H$, $P \subseteq \{x \in \mathbb{R}^n \mid c^T x \leq \gamma\}$, $Q \subseteq \{x \in \mathbb{R}^n \mid c^T x \geq \gamma\}$) genau dann, wenn der Durchschnitt des relativ Inneren von P mit dem relativ Inneren von Q leer ist.*

Beweis : Siehe Stoer & Witzgall (1970), Seite 98, Satz (3.3.9) oder Leichtweiss, "Konvexe Mengen", VEB Deutscher Verlag der Wissenschaften, Berlin, 1980, Seite 28, Satz 3.1. $\square$

Der aufmerksame Leser wird bemerkt haben, dass in (4.9) der Vektorraum $\mathbb{R}^n$ (und nicht wie üblich $\mathbb{K}^n$) benutzt wurde. In der Tat ist Satz (4.9) für konvexe Teilmengen von $\mathbb{Q}^n$ im folgenden Sinne falsch. Es ist nicht immer möglich, zwei konvexe Mengen in $\mathbb{Q}^n$, deren relativ innere Mengen kein gemeinsames Element besitzen, durch eine Hyperebene $c^T x = \gamma$ zu trennen, so dass der Vektor $(c^T, \gamma)^T$ rational ist. Man betrachte nur $P = \{x \in \mathbb{Q} \mid x < \sqrt{2}\}$, $Q = \{x \in \mathbb{Q} \mid x > \sqrt{2}\}$. Dieses Beispiel zeigt auch, dass die bisher entwickelten konstruktiven Beweistechniken, die ja für $\mathbb{Q}$ und $\mathbb{R}$ gleichermaßen arbeiten, nicht mächtig genug sind, um (4.9) abzuleiten. Offenbar muss das Vollständigkeitsaxiom der reellen Zahlen auf irgendeine Weise benutzt werden.

Hausaufgabe. Leiten Sie Satz (4.10) aus Folgerung (4.5) ab.

Satz (4.9) können wir jedoch für Polyeder beweisen. Zu seiner Formulierung müssen wir jedoch schärfere Anforderungen an die Darstellung der involvierten Polyeder stellen.

(4.10) Schwacher Trennsatz.

Seien $P = \{x \in \mathbb{K}^n \mid Ax = a, Bx \leq b\}$, $Q = \{x \in \mathbb{K}^n \mid Cx = c, Dx \leq d\}$ zwei nichtleere Polyeder, so dass keine der Ungleichungen $Bx \leq b$ von allen Punkten in P und keine der Ungleichungen $Dx \leq d$ von allen Punkten in Q mit Gleichheit erfüllt wird. Es gibt eine P und Q trennende Hyperebene genau dann, wenn $R := \{x \in \mathbb{K}^n \mid Ax = a, Cx = c, Bx < b, Dx < d\}$ leer ist.

Beweis : Ist $P \cap Q = \emptyset$, so ist $R = \emptyset$, und die Behauptung folgt (in schärferer Form) aus Satz (4.8). Sei also $P \cap Q \neq \emptyset$.

Die Menge R ist offenbar die Lösungsmenge des Systems

$Ax \leq a, -Ax \leq -a, Cx \leq c, -Cx \leq -c, Bx \leq b, -Bx \leq -b, Dx \leq d, -Dx \leq -d.$

Dieses hat nach Folgerung (4.5) genau dann keine Lösung, wenn es einen Vektor $(\overline{u}^T, \overline{\overline{u}}^T, \overline{v}^T, \overline{\overline{v}}^T, w^T, y^T) \neq 0^T$ gibt mit

$$\overline{u}^T A - \overline{\overline{u}}^T A + \overline{v}^T C - \overline{\overline{v}}^T C + w^T B + y^T D = 0^T,$$

$$\overline{u}^T a - \overline{\overline{u}}^T a + \overline{v}^T c - \overline{\overline{v}}^T c + w^T b + y^T d \leq 0^T.$$

Setzen wir $u := \overline{u} - \overline{\overline{u}}, v := \overline{v} - \overline{\overline{v}}$ und

$$\begin{aligned} r^T &:= u^T A + w^T B (= -v^T C - y^T D), \\ \rho &:= \frac{1}{2}(u^T a + w^T b - v^T c - y^T d) \end{aligned}$$

und $H =: \{x \mid r^T x = \rho\}$. Dann ist H eine P und Q trennende Hyperebene, denn

$$\begin{aligned} (1) \quad x \in P \Longrightarrow r^T x &= u^T Ax + w^T Bx \leq u^T a + w^T b \\ &\leq u^T ax + w^T bx - \tfrac{1}{2}(u^T a + w^T b - v^T c - y^T d) \\ &\rho \\ (2) \quad x \in Q \Longrightarrow r^T x &= v^T Cx - y^T Dx \geq -v^T c - y^T d \\ &\geq -v^T c - y^T d + \tfrac{1}{2}(u^T a + w^T b - v^T c - y^T d) \\ &\rho. \end{aligned}$$

Aus der Voraussetzung folgt, dass Punkte $p \in P$ und $q \in Q$ existieren mit $Bp < b$ und $Dq < d$. Die Abschätzungen (1), (2) zeigen, dass aus $(w^T, y^T) \neq 0^T$ folgt, dass $v^T p < \rho$ oder $v^T q > \rho$ gilt. Somit gilt $v \neq 0$ und $P \cup Q \not\subseteq H$; das aber heißt, H ist eine trennende Hyperebene.

Gibt es eine P und Q trennende Hyperebene $H = \{x \mid r^T x = p\}$, dann gibt es einen Punkt in $P \cup Q$, der nicht in H ist; sagen wir $P \not\subseteq H$. Daraus folgt $P = \{x \mid Ax = a, Bx \leq b, v^T x \leq \rho\}$ und $\{x \mid Ax = a, Bx < b\} = \{x \mid Ax = q, Bx < b, v^T x < \rho\}$ und somit $R = \{x \mid Ax = a, Cx = c, Bx < b, v^T x < \rho, Dx < d\} = \emptyset$, dann $\{x \mid Cx = c, Dx < d, v^T x < \rho\} = \emptyset$, weil $Q = \{x \mid Cx = c, Dx \leq d\} \subseteq \{x \mid v^T x \geq \rho\}$. $\square$

Satz (4.10) folgt aus (4.9), wenn man sich überlegt, dass für ein Polytop $P = \{x \mid Ax = a, Bx \leq b\}$ mit der Eigenschaft, dass keine der Ungleichungen in $Bx \leq b$ durch alle Punkte aus P mit Gleichheit erfüllt ist, die Menge $\{x \mid Ax, Bx < b\}$ das relativ Innere von P ist.

Kapitel 5

Der Dualitätssatz der linearen Programmierung

In Satz (1.5) haben wir bereits den sogenannten schwachen Dualitätssatz angegeben und bewiesen. Die Gleichheit der optimalen Zielfunktionswerte kann man (unter anderem) mit Hilfe des Farkas-Lemmas beweisen. Man kann den (nachfolgend formulierten) Dualitätssatz als eine Optimierungsversion des Farkas-Lemmas ansehen. Beide Sätze sind in dem Sinne äquivalent, dass der eine ohne Mühe aus dem anderen gefolgert werden kann und umgekehrt. So wie das Farkas Lemma für die Polyedertheorie von zentraler Bedeutung ist, ist der Dualitätssatz von außerordentlicher Tragweite in der linearen Optimierung und zwar sowohl in algorithmisch-praktischer als auch in theoretischer Hinsicht. Wenn man heutzutage in mathematischen (Forschungs-)Aufsätzen einen Satz wie "... it follows by an LP-argument ..." liest, dann heißt das so gut wie immer, dass der Dualitätssatz in einer seiner Versionen geschickt anzuwenden ist. Wegen seiner Bedeutung widmen wir dem Dualitätssatz ein eigenes Kapitel. (Er hätte natürlich auch in Kapitel 4 eingearbeitet werden können.)

Wir wollen zunächst das folgende lineare Programm

$$\begin{aligned} &\max\ c^T x \\ &Ax \le b \end{aligned} \tag{5.1}$$

wobei $A \in \mathbb{K}^{(m,n)}$, $b \in \mathbb{K}^m$ und $c \in \mathbb{K}^n$ gegeben sind, untersuchen. Unsere Aufgabe besteht darin, unter den Vektoren $x \in \mathbb{K}^n$, die **zulässig** für (5.1) sind, d. h. die $Ax \le b$ erfüllen, einen Vektor $x^* \in \mathbb{K}^n$ zu finden, der **optimal** ist, d. h. $c^T x^* \ge c^T x$ für alle zulässigen x. In Kapitel 1 haben wir bereits eine Motivation für die Dualitätstheorie gegeben, hier wollen wir noch einen weiteren Zugang beschreiben. Zunächst wollen wir (5.1) polyedrisch darstellen. Wir führen dazu eine neue Variable z ein und schreiben (5.1) in der Form

$$\begin{aligned} &\max z \\ (5.2) \qquad &\begin{pmatrix} 1 & -c^T \\ 0 & A \end{pmatrix} \begin{pmatrix} z \\ x \end{pmatrix} \leq \begin{pmatrix} 0 \\ b \end{pmatrix} \end{aligned}$$

Daraus folgt, dass jedes lineare Programm in ein LP umgeformt werden kann, bei dem lediglich Lösungsvektoren gesucht werden, deren erste Komponente so groß wie möglich ist. Schreiben wir (5.2) noch einmal um und bringen wir z auf die rechte Seite, so kann (5.2) folgendermaßen geschrieben werden.

$$\begin{aligned} &\max z \\ (5.3) \qquad &\begin{pmatrix} -c^T \\ A \end{pmatrix} x \leq \begin{pmatrix} -z \\ b \end{pmatrix}. \end{aligned}$$

Für festes $z \in \mathbb{K}$ ist die Lösungsmenge von (5.3) ein Polyeder im $\mathbb{K}^n$. Wir setzen für alle $z \in \mathbb{K}$

$$(5.4) \qquad P_z := \left\{ x \in \mathbb{K}^n \mid \begin{pmatrix} -c^T \\ A \end{pmatrix} x \leq \begin{pmatrix} -z \\ b \end{pmatrix} \right\}.$$

Nunmehr können wir (5.3) wie folgt darstellen:

$$(5.5) \qquad \max\{z \mid P_z \neq \emptyset\}.$$

Wir suchen also ein $z \in \mathbb{K}$, das so groß wie möglich ist unter der Nebenbedingung, dass das Polyeder $P_z \neq \emptyset$ ist.

Üblicherweise nennt man die Aufgabe zu zeigen, dass eine Menge nicht leer ist, ein **primales Problem**. Zu zeigen, dass eine Menge leer ist, ist ein **duales Problem**. Diese beiden Aufgaben sind i. a. nicht von gleicher Schwierigkeit. Betrachten wir z. B. ein Polyeder $P \subseteq \mathbb{Q}^n$. Legen wir einfach eine Abzählung $x_1, x_2, x_3, \ldots$ der Elemente von $\mathbb{Q}^n$ fest und überprüfen wir, ob $x_i \in P$ ist für $i = 1, 2, \ldots$, so sind wir sicher, dass dieses Verfahren nach endlich vielen Schritten abbricht, falls $P \neq \emptyset$ ist. Jedoch kann dieses Verfahren niemals nach endlich vielen Schritten folgern, dass P leer ist. Ein Ziel von Dualitätstheorien ist es, "gute Kriterien" für die Lösung der dualen Probleme zu finden. Das Farkas-Lemma liefert ein solches. Gilt $P = \{x \mid Ax \leq b\}$, so ist nach (4.3) (a) P genau dann leer, wenn $Q = P^{=}(\binom{A^T}{b^T}, \binom{0}{-1})$ nicht leer ist. Durch Anwendung des obigen Abzählverfahrens auf P und Q gleichzeitig können wir also nach endlich vielen Schritten entscheiden, ob P leer ist oder nicht.

Das angegebene Abzählverfahren sollte nicht als ein ernsthafter Vorschlag zur algorithmischen Lösung des primalen und dualen Problems angesehen werden. Wir werden später sehen, dass dies alles viel besser und schneller gemacht werden kann. Diese Bemerkungen sollten lediglich das Problembewusstsein des Lesers schärfen!

Zurück zu (5.5). Es kann sein, dass $P_z \neq \emptyset$ gilt für alle $z \in \mathbb{K}$. Dann hat (5.1) keine Optimallösung. Andernfalls können wir versuchen den Maximalwert in (5.5) nach oben zu beschränken. Die bestmögliche Schranke ist natürlich gegeben durch

$$\inf\{z \mid P_z = \emptyset\}. \tag{5.6}$$

Nach (4.2) (a) ist $P_z = \emptyset$ äquivalent dazu, dass das System $y^T A = \lambda c^T$, $y^T b < \lambda z$, $y \geq 0$, $\lambda \geq 0$ eine Lösung hat. Man überlegt sich leicht, dass dieses System — im Falle der Lösbarkeit von (5.1) — genau dann lösbar ist, wenn $y^T A = c^T$, $y^T b < z$, $y \geq 0$ lösbar ist. Wenn wir das kleinste z in (5.6) finden wollen, müssen wir also einen möglichst kleinen Wert $y^T b$ bestimmen. (5.6) ist somit äquivalent zu

$$\begin{array}{ll} \min & y^T b \\ y^T A & = c^T \\ y & \geq 0 \end{array} \tag{5.7}$$

Problem (5.7) ist offenbar wiederum ein lineares Programm. Wir nennen es das zum **primalen Problem** (5.1) **duale lineare Programm**. Wir wollen nun zeigen, dass (5.1) und (5.7) den gleichen Optimalwert haben.

(5.8) Dualitätssatz. *Es seien $A \in \mathbb{K}^{(m,n)}$, $b \in \mathbb{K}^m$ und $c \in \mathbb{K}^n$, dann haben die beiden linearen Programme*

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

optimale Lösungen, deren Zielfunktionswerte gleich sind, genau dann, wenn sie zulässige Lösungen besitzen.

Beweis : Wenn (P) und (D) optimale Lösungen haben, dann haben sie auch zulässige. Haben (P) und (D) zulässige Lösungen, so gilt nach (1.5), dass der Wert von (P) nicht größer als der von (D) ist. (P) und (D) haben zulässige Lösungen mit gleichem Zielfunktionswert genau dann, wenn das System

$$(1)\qquad \begin{array}{rcl} \begin{pmatrix} 0 \\ b^T \end{pmatrix} y + \begin{pmatrix} A \\ -c^T \end{pmatrix} x & \leq & \begin{pmatrix} b \\ 0 \end{pmatrix} \\ A^T y & = & c \\ y & \geq & 0 \end{array}$$

eine Lösung hat. Dies ist nach (4.1) äquivalent dazu, dass das System

$$(2)\qquad \begin{array}{rcl} zb^T + v^T A^T & \geq & 0^T \\ u^T A - zc^T & = & 0^T \\ u & \geq & 0 \\ z & \geq & 0 \\ u^T b + v^T c & < & 0 \end{array}$$

keine Lösung hat. Nehmen wir an, dass (2) eine Lösung (u^T, v^T, z) hat. Gibt es eine Lösung von (2) mit $z = 0$, so hat nach (4.1) das System

$$\begin{array}{rrl} & Ax & \leq b \\ A^T y & & = c \\ y & & \geq 0 \end{array}$$

keine Lösung. Das aber heißt, dass (P) oder (D) keine zulässige Lösung haben. Widerspruch! Gibt es eine Lösung von (2) mit $z > 0$, so können wir durch Skalieren eine Lösung finden mit $z = 1$. Daraus folgt

$$0 > u^T b + v^T c \geq -u^T A v + v^T A^T u = 0.$$

Widerspruch! Dies impliziert, dass (2) inkonsistent ist. Also hat nach (4.1) das System (1) eine Lösung, und Satz (5.8) ist bewiesen. □

(5.9) Folgerung. P bzw. D seien die Lösungsmengen von (P) bzw. (D). Wir setzen

$$z^* := \begin{cases} +\infty, & \text{falls (P) unbeschränkt,} \\ -\infty, & \text{falls } P = \emptyset, \\ \max\{c^T x \mid x \in P\}, & \text{andernfalls,} \end{cases}$$

$$u^* \;:=\; \begin{cases} -\infty, & \text{falls (D)unbeschränkt,} \\ +\infty, & \text{falls } D = \emptyset, \\ \min\{y^T b, y \in D\}, & \text{andernfalls.} \end{cases}$$

Dann gilt

(a) $-\infty < z^* = u^* < +\infty \iff z^*$ endlich $\iff u^*$ endlich.

(b) $z^* = +\infty \;\Rightarrow\; D = \emptyset$.

(c) $u^* = -\infty \;\Rightarrow\; P = \emptyset$.

(d) $P = \emptyset \;\Rightarrow\; D = \emptyset$ oder $u^* = -\infty$.

(e) $D = \emptyset \;\Rightarrow\; P = \emptyset$ oder $z^* = +\infty$.

Beweis : (a) Aus $z^* = u^*$ endlich folgt natürlich u^* und z^* endlich.

Sei z^* endlich, dann ist $P \neq \emptyset$ und

$$(1) \qquad \begin{aligned} -c^T x &\leq -z \\ Ax &\leq b \end{aligned}$$

ist für $z = z^*$ lösbar, jedoch unlösbar für jedes $z > z^*$. Sei also $z > z^*$, dann ist nach (4.2) (a)

$$(2) \qquad \begin{array}{rrcl} -uc^T & +\, y^T A & = & 0^T \\ -uz & +\, y^T b & < & 0 \\ & u, y & \geq & 0 \end{array}$$

lösbar. Hat dieses System eine Lösung mit $u = 0$, so hat $y^T A = 0, y^T b < 0, y \geq 0$ eine Lösung. Also folgt aus (4.2) (a), dass $P = \emptyset$, ein Widerspruch. Also gibt es eine Lösung von (2) mit $u > 0$. Durch Skalieren können wir eine derartige Lösung mit $u = 1$ finden. Daraus folgt, dass das System

$$\begin{aligned} y^T b &< z \\ y^T A &= c^T \\ y &\geq 0 \end{aligned}$$

eine Lösung besitzt. Folglich ist $D \neq \emptyset$, und aus Satz (5.8) folgt die Behauptung.

Ist u^* endlich, so verläuft der Beweis analog.

(b) Ist $D \neq \emptyset$, so impliziert der schwache Dualitätssatz (1.5)$z^* \leq \min\{y^T b \mid y \in D\} = u^* < +\infty$.

(c) Analog zu (b).

(d) Angenommen $P = \emptyset$ und $u^* > -\infty$. Dann gilt entweder $u^* = +\infty$ und somit $D = \emptyset$ oder nach (a) folgt $z^* = u^*$ endlich, also $P \neq \emptyset$. Widerspruch!

(e) Analog zu (d). $\square$

Man könnte vermuten, dass in (5.9) (b) und (c) auch die umgekehrte Richtung gilt. Die Aussagen (d) und (e) zeigen aber, dass beim Versuch eines Beweises der Rückrichtung Komplikationen auftreten können. Das nachfolgende Beispiel zeigt, dass es in der Tat zueinander duale lineare Programme gibt, die beide leere Lösungsmengen haben.

(5.10) Beispiel. Es seien

$$\begin{aligned} P &= \left\{ x \in \mathbb{K}^2 \mid \begin{pmatrix} 1 & -1 \\ -1 & 1 \end{pmatrix} x \leq \begin{pmatrix} 0 \\ -1 \end{pmatrix} \right\}, \\ D &= \left\{ y \in \mathbb{K}^2 \mid \begin{pmatrix} 1 & -1 \\ -1 & 1 \end{pmatrix} y = \begin{pmatrix} 1 \\ 1 \end{pmatrix}, y \geq 0 \right\}. \end{aligned}$$

Dann sind die linearen Programme

$$\begin{array}{lcr} \max\ x_1 + x_2 & \text{und} & \min -y_2 \\ x \in P & & y \in D \end{array}$$

zueinander dual, und es gilt:

(a) $P = \emptyset$, da nach (4.2) (a) das System $(u_1, u_2) \begin{pmatrix} 1 & -1 \\ -1 & 1 \end{pmatrix} = (0, 0)$,

$u_1, u_2 \geq 0$, $-u_2 < 0$ eine Lösung hat (z. B. $u_1 = u_2 = 1$).

(b) $D = \emptyset$, da nach (4.2) (c) das System $(v_1, v_2) \begin{pmatrix} 1 & -1 \\ -1 & 1 \end{pmatrix} \geq (0, 0)$,

$v_1 + v_2 < 0$, eine Lösung hat (z. B. $v_1 = v_2 = -1$).

□

Wir haben bereits die Sprechweise “zueinander duale lineare Programme” benutzt, ohne dafür eine Begründung zu geben. Sie erfolgt hiermit:

(5.11) Bemerkung. Gegeben seien dimensionsverträgliche Matrizen A, B, C, D und Vektoren a, b, c, d, dann gilt: Das zum **primalen linearen Programm**

$$(5.12) \qquad \begin{array}{ll} \max c^T x + d^T y & \\ Ax + By & \leq a \\ Cx + Dy & = b \\ x & \geq 0 \end{array}$$

duale lineare Programm ist

$$(5.13) \qquad \begin{array}{ll} \min u^T a + v^T b & \\ u^T A + v^T C & \geq c^T \\ u^T B + v^T D & = d^T \\ u & \geq 0. \end{array}$$

Das zum linearen Programm (5.13) duale Programm ist (5.12).

Beweis : Wir benutzen die Transformationsregeln (2.4) und schreiben (5.12) in der Form

$$\begin{array}{lcr} \max c^T x + d^T y & & \\ Ax + By & \leq & a \\ Cx + Dy & \leq & b \\ -Cx - Dy & \leq & -b \\ -Ix \; + \; 0y & \leq & 0 \end{array}$$

Das hierzu duale Progamm ist nach Definition

$$\begin{array}{rl} \min\ u^Ta + v_1^Tb - v_2^Tb & \\ u^TA + v_1^TC - v_2^TC - w^T & = c^T \\ u^TB + v_1^TD - v_2^TD & = d^T \\ u, v_1, v_2, w & \geq 0. \end{array}$$

Setzen wir $v := v_1 - v_2$ und lassen wir w weg, so erhalten wir (5.13). Analog folgt, dass (5.12) dual zu (5.13) ist. □

Von nun an können wir also sagen, dass das duale Programm zum dualen Programm das primale ist, bzw. von einem Paar dualer Programme sprechen. Wir wollen nun zur Übersicht und als Nachschlagewerk in Tabelle 5.1 eine Reihe von Paaren dualer Programme auflisten. Die Korrektheit der Aussagen ergibt sich direkt aus (5.11).

primales LP		duales LP	
(P_1)	$\max c^Tx,\ Ax \leq b,\ x \geq 0$	(D_1)	$\min y^Tb,\ y^TA \geq c^T,\ y \geq 0$
(P_2)	$\min c^Tx,\ Ax \geq b,\ x \geq 0$	(D_1)	$\max y^Tb,\ y^TA \leq c^T,\ y \geq 0$
(P_3)	$\max c^Tx,\ Ax = b,\ x \geq 0$	(D_3)	$\min y^Tb,\ y^TA \geq c^T$
(P_4)	$\min c^Tx,\ Ax = b,\ x \geq 0$	(D_4)	$\max y^Tb,\ y^TA \leq c^T$
(P_5)	$\max c^Tx,\ Ax \leq b$	(D_5)	$\min y^Tb,\ y^TA = c^T,\ y \geq 0$
(P_6)	$\min c^Tx,\ Ax \geq b$	(D_6)	$\max y^Tb,\ y^TA = c^T,\ y \geq 0$

Tabelle 5.1

Aus den obigen primal-dual Relationen kann man die folgenden Transformationsregeln in Tabelle 5.2 ableiten.

primal	dual
Gleichung oder Ungleichung	Variable
Ungleichung	nichtnegative Variable
Gleichung	nicht vorzeichenbeschränkte Variable
nichtnegative Variable	Ungleichung
nicht vorzeichenbeschränkte Variable	Gleichung

Tabelle 5.2

Speziell bedeutet dies z. B. bezüglich des LP $\max\ c^Tx, Ax \leq b, x \geq 0$

– Jeder primalen Ungleichung $A_{i.}x \leq b_i$ ist eine nichtnegative duale Variable y_i zugeordnet.

– Jeder primalen nichtnegativen Variablen x_j ist die duale Ungleichung $y^T A_{.j} \geq c_j$ zugeordnet.

Aus dem Vorhergehenden dürfte klar sein, dass der Dualitätssatz nicht nur für das spezielle Paar linearer Programme (P), (D) aus Satz (5.8) gilt, sondern für alle Paare dualer linearer Programme. Um später darauf Bezug nehmen zu können, wollen wir den Dualitätssatz in voller Allgemeinheit formulieren.

(5.14) Allgemeiner Dualitätssatz. *Es seien (P) ein lineares Programm und (D) das zu (P) duale Programm.*

(a) Haben (P) und (D) zulässige Lösungen, so haben (P) und (D) Optimallösungen und die Zielfunktionswerte aller Optimallösungen stimmen überein.

(b) Hat eines der beiden Programme (P) oder (D) eine Optimallösung, so auch das andere

(c) Hat eines der Programme (P) oder (D) keine zulässige Lösung, so ist das andere unbeschränkt oder hat keine zulässige Lösung.

(d) Ist eines der Programme (P) oder (D) unbeschränkt, so hat das andere keine zulässige Lösung.

Wir wollen nun zwei weitere wichtige Sätze zur Charakterisierung von Optimallösungen linearer Programme formulieren und beweisen.

(5.15) Satz vom schwachen komplementären Schlupf. *Es seien A, B, C, D und a, b, c, d dimensionsverträglich und (5.12), (5.13) das zugehörige Paar dualer linearer Programme. Die Vektoren $\binom{\overline{x}}{\overline{y}}$ bzw. $\binom{\overline{u}}{\overline{v}}$ seien zulässig für (5.12) bzw. (5.13), dann sind die folgenden Aussagen äquivalent:*

(a) $\binom{\overline{x}}{\overline{y}}$ ist optimal für (5.12) und $\binom{\overline{u}}{\overline{v}}$ ist optimal für (5.13).

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

(c) Für alle Komponenten $\overline{u}_i$ der Duallösung gilt: $\overline{u}_i > 0 \;\Rightarrow\; A_{i.}\overline{x} + B_{i.}\overline{y} = a_i$. Für alle Komponenten $\overline{x}_j$ der Primallösung gilt: $\overline{x}_j > 0 \;\Rightarrow\; \overline{u}^T A_{.j} + \overline{v}^T C_{.j} = c_j$.

(d) Für alle Zeilenindizes i von A und B gilt: $A_i.\overline{x} + B_i.\overline{y} < a_i \;\Rightarrow\; \overline{u}_i = 0$.
Für alle Spaltenindizes j von A und C gilt: $\overline{u}^T A_{.j} + \overline{v}^T C_{.j} > c_j \;\Rightarrow\; \overline{x}_j = 0$.

Beweis :

(a) $\Longleftrightarrow\; c^T\overline{x} + d^T\overline{y} = \overline{u}^T a + \overline{v}^T b$ (nach (5.14))
$\Longleftrightarrow\; c^T\overline{x} + (\overline{u}^T B + \overline{v}^T D)\overline{y} - \overline{u}^T a - \overline{v}^T(C\overline{x} + D\overline{y}) = 0$
$\Longleftrightarrow\; c^T\overline{x} + \overline{u}^T B\overline{y} + \overline{v}^T D\overline{y} - \overline{u}^T a - \overline{v}^T C\overline{x} + \overline{v}^T D\overline{y} - \overline{u}^T A\overline{x} + \overline{u}^T A\overline{x} = 0$
$\Longleftrightarrow\; c^T\overline{x} - (\overline{u}^T A + \overline{v}^T C)\overline{x} - \overline{u}^T a + \overline{u}^T(A\overline{x} + B\overline{y}) = 0$
$\Longleftrightarrow$ (b)

(b) $\Longrightarrow$ (c)

Es seien $t^T := c^T - (\overline{u}^T A + \overline{v}^T C)$ und $s := a - (A\overline{x} + B\overline{y})$. Nach Voraussetzung gilt also $t \leq 0$ und $s \geq 0$. Aus $\overline{x} \geq 0$ und $\overline{u} \geq 0$ folgt daher $t^T\overline{x} \leq 0$ und $\overline{u}^T s \geq 0$, mithin $t^T\overline{x} - \overline{u}^T s \leq 0$. Es kann also $t^T\overline{x} - \overline{u}^T s = 0$ nur dann gelten, wenn $t^T\overline{x} = 0$ und $\overline{u}^T s = 0$. Daraus ergibt sich (c).
(c) $\Longrightarrow$ (b) trivial.
(c) $\Longleftrightarrow$ (d) durch Negation.

$\square$

Aus (5.15) folgt für jedes Paar dualer linearer Programme durch Spezialisierung die Gültigkeit eines Satzes vom schwachen komplementären Schlupf.

(5.16) Satz vom starken komplementären Schlupf. *Besitzen beide dualen linearen Programme*

$$\text{(P)} \quad \begin{array}{r} \max\ c^T x \\ Ax \leq b \end{array} \qquad \text{und} \qquad \text{(D)} \quad \begin{array}{r} \min\ u^T b \\ u^T A = c^T \\ u \geq 0 \end{array}$$

zulässige Lösungen, so existieren optimale Lösungen $\overline{x}, \overline{u}$, so dass für alle Zeilenindizes i von A gilt:

$$\begin{array}{rcl} \overline{u}_i > 0 & \Longleftrightarrow & A_i.\overline{x} = b_i \\ \text{bzw.}(A_i.\overline{x} < b_i & \Longleftrightarrow & \overline{u}_i = 0) \end{array}$$

Beweis : Aufgrund von Satz (5.15) genügt es zu zeigen, dass optimale Lösungen $\overline{x}, \overline{u}$ existieren mit

$$\begin{array}{lrcl} \text{(i)} & A_i.\overline{x} = b_i & \Longrightarrow & \overline{u}_i > 0 \\ \text{bzw. ((ii)} & \overline{u}_i = 0 & \Longrightarrow & A_i.\overline{x} < b_i) \end{array}$$

Setzen wir

$$\overline{A} := \begin{pmatrix} -c^T & b^T \\ A & 0 \\ 0 & A^T \\ 0 & -A^T \\ 0 & -I \end{pmatrix}, \quad \overline{a} := \begin{pmatrix} 0 \\ b \\ c \\ -c \\ 0 \end{pmatrix}, \quad \overline{B} := (A, -I), \quad \overline{b} := b$$

dann gilt:

$$\begin{aligned} \overline{x},\ \overline{u} \text{ optimal} \quad &\Longleftrightarrow \quad \overline{A}\tbinom{x}{u} \ \le\ \overline{a}, \\ \overline{x},\ \overline{u} \text{ erfüllen (i) und (ii)} \quad &\Longleftrightarrow \quad \overline{B}\tbinom{x}{u} \ <\ \overline{b}. \end{aligned}$$

Zum Beweis der Behauptung genügt es also zu zeigen, dass

$$\overline{A}\begin{pmatrix} \overline{x} \\ \overline{u} \end{pmatrix} \le \overline{a}, \quad \overline{B}\begin{pmatrix} \overline{x} \\ \overline{u} \end{pmatrix} < \overline{b}$$

konsistent ist. Mit Satz (5.14) hat nach Voraussetzung $\overline{A}\binom{x}{u} \le \overline{a}$ eine Lösung, also ist nach Folgerung (4.5) die Konsistenz dieses Systems äquivalent zur Inkonsistenz von

$$p \ge 0,\ 0 \ne q \ge 0,\ p^T\overline{A} + q^T\overline{B} = 0^T,\ p^T\overline{a} + q^T\overline{b} \le 0.$$

Angenommen, dieses System besitzt eine Lösung, sagen wir $p^T = (\lambda, p_1^T, p_2^T, p_3^T, p_4^T) \ge 0$, $0 \ne q \ge 0$, dann gilt:

$$\begin{array}{lcr} (p_1+q)^T A & = & \lambda c^T \\ (p_2-p_3)^T A^T - (p_4+q)^T & = & -\lambda b^T \\ (p_1+q)^T b + (p_2-p_3)^T c & \le & 0. \end{array}$$

Daraus folgt

$$\begin{aligned} \lambda\ \ & ((p_1+q)^T b + (p_2-p_3)^T c) \\ & = (p_1+q)^T A(p_3-p_2) + (p_1+q)^T(p_4+q) + (p_2-p_3)^T A^T(p_1+q) \\ & = p_1^T p_4 + p_1^T q + q^T p_4 + q^T q \\ & \ge q^T q \\ & > 0. \end{aligned}$$

Daraus folgt $(p_1+q)^T b + (p_2-p_3)^T c > 0$, ein Widerspruch. □

(5.17) Hausaufgabe. Formulieren und beweisen Sie den Satz vom starken komplementären Schlupf für das Paar dualer linearer Programme (5.12), (5.13).

□

Es gibt eine natürliche Merkregel für die verschiedenen Sätze vom komplementären Schlupf. Wir wissen bereits, dass zu jeder primalen Variablen eine duale Restriktion (auch komplementäre Restriktion genannt) und zu jeder primalen Restriktion eine duale (komplementäre) Variable gehören. Die Sätze (5.15) und (5.16) zeigen nun, dass zur Charakterisierung von Optimallösungen vorzeichenbeschränkte Variable und Ungleichungen besonders wichtig sind. Zu jeder vorzeichenbeschränkten primalen (dualen) Variablen gehört eine duale (primale) Ungleichung und umgekehrt. Ist $a^T x \leq \alpha$ eine Ungleichung und $\overline{x}$ ein Vektor, so nennen wir die Ungleichung **straff** (bezüglich $\overline{x}$), falls $a^T \overline{x} = \alpha$ gilt, andernfalls nennen wir sie **locker**. Der Satz vom schwachen komplementären Schlupf (5.15) sagt dann aus: Gewisse Vektoren sind optimal für (5.12) bzw. (5.13) genau dann, wenn für jede lockere Nichtnegativitätsbedingung die komplementäre Ungleichung straff ist, d. h. wenn in der komplementären Ungleichung kein Schlupf auftritt. Satz (5.16) kann wie folgt formuliert werden. Es gibt Optimallösungen, bei denen Schlüpfe komplementär auftreten, bzw. bei denen die duale Nichtnegativitätsbedingung genau dann locker ist, wenn die komplementäre primale Ungleichung straff ist.

Kapitel 6

Grundlagen der Polyedertheorie

Wir wollen nun die Polyedertheorie weiter ausbauen, weitere Darstellungen von Polyedern angeben und neue Operationen mit Polyedern einführen. Wir werden dabei meistens von der geometrischen Anschauung ausgehen und die Begriffe von dieser Sicht aus motivieren. Wir werden jedoch unser besonderes Augenmerk auf die analytische Beschreibung der geometrischen Konzepte legen.

6.1 Transformationen von Polyedern

Wir haben bereits in Kapitel 3 Projektionen von Polyedern entlang eines Richtungsvektors c untersucht und in Folgerung (3.9) festgestellt, dass eine derartige Projektion auf ein Polyeder wieder ein Polyeder ist. Wenn man mehrfach hintereinander projiziert, bleibt diese Eigenschaft natürlich erhalten. Sind $A \in \mathbb{K}^{(m,n)}$, $b \in \mathbb{K}^m$ und $k, r \geq 0$ mit $k + r = n$, so nennt man die Menge

$$Q := \left\{ x \in \mathbb{K}^k \mid \exists\, y \in \mathbb{K}^r \text{ mit } \begin{pmatrix} x \\ y \end{pmatrix} \in P(\tilde{A}, b) \right\}$$

eine **Projektion von** $P(A, b)$ **auf** $\mathbb{K}^k$. Hierbei ist $\tilde{A} \in \mathbb{K}^{(m,n)}$ eine Matrix, die durch Spaltenvertauschung aus A hervorgeht. Offensichtlich folgt aus unseren Resultaten aus Kapitel 3:

(6.1) Bemerkung. *Jede Projektion eines Polyeders $P(A, b) \subseteq \mathbb{K}^n$ auf $\mathbb{K}^k$, $k \leq n$, ist ein Polyeder.*

□

Dieser Sachverhalt ist in größerer Allgemeinheit gültig. Erinnern wir uns daran, dass jede **affine Abbildung** $f : \mathbb{K}^n \to \mathbb{K}^k$ gegeben ist durch eine Matrix $D \in \mathbb{K}^{(k,n)}$ und einen Vektor $d \in \mathbb{K}^k$, so dass

$$f(x) = Dx + d \quad \forall\, x \in \mathbb{K}^n.$$

Für derartige Abbildungen gilt:

(6.2) Satz. *Affine Bilder von Polyedern sind Polyeder.*

Beweis : Seien $P = P(A, b) \subseteq \mathbb{K}^n$ ein Polyeder und $f(x) = Dx + d$ eine affine Abbildung von $\mathbb{K}^n$ in den $\mathbb{K}^k$, dann gilt

$$\begin{aligned} f(P) &= \{y \in \mathbb{K}^k \mid \exists\, x \in \mathbb{K}^n \text{ mit } Ax \le b \text{ und } y = Dx + d\} \\ &= \{y \in \mathbb{K}^k \mid \exists\, x \in \mathbb{K}^n \text{ mit } B\tbinom{x}{y} \le \overline{b}\}, \end{aligned}$$

wobei

$$B := \begin{pmatrix} A & 0 \\ D & -I \\ -D & I \end{pmatrix}, \quad \overline{b} := \begin{pmatrix} b \\ -d \\ d \end{pmatrix}.$$

Wenden wir nun unser Verfahren (3.6) iterativ auf $B, \overline{b}$ und die Richtungsvektoren $e_1, e_2, \dots, e_n$ an, so erhalten wir nach Satz (3.8) ein System $C\binom{x}{y} \le c$ mit $C = (0, \overline{C})$, und es gilt:

$$\begin{aligned} \forall\, y \in \mathbb{K}^k : (\exists\, x \in \mathbb{K}^n \text{ mit } B\tbinom{x}{y} \le \overline{b} &\iff \exists\, x \in \mathbb{K}^n \text{ mit } C\tbinom{x}{y} \le c \\ &\iff \overline{C}y \le c). \end{aligned}$$

Daraus folgt $f(P) = \{y \in \mathbb{K}^k \mid \overline{C}y \le c\}$ ist ein Polyeder. □

Man beachte, dass der Beweis von (6.2) sogar ein Verfahren zur expliziten Konstruktion des affinen Bildes von $P(A, b)$ beinhaltet. Aus (6.2) ergibt sich direkt die folgende (auch aus anderen Gründen unmittelbar einsichtige) Beobachtung.

(6.3) Folgerung *(Satz von Weyl). Für jede Matrix $A \in \mathbb{K}^{(m,n)}$ gilt:*

$$\left.\begin{array}{l} \operatorname{lin}(A) \\ \operatorname{aff}(A) \\ \operatorname{conv}(A) \\ \operatorname{cone}(A) \end{array}\right\} \quad \textit{ist ein Polyeder.}$$

Beweis : Wir zeigen die Behauptung für die konische Hülle. Alle anderen Fälle beweist man analog.

$$\begin{aligned} \operatorname{cone}(A) &= \{x \in \mathbb{K}^m \mid \exists\, y \geq 0 \text{ mit } x = Ay\} \\ &= \{x \in \mathbb{K}^m \mid \exists\, y \in \mathbb{K}^n \text{ mit } \begin{pmatrix} I & -A \\ -I & A \\ 0 & -I \end{pmatrix} \binom{x}{y} \leq 0\}. \end{aligned}$$

Die letzte Menge ist die Projektion eines Polyeders im $\mathbb{K}^{m+n}$ auf den $\mathbb{K}^m$, also nach (6.1) bzw. (6.2) ein Polyeder. □

Offenbar besagt die obige Folgerung nichts anderes als: Die lineare, affine, konvexe oder konische Hülle einer endlichen Teilmenge des $\mathbb{K}^n$ ist ein Polyeder. Für die konische Hülle hat dies WEYL (1935) gezeigt (daher der Name für Folgerung (6.3)).

(6.4) Folgerung. *Die Summe $P = P_1 + P_2$ zweier Polyeder P_1, P_2 ist ein Polyeder.*

Beweis : Es seien $P_1 = P(A, a)$, $P_2 = P(B, b)$, dann gilt:

$$\begin{aligned} P = P_1 + P_2 &= \{x + y \in \mathbb{K}^n \mid Ax \leq a, By \leq b\} \\ &= \{z \in \mathbb{K}^n \mid \exists\, x, y \in \mathbb{K}^n \text{ mit } Ax \leq a, By \leq b, z = x + y\} \\ &= \{z \in \mathbb{K}^n \mid \exists\, x, y \in \mathbb{K}^n \text{ mit } A(z - y) \leq a, B(z - x) \leq b\} \\ &= \{z \in \mathbb{K}^n \mid \exists\, x, y \in \mathbb{K}^n \text{ mit } D \begin{pmatrix} x \\ y \\ z \end{pmatrix} \leq \binom{a}{b}\} \end{aligned}$$

mit

$$D = \begin{pmatrix} 0 & -A & A \\ -B & 0 & B \end{pmatrix}.$$

Also ist P die Projektion eines Polyeders des $\mathbb{K}^{3n}$ auf den $\mathbb{K}^n$, und somit nach (6.1) ein Polyeder. □

Verbinden wir nun die Erkenntnis aus (6.3), dass $\operatorname{conv}(A)$ und $\operatorname{cone}(B)$ Polyeder sind, mit (6.4), so erhalten wir:

(6.5) Folgerung. *Es seien $A \in \mathbb{K}^{(m,n)}$, $B \in \mathbb{K}^{(m,n')}$, dann gilt*

$$P = \operatorname{conv}(A) + \operatorname{cone}(B)$$

ist ein Polyeder. □

Die obige Folgerung erscheint (durch geschickte Vorbereitung) völlig trivial, sie ist jedoch eine durchaus beachtenswerte Erkenntnis, denn wir werden bald zeigen, dass in der Tat alle Polyeder von der Form $\operatorname{conv}(A) + \operatorname{cone}(B)$ sind.

6.2 Kegelpolarität

Es gibt mehrere Möglichkeiten die Umkehrung von (6.5) zu beweisen. Eine besondere elegante, die eine geometrische Version des Farkas-Lemmas benutzt, führt über die Kegelpolarität. Diese Operation mag zunächst nur als technisches Hilfsmittel erscheinen. Sie und ihre Verallgemeinerungen (allgemeine Polaritäten, Blocker, Antiblocker) sind jedoch bedeutende Methoden in der Polyedertheorie und der linearen sowie ganzzahligen Optimierung.

Wir beginnen mit einer Neuinterpretation des Farkas-Lemmas (4.2) (c). Dieses besagt

$$\exists\, x \geq 0, Ax = b \Longleftrightarrow \forall\, u(A^T u \geq 0 \Rightarrow u^T b \geq 0).$$

Durch diese Aussage sind offenbar auch alle rechten Seiten b charakterisiert, für die $x \geq 0$, $Ax = b$ eine Lösung hat. Nach Definition gilt $\text{cone}(A) = \{b \in \mathbb{K}^m \mid \exists\, x \geq 0 \text{ mit } Ax = b\}$, also können wir aus (4.2) (c) folgern:

(6.6) Bemerkung. *Für alle Matrizen $A \in \mathbb{K}^{(m,n)}$ gilt:*

$$\text{cone}(A) = \{b \in \mathbb{K}^m \mid u^T b \leq 0 \;\forall\, u \in P(A^T, 0)\}.$$

□

Bemerkung (6.6) kann man geometrisch wie folgt beschreiben. Die Menge der zulässigen rechten Seiten b von $Ax = b$, $x \geq 0$ ist genau die Menge aller Vektoren $b \in \mathbb{K}^m$, welche einen stumpfen Winkel mit allen Vektoren des Kegels $P(A^T, 0)$ bilden. Allgemeiner definieren wir nun für jede beliebige Menge $S \subseteq \mathbb{K}^n$

$$S^\circ := \{y \in \mathbb{K}^n \mid y^T x \leq 0 \;\forall\, x \in S\}.$$

S° ist die Menge aller Vektoren, die einen stumpfen Winkel mit allen Vektoren aus S bilden. S° heißt **polarer Kegel** von S. (Überzeugen Sie sich, dass S° ein Kegel ist!) Wir erinnern hier an das in der linearen Algebra definierte **orthogonale Komplement**

$$S^\perp := \{y \in \mathbb{K}^n \mid y^T x = 0 \;\forall\, x \in S\}.$$

Offensichtlich gilt $S^\perp \subseteq S^\circ$. Unter Benutzung der obigen Definition können wir Bemerkung (6.6) nun auch wie folgt aufschreiben.

(6.7) Folgerung. *Für alle Matrizen $A \in \mathbb{K}^{(m,n)}$ gilt*

$$P(A, 0)^\circ = \text{cone}(A^T).$$

□

Folgerung (6.7) und die vorher gemachten Beobachtungen wollen wir an einem Beispiel erläutern. Es sei

$$A = \begin{pmatrix} -3 & 1 \\ 1 & -2 \end{pmatrix},$$

dann sind die Kegel $P(A, 0)$ und $P(A, 0)^\circ$ in Abbildung 6.1 gezeichnet.

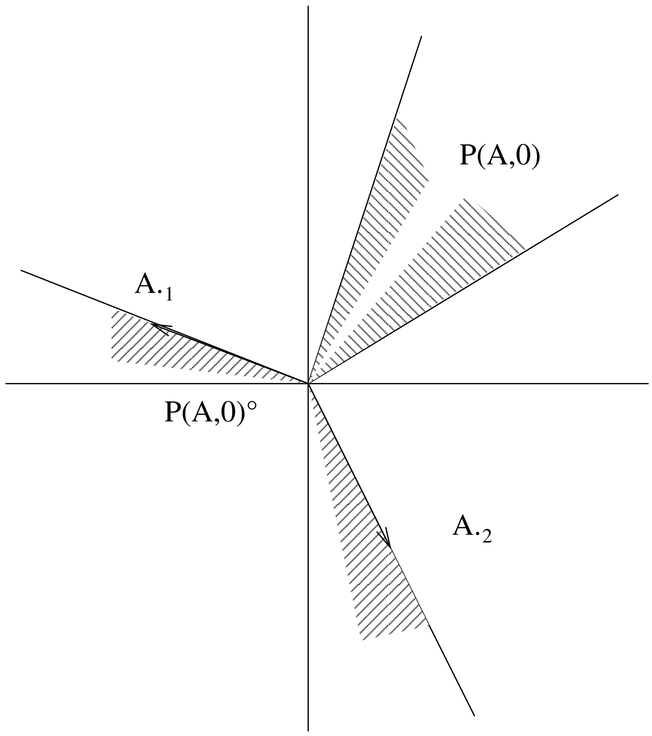


Abb. 6.1

$P(A, 0)^\circ$ besteht also aus allen Vektoren, die mit den Elementen des Kegels $P(A, 0)$ einen stumpfen Winkel bilden, und das sind gerade diejenigen Vektoren, die als konische Kombination der Normalenvektoren $A_{i.}$ dargestellt werden können, also $P(A, 0)^\circ = \text{cone}(\binom{-3}{1}, \binom{1}{-2})$. Ferner gilt: $Ax = b$, $x \geq 0$ ist genau dann lösbar, wenn $b \in P(A, 0)^\circ$ gilt. Daraus folgt z. B., dass $Ax = \binom{0}{1}$, $x \geq 0$ nicht lösbar ist, während $Ax = \binom{-1}{0}$, $x \geq 0$ eine Lösung hat.

Aus der Definition des polaren Kegels und des orthogonalen Komplements ergeben sich unmittelbar einige triviale Beziehungen, deren Beweis wir dem Leser zur Übung überlassen. Wir schreiben im Weiteren

$$S^{\circ\circ} := (S^\circ)^\circ.$$

(6.8) Hausaufgabe. Für $S, S_i \subseteq \mathbb{K}^n$, $i = 1, \dots, k$ gilt:

(a) $S_i \subseteq S_j \Longrightarrow S_j^\circ \subseteq S_i^\circ$

(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) $S = \text{lin}(S) \Longrightarrow S^\circ = S^\perp$. Gilt die Umkehrung?

(f) Ersetzen wir in (a),. . .,(d) "$\circ$" durch "$\perp$", sind dann auch noch alle Behauptungen wahr? □

(6.9) Hausaufgabe. Für welche Mengen $S \subseteq \mathbb{K}^n$ gilt

(a) $S^\circ = S^{\circ\circ\circ}$,

(b) $S = S^\circ$?

□

Die Aussage (6.8) (d) impliziert insbesondere:

(6.10) Folgerung. $\text{cone}(A^T)^\circ = P(A, 0)$.

Beweis : $(\text{cone}(A^T))^\circ = \text{cone}((A^T)^\circ) = (A^T)^\circ = \{x \mid Ax \leq 0\} = P(A, 0)$. □

Das folgende Korollar aus (6.7) und (6.10) wird in der Literatur häufig mit einem Namen belegt.

(6.11) Polarensatz. *Für jede Matrix $A \in \mathbb{K}^{(m,n)}$ gilt:*

$$\begin{aligned} P(A,0)^{\circ\circ} &= P(A,0), \\ \text{cone}(A)^{\circ\circ} &= \text{cone}(A). \end{aligned}$$

Beweis :

$$\begin{aligned} P(A,0) &\overset{(6.10)}{=} \text{cone}(A^T)^\circ &&\overset{(6.7)}{=} P(A,0)^{\circ\circ}, \\ \text{cone}(A) &\overset{(6.7)}{=} P(A^T,0)^\circ &&\overset{(6.10)}{=} \text{cone}(A)^{\circ\circ}. \end{aligned}$$

□

Unser kurzer Exkurs über Kegelpolarität ist damit beendet.

6.3 Darstellungssätze

Wir wollen nun zeigen, dass Polyeder nicht nur in der Form $P(A, b)$ dargestellt werden können und benutzen dazu die bisher entwickelte Maschinerie.

(6.12) Satz *(Minkowski (1896)). Eine Teilmenge $K \subseteq \mathbb{K}^n$ ist genau dann ein polyedrischer Kegel, wenn K die konische Hülle von endlich vielen Vektoren ist. Mit anderen Worten: Zu jeder Matrix $A \in \mathbb{K}^{(m,n)}$ gibt es eine Matrix $B \in \mathbb{K}^{(n,k)}$, so dass*

$$P(A, 0) = \operatorname{cone}(B)$$

gilt und umgekehrt.

Beweis : $P(A, 0) \overset{(6.10)}{=} \operatorname{cone}(A^T)^\circ \overset{(6.3)}{=} P(B^T, 0)^\circ \overset{(6.7)}{=} \operatorname{cone}(B)$. $\square$

(6.13) Satz. *Es seien $A \in \mathbb{K}^{(m,n)}$, $b \in \mathbb{K}^m$, dann existieren endliche Mengen $V, E \subseteq \mathbb{K}^n$ mit*

$$P(A, b) = \operatorname{conv}(V) + \operatorname{cone}(E).$$

Beweis : Setze

$$H := P\left(\begin{pmatrix} A & -b \\ 0^T & -1 \end{pmatrix}, \begin{pmatrix} 0 \\ 0 \end{pmatrix}\right),$$

dann gilt: $x \in P(A, b) \iff \binom{x}{1} \in H$. H ist nach Definition ein polyedrischer Kegel. Also gibt es nach Satz (6.12) eine Matrix $B \in \mathbb{K}^{(n+1,k)}$ mit $H = \operatorname{cone}(B)$. Aufgrund der Definition von H hat die letzte Zeile von B nur nichtnegative Elemente. Durch Skalieren der Spalten von B und Vertauschen von Spalten können wir B in eine Matrix $\overline{B}$ so umformen, dass gilt

$$\overline{B} = \begin{pmatrix} V & E \\ \mathbb{1}^T & 0^T \end{pmatrix}, \quad \operatorname{cone}(\overline{B}) = H.$$

Daraus folgt:

$$\begin{aligned} x \in P(A, b) &\iff \tbinom{x}{1} \in H \\ &\iff x = V\lambda + E\mu \text{ mit } \lambda^T\mathbb{1} = 1, \lambda, \mu \geq 0 \\ &\iff x \in \operatorname{conv}(V) + \operatorname{cone}(E). \end{aligned}$$

$\square$

(6.14) Folgerung. *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 (6.3) ein Polyeder. Ist $x \in P$, so gilt $x = \sum_{i=1}^k \lambda_i v_i$, $v_i \in V$, $\lambda_i \geq 0$, $\sum_{i=1}^k \lambda_i = 1$, und somit $\|x\| \leq \sum_{i=1}^k \|v_i\|$, d. h. $P \subseteq \{x \mid \|x\| \leq \sum_{v \in V} \|v\|\}$. Also ist P beschränkt, d. h. P ist ein Polytop.

Ist umgekehrt P ein Polytop, so gibt es nach Satz (6.13) endliche Mengen V, E mit $P = \text{conv}(V) + \text{cone}(E)$. Gibt es einen Vektor $e \in E$ mit $e \neq 0$, so gilt für alle $n \in \mathbb{N}$, $x + ne \in P$ für alle $x \in \text{conv}(V)$. Also ist P unbeschränkt, falls $E \setminus \{0\} \neq \emptyset$. Daraus folgt $E \in \{\emptyset, \{0\}\}$, und dann gilt trivialerweise $\text{conv}(V) = \text{conv}(V) + \text{cone}(E) = P$. □

(6.15) 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 (6.12), (6.13), (6.14) und (6.5). □

Ist $P \subseteq \mathbb{K}^n$ ein Polyeder, so wissen wir nunmehr, dass es für P zwei mögliche Darstellungen gibt. Es gilt nämlich

$$P = P(A, b) = \text{conv}(V) + \text{cone}(E),$$

wobei A eine (m, n)-Matrix, $b \in \mathbb{K}^m$ und V, E endliche Mengen sind. Diese beiden Darstellungen sind grundsätzlich verschieden, was natürlich in vierlei Hinsicht nützlich sein kann. Manche Aussagen über Polyeder sind völlig trivial, wenn man von der einen Beschreibung ausgeht, während sie aus der anderen Beschreibung nicht unmittelbar folgen.

Die Darstellung $P(A, b)$ nennt man auch **äußere Beschreibung** von P. Der Grund für diese Bezeichnung liegt darin, dass man wegen

$$P = \bigcap_{i=1}^m \{x \mid A_{i\cdot}x \leq b_i\} \subseteq \{x \mid A_{i\cdot}x \leq b_i\},$$

das Polyeder P als Durchschnitt von größeren Mengen betrachten kann. P wird sozusagen "von außen" durch sukzessives Hinzufügen von Ungleichungen (bzw. Halbräumen) konstruiert.

Hingegen nennt man $\text{conv}(V) + \text{cone}(E)$ eine **innere Beschreibung** von P. Ist $E = \emptyset$, so ist die Bezeichnung offensichtlich, denn $V \subseteq P$ und somit wird P

durch konvexe Hüllenbildung von Elementen von sich selbst erzeugt. Analoges gilt, wenn P ein polyedrischer Kegel ist. Sind jedoch V und E nicht leer, dann ist E nicht notwendigerweise eine Teilmenge von P, jedoch gelingt es eben aus den Vektoren $v \in V$ zusammen mit den Vektoren $e \in E$ das Polyeder P "von innen her" zu konstruieren.

Die Sätze (6.12), (6.14) und (6.15) beinhalten weitere wichtige Charakterisierungen von polyedrischen Kegeln, Polytopen und Polyedern. Wir erinnern uns aus der linearen Algebra daran, dass jeder lineare Teilraum L des $\mathbb{K}^n$ eine endliche Basis hat, d. h. eine endliche Teilmenge B besitzt, so dass B linear unabhängig ist und $L = \text{lin}(B)$ gilt. Die linearen Teilräume des $\mathbb{K}^n$ sind also diejenigen Teilmengen des $\mathbb{K}^n$, deren Elemente durch Linearkombinationen einer endlichen Menge erzeugt werden können. Nach (6.14) sind Polytope genau diejenigen Teilmengen des $\mathbb{K}^n$, die durch Konvexkombinationen einer endlichen Menge erzeugt werden können.

Wir werden in Kapitel 8 sehen, dass es sogar eine eindeutig bestimmte minimale (im Sinne der Mengeninklusion) endliche Menge $V \subseteq \mathbb{K}^n$ gibt mit $P = \text{conv}(V)$, d. h. Polytope haben sogar eine eindeutig bestimmte "konvexe Basis". Nach (6.12) sind polyedrische Kegel genau diejenigen Teilmengen des $\mathbb{K}^n$, die ein endliches "Kegelerzeugendensystem" haben. Auch hier gibt es natürlich minimale endliche Mengen, die die Kegel konisch erzeugen. Aber nur unter zusätzlichen Voraussetzungen haben zwei minimale konische Erzeugendensysteme auch gleiche Kardinalität, und Eindeutigkeit gilt lediglich bis auf Multiplikation mit positiven Skalaren. Häufig nennt man eine Teilmenge T des $\mathbb{K}^n$ **endlich erzeugt**, falls $T = \text{conv}(V) + \text{cone}(E)$ für endliche Mengen V, E gilt. Nach (6.15) sind also die Polyeder gerade die endlich erzeugten Teilmengen des $\mathbb{K}^n$. Fassen wir zusammen, so gilt:

(6.16) Bemerkung. Ist $T \subseteq \mathbb{K}^n$, so gilt

(a) T ist ein linearer Teilraum $\Longleftrightarrow$ T ist die lineare Hülle einer endlichen Menge.

(b) T ist ein affiner Teilraum $\Longleftrightarrow$ T ist die affine Hülle einer endlichen Menge.

(c) T ist ein polyedrischer Kegel $\Longleftrightarrow$ T ist die konische Hülle einer endlichen Menge.

(d) T ist ein Polytop $\Longleftrightarrow$ T ist die konvexe Hülle einer endlichen Menge.

(e) T ist ein Polyeder $\Longleftrightarrow$ T ist endlich erzeugt.

$\square$

Kapitel 7

Seitenflächen von Polyedern

Wir wollen uns nun mit den “Begrenzungsflächen” von Polyedern P beschäftigen und diese charakterisieren. Wir gehen dabei so vor, dass wir die Objekte, die wir untersuchen wollen, zunächst “abstrakt”, d. h. durch eine allgemeine Definition einführen, und dann werden wir diese Objekte **darstellungsabhängig** kennzeichnen. Das heißt, wir werden jeweils zeigen, wie die betrachteten Objekte “aussehen”, wenn P durch $P(A, b)$ oder durch $\text{conv}(V)+\text{cone}(E)$ gegeben ist. Wir werden uns meistens auf die Darstellung $P(A, b)$ beziehen, da diese für die Optimierung die wichtigere ist. In Abschnitt 7.1 werden wir jedoch zeigen, wie man Charakterisierungen bezüglich einer Darstellung auf Charakterisierungen bezüglich der anderen Darstellung transformieren kann, so dass aus den hier vorgestellten Resultaten die Kennzeichnungen bezüglich der Darstellung $\text{conv}(V) + \text{cone}(E)$ (mit etwas technischem Aufwand) folgen.

Wir wollen nochmals auf eine die Schreibtechnik vereinfachende Konvention hinweisen. Wir betrachten Matrizen und endliche Mengen als (im Wesentlichen) ein und dieselben Objekte. Ist z. B. A eine (m, n)-Matrix, so schreiben wir

$$\text{conv}(A)$$

und meinen damit die konvexe Hülle der Spaltenvektoren von A. Ist umgekehrt z. B. $V \subseteq \mathbb{K}^n$ eine endliche Menge, so fassen wir V auch als eine $(n, |V|)$-Matrix auf und schreiben

$$V^T,$$

um die Matrix zu bezeichnen, deren Zeilen die Vektoren aus V sind. V^T ist natürlich nicht eindeutig definiert, da wir keine Reihenfolge der Vektoren aus V angegeben haben. Wir werden diese Schreibweise jedoch nur dann benutzen, wenn es auf die Reihenfolge der Zeilen nicht ankommt.

7.1 Die γ-Polare und gültige Ungleichungen

In Abschnitt 6 haben wir bereits einen Polarentyp, die Kegelpolare, zur Beweisvereinfachung eingeführt. Hier wollen wir eine weitere Polare betrachten, die es uns ermöglichen wird, eine Charakterisierung bezüglich $P(A, b)$ in eine Charakterisierung bezüglich $\operatorname{conv}(V) + \operatorname{cone}(E)$ zu übertragen.

(7.1) Definition. *Es seien $S \subseteq \mathbb{K}^n$, $a \in \mathbb{K}^n$, $\alpha \in \mathbb{K}$.*

(a) Die Ungleichung $a^T x \leq \alpha$ heißt **gültig** *bezüglich S, falls $S \subseteq \{x \in \mathbb{K}^n \mid a^T x \leq \alpha\}$.*

(b) Die Menge $S^\gamma := \{\binom{a}{\alpha} \in \mathbb{K}^{n+1} \mid a^T x \leq \alpha \; \forall\, x \in S\}$ heißt **γ-Polare** *von S.*

(c) Eine Hyperebene $H = \{x \mid a^T x = \alpha\}$ heißt **Stützhyperebene** *von S, falls $\binom{a}{\alpha} \in S^\gamma$ und $S \cap H \neq \emptyset$.* □

Zu sagen, dass $a^T x \leq \alpha$ gültig bezüglich S ist, heißt nichts anderes als: S ist im Halbraum $a^T x \leq \alpha$ enthalten. Die γ-Polare S^γ kann als die „Menge aller gültigen Ungleichungen bezüglich S“ betrachtet werden. Wir wollen nun die γ-Polare eines Polyeders charakterisieren.

(7.2) Satz. *Es sei $P \subseteq \mathbb{K}^n$, $P \neq \emptyset$, ein Polyeder mit den Darstellungen $P = P(A, b) = \operatorname{conv}(V) + \operatorname{cone}(E)$, dann gilt:*

$$(a) \quad P^\gamma = \left\{ \begin{pmatrix} a \\ \alpha \end{pmatrix} \in \mathbb{K}^{n+1} \mid \exists\, u \geq 0, u^T A = a^T, u^T b \leq \alpha \right\}$$

$$= \operatorname{cone} \begin{pmatrix} A^T & 0 \\ b^T & 1 \end{pmatrix}.$$

$$(b) \quad P^\gamma = \left\{ \begin{pmatrix} a \\ \alpha \end{pmatrix} \in \mathbb{K}^{n+1} \mid \begin{pmatrix} V^T & -\mathbb{1} \\ E^T & 0 \end{pmatrix} = \begin{pmatrix} a \\ \alpha \end{pmatrix} \leq 0 \right\}$$

$$= P\left(\begin{pmatrix} V^T & -\mathbb{1} \\ E^T & 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 0 \end{pmatrix} \right).$$

Beweis :

$$(a)\quad \begin{pmatrix} a \\ \alpha \end{pmatrix} \in P^\gamma \iff Ax \le b, a^T x > \alpha \quad \textit{inkonsistent}$$

$$\overset{(4.5)}{\iff} \exists\, u \ge 0, v > 0 \textit{ mit } u^T A - v a^T = 0, u^T b - v\alpha \le 0$$

$$\iff \exists\, u \ge 0 \textit{ mit } u^T A = a^T, u^T b \le \alpha$$

$$\iff \exists\, u \ge 0, \lambda \ge 0 \textit{ mit } \begin{pmatrix} a \\ \alpha \end{pmatrix} = \begin{pmatrix} A^T & 0 \\ b^T & 1 \end{pmatrix} \begin{pmatrix} u \\ \lambda \end{pmatrix}$$

$$\iff \begin{pmatrix} a \\ \alpha \end{pmatrix} \in \operatorname{cone} \begin{pmatrix} A^T & 0 \\ b^T & 1 \end{pmatrix}.$$

$$(b)\quad \begin{pmatrix} a \\ \alpha \end{pmatrix} \in P^\gamma \implies a^T v \le \alpha \quad \forall\, v \in V \quad \textit{und}$$

$$a^T(v + \lambda e) \le \alpha \quad \forall\, v \in V, e \in E, \lambda \ge 0$$

$$\implies a^T e \le 0$$

(*andernfalls wäre* $a^T(v + \lambda e) > \alpha$ *für genügend großes* λ)

$$\implies \begin{pmatrix} V^T & -\mathbb{1} \\ E^T & 0 \end{pmatrix} \begin{pmatrix} a \\ \alpha \end{pmatrix} \le 0.$$

Gilt umgekehrt das letztere Ungleichungssystem, und ist $x \in P$, so existieren $v_1, \dots, v_p \in V$ und $e_1, \dots, e_q \in E$, $\lambda_1, \dots, \lambda_p \ge 0$, $\sum_{i=1}^p \lambda_i = 1$, $\mu_1, \dots, \mu_q \ge 0$, so dass

$$x = \sum_{i=1}^p \lambda_i v_i + \sum_{j=1}^q \mu_j e_j.$$

Und daraus folgt

$$a^T x = \sum_{i=1}^p \lambda_i a^T v_i + \sum_{j=1}^q \mu_j a^T e_j \le \sum_{i=1}^q \lambda_i \alpha + \sum_{j=1}^q \mu_j 0 = \alpha,$$

also gilt $\begin{pmatrix} a \\ \alpha \end{pmatrix} \in P^\gamma$. □

(7.3) Folgerung. *Die γ-Polare eines Polyeders $\emptyset \ne P \subseteq \mathbb{K}^n$ ist ein polyedrischer Kegel im $\mathbb{K}^{n+1}$.*

□

(7.4) Folgerung. *Ist* $\emptyset \neq P = P(A, b) = \text{conv}(V) + \text{cone}(E)$ *ein Polyeder und* $a^T x \leq \alpha$ *eine Ungleichung, dann sind äquivalent:*

(i) $a^T x \leq \alpha$ *ist gültig bezüglich* 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) $\binom{a}{\alpha} \in P^\gamma$

□

7.2 Seitenflächen

Wir werden nun diejenigen Teilmengen von Polyedern untersuchen, die als Durchschnitte des Polyeders mit Stützhyperebenen entstehen.

(7.5) Definition. $P \subseteq \mathbb{K}^n$ *sei ein Polyeder. Eine Menge* $F \subseteq P$ *heißt* **Seitenfläche** *von* P, *wenn es eine gültige Ungleichung* $c^T x \leq \gamma$ *bezüglich* P *gibt mit*

$$F = P \cap \{x \mid c^T x = \gamma\}.$$

Eine Seitenfläche F *von* P *heißt* **echt**, *wenn* $F \neq P$ *gilt.* F *heißt* **nichttrivial**, *wenn* $\emptyset \neq F \neq P$ *gilt. Ist* $c^T x \leq \gamma$ *gültig bezüglich* P, *dann heißt*

$$F := P \cap \{x \mid c^T x = \gamma\}$$

die von $c^T x \leq \gamma$ **induzierte** *oder* **definierte** *Seitenfläche.*

□

Offenbar sind Seitenflächen von Polyedern wiederum Polyeder. Eine Seitenfläche F kann durchaus von verschiedenen Ungleichungen definiert werden. Trivialerweise gilt $F = P \cap \{x \mid c^T x = \gamma\} = P \cap \{x \mid \lambda c^T x = \lambda\gamma\}$ für alle $\lambda > 0$, aber auch zwei Ungleichungen die nicht durch Skalierung auseinander hervorgehen, können F definieren.

(7.6) Beispiel. Sei $P(A,b) \subseteq \mathbb{K}^2$ das durch

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

definierte Polyeder (siehe Abbildung 7.1), dann ist das Geradenstück zwischen $(1,1)^T$ und $(0,2)^T$ eine Seitenfläche F definiert durch $x_1+x_2 \leq 2$. Der Punkt$G = \{(1,1)^T\}$ ist ebenfalls eine Seitenfläche von P, denn

$$\begin{aligned} G &= P(A,b) \cap \{x \mid 2x_1 + x_2 = 3\} \\ &= P(A,b) \cap \{x \mid 3x_1 + x_2 = 4\}, \end{aligned}$$

und, wie man sieht, kann $\{\binom{1}{1}\}$ durch zwei völlig unterschiedliche Ungleichungen definiert werden.

□

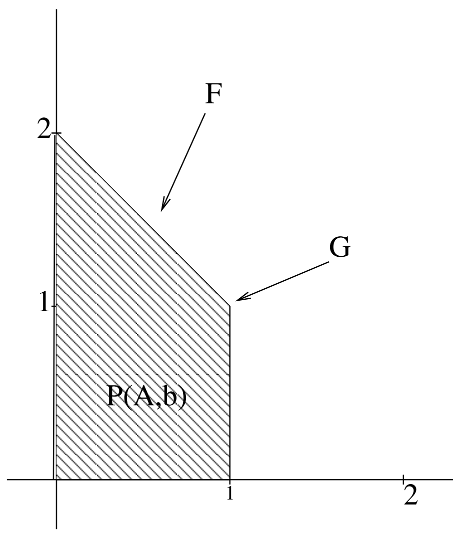


Abb. 7.1

(7.7) Folgerung. *Sei $P \subseteq \mathbb{K}^n$ ein Polyeder, dann gilt:*

(a) P ist eine Seitenfläche von sich selbst.

(b) $\emptyset$ ist eine Seitenfläche von P.

(c) Ist $F = \{x \in P \mid c^T x = \gamma\}$ eine nichttriviale Seitenfläche von P, dann gilt $c \neq 0$.

Beweis : *(a)* $P = \{x \in P \mid 0^T x = 0\}$
(b) $\emptyset = \{x \in P \mid 0^T x = 1\}$
(c) *klar.*

$\square$

(7.8) Satz. *Seien* $P = P(A, b) \neq \emptyset$ *ein Polyeder,* $c \in \mathbb{K}^n$, $\gamma \in \mathbb{K}$ *und*

$$z^* = \begin{cases} +\infty, & \text{falls } c^T x \text{ auf } P \text{ unbeschränkt,} \\ \max\{c^T x \mid x \in P\}, & \text{andernfalls.} \end{cases}$$

(a) $c^T x \leq \gamma$ *ist gültig bezüglich* $P \iff \gamma \geq z^*$.

(b) $z^* < +\infty \Rightarrow F = \{x \in P \mid c^T x = z^*\}$ *ist eine nichtleere Seitenfläche von* P*, und* $\{x \mid c^T x = z^*\}$ *ist eine Stützhyperebene von* P*, falls* $c \neq 0$.

(c) Die Menge der Optimallösungen eines linearen Programms ist eine Seitenfläche des durch die Nebenbedingungen definierten Polyeders.

Beweis : *(a) Ist* $\gamma < z^*$*, dann existiert ein* $x^* \in P$ *mit* $c^T x^* > \gamma$*, also ist* $c^T x \leq \gamma$ *nicht gültig. Ist* $\gamma \geq z^*$*, so gilt* $c^T x \leq z^* \leq \gamma \; \forall \, x \in P$*, also ist* $c^T x \leq \gamma$ *gültig.*

(b) Nach (a) ist $c^T x \leq z^*$ *gültig bezüglich* P *und nach Definition existiert ein* $x^* \in P$ *mit* $c^T x^* = z^*$*, also ist* F *eine nichtleere Seitenfläche von* P*. Offenbar ist* $\{x \mid c^T x = z^*\}$ *eine Stützhyperebene, falls* $c \neq 0$.

(c) folgt aus (b). $\square$

(7.9) Definition. *Sei* $P = P(A, b) \subseteq \mathbb{K}^n$ *ein Polyeder und* M *die Zeilenindexmenge von* A*. Für* $F \subseteq P$ *sei*

$$\mathrm{eq}(F) := \{i \in M \mid A_{i.} x = b_i \; \forall \, x \in F\},$$

d. h. $\mathrm{eq}(F)$ *(genannt* **Gleichheitsmenge** *oder* **equality set von** F*) ist die* **Menge der** *für alle* $x \in F$ **bindenden Restriktionen**. *Für* $I \subseteq M$ *sei*

$$\mathrm{fa}(I) = \{x \in P \mid A_{I.} x = b_I\}.$$

Wir nennen $\mathrm{fa}(I)$ *die* **von** I **induzierte Seitenfläche**.

$\square$

Offenbar ist $\text{fa}(I)$ tatsächlich eine Seitenfläche von $P = P(A,b)$, denn setzen wir $c^T := \sum_{i\in I} A_{i.}$, $\gamma := \sum_{i\in I} b_i$, so ist $c^T x \leq \gamma$ gültig bezüglich $P(A,b)$, und, wie man leicht sieht, gilt

$$\text{fa}(I) = \{x \in P \mid c^T x = \gamma\}.$$

Zur Veranschaulichung der oben definierten Abbildungen fa und eq betrachten wir Beispiel (7.6). Für das in (7.6) definierte Polyeder $P(A,b)$ gilt $M = \{1,2,3,4\}$ und

$$\begin{aligned} \text{fa}(\{1,2\}) &= G, \\ \text{eq}\left(\left\{\tbinom{0}{2}\right\}\right) &= \{1,3\}. \end{aligned}$$

Wir zeigen nun, dass man die Gleichheitsmenge einer Seitenfläche explizit berechnen kann.

(7.10) Satz. *Seien $P = P(A,b)$ ein Polyeder und $\emptyset \neq F = \{x \in P \mid c^T x = \gamma\}$ eine Seitenfläche von P. Dann gilt*

$$\text{eq}(F) = \{i \in M \mid \exists\, u \geq 0 \text{ mit } u_i > 0 \text{ und } u^T A = c^T, u^T b = \gamma\}.$$

Beweis : Nach Voraussetzung und Folgerung (5.9) haben die beiden linearen Programme

$$\text{(P)} \quad \begin{aligned} \max\ & c^T x \\ & Ax \leq b \end{aligned} \qquad \text{und} \qquad \text{(D)} \quad \begin{aligned} \min\ & u^T b \\ & u^T A = c^T \\ & u \geq 0 \end{aligned}$$

optimale Lösungen mit gleichem Zielfunktionswert γ, und F ist die Menge der Optimallösungen von (P). Sei nun $i \in \text{eq}(F)$. Aufgrund des Satzes (5.16) vom starken komplementären Schlupf existieren Optimallösungen $\overline{x}$, $\overline{u}$ von (P), (D) mit $\overline{u}_j > 0 \Leftrightarrow A_{j.}\overline{x} = b_j$.. Wegen $\overline{x} \in F$ gilt $A_{i.}\overline{x} = b_i$, also gibt es einen Vektor u mit den geforderten Eigenschaften.

Gibt es umgekehrt einen Vektor $u \geq 0$ mit $u_i > 0$ und $u^T A = c^T$, $u^T b = \gamma$, so ist u optimal für (D), und aus dem Satz vom schwachen komplementären Schlupf (5.15) folgt $A_{i.}x = b_i$ für alle $x \in F$, d. h. $i \in \text{eq}(F)$. □

(7.11) Satz. *Seien $P = P(A,b)$ ein Polyeder und $F \neq \emptyset$ eine Teilmenge von P, dann sind äquivalent:*

(i) *F ist eine Seitenfläche von P.*

(ii) *$\exists\, I \subseteq M$ mit $F = \text{fa}(I) = \{x \in P \mid A_{I.}x = b_I\}$.*

(iii) *$F = \text{fa}(\text{eq}(F))$.*

Beweis : Gelten (ii) oder (iii), dann ist F offenbar eine Seitenfläche von P.

(i) $\Rightarrow$ (ii) Sei $F = \{x \in P \mid c^T x = \gamma\}$ eine Seitenfläche. Setzen wir $I := \text{eq}(F)$, dann gilt nach Definition $F \subseteq \{x \in P \mid A_{I.}x = b_I\} =: F'$. Ist $x \in F'$, so gilt $x \in P$; es bleibt zu zeigen, dass $c^T x = \gamma$ gilt. Zu jedem $i \in I$ gibt es nach (7.10) einen Vektor $u^{(i)}$ mit $u^{(i)} \geq 0$, $u_i^{(i)} > 0$, $(u^{(i)})^T A^T = c^T$ und $(u^{(i)})^T b = \gamma$. Setze

$$u := \sum_{i \in I} \frac{1}{|I|} u^{(i)},$$

dann gilt nach Konstruktion $u_i > 0 \; \forall\, i \in I$ und ferner $u_i = 0 \; \forall\, i \in M \setminus I$ (andernfalls wäre $i \in I$ nach (7.10)). Die Vektoren x und u sind zulässig für die linearen Programme (P), (D) des Beweises von (5.10). Aus dem Satz vom schwachen komplementären Schlupf (5.15) folgt, dass sie auch optimal sind. Daraus folgt $c^T x = \gamma$ und somit $x \in F$.

(ii) $\Rightarrow$ (iii) folgt direkt aus dem obigen Beweis. □

Aus Satz (7.11) folgt, dass zur Darstellung einer Seitenfläche von $P(A, b)$ keine zusätzliche Ungleichung benötigt wird. Man braucht lediglich in einigen der Ungleichungen des Systems $Ax \leq b$ Gleichheit zu fordern. Da jede nichtleere Seitenfläche auf diese Weise erzeugt werden kann, folgt:

(7.12) Folgerung. *Sind $A \in \mathbb{K}^{(m,n)}$, $b \in \mathbb{K}^m$, dann hat das Polyeder $P(A, b)$ höchstens $2^m + 1$ Seitenflächen.*

Beweis : $M = \{1, \ldots, m\}$ hat 2^m Teilmengen. Für jede Teilmenge $I \subseteq M$ ist $P \cap \{x \mid A_{I.}x = b_I\}$ eine Seitenfläche von P. Dazu kommt u. U. noch die leere Seitenfläche.

□

Man kann Seitenflächen auf ähnliche Weise durch Einführung von Abbildungen analog zu eq bzw. fa bezüglich der Darstellung $P = \text{conv}(V) + \text{cone}(E)$ charakterisieren. Diese Kennzeichnungen von Seitenflächen sind jedoch technisch aufwendiger. Der interessierte Leser sei dazu auf Bachem & Grötschel, "New Aspects of Polyhedral Theory", in: B. Korte (ed.), "Modern Applied Mathematics — Optimization and Operations Research", North-Holland, Amsterdam 1982 verwiesen.

7.3 Dimension

Wir wollen nun zeigen, dass man auch die Dimension einer Seitenfläche eines Polyeders explizit berechnen kann. Zunächst führen wir einen Hilfsbegriff ein.

(7.13) Definition. *Ein Element x eines Polyeders P heißt* **innerer Punkt** *von P, wenn x in keiner echten Seitenfläche von P enthalten ist.*

□

Achtung! Innere Punkte eines Polyeders P sind nicht notwendig auch topologisch innere Punkte im Sinne der natürlichen Topologie des $\mathbb{K}^n$. Unsere inneren Punkte sind topologisch innere Punkte im Sinne der Relativtopologie auf P.

(7.14) Satz. *Jedes nichtleere Polyeder besitzt innere Punkte.*

Beweis : Sei $P = P(A, b)$ und $I = \mathrm{eq}(P(A, b))$, $J = M \setminus I$. Gilt $I = M$, so hat P keine echten Seitenflächen, also ist jedes Element von P ein innerer Punkt. Andernfalls ist das System $Ax \leq b$ äquivalent zu

$$\begin{aligned} A_{I.}x &= b_I, \\ A_{J.}x &\leq b_J. \end{aligned}$$

P hat innere Punkte heißt dann, dass es ein x gibt mit $A_{I.}x = b_I$ und $A_{J.}x < b_J$. Zu jedem $i \in J$ existiert nach Definition ein Vektor $y_i \in P$ mit $A_{i.}y_i < b_i$. Setze $y := \frac{1}{|J|} \sum_{i \in J} y_i$, dann ist y Konvexkombination von Elementen von P, also $y \in P$, und es gilt $A_{J.}y < b_J$. Mithin ist y ein innerer Punkt von P. □

(7.15) Satz. *Sei F Seitenfläche eines Polyeders $P(A, b)$ und $\overline{x} \in F$. Der Vektor $\overline{x}$ ist ein innerer Punkt von F genau dann, wenn $\mathrm{eq}(\{\overline{x}\}) = \mathrm{eq}(F)$.*

Beweis : $\overline{x}$ ist genau dann ein innerer Punkt von F, wenn die kleinste (im Sinne der Mengeninklusion) Seitenfläche von F, die $\overline{x}$ enthält, F selbst ist. Offenbar ist $\mathrm{fa}(\mathrm{eq}(\{\overline{x}\}))$ die minimale Seitenfläche von P, die $\overline{x}$ enthält. Daraus folgt die Behauptung. □

(7.16) Satz. *Ist $F \neq \emptyset$ eine Seitenfläche des Polyeders $P(A, b) \subseteq \mathbb{K}^n$, dann gilt*

$$\dim(F) = n - \mathrm{rang}(A_{\mathrm{eq}(F).}).$$

Beweis : Sei $I := \text{eq}(F)$. Aus der linearen Algebra wissen wir, dass $n = \text{rang}(A_{I.}) + \dim(\text{Kern } (A_{I.}))$ gilt. Zu zeigen ist also: $\dim(F) = \dim(\text{Kern } (A_{I.}))$. Seien $r := \dim(\text{Kern } (A_{I.}))$ und $s := \dim(F)$.

"$r \geq s$": Da $\dim(F) = s$, gibt es $s+1$ affin unabhängige Vektoren $x_0, x_1, \ldots, x_s \in F$. Dann sind die Vektoren $x_1 - x_0, \ldots, x_s - x_0$ linear unabhängig und erfüllen $A_{I.}(x_i - x_0) = 0$. Kern $(A_{I.})$ enthält also mindestens s linear unabhängige Vektoren, also gilt $r \geq s$.

"$s \geq r$": Nach (7.14) besitzt F einen inneren Punkt $\overline{x} \in F$. Nach (7.15) gilt $\text{eq}(\{\overline{x}\}) = \text{eq}(F) = I$, und daraus folgt für $J := M \setminus I$:

$$\begin{aligned} A_{I.}\overline{x} &= b_I, \\ A_{J.}\overline{x} &< b_J. \end{aligned}$$

Ist $r = 0$, so gilt $s \geq 0$ wegen $\overline{x} \in F$. Sei also $r \geq 1$, und $\{x_1, \ldots, x_r\}$ sei eine Basis von Kern $(A_{I.})$. Für $p = 1, \ldots, r$ und $j \in J$ setze:

$$\begin{aligned} \delta_{jp} &:= \begin{cases} \infty, & \text{falls} A_{j.}x_p = 0 \\ \dfrac{b_j - A_{j.}\overline{x}}{A_{j.}x_p} & \text{andernfalls} \end{cases} \\ \varepsilon &:= \min\{\delta_{jp} \mid j \in J, p \in \{1, \ldots, r\}\}. \end{aligned}$$

(Setze $\varepsilon \neq 0$ beliebig, falls $\delta_{jp} = \infty$ für alle j, p.) Für $i \in I$ und alle $p \in \{1, \ldots, r\}$ gilt nun

$$A_{i.}(\overline{x} + \varepsilon x_p) = A_{i.}\overline{x} + \varepsilon A_{i.}x_p = A_{i.}\overline{x} = b_i,$$

da $A_{i.}x_p = 0$. Für $j \in J$ gilt

$$\begin{aligned} A_{j.}(\overline{x} + \varepsilon x_p) &= A_{j.}\overline{x} + \varepsilon A_{j.}x_p \\ &\leq A_{j.}\overline{x} + \delta_{jp}A_{j.}x_p \\ &= A_{j.}\overline{x} + b_j - A_{j.}\overline{x} \\ &= b_j. \end{aligned}$$

Daraus folgt $\overline{x} + \varepsilon x_p \in F$ für alle $p \in \{1, \ldots, r\}$. Da die Vektoren $\varepsilon x_1, \ldots, \varepsilon x_r$ linear unabhängig sind, sind die Vektoren $\overline{x}, \overline{x}+\varepsilon x_1, \ldots, \overline{x}+\varepsilon \overline{x}_r$ affin unabhängig. Das heißt, F enthält mindestens $r + 1$ affin unabhängige Vektoren, und somit gilt $\dim(F) = s \geq r$. □

(7.17) Folgerung. *$P = P(A, b) \subseteq \mathbb{K}^n$ sei ein nichtleeres Polyeder, dann gilt:*

(a) $\dim(P) = n - \text{rang}(A_{\text{eq}(P).})$.

(b) Ist $\text{eq}(P) = \emptyset$*, dann ist P volldimensional (d. h.* $\dim(P) = n$*).*

(c) Ist F eine echte Seitenfläche von P, dann gilt $\dim(F) \leq \dim(P) - 1$.

□

Mit Hilfe von Satz (7.16) kann man auch die affine Hülle einer Seitenfläche auf einfache Weise bestimmen.

(7.18) Satz. *Sei $F \neq \emptyset$ eine Seitenfläche des Polyeders $P(A, b)$, dann gilt*

$$\text{aff}(F) = \{x \mid A_{\text{eq}(F).}x = b_{\text{eq}(F)}\}.$$

Beweis : Es seien $I := \text{eq}(F)$ und $T := \{x \mid A_{I.}x = b_I\}$. Offenbar ist T ein affiner Raum und wegen $F \subseteq T$ gilt $\text{aff}(F) \subseteq \text{aff}(T) = T$. Sei $s = \dim(F)$, dann folgt aus Satz (7.16), dass $\dim(\text{Kern } (A_{I.})) = s$ und somit $\dim(T) = s$ gilt. Aus $\dim(\text{aff}(F)) = \dim T$ und $\text{aff}(F) \subseteq T$ folgt $\text{aff}(F) = T$. □

7.4 Facetten und Redundanz

Wie wir bereits bemerkt haben, kann man zu einem Ungleichungssystem $Ax \leq b$ beliebig viele Ungleichungen hinzufügen, ohne die Lösungsmenge des Systems zu ändern. Wir wollen nun untersuchen, wie man ein gegebenes Polyeder mit möglichst wenigen Ungleichungen darstellen kann. Dies ist speziell für die lineare Optimierung wichtig, da der Rechenaufwand zur Auffindung einer Optimallösung in der Regel von der Anzahl der vorgelegten Ungleichungen abhängt. Gesucht wird also eine Minimaldarstellung eines Polyeders, um rechentechnische Vorteile zu haben. Es wird sich zeigen, dass hierbei diejenigen Ungleichungen, die maximale echte Seitenflächen eines Polyeders definieren, eine wesentliche Rolle spielen. Deshalb wollen wir derartige Seitenflächen untersuchen.

(7.19) Definition. *$Ax \leq b$ sei ein Ungleichungssystem, und M sei die Zeilenindexmenge von A.*

(a) *Sei $I \subseteq M$, dann heißt das System $A_{I}.x \leq b_I$* **unwesentlich** *oder* **redundant** *bezüglich $Ax \leq b$, wenn $P(A, b) = P(A_{M \setminus I}., b_{M \setminus I})$ gilt.*

(b) *Enthält $Ax \leq b$ ein unwesentliches Teilsystem $A_{I}.x \leq b_I$, dann heißt $Ax \leq b$* **redundant**, *andernfalls* **irredundant**.

(c) *Eine Ungleichung $A_{i}.x \leq b_i$ heißt* **wesentlich** *oder* **nicht redundant** *bezüglich $Ax \leq b$, wenn $P(A, b) \neq P(A_{M \setminus \{i\}}., b_{M \setminus \{i\}})$ gilt.*

(d) *Eine Ungleichung $A_{i}.x \leq b_i$ heißt* **implizite Gleichung** *bezüglich $Ax \leq b$, wenn $i \in \mathrm{eq}(P(A, b))$ gilt.*

(e) *Ein System $Ax \leq a$, $Bx = b$ heißt* **irredundant**, *wenn $Ax \leq a$ keine unwesentliche Ungleichung bezüglich des Systems $Ax \leq a$, $Bx \leq b$, $-Bx \leq -b$ enthält und B vollen Zeilenrang hat.*

(f) *Eine nichttriviale Seitenfläche F von $P(A, b)$ heißt* **Facette** *von $P(A, b)$, falls F in keiner anderen echten Seitenfläche von $P(A, b)$ enthalten ist.*

□

Wir weisen ausdrücklich darauf hin, dass Redundanz bzw. Irredundanz keine Eigenschaft des Polyeders $P(A, b)$ ist, sondern eine Eigenschaft des Ungleichungssystems $Ax \leq b$. Wir werden sehen, dass ein Polyeder viele irredundante Beschreibungen haben kann. Ferner ist auch die Annahme falsch, dass man immer durch das gleichzeitige Weglassen aller unwesentlichen Ungleichungen eines Systems $Ax \leq b$ eine irredundante Beschreibung von $P(A, b)$ enthält. Wir wollen nun zunächst unwesentliche Ungleichungen charakterisieren.

(7.20) Satz. *Ein Ungleichungssystem $A_{I}.x \leq b_I$ ist unwesentlich bezüglich $Ax \leq b$ genau dann, wenn es eine Matrix $U \in \mathbb{K}_+^{(|I|,m)}$ gibt mit $UA = A_{I}.$, $Ub \leq b_I$ und $U._I = 0$.*

Beweis : Für jedes $i \in I$ ist nach (7.4) die Ungleichung $A_{i}.x \leq b_i$ gültig bezüglich $P(A_{M \setminus I}., b_{M \setminus I})$ genau dann, wenn es einen Vektor $\overline{u}_i \geq 0$ gibt mit $\overline{u}_i^T A_{M \setminus I}. = A_{i}.$ und $\overline{u}_i^T b \leq b_i$. Dies ist genau dann der Fall, wenn es eine Matrix $U \in \mathbb{K}_+^{(|I|,m)}$ gibt mit $UA = A_{I}.$, $Ub \leq b_I$ und $U._I = 0$. □

(7.21) Folgerung. *$A_{i}.x \leq b_i$ ist genau dann redundant bezüglich $Ax \leq b$, wenn es einen Vektor $u \in \mathbb{K}_+^m$ gibt mit $u^T A = A_{i}.$, $u^T b \leq b_i$, $u_i = 0$.*

□

Der nächste Satz zeigt, wann man Ungleichungen nicht mehr weglassen kann, ohne das Polyeder zu ändern.

(7.22) Satz. *$P = P(A, b) \neq \emptyset$ sei ein Polyeder. Sei $\emptyset \neq I \subseteq M \setminus \mathrm{eq}(P)$ und $P' := P(A_{M\setminus I}., b_{M\setminus I})$. Dann gilt*

$P \neq P' \iff \exists$ nichttriviale Seitenfläche $F \subseteq P$ mit $\mathrm{eq}(F) \subseteq I \cup \mathrm{eq}(P)$.

Beweis : Im Weiteren bezeichnen wir mit $\mathrm{eq}_{P'}$ die "equality set"-Abbildung bezüglich P'. Es gilt offenbar $\mathrm{eq}_{P'}(F) \subseteq \mathrm{eq}(F)$.

"$\Longleftarrow$" Angenommen, es gilt $P = P'$, und F sei eine beliebige nichttriviale Seitenfläche von P (und somit auch von P'). Da F eine nichttriviale Seitenfläche von P' ist, gibt es ein $i \in (M \setminus I) \setminus \mathrm{eq}_{P'}(P')$ mit $i \in \mathrm{eq}_{P'}(F)$. Daraus folgt $\mathrm{eq}(F) \not\subseteq I \cup \mathrm{eq}(P)$.

"$\Longrightarrow$" Angenommen, es gilt $P \neq P'$. Wegen $P \subseteq P'$ heißt dies, es existiert ein Vektor $v \in P' \setminus P$, und somit gibt es eine Indexmenge $\emptyset \neq K \subseteq I$ mit der Eigenschaft $A_i.v \leq b_i \ \forall\, i \in M \setminus K$ und $A_i.v > b_i \ \forall\, i \in K$. Nach Satz (7.14) hat P einen inneren Punkt, sagen wir w, d. h. es gilt $A_i.w = b_i \ \forall\, i \in \mathrm{eq}(P)$ und $A_i.w < b_i \ \forall\, i \in M \setminus \mathrm{eq}(P)$.

Wir betrachten nun einen Punkt y auf der Strecke zwischen v und w, d. h. $y = \lambda w + (1 - \lambda)v$ mit $0 \leq \lambda \leq 1$. Aufgrund der Voraussetzungen gilt:

$$\begin{aligned} A_i.y &= b_i \quad \forall\, i \in \mathrm{eq}(P) \\ A_i.y &< b_i \quad \forall\, i \in M \setminus (\mathrm{eq}(P) \cup K). \end{aligned}$$

Ist $i \in K$, so gilt

$$\begin{aligned} A_i.y \leq b_i \quad &\iff \lambda A_i.w + (1 - \lambda)A_i.v \leq b_i \\ &\iff \lambda A_i.(w - v) \leq b_i - A_i.v \\ &\iff \lambda \geq \frac{b_i - A_i.v}{A_i.(w - v)} \qquad (\text{da } A_i.(w - v) < 0). \end{aligned}$$

Setzen wir

$$\begin{aligned} \mu &:= \max\left\{\frac{b_i - A_i.v}{A_i.(w - v)} \mid i \in K\right\}, \\ L &:= \left\{i \in K \mid \mu = \frac{b_i - A_i.v}{A_i.(w - v)}\right\}, \end{aligned}$$

dann gilt $z := \mu w + (1-\mu)v \in P, \emptyset \neq L \subseteq K \subseteq I$ und

$$\begin{array}{ll} A_{i.}z = b_i & \forall\, i \in L \\ A_{i.}z < b_i & \forall\, i \in K \setminus L. \end{array}$$

Daraus folgt, z ist ein innerer Punkt von $F := \text{fa}(L \cup \text{eq}(P))$. Nach (7.15) gilt dann $\text{eq}(F) = \text{eq}(\{z\}) = L \cup \text{eq}(P)$, und das bedeutet, dass F eine nichttriviale Seitenfläche von P mit $\text{eq}(F) \subseteq I \cup \text{eq}(P)$ ist. □

Wir werden nun wichtige Eigenschaften von Facetten bestimmen, Nichtredundanz kennzeichnen und Facetten charakterisieren.

(7.23) Satz. *Sei F eine Facette von $P = P(A, b)$, dann gilt:*

(a) $\text{eq}(P) \subset \text{eq}(F)$.

(b) Für alle $i \in \text{eq}(F) \setminus \text{eq}(P)$ *gilt*

$$F = \text{fa}(\{i\}) = \{x \in P \mid A_{i.}x = b_i\}.$$

Beweis : (a) gilt offensichtlich für alle nichttrivialen Seitenflächen von P.
(b) Die Abbildung fa ist inklusionsumkehrend, d. h.
$I \subseteq J \Rightarrow \text{fa}(I) \supseteq \text{fa}(J)$.

Daraus folgt $F = \text{fa}(\text{eq}(F)) \subseteq \text{fa}(\{i\})$. Da $i \notin \text{eq}(P)$, muss $\text{fa}(\{i\})$ eine echte Seitenfläche von P sein. Aus der Maximalität von F folgt die Behauptung. □

(7.24) Folgerung. *Sei $P = P(A, b)$ ein Polyeder und $\mathcal{F}$ die Menge der Facetten von P. Dann gilt:*

(a) $F_1, F_2 \in \mathcal{F},\ F_1 \neq F_2 \Longrightarrow \text{eq}(F_1) \cap \text{eq}(F_2) = \text{eq}(P)$.

(b) $|\mathcal{F}| \leq m - |\text{eq}(P)|$.

(c) Es gibt eine Menge $I \subseteq M$ mit folgenden Eigenschaften

(c_1) $I \subseteq M \setminus \text{eq}(P)$

(c_2) $|I| = |\mathcal{F}|$

(c_3) $F \in \mathcal{F} \Longleftrightarrow \exists$ *genau ein* $i \in I$ *mit* $F = \text{fa}(\{i\})$.

□

Jede Menge $I \subseteq M$ mit den Eigenschaften (c_1), (c_2), (c_3) wollen wir **Facetten-Indexmenge** nennen. Satz (7.23) (b) zeigt, dass man Facetten von P dadurch erhält, dass man in nur einer Ungleichung $A_{i.}x \leq b_i$ des Systems $Ax \leq b$ Gleichheit fordert. Jedoch ist es keineswegs so, dass für alle $i \in M$ die Menge $\text{fa}(\{i\})$ eine Facette von P ist! Dies gilt nur für solche $i \in M$, die in einer Facettenindexmenge enthalten sind.

(7.25) Satz. *Seien $P = P(A, b) \neq \emptyset$ ein Polyeder und $\mathcal{F}$ die Menge der Facetten von P. Seien M die Zeilenindexmenge von A, $I \subseteq M \setminus \text{eq}(P)$ und $J \subseteq \text{eq}(P)$. Sei $P' := \{x \mid A_{J.}x = b_J,\ A_{I.}x \leq b_I\}$, dann gilt:*

$$(a)\quad P = P' \iff \begin{array}{ll} (a_1) & \forall F \in \mathcal{F} \text{ gilt } \mathcal{I} \cap \text{eq}(\mathcal{F}) \neq \emptyset \text{ und} \\ (a_2) & \text{rang}(A_{J.}) = \text{rang}(A_{\text{eq}(P).}). \end{array}$$

$$(b)\quad P = P(A_{I\cup\text{eq}(P).}, b_{I\cup\text{eq}(P)}) \iff \forall F \in \mathcal{F} \text{ gilt } \mathcal{I} \cap \text{eq}(\mathcal{F}) \neq \emptyset.$$

Beweis : Mit $J = \text{eq}(P)$ folgt (b) direkt aus (a). Wir beweisen (a).

"$\Longrightarrow$ " Nach Definition gilt offenbar $J = \text{eq}_{P'}(P')$. Angenommen (a_2) ist nicht erfüllt, d. h. $\text{rang}(A_{J.}) < \text{rang}(A_{\text{eq}(P).})$. Dann folgt aus der Dimensionsformel (7.17) (a) $\dim(P') > \dim(P)$ und somit muss $P \neq P'$ gelten. Widerspruch !

Angenommen (a_1) ist nicht erfüllt. Dann gibt es eine Facette F von P mit $\text{eq}(F) \subseteq M \setminus I = (M \setminus I) \cup \text{eq}(P)$. Folglich gilt $P \neq P'$ nach Satz (7.22). Widerspruch !

"$\Longleftarrow$ " Wir zeigen zunächst, dass unter der Voraussetzung (a_2) gilt:

$$A_{J.}x = b_J \implies A_{\text{eq}(P).}x = b_{\text{eq}(P)}.$$

Da $P' \neq \emptyset$, gilt $\text{rang}(A_{J.}, b_J) = \text{rang}(A_{J.}) = \text{rang}(A_{\text{eq}(P).}) = \text{rang}(A_{\text{eq}(P).}, b_{\text{eq}(P)})$. Das heißt, für alle $i \in \text{eq}(P)$ existieren $K \subseteq J$ und λ_k, $k \in K$, mit $A_{i.} = \sum_{k\in K} \lambda_k A_{k.}$, $b_i = \sum_{k\in K} \lambda_k b_k$. Erfüllt also der Vektor x das System $A_{J.}x = b_J$, so gilt für alle $i \in \text{eq}(P)$

$$A_{i.}x = \sum_{k\in K} \lambda_k A_{k.}x = \sum_{k\in K} \lambda_k b_k = b_i.$$

Nach (a_1) gilt für jede Facette F von P: $\text{eq}(F) \not\subseteq M \setminus I$, und da Facetten maximale echte Seitenflächen sind und eq inklusionsumkehrend ist, folgt daraus $\text{eq}(G) \not\subseteq M \setminus I$ für alle echten Seitenflächen G von P. Aus Satz (7.22) folgt daher $P = P'$. □

(7.26) Folgerung. *Seien $P = P(A,b) \neq \emptyset$ ein Polyeder, $I \subseteq M \setminus \mathrm{eq}(P)$, $J \subseteq \mathrm{eq}(P)$ und $P = \{x \mid A_{J}.x = b_J,\ A_{I}.x \leq b_I\}$. Diese Darstellung von P ist genau dann irredundant, wenn gilt:*

(a) I ist eine Facetten-Indexmenge von P.

(b) $A_{J}.$ ist eine $(\mathrm{rang}(\mathrm{A}_{\mathrm{eq(P)}}.), \mathrm{n})$-Matrix mit vollem Zeilenrang.

□

(7.27) Folgerung. *Sei $P = P(A,b) \subseteq \mathbb{K}^n$ ein volldimensionales Polyeder (also $\mathrm{eq}(P) = \emptyset$, bzw. $\dim(P) = n$), dann gilt für alle $I \subseteq M$*

$$\begin{array}{ll} P(A_{I}., b_I) & \text{ist eine irredundante Beschreibung von } P \\ & \iff I \text{ ist Facetten-Indexmenge von } P. \end{array}$$

□

(7.28) Satz. *Sei $P = P(A,b)$ ein Polyeder, und F sei eine nichttriviale Seitenfläche von P. Dann sind äquivalent:*

(i) F ist eine Facette von P.

(ii) F ist eine maximale echte Seitenfläche von P.

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

(iv) F enthält $\dim(P)$ affin unabhängige Vektoren.

(v) Sei $c^T x \leq \gamma$ eine bezüglich P gültige Ungleichung mit $F = \{x \in P \mid c^T x = \gamma\}$, dann gilt für alle gültigen Ungleichungen $d^T x \leq \delta$ mit $F \subseteq \{x \in P \mid d^T x = \delta\}$: Es gibt einen Vektor $u \in \mathbb{K}^{\mathrm{eq}(P)}$ und $\alpha \in \mathbb{K}$, $\alpha \geq 0$ mit

$$\begin{aligned} d^T &= \alpha c^T + u^T A_{\mathrm{eq}(P)}. \,, \\ \delta &= \alpha\gamma + u^T b_{\mathrm{eq}(P)} \,. \end{aligned}$$

Beweis : (i) $\iff$ (ii): nach Definition.

(iv) $\iff$ (iii): trivial.

(iii) $\Longrightarrow$ (ii): Angenommen F ist keine Facette, dann existiert eine echte Seitenfläche G von P mit $F \subset G \subset P$. Aus (7.17) (c) folgt dann $\dim(F) \leq \dim(G) - 1 \leq \dim(P) - 2$, Widerspruch!

(i) $\Longrightarrow$ (v): Sei F eine beliebige Facette von P. Wir nehmen zunächst an, dass $A_{I.}x \leq b_I$, $A_{J.}x = b_J$ eine irredundante Darstellung von $P(A, b)$ ist mit $1 \in I$ und dass $F = \{x \in P \mid A_{1.}x = b_1\}$ gilt. Sei nun $d^T x \leq \delta$ eine gültige Ungleichung mit $F \subseteq \{x \in P \mid d^T x = \delta\}$. Aufgrund von Folgerung (7.4) gibt es Vektoren $v \geq 0$ und w mit $v^T A_{I.} + w^T A_{J.} = d^T$ und $v^T b_I + w^T b_J \leq \delta$ (in der Tat gilt hier Gleichheit, da $\{x \mid d^T x = \delta\}$ eine Stützhyperebene ist). Angenommen, es gibt einen Index $i \in I \setminus \{1\}$ mit $v_i > 0$, dann gilt nach (7.10) $i \in \text{eq}(F)$. Dies ist aber ein Widerspruch dazu, dass I eine Facettenindexmenge ist. Hieraus folgt (v).

(v) $\Longrightarrow$ (iii): Da F eine echte Seitenfläche von P ist, gilt $\dim(F) \leq \dim(P) - 1$. Angenommen $\dim(F) \leq \dim(P) - 2$. O. b. d. A. können wir annehmen, dass $F = \{x \in P \mid A_{1.}x = b_1\}$ gilt. Aus (7.16) folgt

$$\text{rang}(\text{A}_{\text{eq(F)}.}) \geq \text{rang}(\text{A}_{\text{eq(P)}.}) + 2.$$

Mithin gibt es einen Index $i \in \text{eq}(F) \setminus (\text{eq}(P) \cup \{1\})$, so dass der Zeilenvektor $A_{i.}$ linear unabhängig von den Zeilenvektoren $A_{j.}$, $j \in \text{eq}(P) \cup \{1\}$, ist. Das aber heißt, dass das System

$$A_{i.} = \alpha A_{1.} + u^T A_{\text{eq}(P).}$$

keine Lösung α, u hat. Wegen $F \subseteq \{x \in P \mid A_{i.}x = b_i\}$ ist dies ein Widerspruch zu (v). □

(7.29) Folgerung. *Seien $P = P(A, b) \subseteq \mathbb{K}^n$ ein volldimensionales Polyeder und $F = \{x \in P \mid c^T x = \gamma\}$ eine Seitenfläche von P. Dann sind äquivalent:*

(i) F *ist Facette von* P.

(ii) $\dim(F) = n - 1$.

(iii) Für alle gültigen Ungleichungen $d^T x \leq \delta$, $d \neq 0$, *mit* $F \subseteq \{x \in P \mid d^T x = \delta\}$ *gilt: Es existiert ein* $\alpha > 0$ *mit*

$$\begin{aligned} d^T &= \alpha c^T, \\ \delta &= \alpha \gamma. \end{aligned}$$

□

(7.30) Beispiel. Wir betrachten das Polyeder $P = P(A,b) \subseteq \mathbb{R}^2$, das wie folgt gegeben ist.

$$A = \begin{pmatrix} A_{1.} \\ A_{2.} \\ \cdot \\ \cdot \\ \cdot \\ A_{6.} \end{pmatrix} = \begin{pmatrix} 1 & -1 \\ -2 & 2 \\ 1 & 0 \\ 0 & 1 \\ -1 & 0 \\ -1 & -1 \end{pmatrix} \qquad b = \begin{pmatrix} 0 \\ 0 \\ 2 \\ 2 \\ -1 \\ -2 \end{pmatrix}.$$

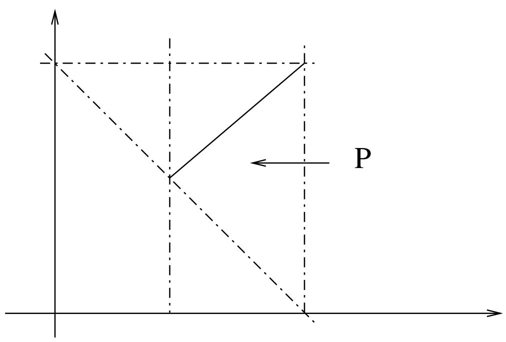


Abb. 7.2

P hat 4 Seitenflächen, nämlich $\emptyset$, P und $F_1 = \{\binom{2}{2}\}$, $F_2 = \{\binom{1}{1}\}$. F_1 und F_2 sind Facetten von P. Es gilt $\text{eq}(P) = \{1,2\}$, $\text{eq}(F_1) = \{1,2,3,4\}$, $\text{eq}(F_2) = \{1,2,5,6\}$, $\text{eq}(\emptyset) = \{1,\ldots,6\}$. Die Mengen $\{3,5\}$, $\{3,6\}$, $\{4,5\}$, $\{4,6\}$ sind die Facettenindexmengen von P. Eine irredundante Beschreibung von P ist z. B. gegeben durch

$$P = \{x \mid A_{1.}x = 0,\ A_{3.}x \le b_3,\ A_{5.}x \le b_5\}.$$

Übrigens sind die Ungleichungen $A_{i.}x \le b_i$, $i = 3,4,5,6$ redundant bezüglich $P(A,b)$. Die Ungleichungssysteme $A_{I.}x \le b_I$ mit $I = \{3,5\}$ oder $I = \{4,6\}$ sind z. B. ebenfalls redundant. Aber $A_{I.}x \le b_I$ ist nicht redundant bezüglich $P(A,b)$, falls $I = \{3,4,5\}$. □

Kapitel 8

Ecken und Extremalen

Im vorhergehenden Kapitel haben wir allgemeine Seitenflächen und speziell maximale Seitenflächen (Facetten) von Polyedern behandelt. Die Facetten sind bei der äußeren Beschreibung von Polyedern wichtig. Wir werden nun minimale Seitenflächen untersuchen und sehen, dass sie bei der inneren Darstellung von Bedeutung sind.

(8.1) Definition. *Es seien $P \subseteq \mathbb{K}^n$ ein Polyeder und F eine Seitenfläche von P. Gibt es $x \in \mathbb{K}^n$ und $0 \neq z \in \mathbb{K}^n$ mit*

$$\left\{ \begin{array}{l} F = \{x\} \\ F = x + \operatorname{lin}(\{z\}) \\ F = x + \operatorname{cone}(\{z\})\} \end{array} \right\}, \quad \textit{so heißt } F \quad \left\{ \begin{array}{l} \textbf{Ecke} \\ \textbf{Extremallinie} \\ \textbf{Extremalstrahl} \end{array} \right\}.$$

□

Ist $F = \{x\}$ eine Ecke von P, so sagen wir einfach “x ist eine Ecke von P”. Wir werden uns im weiteren insbesondere für Ecken interessieren und sie charakterisieren. Offenbar sind Ecken Seitenflächen der Dimension 0, während Extremallinien und Extremalstrahlen Seitenflächen der Dimension 1 sind. Allgemein heißen Seitenflächen der Dimension 1 **Kanten**. Kanten sind entweder Extremallinien, Extremalstrahlen oder Verbindungsstrecken zwischen zwei Ecken. Sind zwei Ecken x, y eines Polyeders P durch eine Kante verbunden, d. h. $\operatorname{conv}(\{x, y\})$ ist eine Seitenfläche von P, so nennt man x und y **adjazent** auf P.

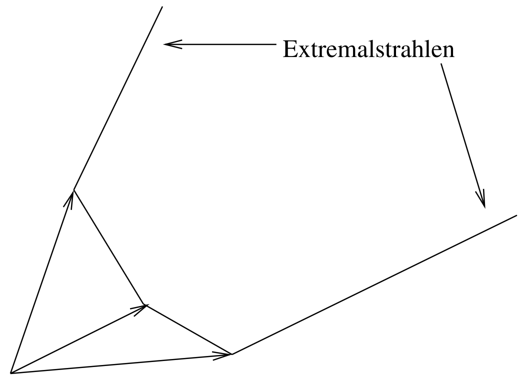


Abb. 8.1

8.1 Rezessionskegel, Linienraum, Homogenisierung

An dieser Stelle ist es nützlich einige weitere Objekte einzuführen, die man Polyedern (bzw. allgemeinen Mengen) zuordnen kann. Das Studium dieser Objekte ist für sich selbst betrachtet sehr interessant. Wir wollen diese Mengen jedoch nur als Hilfsmittel zur Vereinfachung von Beweisen verwenden, weswegen wir nicht weiter auf theoretische Untersuchungen dieser Mengen eingehen werden.

(8.2) Definition. *Sei $S \subseteq \mathbb{K}^n$ eine beliebige Menge. Wir definieren*

$$
\begin{array}{lrl}
(a) & \operatorname{rec}(S) & := \{y \in \mathbb{K}^n \mid \exists\, x \in S, \text{ so dass } \forall\, \lambda \geq 0 \text{ gilt } x + \lambda y \in S\}, \\
(b) & \operatorname{lineal}(S) & := \{y \in \mathbb{K}^n \mid \exists\, x \in S, \text{ so dass } \forall\, \lambda \in \mathbb{K} \text{ gilt } x + \lambda y \in S\}, \\
(c) & \operatorname{hog}(S) & := \{\binom{x}{1} \in \mathbb{K}^{n+1} \mid x \in S\}^{\circ\circ}.
\end{array}
$$

Die Menge $\operatorname{rec}(S)$ *heißt* **Rezessionskegel** *von* S, $\operatorname{lineal}(S)$ *heißt* **Linealitätsraum** *oder* **Linienraum** *von* S, *und* $\operatorname{hog}(S)$ *heißt* **Homogenisierung** *von* S.

□

Wir wollen nun die oben eingeführten Mengen bezüglich Polyedern charakterisieren. Nennen wir einen Vektor y mit $x+\lambda y \in S$ für alle $\lambda \geq 0$ eine "Richtung nach Unendlich", so besteht der Rezessionskegel einer Menge S aus allen Richtungen nach Unendlich. Für Polyeder gilt Folgendes:

(8.3) Satz. *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 : (a) $\operatorname{rec}(P) = P(A, 0)$. Ist $y \in \operatorname{rec}(P)$, so existiert ein $x \in P$ mit $x + \lambda y \in P \;\forall\, \lambda \geq 0$. Daraus folgt $b \geq A(x + \lambda y) = Ax + \lambda Ay$. Gäbe es eine Komponente von Ay, die größer als Null ist, sagen wir $(Ay)_i > 0$, so wäre der Vektor $x + \lambda_0 y$ mit

$$\lambda_0 = \frac{b_i - (Ax)_i}{(Ay)_i} + 1$$

nicht in $P(A, b)$, Widerspruch! Ist $y \in P(A, 0)$, so 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 $y \in \operatorname{rec}(P)$. (b) $P(A, 0) = \operatorname{cone}(E)$. Trivial! $\square$

Insbesondere folgt aus (8.3), dass $\operatorname{rec}(P) = \{y \in \mathbb{K}^n \mid \forall\, x \in P, \text{ und } \forall\, \lambda \geq 0 \text{ gilt } x + \lambda y \in P\}$. Ist P ein Kegel, so gilt natürlich $P = \operatorname{rec}(P)$, und offenbar ist ein Polyeder P genau dann ein Polytop, wenn $\operatorname{rec}(P) = \{0\}$. Abbildung 8.2 zeigt ein Polyeder und seinen Rezessionskegel.

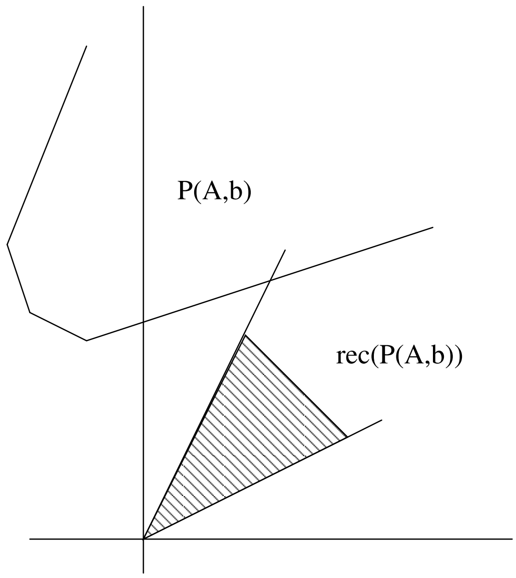


Abb. 8.2

Aus Definition (8.2) 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 lineare Teilraum $L \subseteq \mathbb{K}^n$, so dass $x + L \subseteq P$ für alle $x \in P$ gilt. Analytisch können wir $\operatorname{lineal}(P)$ wie folgt darstellen.

(8.4) Satz. *Sei* $P = P(A, b) = \operatorname{conv}(V) + \operatorname{cone}(E)$ *ein nichtleeres Polyeder,* *dann gilt*

$$\text{lineal}(P) = \{x \mid Ax = 0\} = \text{cone}(\{e \in E \mid -e \in \text{cone}(E)\}).$$

Beweis : Wegen $\text{lineal}(P) = \text{rec}(P) \cap (-\text{rec}(P))$ folgt die Behauptung direkt aus (8.3). □

Die Definition der Homogenisierung erscheint etwas kompliziert: Man wende zweimal die Kegelpolarität auf die Menge S an! Geometrisch betrachtet ist $\text{hog}(S)$ der Durchschnitt aller Ungleichungen mit rechter Seite 0, die gültig bezüglich $\{\binom{x}{1} \mid x \in S\}$ sind.

(8.5) Satz. *Sei* $P = P(A, b) = \text{conv}(V) + \text{cone}(E)$ *ein nichtleeres Polyeder. Sei*

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

dann gilt

$$\text{hog}(P) = P(B, 0) = \text{cone}(\{\tbinom{v}{1} \mid v \in V\}) + \text{cone}(\{\tbinom{e}{0} \mid e \in E\}).$$

Beweis : Setzen wir $P_1 := \{\binom{x}{1} \in \mathbb{K}^{n+1} \mid x \in P\}$, so gilt

$$P_1 = \text{conv}\{\tbinom{v}{1} \mid v \in V\} + \text{cone}\{\tbinom{e}{0} \mid e \in E\}.$$

Aus Folgerung (7.4) (iii) ergibt sich dann:

$$\begin{aligned} P_1^\circ &= \{z \in \mathbb{K}^{n+1} \mid z^T u \leq 0 \ \forall\, u \in P_1\} \\ &= \{z \in \mathbb{K}^{n+1} \mid z^T \tbinom{v}{1} \leq 0 \ \forall\, v \in V, z^T \tbinom{e}{0} \leq 0 \ \forall\, e \in E\} \\ &= \left\{ z \mid \begin{pmatrix} V^T, & \emptyset \\ E^T, & 0 \end{pmatrix} z \leq 0 \right\} = P\left(\begin{pmatrix} V^T, & \emptyset \\ E^T, & 0 \end{pmatrix}, 0 \right). \end{aligned}$$

Mit Folgerung (6.7) erhalten wir nun

$$\text{hog}(P) = P_1^{\circ\circ} = P\left(\begin{pmatrix} V^T, & \emptyset \\ E^T, & 0 \end{pmatrix}, 0 \right)^\circ = \text{cone}\begin{pmatrix} V & E \\ \emptyset^T & 0 \end{pmatrix}.$$

Die zweite Charakterisierung von $\text{hog}(P)$ folgt aus einer anderen Darstellung von P_1°. Es gilt nämlich mit Satz (7.2):

$$
\begin{aligned}
P_1^\circ &= \{\tbinom{y}{\lambda} \in \mathbb{K}^{n+1} \mid y^T x + \lambda 1 \le 0 \;\forall x \in P\} = \{\tbinom{y}{\lambda} \in \mathbb{K}^{n+1} \mid y^T x \le -\lambda \;\forall x \in P\} \\
&= \{\tbinom{y}{\lambda} \in \mathbb{K}^{n+1} \mid \tbinom{y}{-\lambda} \in P^\gamma\} = \{\tbinom{y}{\lambda} \in \mathbb{K}^{n+1} \mid \tbinom{y}{-\lambda} \in \text{cone}\begin{pmatrix} A^T & 0 \\ b^T & 1 \end{pmatrix}\} \\
&= \text{cone}\begin{pmatrix} A^T & 0 \\ -b^T & -1 \end{pmatrix}.
\end{aligned}
$$

Folgerung (6.10) impliziert nun

$$
\text{hog}(P) = P_1^{\circ\circ} = \left(\text{cone}\begin{pmatrix} A^T & 0 \\ -b^T & -1 \end{pmatrix}\right)^\circ = P\left(\begin{pmatrix} A & -b \\ 0 & -1 \end{pmatrix}, 0\right).
$$

$\square$

In Abbildung 8.3 sind ein Polyeder $P \subseteq \mathbb{R}^1$, die im obigen Beweis definierte Menge P_1 und $\text{hog}(P)$ dargestellt.

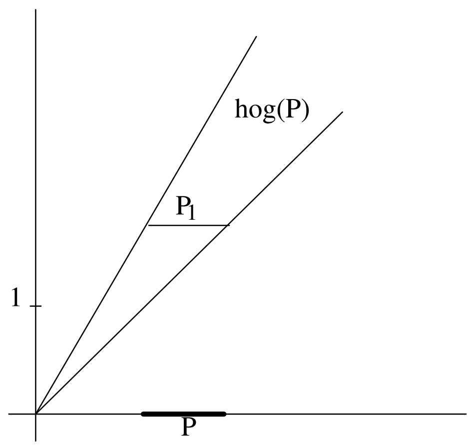


Abb. 8.3

(8.6) Bemerkung. Sei $P \subseteq \mathbb{K}^n$ ein Polyeder, dann gilt:

$$
\begin{aligned}
&(a) \quad x \in P &&\iff \tbinom{x}{1} \in \text{hog}(P). \\
&(b) \quad x \in \text{rec}(P) &&\iff \tbinom{x}{0} \in \text{hog}(P).
\end{aligned}
$$

Beweis : (a) ist trivial.

(b) Sei $P = P(A, b)$ eine Darstellung von P, dann gilt $\text{hog}(P) = P(B, 0)$ mit $B = \begin{pmatrix} A, & -b \\ 0, & -1 \end{pmatrix}$. Folglich gilt nach (8.5) und (8.3)

$$
\tbinom{x}{0} \in \text{hog}(P) \iff \tbinom{x}{0} \in P(B, 0) \iff Ax \le 0 \iff x \in \text{rec}(P).
$$

$\square$

8.2 Charakterisierungen von Ecken

Wir beginnen mit der Kennzeichnung von Ecken und Extremalstrahlen von Kegeln.

(8.7) Satz. *Sei $K \subseteq \mathbb{K}^n$ ein polyedrischer Kegel, dann gilt:*

(a) Ist x 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 $F = \operatorname{cone}(\{z\})$ eine Seitenfläche von K ist.

Beweis : Nach (2.6) gibt es eine Matrix A mit $K = P(A, 0)$. Aufgrund von (7.11) ist daher jede nichtleere Seitenfläche von K ein Kegel, der den Nullvektor enthält.

(a) Ist x Ecke, dann gilt also $0 \in \{x\}$ und somit $x = 0$.

(b) "$\Longleftarrow$" nach Definition.

"$\Longrightarrow$" Nach (8.1) existieren $x \in \mathbb{K}^n$ und $z \in \mathbb{K}^n \setminus \{0\}$ mit $F = x + \operatorname{cone}(\{z\})$. Nach obiger Bemerkung gilt $0 \in F$, und somit existiert ein $\overline{\lambda} \geq 0$ mit $0 = x + \overline{\lambda} z$, d. h. $x = -\overline{\lambda} z$. Da F ein Kegel ist, gilt $\{\lambda x \mid \lambda \in \mathbb{K}_+\} = \{\lambda(-\overline{\lambda} z) \mid \lambda \in \mathbb{K}_+\} \subseteq F$. Falls $\overline{\lambda} \neq 0$, gilt aber $\{-\lambda z \mid \lambda \geq 0\} \not\subseteq x + \operatorname{cone}(\{z\})$, ein Widerspruch. Daraus folgt $x = 0$. $\square$

Der folgende Hilfssatz über innere Punkte wird im Weiteren benötigt.

(8.8) Lemma. *Ist F eine nichtleere Seitenfläche von $P = P(A, b)$, gilt $I = \operatorname{eq}(F)$, und ist $B = \{y^1, \ldots, y^k\}$ eine Basis des Kerns von $A_{I.}$, dann gibt es zu jedem inneren Punkt $x \in F$ von F ein $\varepsilon > 0$, so dass $x \pm \varepsilon y^j \in P$ für alle $j = 1, \ldots, k$ gilt.*

Beweis : Sei x innerer Punkt von F und $J = \{1, \ldots, m\} \setminus I$. Nach (7.15) gilt $\operatorname{eq}(\{x\}) = I$, also $A_{J.}x < b_J$ und ferner $A_{I.}(x \pm \varepsilon y^j) = b_I$ für alle $\varepsilon \in \mathbb{K}$. Für ε genügend klein gilt dann offenbar auch $A_{J.}(x \pm \varepsilon y^j) \leq b_J$. $\square$

Wir wollen nun die Ecken eines Polyeders charakterisieren.

(8.9) Satz. *Seien $P = P(A, b) \subseteq \mathbb{K}^n$ ein Polyeder und $x \in P$. Dann sind die folgenden Aussagen äquivalent:*

(1) x ist eine Ecke von P.

(2) $\{x\}$ *ist eine nulldimensionale Seitenfläche von* P*.*

(3) x *ist keine echte Konvexkombination von Elementen von* P*, d. h.* $y, z \in P$*,* $y \neq z$*,* $0 < \lambda < 1 \implies x \neq \lambda y + (1 - \lambda)z$*.*

(4) $P \setminus \{x\}$ *ist konvex.*

(5) $\operatorname{rang}(A_{\operatorname{eq}(\{x\}).}) = n$*.*

(6) $\exists\, c \in \mathbb{K}^n \setminus \{0\}$*, so dass* x *die eindeutig bestimmte Optimallösung des linearen Programms* $\max c^T y$*,* $y \in P$ *ist.*

Beweis : (1) $\iff$ (2). Definition!

(1) $\implies$ (6). Nach Definition ist $\{x\}$ eine Seitenfläche, also existiert eine bezüglich P gültige Ungleichung $c^T y \leq \gamma$, so dass $\{y \in P \mid c^T y = \gamma\} = \{x\}$ gilt. Folglich ist x die eindeutig bestimmte Optimallösung von $\max c^T y$, $y \in P$. Ist $P \neq \{x\}$, so ist $c \neq 0$, andernfalls kann $c \neq 0$ gewählt werden.

(6) $\implies$ (3). Sei x die eindeutige Optimallösung von $\max c^T u$, $u \in P$ mit Wert γ. Gilt $x = \lambda y + (1 - \lambda)z$ für $y, z \in P$, $y \neq z$, $0 < \lambda < 1$, dann folgt $\gamma = c^T x = c^T(\lambda y + (1 - \lambda)z) = \lambda c^T y + (1 - \lambda)c^T z \leq \lambda\gamma + (1 - \lambda)\gamma = \gamma$. Dies impliziert $c^T y = \gamma = c^T z$. Widerspruch!

(3) $\iff$ (4) trivial.

(3) $\implies$ (5). Sei $I = \operatorname{eq}(\{x\})$, dann ist x innerer Punkt von $F = \{y \in P \mid A_{I.}y = b_I\}$. Angenommen $\operatorname{rang}(A_{I.}) < n$, dann existiert ein Vektor $y \neq 0$ im Kern von $A_{I.}$, und nach Lemma (8.8) ein $\varepsilon > 0$, so dass $x \pm \varepsilon y \in P$. Daraus folgt $x = \frac{1}{2}(x + \varepsilon y) + \frac{1}{2}(x - \varepsilon y)$, also ist x echte Konvexkombination von Elementen von P, Widerspruch!

(5) $\implies$ (2). Sei $I = \operatorname{eq}(\{x\})$, dann hat $A_{I.}y = b_I$ nach Voraussetzung eine eindeutig bestimmte Lösung, nämlich x. Daraus folgt $F = \{y \in P \mid A_{I.}y = b_I\} = \{x\}$, also ist $\{x\}$ eine nulldimensionale Seitenfläche. □

Für Polyeder der Form $P^=(A, b) = \{x \in \mathbb{K}^n \mid Ax = b, x \geq 0\}$ gibt es eine weitere nützliche Kennzeichnung der Ecken. Ist $x \in \mathbb{K}^n$, so setzen wir

$$\operatorname{supp}(x) = \{i \in \{1, \ldots, n\} \mid x_i \neq 0\}.$$

Die Indexmenge $\operatorname{supp}(x)$ heißt **Träger** von x.

(8.10) Satz. *Für* $x \in P^=(A, b) \subseteq \mathbb{K}^n$ *sind folgende Aussagen äquivalent:*

(1) x *ist Ecke von* $P^=(A,b)$.

(2) $\text{rang}(A_{\cdot\text{supp}(x)}) = |\text{supp}(x)|$.

(3) Die Spaltenvektoren $A_{\cdot j}$, $j \in \text{supp}(x)$, *sind linear unabhängig.*

Beweis : (2) $\iff$ (3) trivial.

(1) $\iff$ (2). Sei $D = \begin{pmatrix} A \\ -A \\ -I \end{pmatrix}$, $d = \begin{pmatrix} b \\ -b \\ 0 \end{pmatrix}$, dann gilt $P(D,d) = P^=(A,b)$. Mit Satz (8.9) gilt nun:

$$\begin{aligned}
x \text{ Ecke von } P^=(A,b) \quad & \iff x \text{ Ecke von } P(D,d) \\
& \iff \text{rang}(D_{\text{eq}(\{x\})\cdot}) = n \\
& \iff \text{rang}\begin{pmatrix} A \\ -A \\ I_{J\cdot} \end{pmatrix} = n \text{ mit } J = \{j \mid x_j = 0\} \\
& \qquad\qquad\qquad\qquad\quad = \{1,\dots,n\} \setminus \text{supp}(x) \\
\text{(Streichen der Spalten } j \in J) \quad & \iff \text{rang}\begin{pmatrix} A_{\cdot,\text{supp}(x)} \\ -A_{\cdot,\text{supp}(x)} \end{pmatrix} = n - |J| = |\text{supp}(x)| \\
& \iff \text{rang}(A_{\cdot,\text{supp}(x)}) = |\text{supp}(x)|.
\end{aligned}$$

□

8.3 Spitze Polyeder

Nicht alle Polyeder haben Ecken, so z. B. der Durchschnitt von weniger als n Halbräumen des $\mathbb{K}^n$. Polyeder mit Ecken sind besonders für die lineare Optimierung wichtig.

(8.11) Definition. *Ein Polyeder heißt* **spitz**, *wenn es eine Ecke besitzt.*

□

Wir wollen im nachfolgenden spitze Polyeder charakterisieren und einige Aussagen über spitze Polyeder beweisen.

(8.12) Satz. *Sei* $P = P(A,b) \subseteq \mathbb{K}^n$ *ein nichtleeres Polyeder, dann sind äquivalent:*

(1) P *ist spitz.*

(2) $\operatorname{rang}(A) = n$.

(3) $\operatorname{rec}(P)$ *ist spitz, d. h.* 0 *ist eine Ecke von* $\operatorname{rec}(P)$.

(4) Jede nichtleere Seitenfläche von P *ist spitz.*

(5) $\operatorname{hog}(P)$ *ist spitz.*

(6) P *enthält keine Gerade.*

(7) $\operatorname{rec}(P)$ *enthält keine Gerade.*

(8) $\operatorname{lineal}(P) = \{0\}$.

Beweis : (1) $\Longrightarrow$ (2). Ist x eine Ecke von P, so gilt nach (8.9) $n = \operatorname{rang}(A_{\mathrm{eq}(\{x\}).}) \leq \operatorname{rang}(A) \leq n$, also $\operatorname{rang}(A) = n$.

(2) $\Longrightarrow$ (1). Sei $x \in P$ so gewählt, dass $I := \mathrm{eq}(\{x\})$ maximal bezüglich Mengeninklusion ist. Sei $F = \{y \in P \mid A_{I.}y = b_I\}$, dann ist x innerer Punkt von F. Ist $\operatorname{rang}(A_{I.}) < n$, dann enthält der Kern von $A_{I.}$ einen Vektor $y \neq 0$, und mit Lemma (8.8) gibt es ein $\varepsilon > 0$, so dass $x \pm \varepsilon y \in P$. Die Gerade $G = \{x + \lambda y \mid \lambda \in \mathbb{K}\}$ trifft mindestens eine der Hyperebenen $H_j = \{y \mid A_{j.}y = b_j\}$, $j \notin I$, da $\operatorname{rang}(A) > \operatorname{rang}(A_{I.})$. Also muss es ein $\delta \in \mathbb{K}$ geben, so dass $x + \delta y \in P$ und $\mathrm{eq}(\{x + \delta y\}) \supset I$. Dies widerspricht der Maximalität von I.

Aus der Äquivalenz von (1) und (2) folgt direkt die Äquivalenz der Aussagen (3), (4), (5), da

$$\begin{aligned}
\operatorname{rec}(P) &= P(A, 0), \text{ nach (8.3)},\\
\operatorname{hog}(P) &= P\left(\begin{pmatrix} A, & -b \\ 0, & -1 \end{pmatrix}, 0\right), \text{ nach (8.5)},\\
F &= \{x \in \mathbb{K}^n \mid Ax \leq b, A_{\mathrm{eq}(F).}x \leq b_{\mathrm{eq}(F)}, -A_{\mathrm{eq}(F).}x \leq -b_{\mathrm{eq}(F)}\}
\end{aligned}$$

für alle Seitenflächen F von P gilt.

(3) $\Longrightarrow$ (6). Angenommen P enthält eine Gerade $G = \{u + \lambda v \mid \lambda \in \mathbb{K}\}$, $v \neq 0$, dann gilt $b \geq A(u + \lambda v) = Au + \lambda Av$ für alle $\lambda \in \mathbb{K}$. Daraus folgt $A(\lambda v) \leq 0$ für alle $\lambda \in \mathbb{K}$ und somit $v, -v \in \operatorname{rec}(P)$, d. h. $0 = \frac{1}{2}v + \frac{1}{2}(-v)$ ist eine Konvexkombination. Also ist 0 keine Ecke von $\operatorname{rec}(P)$.

(6) $\Longrightarrow$ (3). Ist $\operatorname{rec}(P)$ nicht spitz, so ist 0 echte Konvexkombination von Vektoren aus $\operatorname{rec}(P)$, sagen wir $0 = \lambda u + (1 - \lambda)v$, $u \neq 0 \neq v$, $0 < \lambda < 1$. Dann aber

ist $G = \{\lambda u \mid \lambda \in \mathbb{K}\}$ eine Gerade in $\operatorname{rec}(P)$, und für alle $x \in P$ ist $x + G$ eine Gerade in P.

Die Äquivalenz von (7) und (8) zu den übrigen Aussagen ist nun offensichtlich. □

(8.13) Folgerung. *Sei $P = P^=(A, b) \subseteq \mathbb{K}^n$, dann gilt*

$$P \neq \emptyset \iff P \text{ spitz.}$$

Beweis : Es ist $P = P(D, d)$ mit $D = \begin{pmatrix} A \\ -A \\ -I \end{pmatrix}$, $d = \begin{pmatrix} b \\ -b \\ 0 \end{pmatrix}$, und offenbar hat D den Rang n. Aus (8.12) folgt dann die Behauptung. □

(8.14) Folgerung. *Sei $P \subseteq \mathbb{K}^n$ ein Polytop, dann gilt*

$$P \neq \emptyset \iff P \text{ spitz.}$$

Beweis : Da P beschränkt ist, gibt es einen Vektor u mit $P \subseteq \{x \mid x \leq u\}$. Gilt $P = P(A, b)$, so folgt daraus $P = P(D, d)$ mit $D = \binom{A}{I}$, $d = \binom{b}{u}$. D hat den Rang n, daraus folgt mit (8.12) die Behauptung. □

(8.15) Folgerung. *Das Polyeder P sei spitz, und das lineare Programm $\max c^T x$, $x \in P$ habe eine Optimallösung, dann hat dieses lineare Programm auch eine optimale Lösung, die eine Ecke von P ist (eine sogenannte optimale* **Ecklösung***).*

Beweis : $F = \{x \in P \mid c^T x = \max\{c^T y \mid y \in P\}\}$ ist eine nichtleere Seitenfläche von P, die nach Satz (8.12) eine Ecke hat. □

(8.16) Folgerung. *Ist P ein nichtleeres Polytop, dann hat jedes lineare Programm $\max c^T x$, $x \in P$ eine optimale Ecklösung.*

□

Das folgende Korollar wird sich als sehr wichtig erweisen.

(8.17) Folgerung. *Lineare Programme der Form*

$$\begin{array}{rl} \max c^T x & \\ Ax = b & \\ x \geq 0 & \end{array} \qquad \text{oder} \qquad \begin{array}{rl} \max c^T x & \\ Ax \leq b & \\ x \geq 0 & \end{array}$$

besitzen genau dann eine endliche Optimallösung, wenn sie eine optimale Ecklösung besitzen.

Beweis : Nach (8.13) ist $P^{=}(A,b)$ spitz (falls nicht leer) und somit hat das erste der obigen LP's nach (8.15) eine optimale Ecklösung, falls es überhaupt eine optimale Lösung hat. Analog folgt die Behauptung für das zweite LP. □

8.4 Extremalen von spitzen Polyedern

Wir wollen nachfolgend einige Aussagen über spitze Polyeder beweisen, die sich — entsprechend modifiziert — auch für allgemeine Polyeder zeigen lassen. Dabei treten jedoch einige unschöne technische Komplikationen auf, so dass wir hier auf die Behandlung dieser Verallgemeinerung verzichten.

(8.18) Definition. *Sei P ein Polyeder. Ein Vektor $z \in \operatorname{rec}(P) \setminus \{0\}$ heißt* **Extremale** *(oder Extremalvektor) von P, wenn $\operatorname{cone}(\{z\})$ ein Extremalstrahl von $\operatorname{rec}(P)$ ist.*

□

Nur spitze Polyeder haben Extremalen. Denn ist P nicht spitz, so ist nach (8.12) $\operatorname{rec}(P)$ nicht spitz, also ist der Nullvektor eine echte Konvexkombination zweier von Null verschiedenen Vektoren, sagen wir $0 = \lambda u + (1-\lambda)v$, $0 < \lambda < 1$. Ist $F = \operatorname{cone}(\{z\})$ ein Extremalstrahl von $\operatorname{rec}(P)$, so gibt es eine bezüglich $\operatorname{rec}(P)$ gültige Ungleichung $c^T x \leq 0$ mit $F = \{x \in \operatorname{rec}(P) \mid c^T x = 0\}$. Nun gilt $0 = c^T 0 = c^T(\lambda u + (1-\lambda)v) = \lambda c^T u + (1-\lambda)c^T v \leq 0$. Aus $c^T u \leq 0$, $c^T v \leq 0$ folgt $c^T u = c^T v = 0$ und somit $u, v \in F$, ein Widerspruch. Aussagen über Extremalen machen also nur für spitze Polyeder Sinn.

Ist K speziell ein spitzer polyedrischer Kegel, so ist (wegen $\operatorname{rec}(K) = K$) eine Extremale von K ein Vektor $z \in K$, so dass $\operatorname{cone}(\{z\})$ ein Extremalstrahl von K ist. Das heißt, jeder auf einem Extremalstrahl von K gelegener und von Null verschiedener Vektor ist eine Extremale von K.

(8.19) Satz. *Seien $P = P(A,b) \subseteq \mathbb{K}^n$ ein spitzes Polyeder und $z \in \operatorname{rec}(P) \setminus \{0\}$. Dann sind äquivalent:*

(1) z *ist eine Extremale von* P.

(2) $\operatorname{cone}(\{z\})$ *ist ein Extremalstrahl von* $\operatorname{rec}(P)$.

(3) z *läßt sich nicht als echte konische Kombination zweier linear unabhängiger Elemente von* $\operatorname{rec}(P)$ *darstellen.*

(4) $(\operatorname{rec}(P) \setminus (\operatorname{cone}\{z\})) \cup \{0\}$ *ist ein Kegel.*

(5) rang$(A_{\text{eq}(\{z\}).}) = n - 1$ *(*eq bezüglich $Ax \leq 0$*).*

Beweis : (1) $\Longleftrightarrow$ (2) Definition.

(2) $\Longrightarrow$ (3). Ist $F := \operatorname{cone}(\{z\})$ ein Extremalstrahl von $\operatorname{rec}(P)$, so ist F eine eindimensionale Seitenfläche von $\operatorname{rec}(P)$, d. h. F kann keine zwei linear unabhängigen Vektoren enthalten. Insbesondere gibt es eine bezüglich $\operatorname{rec}(P)$ gültige Ungleichung $c^T x \leq 0$ mit $F = \{x \in \operatorname{rec}(P) \mid c^T x = 0\}$. Gibt es zwei linear unabhängige Vektoren $u, v \in \operatorname{rec}(P)$ und $\lambda, \mu > 0$ mit $z = \lambda u + \mu v$, so gilt $0 = c^T z = c^T(\lambda u + \mu v) = \lambda c^T u + \mu c^T v \leq 0$. Daraus folgt $c^T u = c^T v = 0$, d. h. $u, v \in F$, ein Widerspruch.

(3) $\Longleftrightarrow$ (4) trivial.

(3) $\Longrightarrow$ (5). Sei $I = \text{eq}(\{z\})$, dann ist z innerer Punkt von $F := \{x \in \operatorname{rec}(P) \mid A_{I.}x = 0\}$. Ist rang$(A_{I.}) < n - 1$, dann enthält der Kern von $A_{I.}$ einen von z linear unabhängigen Vektor u. Nach Lemma (8.8) gibt es ein $\varepsilon > 0$, so dass $z \pm \varepsilon u \in \operatorname{rec}(P)$ gilt. Dann aber gilt $z = \frac{1}{2}(z + \varepsilon u) + \frac{1}{2}(z - \varepsilon u)$, d. h. z ist echte konische Kombination von zwei linear unabhängigen Elementen von $\operatorname{rec}(P)$, Widerspruch. Offenbar ist rang$(A_{I.}) \neq n$.

(5) $\Longrightarrow$ (2). Sei $I = \text{eq}(\{z\})$, dann folgt für $F = \{x \in \operatorname{rec}(P) \mid A_{I.}x = 0\}$ aus der Voraussetzung, dass $\dim(F) = 1$ gilt. Da $\operatorname{rec}(P)$ spitz ist, enthält nach (8.12) $\operatorname{rec}(P)$ keine Gerade, also muss die eindimensionale Seitenfläche F der Strahl $\operatorname{cone}(\{z\})$ sein. $\square$

Wir wollen nun noch eine Beziehung zwischen den Ecken und Extremalen eines spitzen Polyeders und den Extremalen seiner Homogenisierung aufzeigen.

(8.20) Satz. *Sei* $P \subseteq \mathbb{K}^n$ *ein spitzes Polyeder, dann gilt:*

(a) x *ist Ecke von* $P \iff \binom{x}{1}$ *ist Extremale von* $\operatorname{hog}(P)$.

(b) z *ist Extremale von* $P \iff \binom{z}{0}$ *ist Extremale von* $\operatorname{hog}(P)$.

Beweis : Sei $P = P(A, b)$, dann gilt nach Satz (8.5) $\mathrm{hog}(P) = P(B, 0)$ mit

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

(a) Sei $I = \mathrm{eq}(\{x\})$ bezüglich $P(A, b)$. x ist Ecke von $P \overset{(8.9)}{\Longleftrightarrow} \mathrm{rang}(A_{I.}) = n \Longleftrightarrow \mathrm{rang}(B_{\mathrm{eq}(\{\binom{x}{1}\}).}) = n$ (denn die neu hinzugekommene Ungleichung ist nicht mit Gleichheit erfüllt) $\overset{(8.19)}{\Longleftrightarrow} \binom{x}{1}$ ist Extremale von $\mathrm{hog}(P)$.

(b) z ist Extremale von $P \Longleftrightarrow z$ ist Extremale von $\mathrm{rec}(P) \overset{(8.19)}{\Longleftrightarrow} \mathrm{rang}(A_{\mathrm{eq}(\{z\}).}) = n - 1 \Longleftrightarrow \mathrm{rang}(B_{\mathrm{eq}(\{\binom{z}{0}\}).}) = n \overset{(8.19)}{\Longleftrightarrow} \binom{z}{0}$ ist Extremale von $\mathrm{hog}(P)$. □

8.5 Einige Darstellungssätze

Wir knüpfen hier an Abschnitt 6.3 an, wo wir bereits verschiedene Darstellungssätze bewiesen haben. Einige dieser Sätze können wir nun mit Hilfe der in diesem Kapitel gewonnenen Erkenntnisse verschärfen.

Ist K ein polyedrischer Kegel und gilt $K = \mathrm{cone}(E)$, dann nennen wir E eine **Kegelbasis** von K, wenn es keine echte Teilmenge E' von E gibt mit $K = \mathrm{cone}(E')$ und wenn jede andere minimale Menge F mit $K = \mathrm{cone}(F)$ dieselbe Kardinalität wie E hat. Ist P ein Polytop, dann heißt eine Menge V mit $P = \mathrm{conv}(V)$ **konvexe Basis** von P, wenn V keine echte Teilmenge besitzt, deren konvexe Hülle P ist, und wenn jede andere minimale Menge W mit $P = \mathrm{conv}(W)$ dieselbe Kardinalität wie V hat.

Trivialerweise sind die Elemente einer Kegelbasis E **konisch unabhängig**, d. h. kein $e \in E$ ist konische Kombination der übrigen Elemente von E; und die Elemente einer konvexen Basis sind **konvex unabhängig**, d. h. kein Element von V ist Konvexkombination der übrigen Elemente von V. Es gilt aber keineswegs, dass jeder Vektor $x \in \mathrm{cone}(E)$ bzw. $x \in \mathrm{conv}(V)$ eine eindeutige konische bzw. konvexe Darstellung durch Vektoren aus E bzw. V hat. In dieser Hinsicht unterscheiden sich also Kegelbasen und konvexe Basen von Vektorraumbasen.

(8.21) Satz. *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 man dadurch erhält, dass man aus jedem Extremalstrahl von K genau einen von Null verschiedenen Vektor (also eine Extremale von K) auswählt.*

Beweis : Ist z Extremale von K, so ist $K' := (K \setminus \text{cone}\{z\}) \cup \{0\}$ nach (8.19) ein Kegel. Folglich gilt $\text{cone}(E) \subseteq K'$ für alle Teilmengen E von K'. Also muss jede Kegelbasis von K mindestens ein (aus der Basiseigenschaft folgt sofort "genau ein") Element von $\text{cone}\{z\} \setminus \{0\}$ enthalten.

Zum Beweis, dass jede wie in (2) spezifizierte Menge eine Kegelbasis ist, benutzen wir Induktion über $d = \dim K$. Für Kegel der Dimension 1 ist die Behauptung trivial. Sei die Behauptung für Kegel der Dimension d richtig und K ein Kegel mit $\dim K = d + 1$. Sei $y \in K \setminus \{0\}$ 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 (8.9)). Sei $z \in \{x \mid c^T x = 0\} \setminus \{0\}$. Dann ist für die Gerade

$$G = \{y + \lambda z \mid \lambda \in \mathbb{K}\}$$

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 der eindeutige Maximalpunkt). Folglich gibt es zwei Punkte z_1 und z_2, die auf echten Seitenflächen, sagen wir F_1 und F_2, von K liegen, so dass $K \cap G = \text{conv}(\{z_0, z_1\})$. Die Dimensionen der Seitenflächen F_1, F_2 sind höchstens d, F_1 und F_2 sind Kegel, und die Extremalstrahlen von F_1 und F_2 sind Extremalstrahlen von K. Nach Induktionsvoraussetzung werden z_1 und z_2 durch die in (2) festgelegten Mengen bezüglich F_1 und F_2 konisch erzeugt. Daraus folgt, dass y durch jede Menge des Typs (2) konisch erzeugt werden kann. Dies impliziert die Behauptung. □

(8.22) Folgerung. *Jeder spitze polyedrische Kegel besitzt eine — bis auf positive Skalierung der einzelnen Elemente — eindeutige Kegelbasis.*

□

(8.23) Folgerung. *Jeder spitze polyedrische Kegel $K \neq \{0\}$ ist die Summe seiner Extremalstrahlen, d. h. sind $\text{cone}(\{e_i\})$, $i = 1, \dots, k$ die Extremalstrahlen von K, so gilt*

$$K = \text{cone}(\{e_1, \dots, e_k\}) = \sum_{i=1}^{k} \text{cone}(\{e_i\}).$$

Beweis : Klar. □

Der folgende Satz verschärft (6.13) für spitze Polyeder.

(8.24) Satz. *Jedes spitze Polyeder P lässt sich darstellen als die Summe der konvexen Hülle seiner Ecken und der konischen Hülle seiner Extremalen, d. h. sind V die Eckenmenge von P und $\operatorname{cone}(\{e\})$, $e \in E$, die Extremalstrahlen von $(\operatorname{rec}(P))$ (bzw. ist E eine Kegelbasis von $\operatorname{rec}(P)$), so gilt*

$$P = \operatorname{conv}(V) + \operatorname{cone}(E).$$

Beweis : Sei hog(P) die Homogenisierung von P. Da P spitz ist, ist hog(P) nach (8.12) ein spitzer Kegel. Nach (8.23) ist hog(P) die Summe seiner Extremalstrahlen $\operatorname{cone}(\{e'_i\})$ $i = 1, \ldots, k$. O. b. d. A. können wir annehmen, dass $e'_i = \binom{v_i}{1}$, $i = 1, \ldots, p$, und $e'_i = \binom{e_i}{0}$, $i = p+1, \ldots, k$ gilt. Aus (8.20) folgt: $V = \{v_1, \ldots, v_p\}$ ist die Eckenmengen von P und $E = \{e_{p+1}, \ldots, e_k\}$ ist die Extremalenmenge von P. Nach (8.6) gilt $x \in P \iff \binom{x}{1} \in$ hog(P) und somit folgt $x \in P \iff x = \sum_{i=1}^{p} \lambda_i v_i + \sum_{i=p+1}^{k} \mu_i e_i$, $\lambda_i, \mu_i \geq 0$, $\sum_{i=1}^{p} \lambda_i = 1$, d. h. $x \in \operatorname{conv}(V) + \operatorname{cone}(E)$. □

(8.25) Folgerung *(Satz von Krein-Milman). Sei P ein Polyeder und V die Menge seiner Ecken, dann gilt*

$$P \text{ ist ein Polytop} \iff P = \operatorname{conv}(V).$$

□

(8.26) Folgerung. *Polytope haben eine eindeutige konvexe Basis.*

□

Für die lineare Optimierung ist die folgende Beobachtung wichtig.

(8.27) Satz. *Seien $P \subseteq \mathbb{K}^n$ 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 : Seien V die Eckenmenge von P und E eine Kegelbasis von $\operatorname{rec}(P)$, dann gilt nach (8.24) $P = \operatorname{conv}(V) + \operatorname{cone}(E)$. Es ist $\gamma := \max\{c^T v \mid v \in V\} < \infty$, da V endlich und $\gamma = \max\{c^T x \mid x \in \operatorname{conv}(V)\}$. Gilt $c^T e \leq 0, \forall\, e \in E$, so ist $\gamma = \max\{c^T x \mid x \in P\} < \infty$. Falls also $\max\{c^T x \mid x \in P\}$ unbeschränkt ist, muss für mindestens ein $e \in E$ gelten $c^T e > 0$. Die umgekehrte Richtung ist trivial. □

Wie bereits bemerkt, haben Elemente von spitzen Polyedern P i. a. keine eindeutige Darstellung als konvexe und konische Kombination von Ecken und Extremalen. Man kann jedoch zeigen, dass zu einer derartigen Darstellung von Elementen von P nicht allzu viele Ecken und Extremalen benötigt werden.

(8.28) Satz. *Es seien $K \subseteq \mathbb{K}^n$ ein spitzer Kegel und $0 \neq x \in K$. Dann gibt es Extremalen $y_1, \ldots, y_d$ von K, wobei $d \leq \dim(K) \leq n$ gilt, mit*

$$x = \sum_{i=1}^{d} y_i.$$

Beweis : Es seien $\operatorname{cone}(\{e_i\})$ die Extremalstrahlen von K, $i = 1, \ldots, k$, dann gibt es nach (8.23) Skalare $\lambda_i \geq 0$, $i = 1, \ldots, k$, mit

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

Unter allen möglichen Darstellungen von x dieser Art sei die obige eine, so dass $I = \{i \in \{1, \ldots, k\} \mid \lambda_i > 0\}$ minimal ist. Sagen wir, es gilt $I = \{1, \ldots, d\}$. Angenommen die Vektoren e_i, $i \in I$ sind linear abhängig, dann gibt es $\mu_1, \ldots, \mu_d \in \mathbb{K}$, nicht alle μ_i Null, so dass gilt

$$\sum_{i=1}^{d} \mu_i e_i = 0.$$

Angenommen $\mu_i \geq 0$ für $i = 1, \ldots, d$, und o. B. d. A. sei $\mu_1 > 0$. Dann ist $-e_1 = \sum_{i=2}^{d} \frac{\mu_i}{\mu_1} e_i$ eine konische Kombination, und nach Satz (8.4) gilt $e_1 \in \operatorname{lineal}(K)$. Dies widerspricht nach (8.12) der Voraussetzung K ist spitz.

O. B. d. A. können wir daher annehmen, dass gilt $\mu_1 < 0$ und

$$\frac{\lambda_1}{\mu_1} = \max\{\frac{\lambda_i}{\mu_i} \mid \mu_i < 0\}.$$

Daraus folgt

$$\begin{aligned} e_1 &= -\textstyle\sum_{i=2}^{d} \frac{\mu_i}{\mu_1} e_i\,, \\ x &= \textstyle\sum_{i=2}^{d} (\lambda_i - \frac{\lambda_1}{\mu_1}\mu_i) e_i\,. \end{aligned}$$

Diese Darstellung von x ist eine konische Kombination, denn

$$\mu_i \geq 0 \implies (\lambda_i - \tfrac{\lambda_1}{\mu_1}\mu_i) \geq 0\,,$$
$$\mu_i < 0 \implies \tfrac{\lambda_i}{\mu_i} \leq \tfrac{\lambda_1}{\mu_1} \implies \lambda_i \geq \tfrac{\lambda_1}{\mu_1}\mu_i,$$

also kann x mit weniger als d Extremalen konisch dargestellt werden, Widerspruch zur Minimalität! Da ein Kegel höchstens $\dim(K)$ linear unabhängige Vektoren enthält, folgt $d \leq \dim(K)$. Setzen wir $y_i = \lambda_i e_i$, $i = 1, \ldots, d$, so sind die Vektoren y_i Extremalen von K mit der gewünschten Eigenschaft. □

(8.29) Folgerung. *Ist $P \subseteq \mathbb{K}^n$ ein spitzes Polyeder und ist $x \in P$, dann gibt es Ecken $v_0, v_1, \ldots, v_k$ und Extremalen $e_{k+1}, e_{k+2}, \ldots, e_d$ von P mit $d \leq \dim(P)$ und nichtnegative Skalare $\lambda_0, \ldots, \lambda_k$ 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 (8.12) ist hog(P) ein spitzer Kegel, und die Dimension von hog(P) ist $\dim(P) + 1$. Nach (8.6) gilt $x \in P \iff \binom{x}{1} \in$ hog(P). Nach Satz (8.28) ist $\binom{x}{1}$ konische Kombination von $d + 1 \leq \dim(P) + 1$ Extremalen von hog(P). O. B. d. A. können wir annehmen, dass gilt

$$\begin{pmatrix} x \\ 1 \end{pmatrix} = \sum_{i=0}^{k} \begin{pmatrix} y_i \\ \lambda_i \end{pmatrix} + \sum_{i=k+1}^{d} \begin{pmatrix} e_i \\ 0 \end{pmatrix},$$

wobei $\lambda_i > 0$, $i = 0, \ldots, k$. Für $v_i := \frac{1}{\lambda_i} y_i$ gilt dann

$$x = \sum_{i=0}^{k} \lambda_i v_i + \sum_{i=k+1}^{d} e_i\,, \quad \sum_{i=0}^{k} \lambda_i = 1.$$

Ferner sind nach (8.20) die Vektoren v_i Ecken von P und die Vektoren e_i Extremalen von P. Also haben wir die gewünschte Kombination von x gefunden. □

Das folgende direkte Korollar von (8.29) wird in der Literatur häufig **Satz von Caratheodory** genannt.

(8.30) Folgerung. *Ist $P \subseteq \mathbb{K}^n$ ein Polytop, dann ist jedes Element von P Konvexkombination von höchstens $\dim(P) + 1$ Ecken von P.*

□

Kapitel 9

Die Grundversion der Simplex-Methode

In Satz (7.8) haben wir gezeigt, dass die Menge der Optimallösungen eines linearen Programms eine Seitenfläche des durch die Restriktionen definierten Polyeders ist. Dieses Ergebnis haben wir in Folgerung (8.15) dahingehend verschärft, dass wir nachweisen konnten, dass bei spitzen Polyedern das Optimum, falls es existiert, stets in einer Ecke angenommen wird. Wollen wir also ein Verfahren entwerfen, das lineare Programme über spitzen Polyedern löst, so genügt es, ein Verfahren anzugeben, dass eine optimale Ecke derartiger Polyeder findet.

Aus Kapitel 2 wissen wir, dass wir uns auf lineare Programme über Polyedern der Form $P^{=}(A,b) = \{x \in \mathbb{R}^n \mid Ax = b, x \geq 0\}$ beschränken können. In Folgerung (8.13) haben wir gezeigt, dass ein solches Polyeder stets spitz ist, falls es nicht leer ist. Wenn wir also einen Algorithmus angeben können, der die optimale Ecke eines linearen Programms der Form $\max c^T x,\ x \in P^{=}(A,b)$ findet, so können wir mit diesem alle linearen Programmierungsprobleme lösen.

(9.1) Definition. *Ein lineares Programm der Form*

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

heißt ein lineares Programm in **Standardform**. *Wenn nichts anderes gesagt wird, nehmen wir immer an, dass $A \in \mathbb{K}^{(m,n)}$, $b \in \mathbb{K}^m$, und $c \in \mathbb{K}^n$ gilt.*

□

Ist der Rang von A gleich n, so wissen wir, dass das Gleichungssystem $Ax = b$ entweder gar keine oder eine eindeutig bestimmte Lösung hat, die mit dem Gauß'schen Eliminationsverfahren berechnet werden kann. Da also entweder keine Lösung existiert oder der einzige zulässige Punkt leicht bestimmt werden kann, liegt kein wirkliches Optimierungsproblem vor. Wir wollen daher voraussetzen, dass $\text{rang}(A) < n$ gilt. Ferner nehmen wir an, um die Darstellung einfacher zu machen, dass die Zeilen von A linear unabhängig sind und dass $P^=(A, b) \neq \emptyset$ ist. Wir treffen also folgende Verabredung:

(9.2) Generalvorausssetzungen für Kapitel 9.

(a) A ist eine (m, n)-Matrix mit $m < n$,
(b) $\text{rang}(A) = m$,
(c) $P^=(A, b) \neq \emptyset$.

□

Wir werden später zeigen, dass lineare Programme, bei denen die Voraussetzungen (9.2) nicht gegeben sind, auf solche zurückgeführt werden können, die diese erfüllen, dass also alle Voraussetzungen (9.2) o. B. d. A. gemacht werden können.

Um die weitere Darstellung schreibtechnisch zu vereinfachen, legen wir für dieses Kapitel folgendes fest (vergleiche Abschnitt 1.3.):

(9.3) Konventionen in Kapitel 9. *Es sei $A \in \mathbb{K}^{(m,n)}$, wobei nach (9.2) $m < n$ gelten soll.*

(a) Mit $\{1, \dots, m\}$ bezeichnen wir die Zeilenindexmenge von A, mit $\{1, \dots, n\}$ die Spaltenindexmenge.

(b) B und N bezeichnen stets Spaltenindexvektoren von A, wobei $B = (p_1, \dots, p_m) \in \{1, \dots, n\}^m$ und $N = (q_1, \dots, q_{n-m}) \in \{1, \dots, n\}^{n-m}$ gilt. Ferner kommt kein Spaltenindex, der in B vorkommt, auch in N vor, und umgekehrt. (Um Transpositionszeichen zu sparen, schreiben wir B und N immer als Zeilenvektoren.)

(c) Wir werden aber B und N auch einfach als Mengen auffassen (wenn die Anordnung der Indizes keine Rolle spielt), dann gilt also nach (b) $B \cap N = \emptyset$, $B \cup N = \{1, \dots, n\}$. Insbesondere können wir dann $i \in B$ und $j \in N$ schreiben.

(d) Zur schreibtechnischen Vereinfachung schreiben wir im folgenden A_B und A_N statt $A_{.B}$ und $A_{.N}$.

(e) *Ist ein Spaltenindexvektor B wie oben angegeben definiert, so bezeichnet, falls nicht anders spezifiziert, $N = (q_1, \dots, q_{n-m})$ immer den Spaltenindexvektor, der alle übrigen Spaltenindizes enthält, wobei $q_i < q_j$ für $i < j$ gilt.*

□

9.1 Basen, Basislösungen, Entartung

(9.4) Definition. *Gegeben sei ein Gleichungssystem $Ax = b$ mit $A \in \mathbb{K}^{(m,n)}$, $b \in \mathbb{K}^m$ und* rang$(A) = m$. *Spaltenindexvektoren B und N seien entsprechend Konvention (9.3) gegeben.*

(a) *Ist A_B regulär, so heißt A_B* **Basismatrix** *oder kurz* **Basis** *von A, und A_N heißt* **Nichtbasismatrix** *oder* **Nichtbasis** *von A. Der Vektor $x \in \mathbb{K}^n$ mit $x_N = 0$ und $x_B = A_B^{-1}b$ heißt* **Basislösung von** $Ax = b$ **zur Basis** A_B *oder die* **zu** A_B **gehörige Basislösung** *oder kurz* **Basislösung**.

(b) *Ist A_B eine Basis, dann heißen die Variablen x_j, $j \in B$,* **Basisvariable** *und die Variablen x_j, $j \in N$,* **Nichtbasisvariable**.

(c) *Ist A_B eine Basis, dann heißen A_B und die zugehörige Basislösung x* **zulässig** *(bezüglich $P^=(A, b)$), wenn $A_B^{-1}b \geq 0$ gilt.*

(d) *Eine Basislösung x zur Basis A_B heißt* **nichtentartet** *(oder* **nichtdegeneriert***), falls $x_B = A_B^{-1}b > 0$, andernfalls heißt sie* **entartet** *(oder* **degeneriert***).*

□

Die oben eingeführten Begriffe gehören zur Standardterminologie der linearen Programmierung. Eine erste begriffliche Brücke zur Polyedertheorie schlägt der nächste Satz.

(9.5) Satz. *Seien $P = P^=(A, b) \subseteq \mathbb{K}^n$ ein Polyeder mit* rang$(A) = m < n$ *und $x \in P$. Dann sind äquivalent*

(1) *x ist eine Ecke von $P^=(A, b)$.*

(2) *x ist eine zulässige Basislösung (d. h. es gibt eine Basis A_B von A mit der Eigenschaft $x_B = A_B^{-1}b \geq 0$ und $x_N = 0$).*

Beweis : (1) $\Longrightarrow$ (2). Sei $I := \operatorname{supp}(x)$. Ist x eine Ecke von $P^=(A,b)$, so sind die Spaltenvektoren $A_{.j}$, $j \in I$, nach (8.10) linear unabhängig. Wegen $\operatorname{rang}(A) = m$ gibt es eine Menge $J \in \{1,\dots,n\} \setminus I$, $|J| = m - |I|$, so dass die Spalten $A_{.j}$, $j \in I \cup J$, linear unabhängig sind. Folglich ist A_B mit $B = I \cup J$ eine Basis von A. Nehmen wir o. B. d. A. an, dass B aus den ersten m Spalten von A besteht, also $A = (A_B, A_N)$ mit $N = \{m+1,\dots,n\}$ gilt, dann erhalten wir aus $x^T = (x_B^T, x_N^T) = (x_B^T, 0)$ folgendes: $A_B^{-1}b = A_B^{-1}(Ax) = A_B^{-1}(A_B, A_N)\binom{x_B}{0} = (I, A_B^{-1}A_N)\binom{x_B}{0} = x_B$. Da $x \geq 0$ gilt, ist x eine Basislösung.

(2) $\Longrightarrow$ (1) folgt direkt aus (8.10). $\square$

Damit sehen wir, dass Ecken von $P^=(A,b)$ zulässsigen Basislösungen von $Ax = b$, $x \geq 0$ entsprechen. Also gibt es zu jeder Ecke eine zulässige Basis von A. Sind zwei Ecken verschieden, so sind natürlich die zugehörigen Basen verschieden. Dies gilt aber nicht umgekehrt!

$$\text{Ecke} \overset{\text{eindeutig}}{\longleftrightarrow} \text{zulässige Basislösung} \overset{\text{nichteindeutig}}{\longleftrightarrow} \text{zulässige Basis.}$$

Wir wollen nun Entartung und ihre Ursachen untersuchen und betrachten zunächst drei Beispiele.

(9.6) Beispiel (Degeneration).

(a) Im $\mathbb{K}^2$ gibt es keine "richtigen" degenerierten Probleme, da man solche Probleme durch Entfernung von redundanten Ungleichungen nichtdegeneriert machen kann. Im folgenden Beispiel ist die Ungleichung $2x_1 + x_2 \leq 2$ redundant.

$$\begin{aligned} x_1 + x_2 &\leq 1 \\ 2x_1 + x_2 &\leq 2 \\ x_1, x_2 &\geq 0 \end{aligned}$$

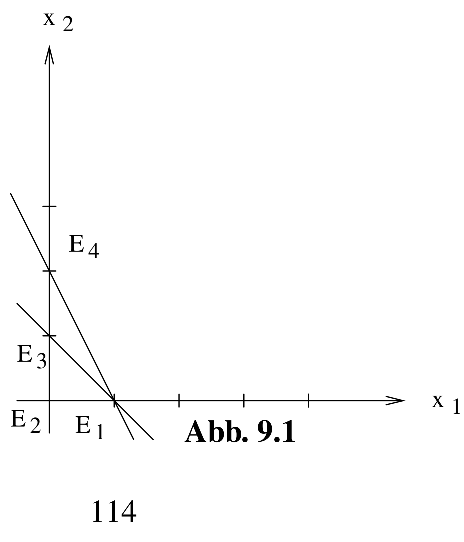


Abb. 9.1

Das durch das Ungleichungssystem definierte Polyeder (siehe Abbildung 9.1) hat die drei Ecken E_1, E_2, E_3. Wir formen durch Einführung von Schlupfvariablen um:

$$\begin{aligned} x_1 + x_2 + s_1 \quad\quad & = 1 \\ 2x_1 + x_2 \quad\quad + s_2 & = 2 \\ x_1, x_2, s_1, s_2 \quad\quad & \geq 0. \end{aligned}$$

Also ist $P^=(A, b)$ gegeben durch:

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

Die folgende Tabelle 9.1 enthält die Basen von A und die zugehörigen Basislösungen.

Spalten in B	A_B	x_B	x_N	x	Ecke
1,2	$\begin{pmatrix} 1 & 1 \\ 2 & 1 \end{pmatrix}$	$(x_1, x_2) = (1, 0)$	(s_1, s_2)	$(1, 0, 0, 0)$	E_1
1,3	$\begin{pmatrix} 1 & 1 \\ 2 & 0 \end{pmatrix}$	$(x_1, s_1) = (1, 0)$	(x_2, s_2)	$(1, 0, 0, 0)$	E_1
1,4	$\begin{pmatrix} 1 & 0 \\ 2 & 1 \end{pmatrix}$	$(x_1, s_2) = (1, 0)$	(x_2, s_1)	$(1, 0, 0, 0)$	E_1
2,3	$\begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix}$	$(x_2, s_1) = (2, -1)$	(x_1, s_2)	$(0, 2, -1, 0)$	E_4
2,4	$\begin{pmatrix} 1 & 0 \\ 1 & 1 \end{pmatrix}$	$(x_2, s_2) = (1, 1)$	(x_1, s_1)	$(0, 1, 0, 1)$	E_3
3,4	$\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$	$(s_1, s_2) = (1, 1)$	(x_1, x_2)	$(0, 0, 1, 1)$	E_2

Tabelle 9.1

Im obigen Falle ist jede $(2, 2)$-Untermatrix von A eine Basis. Dies ist natürlich nicht immer so! Zur Ecke E_1 gehören drei verschiedene Basen. Hier liegt also Entartung vor. Alle anderen Basislösungen sind nichtdegeneriert. Die zu $B = (2, 3)$ gehörige Basis ist unzulässig. Zu unzulässigen Basislösungen sagt man manchmal auch unzulässige Ecke von $P^=(A, b)$, E_4 ist also eine unzulässige Ecke.

(b) Bei dreidimensionalen Polyedern im $\mathbb{K}^3$ tritt echte Entartung auf. So ist z. B. die Spitze der folgenden Pyramide entartet, während alle übrigen vier Ecken nichtentartet sind.

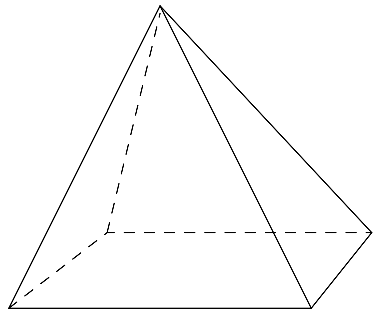


Abb. 9.2

(c) Dagegen sind alle Ecken der obigen Doppelpyramide entartet.

$\square$

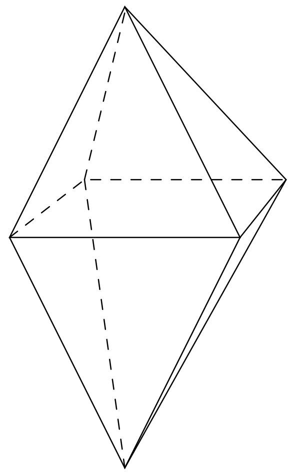


Abb. 9.3

Die Sätze (9.5) und (8.10) (und ihre Beweise) sagen uns, wie Ecken von $P^{=}(A, b)$ mit den Basen von A_B zusammenhängen. Aus der linearen Algebra wissen wir zunächst, dass jeder Punkt x im $\mathbb{K}^n$ als Durchschnitt von n Hyperebenen dargestellt werden kann, deren Normalenvektoren linear unabhängig sind. Algebraisch ausgedrückt, jeder Punkt $x \in \mathbb{K}^n$ ist eindeutig bestimmte Lösung eines regulären (n, n)-Gleichungssystems $Dy = d$ (mit $\text{rang}(D) = n$).

Ecken von $P^=(A, b)$ erhalten wir dadurch, dass wir nur Lösungen von regulären Gleichungssystemen

$$\begin{aligned} Ay &= b \\ e_j^T y &= 0, \quad j \in J \end{aligned}$$

mit $J \subseteq \{1, \ldots, n\}$ betrachten (und prüfen, ob die Lösungen in $P^=(A, b)$ enthalten sind). Dabei kann es zu folgenden Situationen kommen:

Ist A_B eine Basis von A mit $(A_B^{-1}b)_i \neq 0$ für alle $i \in B$, so gibt es nur eine einzige Möglichkeit das System $Ay = b$ durch ein System $e_j^T y = 0$, $j \in J$, so zu ergänzen, dass das Gesamtsystem regulär wird. Man muss $J = \{1, \ldots, n\} \setminus B$ wählen, andernfalls sind nicht genügend Gleichungen vorhanden. Daraus folgt speziell:

(9.7) Bemerkung. Ist x eine nichtdegenerierte Basislösung von $Ay = b$, $y \geq 0$, dann gibt es eine eindeutig bestimmte zu x gehörige Basis A_B von A ($x_B = A_B^{-1}b$, $x_N = 0$).

$\square$

Die Umkehrung gilt i. a. nicht, wie das folgende Beispiel zeigt.

(9.8) Beispiel. $P^=(A, b) \subseteq \mathbb{R}^3$ sei gegeben durch

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

Die Matrix A besitzt zwei Basen A_{B_1}, A_{B_2} mit $B_1 = (1, 3)$, $B_2 = (2, 3)$. Die zu A_{B_1} gehörige Basislösung ist $x_1^T = (1, 0, 0)$, die zu A_{B_2} gehörige Basislösung ist $x_2^T = (0, 1, 0)$. Beide Basislösungen sind entartet, aber die zugehörigen Basen sind eindeutig bestimmt. $\square$

Ist A_B eine Basis von A und x die zugehörige Basislösung, so dass einige Komponenten von $A_B^{-1}b$ Null sind (d. h. x ist entartet), so enthält das Gleichungssystem $Ay = b$, $e_j^T y = 0$ für alle $j \in J = \{j \mid x_j = 0\}$ mehr als n Gleichungen (also mehr als notwendig). I. a. ist dann x Lösung von mehreren regulären (n, n)-Untersystemen dieses Gleichungssystems. Man hat also keine eindeutige Zuordnung mehr von der Ecke x zu einem Gleichungssystem der Form $Ay = b$, $e_j^T y = 0$ $j \in J$. Wir werden sehen, dass Entartung zu rechentechnischen Problemen führt. Entartung kann drei Ursachen haben

- überflüssige Variable (im Beispiel (9.8) kann x_3 weggelassen werden, dann sind alle Basislösungen nichtentartet),

– redundante Ungleichungen (Beispiel (9.6) (a)),

– geometrische Gründe (Beispiele (9.6) (b) und (c)). Die durch überflüssige Variablen oder redundante Ungleichungen hervorgerufene Entartung kann i. a. beseitigt werden durch Weglassen der überflüssigen Variablen bzw. der redundanten Ungleichungen. Es gibt jedoch Polyeder — wie z. B. die Doppelpyramide in (9.6) (c) — die genuin entartet sind. Der Grund liegt hier darin, dass die Ecken überbestimmt sind, d. h. es gibt Ecken x von P, in denen mehr als $\dim(P)$ Facetten zusammentreffen. Dann bestimmen je $\dim(P)$ der diese Facetten definierenden Ungleichungen eine Basis von A, die x als zugehörige Basislösung liefert.

9.2 Basisaustausch (Pivoting), Simplexkriterium

Da jede zulässige Basis der Matrix A eine Ecke von $P^{=}(A, b)$ definiert, kann man ein lineares Programm der Form (9.1) dadurch lösen, dass man alle Basen von A bestimmt — dies sind endlich viele —, den Zielfunktionswert der zugehörigen Basislösung errechnet und die beste Lösung auswählt. Da eine (m, n)-Matrix bis zu $\binom{n}{m}$ zulässige Basen haben kann, ist dieses Verfahren aufgrund des gewaltigen Rechenaufwandes (in jedem Schritt Bestimmung einer inversen Matrix) so gut wie undurchführbar.

Man sollte also versuchen, z. B. ein Verfahren so zu entwerfen, dass man von einer zulässigen Basislösung ausgehend eine neue Basislösung bestimmt, die zulässig ist und einen besseren Zielfunktionswert hat. Ferner muss das Verfahren so angelegt sein, dass der Übergang von einer Basis zur nächsten nicht allzuviel Rechenaufwand erfordert.

Im Prinzip kann man damit nicht gewährleisten, dass nicht alle Basen enumeriert werden, aber aufgrund heuristischer Überlegungen scheint ein solcher Ansatz, wenn er realisiert werden kann, besser als das obige Enumerationsverfahren funktionieren zu können. Im weiteren werden wir zeigen, wie man einen Basisaustausch, der von einer zulässigen Basislösung zu einer besseren zulässigen Basislösung mit relativ wenig Rechenaufwand führt, durchführen kann.

(9.9) Satz. *Gegeben sei ein Gleichungssystem $Ay = b$ mit $\operatorname{rang}(A) = m$, und A_B sei eine Basis von A. Dann gilt*

$$x \text{ erfüllt } Ay = b \iff x \text{ erfüllt } x_B = A_B^{-1}b - A_B^{-1}A_N x_N.$$

Beweis : O. B. d. A. sei $B = (1, \dots, m)$, d. h. $A = (A_B, A_N)$, $x = \begin{pmatrix} x_B \\ x_N \end{pmatrix}$, dann gilt

$$
\begin{aligned}
Ax = b \iff (A_B, A_N)\begin{pmatrix} x_B \\ x_N \end{pmatrix} = b &\iff A_B^{-1}(A_B, A_N)\begin{pmatrix} x_B \\ x_N \end{pmatrix} = A_B^{-1}b \\
&\iff (I, A_B^{-1}A_N)\begin{pmatrix} x_B \\ x_N \end{pmatrix} = A_B^{-1}b \\
&\iff x_B + A_B^{-1}A_N x_N = A_B^{-1}b \\
&\iff x_B = A_B^{-1}b - A_B^{-1}A_N x_N
\end{aligned}
$$

□

Die obige Beziehung ist simpel, von rechentechnischer Bedeutung ist jedoch die Tatsache, dass wir die Basisvariablen x_j, $j \in B$, in Abhängigkeit von den Nichtbasisvariablen darstellen können.

Wir wollen uns nun überlegen, wie wir den Übergang von einer Ecke zu einer anderen Ecke vollziehen können, d. h. wie wir aus einer Basis eine andere Basis ohne viel Aufwand konstruieren können. Um dies rechentechnisch exakt vorführen zu können, müssen wir die Indexmengen B, N wie in (9.3) (b) angegeben als Vektoren auffassen.

(9.10) Satz *(Basisaustausch, Pivotoperation). Gegeben sei ein Gleichungssystem $Ay = b$ mit $\operatorname{rang}(A) = m$. $B = (p_1, \dots, p_m)$ und $N = (q_1, \dots, q_{n-m})$ seien Spaltenindexvektoren von A wie in (9.3) (b) festgelegt, so dass A_B eine Basis von A ist. Wir setzen*

$$
\begin{aligned}
\overline{A} &:= A_B^{-1}A_N = (\overline{a}_{rs})_{\substack{1 \le r \le m \\ 1 \le s \le n-m}}, \\
\overline{b} &:= A_B^{-1}b \in \mathbb{K}^m.
\end{aligned}
$$

Ist $\overline{a}_{rs} \neq 0$, so ist $A_{B'}$ mit $B' := (p_1, \dots, p_{r-1}, q_s, p_{r+1}, \dots, p_m)$ eine Basis von A. Ferner gilt:

$$A_{B'}^{-1} = EA_B^{-1},$$

*wobei E eine sogenannte (m, m)-**Elementarmatrix** ist, die wie folgt definiert ist:*

$$
E = \begin{pmatrix}
1 & & \eta_1 & & \\
& \ddots & \vdots & & \\
& & 1\eta_{r-1} & & \\
& & \eta_r & & \\
& & \eta_{r+1}1 & & \\
& & \vdots & \ddots & \\
& & \eta_m & & 1
\end{pmatrix}.
$$

Dabei ist die r-te Spalte von E der Vektor $\eta = (\eta_1, \ldots, \eta_m)^T$ (genannt **Eta-Spalte***) gegeben durch*

$$\begin{aligned} \eta_r &:= \tfrac{1}{\overline{a}_{rs}} \\ \eta_i &:= -\tfrac{\overline{a}_{is}}{\overline{a}_{rs}} \quad i \in \{1, \ldots, m\} \setminus \{r\}. \end{aligned}$$

Das Element $\overline{a}_{rs}$ heißt **Pivot-Element***.*

Sind x, x' die zu A_B, A'_B gehörigen Basislösungen, so gilt:

$$\begin{aligned} x'_{p_i} &= \overline{b}_i - \frac{\overline{a}_{is}}{\overline{a}_{rs}}\overline{b}_r = E_i.\overline{b} = E_i.x_B,\ 1 \le i \le m,\ i \ne r && (neue Basisvariable) \\ x'_{q_s} &= \frac{1}{\overline{a}_{rs}}\overline{b}_r\ (= E_r.\overline{b}_r = E_r.x_B) && (neue Basisvariable) \\ x'_j &= 0 \ \text{ andernfalls} && (neue Nichtbasisvariable). \end{aligned}$$

Beweis : Es seien

$$d := A_{.q_s},\ \overline{d} := A_B^{-1}d = \overline{A}_{.s}, F := A_B^{-1}A_{B'}.$$

$A_{B'}$ erhält man aus A_B dadurch, dass die r-te Spalte $A_{.p_r}$ von A_B durch d ersetzt wird. F ist daher eine (m, m)-Elementarmatrix, deren r-te Spalte der Vektor $\overline{d}$ ist. Offenbar gilt daher

$$F \cdot E = \begin{pmatrix} 1 & & & \overline{d}_1 & & & \\ & \ddots & & \vdots & & & \\ & & 1 & \overline{d}_{r-1} & & & \\ & & & \overline{d}_r & & & \\ & & & \overline{d}_{r+1} & 1 & & \\ & & & \vdots & & \ddots & \\ & & & \overline{d}_m & & & 1 \end{pmatrix} \begin{pmatrix} 1 & & & \eta_1 & & & \\ & \ddots & & \vdots & & & \\ & & 1 & \eta_{r-1} & & & \\ & & & \eta_r & & & \\ & & & \eta_{r+1} & 1 & & \\ & & & \vdots & & \ddots & \\ & & & \eta_m & & & 1 \end{pmatrix} = I,$$

daraus folgt $E = F^{-1}$ ($\overline{d}_r \ne 0$ war vorausgesetzt). Da F und A_B invertierbar sind, ergibt sich aus $A_{B'} = A_B F$ sofort $A_{B'}^{-1} = F^{-1}A_B^{-1} = EA_B^{-1}$. Die übrigen Formeln ergeben sich durch einfache Rechnung. □

Bei dem in Satz (9.10) beschriebenen Basisaustausch wird also von einer Basis zu einer anderen Basis übergegangen, indem eine Nichtbasisvariable x_{q_s} und eine Basisvariable x_{p_r} gesucht werden, so dass $\overline{a}_{rs} \ne 0$ gilt. Wenn diese Bedingung erfüllt ist, kann die Nichtbasisvariable x_{q_s} zu einer Basisvariablen gemacht werden, wobei x_{p_r} nichtbasisch wird. Alle übrigen Basisvariablen bleiben erhalten.

(9.11) Beispiel. Wir betrachten das Gleichungssystem $Ay = b$ gegeben durch

$$A = \begin{pmatrix} 1 & 0 & 2 & -\frac{3}{2} & -4 & \frac{11}{2} & \frac{11}{2} \\ 0 & 1 & 0 & 0 & 3 & 0 & 2 \\ 0 & 0 & -1 & \frac{1}{2} & 2 & -\frac{3}{2} & -\frac{1}{2} \\ 0 & 0 & 0 & \frac{1}{2} & 1 & -\frac{1}{2} & -\frac{1}{2} \end{pmatrix} \quad b = \begin{pmatrix} 1 \\ 2 \\ -1 \\ 2 \end{pmatrix}.$$

Es seien $B = (1,2,3,4)$, $N = (5,6,7)$, dann ist A_B eine Basis von A, und es gilt

$$A_B^{-1} = \begin{pmatrix} 1 & 0 & 2 & 1 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & -1 & 1 \\ 0 & 0 & 0 & 2 \end{pmatrix}, \quad \overline{A} = A_B^{-1} A_N = \begin{pmatrix} 1 & 2 & 4 \\ 3 & 0 & 2 \\ -1 & 1 & 0 \\ 2 & -1 & -1 \end{pmatrix}.$$

Die zugehörige Basislösung ist $x_B^T = (1,2,3,4)$, $x_N^T = (0,0,0)$, d. h. $x^T = (1,2,3,4,0,0,0)$.

Das Element $\overline{a}_{23} = 2$ ist von Null verschieden. Wir können es also als Pivotelement $\overline{a}_{rs}$ ($r = 2$, $s = 3$) wählen. Daraus ergibt sich:

$$B' = (1,7,3,4), \; N' = (5,6,2)$$

$$E = \begin{pmatrix} 1 & -2 & 0 & 0 \\ 0 & \frac{1}{2} & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & \frac{1}{2} & 0 & 1 \end{pmatrix}, \quad EA_B^{-1} = A_{B'}^{-1} = \begin{pmatrix} 1 & -2 & 2 & 1 \\ 0 & \frac{1}{2} & 0 & 0 \\ 0 & 0 & -1 & 1 \\ 0 & \frac{1}{2} & 0 & 2 \end{pmatrix}$$

und

$$A_{B'}^{-1} A = EA_B^{-1} A = \begin{pmatrix} 1 & -2 & 0 & 0 & -5 & 2 & 0 \\ 0 & \frac{1}{2} & 0 & 0 & \frac{3}{2} & 0 & 1 \\ 0 & 0 & 1 & 0 & -1 & 1 & 0 \\ 0 & \frac{1}{2} & 0 & 1 & \frac{7}{2} & -1 & 0 \end{pmatrix}, \quad \overline{A}' = \begin{pmatrix} -5 & 2 & -2 \\ \frac{3}{2} & 0 & \frac{1}{2} \\ -1 & 1 & 0 \\ \frac{7}{2} & -1 & \frac{1}{2} \end{pmatrix}.$$

Die zugehörige Basislösung ist $x_{B'}^T = (-3,1,3,5)$, $x_{N'}^T = (0,0,0)$, daraus ergibt sich $x^T = (-3,0,3,5,0,0,1)$.

□

(9.12) Bemerkung. Die neue Basisinverse $A_{B'}^{-1}$ kann also wegen $A_{B'}^{-1} = EA_B^{-1}$ leicht aus A_B^{-1} berechnet werden. Tatsächlich ist jedoch dieses "Pivoting" der numerisch aufwendigste und schwierigste Schritt im Simplexverfahren. Es gibt hier zahllose "Update"-Varianten, von denen wir einige (Produktform der Inversen, LU-Dekomposition, Cholesky-Zerlegung) später bzw. in den Übungen noch kurz besprechen wollen, durch die eine schnelle und numerisch stabile Berechnung von $A_{B'}^{-1}$ erreicht werden soll. Insbesondere für große und dünn besetzte Matrizen $(n, m \geq 5000)$ (large-scale-linear-programming) gibt es spezielle Techniken.

$\square$

Die äquivalente Darstellung des Gleichungssystems $Ax = b$ in Bemerkung (9.9) gibt uns ebenso die Möglichkeit, die Kosten auf einfache Weise über die Nichtbasisvariablen zu berechnen, woraus sich ein sehr einfaches Optimalitätskriterium gewinnen lässt.

(9.13) Satz. *Gegeben sei ein lineares Programm in Standardform (9.1), und A_B sei eine Basis von A.*

(a) *Für den Zielfunktionswert $c^T x$ von $x \in P^=(A, b)$ gilt*

$$c^T x = c_B^T A_B^{-1} b + (c_N^T - c_B^T A_B^{-1} A_N) x_N.$$

Der Term $\overline{c}^T := c_N^T - c_B^T A_B^{-1} A_N$ heißt **reduzierte Kosten** *(von x), die Komponenten $\overline{c}_i$ von $\overline{c}$ heißen* **reduzierte Kostenkoeffizienten**.

(b) *(Simplexkriterium)*
Ist A_B eine zulässige Basis und sind die reduzierten Kosten nicht-positiv, d. h.

$$\overline{c}^T = c_N^T - c_B^T A_B^{-1} A_N \leq 0,$$

dann ist die zugehörige Basislösung x mit $x_B = A_B^{-1} b$, $x_N = 0$ optimal für (9.1).

(c) *Ist A_B eine zulässige Basis, und ist die zugehörige Basislösung nichtdegeneriert und optimal, dann gilt $\overline{c} \leq 0$.*

Beweis :

(a) Nach (9.9) gilt $Ax = b \iff x_B = A_B^{-1} b - A_B^{-1} A_N x_N$ und damit gilt

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

(b) Sei $y \in P^=(A,b)$ beliebig. Wenn wir zeigen können, dass $c^T x \geq c^T y$ gilt, sind wir fertig.

$$c^T y = c_B^T A_B^{-1} b + \underbrace{\overline{c}^T}_{\leq 0} \underbrace{y_N}_{\geq 0} \leq c_B^T A_B^{-1} b = c_B^T x_B = c^T x.$$

(c) Sei die Basislösung x mit $x_B = A_B^{-1} b$, $x_N = 0$ optimal. Dann gilt für alle $y \in P^=(A,b)$:

$$\begin{aligned} c^T x \geq c^T y \iff & c_B^T A_B^{-1} b + \overline{c}^T x_N \geq c_B^T A_B^{-1} b + \overline{c}^T y_N \\ \iff & \qquad 0 \geq \overline{c}^T y_N \,. \end{aligned}$$

Angenommen die i-te Komponente $\overline{c}_i$ von $\overline{c}^T$ wäre größer als Null. Nach Voraussetzung ist x nichtdegeneriert, also $A_B^{-1} b > 0$. Sei e_i der i-te Einheitsvektor in $\mathbb{K}^{n-m}$, dann gibt es somit ein $\lambda > 0$ mit

$$A_B^{-1} b \geq A_B^{-1} A_N \lambda e_i.$$

Der Vektor x^λ mit $x_B^\lambda = A_B^{-1} b - A_B^{-1} A_N \lambda e_i$, $x_N = \lambda e_i$ ist somit ein für (9.1) zulässiger Vektor, und es gilt

$$c^T x^\lambda = c_B^T A_B^{-1} b + \overline{c}^T \lambda e_i = c^T x + \lambda \overline{c}_i > c^T x.$$

Widerspruch! □

Aus Satz (9.10) wissen wir, wie ein Basis- bzw. Eckenaustausch vorgenommen werden kann, Satz (9.13) besagt, unter welchen Umständen wir eine zulässige Basis verbessern können, bzw. wann sie optimal ist. Setzen wir diese Informationen zusammen, so gilt:

(9.14) Satz *(Basisverbesserung). Gegeben sei ein lineares Programm in Standardform (9.1). Sei A_B eine zulässige Basis mit Basislösung x. Sei $\overline{A} = A_B^{-1} A_N$, $\overline{b} = A_B^{-1} b$, und $\overline{c}^T = c_N^T - c_B^T A_B^{-1} A_N$ seien die reduzierten Kosten. Sei $q_s \in N$ ein Index mit $\overline{c}_s > 0$, dann gilt*

(a) Ist $\overline{A}_{.s} \leq 0$, dann ist $c^T x$ auf $P^=(A,b)$ unbeschränkt.

(b) Ist $\overline{A}_{.s} \not\leq 0$, dann setzen wir

$$\lambda_0 := \min\Big\{ \frac{\overline{b}_i}{\overline{a}_{is}} \mid i = 1, \ldots, m, \overline{a}_{is} > 0 \Big\},$$

und wählen einen Index

$$r \in \Big\{ i \in \{1, \dots, m\} \mid \frac{\overline{b}_i}{\overline{a}_{is}} = \lambda_0 \Big\}.$$

Dann ist $A_{B'}$ *mit* $B' = (p_1, \dots, p_{r-1}, q_s, p_{r+1}, \dots, p_m)$ *eine zulässige Basis mit Basislösung* x'*, so dass* $c^T x' \geq c^T x$ *gilt.*

(c) *Gelten die Voraussetzungen von (b), und ist* A_B *nichtdegeneriert, dann gilt* $c^T x' > c^T x$.

Beweis : (a) Nach (9.9) gilt: $y \in P^=(A,b) \iff y_B = A_B^{-1} b - A_B^{-1} A_N y_N \geq 0$ und $y_N \geq 0$. Für beliebiges $\lambda \geq 0$ ist daher der Vektor $y^\lambda \in \mathbb{K}^n$ mit $y_N^\lambda := \lambda e_s$ und $y_B^\lambda := A_B^{-1} b - \overline{A} y_N^\lambda = A_B^{-1} b - \lambda A_{.s}$ wegen $e_s \geq 0$ und $A_B^{-1} b - \lambda A_{.s} \geq A_B^{-1} b \geq 0$ in $P^=(A,b)$. Da $c^T y^\lambda = c_B^T A_B^{-1} b + \overline{c} y_N^\lambda = c_B^T A_B^{-1} b + \lambda \overline{c}_s$ mit λ über alle Schranken wächst, ist $c^T x$ auf $P^=(A,b)$ unbeschränkt.

(b) Wählen wir r und s wie im Satz angegeben, so ist $\overline{a}_{rs} > 0$. Nach Satz (9.10) ist dann $B' = (p_1, \dots, p_{r-1}, q_s, p_{r+1}, \dots, p_m)$ eine Basis mit Basislösung x', wobei

$$\begin{aligned}
x'_{p_i} &= \overline{b}_i - \frac{\overline{b}_r}{\overline{a}_{rs}} \overline{a}_{is} \begin{cases} \geq \overline{b}_i - \dfrac{\overline{b}_i}{\overline{a}_{is}} \overline{a}_{is}, & \text{falls} \quad \overline{a}_{is} > 0, i \neq r \quad \text{(Wahl von } \lambda_0\text{!)}, \\ \geq \overline{b}_i, & \text{falls} \quad \overline{a}_{is} \leq 0, \end{cases} \\
x'_{q_s} &= \frac{\overline{b}_r}{\overline{a}_{rs}} \geq 0, \\
x'_i &= 0 \quad \text{andernfalls.}
\end{aligned}$$

Also ist x' eine zulässige Basislösung.

Wir zeigen nun $c^T x' \geq c^T x$. Zunächst gilt für den in Satz (9.10) definierten Vektor η:

$$c_{B'}^T \eta - c_{p_r} = \frac{\overline{c}_s}{\overline{a}_{rs}} > 0,$$

denn

$$\begin{aligned}
0 < \overline{c}_s &= (c_N^T - c_B^T A_B^{-1} A_N)_s = c_{q_s} - c_B^T \overline{A}_{.s} = c_{q_s} - \textstyle\sum_{i=1}^m c_{p_i} \overline{a}_{is} \\
&= (c_{q_s} - \textstyle\sum_{i \neq r} c_{p_i} \overline{a}_{is}) - c_{p_r} \overline{a}_{rs} = \overline{a}_{rs} (c_{B'}^T \eta - c_{p_r}),
\end{aligned}$$

und aus $\overline{a}_{rs} > 0$ folgt nun die Behauptung. Somit gilt:

$$\begin{aligned}
c^T x' &= c_{B'}^T x'_{B'} = c_{B'}^T A_{B'}^{-1} b \overset{(9.10)}{=} c_{B'}^T E A_B^{-1} b = c_{B'}^T E x_B = \\
&= \sum_{i \neq r} c_{p_i} x_{p_i} + c_{B'}^T \eta x_{p_r} \\
&= c_B^T x_B + (c_{B'}^T \eta - c_{p_r}) x_{p_r} \\
&\geq c_B^T x_B = c^T x.
\end{aligned}$$

Hieraus folgt (b). (c) Ist x nichtdegeneriert, so gilt $x_{p_r} > 0$, und somit ist die letzte Ungleichung in der obigen Abschätzung strikt. Daraus ergibt sich (c). □

Der obige Beweis macht den geometrischen Inhalt des Satzes (9.14) nicht deutlich. Dem Leser wird dringend empfohlen, sich zu überlegen, welche geometrische Bedeutung der Parameter λ_0 in (b) hat und welche Beziehung $\overline{A}_{\cdot s}$ im Falle $\overline{A}_{\cdot s} \leq 0$ zum Rezessionskegel von $P^=(A, b)$ hat.

9.3 Das Simplexverfahren

Die Grundidee des Simplexverfahrens haben wir schon erläutert, sie kann kurz wie folgt beschrieben werden:

- Finde eine zulässige Basis A_B.
- Konstruiere aus A_B eine zulässige Basis $A_{B'}$, so dass die zulässige Basislösung von $A_{B'}$ besser als die von A_B ist.
- Gibt es keine bessere Basislösung, so ist die letzte optimal.

Die vorhergehenden Sätze geben uns nun die Möglichkeit, die oben informell beschriebene Idee, analytisch darzustellen und algorithmisch zu realisieren. Die Grundversion des Simplexverfahrens lautet dann wie folgt:

(9.15) Grundversion des Simplexverfahrens.

Input: $A' \in \mathbb{K}^{(m,n)}$, $b' \in \mathbb{K}^m$, $c \in \mathbb{K}^n$

Output:

Eine optimale Lösung x des linearen Programms

$$
(9.1) \qquad \begin{aligned} &\max c^T x \\ &A'x = b' \\ &x \geq 0, \end{aligned}
$$

falls eine solche existiert. Andernfalls stellt das Verfahren fest, dass (9.1) unbeschränkt ist oder keine zulässige Lösung besitzt.

Phase I: *(Test auf Zulässigkeit, Bestimmung einer zulässigen Basislösung)*

Bestimme ein Subsystem $Ax = b$, $x \geq 0$ von $A'x = b'$, $x \geq 0$, so dass $P^=(A,b) = P^=(A',b')$ gilt und die Zusatzvoraussetzungen $\operatorname{rang}(A) =$ Zeilenzahl von $A <$ Spaltenzahl von A erfüllt sind (falls möglich). Sei $\operatorname{rang}(A) = m < n$. Bestimme weiterhin eine zulässige Basis, d. h. finde $B = (p_1, \ldots, p_m) \in \{1, \ldots, n\}^m$, so dass A_B eine zulässige Basis von A ist, und sei $N = (q_1, \ldots, q_{n-m})$ wie üblich definiert. Berechne

$$
\begin{aligned}
&A_B^{-1} \\
&\overline{A} &&= A_B^{-1} A_N, \\
&\overline{b} &&= A_B^{-1} b, \\
&\overline{c}^T &&= c_N^T - c_B^T A_B^{-1} A_N, \\
&c_0 &&= c_B^T \overline{b}.
\end{aligned}
$$

Wie die gesamte Phase I durchgeführt wird und wie zu verfahren ist, wenn die gewünschten Resultate nicht erzielt werden können, wird in (9.24) angegeben. Im weiteren gehen wir davon aus, dass Phase I erfolgreich war.

Phase II: *(Optimierung)*

(II.1) *(Optimalitätsprüfung)*

Gilt $\overline{c}_i \leq 0$ für $i = 1, \ldots, n-m$, so ist die gegenwärtige Basislösung optimal (siehe (9.13) (b)). Gib den Vektor x mit $x_B = \overline{b}$, $x_N = 0$ aus. Der Optimalwert von (9.1) ist $c^T x = c_B^T \overline{b} = c_0$. STOP!

Falls $\overline{c}_i > 0$ für ein $i \in \{1, \ldots, n-m\}$ gehe zu (II.2).

(II.2) *(Bestimmung der Austausch- oder Pivotspalte)*

Wähle einen Index $s \in \{1, \dots, n-m\}$, so dass $\overline{c}_s > 0$ gilt. (Zur konkreten Realisierung dieser Auswahl, falls mehrere reduzierte Kostenkoeffizienten positiv sind, gibt es sehr viele verschiedene Varianten, die wir in Abschnitt 10.2 besprechen werden.)

(II.3) *(Prüfung auf Beschränktheit des Optimums)*

Gilt $\overline{A}_{\cdot s} \leq 0$, so ist das lineare Programm unbeschränkt (siehe (9.14) (a)), STOP!

Falls $\overline{a}_{is} > 0$ für ein $i \in \{1, \dots, m\}$, gehe zu (II.4).

(II.4) *(Berechne)*

$$\lambda_0 := \min\left\{ \frac{\overline{b}_i}{\overline{a}_{is}} \mid \overline{a}_{is} > 0, i = 1, \dots, m \right\}$$

(II.5) *(Bestimmung der Austausch- oder Pivotzeile)*

Wähle einen Index $r \in \{1, \dots, m\}$, so dass gilt

$$\frac{\overline{b}_r}{\overline{a}_{rs}} = \lambda_0.$$

(Hierzu gibt es verschiedene Möglichkeiten, die wir in 10.2 erläutern werden.)

(II.6) *(Pivotoperation, Basisaustausch)*

Setze

$$\begin{array}{ll} B' & := (p_1, \dots, p_{r-1}, q_s, p_{r+1}, \dots, p_m), \\ N' & := (q_1, \dots, q_{s-1}, p_r, q_{s+1}, \dots, q_{n-m}), \\ A_{B'}^{-1} & := E A_B^{-1} \ \text{(siehe (9.10)).} \end{array}$$

(II.7) *(Updating)*

Berechne $\overline{A}$, $\overline{b}$, $\overline{c}$, und c_0 neu!

(Hierzu gibt es viele numerische Varianten. Wir geben nachfolgend ein didaktisch klares Verfahren an, das aber numerisch umständlich und höchstens für Handrechnung und kleine lineare Programme geeignet ist. Bessere Methoden werden in den Übungen besprochen.)

(a) *Neuberechnung von $\overline{A}$:*

Setze $\overline{\overline{a}}_{rs} := \frac{1}{\overline{a}_{rs}}$.

Für $i = 1, \ldots, m$, $i \neq r$ *führe aus*

$$\overline{\overline{a}}_{is} := -\overline{\overline{a}}_{rs}\overline{a}_{is}.$$

Für $j = 1, \ldots, n-m$, $j \neq s$ *führe aus*

$$\overline{\overline{a}}_{rj} := \overline{\overline{a}}_{rs}\overline{a}_{rj}.$$

Für $j = 1, \ldots, n-m$, $j \neq s$, *und* $i = 1, \ldots, m$, $i \neq r$ *führe aus*

$$\overline{\overline{a}}_{ij} := \overline{a}_{ij} - \frac{\overline{a}_{is}}{\overline{a}_{rs}}\overline{a}_{rj} = \overline{a}_{ij} + \overline{\overline{a}}_{is}\overline{a}_{rj}.$$

(b) *Neuberechnung von $\overline{b}$*

$$\overline{\overline{b}}_r := \frac{\overline{b}_r}{\overline{a}_{rs}}.$$

Für $i = 1, \ldots, m$, $i \neq r$ *führe aus*

$$\overline{\overline{b}}_i := \overline{b}_i - \frac{\overline{a}_{is}}{\overline{a}_{rs}}\overline{b}_r.$$

(c) *Neuberechnung von $\overline{c}$ und c_0.*

Setze für $j = 1, \ldots, n-m$, $j \neq s$

$$\overline{\overline{c}}_j := \overline{c}_j - \frac{\overline{a}_{rj}}{\overline{a}_{rs}}\overline{c}_s,\ \overline{\overline{c}}_s := -\frac{\overline{c}_s}{\overline{a}_{rs}},\ c_0 := c_0 + \overline{c}_s\frac{\overline{b}_r}{\overline{a}_{rs}}.$$

(d) *Setze* $\overline{A} := \overline{\overline{A}}$, $\overline{b} := \overline{\overline{b}}$, $\overline{c} := \overline{\overline{c}}$ *und gehe zu (II.1).*

□

(9.16) Die Tableauform der Simplexmethode. Zur Veranschaulichung der Simplexmethode (nicht zur Verwendung in kommerziellen Codes) benutzt man manchmal ein Tableau T. Das Tableau entspricht dem Gleichungssystem

$$\begin{pmatrix} c^T \\ A \end{pmatrix} x = \begin{pmatrix} c^T x \\ b \end{pmatrix}$$

bzw. bei gegebener Basis A_B von A und geeigneter Permutation der Spalten dem System

$$\begin{array}{|c|c|}\hline c_N^T - c_B^T A_B^{-1} A_N & 0 \\ \hline A_B^{-1} A_N & I \\ \hline \end{array} \begin{pmatrix} x_N \\ x_B \end{pmatrix} = \begin{array}{|c|}\hline c^T x - c_B^T A_B^{-1} b \\ \hline A_B^{-1} b \\ \hline \end{array}$$

□

(9.17) Definition. *Ist* $\max\{c^T x \mid Ax = b, x \geq 0\}$ *ein LP in Standardform und ist* A_B *eine zulässige Basis von* A*, so heißt die* $(m+1, n+1)$*-Matrix*

$$T_B := \begin{pmatrix} c^T - c_B^T A_B^{-1} A, & -c_B^T A_B^{-1} b \\ A_B^{-1} A, & A_B^{-1} b \end{pmatrix}$$

ein **Simplextableau zur Basis** B *mit Zeilen* $i = 0, 1, \ldots, m$ *und Spalten* $j = 1, \ldots, n+1$.

□

Da ein Simplextableau stets eine volle (m, m)-Einheitsmatrix enthält, ist eine solche Darstellung für den Rechner redundant. Zum Kennenlernen der Simplexmethode und zur Übung der Rechentechnik ist sie allerdings sehr nützlich.

(9.18) Update Formeln für ein Simplextableau. *Sei* T_B *ein Simplextableau zur Basis* B*. Ist* $q_s \in N$*, so daß* $B' = (p_1, \ldots, p_{r-1}, q_s, p_{r+1}, \ldots, p_m)$ *wieder eine zulässige Basis ist, so gelten für das Simplextableau* $T_{B'}$ *folgende Transformationsregeln*

$$(a) \qquad \begin{aligned} (T_{B'})_{i.} &= (T_B)_{i.} - \tfrac{\overline{a}_{is}}{\overline{a}_{rs}} (T_B)_{r.}, \quad \text{für } i = 0, \ldots, m,\ i \neq r, \\ (T_{B'})_{r.} &= \tfrac{1}{\overline{a}_{rs}} (T_B)_{r.} \end{aligned}$$

oder

$$(b) \qquad T_{B'} = E' \cdot T_B$$

mit

$$E' = \begin{pmatrix} 1 & & & \eta'_0 & & & \\ & \ddots & & \vdots & & & \\ & & 1 & \eta'_{r-1} & & & \\ & & & \eta'_r & & & \\ & & & \eta'_{r+1} & 1 & & \\ & & & \vdots & & \ddots & \\ & & & \eta'_m & & & 1 \end{pmatrix}, \qquad \begin{aligned} \eta'_i &= -\tfrac{(T_B)_{is}}{\overline{a}_{rs}},\ i \neq r \\ \eta'_r &= \tfrac{1}{\overline{a}_{rs}} \end{aligned}$$

(Die Transformationsformeln (9.18) sind im wesentlichen die gleichen wie in Satz (9.10), nur dass diesmal die Zielfunktion gleich mittransformiert wird.)

Beweis : Formel (a):

$$T_{B'} := \begin{pmatrix} c^T - c_{B'}^T A_{B'}^{-1} A, & -c_{B'}^T A_{B'}^{-1} b \\ A_{B'}^{-1} A, & A_{B'}^{-1} b \end{pmatrix},$$

d. h. mit Satz (9.10) gilt $(A_{B'}^{-1} = EA_B^{-1})$:

$$T_{B'} := \begin{pmatrix} (c^T, 0) - c_{B'}^T E(A_B^{-1}A, A_B^{-1}b) \\ E(T_B)_{\{1,\dots,m\}.} \end{pmatrix}.$$

Sei $K := \{1, \dots, m\}$. Aus $(T_{B'})_{K.} = E(T_B)_{K.}$ ergeben sich die beiden unteren Transformationsformeln in (a). Sei $z \in \mathbb{K}^m$ und $y = A_B^{-1}z$, dann gilt

$$c_{B'}^T A_{B'}^{-1} z = c_{B'}^T E A_B^{-1} z = c_{B'}^T E y = c_B^T y + (c_{B'}^T \eta - c_{p_r}) y_r$$

(mit η aus Satz (9.10)). Mit (9.13) folgt nun $c_{B'}^T \eta - c_{p_r} = \dfrac{\overline{c}_s}{\overline{a}_{rs}} = \dfrac{(T_B)_{0s}}{\overline{a}_{rs}}$. Also gilt:

$$\begin{aligned} (c^T, 0) - c_{B'}^T A_{B'}^{-1}(A, b) &= (c^T, 0) - c_B^T A_B^{-1}(A, b) - (T_B)_{0s} \overline{a}_{rs} (A_B^{-1}(A, b))_{r.} \\ &= (T_B)_{0.} - \tfrac{(T_B)_{0s}}{\overline{a}_{rs}} (T_B)_{r.} \end{aligned}$$

Die Formeln (b) ergeben sich direkt aus (a). □

(9.19) Beispiel. Wir betrachten das lineare Programm

$$\begin{array}{ll} \max x_1 + 2x_2 & \\ x_1 & \leq 4 \\ 2x_1 + x_2 & \leq 10 \\ -x_1 + x_2 & \leq 5 \\ x_1, x_2 & \geq 0, \end{array}$$

dessen Lösungsmenge in Abbildung 9.4 graphisch dargestellt ist.

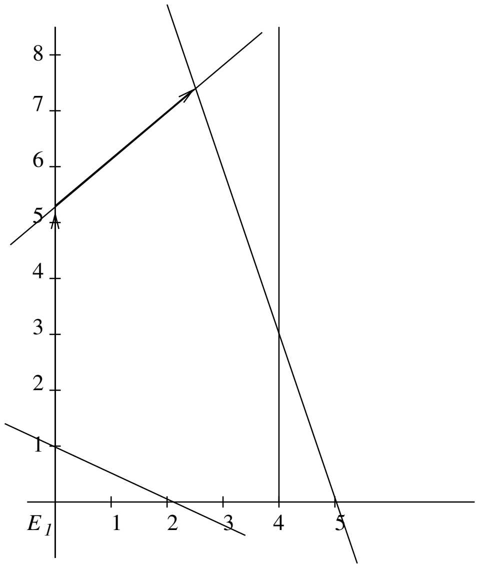


Abb. 9.4

Wir führen Schlupfvariablen x_3, x_4, x_5 ein und transformieren unser LP in folgende Form.

$$\max\ (1,\ 2,\ 0,\ 0,\ 0) \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \end{pmatrix}$$

$$\begin{pmatrix} 1 & 0 & 1 & 0 & 0 \\ 2 & 1 & 0 & 1 & 0 \\ -1 & 1 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \end{pmatrix} = \begin{pmatrix} 4 \\ 10 \\ 5 \end{pmatrix}$$

$$x_i \geq 0 \ i = 1, \ldots, 5.$$

Die Matrix A des Gleichungssystems des transformierten Systems hat 3 Zeilen und 5 Spalten und enthält eine Einheitsmatrix. Folglich hat sie vollen Zeilenrang.

Daraus folgt, dass die Spalten 3, 4, 5 eine Basis von A definieren, und diese ist zulässig, da die zugehörige Basislösung $x_3 = 4$, $x_4 = 10$, $x_5 = 5$, $x_1 = 0$, $x_2 = 0$ nichtnegativ ist. Diese Basislösung entspricht übrigens der Ecke $x_1 = 0$, $x_2 = 0$ des ursprünglichen Problems. Wir stellen nun das zur Anfangsbasis gehörige Tableau auf und führen den Simplexalgorithmus durch. Die Anfangsbasis ist gegeben durch $B = (3, 4, 5)$, und somit ist $N = (1, 2)$.

$$
\begin{array}{cc|rrrrr|r}
 & & \mathit{1} & \mathit{2} & \mathit{3} & \mathit{4} & \mathit{5} & \\
 & & 1 & 2 & 0 & 0 & 0 & 0 \\
\hline
T_0 : & \mathit{3} & 1 & 0 & 1 & 0 & 0 & 4 \\
 & \mathit{4} & 2 & 1 & 0 & 1 & 0 & 10 \\
 & \mathit{5} & -1 & \boxed{1} & 0 & 0 & 1 & 5
\end{array}
\qquad \overline{a}_{rs} = \overline{a}_{32} = 1
$$

Spaltenauswahl: Steilster Anstieg, d. h. der größte reduzierte Kostenkoeffizient wird gewählt. Das Pivotelement ist: $\overline{a}_{rs} = \overline{a}_{32} = 1$.

$$
\begin{array}{cc|rrrrr|r}
 & & 3 & 0 & 0 & 0 & -2 & -10 \\
\hline
T_1 : & \mathit{3} & 1 & 0 & 1 & 0 & 0 & 4 \\
 & \mathit{4} & \boxed{3} & 0 & 0 & 1 & -1 & 5 \\
 & \mathit{2} & -1 & 1 & 0 & 0 & 1 & 5
\end{array}
\qquad \overline{a}_{rs} = \overline{a}_{21} = 3
$$

$$
\begin{array}{cc|rrrrr|r}
 & & 0 & 0 & 0 & -1 & -1 & -15 \\
\hline
T_2 : & \mathit{3} & 0 & 0 & 1 & -\frac{1}{3} & \frac{1}{3} & \frac{7}{3} \\
 & \mathit{1} & 1 & 0 & 0 & \frac{1}{3} & -\frac{1}{3} & \frac{5}{3} \\
 & \mathit{2} & 0 & 1 & 0 & \frac{1}{3} & \frac{2}{3} & \frac{20}{3}
\end{array}
$$

Das Tableau T_2 ist optimal. Die optimale Lösung ist $(\frac{5}{3}, \frac{20}{3}, \frac{7}{3}, 0, 0)$, bzw. $x_1 = \frac{5}{3}$, $x_2 = \frac{20}{3}$.

Wir sehen an diesem Beispiel, dass es lästig ist, immer die Einheitsmatrix im Tableau mitzuschleppen. Wenn wir uns die gegenwärtigen Basis- und Nichtbasisvariablen geeignet (z. B. am Rand des Tableaus) merken, können wir die Einheitsmatrix weglassen. Ein derartiges Tableau nennt man **verkürztes Tableau**. Wir schreiben die vorhergehende Rechnung noch einmal in dieser verkürzten Form auf.

$$
\begin{array}{rr|rc}
\mathit{1} & \mathit{2} & & \\
1 & 2 & 0 & \\
\hline
1 & 0 & 4 & \mathit{3} \\
2 & 1 & 10 & \mathit{4} \\
-1 & 1 & 5 & \mathit{5}
\end{array}
$$

$$
\begin{array}{rr|rl}
\mathit{1} & \mathit{5} & & \\
3 & -2 & -10 & \\
\hline
1 & 0 & 4 & \mathit{3} \\
3 & -1 & 5 & \mathit{4} \\
-1 & 1 & 5 & \mathit{2}
\end{array}
$$

$$
\begin{array}{rr|rl}
\mathit{4} & \mathit{5} & & \\
-1 & -1 & -15 & \\
\hline
-\frac{1}{3} & \frac{1}{3} & \frac{7}{3} & \mathit{3} \\
\frac{1}{3} & -\frac{1}{3} & \frac{5}{3} & \mathit{1} \\
\frac{1}{3} & \frac{2}{3} & \frac{20}{3} & \mathit{2}
\end{array}
$$

$\square$

Ein weiteres Beispiel soll zur Einübung der Rechentechnik dienen.

(9.20) Beispiel. Wir betrachten das folgende lineare Programm.

$$
\begin{array}{rl}
\max x_1 + x_2 - x_3 & \\
3x_1 - 2x_2 & \leq 3 \\
2x_1 \quad + \quad x_3 & \leq 4 \\
x_1 + 3x_2 - 2x_3 & \leq 6 \\
x_1, x_2, x_3 & \geq 0
\end{array}
$$

Wir führen Schlupfvariablen ein und erhalten das folgende zulässige Anfangstableau.

$$
T_0 : \quad
\begin{array}{c|rrrrrr|r|l}
 & 1 & 1 & -1 & 0 & 0 & 0 & 0 & \\
\hline
\mathit{4} & \boxed{3} & -2 & 0 & 1 & 0 & 0 & 3 & s = 1 \\
\mathit{5} & 2 & 0 & 1 & 0 & 1 & 0 & 4 & r = 1 \\
\mathit{6} & 1 & 3 & -2 & 0 & 0 & 1 & 6 &
\end{array}
$$

$$
T_1 : \quad
\begin{array}{c|rrrrrr|r|l}
 & 0 & \frac{5}{3} & -1 & -\frac{1}{3} & 0 & 0 & -1 & \\
\hline
\mathit{1} & 1 & -\frac{2}{3} & 0 & \frac{1}{3} & 0 & 0 & 1 & s = 2 \\
\mathit{5} & 0 & \frac{4}{3} & 1 & -\frac{2}{3} & 1 & 0 & 2 & r = 3 \\
\mathit{6} & 0 & \boxed{\frac{11}{3}} & -2 & -\frac{1}{3} & 0 & 1 & 5 &
\end{array}
$$

$$
\begin{array}{lll|l}
 & & 0 \quad 0 \quad -\frac{1}{11} \quad -\frac{6}{33} \quad 0 \quad -\frac{5}{11} & -\frac{36}{11} \\
\hline
T_2 : & 1 & 1 \quad 0 \quad -\frac{4}{11} \quad \frac{9}{33} \quad 0 \quad \frac{2}{11} & \frac{21}{11} \\
 & 5 & 0 \quad 0 \quad \frac{19}{11} \quad -\frac{18}{33} \quad 1 \quad -\frac{4}{11} & \frac{2}{11} \\
 & 2 & 0 \quad 1 \quad -\frac{6}{11} \quad -\frac{1}{11} \quad 0 \quad \frac{3}{11} & \frac{15}{11}
\end{array}
$$

Als Optimallösung ergibt sich somit: $x_1 = \frac{21}{11}$, $x_2 = \frac{15}{11}$, $x_3 = 0$, mit $c^T x = \frac{36}{11}$. □

Über die Grundversion der Simplexmethode können wir die folgende Aussage machen.

(9.21) Satz. *Sei (9.1) ein lineares Programm, so dass $P^{=}(A, b) \neq \emptyset$ und alle zulässigen Basislösungen nicht entartet sind. Dann findet die Grundversion des Simplexalgorithmus (d. h. alle Schritte, bei denen eine nicht spezifizierte Auswahl besteht, werden in irgendeiner beliebigen Form ausgeführt) nach endlich vielen Schritten eine optimale Lösung oder stellt fest, dass das Problem unbeschränkt ist.*

Beweis : Bei jedem Durchlauf des Simplexalgorithmus wird eine neue Basis erzeugt. Da jede Basislösung nicht entartet ist, hat die neue Basislösung nach (9.14) (c) einen strikt besseren Zielfunktionswert. Ist $A_{B_1}, \ldots, A_{B_i}, \ldots$ die Folge der mit dem Simplexalgorithmus generierten Basen, dann folgt daraus, dass keine Basis mehrfach in dieser Folge auftreten kann. Da die Anzahl der verschiedenen Basen von A endlich ist, ist die Folge der Basen endlich. Ist A_{B_k} die letzte erzeugte Basis, so ist sie entweder optimal oder in Schritt (II.3) wurde Unbeschränktheit festgestellt. □

Besitzt (9.1) degenerierte Ecken, so kann es sein, dass die Grundversion des Simplexverfahrens irgendwann einmal nur noch zulässige Basen erzeugt, die alle zu derselben Ecke gehören und bei der die gleichen Matrizen $\overline{A}$ immer wiederkehren. Man sagt, das Verfahren kreist oder **kreiselt**. In der Tat sind Beispiele konstruiert worden, bei denen dieses Phänomen auftreten kann, siehe (9.22).

(9.22) Beispiel (Kreiseln). Gegeben sei das folgende LP:

$$
\begin{array}{rcl}
\max \frac{4}{5}x_1 - 18x_2 - x_3 - x_4 & & \\
\frac{16}{5}x_1 - 84x_2 - 12x_3 + 8x_4 & \leq & 0 \\
\frac{1}{5}x_1 - 5x_2 - \frac{2}{3}x_3 + \frac{1}{3}x_4 & \leq & 0 \\
x_1 \qquad\qquad\qquad\qquad & \leq & 1 \\
x_1, x_2, x_3, x_4 & \geq & 0
\end{array}
$$

Das verkürzte Tableau zur Anfangsbasis $B = (5, 6, 7)$ lautet:

$$\begin{array}{rrrr|rl}
\mathit{1} & \mathit{2} & \mathit{3} & \mathit{4} & & \\
\frac{4}{5} & -18 & -1 & -1 & 0 & \\
\hline
\boxed{\frac{16}{5}} & -84 & -12 & 8 & 0 & \mathit{5} \\
\frac{1}{5} & -5 & -\frac{2}{3} & \frac{1}{3} & 0 & \mathit{6} \\
1 & 0 & 0 & 0 & 1 & \mathit{7}
\end{array}$$

Wir führen nun den Simplexalgorithmus aus. Die Pivotelemente sind in den Tableaus gekennzeichnet.

$$\begin{array}{rrrr|rl}
\mathit{5} & \mathit{2} & \mathit{3} & \mathit{4} & & \\
-\frac{1}{4} & 3 & 2 & -3 & 0 & \\
\hline
\frac{5}{16} & -\frac{105}{4} & -\frac{15}{4} & \frac{5}{2} & 0 & \mathit{1} \\
-\frac{1}{16} & \boxed{\frac{1}{4}} & \frac{1}{12} & -\frac{1}{6} & 0 & \mathit{6} \\
-\frac{5}{16} & \frac{105}{4} & \frac{15}{4} & -\frac{5}{2} & 1 & \mathit{7}
\end{array}
\qquad
\begin{array}{rrrr|rl}
\mathit{5} & \mathit{6} & \mathit{3} & \mathit{4} & & \\
\frac{1}{2} & -12 & 1 & -1 & 0 & \\
\hline
-\frac{25}{4} & 105 & \boxed{5} & -15 & 0 & \mathit{1} \\
-\frac{1}{4} & 4 & \frac{1}{3} & -\frac{2}{3} & 0 & \mathit{2} \\
\frac{25}{4} & -105 & -5 & 15 & 1 & \mathit{7}
\end{array}$$

$$\begin{array}{rrrr|rl}
\mathit{5} & \mathit{6} & \mathit{1} & \mathit{4} & & \\
\frac{7}{4} & -33 & -\frac{1}{5} & 2 & 0 & \\
-\frac{5}{4} & 21 & \frac{1}{5} & -3 & 0 & \mathit{3} \\
\frac{1}{6} & -3 & -\frac{1}{15} & \boxed{\frac{1}{3}} & 0 & \mathit{2} \\
0 & 0 & 1 & 0 & 1 & \mathit{7}
\end{array}
\qquad
\begin{array}{rrrr|rl}
\mathit{5} & \mathit{6} & \mathit{1} & \mathit{2} & & \\
\frac{3}{4} & -15 & \frac{1}{5} & -6 & 0 & \\
\boxed{\frac{1}{4}} & -6 & -\frac{2}{5} & 9 & 0 & \mathit{3} \\
\frac{1}{2} & -9 & -\frac{1}{5} & 3 & 0 & \mathit{4} \\
0 & 0 & 1 & 0 & 1 & \mathit{7}
\end{array}$$

$$\begin{array}{rrrr|rl}
\mathit{3} & \mathit{6} & \mathit{1} & \mathit{2} & & \\
-3 & 3 & \frac{7}{5} & -33 & 0 & \\
\hline
4 & -24 & -\frac{8}{5} & 36 & 0 & \mathit{5} \\
-2 & \boxed{3} & \frac{3}{5} & -15 & 0 & \mathit{4} \\
0 & 0 & 1 & 0 & 1 & \mathit{7}
\end{array}
\qquad
\begin{array}{rrrr|rl}
\mathit{3} & \mathit{4} & \mathit{1} & \mathit{2} & & \\
-1 & -1 & \frac{4}{5} & -18 & 0 & \\
\hline
-12 & 8 & \frac{16}{5} & -84 & 0 & \mathit{5} \\
-\frac{2}{3} & \frac{1}{3} & \frac{1}{5} & -5 & 0 & \mathit{6} \\
0 & 0 & 1 & 0 & 1 & \mathit{7}
\end{array}$$

Das letzte Tableau ist bis auf Spaltenvertauschung das gleiche wie das erste. Alle Basen der oben generierten Folge gehören zur Ecke $x^T = (0, 0, 0, 0)$ des Ausgangsproblems. Die bei der Berechnung des Beispiels benutzte Variante des Simplexverfahrens würde also nicht nach endlich vielen Schritten abbrechen. $\square$

Wie Satz (9.21) zeigt, funktioniert die Grundversion der Simplexmethode, gleichgültig welche Arten von Auswahlregeln man in den Schritten (II.2) und (II.5) benutzt, wenn alle Basislösungen nicht entartet sind. Es ist daher sinnvoll zu fragen, ob man nicht das Ausgangsproblem so modifizieren (stören) kann, dass das neue Problem nicht entartet ist. Dass dies geht, zeigt der folgende Satz. Wie üblich bezeichnen wir das LP $\max\{c^T x \mid Ax = b, x \geq 0\}$ mit (9.1). Sei für ein $\varepsilon^T = (\varepsilon_1, \ldots, \varepsilon_n), \varepsilon > 0$

$$\begin{array}{ll} \max c^T x & \\ Ax = & b + A\varepsilon \\ x \geq & 0. \end{array}$$

(9.23) Satz (Perturbationsmethode).

(a) *Das lineare Programm (9.1) ist genau dann lösbar, wenn das LP (9.1ε) lösbar ist für jedes $\varepsilon > 0$.*

(b) *Es gibt ein $\varepsilon_0 > 0$ derart, dass für alle $0 < \varepsilon < \varepsilon_0$ jede zulässige Basis von (9.1ε) zum einen eine nichtdegenerierte Basislösung von (9.1ε) bestimmt und zum anderen eine zulässige Basis von (9.1) ist. Ferner bestimmt die zu einer optimalen Basislösung von (9.1ε) gehörige Basis eine optimale Basislösung von (9.1).*

Beweis : P. Kall, "Mathematische Methoden des Operations Research", Teubner Studienbücher, Stuttgart, S. 52–53. $\square$

Aufgrund von Satz (9.23) könnte also jedes LP durch Störung in ein nichtentartetes LP umgewandelt werden, und damit wäre die Konvergenz der Grundversion des Simplexalgorithmus gewährleistet. In der Praxis wird diese Methode jedoch kaum angewendet, da das ε_0 nicht einfach zu bestimmen ist und der Ansatz mit irgendeiner Zahl $\varepsilon > 0$ nicht zum Ziele führen muss. Ferner könnten durch eine geringe Störung des Problems durch Rundungsfehler numerische Schwierigkeiten auftreten, deren Auswirkungen nicht immer abschätzbar sind.

Wir werden im nächsten Kapitel andere Methoden zur Vermeidung des Kreisens kennenlernen. In der Praxis werden diese häufig jedoch gar nicht implementiert, da bei praktischen Problemen trotz gelegentlicher Degeneration ein Kreisen sehr selten beobachtet wurde. Einer der Gründe dafür dürfte darin zu sehen sein, dass der Computer durch seine begrenzte Rechengenauigkeit sozusagen von allein eine Störungsmethode durchführt. So wird bei jedem Auf- oder Abrunden eine kleine Änderung des vorliegenden Programms vorgenommen, die sich positiv auf die Konvergenzeigenschaften des Simplexverfahrens auswirkt.

9.4 Die Phase I

Zur vollständigen Beschreibung der Grundversion der Simplexmethode fehlt noch eine Darstellung der Phase I von (9.15). Diese wollen wir nun nachholen. Ist das Ungleichungssystem $Ax = b, x \geq 0$ gegeben, so können wir stets o. B. d. A. $b \geq 0$ annehmen, denn Multiplikation einer Gleichung mit -1 verändert das Polyeder nicht.

(9.24) Phase I des Simplexverfahrens.

Input: $A \in \mathbb{K}^{(m,n)}$, $b \in \mathbb{K}^m$ *mit* $b \geq 0$.

Output:

(a) $P^=(A, b) = \emptyset$ *oder*

(b) $P^=(A, b) = \{x\}$ *oder*

(c) Ausgabe von $I \subseteq \{1, \ldots, m\}$ *und* $B = (p_1, \ldots, p_k)$ *mit folgenden Eigenschaften:*

  (1) Mit $A' := A_{I.}$, $b' := b_I$ *gilt* $P^=(A', b') \neq \emptyset$, $\text{rang}(A') = |I| = k$, $k < n$ *und* $P^=(A', b') = P^=(A, b)$.

  (2) A'_B *ist eine zulässige Basis von* A'.

Wir formulieren zunächst ein lineares "Hilfsprogramm". Sei $D := (A, I)$ und betrachte das LP:

$$(9.25) \qquad \begin{array}{ll} \max \mathbb{1}^T Ax & \\ D\binom{x}{y} & = b \\ x, y & \geq 0 \end{array}$$

wobei $x^T = (x_1, \dots, x_n)$, $y^T = (y_{n+1}, \dots, y_{n+m})$ gesetzt wird. (9.25) erfüllt die Zusatzvoraussetzungen (9.2) (a), (b), (c), und mit $B = (n+1, \dots, n+m)$ ist D_B eine zulässige Basis mit zugehöriger Basislösung $x = 0$, $y = b$. Es gilt offenbar

$$\begin{array}{ll} D\binom{x}{y} & = b \\ x, y & \geq 0 \end{array} \quad \Longleftrightarrow \quad \begin{array}{ll} Ax + y & = b \\ x, y & \geq 0 \end{array}$$

daraus folgt $\mathbb{1}^T Ax = \mathbb{1}^T b - \mathbb{1}^T y$, d. h. (9.25) ist äquivalent zu $\max\{\mathbb{1}^T b - \mathbb{1}^T y \mid Ax + y = b,\ x, y \geq 0\}$ bzw. zu

$$\text{(9.26)} \qquad \begin{array}{ll} \mathbb{1}^T b - \min \mathbb{1}^T y & \\ Ax + y = & b \\ x, y \geq & 0. \end{array}$$

(9.26) bzw. (9.25) besagen also, dass die künstlich eingeführten Variablen möglichst klein gewählt werden sollen.

(I.1) *Wende die Grundversion (9.15) des Simplexverfahrens auf (9.25) an. Die Matrix D hat vollen Zeilenrang und weniger Zeilen als Spalten, $B = (n + 1, \dots, n + m)$ definiert eine zulässige Basis, zu der die Matrix $\overline{A}$ und die Vektoren $\overline{b}$, $\overline{c}$ und c_0 trivial zu berechnen sind. Wir können also direkt mit diesen Daten mit Phase II beginnen.*

Das LP (9.26) ist offenbar durch $\mathbb{1}^T b - 0 = \mathbb{1}^T b$ nach oben beschränkt, also ist (9.25) durch $\mathbb{1}^T y$ nach oben beschränkt. Der Dualitätssatz impliziert, dass (9.25) eine optimale Lösung besitzt. Somit endet das Simplexverfahren mit einer optimalen Basis D_B und zugehöriger Basislösung $z = \binom{x}{y}$. Wir werden nun die optimale Lösung bzw. die optimale Basis analysieren, um aus Eigenschaften der Lösung bzw. Basis die Schlussfolgerungen (a), (b) oder (c) ziehen zu können.

(I.2) *Falls $\mathbb{1}^T Ax < \mathbb{1}^T b$, so wird das Minimum in (9.26) nicht in $y = 0$ angenommen. Hieraus folgt offenbar, dass $P^=(A, b) = \emptyset$ gilt, und wir können das Verfahren mit Antwort (a) beenden, STOP!* Die folgenden Schritte behandeln den Fall $\mathbb{1}^T Ax = \mathbb{1}^T b$. D. h. es gilt $\mathbb{1}^T Ax = \mathbb{1}^T b - \mathbb{1}^T y = \mathbb{1}^T b$ bzw. $\mathbb{1}^T y = 0$. Somit gilt $y = 0$, und x ist zulässig bzgl. $Ax = b$, $x \geq 0$. Wir müssen nun noch evtl. Zeilen von A streichen, um die Rangbedingung zu erfüllen.

(I.3) *Falls $B \cap \{n + 1, \dots, n + m\} = \emptyset$, so befinden sich in der optimalen Basis keine künstlichen Variablen, d. h. es gilt $D_B = A_B$. Da $D_B = A_B$ regulär* *ist, gilt* $\text{rang}(A) = m$. *Falls* $N \cap \{1, \dots, n\} = \emptyset$, *so sind alle Nichtbasisvariablen künstliche Variablen, und wir haben* $m = n$. *Dann gilt* $x = A^{-1}b$ *und* $P^{=}(A, b) = \{x\}$. *In diesem Falle können wir das Verfahren beenden mit Antwort (b), STOP! Der Vektor* x *ist die Optimallösung eines jeden LP über* $P^{=}(A, b)$. *Andernfalls ist* $m < n$ *und* A_B *ein zulässige Basis von* A. *Wir setzen* $I = M$ *und schließen das Verfahren mit Antwort (c) ab, STOP!*

Die beiden abschließenden Schritte sind für den Fall vorgesehen, dass sich noch künstliche Variablen in der Basis befinden, d. h. falls $B \cap \{n+1, \dots, n+m\} \neq \emptyset$. Wir versuchen zunächst, diese künstlichen Variablen aus der Basis zu entfernen. Da alle künstlichen Variablen $y_{n+1}, \dots, y_{n+m}$ Null sind, ist die Basislösung $z = (z_B, z_N)$, $z_B = D_B^{-1}b$ degeneriert. Sei o. B. d. A.

$$z_B^T = (y_{p_1}, \dots, y_{p_t}, x_{p_{t+1}}, \dots, x_{p_m}),$$

d. h. $B \cap \{n+1, \dots, n+m\} = \{p_1, \dots, p_t\}$. Wenn wir also künstliche Variablen aus der Basis entfernen wollen, müssen wir prüfen, ob wir sie aus der Basis "hinauspivotisieren" können, d. h. ob in den Zeilen $\overline{D}_{i.}$ $(i = 1, \dots, t)$ $(\overline{D} = (\overline{d}_{rs}) = D_B^{-1}D_N)$ von Null verschiedene Pivotelemente vorhanden sind. Da wir ja nur Nichtbasisvariable gegen degenerierte Basisvariable austauschen, also Nullen gegen Nullen tauschen, können wir auch Pivotoperationen auf negativen Elementen zulassen.

(I.4) *Falls es* $r \in \{1, \dots, t\}$ *und* $s \in \{1, \dots, n\}$ *gibt, so dass* z_{q_s} *keine künstliche Variable ist und* $\overline{d}_{rs} \neq 0$ *gilt, dann*

führe eine Pivotoperation mit dem Pivotelement $\overline{d}_{rs}$ *entsprechend Satz (9.10) durch. Wir erhalten dadurch eine neue Basis* $D_{B'}$, *mit einer künstlichen Variablen weniger, setzen* $B := B'$ *und gehen zu (I.3).*

(I.5) *Falls* $\overline{d}_{ij} = 0 \; \forall\, i \in \{1, \dots, t\} \; \forall\, s \in \{1, \dots, n\}$, *für die* z_{q_s} *keine künstliche Variable ist, dann hat das Gleichungssystem* $z_B = D_B^{-1}b - D_B^{-1}D_N z_N$ *bis auf Permutationen der Zeilen und Spalten folgende Form:*

$$\begin{pmatrix} z_{p_1} \\ \vdots \\ z_{p_t} \end{pmatrix} = \begin{bmatrix} 0 \\ \vdots \\ 0 \end{bmatrix} - \left[\begin{array}{c|c} \quad 0 \quad & \quad * \quad \end{array} \right] \begin{pmatrix} x_N \\ y_N \end{pmatrix}$$

und

$$\begin{pmatrix} z_{p_{t+1}} \\ \vdots \\ z_{p_m} \end{pmatrix} = \begin{bmatrix} * \end{bmatrix} - \left[\begin{array}{c|c} \quad * \quad & \quad * \quad \end{array} \right] \begin{pmatrix} x_N \\ y_N \end{pmatrix}$$

(mit $y_B = (z_{p_1}, \dots, z_{p_t})$, $x_B = (z_{p_{t+1}}, \dots, z_{p_m})$, $z_N = (x_N, y_N)$).

Setzen wir also die künstlichen Variablen Null, so ist der obere Teil des Gleichungssystems

$$y_B = 0 - 0x_N$$

unabhängig von der Belegung von x_N stets erfüllt, d. h. wir können alle Zeilen $D_{1.}, \dots, D_{t.}$ streichen. Ist $I = \{t+1, \dots, m\}$ und $J = \{j \in \{1, \dots, n\} \mid z_{q_j}$ ist keine künstliche Variable $\}$ so gilt

$$x_B = (D_B^{-1}b)_I - \overline{D}_{IJ}x_N$$

und $B = (p_{t+1}, \dots, p_m)$ ist eine zulässige Basis von $A' := A_{I.}$ mit $\operatorname{rang}(A') = k := m - t$.

Ist $k = n$, so gilt wieder $P^=(A, b) = \{x\}$, und wir enden mit Antwort (b) andernfalls gilt $k < n$ und wir beenden das Verfahren mit Antwort (c), STOP!

□

Die Korrektheit des Phase I-Verfahrens (9.24) ist aufgrund der oben gemachten Bemerkungen offensichtlich. (9.24) zeigt auch, dass man bei der Lösung von linearen Programmen im wesentlichen mit der Grundversion des Simplexalgorithmus zur Lösung von Phase II auskommt. Es ist lediglich zusätzlich etwas Buchhaltung (Einführung künstlicher Variablen, Streichen von Spalten und Zeilen) zusätzlich durchzuführen. Außerdem sind u. U. im Schritt (I.4) Pivotoperationen auf negativen Pivotelementen erforderlich. Die Pivotformeln ändern sich dadurch aber nicht.

(9.27) Bemerkung. Liegen Probleme mit nicht vorzeichenbeschränkten Variablen vor, so wendet man die Transformationsregeln (2.4) aus Kapitel 2 an, um das Problem in ein LP mit vorzeichenbeschränkten Variablen zu transformieren.

□

Zur Auffindung einer Anfangslösung gibt es noch andere Varianten (z. B. die M-Methode), jedoch wird in kommerziellen Codes meist die Zweiphasenmethode verwendet.

(9.28) Beispiel (für Phase I+II). Gegeben sei das folgende lineare Programm in

Standardform

$$
\begin{array}{rcr}
\max x_1 - x_2 + 2x_3 & & \\
2x_1 - 3x_2 + x_3 & = & 3 \\
-x_1 + 2x_2 + x_3 & = & -1 \\
3x_1 - 5x_2 & = & 4 \\
x_1, x_2, x_3 & \geq & 0
\end{array}
$$

Durch Multiplikation der zweiten Zeile mit -1 machen wir die rechten Seiten nichtnegativ. Es gilt dann $\mathbb{1}^T A = (6, -10, 0)$, $\mathbb{1}^T b = 8$. Wir stellen nun unser Simplextableau auf, wobei wir künstliche Variablen s_1, s_2, s_3 einführen. Wir fügen dem Simplextableau eine neue oberste Zeile für die künstliche Zielfunktion hinzu.

		x_1	x_2	x_3	s_1	s_2	s_3		
		6	-10	0	0	0	0	0	$\leftarrow$ künstliche Zielfunktion
		1	-1	2	0	0	0	0	$\leftarrow$ akt. Zielfunktion
T_0 :	s_1	2	-3	1	1	0	0	3	
	s_2	$\boxed{1}$	-2	-1	0	1	0	1	
	s_3	3	-5	0	0	0	1	4	

		0	2	6	0	-6	0	-6
		0	1	3	0	-1	0	-1
T_1 :	s_1	0	1	$\boxed{3}$	1	-2	0	1
	x_1	1	-2	-1	0	1	0	1
	s_3	0	1	3	0	-3	1	1

		x_1	x_2	x_3	s_1	s_2	s_3	
		0	0	0	-2	-2	0	-8
		0	0	0	-1	1	0	-2
T_2 :	x_3	0	$\frac{1}{3}$	1	$\frac{1}{3}$	$-\frac{2}{3}$	0	$\frac{1}{3}$
	x_1	1	$-\frac{5}{3}$	0	$\frac{1}{3}$	$\frac{1}{3}$	0	$\frac{4}{3}$
	s_3	0	0	0	-1	-1	1	0

Das Tableau T_2 ist optimal bezüglich der künstlichen Zielfunktion. Es gilt $\mathbb{1}^T Ax = \mathbb{1}^T b$. $B \cap \{n+1, \dots, n+m\} = \{n+3\} = \{p_3\} \neq \emptyset$ (d. h. s_3 ist in der Basis). Schritt (I.4) wird nicht ausgeführt, da $\overline{d}_{32} = 0$, also dürfen wir entsprechend (I.5) die letzte Zeile und die Spalten 4, 5 und 6 streichen. Dies ergibt das folgende zulässige Tableau für das ursprüngliche Problem

	x_1	x_2	x_3	
	0	0	0	-2
x_3	0	$\frac{1}{3}$	1	$\frac{1}{3}$
x_1	1	$-\frac{5}{3}$	0	$\frac{4}{3}$

Dieses Tableau ist optimal bezüglich der ursprünglichen Zielfunktion. $x_1 = \frac{4}{3}$, $x_2 = 0$, $x_3 = \frac{1}{3}$, $c^T x = 2$.

Kapitel 10

Varianten der Simplex-Methode

Nachdem wir nun wissen, wie der Simplexalgorithmus im Prinzip funktioniert, wollen wir uns in diesem Kapitel überlegen, wie die noch offen gelassenen Schritte am besten ausgeführt und wie die verschiedenen Updates numerisch günstig implementiert werden können.

10.1 Der revidierte Simplexalgorithmus

Betrachtet man die Grundversion des Simplexalgorithmus (9.15), so stellt man fest, dass während einer Iteration i. a. nicht sämtliche Spalten der Matrix $\overline{A}$ benötigt werden. Der revidierte Simplexalgorithmus nutzt diese Tatsache aus, indem er stets nur solche Spalten von $\overline{A}$ berechnet, die im jeweiligen Schritt benötigt werden.

Alle numerischen Implementationen des Simplexverfahrens benutzen als Grundgerüst den revidierten Simplexalgorithmus. Wir wollen deshalb im folgenden den revidierten Simplexalgorithmus in Unterprogramme (Subroutines) zerlegen und später zeigen, wie diese Subroutines bei den jeweiligen numerischen Implementationsvarianten realisiert werden. Wir gehen immer davon aus, dass wir ein LP in Standardform (9.1) vorliegen haben. Alle Bezeichnungen werden, wie in Kapitel 9 festgelegt, verwendet.

(10.1) Algorithmus (Revidierter Simplexalgorithmus). *Gegeben seien:* A, A_B^{-1}, $\overline{b}$, c *und* B.

(1) BTRAN *(Backward Transformation)*

Berechnet: $\pi^T := c_B^T A_B^{-1}$ *(Schattenpreise)*

(2) PRICE *(Pivotspaltenauswahl)*

Berechnet die reduzierten Kostenkoeffizienten: $\overline{c}_j := (c_N^T)_j - \pi^T A_N e_j$, *für* $j = 1, \ldots, n-m$ *und wählt einen Index* s *mit* $\overline{c}_s > 0$.

(1) FTRAN *(Forward Transformation)*

Aktualisiert die Pivotspalte:

$$\overline{d} := A_B^{-1} d = A_B^{-1} A_{\cdot q_s}.$$

(4) CHUZR *(Pivotzeilenauswahl)*

Berechnet: $\lambda_0 = \min\{\frac{\overline{b}_i}{\overline{d}_i} \mid \overline{d}_i > 0,\ i = 1, \ldots, m\}$ *und wählt einen Index* $r \in \{i \in \{1, \ldots, m\} \mid \frac{\overline{b}_i}{\overline{d}_i} = \lambda_0\}$.

(4) WRETA *(Updating der Basis)*

Entferne das r*-te Element von* B *und ersetze es durch* q_s. *Der neue Spaltenindexvektor heiße* B'.

Berechnet: $A_{B'}^{-1}$, $\overline{b} = A_{B'}^{-1} b$.

(4) *Abbruchkriterium wie üblich (siehe Schritt (II.1) von (9.15)).*

□

Wir werden später und in den Übungen auf numerische Tricks zur Implementation der Updates eingehen. Es seien hier jedoch bereits einige Vorteile der revidierten Simplexmethode festgehalten:

– Wesentlich weniger Rechenaufwand, speziell bei Programmen mit erheblich mehr Variablen als Gleichungen.

– Die Ausgangsmatrix A kann auf externem Speicher bzw. bei dünn besetzten Matrizen durch spezielle Speichertechniken gehalten werden. Spalten von A werden je nach Bedarf generiert. (Sparse-Matrix-Techniques).

– Die Kontrolle über die Rechengenauigkeit ist besser. Insbesondere kann bei Bedarf mit Hilfe eines Inversionsunterprogramms A_B^{-1} neu berechnet werden.

– Es gibt besondere Speichertechniken, die eine explizite Speicherung von A_B^{-1} vermeiden. Man merkt sich lediglich die Etavektoren und berechnet dann über die Formel aus (9.10) aus der ersten Basisinversen durch Linksmultiplikation mit den Elementarmatrizen die gegenwärtige Basisinverse.

10.2 Spalten- und Zeilenauswahlregeln

Wir wollen uns nun damit beschäftigen, wie die noch unspezifizierten Schritte des Simplexverfahrens “vernünftig” ausgeführt werden können.

(10.2) Varianten der PRICE-Routine (10.1) (2). Sei $S := \{j \in \{1, \ldots, n - m\} \mid \overline{c}_j > 0\} \neq \emptyset$. Folgende Regeln werden häufig angewendet, bzw. sind in der Literatur eingehend diskutiert worden.

(1) **Kleinster-Index-Regel:** Wähle $s = \min\{j \in S\}$

(2) **Kleinster-Variablenindex-Regel:** Wähle $s \in S$, so dass $q_s = \min\{q_j \in N \mid j \in S\}$.

(3) **Steilster-Anstieg-Regel:** Wähle $s \in S$, so dass $\overline{c}_s = \max\{\overline{c}_j \mid j \in S\}$.

(4) **Größter-Fortschritt-Regel:** Berechne für jedes $j \in S$ zunächst $\lambda_0^j = \min\{\frac{\overline{b}_i}{\overline{a}_{ij}} \mid \overline{a}_{ij} > 0, i = 1, \ldots, m\}$ und $g_j = \overline{c}_j \lambda_0^j$.
Wähle $s \in S$, so dass $g_s = \max\{g_j \mid j \in S\}$.

(5) **Varianten von (4):** Es gibt unzählige Varianten von (4), einige davon sind beschrieben in:

D. Goldfarb, J. K. Reid: “A practicable steepest-edge simplex algorithm”, Mathematical Programming 12 (1977), 361–371.

P. M. S. Harris: “Pivot selection methods for the Devex LP code”, Mathematical Programming 5 (1973), 1–28.

H. Crowder, J. M. Hattingh: “Partially normalized pivot selection in linear programming”, Math. Progr. Study 4 (1975), 12–25.

□

Die Regel (10.2) (1) ist die rechentechnisch einfachste. Man durchläuft den Vektor $\overline{c}$, sobald ein Index s mit $\overline{c}_s > 0$ gefunden ist, hört man auf und geht zum nächsten Schritt. Die reduzierten Kosten müssen also nicht alle berechnet werden. Dadurch wird viel Aufwand gespart, aber aufgrund der simplen Wahl von s ist die Gesamtzahl der Pivotoperationen im Vergleich zu anderen Regeln recht hoch.

Ähnliches gilt für Regel (2). Hier müssen jedoch alle reduzierten Kostenkoeffizienten berechnet werden. Diese Regel ist jedoch aus einem theoretischen Grund, der noch diskutiert werden soll, interessant.

Die Regel (3) ist die Regel, die bei einfachen Implementationen (keine sophistifizierten Tricks) am häufigsten verwendet wird und bei Problemen bis zu mittleren Größenordnungen (jeweils bis zu 500 Variablen und Zeilen) recht gute Ergebnisse zeigt. Ihr liegt die Idee zu Grunde, dass diejenige Variable in die Basis genommen werden sollte, die pro Einheit den größten Zuwachs in der Zielfunktion bringt.

Es kann natürlich sein, dass die Nichtbasisvariable mit dem größten Zuwachs pro Einheit nur um sehr wenige Einheiten erhöht werden kann und dass ein Wechsel zu einer anderen Basis einen wesentlich größeren Gesamtfortschritt bringt. Hierzu ist Regel (4) geschaffen. Bei ihr wird für jeden möglichen Basiswechsel der tatsächliche Zuwachs der Zielfunktion g_j berechnet, und es wird der Basiswechsel vorgenommen, der den insgesamt größten Fortschritt bringt. Diese Verbesserung wird natürlich durch einen erheblichen rechnerischen Mehraufwand erkauft, bei dem es fraglich ist, ob er sich lohnt.

Aufgrund von Erfahrungen in der Praxis kann gesagt werden, dass sich die trivialen Regeln (1), (2) und (3) für kleinere bis mittlere Problemgrößen bewährt haben, jedoch für große LPs, Modifizierungen von (10.2) (4), wie sie in den Literaturangaben beschrieben sind, benutzt werden. Die trivialen Regeln führen im allgemeinen zu insgesamt mehr Pivotoperationen. Die komplizierten Regeln versuchen die Anzahl der Pivotoperationen minimal zu gestalten. Hierbei ist ein wesentlich höherer Rechenaufwand erforderlich (viele Spalten von $\overline{A}$ sind zu generieren und für jede Spalte ist λ_0 zu berechnen), der bei kleineren Problemen durchaus einem mehrfachen Basisaustausch entspricht und sich daher nicht auszahlt. Bei großen Problemen ist i. a. der Basiswechsel rechenaufwendiger als die komplizierte Auswahl der Pivotspalte. Hier zahlt sich dann die sorgfältige Auswahl der Pivotspalte bei der Gesamtrechenzeit aus.

Bei der Auswahl von Pivotzeilen gibt es nicht so viele interessante Regeln, aber hier kann durch geschickte Wahl die Endlichkeit des Simplexverfahrens erzwungen werden.

Im folgenden gehen wir davon aus, dass der Spaltenindex s durch eine Spaltenauswahlregel (10.2) bestimmt wurde und setzen $R := \{i \in \{1 \ldots, m\} \mid \frac{\overline{b}_i}{\overline{a}_{is}} = \lambda_0\}$. Wir treffen zunächst eine Definition.

(10.3) Definition. *Ein von Null verschiedener Vektor $x^T = (x_1, \ldots, x_n) \in \mathbb{K}^n$ heißt* **lexikographisch positiv** *(wir schreiben: $x \succ 0$), wenn die erste von Null verschiedene Komponente positiv ist (d. h. ist $i := \min \operatorname{supp}(x)$, dann ist $x_i > 0$). Wir schreiben $x \succ y$, falls $x - y \succ 0$ gilt.*

□

(10.4) Bemerkung. "$\succ$" definiert eine totale Ordnung im $\mathbb{K}^n$. Wir schreiben $x \succeq y \iff x \succ y \ \vee \ x = y$ und bezeichnen mit lex–$\min S$ das lexikographische Minimum einer endlichen Menge $S \subseteq \mathbb{K}^n$.

$\square$

(10.5) 1. Lexikographische Zeilenauswahlregel. Wähle $r \in R$ so dass

$$\frac{1}{\overline{a}_{rs}}(A_B^{-1}A)_{r.} = \text{lex-}\min\{\frac{1}{\overline{a}_{is}}(A_B^{-1}A)_{i.} \mid \overline{a}_{is} > 0,\ i \in \{1, \ldots, m\}\}.$$

$\square$

Mit Regel (10.5) wird jedes Simplexverfahren unabhängig von der Spaltenauswahlregel endlich.

(10.6) (Endlichkeit des Simplexalgorithmus). Wird im Simplexverfahren die 1. Lexikographische Regel (10.5) verwendet, so endet der Algorithmus nach endlich vielen Schritten, gleichgültig, welche Spaltenauswahlregel verwendet wird.

Beweis : Wir zeigen zuerst:

$$(a) \qquad \frac{1}{\overline{a}_{rs}}(A_B^{-1}A)_{r.} \prec \frac{1}{\overline{a}_{is}}(A_B^{-1}A)_{i.} \ \forall\, i \neq r,$$

das heißt, das lexikographische Minimum wird eindeutig angenommen. Seien r und r' Indizes mit

$$(*) \qquad \frac{1}{\overline{a}_{rs}}(A_B^{-1}A)_{r.} = \frac{1}{\overline{a}_{r's}}(A_B^{-1}A)_{r'.}$$

und o. B. d. A. sei $A = (A_B, A_N)$. Dann gilt $(A_B^{-1}A)_{r.} = (A_B^{-1})_{r.}A = (e_r^T, \overline{A}_{r.})$ und $(A_B^{-1}A)_{r'.} = (e_{r'}^T, \overline{A}_{r'.})$ und $(*)$ impliziert $(e_r^T, \overline{A}_{r.}) = \frac{\overline{a}_{rs}}{\overline{a}_{r's}}(e_{r'}^T, \overline{A}_{r'.})$. Damit folgt $e_r = \frac{\overline{a}_{rs}}{\overline{a}_{r's}} e_{r'}$ und somit $r = r'$. Also gilt (a).

Sei A_B nun eine zulässige Basis von A (z. B. die Startbasis (evtl. auch die der Phase I)) und sei T_B das zugehörige Simplextableau (vgl. (9.16)). O. B. d. A. sei

$$T_B = \begin{pmatrix} 0 & \overline{c} & -c_B^T A_B^{-1} b \\ I & \overline{A} & A_B^{-1} b \end{pmatrix}$$

(evtl. ist eine Permutation der Spalten der Ausgangsmatrix notwendig). Sei $B' = (p_1, \ldots, p_{r-1}, q_s, p_{r+1}, \ldots, p_m)$. Dann gilt

$$(b) \qquad (T_{B'})_{r.} \succ 0$$

denn $(T_B)_{r\cdot} \succ 0$ und $\overline{a}_{rs} > 0$ (vgl. Formel (9.18) (a)). Da nach Voraussetzung $(T_B)_{i\cdot} \succ 0 \; \forall \, i = 1, \dots, m$, folgt

$$(c) \qquad (T_{B'})_{i\cdot} \succ 0 \; \forall \, i \text{ mit } \overline{a}_{is} \leq 0$$

vgl. (9.18) (a).

Aus (a) folgt nun $(T_B)_{i\cdot} \succ \frac{\overline{a}_{is}}{\overline{a}_{rs}} (T_B)_{r\cdot}$ für alle $i \neq r$ mit $\overline{a}_{is} > 0$, d. h.

$$(d) \qquad (T_{B'})_{i\cdot} \succ 0 \; \forall \, i \text{ mit } \overline{a}_{is} > 0.$$

Somit sind bei Anwendung der 1. Lexikographischen Regel stets alle Zeilen $(T_B)_{i\cdot}$ $(i \neq 0)$ des Simplextableaus lexikographisch positiv, d. h. es gilt für 2 aufeinanderfolgende Basen B und B' (vgl. (9.18) (a))

$$(T_{B'})_{o\cdot} - (T_B)_{o\cdot} = -\frac{(T_B)_{os}}{\overline{a}_{rs}} (T_B)_{r\cdot} \prec 0,$$

denn $(T_B)_{os} > 0$. Also sind alle Simplextableaus verschieden und da nur endlich viele zulässige Basislösungen und somit Simplextableaus existieren, folgt nun die Behauptung. □

(10.7) 2. Lexikographische Zeilenauswahlregel. Wähle $r \in R$, so dass

$$\frac{1}{\overline{a}_{rs}} (A_B^{-1})_{r\cdot} = \text{lex}-\min \left\{ \frac{1}{\overline{a}_{is}} (A_B^{-1})_{i\cdot} \mid \overline{a}_{is} > 0, \; i \in \{1, \dots, m\} \right\}.$$

□

Die Anwendung der 2. Lexikographischen Regel garantiert auch Endlichkeit des Simplexverfahrens, analog zu Satz (10.6) (siehe Kall oder Dantzig S. 269, diese folgt aus der Störungsmethode).

(10.8) Weitere Zeilenauswahlregeln.

(1) **Kleinster-Index-Regel:** $r := \min R$.

(2) **Kleinster-Variablenindex-Regel:** Wähle $r \in R$, so dass $p_r = \min\{p_i \in B \mid i \in R\}$.

Keine der beiden Regeln in (10.8) kann für sich allein Endlichkeit des Simplexverfahrens bewirken. Aber eine Kombination von (10.8) (2) und (10.2) (2) schafft es.

(10.9) Bland-Regel: R. Bland ("New finite pivoting rules for the simplex method", Mathematics of Operations Research 2 (1977), 103–107) hat gezeigt, dass das Simplexverfahren auch endlich ist, wenn sowohl bei der Spaltenauswahl (PRICE-Routine) als auch bei der Zeilenauswahl die Kleinster-Variablen-Index Regel angewendet wird.

$\square$

Von Avis und Chvátal ("Notes on Bland's pivoting rule", Mathematical Programming Studies 8 (1978), 24–34) sind Untersuchungen über die Praktikabilität der Bland-Regel gemacht worden. Dabei hat sich gezeigt, dass i. a. die Anzahl der Pivotoperationen bei Anwendung der Bland-Regel wesentlich höher liegt als z. B. bei der Steilster-Anstieg Regel (10.2) (3). Für Computerimplementierungen scheint die Bland-Regel selbst bei hochdegenerierten Problemen nicht gut geeignet zu sein.

(10.10) Worst-Case Verhalten des Simplexverfahrens. Zu fast allen bekannten Pivotauswahlregeln (speziell zu allen, die hier genannt wurden) kennt man heute eine Klasse von Polyedern und Zielfunktionen, so dass der Simplexalgorithmus bei Anwendung einer zugehörigen Auswahlregel durch alle Ecken des Polyeders läuft. Da die Anzahl der Ecken dieser Polyeder exponentiell mit der Anzahl der Variablen wächst, sagt man, dass das Simplexverfahren ein exponentielles worst-case Verhalten besitzt. Es ist jedoch ein offenes Problem, ob es eine Pivot-Auswahlregel gibt, für die das Simplexverfahren polynomial ist.

10.3 Die Behandlung oberer Schranken

In linearen Programmen kommen häufig Beschränkungen der folgenden Form vor:

$$0 \leq x \leq u.$$

Da diese strukturell sehr einfach sind, kann man sie algorithmisch besser behandeln als allgemeine Ungleichungen. Normalerweise führen wir Ungleichungen durch Einfügung von Schlupfvariablen wie folgt in Gleichungen und Nichtnegativitätsbedingungen über:

$$x + \overline{x} = u, \ x \geq 0, \overline{x} \geq 0.$$

Ist jedoch der Wert von x (bzw. $\overline{x}$) festgelegt, so ist der Wert der Komplementärvariablen $\overline{x}$ (bzw. x) bereits eindeutig bestimmt. Diesen Vorteil kann man nun so ausnutzen, dass man im Simplexverfahren nur eine der beiden Variablen mitschleppt und die zusätzliche Gleichung völlig weglässt. Für jede Zeile oder Spalte muss jedoch festgehalten werden, ob sie x oder der Komplementärvariablen $\overline{x} = u - x$ entspricht. Die Zeilenauswahlregeln sind ebenfalls etwas komplizierter, da eine neue in die "Basis" hineinzunehmende Variable nur maximal bis zu ihrer Beschränkung erhöht werden darf und die übrigen Basisvariablen ebenfalls ihre Schranken nicht überschreiten dürfen.

Wir wollen im folgenden die sogenannte **Obere-Schranken-Technik** zur Behandlung von linearen Programmen der Form

$$
\begin{aligned}
&\max c^T x \\
&Ax = b \\
&0 \le x \le u
\end{aligned}
\tag{10.11}
$$

besprechen. Diese "upper-bound-technique" behandelt nur explizit nach oben beschränkte Variablen. Sind einige der Variablen nicht explizit nach oben beschränkt, so legen wir, um die weitere Diskussion und die Formeln zu vereinfachen, fest, dass ihre obere Schranke $+\infty$ ist. Das heißt, die Komponenten von u haben Werte aus der Menge $\mathbb{R}_+ \cup \{+\infty\}$.

Weiter benötigen wir eine erweiterte Definition von Basis und Nichtbasis. Wir halten uns im Prinzip an Konvention (9.3), lassen jedoch zu, dass der Vektor der Indizes von Nichtbasisvariablen positive und negative Indizes enthalten kann. Durch das Vorzeichen wollen wir uns merken, ob eine Nichtbasisvariable die obere oder untere Schranke annimmt, und zwar legen wir fest, dass für eine Nichtbasisvariable q_s der Wert x_{q_s} Null ist, falls das Vorzeichen von q_s positiv ist, andernfalls ist der Wert von x_{q_s} die obere Schranke u_{q_s}. Um dies formeltechnisch einfach aufschreiben zu können, treffen wir die folgenden Vereinbarungen. Ist $B = (p_1, \dots, p_m)$ ein Spaltenindexvektor, so dass A_B eine Basis ist und $N = (q_1, \dots, q_{n-m})$ wie in (9.3) definiert, dann sei

$$
\overline{N} := (\overline{q}_1, \dots, \overline{q}_{n-m}) \text{ mit } \overline{q}_i = q_i \text{ oder } \overline{q}_i = -q_i
\tag{10.12}
$$

$$
\begin{aligned}
N^+ &:= (q_i \mid \overline{q}_i > 0) \\
N^- &:= (q_i \mid \overline{q}_i < 0)
\end{aligned}
\tag{10.13}
$$

Die in B, N^+ und N^- enthaltenen Indizes bilden also eine Partition der Spaltenindexmenge $\{1, \dots, n\}$ von A. Bei einer Rechnerimplementation genügt es natürlich, sich $\overline{N}$ zu merken, denn $N = (|\overline{q}_1|, \dots, |\overline{q}_{n-m}|)$. Ist A_B regulär, so nennen wir A_B eine E-**Basis** (erweiterte Basis) von A und $\overline{N}$ eine E-**Nichtbasis** von

A. Der Vektor $x \in \mathbb{K}^n$ mit

$$(10.14) \qquad \begin{aligned} x_{N^+} &= 0 \\ x_{N^-} &= u_{N^-} \\ x_B &= A_B^{-1} b - A_B^{-1} A_{N^-} u_{N^-} \end{aligned}$$

heißt E**-Basislösung** zur E-Basis A_B.

$$\text{Eine} \left\{ \begin{array}{l} E\text{-Basis } A_B \\ E\text{-Basislösung } x \end{array} \right\} \text{heißt } \textbf{zulässig}\text{, wenn}$$

$0 \leq x_B \leq u_B$ und heißt **nicht degeneriert** (**nicht entartet**), falls $0 < x_B < u_B$, andernfalls **degeneriert** (**entartet**). Gibt es keine oberen Schranken, so gilt offenbar $N = \overline{N} = N^+$ und alle Formeln reduzieren sich auf den bereits behandelten Fall.

(10.15) Satz. *Gegeben sei ein Polyeder* $P := \{x \in \mathbb{K}^n \mid Ax = b, 0 \leq x \leq u\}$ *mit* $\operatorname{rang}(A) = m$. *Ein Vektor* $x \in \mathbb{K}^n$ *ist genau dann eine Ecke von* P, *wenn* x *zulässige* E*-Basislösung ist.*

Beweis : Es gilt $P = P(D, d)$ mit

$$D = \begin{pmatrix} A \\ -A \\ I \\ -I \end{pmatrix} \quad d = \begin{pmatrix} b \\ -b \\ u \\ 0 \end{pmatrix}.$$

Also folgt mit Satz (8.9): x Ecke von $P \iff \operatorname{rang}(D_{\mathrm{eq}(\{x\})}) = n$. Seien $J = \mathrm{eq}(\{x\})$ und J_1, J_2, J_3, J_4 so gewählt, dass

$$D_{J.} = \left[\begin{array}{c|c|c} \begin{matrix} A_{J_1} \\ -A_{J_2} \end{matrix} & \multicolumn{2}{c}{*} \\ \hline 0 & I_{J_3} & 0 \\ \hline 0 & 0 & -I_{J_4} \end{array}\right]$$

Gilt $\operatorname{rang}(D_{J.}) = n$, so besitzt $A_{J_1 \cup J_2 .}$ vollen Rang und mit $K := \{1, \ldots, n\} \setminus (J_3 \cup J_4)$ ist $A_B := A_{J_1 \cup J_2, K}$ regulär. Man verifiziert sofort, dass A_B zulässig ist, wenn $N^- = J_3$, $N^+ = J_4$ gewählt wird. Die Rückrichtung ist nun evident. $\square$

Wir kennzeichnen also im Weiteren Basen, Nichtbasen und Basislösung von (10.11) mit einem E, um sie von den in Kapitel 9 eingeführten Begriffen zu unterscheiden.

(10.16) Algorithmus. *(Upper-Bound-Technique)*

zur Lösung von linearen Programmen der Form (10.11).

Wie nehmen an, dass $u \in (\mathbb{R}_+ \cup \{+\infty\})^n$ gilt und dass eine zulässige E-Basis A_B vorliegt. Wie beschreiben lediglich die Phase II. Es seien also gegeben: zulässige E - Basis A_B, A_B^{-1}, $\overline{b} = A_B^{-1}b$, c, $B = (p_1, \dots, p_m)$ und $\overline{N} = (\overline{q}_1, \dots, \overline{q}_{n-m})$ (N, N^+, N^- können daraus bestimmt werden), $\overline{\overline{b}} = A_B^{-1}b - A_B^{-1}A_{N^-}u_{N^-}$.

(1) BTRAN:

Berechne $\pi^T := c_B^T A_B^{-1}$.

(2) PRICE:

Berechne $\overline{c}^T := c_N^T - \pi^T A_N$ und wähle ein $s \in \{1, \dots, n-m\}$ mit

$$\overline{c}_s \begin{cases} > 0 & \textit{falls} \quad \overline{q}_s > 0 \\ < 0 & \textit{falls} \quad \overline{q}_s < 0 \end{cases}$$

mittels irgendeiner der Pivotspaltenauswahlregeln. Gibt es keinen solchen Index s, so ist die aktuelle E-Basislösung optimal.

Begründung: Offenbar gilt $x \in \{y \in \mathbb{K}^n \mid Ay = b, 0 \le y \le u\} \iff x_B = A_B^{-1}b - A_B^{-1}A_{N^+}x_{N^+} - A_B^{-1}A_{N^-}x_{N^-}$ und $0 \le x_B, x_{N^+}, x_{N^-} \le u$. Also ist

$$\begin{aligned} c^T x &= c_B^T x_B + c_{N^+}^T x_{N^+} + c_{N^-}^T x_{N^-} \\ &= c_B^T A_B^{-1} b + (c_{N^+}^T - c_B^T A_B^{-1} A_{N^+}) x_{N^+} + (c_{N^-}^T - c_B^T A_B^{-1} A_{N^-}) x_{N^-} \\ &= \pi^T b + (c_{N^+}^T - \pi^T A_{N^+}) x_{N^+} + (c_{N^-}^T - \pi^T A_{N^-}) x_{N^-}. \end{aligned}$$

Da für die gegenwärtige E-Basislösung gilt $x_{N^-} = u_{N^-}$, $x_{N^+} = 0$, kann der Zielfunktionswert nur verbessert werden, wenn für eine Nichtbasisvariable q_s entweder gilt $\overline{q}_s > 0$ und $\overline{c}_s = (c_N^T - \Pi^T A_N)_s > 0$ oder $\overline{q}_s < 0$ und $\overline{c}_s < 0$.

(3) FTRAN:

Berechne $\overline{d} := A_B^{-1} A_{\cdot q_s} = \overline{A}_{\cdot s}$.

(4) CHUZR:

Setze $\sigma = \operatorname{sign}(\overline{q}_s)$ und berechne

$$\begin{aligned} \lambda_0 &:= \min\{\tfrac{\overline{\overline{b}}_i}{\sigma \overline{a}_{is}} \mid \sigma \overline{a}_{is} > 0,\ i = 1, \dots, m\} \\ \lambda_1 &:= \min\{\tfrac{\overline{\overline{b}}_i - u_{p_i}}{\sigma \overline{a}_{is}} \mid \sigma \overline{a}_{is} < 0,\ i = 1, \dots, m\} \\ \lambda_2 &:= u_{q_s} \end{aligned}$$

(a) *Ist keine der Zahlen λ_0, λ_1, λ_2 endlich, so ist das Programm (10.11) unbeschränkt (das kann natürlich nur vorkommen, wenn x_{q_s} nicht explizit beschränkt ist).*

(b) *Ist $\lambda_0 = \min\{\lambda_0, \lambda_1, \lambda_2\}$, so wähle einen Index*

$$r \in \{i \in \{1, \dots, m\} \mid \frac{\overline{\overline{b}}_i}{\sigma \overline{a}_{is}} = \lambda_0, \sigma \overline{a}_{is} > 0\}$$

mittels irgendeiner Pivotzeilenauswahlregel.

(c) *Ist $\lambda_1 = \min\{\lambda_0, \lambda_1, \lambda_2\}$ so wähle*

$$r \in \{i \in \{1, \dots, m\} \mid \frac{\overline{\overline{b}}_i - u_{p_i}}{\sigma \overline{a}_{is}} = \lambda_1, \sigma \overline{a}_{is} < 0\}$$

mittels irgendeiner Pivotzeilenauswahlregel.

(d) *Ist $\lambda_2 = \min\{\lambda_0, \lambda_1, \lambda_2\}$ so setze $\overline{q}_s := -\overline{q}_s$, berechne $\overline{\overline{b}}$ neu und gehe zurück zur PRICE-Routine (2).*

(5) WRETA:

Setze $B' = (p_1, \dots, p_{r-1}, q_s, p_{r+1}, \dots, p_m)$, $\overline{N} = (\overline{q}_1, \dots, \overline{q}_{s-1}, \overline{q}, \overline{q}_{s+1}, \dots, \overline{q}_{n-m})$, wobei $\overline{q} = p_r$, falls $\lambda_0 = \min\{\lambda_0, \lambda_1, \lambda_2\}$, bzw. $\overline{q} = -p_r$, falls $\lambda_1 = \min\{\lambda_0, \lambda_1, \lambda_2\}$. Berechne $A_{B'}^{-1}$, $\overline{\overline{b}} = A_{B'}^{-1} b - A_{B'}^{-1} A_{N^-} u_{N^-}$.

Begründung: Die Pivotzeilenauswahl (CHUZR-Routine) dient dazu, die Zeilenauswahl so zu bestimmen, dass die transformierten Variablen nach erfolgter Pivotoperation wieder zulässig sind. Entsprechend der Transformationsregeln beim Pivotisieren mit dem Pivotelement $\overline{a}_{rs}$ (vergleiche (9.10)) muss gewährleistet sein, dass nach Ausführung des Pivotschrittes für die neue E-Basislösung folgendes gilt:

$\alpha)$ $0 \leq x'_{p_i} = \overline{\overline{b}}_i - \frac{\overline{a}_{is}}{\overline{a}_{rs}} \overline{\overline{b}}_r \leq u_{p_i}$, für $i \neq r$,

$\beta)$ $0 \leq x'_{q_s} = \frac{1}{\overline{a}_{rs}} \overline{\overline{b}}_r \leq u_{q_s}$,

$\gamma)$ $x'_{p_r} \in \{0, u_{p_r}\}$,

δ) $x'_{q_i} \in \{0, u_{q_i}\}$, für $i \neq s$.

Wollen wir nun den Wert der Variablen x_{q_s} um $\lambda \geq 0$ ändern (d. h. erhöhen, falls $\overline{q}_s > 0$, bzw. um λ erniedrigen, falls $\overline{q}_s < 0$), so können wir das so lange tun bis entweder

– eine Basisvariable den Wert ihrer oberen oder unteren Schranke annimmt oder

– die Nichtbasisvariable q_s den Wert ihrer anderen Schranke annimmt.

Aus letzterem folgt natürlich, dass $\lambda \leq u_{q_s}$ gelten muss, ansonsten würden wir bei Erhöhung ($\overline{q}_s > 0$) die obere Schranke bzw. bei Erniedrigung ($\overline{q}_s < 0$) die untere Schranke für x_{q_s} verletzen. Dies erklärt die Bestimmung von λ_2 in (4).

Ist $\overline{q}_s > 0$, so bewirkt eine Erhöhung von x_{q_s} um den Wert λ, dass $x'_{p_i} = \overline{\overline{b}}_i - \lambda\overline{a}_{is}$ gilt. Aus der Bedingung α) erhalten wir daher die folgenden Schranken für λ:

$$\begin{aligned} \lambda &\leq \frac{\overline{\overline{b}}_i}{\overline{a}_{is}} && \text{für } \overline{a}_{is} > 0 \\ \lambda &\leq \frac{\overline{\overline{b}}_i - u_{p_i}}{\overline{a}_{is}} && \text{für } \overline{a}_{is} < 0. \end{aligned}$$

Ist $\overline{q}_s < 0$, so bewirkt die Verminderung von $x_{q_s} (= u_{q_s})$ um den Wert $\lambda \geq 0$, dass $x'_{p_i} = \overline{\overline{b}}_i + \lambda\overline{a}_{is}$ gilt. Aus α) erhalten wir somit:

$$\begin{aligned} \lambda &\leq -\frac{\overline{\overline{b}}_i}{\overline{a}_{is}} && \text{für } \overline{a}_{is} < 0 \\ \lambda &\leq \frac{u_{p_i} - \overline{\overline{b}}_i}{\overline{a}_{is}} && \text{für } \overline{a}_{is} > 0. \end{aligned}$$

Dies begründet die Definition von λ_0 und λ_1. Sei nun $\lambda_{\min} = \min\{\lambda_0, \lambda_1, \lambda_2\}$. Gilt $\lambda_{\min} = \lambda_2$, so wird einfach der Wert einer Nichtbasisvariablen von der gegenwärtigen Schranke auf die entgegengesetzte Schranke umdefiniert. Die Basis ändert sich dadurch nicht, jedoch konnte aufgrund der Auswahl in PRICE eine Zielfunktionsverbesserung erreicht werden. Gilt $\lambda_{\min} = \lambda_0$ oder $\lambda_{\min} = \lambda_1$, so führen wir einen üblichen Pivotschritt durch. Bei $\lambda_{\min} = \lambda_0$ wird die neue Nichtbasisvariable x_{p_r} mit Null festgesetzt, da die untere Schranke von x_{p_r} die stärkste Einschränkung für die Veränderung von x_{q_s} darstellte. Bei $\lambda_{\min} = \lambda_1$, wird analog x_{p_r} eine Nichtbasisvariable, die den Wert ihrer oberen Schranke annimmt. □

Wir wollen nun anhand eines Beispiels die Ausführung von Algorithmus (10.16) in Tableautechnik demonstrieren. Dabei werden wir keine neuen Tableau-Update-Formeln definieren, sondern mit den bekannten Formeln für die Pivotschritte rechnen. Wir führen lediglich eine zusätzliche Spalte ein, in der jeweils aus $\overline{b}$ und c_0 der

Wert der gegenwärtigen E-Basislösung berechnet wird. Ist also c_0 der gegenwärtige Wert der Basislösung und $x_B = A_B^{-1} b = \overline{b}$ so hat die zusätzliche Spalte in der ersten Komponente den Eintrag $\overline{\overline{c}}_0 := -c_0 - \overline{c}_{N^-} u_{N^-}$, und die restlichen Komponenten berechnen sich durch

$$\overline{\overline{b}} := \overline{b} - \overline{A}_{N^-} u_{N^-}.$$

(10.17) Beispiel. Wir betrachten das folgende LP, dessen Lösungsmenge in Abbildung 10.1 dargestellt ist

$$\begin{array}{ll}
\max\ 2x_1 + x_2 & \\
-x_1 + x_2 & \leq \frac{1}{2} \\
\quad x_1 + x_2 & \leq 2 \\
\quad x_1 & \leq \frac{3}{2}\ (= u_1) \\
x_2 & \leq 1\ (= u_2) \\
x_1, x_2 & \geq 0.
\end{array}$$

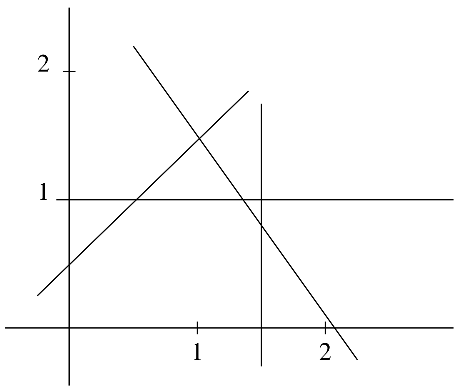


Abb. 10.1

Wir führen 2 Schlupfvariablen x_3, x_4 ein (mit oberen Schranken $+\infty$) und erhalten als Anfangstableau T_0 (mit zusätzlicher Spalte $\binom{\overline{\overline{c}}_0}{\overline{\overline{b}}}$, die mit der Spalte $\binom{-c_0}{\overline{b}}$ identisch ist, da keine Nichtbasisvariable von Null verschieden ist):

$$T_0: \quad \begin{array}{c|cccc|c|c}
 & \mathit{1} & \mathit{2} & \mathit{3} & \mathit{4} & & \\
 & 2 & 1 & 0 & 0 & 0 & 0 \\ \hline
\mathit{3} & -1 & 1 & 1 & 0 & \frac{1}{2} & \frac{1}{2} \\
\mathit{4} & 1 & 1 & 0 & 1 & 2 & 2
\end{array}
\quad = \quad
\begin{array}{cc|c|c}
\overline{c} & 0 & -c_0 & \overline{\overline{c}}_0 \\ \hline
\overline{A} & I & \overline{b} & \overline{\overline{b}}
\end{array}
\quad
\begin{array}{l}
B = (3,4) \\
\overline{N} = (1,2)
\end{array}$$

Wir wählen die erste Spalte als Pivotspalte, d. h. $\overline{q}_1 = 1$. Schritt (4) CHUZR ergibt

$$\lambda_0 = \frac{\overline{\overline{b}}_2}{\overline{a}_{21}} = \frac{2}{1} = 2,\ \lambda_1 \text{ ist wegen } u_3 = \infty \text{ nicht definiert },\ \lambda_2 = u_1 = \frac{3}{2},$$

d. h. $\lambda_{\min} = \lambda_2$. Wir führen also Schritt (4) (d) aus, setzen $\overline{q}_1 := -\overline{q}_1$, d. h. $\overline{q}_1 = -1$, und berechnen $\overline{\overline{b}}$ neu. Die neue letzte Spalte, d. h. die neue E-Basislösung zur Basis A_B erhalten wir aus der (normalen) Basislösung wegen $N^- = (1)$ wie folgt:

$$\overline{\overline{b}} = \overline{b} - \overline{A}_{.1} u_1 = \begin{pmatrix} \frac{1}{2} \\ 2 \end{pmatrix} - \frac{3}{2} \begin{pmatrix} -1 \\ 1 \end{pmatrix} = \begin{pmatrix} 2 \\ \frac{1}{2} \end{pmatrix},$$

und der Zielfunktionswert erhöht sich um $\overline{c}_1.u_1 = 2 \cdot \frac{3}{2} = 3$. Mithin erhalten wir als nächstes Tableau (mit $\overline{q}_1 := -\overline{q}_1$)

$$\begin{array}{llrrrr|c|c}
 & & \mathit{-1} & \mathit{2} & \mathit{3} & \mathit{4} & & \\
 & & 2 & 1 & 0 & 0 & 0 & -3 \\ \cline{3-8}
T_1 : & \mathit{3} & -1 & 1 & 1 & 0 & \frac{1}{2} & 2 \\
 & \mathit{4} & 1 & 1 & 0 & 1 & 2 & \frac{1}{2}
\end{array}
\qquad
\begin{array}{l}
B = (3,4) \\
\overline{N} = (-1,2)
\end{array}$$

Als Pivotspalte kommt nur die 2. Spalte ($\overline{q}_2 = 2$) in Frage. CHUZR ergibt

$$\lambda_0 = \frac{\overline{\overline{b}}_2}{\overline{a}_{22}} = \frac{1}{2}, \ \lambda_1 \text{ nicht definiert}\ , \ \lambda_2 = u_2 = 1.$$

Die Wahl der Austauschzeile in (4) (b) ergibt ein eindeutig bestimmtes $r = 2$. Wir führen dann (5) in Tableautechnik aus. Wir führen also einen Standardpivotschritt auf dem Element $\overline{a}_{22} = 1$ durch und korrigieren anschließend $\overline{b}$ und c_0, um $\overline{\overline{b}}$ und $\overline{\overline{c}}_0$ zu erhalten.

$$\begin{array}{llrrrr|c|c}
 & & \mathit{-1} & \mathit{2} & \mathit{3} & \mathit{4} & & \\
 & & 1 & 0 & 0 & -1 & -2 & -\frac{7}{2} \\ \cline{3-8}
T_2 : & \mathit{3} & -2 & 0 & 1 & -1 & -\frac{3}{2} & \frac{3}{2} \\
 & \mathit{2} & 1 & 1 & 0 & 1 & 2 & \frac{1}{2}
\end{array}
\qquad
\begin{array}{l}
B = (3,2) \\
\overline{N} = (-1,4)
\end{array}$$

Dieses Tableau ist optimal, mit $x_1 = \frac{3}{2}$, $x_2 = \frac{1}{2}$. Zur Erläuterung der übrigen Fälle beginnen wir noch einmal mit Tableau T_0, wählen aber die 2. Spalte als Pivotspalte. CHUZR ergibt

$$\lambda_0 = \frac{\overline{\overline{b}}_1}{\overline{a}_{12}} = \frac{1}{2}, \ \lambda_1 \text{ nicht definiert}\ , \ \lambda_2 = 1.$$

In diesem Falle machen wir einen Standardpivotschritt mit dem Pivotelement

$\overline{a}_{rs} = \overline{a}_{12} = 1.$

$$T_3: \quad \begin{array}{c|rrrr|r|r} & 1 & 2 & 3 & 4 & & \\ & 3 & 0 & -1 & 0 & -\frac{1}{2} & -\frac{1}{2} \\ \hline 2 & -1 & 1 & 1 & 0 & \frac{1}{2} & \frac{1}{2} \\ 4 & 2 & 0 & -1 & 1 & \frac{3}{2} & \frac{3}{2} \end{array} \qquad \begin{array}{l} B = (2,4) \\ \overline{N} = (1,3) \end{array}$$

Die gegenwärtige Basislösung wird wie üblich in eine E-Basislösung transformiert. Als Pivotspalte kommt nur die erste in Frage. CHUZR ergibt

$$\lambda_0 = \frac{\overline{\overline{b}}_2}{\overline{a}_{21}} = \frac{3}{4},\ \lambda_1 = \frac{\overline{\overline{b}}_1 - u_2}{\overline{a}_{11}} = \frac{1}{2},\ \lambda_2 = u_1 = \frac{3}{2}\ (\lambda_{\min} = \lambda_1)$$

Wir führen nun Schritt (5) des Algorithmus (10.6) durch einen Pivotschritt auf $\overline{a}_{rs} = \overline{a}_{11} = -1$ aus und berechnen das gesamte Tableau neu. Anschließend müssen wir die rechteste Spalte des Tableaus korrigieren. Da $\lambda_{\min} = \lambda_1$ wird die neue Nichtbasisvariable x_2 mit ihrer oberen Schranke $u_2 = 1$ belegt (und $\overline{q}_1 = -2$ gesetzt). Wir müssen daher vom neuberechneten $\overline{b} = (-\frac{1}{2}, \frac{5}{2})^T$ noch das u_2-fache der zum Index 2 gehörigen Spalte von $\overline{A}$ also $\overline{A}_{.1} = (-1, 2)^T$) subtrahieren. Vom Zielfunktionswert c_0 subtrahieren wir $u_2 \cdot \overline{c}_2 = 1 \cdot 3$. Daraus ergibt sich

$$T_4: \quad \begin{array}{c|rrrr|r|r} & 1 & -2 & 3 & 4 & & \\ & 0 & 3 & 2 & 0 & 1 & -2 \\ \hline 1 & 1 & -1 & -1 & 0 & -\frac{1}{2} & \frac{1}{2} \\ 4 & 0 & 2 & 1 & 1 & \frac{5}{2} & \frac{1}{2} \end{array} \qquad \begin{array}{l} B = (1,4) \\ \overline{N} = (-2,3) \end{array}$$

$$\overline{\overline{b}} = \begin{pmatrix} -\frac{1}{2} \\ \frac{5}{2} \end{pmatrix} - 1 \begin{pmatrix} -1 \\ 2 \end{pmatrix}.$$

Als Pivotspalte müssen wir nun die 3. Spalte von T_4, also $\overline{A}_{.2}$, wählen.

$$\lambda_0 = \frac{\overline{\overline{b}}_2}{\overline{a}_{22}} = \frac{1}{2},\ \lambda_1 = \frac{\overline{\overline{b}}_1 - u_1}{\overline{a}_{12}} = 1,\ \lambda_2 \text{nicht definiert.}$$

Somit führen wir einen Standardpivotschritt auf $\overline{a}_{22} = 1$ aus

$$T_5: \quad \begin{array}{c|rrrr|r|r} & 1 & -2 & 3 & 4 & & \\ & 0 & -1 & 0 & -2 & -4 & -3 \\ \hline 1 & 1 & 1 & 0 & 1 & 2 & 1 \\ 3 & 0 & 2 & 1 & 1 & \frac{5}{2} & \frac{1}{2} \end{array} \qquad \begin{array}{l} B = (1,3) \\ \overline{N} = (-2,4) \end{array}$$

Als Pivotspalte kommt nur $\overline{A}_{.1}$, also die 2. Spalte von T_5, in Frage, d. h. $q_s = q_1 = 2$, $\overline{q}_1 = -2$. Wir erhalten durch CHUZR

$$\lambda_0 \text{ undefiniert}, \lambda_1 = \frac{\overline{\overline{b}}_1 - u_1}{\sigma \overline{a}_{11}} = \frac{1}{2}, \ \lambda_2 = u_2 = 1.$$

Unser Pivotelement ist somit $\overline{a}_{11} = 1$. Wie üblich führen wir einen Standardpivotschritt durch. Da wir die neue Nichtbasisvariable x_1 mit ihrer oberen Schranke $u_1 = \frac{3}{2}$ belegen, ziehen wir wieder vom neuberechneten $\overline{b}$ das u_1-fache der zu 1 gehörigen Spalte von $\overline{A}$ (das ist $\overline{A}_{.1} = (1, -2)^T$) ab, um $\overline{\overline{b}}$ zu erhalten.

$$\begin{array}{cc|rrrr|r|r}
 & & \mathit{-1} & \mathit{2} & \mathit{3} & \mathit{4} & & \\
 & & 1 & 0 & 0 & -1 & -2 & -\frac{7}{2} \\
\hline
T_6 : & \mathit{2} & 1 & 1 & 0 & 1 & 2 & \frac{1}{2} \\
 & \mathit{3} & -2 & 0 & 1 & -1 & -\frac{3}{2} & \frac{3}{2}
\end{array} \qquad \begin{array}{l} B = (2,3) \\ \overline{N} = (-1, 4) \end{array}$$

Wir haben wieder die Optimallösung erreicht. $\square$

Analog zu oberen Schranken können natürlich auch untere Schranken behandelt werden. Ferner kann man auch allgemeine obere Schranken (generalized upper bounds, GUB), dies sind Restriktionen der folgenden Form: $\sum x_i \leq a$, auf ähnliche Weise berücksichtigen. Bei solchen Restriktionen können ebenfalls Pivotschemata entworfen werden, die wesentlich weniger Speicher- und Rechenaufwand erfordern als die Standardmethode. Weiterhin gibt es effiziente Techniken zur Behandlung von sogenannten variable upper bounds (VUB) der Form $0 \leq x \leq y$.

10.4 Das duale Simplexverfahren

Wir wollen nun eine Variante des Simplexverfahrens darstellen, die — wie wir noch sehen werden — bei bestimmten Problemstellungen von Vorteil ist. Sei A_B eine Basis von A, und betrachte das Paar dualer linearer Programme:

$$(P) \quad \begin{array}{l} \max c^T x \\ Ax = b \\ x \geq 0 \end{array} \qquad\qquad (D) \quad \begin{array}{l} \min u^T b \\ u^T A \geq c^T \end{array}$$

bzw.

$$\begin{array}{ll} \max c^T x & \\ x_B & = A_B^{-1} b - A_B^{-1} A_N x_N \\ x_N, x_B & \geq 0 \end{array} \qquad\qquad \begin{array}{ll} \min u^T b & \\ u^T A_B & \geq c_B^T \\ u^T A_N & \geq c_N^T \end{array}$$

(10.18) Definition. *Die Basis A_B von A heißt* **primal zulässig**, *falls $A_B^{-1}b \geq 0$ und* **dual zulässig**, *falls $\overline{c}^T = c_N^T - c_B^T A_B^{-1} A_N \leq 0$. Die zugehörigen Basislösungen x bzw. u mit $x_B = A_B^{-1}b$, $x_N = 0$ bzw. $u^T = c_B^T A_B^{-1}$ heißen dann* **primal** *bzw.* **dual zulässig**.

(10.19) Satz. *Sei $P = \{u \in \mathbb{K}^m \mid u^T A \geq c^T\}$. Der Vektor u ist genau dann eine Ecke von P, wenn u eine dual zulässige Basislösung ist.*

Beweis : Sei $\overline{u}$ Ecke von $P = P(-A^T, -c)$ und $I = \text{eq}(\{\overline{u}\})$. Mit Satz (8.9) folgt $\text{rang}((-A^T)_{I.}) = m$, d. h. es existiert $B \subseteq I$ mit A_B Basis von A, und es gilt $\overline{u}^T A_B = c_B^T$, $\overline{u}^T A_N \geq c_N^T$. Also ist $\overline{u}^T = c_B^T A_B^{-1}$ und $c_B^T A_B^{-1} A_N \geq c_N^T$, d. h. A_B ist dual zulässig. Ist umgekehrt A_B dual zulässig, so ist $\overline{u} := c_B^T A_B^{-1}$ aufgrund von Satz (8.9) eine Ecke von P. □

(10.20) Bemerkung. Ist A_B eine Basis von A, und sind x bzw. u die zu A_B gehörenden (primalen bzw. dualen aber nicht notwendigerweise zulässigen) Basislösungen, so gilt: $c^T x = u^T b$.

Beweis : $u^T b = c_B^T A_B^{-1} b = c_B^T x_B = c^T x$. □

(10.21) Folgerung. *Ist A_B eine Basis von A, so ist A_B optimal genau dann, wenn A_B primal und dual zulässig ist.*

Beweis : (Dualitätssatz).

(10.22) Folgerung. *Ist A_B eine optimale Basis für das Programm (P), dann ist $c_B^T A_B^{-1}$ eine optimale Lösung des zu (P) dualen Programms (D).*

□

Der Vektor $\pi := c_B^T A_B^{-1}$ (mit A_B dual zulässig) heißt der Vektor der **Schattenpreise** (vgl. ökonomische Interpretation der Dualität und Schritt (1) in (10.1)).

Wir wollen nun die dem dualen Simplexverfahren zugrunde liegende Idee entwickeln und bemerken, dass — im Prinzip — das duale Simplexverfahren das primale Simplexverfahren angewendet auf das duale Problem ist. Sei A_B eine Basis von A, so lässt sich (9.1) umformen in:

$$
(10.23) \qquad \begin{array}{ll}
\max c_B^T A_B^{-1} b + \overline{c}^T x_N & \\
\overline{A} x_N + I x_B & = A_B^{-1} b \; (= \overline{b}) \\
x_B, x_N & \geq 0
\end{array}
$$

oder

$$(10.24) \qquad \begin{array}{ll} c_B^T A_B^{-1} b + \max \overline{c}^T x_N & \\ \overline{A} x_N & \leq \overline{b} \\ x_N & \geq 0 \end{array}$$

Das zu (10.24) duale Programm lautet (bei Weglassung des konstanten Terms $c_B^T A_B^{-1} b$ der Zielfunktion):

$$(10.25) \qquad \begin{array}{ll} \min u^T \overline{b} & \\ \overline{A}^T u & \geq \overline{c} \\ u & \geq 0 \end{array}$$

Die Einführung von Schlupfvariablen y_N (und Variablenumbenennung $y_B := u$) ergibt:

$$(10.26) \qquad \begin{array}{ll} \max -\overline{b}^T y_B & \\ -\overline{A}^T y_B + I y_N & = -\overline{c} \\ y & \geq 0 \end{array}$$

Durch eine lange Kette von Umformungen ist es uns also gelungen, ein LP in Standardform (9.1) in ein anderes LP in Standardform (10.26) zu transformieren. Was haben wir gewonnen? Nicht viel, es sei denn, die Basis A_B ist **dual zulässig**. Denn in diesem Falle gilt, dass die rechte Seite von (10.26), also $-\overline{c}$, nichtnegativ ist. Also ist die Matrix I eine zulässige (Ausgangs-)basis für (10.26), und wir können auf (10.26) den Simplexalgorithmus (direkt mit Phase II beginnend) anwenden.

Eine besonders interessante Anwendung ergibt sich, wenn das ursprüngliche Problem die Form

$$\begin{array}{l} \max c^T x \\ Ax \leq b \\ x \geq 0 \end{array}$$

hat, wobei $c \leq 0$ und $b \ngeq 0$ ist. Hier ist eine dual zulässige Basis direkt gegeben, während eine primal zulässige Basis erst mittels Phase I gefunden werden müsste.

(10.27) Algorithmus (duale Simplexmethode).

Input: $A' \in \mathbb{K}^{(m,n)}$, $b' \in \mathbb{K}^m$, $c \in \mathbb{K}^n$.

Output: *optimale Lösung* x *des LP* $\max\{c^T x \mid A'x = b', x \geq 0\}$.

Phase I:

Bestimmung eines Subsystems $Ax = b$, $x \geq 0$ mit $P^=(A,b) = P^=(A',b')$, welches (9.2) erfüllt (falls möglich) und Bestimmung einer dual zulässigen Basis A_B von A. Berechne: $\overline{A} = A_B^{-1}A_N$, $\overline{b} = A_B^{-1}b$, $\overline{c}^T = c_N^T - c_B^T A_B^{-1} A_N$.
(Dieser Teil des Algorithmus wird analog zur Phase I (9.24) der Grundversion des Simplexalgorithmus ausgeführt.)

Phase II: *Optimierung*

(II.1) *(Optimalitätsprüfung)*
Gilt $\overline{b}_i \geq 0$ $(i = 1,\dots,m)$, so ist die gegenwärtige Basislösung optimal (A_B ist primal und dual zulässig). Setze $x_B = \overline{b}$ und $x_N = 0$, andernfalls gehe zu (II.2).

(II.2) *(Bestimmung der Pivotzeile)*
Wähle einen Index r mit $\overline{b}_r < 0$.

(II.3) *(Prüfung auf Beschränktheit des Optimums)*
Gilt $\overline{A}_{r.} \geq 0$, so hat das duale Programm keine endliche Lösung, also ist $P^=(A,b) = \emptyset$. STOP!

(II.4) *Berechne* $\lambda_0 := \min\left\{ \frac{\overline{c}_j}{\overline{a}_{rj}} \mid \overline{a}_{rj} < 0, j = 1,\dots,n \right\}$.

(II.5) *(Bestimmung der Pivotspalte)*
Wähle Index $s \in \{j \in \{1,\dots,n-m\} \mid \frac{\overline{c}_j}{\overline{a}_{rj}} = \lambda_0\}$.

(II.6) *(Pivotoperation)*
Setze $A_{B'}^{-1} := EA_B^{-1}$ mit E aus Satz (9.10), und berechne alle notwendigen Parameter neu. Gehe zu (II.1).

□

(10.28) Tableauform des dualen Simplexalgorithmus. Wollen wir das duale Simplexverfahren auf das Problem (10.23) (mit dual zulässiger Basis A_B) anwenden, so können wir das verkürzte Tableau

$$VT := \begin{array}{|c|c|} \hline -\overline{b}^T & -z_0 \\ \hline -\overline{A}^T & -\overline{c} \\ \hline \end{array}$$

verwenden und auf dieses Tableau die Update Formeln des verkürzten Simplextableaus (diese sind in Schritt (II.7) von (9.15) angegeben) anwenden. Jedoch zeigt eine einfache Überlegung, dass wir bei der Auswahl der Indizes r und s entsprechend (II.2) bzw. (II.5) die Updateformeln (II.7) aus (9.15) auch direkt auf die Matrix $\overline{A}$ bzw. $\overline{b}$ und $\overline{c}$ anwenden können und zwar genau in der Form wie sie in (9.15) (II.7) aufgeschrieben wurden (um das neue Tableau VT' zu erhalten).

$$VT' := \begin{array}{|c|c|} \hline -(\overline{b}')^T & -z_0' \\ \hline -(\overline{A}')^T & -\overline{c}' \\ \hline \end{array}$$

Wir führen also die Transponierung bzw. den Vorzeichenwechsel nicht explizit durch, sondern beachten den Vorzeichenwechsel bzw. die Transponierung implizit bei der Bestimmung der Pivotzeile bzw. Pivotspalte.

Dann könnten wir aber auch wieder das primale (verkürzte) Tableau verwenden und anhand dieses Tableaus sowohl den primalen als auch den dualen Simplexalgorithmus durchführen. $\square$

(10.29) Das Tucker-Tableau. Für die nachfolgende spezielle Form linearer Programme erhalten wir primale und duale Optimallösungen direkt aus der Tableaurechnung. Wir betrachten:

$$\begin{array}{llcll} & \max c^T x & \qquad & & \min u^T b \\ (P) & Ax \le b & & (D) & u^T A \ge c^T \\ & x \ge 0 & & & u \ge 0 \end{array}$$

oder mit Schlupfvariablen

$$\begin{array}{lrlcllr} \max z_0 & & & \qquad & \max z_0 & & \\ c^T x - z_0 & = & 0 & & u^T b + z_0 & = 0 & \\ Ax - b & = & -u & & -c^T + u^T A & = x^T & \\ x, u & \ge & 0 & & u, x & \ge 0 & \end{array}$$

oder in Tableauschreibweise

$$\begin{array}{|c|c|} \hline c_1, c_2, \ldots, c_n & z_0 \\ \hline & b_1 \\ A & \vdots \\ & b_m \\ \hline \end{array}$$

$$\begin{aligned} N &= \text{Nichtbasisvariable des primalen Programms}\\ &= \text{Basisvariable des dualen Programms}\\ B &= \text{Basisvariable des primalen Programms}\\ &= \text{Nichtbasisvariable des dualen Programms} \end{aligned}$$

Ein solches Tucker-Tableau heißt **primal zulässig**, wenn $b \geq 0$, und **dual zulässig**, wenn $c \leq 0$.

Führen wir den primalen oder dualen Simplexalgorithmus bis zum Optimaltableau durch, so sind die jeweiligen Optimallösungen gegeben durch:

$$\left.\begin{array}{ll} x_{B(i)} = \overline{b}_i & i = 1, \ldots, m \\ x_{N(i)} = 0 & i = 1, \ldots, n \end{array}\right\} \text{ optimale Lösung des primalen Programms.}$$

$$\left.\begin{array}{ll} u_{N(i)} = -\overline{c}_i & i = 1, \ldots, n \\ u_{B(i)} = \ \ 0 & i = 1, \ldots, m \end{array}\right\} \text{ optimale Lösung des dualen Programms.}$$

□

Hat man eine Optimallösung von einem LP in Standardform ausgehend berechnet, so lässt sich die Optimallösung des dualen Programms nicht ohne Weiteres aus dem Tableau ablesen.

(10.30) Beispiel (dualer Simplexalgorithmus). Wir betrachten die folgenden zueinander dualen Programme (P) und (D). Die Lösungsmenge von (P) sei mit P bezeichnet, die von (D) mit D. P und D sind in Abbildung 10.2 dargestellt.

$$(P)\quad \begin{array}{lll} \max -x_1 - 2x_2 & & \\ -x_1 - \ \ x_2 & \leq -2 & (x_3) \\ -2x_1 - \ \ x_2 & \leq -3 & (x_4) \\ x_1, x_2 & \geq \ \ 0 & \end{array} \qquad (D)\quad \begin{array}{lll} \min -2y_3 - 3y_4 & & \\ -y_3 - 2y_4 & \geq -1 & (-y_1) \\ -y_3 - \ \ y_4 & \geq -2 & (-y_2) \\ y_3, y_4 & \geq \ \ 0 & \end{array}$$

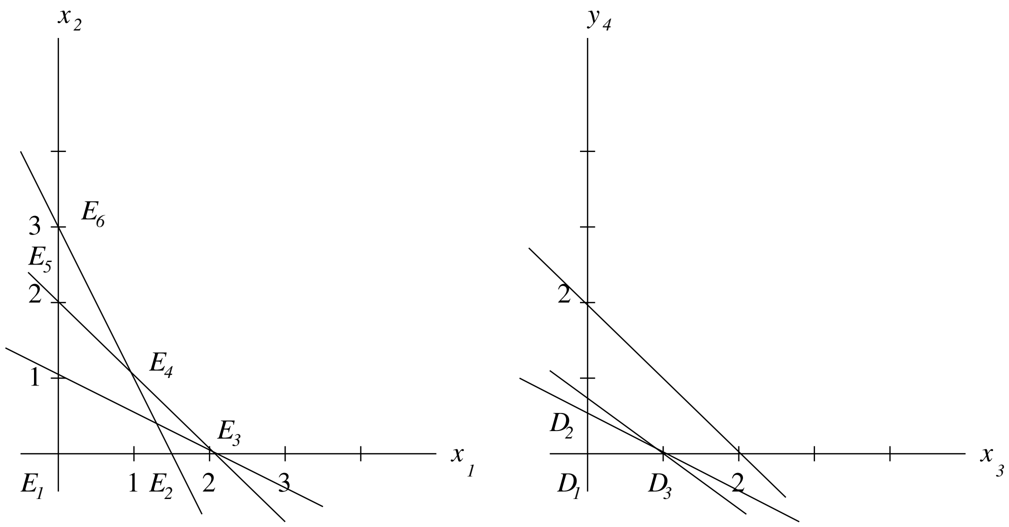


Abb. 10.2

Wir schreiben das Tucker-Tableau zu diesem Problem auf.

$$
\begin{array}{c|cc|c}
 & 1 & 2 & \\
 & -1 & -2 & 0 \\
\hline
3 & -1 & -1 & -2 \\
4 & -2 & -1 & -3
\end{array}
\qquad
\begin{array}{rcl}
x_1 &=& 0 \\
x_2 &=& 0 \\
x_3 &=& -2 \\
x_4 &=& -3
\end{array}
\qquad
\begin{array}{rcl}
y_3 &=& 0 \\
y_4 &=& 0 \\
y_1 &=& 1 \\
y_2 &=& 2
\end{array}
$$

$$
\begin{array}{rcl}
N = (1,2) &=& \text{primale Nichtbasis, duale Basis} \\
B = (3,4) &=& \text{primale Basis, duale Nichtbasis}
\end{array}
$$

Die gegenwärtige Basis ist dual aber nicht primal zulässig. Wir machen einen Update mit dem Pivotelement $\overline{a}_{rs} = \overline{a}_{21} = -2$. Das neue Tableau hat folgende Form.

$$
\begin{array}{c|cc|c}
 & 4 & 2 & \\
 & -\frac{1}{2} & -\frac{3}{2} & -\frac{3}{2} \\
\hline
3 & -\frac{1}{2} & -\frac{1}{2} & -\frac{1}{2} \\
1 & -\frac{1}{2} & \frac{1}{2} & \frac{3}{2}
\end{array}
\qquad
N = (4,2),\ B = (3,1)
$$

$$
\left.\begin{array}{rl}
x_1 &= \frac{3}{2} \\
x_2 &= 0
\end{array}\right\} = E_2
\qquad
\left.\begin{array}{rl}
y_3 &= 0 \\
y_4 &= \frac{1}{2}
\end{array}\right\} = D_2
$$

Es folgt ein Pivotschritt mit $\overline{a}_{rs} = \overline{a}_{11} = -\frac{1}{2}$.

$$\begin{array}{c|cc|c} & 3 & 2 & \\ & -1 & -1 & -2 \\ \hline 4 & -2 & 1 & 1 \\ 1 & -1 & 1 & 2 \end{array} \qquad N = (3,2),\ B = (4,1)$$

$$\left.\begin{array}{ll} x_1 & = 2 \\ x_2 & = 0 \end{array}\right\} = E_3 \qquad \left.\begin{array}{ll} y_3 & = 1 \\ y_4 & = 0 \end{array}\right\} = D_3$$

Dieses Tableau ist optimal. □

10.5 Zur Numerik des Simplexverfahrens

Dieser Abschnitt ist nur äußerst kursorisch und oberflächlich ausgearbeitet. Eine sorgfältige Behandlung des Themas würde eine gesamte Spezialvorlesung erfordern. Wir empfehlen dem Leser die Bemerkungen zur Numerik in

V. Chvátal, "Linear Programming", Freeman, New York, 1983, (80 SK 870 C 564)

oder das Buch

M. Bastian, "Lineare Optimierung großer Systeme", Athenäum/Hain/Skriptor/Hanstein, Königstein, 1980, (80 ST 130 B 326)

zu lesen, das einem Spezialaspekt dieses Themas gewidmet ist und diesen einigermaßen erschöpfend behandelt. Das Buch

B. A. Murtagh, "Advanced Linear Programming: Computation and Practice", McGraw-Hill, New York, 1981, (80 QH 400 M 984).

ist ganz Rechen- und Implementierungstechniken der linearen Optimierung gewidmet. Generelle Methoden zur Behandlung spezieller (z. B. dünn besetzter oder triangulierter oder symmetrischer) Matrizen im Rahmen numerischer Verfahren (etwa Gauß-Elimination, Matrixinvertierung) finden sich in

S. Pissanetsky, "Sparse Matrix Technology", Academic Press, London, 1984, (80 SK 220 P 673).

Generell wird bei Implementationen die revidierte Simplexmethode verwendet. Es gibt zwei grundsätzliche Varianten für die Reinversion: **Produktform der Inversen (PFI)**, **Eliminationsform der Inversen (EFI)**.

(10.31) Produktform der Inversen. *B^{-1} liegt in Produktform vor: $B^{-1} = E_k \cdot E_{k-1} \cdot \ldots \cdot E_1$ mit E_i aus Satz (9.10).*

Prinzip: *Gauß-Jordan Verfahren.*

Freiheitsgrade:

a) Positionen der Pivotelemente,

b) Reihenfolge der Pivots.

a) hat Einfluss auf die numerische Stabilität

b) hat Einfluss auf die Anzahl NNE (nicht Null Elemente) im Etafile (Speicherung der Spalten η von E_i).

Beispiel.

$$B = \begin{pmatrix} 1 & 1 & 1 & 1 \\ -1 & 1 & 0 & 0 \\ -1 & 1 & 0 & 0 \\ -1 & 0 & 0 & 1 \end{pmatrix}$$

besitzt folgende Produktform-Darstellungen der Inversen:

a) Pivotisieren auf der Hauptdiagonalen von “links oben” nach “rechts unten” ergibt

$$B^{-1} = \begin{pmatrix} 1 & & & -\frac{1}{4} \\ & 1 & & -\frac{1}{4} \\ & & 1 & -\frac{1}{4} \\ & & & \frac{3}{4} \end{pmatrix} \begin{pmatrix} 1 & & -\frac{1}{3} & \\ & 1 & -\frac{1}{3} & \\ & & \frac{2}{3} & \\ & & -\frac{1}{3} & 1 \end{pmatrix} \begin{pmatrix} 1 & -\frac{1}{2} & & \\ & \frac{1}{2} & & \\ & -\frac{1}{2} & 1 & \\ & -\frac{1}{2} & & 1 \end{pmatrix} \begin{pmatrix} 1 & & & \\ 1 & 1 & & \\ 1 & & 1 & \\ 1 & & & 1 \end{pmatrix}$$

Zu speichern sind 16 Werte (+ Positionen).

b) Pivotisieren auf der Hauptdiagonalen von “rechts unten” nach “links oben” ergibt:

$$B^{-1} = \begin{pmatrix} \frac{1}{4} & & & \\ \frac{1}{4} & 1 & & \\ \frac{1}{4} & & 1 & \\ \frac{1}{4} & & & 1 \end{pmatrix} \begin{pmatrix} 1 & -1 & & \\ & 1 & & \\ & 0 & 1 & \\ & 0 & & 1 \end{pmatrix} \begin{pmatrix} 1 & & -1 & \\ & 1 & 0 & \\ & & 1 & \\ & & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & & & -1 \\ & 1 & & 0 \\ & & 1 & 0 \\ & & & 1 \end{pmatrix}$$

Zu speichern sind 10 Werte (+ Positionen)

Folgende Gegebenheiten sollten bei der Pivotauswahl berücksichtigt werden:

(10.32) Es ist günstig, in einer Spalte mit wenigen NNE zu pivotisieren, da dann die Etavektoren dünn besetzt sind.

(10.33) Es ist günstig, in einer Zeile mit wenigen NNE zu pivotisieren, da dann in anderen Zeilen der umgeformten Restmatrix potentiell weniger neue NNE entstehen.

(10.34) Es ist aus Gründen der Stabilität nicht günstig, ein Pivotelement zu wählen, dessen Betrag sehr klein ist.

In den Inversionsroutinen, die bei den ersten Implementationen des Simplexverfahrens benutzt wurden, beachtete man nur die numerische Stabilität, nahm sich die Spalten der Basismatrix in der Reihenfolge, in der sie abgespeichert waren, multiplizierte sie mit den bereits bestimmten Elementarmatrizen und pivotisierte auf der Komponente mit dem größten Absolutbetrag. Später nahm man eine Vorsortierung der Basisspalten in der Weise vor, so dass die dünn-besetzten zuerst bearbeitet wurden: dies war ohne großen Aufwand möglich, da aufgrund der spaltenweisen Speicherung der NNE die “column counts” (Anzahl NNE einer bestimmten Spalte) bekannt waren.

Heutzutage wird die Inversion üblicherweise in 2 Phasen zerlegt:

(10.35) Boolesche Phase. Entscheidung über Pivotposition!

(10.36) Numerische Phase. Rechentechnisch günstige Ausführung des Pivotschrittes!

Die folgende Beobachtung über Dreiecksmatrizen ist für Implementationen nützlich.

(10.37) Bemerkung. Sei B eine untere (m, m)-Dreiecksmatrix mit $b_{ii} \neq 0$ $i = 1, \dots, m$. Dann ist durch

$$B^{-1} = E_m \cdot E_{m-1} \cdot \ldots \cdot E_2 \cdot E_1$$

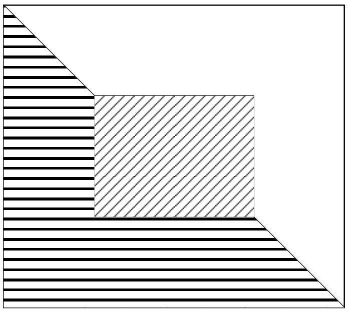
und

$$E_j = \begin{pmatrix} 1 & & & 0 & & & \\ & \ddots & & & & & \\ & & 1 & & & & \\ & & & \frac{1}{b_{jj}} & & & \\ & & & -d_{j+1,j} & 1 & & \\ & & & \vdots & & \ddots & \\ & & & -d_{mj} & & & 1 \end{pmatrix}, \quad d_{ij} = \frac{b_{ij}}{b_{jj}},\ i > j$$

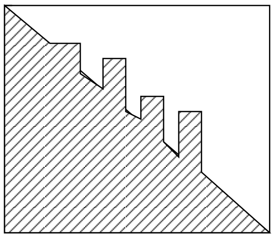
eine Darstellung von B^{-1} in Produktform gegeben.

□

Einige Inversionsverfahren versuchen, diese schöne Eigenschaft von L-Matrizen (lower triangular) auch bei der Inversion anderer dünn besetzter Matrizen auszunutzen. Die gegebenen Basismatrizen werden dabei durch implizites Vertauschen von Zeilen und Spalten so umgeformt, dass sie L-Matrizen möglichst ähnlich werden und z. B. folgendes Aussehen haben:



"Bump"



Bump strukturiert
in L–Matrix mit "Spikes"

Abb. 10.3

Der Bereich, der nicht in Dreiecksform gebracht werden kann, wird **Bump** genannt.

Diese Methode wird z. B. verwendet in der Preassigned Pivot Procedure (Hellerman und Rarick) und ist implementiert in OPTIMA (CDC) und MPS/III (IBM).

Wir betrachten nun

(10.38) Eliminationsform der Inversen (EFI)

Die Grundlage dieser Methode bildet die folgende Beobachtung:

(10.39) Satz (LU-Zerlegung). *Ist $\overline{B}$ eine reguläre (m,m)-Matrix, so gibt es eine Matrix B, die durch Spaltenvertauschungen aus $\overline{B}$ hervorgeht und sich in der Form $B = LU$ darstellen lässt, wobei L eine untere und U eine obere Dreiecksmatrix ist. Es gilt $B^{-1} = U^{-1}L^{-1}$.*

Prinzipielles Vorgehen: Gauß'sches Eliminationsverfahren.

Die Festlegung der Pivotpositionen und ihrer Reihenfolge erfolgt wie bei der PFI in einer vorgeschalteten Booleschen Phase. Da $U^{-1} = U_2 \cdot U_3 \cdot \ldots U_m$ mit

$$U_i = \begin{pmatrix} 1 & & -u_{1i} & & \\ & \ddots & -u_{i-1,i} & & \\ & & 1 & & \\ & & & \ddots & \\ & & & & 1 \end{pmatrix}$$

gilt, ist der Rechenaufwand bei der Inversion geringer als bei der PFI, was sofort aus der Tatsache folgt, dass die U_i bereits durch die Spalten von U gegeben sind und die L_i sich durch weniger Rechenoperationen ergeben als die entsprechenden Elementarmatrizen bei der PFI.

Die nachfolgende Beschreibung der Inversionsroutine des LP-Codes MPSX/370 der Firma IBM ist dem oben genannten Buch von Bastian (Seite 31 ff) entnommen.

(10.40) Die Inversionsroutine von MPSX/370. Die Routine INVERT von MPSX/370 beruht auf der im vorhergehenden besprochenen LU-Zerlegung der Basismatrix. Sie ist zur Inversion großer dünn-besetzter Matrizen entwickelt worden; es wird daher versucht, durch eine sehr weitgehende Listenverarbeitung den Aufwand proportional zur Anzahl der von Null verschiedenen Elemente (NNE) in der Darstellung der Inversen zu halten.

(a) **Die Boolesche Phase**

In der Booleschen Phase dieser Inversionsroutine werden die Pivotpositionen so vorbestimmt, dass die Anzahl der NNE in der Darstellung der Inversen möglichst gering wird (vgl. Benichou u. a.).

Zunächst wird die Besetzungsstruktur der Ausgangsmatrix in die Form zweier Listen übertragen: die eine Liste enthält spaltenweise die Zeilenindices der NNE, und in der anderen werden entsprechend zeilenweise die Spaltenindices gespeichert. Darüber hinaus werden noch column und row counts (Anzahl der NNE in einer Spalte bzw. Zeile) festgehalten. Auf diese Weise wird ein rascher Zugriff auf Spalten oder Zeilen nach dem Kriterium der Besetzungsdichte ermöglicht.

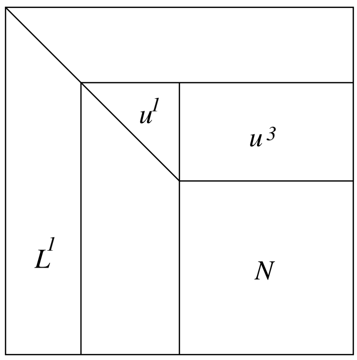
Die Bestimmung der Pivotpositionen erfolgt in drei Schritten, wobei die beiden ersten Schritte der Identifizierung des "Bumps" in den oben besprochenen Triangularisierungsverfahren entsprechen:

1. Wähle Spalten mit row count 1; dadurch sind die zugehörigen Pivotzeilen ebenfalls eindeutig festgelegt.

   Die Spalten- und Zeilenindices der NNE dieser Spalten werden aus den beiden Listen entfernt und die row und column counts angepasst. Dadurch entstehen möglicherweise wieder Zeilen mit row count 1, die anschließend ausgewählt werden.

2. Entsprechend wie im Schritt 1 werden dann Spalten mit column count 1 ausgewählt, Pivotspalten und -zeilen aus den Indices-Listen gestrichen und row sowie column counts angepasst.

   Das Ergebnis der beiden ersten Schritte ist die Umordnung von Spalten der Ausgangsmatrix so, dass die vorbestimmten Pivotelemente auf der Hauptdiagonale liegen. Durch symmetrisches Vertauschen der Zeilen und Spalten gemäß der vorgesehenen Pivotfolge erhält man eine Matrix B der folgenden Gestalt:

   $B =$ 

   Der Nukleus N entspricht dem Bump beim Verfahren von Hellerman und Rarick und in der Tat könnte dies Verfahren auch zur Zerlegung des Nukleus benutzt werden.

3. Stattdessen beruht der Schritt 3 der Reinversionsroutine von MPSX/370 auf einer LU-Zerlegung des Nukleus in der Booleschen Matrix, wobei die folgenden einfachen Auswahlregeln angewandt werden:
   - wähle unter den Spalten mit minimalem column count eine solche als Pivotspalte, die ein NNE in einer Zeile mit möglichst kleinem row count besitzt;
   - wähle als Pivotzeile eine Zeile mit minimalem row count unter den Zeilen mit NNE in der Pivotspalte.

   Ist eine Pivotposition bestimmt worden, so werden die beiden Indices-Listen aktualisiert; die Anpassung der row und column counts erfolgt entsprechend.

   Durch besondere Maßnahmen wird versucht, das Durchsuchen von Listen sowie die Suche nach NNE in Vektoren einzuschränken (eine detaillierte Beschreibung der eingesetzten Methoden findet man bei Gustavson). So wird die Pivotwahl z. B. dadurch vereinfacht, dass zu jeder Spalte j, in der noch nicht pivotisiert wurde, nicht nur der column count sondern auch der minimale row count unter den Zeilen mit NNE in Spalte j mitgeführt wird. Spalten mit gleichem column count und minimalem row count werden verkettet. Zusätzlich werden die Listen der Zeilen- und Spaltenindices von NNE zu der vorbestimmten Pivotposition für die numerische Phase aufgehoben:
   - die Liste der Spaltenindices in der Pivotzeile gibt die Spalten an, die aktualisiert werden müssen (durch Umspeicherung erhält man zu jeder Spalte die Adressen derjenigen Etavektoren, die zur Aktualisierung dieser Spalte herangezogen werden müssen);
   - die Liste der Zeilenindices in der Pivotspalte entspricht der Besetzung mit NNE in der aktualisierten Pivotspalte, und ihre Speicherung verhindert, dass die gesamte Spalte nach NNE durchsucht werden muss.

   Werden die Listen im Verlaufe zu umfangreich und damit auch die Aktualisierung zu aufwendig, so wird die Boolesche Matrix explizit als Matrix umgespeichert (ein Byte pro Element). Ist diese Matrix vollbesetzt oder erreicht sie eine vom Benutzer zu spezifizierende Dichte, so wird die Boolesche Phase abgebrochen.

(b) **Die Numerische Phase**

In der Numerischen Phase werden die Pivotspalten in der Reihenfolge, die in der Booleschen Phase vorbestimmt wurde, aktualisiert und die Etavektoren gebildet. Die Etavektoren von L^{-1} und U^{-1} werden dabei auf getrennten Files, dem L-File und dem U-File, abgespeichert. Die Etavektoren zu L^1 und zu U^1 ergeben sich direkt aus den Ausgangsdaten und werden sofort extern gespeichert. Bei der anschließenden Zerlegung der Nukleusspalten hält man die neu entstehenden Etaspalten des L-Files zunächst im Hauptspeicher, sofern genug Speicherplatz vorhanden ist.

Sei d diejenige Basisspalte, aus der die Elementarmatrizen L_k und U_k berechnet werden sollen. Ausgangspunkt ist die Spalte

$$\hat{d} := L_{k-1} \dots L_2 \cdot L_1 \cdot d,$$

die man sich in einem Arbeitsbereich erzeugt (man beachte, dass die Etavektoren von $L_1, \dots, L_j$ nicht benötigt werden, wenn L^1 aus j Spalten besteht).

Diejenigen Komponenten von $\hat{d}$, die in Zeilen liegen, in denen bereits pivotisiert wurde, liefern nun den Etavektor von U_k; aus den anderen ergibt sich der Etavektor von L_k.

Wegen der in der Booleschen Phase geleisteten Vorarbeiten sind hierbei nur sehr wenige Suchvorgänge erforderlich:

- die Pivotzeile ist bereits bekannt;
- diejenigen vorausgegangenen Pivots, die für die Berechnung von $\hat{d}$ relevant sind, sind bekannt; das L-File muss also nicht durchsucht werden, sondern auf die benötigten L_j kann direkt zugegriffen werden;
- die Indices der NNE in der aktualisierten Spalte sind von vornherein bekannt; dies erleichtert nicht nur die Abspeicherung der Etavektoren sondern auch das Löschen des Arbeitsbereichs für die nachfolgenden Operationen.

Ist ein vorbestimmtes Pivotelement dem Betrag nach zu klein bzw. durch Differenzbildung tatsächlich gleich Null, so wird die betreffende Spalte zunächst zurückgestellt bis alle anderen vorbestimmten Pivots durchgeführt worden sind. Die Verarbeitung dieser Spalten ist wesentlich aufwendiger, da keine Informationen aus der Booleschen Phase verwendet werden können. Die Pivot-Wahl in den zurückgestellten Spalten sowie in denjenigen Spalten, für die in der Booleschen Phase kein Pivotelement bestimmt wurde, erfolgt nach dem Gesichtspunkt der numerischen Stabilität: man entscheidet sich für die Komponente mit dem größten Absolutbetrag in einer Zeile, in der noch nicht pivotisiert worden ist (Pivoting for Size).

10.6 Spezialliteratur zum Simplexalgorithmus

Die folgende Liste ist eine (unvollständige) Sammlung einiger wichtiger Paper zu dem in den Kapiteln 9 und 10 behandelten Themen.

D. Avis & V. Chvátal [1978]: *Notes on Bland's Pivoting Rule,* Mathematical Programming Studies 8 (1978) 24–34.

R. H. Bartels [1971]: *A Stabilization of the Simplex Method,* Numerische Mathematik 16 (1971) 414–434.

R. H. Bartels & G. H. Golub [1969]: *The Simplex Method of Linear Programming Using LU Decomposition,* Communications of the Association for Computing Machinery 12 (1969) 266–268. 275–278.

E. M. L. Beale & J. A. Tomlin [1970 *Special Facilities in a General Mathematical Programming System for Non-Convex Problems Using Sets of Variables,* in: J. Lawrence (ed.), "Proceedings of the fifth IFORS Conference". Tavistock, London 1970, 447–454.

M. Bénichou, J. M. Gauthier, P. Girodet & G. Hentgès. G. Ribière & O. Vincent [1971]: *Experiments in Mixed Integer Linear Programming,* Mathematical Programming 1 (1971) 76–94.

M. Bénichou, J. M. Gauthier, G. Hentgès & G. Ribière [1977]: *The Efficient Solution of Large-Scale Linear Programming Problems — Some Algorithmic Techniques and Computational Results,* Mathematical Programming 13 (1977) 280–322.

R. G. Bland [1977]: *New Finite Pivoting Rules for the Simplex Method,* Mathematics of Operations Research 2 (1977) 103–107.

R. K. Brayton, F. G. Gustavson & R. A. Willoughby [1970]: *Some Results on Sparse Matrices,* Mathematics of Computation 24 (1970) 937–954.

H. Crowder & J. M. Hattingh [1975]: *Partially Normalized Pivot Selection in Linear Programming,* Mathematical Programming Study 4 (1975) 12–25.

A. R. Curtis & J. K. Reid [1972]: *On the automatic Scaling of Matrices for Gaussian Elimination,* Journal of the Institute of Mathematics and its Applications 10 (1972) 118–124.

G. B. Dantzig, R. P. Harvey, R. D. McKnight & S. S. Smith [1969]: *Sparse Matrix Techniques in Two Mathematical Programming Codes,* in: R. A. Willoughby (ed.), "Sparse Matrix Proceedings" RA-1, IBM Research Center, Yorktown Heights, New York, 1969, 85–99.

R. Fletcher & S. P. J. Matthews [1984]: *Stable Modification of Explicit LU Factors for Simplex Updates*, Mathematical Programming 30 (1984) 267–284.

J. J. H. Forrest & J. A. Tomlin [1972]: *Updated Triangular Factors of the Basis to Maintain Sparsity in the Product Form Simplex Method*, Mathematical Programming 2 (1972) 263–278.

A. M. Geoffrion & R. E. Marsten [1972]: *Integer Programming Algorithms: a Framework and State-of-the-Art Survey*, Management Science 18 (1972) 465–491.

D. Goldfarb & J. K. Reid [1977]: *A Practicable Steepest Edge Simplex Algorithm* Mathematical Programming 12 (1977) 361–371.

D. Goldfarb [1976]: *Using the Steepest Edge Simplex Algorithm to Solve Sparse Linear Programs*, in: J. Bunch and D. Rose (eds.), "Sparse Matrix Computations", Academic Press, New York, 1976, 227–240.

F. G. Gustavson [1972]: *Some Basic Techniques for Solving Sparse Systems of Linear Equations*, in: D. J. Rose & R. A. Willoughby (eds.), "Sparse Matrices and Their Applications", Plenum Press New York, 1972, 41–52.

P. M. J. Harris [1975]: *Pivot Selection Methods of the Devex LP Code*, Mathematical Programming Study 4 (1975) 30–57.

R. G. Jeroslow [1973]: *The Simplex Algorithm with the Pivot Rule of Maximizing Improvement Criterion*, Discrete Mathematics 4 (1973) 367–377.

V. Klee & G. J. Minty [1972]: *How Good is the Simplex Algorithm?*, in: O. Shisha (ed.), "Inequalities - III", Academic Press, New York, 1972, 159–175.

H. Kuhn & R. E. Quandt [1963]: *An Experimental Study of the Simplex Method*, in: N. C. Metropolis et al. (eds.), "Experimental Arithmetic, High-Speed Computing and Mathematics", Proceedings of Symposia on Applied Mathematics XV, American Mathematical Society, Providence, RI 1963, 107–124.

M. A. Saunders [1976]: *The Complexity of LU Updating in the Simplex Method*, in: R. S. Anderssen & R. P. Brent (eds.), "The Complexity of Computational Problem Solving", Queensland University Press, Queensland, 1976, 214–230.

M. A. Saunders [1976]: *A Fast, Stable Implementation of the Simplex Method Using Bartels-Golub Updating*, in: J. R. Bunch & D. J. Rose (eds.), "Sparse Matrix Computations", Academic Press New York, 1976, 213–226.

M. J. Todd [1984]: *Modifying the Forrest-Tomlin and Saunders Updates for Linear Programming Problems with Variable Upper Bounds*, Cornell University, School of OR/IE 1984, TR 617.

J. A. Tomlin 1975]: *On Scaling Linear Programming Problems*, Mathematical Programming Study (1975) 146–166.

Kapitel 11

Postoptimierung und parametrische Programme

Wir haben bisher die Theorie (und Praxis) des Simplexverfahrens entwickelt unter der Annahme, dass die Daten, A, b, und c fest vorgegeben sind. Bei praktischen Problemen können jedoch häufig nicht alle der benötigten Zahlen mit Sicherheit angegeben werden, d. h. es kann Unsicherheit über die Beschränkungen b oder den zu erwartenden Gewinn $c^T x$ geben. Man muss also die Tatsache mit in Betracht ziehen, dass die Wirklichkeit nicht exakt in das lineare Modell abgebildet wurde bzw. werden konnte. Darüber hinaus können im Nachhinein zusätzliche Variablen oder Restriktionen auftreten, die bei Aufstellung des Modells übersehen wurden oder nicht berücksichtigt werden konnten.

Wir werden uns nun überlegen, wie wir gewisse auftretende Änderungen behandeln können, wenn wir z. B. die optimale Lösung des ursprünglichen Programms gefunden haben. Wir wollen uns auf die folgenden Fälle beschränken:

1. Variation der rechten Seite b.
2. Variation der Zielfunktion c.
3. Änderung einer Nichtbasisspalte.
4. Hinzufügung einer neuen Variablen.
5. Hinzufügung einer neuen Restriktion.

Im Prinzip könnten wir natürlich versuchen, eine Theorie zu entwerfen, bei der

Schwankungen aller Daten berücksichtigt werden. Das ist jedoch außerordentlich kompliziert, und es gibt darüber kaum rechnerisch verwertbare Aussagen.

Wir gehen im weiteren davon aus, dass ein LP in Standardform (9.1) gegeben ist und dass wir eine optimale Basis A_B von A kennen.

11.1 Änderung der rechten Seite b

Wir wollen uns zunächst überlegen, welchen Einfluss eine Änderung der rechten Seite auf die Optimallösung bzw. den Wert der Optimallösung hat.

(11.1) Änderung der Optimallösung. Gegeben seien ein LP in Standardform (9.1)

$$\max\{c^T x \mid Ax = b, x \geq 0\}$$

und eine optimale Basis A_B von A. Die neue rechte Seite des LP sei $b' := b + \Delta$. Wir berechnen die neue Basislösung zur Basis A_B. Diese ergibt sich aus:

$$x'_B = A_B^{-1}(b + \Delta) = x_B + A_B^{-1}\Delta,\ x'_N = 0.$$

Gilt $x'_B \geq 0$, so ist die neue Basislösung optimal, da sich ja an den reduzierten Kosten $\overline{c}^T = c_N^T - c_B^T A_B^{-1} A_N$ nichts geändert hat, d. h. es gilt weiterhin $\overline{c} \leq 0$.

Gilt $x'_B \ngeq 0$, so ist die neue Basislösung primal nicht zulässig. Die Optimalitätsbedingung $\overline{c} \leq 0$ ist jedoch weiterhin erfüllt. Das aber heißt nach (10.18), dass die Basis dual zulässig ist. Folglich haben wir mit A_B eine zulässige Basis für die duale Simplexmethode (10.27), und wir können direkt mit Phase II von (10.27) mit der Neuberechnung der Optimallösung beginnen.

□

In diesem Zusammenhang sei auf eine wichtige Funktion, die die Änderung des Zielfunktionswertes eines LP bei Variationen der rechten Seite angibt, hingewiesen. Diese Funktion hat interessante Eigenschaften, von denen wir nachfolgend einige aufführen wollen. Die Funktion

$$L : \{b \in \mathbb{K}^m \mid P^=(A, b) \neq \emptyset\} \longrightarrow \mathbb{K} \cup \{\infty\}$$

definiert durch

$$L(b) := \sup\{c^T x \mid Ax = b, x \geq 0\} \tag{11.2}$$

heißt **Perturbationsfunktion** bzgl. des LPs

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

Es bezeichne

$$\mathrm{RP}(A) := \{b \in \mathbb{K}^m \mid P^=(A,b) \neq \emptyset\}$$

den Definitionsbereich von L (wegen $0 \in \mathrm{RP}(A)$ ist $\mathrm{RP}(A) \neq \emptyset$) und

$$\text{epi}(L) := \left\{ \begin{pmatrix} v \\ z \end{pmatrix} \in \mathbb{K}^{m+1} \mid v \in \mathrm{RP}(A),\ z \in \mathbb{K},\ L(v) \leq z \right\}$$

den Epigraphen von L.

(11.3) Satz. *Die Menge* epi(L) *ist ein polyedrischer Kegel.*

Beweis : Offenbar ist epi(L) eine Projektion des polyedrischen Kegels $\{(x^T, v^T, z)^T \mid Ax - v = 0, x \geq 0,\ c^T x - z \leq 0\}$ auf die (v, z)-Koordinaten. Also ist nach (6.1) epi(L) ein Polyeder, das trivialerweise ein Kegel ist. □

(11.4) Bemerkung.

$$\exists\, b \in \mathrm{RP}(A) \textit{ mit } L(b) < \infty \iff \forall\, b \in \mathrm{RP}(A) \textit{ gilt } L(b) < \infty.$$

Beweis : Sei $V := \{x \in \mathbb{K}^n \mid Ax = 0\}$, dann gibt es zu jedem $b \in \mathbb{K}^m$ einen Vektor $x(b) \in \mathbb{K}^n$ mit $P^=(A,b) = (V + x(b)) \cap \mathbb{K}^n_+$. Folglich gilt für alle $b \in \mathrm{RP}(A) : L(b) = \infty \iff L(0) = \infty$. □

(11.5) Satz. *Ist $L \not\equiv \infty$, so gibt es Vektoren $g^i \in \mathbb{K}^m$ $(i = 1, \ldots, k)$, so daß*

$$L(v) = \max\{v^T g^i \mid i = 1, \ldots, k\}$$

für alle $v \in \mathrm{RP}(A)$ gilt.

Beweis : Mit Satz (11.3) ist $K :=$ epi(L) ein polyedrischer Kegel. Also gibt es nach Bemerkung (2.6) eine $(r, m+1)$-Matrix B, so dass $K = P(B, 0)$ gilt. Da $L(0) \geq 0$, ist $\binom{0}{z} \notin K$ für alle $z < 0$. Also kann die letzte Spalte von B kein Nullvektor sein, und wir können o. B. d. A. annehmen, dass B die Form

$$B = \begin{pmatrix} H & 0 \\ G & -h \end{pmatrix}$$

mit $h \neq 0$ besitzt. Da $L \not\equiv \infty$, folgt mit (11.4) $L(0) < \infty$, und daraus folgt direkt $L(0) = 0$. Also wissen wir, dass $\binom{0}{1} \in K$. Dies impliziert $h > 0$. Sei o. B. d. A. $h = \mathbb{1}$. Dann gilt: $\text{epi}(L) = \{(v^T, z)^T \mid Hv \leq 0,\ Gv \leq z\}$, und es ist $L(v) = z \iff G_{i.}v = z$ für ein i. Bezeichnen g^i $(i = 1, \dots, k)$ die Zeilen von G, so gilt

$$L(v) = \max\{v^T g^i \mid i = 1, \dots, k\}.$$

□

(11.6) Folgerung. *Die Perturbationsfunktion L ist stückweise linear und konkav.*

□

Speziell folgt daraus, dass sich der Wert der Zielfunktion eines linearen Programms stetig mit einer Variation der rechten Seite des LP ändert. Die stetige Änderung lässt sich durch (11.5) explizit angeben. Sie ist "fast überall" linear, bis auf einige "Knicke".

11.2 Änderungen der Zielfunktion c

Wir gehen wieder davon aus, dass wir eine Optimalbasis A_B von (9.1) kennen und dass sich die Zielfunktion c um Δ ändert. Da wir eine Basis A_B gegeben haben, können wir die neuen Kosten der Basislösung ausrechnen. Es gilt

$$\begin{aligned}(c^T + \Delta^T)x &= (c_B^T + \Delta_B^T)A_B^{-1}b + (c_N^T + \Delta_N^T - (c_B^T + \Delta_B^T)A_B^{-1}A_N)x_N \\ &= c_B^T\overline{b} + \overline{c}x_N + \Delta_B^T\overline{b} + \underbrace{(\Delta_N^T - \Delta_B^T\overline{A})}_{=:\overline{\Delta}}x_N.\end{aligned}$$

(a) Wird keiner der Koeffizienten von c_B^T geändert (ist also $\Delta_B = 0$), ändert sich wegen $x_N = 0$ der Zielfunktionswert der Basislösung zur Basis A_B nicht.

(b) Sind die neuen reduzierten Kosten $\overline{c} + \overline{\Delta}$ nicht positiv, so ist das Optimalitätskriterium (9.13) (b) weiterhin erfüllt, d. h. A_B bleibt die optimale Basis, jedoch ändert sich u. U. der Wert der zugehörigen Basislösung wenn $\Delta_B \neq 0$.

(c) Ist einer der Koeffizienten $\overline{c}_j + \overline{\Delta}_j$ positiv, so wenden wir Phase II des primalen Simplexalgorithmus mit (zulässiger) Anfangsbasislösung A_B auf das neue Problem an.

(Zur Berechnung von $\overline{\Delta}$ genügt die Kenntnis von $\overline{A}$ oder A_B^{-1} und A_N.)

An den obigen Berechnungen kann man sehen, wie man eine Schar von linearen Programmen, bei denen entweder nur b oder nur c einer Änderung unterworfen wird, lösen kann.

11.3 Änderungen des Spaltenvektors $A_{.j}$, $j = N(s)$

Wollen wir die s-te Nichtbasisspalte (d. h. die j-te Spalte von A) um Δ ändern und kennen wir die Basisinverse A_B^{-1}, so können wir die neue s-te Spalte von $\overline{A}$ berechnen durch

$$\overline{A}'_{.s} = A_B^{-1}(A_{.j} + \Delta) = \overline{A}_{.s} + A_B^{-1}\Delta.$$

Die Basislösung $\overline{x}$ zu A_B bleibt weiterhin primal zulässig, da diese Änderung keinen Einfluss auf $\overline{x}_B = \overline{b}$, $\overline{x}_N = 0$ hat. Jedoch bleibt $\overline{x}$ i. a. nicht optimal, da sich der s-te Koeffizient der reduzierten Kosten ($j = N(s)$) wie folgt ändert

$$\begin{aligned} \overline{c}'_s &= c_j - c_B^T(\overline{A}_{.s} + A_B^{-1}\Delta) \\ &= c_j - c_B^T\overline{A}_{.s} - c_B^T A_B^{-1}\Delta \\ &= \overline{c}_s - c_B^T A_B^{-1}\Delta = \overline{c}_s - u^T\Delta, \end{aligned}$$

wobei u eine optimale duale Lösung ist.

Die Optimalität der gegenwärtigen Basislösung $\overline{x}$ bleibt genau dann erhalten, wenn $\overline{c}_s \leq u^T\Delta$ gilt. Andernfalls können wir die neue Optimallösung mit Phase II des primalen Simplexalgorithmus berechnen, wobei wir von der zulässigen Basis A_B ausgehen (die revidierte Methode ist natürlich von Vorteil).

(Bei Änderung einer Basisspalte kann man kaum Vorhersagen machen. Es muss neu invertiert werden. Dabei kann sowohl primale als auch duale Zulässigkeit verloren gehen.)

(11.7) Beispiel. Wir betrachten nochmals unser Beispiel (10.30)

$$\begin{array}{lrr} \max\ -x_1 - 2x_2 & & \\ -x_1 - \ \ x_2 & \leq & -2 \\ -2x_1 - \ \ x_2 & \leq & -3 \\ x_1, x_2 & \geq & 0 \end{array}$$

Das optimale (Tucker) Tableau ist:

-1	-1	-2
-2	1	1
-1	1	2

duale Lösung $y_3 = 1$, $y_4 = 0$ primale Lösung $x_1 = 2$, $x_2 = 0$

Die Spalte $A_{.2}$ ist eine Nichtbasisspalte. Wir untersuchen, um wieviel diese Spalte geändert werden kann, ohne Optimalität zu verlieren.

$$A_{.2} = \begin{pmatrix} -1 \\ -1 \end{pmatrix}, \quad \Delta = \begin{pmatrix} \alpha \\ \beta \end{pmatrix}, \quad \overline{c}_2 = -1, \; u = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$$

also $\overline{c}_2' = -1 - (1,0)\binom{\alpha}{\beta} = -1 - \alpha$

$$\overline{c}_2' \le 0 \iff -1 - \alpha \le 0 \iff \alpha \ge -1.$$

Die obige Basislösung ist also für alle linearen Programme der Form

$$\begin{array}{ll} \max\ -x_1 - 2x_2 & \\ -x_1 + (\alpha - 1)x_2 & \le -2 \\ -2x_1 + (\beta - 1)x_2 & \le -3 \\ x_1, x_2 & \ge \ \ 0 \end{array}$$

optimal, wenn nur $\alpha \ge -1$ gilt. Zur geometrischen Interpretation siehe Abbildung 11.1. □

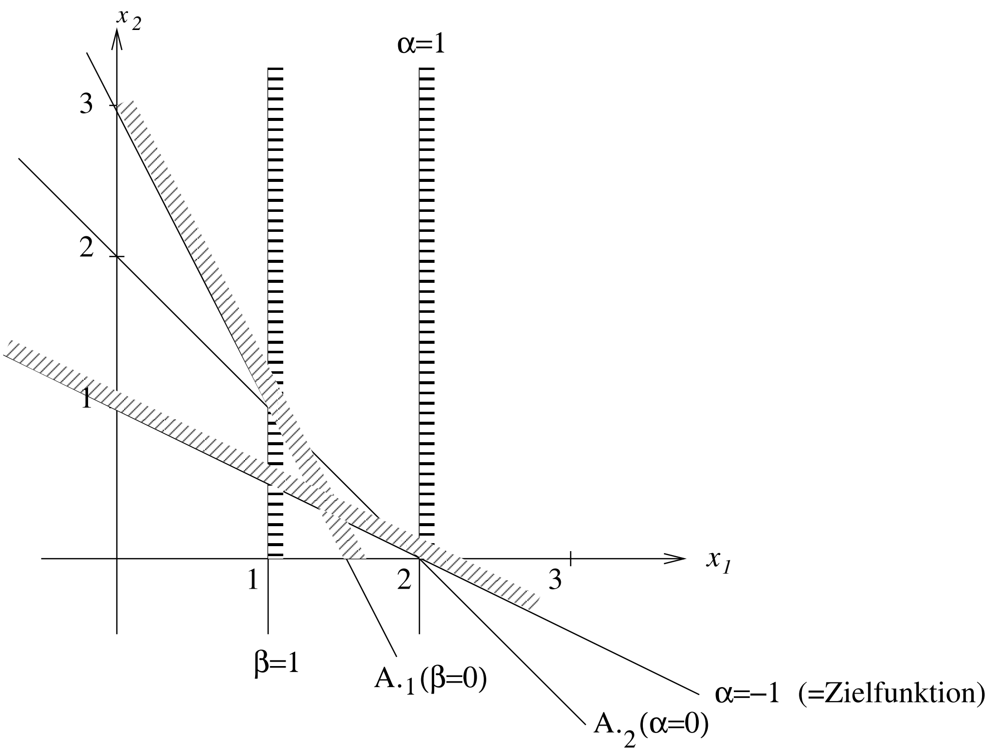


Abb. 11.1

11.4 Hinzufügung einer neuen Variablen

Sei $A_{\cdot n+1}$ die neue Spalte von A und c_{n+1} der Zielfunktionswert der neuen Variablen x_{n+1}. Wir verlängern den Vektor N der Nichtbasisindizes um eine Komponente und setzen $N(n-m+1) := n+1$. Wir berechnen

$$\overline{c}_{n-m+1} := c_{n+1} - \underbrace{c_B^T A_B^{-1}}_{u^T} A_{\cdot n+1}.$$

Gilt $\overline{c}_{n-m+1} \leq 0$, so bleibt aufgrund von (9.13) (b) die alte Lösung optimal. Andernfalls führen wir mit der Anfangsbasis A_B den primalen Simplexalgorithmus aus. Dabei ist

$$\overline{A}_{\cdot n-m+1} = A_B^{-1} A_{\cdot n+1}.$$

Im ersten Schritt wird die neue Variable in die Basis aufgenommen.

11.5 Hinzufügung einer neuen Restriktion

Zu den bestehenden Nebenbedingungen fügen wir die neue Restriktion

$$A_{m+1,.}x = \sum_{i=1}^{n} a_{m+1,i}x_i = b_{m+1}$$

hinzu.

1. Fall: Die Basislösung $\overline{x}$ zu A_B erfüllt $A_{m+1,.}\overline{x} = b_{m+1}$.

In diesem Fall ist $\overline{x}$ auch die optimale Lösung des erweiterten Problems. Ist die neue Zeile linear abhängig von den übrigen Zeilen von A, so ist die neu hinzugefügte Gleichung irrellevant für das Problem und kann gestrichen werden. Ist die neue Zeile linear unabhängig von den übrigen, so ist $\overline{x}$ eine entartete Basislösung. Eine der Nichtbasisvariablen wird mit Wert 0 in die Basis aufgenommen und die Basis entsprechend erweitert.

2. Fall: Die Basislösung $\overline{x}$ erfüllt die neue Gleichung nicht.

Wir führen eine neue Schlupfvariable $x_{n+1} \geq 0$ ein und verändern die neue Gleichung in

$$\sum_{i=1}^{n} a_{m+1,i}x_i \pm x_{n+1} = b_{m+1},$$

wobei wir ein $+$ Zeichen schreiben, falls $A_{m+1,.}\overline{x} > b_{m+1}$ gilt, andernfalls ziehen wir x_{n+1} ab. Wir verlängern B um eine Komponente und setzen $B(m+1) := n+1$, d. h. wir nehmen x_{n+1} mit dem Wert $\overline{b}_{m+1} = -|b_{m+1} - A_{m+1,.}\overline{x}|$ in die Basis auf. Da $\overline{b}_{m+1}$ negativ ist, ist die neue Basis primal nicht zulässig; wir haben jedoch keine Änderung an der Zielfunktion vorgenommen, d. h. die reduzierten Kosten sind weiterhin nicht positiv. Also ist die gegenwärtige neue Basis dual zulässig. Die neue Basis hat die Form

$$\begin{pmatrix} A_B & 0 \\ A_{m+1,B} & 1 \end{pmatrix}.$$

Die Basisinverse lässt sich wie folgt schreiben:

$$\begin{pmatrix} A_B^{-1} & 0 \\ -A_{m+1,B}A_B^{-1} & 1 \end{pmatrix}.$$

Die neue $(m+1)$-te Zeile von $\overline{A}$ lautet:

$$-A_{m+1,B}\overline{A} + A_{m+1,N} = \overline{A}_{m+1,.}\,.$$

Wir führen nun einen dualen Pivotschritt auf der $(m+1)$-ten Zeile von $\overline{A}$ durch, wobei wir versuchen, die Variable x_{n+1} aus der Basis zu entfernen.

Ergibt der duale Beschränktheitstest ($\overline{A}_{m+1,.} \geq 0$ (10.27) (II.3)), dass das duale Problem unbeschränkt ist, so ist das neue primale Problem unzulässig, d. h. die Hyperebene $A_{m+1,.}x = b_{m+1}$ hat einen leeren Schnitt mit der Menge der zulässigen Lösungen des alten Problems, und wir können den Algorithmus beenden.

Andernfalls führen wir Phase II des dualen Simplexalgorithmus (10.27) aus. Endet der Algorithmus mit einer primal und dual zulässigen Lösung, so gibt es eine optimale primale Lösung x^* mit $x^*_{n+1} = 0$. Diese kann wie folgt konstruiert werden, falls x^*_{n+1} nicht bereits Null ist:

Sei x' die primale zulässige Basislösung nach Beendigung des dualen Verfahrens und $\overline{x}$ die Anfangsbasislösung bei Beginn des dualen Programms, d. h. $\overline{x}_{n+1} = \overline{b}_{m+1} < 0$. Setzen wir

$$\lambda := \frac{x'_{n+1}}{x'_{n+1} - \overline{x}_{n+1}},$$

so gilt $\lambda > 0$, da $x'_{n+1} > 0$ und $\overline{x}_{n+1} < 0$. Sei $x^* := \lambda\overline{x} + (1-\lambda)x'$, dann gilt

$$x^*_i \geq 0 \ \text{ für } \ i = 1, \ldots, n, \ \text{ da } \ \overline{x}_i \geq 0, \ x' \geq 0$$

$$\begin{aligned} x^*_{n+1} &= \tfrac{x'_{n+1}}{x'_{n+1} - \overline{x}_{n+1}}\overline{x}_{n+1} + x'_{n+1} - \tfrac{x'_{n+1}}{x'_{n+1} - \overline{x}_{n+1}}x'_{n+1} \\ &= \tfrac{1}{x'_{n+1} - \overline{x}_{n+1}}(x'_{n+1}\overline{x}_{n+1} - x'_{n+1}x'_{n+1} + x'_{n+1}x'_{n+1} - \overline{x}_{n+1}x'_{n+1}) \\ &= 0. \end{aligned}$$

Also gilt $x^* \geq 0$ und ferner

$$\begin{aligned} c^T x^* &= \lambda \underbrace{c^T\overline{x}}_{\geq c^T x'} + (1-\lambda)c^T x' \\ &\geq c^T x'. \end{aligned}$$

Daraus folgen die Zulässigkeit und die Optimalität von x^*.

Kapitel 12

Die Ellipsoidmethode

In (10.10) haben wir angemerkt, dass man Beispiele von Polyedern und Zielfunktionen konstruieren kann, so dass der Simplexalgorithmus alle Ecken der Polyeder durchläuft. Die in den Übungen besprochene Klasse von Klee & Minty-Beispielen, bei denen dieser Fall auftritt, ist definiert durch n Variable und $2n$ Ungleichungen. Die zugehörigen Polyeder haben 2^n Ecken. Es ist offensichtlich, dass 2^n Iterationen des Simplexalgorithmus zu nicht tolerierbaren Rechenzeiten führen. Man kann zwar zeigen (Borgwardt "The Average Number of Pivot Steps Required by the Simplex-Method is Polynomial", Zeitschrift für Operations Research 26 (1982) 157–177), dass das Laufzeitverhalten des Simplexalgorithmus im Durchschnitt gut ist, aber dennoch bleibt die Frage, ob es nicht möglich ist, eine generelle "niedrige" Schranke für die Anzahl der Iterationensschritte des Simplexalgorithmus (mit einer speziellen Zeilen- und Spaltenauswahlregel) zu finden. Dieses Problem ist bis heute ungelöst!

Im Jahre 1979 hat L. G. Khachiyan ("A polynomial algorithm in linear programming", Doklady Akademia Nauk SSSR 244 (1979) 1093–1096, engl. Übersetzung in Soviet Mathematics Doklady 20 (1979) 191–194) gezeigt, dass lineare Programme in "polynomialer Zeit" gelöst werden können. Das Verfahren, dass er dazu angegeben hat, die sogenannte Ellipsoidmethode, arbeitet allerdings völlig anders als der Simplexalgorithmus und scheint in der Praxis im Laufzeitverhalten dem Simplexalgorithmus "im Durchschnitt" weit unterlegen zu sein. Theoretisch beweisbar ist jedoch, dass es keine Beispiele linearer Programme (z. B. vom Klee & Minty-Typ) gibt, die die Ellipsoidmethode zu exorbitanten Laufzeiten zwingen.

Die dieser Methode zugrundeliegenden Ideen benutzen ganz andere geometrische Überlegungen, als wir sie bisher gemacht haben. Außerdem sind zu einer korrekten Implementation so viele numerische Details zu klären, dass der zeitliche

Rahmen dieser Vorlesung durch eine vollständige Diskussion aller Probleme gesprengt würde. Wir wollen daher nur einen Überblick über die Methode geben und nur einige Beweise ausführen.

12.1 Polynomiale Algorithmen

Um das Laufzeitverhalten eines Algorithmus exakt beschreiben zu können, ist es notwendig Maschinenmodelle, Kodierungsvorschriften, Algorithmusbeschreibungen etc. exakt einzuführen. Dies wird z. B. in der Vorlesung "Komplexitätstheorie" gemacht. Hier wollen wir die wichtigsten Konzepte dieser Theorie informell für den Spezialfall der linearen Programmierung einführen. Eine mathematisch exakte Darstellung der Theorie findet man z. B. in dem Buch

M. R. Garey & D. S. Johnson, "Computers and Intractability: A Guide to the Theory of $\mathcal{NP}$-Completeness", Freeman, San Francisco, 1979, (80 ST 230 G 229).

Zunächst müssen wir unser Konzept, dass Zahlen aus dem Körper $\mathbb{K}$ gewählt werden können, aufgeben. Jede Zahl, die wir in unserem Berechnungsmodell betrachten, muss endlich kodierbar sein. Es gibt aber kein Kodierungsschema, so dass z. B. alle reellen Zahlen durch endlich viele Symbole beschrieben werden könnten. Wir beschränken uns daher auf rationale Zahlen. Haben wir eine Ungleichung

$$a^T x \leq \alpha$$

mit $a \in \mathbb{Q}^n$, $\alpha \in \mathbb{Q}$, so können wir auf einfache Weise das kleinste gemeinsame Vielfache p der Nenner der Komponenten von a und des Nenners von α bestimmen. Die Ungleichung

$$pa^T x \leq p\alpha$$

hat offenbar die gleiche Lösungsmenge wie $a^T x \leq \alpha$, während alle Koeffizienten dieser Ungleichung ganzzahlig sind. Der Fall rationaler Daten lässt sich also direkt auf den Fall ganzzahliger Daten reduzieren. Es ist zwar häufig einfacher unter der Voraussetzung ganzzahliger Daten zu rechnen, und fast alle Aufsätze zur Ellipsoidmethode machen diese Annahme, wir wollen aber dennoch für dieses Kapitel voraussetzen:

(12.1) Annahme. *Alle Daten linearer Programme sind rational, d. h. für jedes LP der Form* $\max c^T x$, $Ax \leq b$ *gilt* $c \in \mathbb{Q}^n$, $A \in \mathbb{Q}^{(m,n)}$, $b \in \mathbb{Q}^m$.

□

Nun müssen wir die Frage klären, wie Daten "gegeben" sind. Da unsere Computer üblicherweise mit Binärcodes arbeiten, wollen wir annehmen, dass alle vorkommenden ganzen Zahlen binär kodiert sind. Die binäre Darstellung einer ganzen Zahl n benötigt $\lceil \log_2(|n|+1) \rceil$ Stellen (Bits) und eine Stelle für das Vorzeichen. Wir nennen daher

$$\langle n \rangle := \lceil \log_2(|n|+1) \rceil + 1 \tag{12.2}$$

die **Kodierungslänge** von n. Jede rationale Zahl r hat eine Darstellung $r = \frac{p}{q}$ mit p und q teilerfremd und $q > 0$. Wir nehmen immer an, dass rationale Zahlen in dieser Form dargestellt sind. Also können wir sagen, dass die Kodierungslänge einer rationalen Zahl $r = \frac{p}{q}$ gegeben ist durch

$$\langle r \rangle = \langle p \rangle + \langle q \rangle.$$

Die **Kodierungslänge einer Matrix** $A = (a_{ij}) \in \mathbb{Q}^{(m,n)}$ (oder analog eines Vektors) ist gegeben durch

$$\langle A \rangle := \sum_{i=1}^{m} \sum_{j=1}^{n} \langle a_{ij} \rangle. \tag{12.3}$$

Daraus folgt, daß die Kodierungslänge eines linearen Programms der Form $\max c^T x$, $Ax \leq b$ gegeben ist durch

$$\langle c \rangle + \langle A \rangle + \langle b \rangle.$$

Um über Laufzeiten sprechen zu können, müssen wir uns überlegen, wie wir die Laufzeit eines Algorithmus messen wollen. Wir legen fest, dass wir dazu zunächst die

Anzahl der elementaren Rechenschritte

zählen wollen, wobei elementare Rechenschritte die arithmetischen Operationen

Addition, Subtraktion, Multiplikation, Division, Vergleich

von ganzen Zahlen oder rationalen Zahlen sind. Dies reicht aber noch nicht aus, denn wie jeder weiß, dauert z. B. die Multiplikation großer ganzer Zahlen länger als die Multiplikation kleiner ganzer Zahlen. Wir müssen also die Zahlengrößen in Betracht ziehen und jede elementare Rechenoperation mit der Kodierungslänge der dabei involvierten Zahlen multiplizieren. Für unsere Zwecke ist folgende Definition angemessen.

(12.4) Definition. *Die* **Laufzeit eines Algorithmus** $\mathcal{A}$ *zur Lösung eines Problems* Π *(kurz* $L_{\mathcal{A}}(\Pi)$*) ist die Anzahl der elementaren Rechenschritte, die während* *der Ausführung des Algorithmus durchgeführt wurden, multipliziert mit der Kodierungslänge der (bezüglich ihrer Kodierungslänge) größten Zahl, die während der Ausführung des Algorithmus aufgetreten ist.*

□

Die beiden folgenden Begriffe sind entscheidend für das Verständnis des Weiteren.

(12.5) Definition. *Sei $\mathcal{A}$ ein Algorithmus zur Lösung einer Klasse von Problemen* (z. B. die Klasse aller linearen Optimierungsprobleme). *Die Elemente (Probleme) dieser Klasse (z. B. spezielle lineare Programme) bezeichnen wir mit* Π.

(a) *Die Funktion* $f : \mathbb{N} \to \mathbb{N}$ *definiert durch*

$$f_{\mathcal{A}}(n) := \max\{L_{\mathcal{A}}(\Pi) \mid \text{ die Kodierungslänge der Daten zur Beschreibung von } \Pi \text{ ist höchstens } n\}$$

heißt **Laufzeitfunktion** *von* $\mathcal{A}$.

(b) *Der Algorithmus* $\mathcal{A}$ *hat eine* **polynomiale Laufzeit** *(kurz:* $\mathcal{A}$ *ist ein* **polynomialer Algorithmus**), *wenn es ein Polynom* $p : \mathbb{N} \to \mathbb{N}$ *gibt, so daß*

$$f_{\mathcal{A}}(n) \leq p(n) \quad \forall\, n \in \mathbb{N} \text{ gilt.}$$

□

Polynomiale Algorithmen sind also solche Verfahren, deren Schrittzahl multipliziert mit der Kodierungslänge der größten auftretenden Zahl durch ein Polynom in der Kodierungslänge der Problemdaten beschränkt werden kann. Für die Klasse der linearen Programme der Form $\max c^T x,\ Ax \leq b$ muss also die Laufzeit eines Algorithmus durch ein Polynom in $\langle A \rangle + \langle b \rangle + \langle c \rangle$ beschränkt werden können, wenn er polynomial sein soll. Wie die Klee & Minty-Beispiele zeigen, ist der Simplexalgorithmus kein polynomialer Algorithmus zur Lösung linearer Programme!

12.2 Reduktionen

Die Ellipsoidmethode ist kein Optimierungsverfahren, sondern lediglich eine Methode, die in einem gegebenen Polyeder einen Punkt findet, falls ein solcher existiert. Wir müssen daher das allgemeine lineare Optimierungsproblem auf diesen

Fall zurückführen. Aus Kapitel 2 (allgemeine Transformationsregeln) wissen wir, dass jedes lineare Programm in der Form

$$\begin{aligned} &\max c^T x \\ &Ax \le b \\ &x \ge 0 \end{aligned} \tag{12.6}$$

geschrieben werden kann. Das zu (12.6) duale Programm ist

$$\begin{aligned} &\min b^T y \\ &A^T y \ge c \\ &y \ge 0 \end{aligned} \quad . \tag{12.7}$$

Aus dem Dualitätssatz (5.14) wissen wir, dass (12.6) und (12.7) genau dann optimale Lösungen haben, deren Werte gleich sind, wenn beide Programme zulässige Lösungen haben. Daraus folgt, dass jedes Element $\binom{x}{y}$ des Polyeders P, definiert durch

$$\begin{pmatrix} -c^T & b^T \\ A & 0 \\ 0 & -A^T \\ -I & 0 \\ 0 & -I \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} \le \begin{pmatrix} 0 \\ b \\ -c \\ 0 \\ 0 \end{pmatrix} \tag{12.8}$$

eine Optimallösung x von (12.6) und eine Optimallösung y von (12.7) bestimmt. Dies impliziert, dass es zur Lösung von (12.6) genügt, einen Punkt in (12.8) zu finden.

Der obige "Trick", das Primal- und das Dualprogramm zusammenzufassen bläht natürlich das zu bearbeitende System ungeheuer auf. Eine andere Methode der Reduktion ist die binäre Suche, die wir nun schildern. Zur korrekten Darstellung benötigen wir jedoch noch einige (auch für andere Zwecke) wichtige Hilfssätze. Zunächst wollen wir einige Parameter durch die Kodierungslänge abschätzen. Zum Beweis des ersten Hilfssatzes benötigen wir die bekannte Hadamard-Ungleichung, die besagt, dass das Volumen eines Parallelepipeds in $\mathbb{R}^n$ mit Kantenlängen $d_1, \ldots, d_n$ nicht größer ist als das Volumen des Würfels mit Kantenlängen $d_1, \ldots, d_n$. Bekanntlich ist das Volumen eines Parallelepipeds, das durch die Vektoren $a_1, \ldots, a_n$ aufgespannt wird, nichts anderes als der Absolutbetrag der Determinante der Matrix A mit Spalten $A_{\cdot 1} = a_1, \ldots, A_{\cdot n} = a_n$. Die Hadamard-Ungleichung lautet also

$$|\det A| \le \prod_{j=1}^{n} \|A_{\cdot j}\|_2 \, . \tag{12.9}$$

(12.10) Lemma.

(a) *Für jede Zahl* $r \in \mathbb{Q}$ *gilt:* $|r| \leq 2^{\langle r \rangle - 1} - 1$.

(b) *Für je zwei rationale Zahlen* r, s *gilt:* $\langle rs \rangle \leq \langle r \rangle + \langle s \rangle$.

(c) *Für jeden Vektor* $x \in \mathbb{Q}^n$ *gilt:* $\|x\|_2 \leq \|x\|_1 \leq 2^{\langle x \rangle - n} - 1$.

(d) *Für jede Matrix* $A \in \mathbb{Q}^{(n,n)}$ *gilt:* $|\det(A)| \leq 2^{\langle A \rangle - n^2} - 1$.

Beweis :

(a) und (b) folgen direkt aus der Definition.

(c) Sei $x = (x_1, \ldots, x_n)^T \in \mathbb{Q}^n$, dann gilt nach (a)

$$\begin{aligned} 1 + \|x\|_1 = 1 + \sum_{i=1}^n |x_i| &\leq \prod_{i=1}^n (1 + |x_i|) \\ &\leq \prod_{i=1}^n 2^{\langle x_i \rangle - 1} \\ &= 2^{\langle x \rangle - n}. \end{aligned}$$

Die Ungleichung $\|x\|_2 = \sqrt{\sum_{i=1}^n x_i^2} \leq \sum_{i=1}^n |x_i| = \|x\|_1$ ist trivial.

(d) Aus der Hadamard-Ungleichung (12.9) und (c) folgt

$$\begin{aligned} 1 + |\det(A)| &\leq 1 + \prod_{j=1}^n \|A_{\cdot j}\|_2 \leq \prod_{j=1}^n (1 + \|A_{\cdot j}\|_2) \\ &\leq \prod_{j=1}^n 2^{\langle A_{\cdot j} \rangle - n} = 2^{\langle A \rangle - n^2}. \end{aligned}$$

□

Diesen Hilfssatz können wir zur Abschätzung der "Größe" der Ecken von Polyedern wie folgt benutzen.

(12.11) Satz. *Für jede Ecke* $v = (v_1, \ldots, v_n)^T$ *eines Polyeders* P *der Form* $P(A, b)$, $P^=(A, b)$ *oder* $P = \{x \in \mathbb{R}^n \mid Ax \leq b, x \geq 0\}$, $A \in \mathbb{Q}^{(m,n)}$, $b \in \mathbb{Q}^m$, *gilt*

(a) *Der Absolutbetrag des Zählers von* v_i *ist höchstens* $2^{\langle A \rangle + \langle b \rangle - n^2}$, *der Absolutbetrag des Nenners von* v_i *ist höchstens* $2^{\langle A \rangle - n^2}$, $i = 1, \ldots, n$.

(b) $|v_i| \leq 2^{2\langle A \rangle + \langle b \rangle - 2n^2}$, $i = 1, \ldots, n$.

(c) *Falls* $A \in \mathbb{Z}^{(m,n)}$, *so gilt* $|v_i| \leq 2^{\langle A \rangle + \langle b \rangle - n^2}$ $i = 1, \ldots, n$.

Beweis :

Ist v Ecke von P, so gibt es nach Satz (8.9) eine reguläre Untermatrix und einen entsprechenden Untervektor des Restriktionensystems von P, so dass v die eindeutig bestimmte Lösung des hierdurch bestimmten Gleichungssystems ist. Wir führen den Fall $P = \{x \mid Ax \leq b, x \geq 0\}$ vor. Die beiden anderen Fälle verlaufen analog. Es gibt also eine (n, n)-Matrix D, deren Zeilen Zeilen von A oder negative Einheitsvektoren sind und einen entsprechenden Vektor d, dessen Komponenten Elemente von b oder Nullen sind, so dass v die eindeutige Lösung von $Dx = d$ ist. Nach Cramer's Regel gilt dann für $i = 1, \dots, n$

$$v_i = \frac{\det D_i}{\det D}$$

wobei D_i aus D dadurch entsteht, dass die i-te Spalte von D durch den Vektor d ersetzt wird. Hat D Zeilen, die negative Einheitsvektoren sind, so bezeichnen wir mit $\overline{D}$ die Matrix, die durch Streichen dieser Zeilen und der Spalten, in denen sich die Elemente -1 befinden, entsteht. Aufgrund des Determinantenentwicklungssatzes gilt $|\det D| = |\det \overline{D}|$, und ferner ist $\overline{D}$ eine Untermatrix von A. Daraus folgt mit (12.10) (d)

$$|\det D| = |\det \overline{D}| \leq 2^{\langle \overline{D} \rangle - n^2} \leq 2^{\langle A \rangle - n^2}.$$

Analog folgt

$$|\det D_i| \leq 2^{\langle A \rangle + \langle b \rangle - n^2}.$$

Damit ist (a) bewiesen.

Ist $|\det D| \geq 1$, dies ist z. B. dann der Fall, wenn $A \in \mathbb{Z}^{(m,n)}$ gilt, so erhalten wir mit (a)

$$|v_i| \leq |\det D_i| \leq 2^{\langle A \rangle + \langle b \rangle - n^2}.$$

Hieraus folgt (c). Ist $|\det D| < 1$, so müssen wir $|\det D|$ nach unten abschätzen. Sei $q = \prod_{i,j} q_{ij}$ das Produkt der Nenner der Elemente $d_{ij} = \frac{p_{ij}}{q_{ij}}$ von D. Dann gilt $|\det D| = \frac{q|\det D|}{q}$, und sowohl q als auch $q|\det D|$ sind ganze Zahlen. Aus (12.10) (a), (b) folgt

$$q = \prod_{i,j} q_{ij} \leq 2^{\langle \Pi q_{ij} \rangle} \leq 2^{\sum \langle q_{ij} \rangle} \leq 2^{\langle D \rangle - n^2} \leq 2^{\langle A \rangle - n^2}.$$

Daraus ergibt sich

$$|v_i| \leq \frac{|\det D_i|}{|\det D|} = \frac{|\det D_i| q}{q|\det D|} \leq |\det D_i| q \leq 2^{2\langle A \rangle + \langle b \rangle - 2n^2}.$$

Damit ist auch (b) bewiesen □

Da nach Satz (8.12) alle Polyeder P der Form $P = \{x \mid Ax \leq b, x \geq 0\}$ spitz sind und nach Folgerung (8.15) lineare Programme über spitzen Polyedern optimale Ecklösungen haben (falls sie überhaupt Optimallösungen haben) folgt:

(12.12) Satz. *Das lineare Programm (12.6) hat eine Optimallösung genau dann, wenn die beiden linearen Programme*

$$(12.13)\qquad \begin{array}{ll} \max c^T x & \\ Ax & \leq b \\ x & \geq 0 \\ x_i & \leq 2^{2\langle A\rangle + \langle b\rangle - 2n^2}\,,\ i = 1, \ldots, n \end{array}$$

$$(12.14)\qquad \begin{array}{ll} \max c^T x & \\ Ax & \leq b \\ x & \geq 0 \\ x_i & \leq 2^{2\langle A\rangle + \langle b\rangle - 2n^2} + 1\,,\ i = 1, \ldots, n \end{array}$$

eine Optimallösung haben und die Werte der Optimallösungen übereinstimmen. Die Zielfunktionswerte von (12.13) und (12.14) stimmen genau dann nicht überein, wenn (12.6) unbeschränkt ist. (12.6) ist genau dann nicht lösbar, wenn (12.13) oder (12.14) nicht lösbar ist.

□

Wir haben damit das lineare Programmierungsproblem (12.6) über einem (allgemeinen) Polyeder auf die Lösung zweier linearer Programme über Polytopen reduziert. Wir müssen also im Prinzip nur zeigen, wie man LP's über Polytopen löst.

(12.15) Satz. *Ist $P \neq \emptyset$ ein Polytop der Form $P(A, b)$, $P^=(A, b)$ oder $P = \{x \mid Ax \leq b, x \geq 0\}$ mit $A \in \mathbb{Q}^{(m,n)}$, $b \in \mathbb{Q}^m$, so kann für $c \in \mathbb{Q}^n$ das lineare Programm $\max\ c^T x, x \in P$ nur Optimalwerte in der endlichen Menge*

$$S := \{\tfrac{p}{q} \in \mathbb{Q} \mid |p| \leq n 2^{\langle A\rangle + \langle b\rangle + 2\langle c\rangle - n^2 - n}, 1 \leq q \leq 2^{\langle A\rangle + \langle c\rangle - n^2 - n}\}$$

annehmen.

Beweis : Da alle Optimalwerte Zielfunktionswerte von Ecken von P sind, brauchen wir nur die Werte $c^T v$, v Ecke von P, abzuschätzen. Sei also $v = (v_1, \ldots, v_n)^T$

eine Ecke von P. Wie im Beweis von (12.11) können wir feststellen, dass die Komponenten v_i von v eine Darstellung der Form

$$v_i = \frac{\det D_i}{\det D} =: \frac{p_i}{d}$$

haben, wobei D eine reguläre Untermatrix von A bzw. $\binom{A}{I}$ ist und D_i aus D dadurch hervorgeht, dass die i-te Spalte von D durch einen Untervektor der rechten Seite ersetzt wird. Satz (12.11) zeigt $d \le 2^{\langle A \rangle - n^2}$, $p_i \le 2^{\langle A \rangle + \langle b \rangle - n^2}$. Sei nun $c = (\frac{s_1}{t_1}, \dots, \frac{s_n}{t_n})^T \in \mathbb{Q}^n$ eine teilerfremde Darstellung der Zielfunktion, so gilt

$$c^T v = \sum_{i=1}^{n} \frac{s_i p_i}{t_i d} = \frac{1}{dt} \sum_{i=1}^{n} s_i p_i \bar{t}_i \ \text{ mit } t := t_1 \cdot \ldots \cdot t_n,\ \bar{t}_i := \frac{t}{t_i}.$$

Aus (12.10) (a) folgt $t = t_1 \cdot \ldots \cdot t_n \le 2^{\langle t_1 \rangle - 1} \cdot \ldots \cdot 2^{\langle t_n \rangle - 1} = 2^{\sum(\langle t_i \rangle - 1)} \le 2^{\langle c \rangle - n}$ und somit

$$q := dt \le 2^{\langle A \rangle + \langle c \rangle - n^2 - n}.$$

Analog folgt

$$p := \sum_{i=1}^{n} s_i p_i \bar{t}_i \le \sum_{i=1}^{n} 2^{\langle s_i \rangle - 1} 2^{\langle A \rangle + \langle b \rangle - n^2} 2^{\langle c \rangle - n} \le n 2^{\langle A \rangle + \langle b \rangle + 2\langle c \rangle - n^2 - n}.$$

□

Satz (12.15) gibt uns nun die Möglichkeit, das Verfahren der binären Suche zur Lösung von $\max c^T x, x \in P$ anzuwenden. Diese Methode funktioniert bezüglich der in Satz (12.15) definierten Menge S wie folgt.

(12.16) Binäre Suche.

(1) *Wähle ein Element* $s \in S$, *so dass für* $S' := \{t \in S \mid t < s\}$ *und* $S'' := \{t \in S \mid t \ge s\}$ *gilt*

$$|S'| \le |S''| \le |S'| + 1.$$

(2) *Überprüfe, ob das Polyeder*

$$P_s = \{x \mid Ax \le b, x \ge 0, c^T x \ge s\}$$

nicht leer ist.

(3) *Ist* P_s *leer, so setze* $S := S'$, *andernfalls setze* $S := S''$.

(4) *Ist* $|S| = 1$*, so gilt für* $s \in S$*:* $s = \max\{c^T x \mid Ax \leq b, x \geq 0\}$*, und jeder Punkt in* P_s *ist eine Optimallösung von* $\max\ c^T x, x \in P$*. Andernfalls gehe zu (1).*

□

Die Korrektheit dieses Verfahrens ist offensichtlich. Ist die Binärsuche auch effizient? Da in jedem Schritt die Kardinalität $|S|$ von S (fast) halbiert wird, ist klar, dass höchstens

$$\text{(12.17)} \qquad \overline{\overline{N}} := \lceil \log_2(|S| + 1) \rceil$$

Aufspaltungen von S in zwei Teile notwendig sind, um eine einelementige Menge zu erhalten. Also muss Schritt (2) von (12.16) $\overline{\overline{N}}$-mal ausgeführt werden. Nach (12.15) gilt

$$|S| \leq 2n2^{2\langle A\rangle + \langle b\rangle + 3\langle c\rangle - 2n^2 - 2n} + 1\,,$$

und daraus folgt, dass Test (2) von (12.16) höchstens

$$\text{(12.18)} \qquad \overline{N} := 2\langle A\rangle + \langle b\rangle + 3\langle c\rangle - 2n^2 - 2n + \log_2 n + 2$$

mal durchgeführt wurde. Ist Test (2) in einer Zeit ausführbar, die polynomial in $\langle A\rangle + \langle b\rangle + \langle c\rangle$ ist, so ist auch die binäre Suche ein polynomialer Algorithmus, da ein polynomialer Algorithmus für (2) nur $\overline{N}$-mal also nur polynomial oft ausgeführt werden muss.

(12.19) Folgerung. *Es gibt einen polynomialen Algorithmus zur Lösung linearer Programme genau dann, wenn es einen Algorithmus gibt, der in polynomialer Zeit entscheidet, ob ein Polytop* P *leer ist oder nicht und der, falls* $P \neq \emptyset$*, einen Punkt in* P *findet.*

□

Damit haben wir das lineare Programmierungsproblem reduziert auf die Frage der Lösbarkeit von Ungleichungssystemen, deren Lösungsmenge beschränkt ist.

12.3 Beschreibung der Ellipsoidmethode

In diesem Abschnitt setzen wir $n \geq 2$ voraus. Wir wollen zunächst einige Eigenschaften von Ellipsoiden beschreiben. Wir erinnern daran, dass Ellipsoide volldimensionale konvexe Körper im $\mathbb{R}^n$ sind, die eine besonders einfache Darstellung haben. Und zwar ist eine Menge $E \subseteq \mathbb{R}^n$ ein **Ellipsoid (mit Zentrum a)** genau dann, wenn es einen Vektor $a \in \mathbb{R}^n$ und eine (symmetrische) positiv definite Matrix A gibt, so dass gilt

(12.20) $$E = E(A,a) := \{x \in \mathbb{R}^n \mid (x-a)^T A^{-1}(x-a) \leq 1\}.$$

Aus der linearen Algebra wissen wir, dass für symmetrische Matrizen $A \in \mathbb{R}^{(n,n)}$ die folgenden Aussagen äquivalent sind:

(12.21)

(i) A ist positiv definit.

(ii) A^{-1} ist positiv definit.

(iii) Alle Eigenwerte von A sind positive reelle Zahlen.

(iv) $\det(A_{II}) > 0$ für alle Mengen $I = \{1, \ldots, i\}, i = 1, \ldots, n$.

(v) $A = B^T B$ für eine reguläre Matrix $B \in \mathbb{R}^{(n,n)}$.

Das Ellipsoid $E(A,a)$ ist durch die (ebenfalls positiv definite) Inverse A^{-1} von A definiert. Das erscheint zunächst seltsam, liegt aber daran, dass sich viele Eigenschaften von $E(A,a)$ aus algebraischen Eigenschaften von A ableiten lassen. Z. B. ist der Durchmesser (d. h. die Länge der längsten Achse) von $E(A,a)$ gleich $2\sqrt{\Lambda}$, wobei Λ der größte Eigenwert von A ist. Die längsten Achsen sind durch die Eigenvektoren, die zu Λ gehören, gegeben. Die Symmetrieachsen von $E(A,a)$ entsprechen ebenfalls den Eigenvektoren von A. In Abbildung 12.1 ist das Ellipsoid $E(A,0)$ dargestellt mit

$$A = \begin{pmatrix} 16 & 0 \\ 0 & 4 \end{pmatrix}.$$

Die Eigenwerte von A sind $\Lambda = 16$ und $\lambda = 4$ mit den zugehörigen Eigenvektoren $e_1 = (1,0)^T$ und $e_2 = (0,1)^T$. Der Durchmesser von $E(A,0)$ ist $2\sqrt{\Lambda} = 8$.

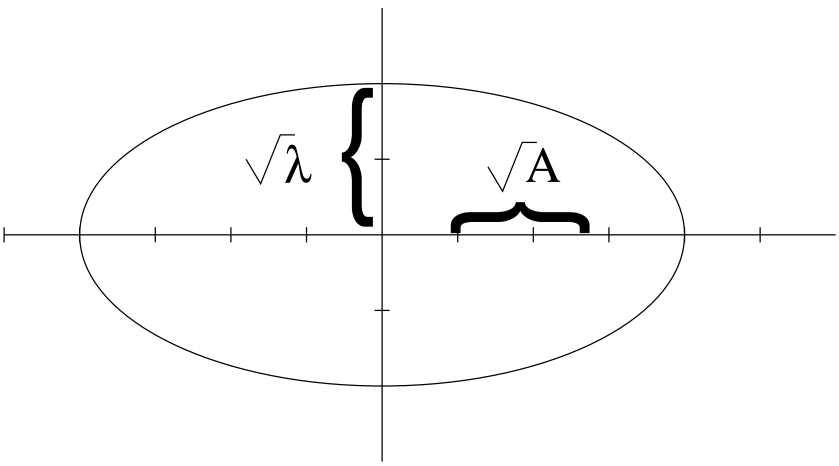


Abb. 12.1

Zu jeder positiv definiten Matrix A gibt es eine eindeutig bestimmte symmetrische reguläre Matrix, die mit $A^{\frac{1}{2}}$ bezeichnet und **Wurzel von** A genannt wird, mit

$$A = A^{\frac{1}{2}} A^{\frac{1}{2}}.$$

Die Einheitskugel $S(0, 1) \subseteq \mathbb{R}^n$ (um den Nullpunkt mit Radius 1) ist das Ellipsoid $E(I, 0)$. Man rechnet leicht nach, dass gilt:

$$E(A, a) = A^{\frac{1}{2}} S(0, 1) + a. \tag{12.22}$$

Damit sei die Aufzählung von Eigenschaften von Ellipsoiden beendet. Wir wollen nun die geometrische Idee, die hinter der Ellipsoidmethode steckt, erläutern.

(12.23) Geometrische Beschreibung der Ellipsoidmethode. *Gegeben sei ein Polytop P. Wir wollen einen Punkt in P finden oder beweisen, dass P leer ist.*

(1) Konstruiere ein Ellipsoid $E_0 = E(A_0, a_0)$, das P enthält. Setze $k := 0$.

(2) Ist das gegenwärtige Ellipsoid "zu klein", so brich ab mit der Antwort "P ist leer".

(3) Teste ob der Mittelpunkt a_k von E_k in P enthalten ist.

(4) Gilt $a_k \in P$, dann haben wir unser Ziel erreicht, STOP.

(5) Gilt $a_k \notin P$, dann gibt es eine P definierende Ungleichung, sagen wir

$$c^T x \leq \gamma,$$

die von a_k verletzt wird, d. h. $c^T a_k > \gamma$. Mit

$$E'_k := E_k \cap \{x \mid c^T x \leq c^T a_k\}$$

bezeichnen wir das Halbellipsoid von E_k, das P enthält. Wir konstruieren nun das Ellipsoid kleinsten Volumens, das E'_k enthält und nennen es E_{k+1}.

(6) Setze $k := k + 1$ und gehe zu (2).

$\square$

Das Prinzip des Verfahrens ist klar. Man zäunt das Polytop P durch ein Ellipsoid E_k ein. Ist das Zentrum von E_k nicht in P, so sucht man sich ein kleineres Ellipsoid E_{k+1}, das P enthält und fährt so fort. Die Probleme, die zu klären sind, sind die folgenden:

– Wie findet man ein Ellipsoid, das P enthält?

– Wie kann man E_{k+1} aus E_k konstruieren?

– Wann kann man abbrechen, d. h. was heißt "zu klein"?

– Wieviele Iterationen des Verfahrens sind durchzuführen?

Die nachfolgenden Sätze beantworten diese Fragen, wobei wir nur einige der Beweise, die zum Nachweis der Polynomialität des Verfahrens notwendig sind, angeben wollen.

Unser Anfangsellipsoid soll eine Kugel mit dem Nullpunkt als Zentrum sein. Enthalten die Restriktionen, die das Polytop P definieren, explizite obere und unter Schranken für die Variablen, sagen wir

$$l_i \leq x_i \leq u_i\,,\ i = 1, \dots, n$$

so sei

(12.24)
$$R := \sqrt{\sum_{i=1}^{n} (\max\{|u_i|, |l_i|\})^2}.$$

Dann gilt trivialerweise $P \subseteq S(0, R) = E(R^2 I, 0)$. Andernfalls kann man zeigen

(12.25) Lemma. *Sei P ein Polyeder der Form $P(A, b)$, $P^=(A, b)$ oder $\{x \in \mathbb{R}^n \mid Ax \leq b, x \geq 0\}$ mit $A \in \mathbb{Q}^{(m,n)}$, $b \in \mathbb{Q}^m$, dann gilt*

(a) Alle Ecken von P sind in der Kugel $S(0, R)$ enthalten mit

$$R := \sqrt{n}2^{2\langle A\rangle + \langle b\rangle - 2n^2}.$$

(b) Ist P ein Polytop, so gilt $P \subseteq S(0, R) = E(R^2 I, 0)$.

Beweis : Nach Satz (12.11) gilt für jede Ecke $v^T = (v_1, \dots, v_n)$ von P

$$|v_i| \leq 2^{2\langle A\rangle + \langle b\rangle - 2n^2} \quad (i = 1, \dots, n),$$

und daraus folgt für die euklidische Norm von v:

$$\|v\| = \sqrt{\sum_{i=1}^{n} v_i^2} \leq \sqrt{n \max\{v_i^2\}} \leq \sqrt{n}\ 2^{2\langle A\rangle + \langle b\rangle - 2n^2}.$$

Also ist jede Ecke von P in $S(0, R)$ enthalten. Ist insbesondere P ein Polytop, so folgt daraus $P \subseteq S(0, R)$. □

Damit haben wir durch (12.24) oder (12.25) ein Anfangsellipsoid E_0 gefunden, mit dem wir die Ellipsoidmethode beginnen können. Die Konstruktion von E_{k+1} aus E_k geschieht wie folgt.

(12.26) Satz. *Sei $E_k = E(A_k, a_k) \subseteq \mathbb{R}^n$ ein Ellipsoid, $c \in \mathbb{R}^n \setminus \{0\}$ und $E'_k := \{x \in \mathbb{R}^n \mid c^T x \leq c^T a_k\} \cap E_k$. Setze*

(12.27) $$d := \frac{1}{\sqrt{c^T A_k c}} A_k c,$$

(12.28) $$a_{k+1} := a_k - \frac{1}{n+1} d,$$

(12.29) $$A_{k+1} := \frac{n^2}{n^2-1}\left(A_k - \frac{2}{n+1} dd^T\right),$$

dann ist A_{k+1} positiv definit und $E_{k+1} := E(A_{k+1}, a_{k+1})$ ist das eindeutig bestimmte Ellipsoid minimalen Volumens, das E'_k enthält.

Beweis : Siehe R. G. Bland, D. Goldfarb & M. J. Todd, "The Ellipsoid Method: A Survey", Operations Research 29 (1981) 1039–1091 oder M. Grötschel, L. Lovász & A. Schrijver, "The Ellipsoid Method and Combinatorial Optimization", Springer, Heidelberg, erscheint 1985 oder 1986. □

Wir wollen kurz den geometrischen Gehalt der Schritte in Satz (12.26) erläutern.

Ist $c \in \mathbb{R}^n$, und wollen wir das Maximum oder Minimum von $c^T x$ über E_k finden, so kann man dies explizit wie folgt angeben. Für den in (12.27) definierten Vektor d und

$$z_{\max} := a_k + d, \quad z_{\min} := a_k - d$$

gilt

$$\begin{aligned} c^T z_{\max} &= \max\{c^T x \mid x \in E_k\} = c^T a_k + \sqrt{c^T A_k c}\,, \\ c^T z_{\min} &= \min\{c^T x \mid x \in E_k\} = c^T a_k - \sqrt{c^T A_k c}\,. \end{aligned}$$

Diese Beziehung kann man leicht — z. B. aus der Cauchy-Schwarz-Ungleichung — ableiten. Daraus folgt, dass der Mittelpunkt a_{k+1} des neuen Ellipsoids E_{k+1} auf dem Geradenstück zwischen a_k und $z_{\min}$ liegt. Die Länge dieses Geradenstücks ist $\|d\|$, und a_{k+1} erreicht man von a_k aus, indem man einen Schritt der Länge $\frac{1}{n+1}\|d\|$ in Richtung $-d$ macht. Der Durchschnitt des Randes des Ellipsoids E_{k+1} mit dem Rand von E'_k wird gebildet durch den Punkt $z_{\min}$ und $E''_k := \{x \mid (x -$

$a_k)^T A_k(x - a_k) = 1\} \cap \{x \mid c^T x = c^T a_k\}$. E''_k ist der Rand eines "$(n-1)$-dimensionalen Ellipsoids" im $\mathbb{R}^n$.

In Abbildung 12.2 sind das Ellipsoid E_k aus Abbildung 12.1 und das Ellipsoid E_{k+1}, definiert durch Satz (12.26) bezüglich des Vektors $c^T = (-1, -2)$ dargestellt. E'_k ist gestrichelt eingezeichnet. Als Anfangsdaten haben wir daher: Die verletzte Ungleichung ist $-x_1 - 2x_2 \leq -3$. Die zugehörige Gerade $\{x \mid -x_1 - 2x_2 = -3\}$ ist ebenfalls gezeichnet.

$$a_k = \begin{pmatrix} 0 \\ 0 \end{pmatrix}, \quad A_k = \begin{pmatrix} 16 & 0 \\ 0 & 4 \end{pmatrix}, \quad c = \begin{pmatrix} -1 \\ -2 \end{pmatrix}, .$$

Die Formeln (12.27), (12.28), (12.29) ergeben:

$$\begin{aligned}
d &= \tfrac{-1}{\sqrt{2}} \tbinom{4}{2}, \\
a_{k+1} &= a_k - \tfrac{1}{3} d = \tfrac{1}{3\sqrt{2}} \tbinom{4}{2} \approx \tbinom{0.9428}{0.4714}, \\
A_{k+1} &= \tfrac{n^2}{n^2-1}(A_k - \tfrac{2}{n+1} dd^T) = \tfrac{4}{9} \begin{pmatrix} 32 & -8 \\ -8 & 8 \end{pmatrix}, \\
E_{k+1} &= E(A_{k+1}, a_{k+1}).
\end{aligned}$$

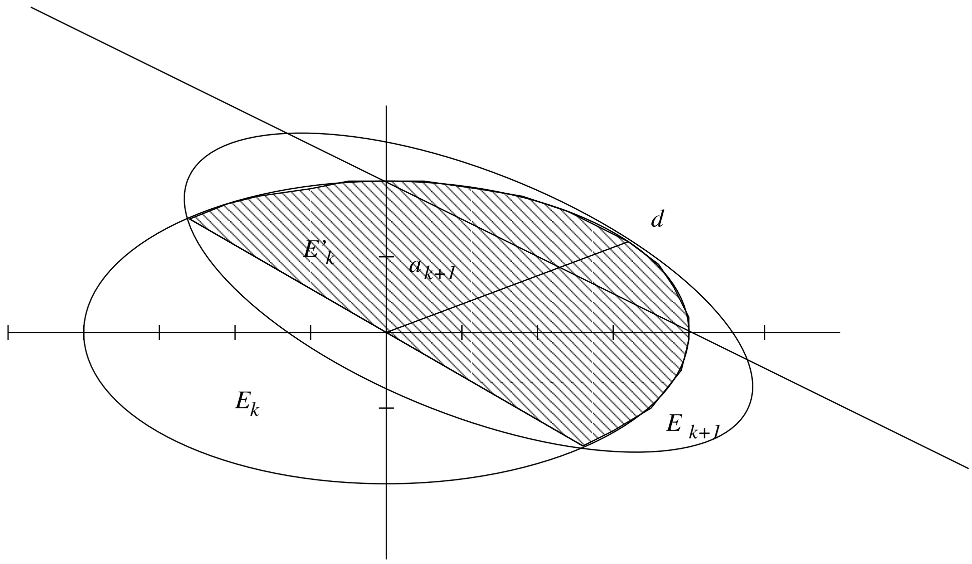


Abb. 12.2

Das Stopkriterium der Ellipsoidmethode beruht auf einem Volumenargument. Nach Konstruktion ist klar, dass das Volumen von E_{k+1} (bezeichnet mit $\text{vol}(E_{k+1})$) kleiner ist als das von E_k. Man kann den Volumenschrumpfungsfaktor explizit berechnen. Er hängt nur von der Dimension n des Raumes $\mathbb{R}^n$ und nicht etwa von Update-Vektor c ab.

(12.30) Lemma.

$$\frac{\mathrm{vol}(E_{k+1})}{\mathrm{vol}(E_k)} = \left(\left(\frac{n}{n+1}\right)^{n+1}\left(\frac{n}{n-1}\right)^{n-1}\right)^{\frac{1}{2}} \leq e^{\frac{-1}{2n}} < 1.$$

Beweis : Siehe Grötschel, Lovász & Schrijver (1985). □

Wir sehen also, dass mit den Formeln aus Satz (12.26) eine Folge von Ellipsoiden konstruiert werden kann, so dass jedes Ellipsoid E_{k+1} das Halbellipsoid E'_k und somit das Polytop P enthält und dass die Volumina der Ellipsoide schrumpfen. Die Folge der Volumina konvergiert gegen Null, muss also einmal das Volumen von P, falls P ein positives Volumen hat, unterschreiten. Daher muss nach endlich vielen Schritten der Mittelpunkt eines der Ellipsoide in P sein, falls $P \neq \emptyset$. Wir wollen nun ausrechnen, nach wievielen Schritten, dies der Fall ist.

(12.31) Lemma. *Seien* $P = P(A, b) \subseteq \mathbb{R}^n$, $\overset{\circ}{P} = \{x \in \mathbb{R}^n \mid Ax < b\}$, *und* $R := \sqrt{n}2^{2\langle A\rangle+\langle b\rangle-2n^2}$.

(a) *Entweder gilt* $\overset{\circ}{P} = \emptyset$ *oder*

$$\mathrm{vol}(\overset{\circ}{P} \cap S(0,R)) \geq 2^{-(n+1)(\langle A\rangle+\langle b\rangle-n^2)}.$$

(b) *Ist* P *volldimensional (d. h.* $\dim P = n$*), dann gilt*

$$\mathrm{vol}(P \cap S(0,R)) = \mathrm{vol}(\overset{\circ}{P} \cap S(0,R)) > 0.$$

□

Lemma (12.31) zusammen mit Lemma (12.25) sind geometrisch und algorithmisch interessant. Die beiden Hilfssätze implizieren folgendes.

- Wenn ein Polyeder P nicht leer ist, dann gibt es Elemente des Polyeders, die "nah" beim Nullvektor liegen, d. h. in $S(0, R)$ enthalten sind.
- Es reicht aus zu überprüfen, ob $P \cap S(0, R)$ leer ist oder nicht. Daraus kann man schließen, dass P leer ist oder nicht.
- Will man einen Konvergenzbeweis über Volumen führen, so sollte man strikte statt normale Ungleichungssysteme betrachten. Denn ein striktes Ungleichungssystem ist entweder unlösbar oder seine Lösungsmenge hat ein positives (relativ großes) Volumen.

Damit können wir die Ellipsoidmethode formulieren.

(12.32) Die Ellipsoidmethode.

Input: $A \in \mathbb{Q}^{(m,n)}$, $b \in \mathbb{Q}^m$.

Output: *Ein Vektor* $x \in \mathbb{Q}^n$ *mit* $Ax \leq b$ *oder die Feststellung, dass* $\{x \in \mathbb{R}^n \mid Ax < b\}$ *leer ist.*

(1) **Initialisierung:** *Setze*

$$\begin{aligned}
A_0 :=\ & R^2 I,\ \text{mit } R = \sqrt{n}\, 2^{2\langle A\rangle + \langle b\rangle - 2n^2} \text{(oder } R \text{ durch andere Vorinformationen kleiner gewählt)},\\
a_0 :=\ & 0,\\
k :=\ & 0,\\
N :=\ & 2n((3n+1)\langle A\rangle + (2n+1)\langle b\rangle - n^3).
\end{aligned}$$

(Das Anfangsellipsoid ist $E_0 := E(A_0, a_0)$.)

(2) **Abbruchkriterium:**

(2.a) Gilt $k = N$, *dann hat* $Ax < b$ *keine Lösung, STOP!*

(2.b) Gilt $Aa_k \leq b$, *dann ist eine Lösung gefunden, STOP!*

(2.c) Andernfalls sei c^T *irgendeine Zeile von* A *derart, dass der Mittelpunkt* a_k *von* E_k *die entsprechende Ungleichung verletzt.*

(3) **Update:** *Setze*

(3.a) $a_{k+1} := a_k - \frac{1}{n+1} \frac{1}{\sqrt{c^T A_k c}} A_k c$

(3.b) $A_{k+1} := \frac{n^2}{n^2-1}(A_k - \frac{2}{n+1} \frac{1}{c^T A_k c} A_k c c^T A_k^T)$
($E_{k+1} := E(A_{k+1}, a_{k+1})$ ist das neue Ellipsoid.)
$k := k + 1$.

Gehe zu (2).

□

(12.33) Satz. *Die Ellipsoidmethode arbeitet korrekt.*

Beweis : Gibt es ein $k \leq N$, so dass $Aa_k \leq b$, so ist offenbar ein Vektor aus $P(A,b)$ gefunden. Bricht die Ellipsoidmethode in Schritt (2.a) ab, so müssen wir zeigen, dass $\overset{\circ}{P}:= \{x \mid Ax < b\}$ kein Element besitzt.

Angenommen $\overset{\circ}{P}\neq \emptyset$. Sei P' der Durchschnitt von $\overset{\circ}{P}$ mit E_0. Dann gilt nach Lemma (12.31), dass das Volumen von P' mindestens $2^{-(n+1)(\langle A\rangle+\langle b\rangle-n^2)}$ beträgt. Ist a_k nicht in $P(A,b)$, $0 \leq k < N$, so wird in (2.c) eine verletzte Ungleichung gefunden, sagen wir $c^T x \leq \gamma$. Wegen $\gamma < c^T a_k$ enthält das Halbellipsoid

$$E'_k := E_k \cap \{x \mid c^T x \leq c^T a_k\}$$

die Menge P'. Das durch die Formeln (3.a), (3.b) konstruierte neue Ellipsoid E_{k+1} ist nach Satz (12.26) das volumenmäßig kleinste Ellipsoid, das E'_k enthält. Wegen $P' \subseteq E'_k$ gilt natürlich $P' \subseteq E_{k+1}$. Daraus folgt

$$P' \subseteq E_k\,,\ 0 \leq k \leq N.$$

Das Volumen des Anfangsellipsoids E_0 kann man wie folgt berechnen:

$$\mathrm{vol}(E_0) = \sqrt{\det(R^2 I)}\; V_n = R^n V_n,$$

wobei V_n das Volumen der Einheitskugel im $\mathbb{R}^n$ ist. Wir machen nun eine sehr grobe Abschätzung. Die Einheitskugel ist im Würfel $W = \{x \in \mathbb{R}^n \mid |x_i| \leq 1, i = 1,\dots,n\}$ enthalten. Das Volumen von W ist offensichtlich 2^n. Daraus folgt

$$\begin{aligned}\mathrm{vol}(E_0) < R^n 2^n &= 2^{n(2\langle A\rangle+\langle b\rangle-2n^2+\log\sqrt{n}+1)}\\ &< 2^{n(2\langle A\rangle+\langle b\rangle-n^2)}\,,\end{aligned}$$

In jedem Schritt der Ellipsoidmethode schrumpft nach (12.30) das Volumen um mindestens den Faktor $e^{\frac{-1}{2n}}$. Aus der Abschätzung von $\mathrm{vol}(E_0)$ und der in (1) von (12.33) angegebenen Formel für N erhalten wir somit

$$\mathrm{vol}(E_N) \leq e^{\frac{-N}{2n}}\,\mathrm{vol}(E_0) < 2^{-(n+1)(\langle A\rangle+\langle b\rangle-n^2)}.$$

Also gilt $\mathrm{vol}(E_N) < \mathrm{vol}(P')$ und $P' \subseteq E_N$. Dies ist ein Widerspruch. Hieraus folgt, dass P' und somit $\{x \mid Ax < b\}$ leer sind, wenn die Ellipsoidmethode in (2.a) abbricht. $\square$

Aus (12.31) (b) folgt nun unmittelbar

(12.34) Folgerung. *Ist $P = P(A,b) \subseteq \mathbb{R}^n$ ein Polyeder, von dem wir wissen, dass es entweder volldimensional oder leer ist, dann findet die Ellipsoidmethode entweder einen Punkt in P oder beweist, dass P leer ist.*

□

Nun kann man natürlich einem durch ein Ungleichungssystem gegebenen Polyeder nicht unmittelbar ansehen, ob es volldimensional ist oder nicht. Ferner können gerade diejenigen Polyeder, die durch Transformation linearer Programme entstehen (siehe (12.8), hier gilt immer $c^T x = b^T y$), nicht volldimensional sein, so dass die Ellipsoidmethode zu keiner befriedigenden Antwort führt. Diesen Defekt kann man jedoch durch die folgende Beobachtung reparieren.

(12.35) Satz. *Seien $A \in \mathbb{Q}^{(m,n)}$ und $b \in \mathbb{Q}^m$, dann hat das Ungleichungssystem*

$$Ax \leq b$$

hat genau dann eine Lösung, wenn das strikte Ungleichungssystem

$$Ax < b + 2^{-2\langle A\rangle - \langle b\rangle}\mathbb{1}$$

eine Lösung hat. Ferner kann man aus einer Lösung des strikten Ungleichungssystems in polynomialer Zeit eine Lösung von $Ax \leq b$ konstruieren.

□

Damit ist die Beschreibung der Ellipsoidmethode (bis auf die Abschätzung der Rechenzeit) vollständig. Wollen wir entscheiden, ob ein Polyeder $P(A, b)$ einen Punkt enthält, können wir die Methode (12.32) auf das Ungleichungssystem $Ax \leq b$ anwenden. Finden wir einen Punkt in $P(A, b)$, dann sind wir fertig. Andernfalls wissen wir, dass $P(A, b)$ nicht volldimensional ist. In diesem Falle können wir auf Satz (12.35) zurückgreifen und starten die Ellipsoidmethode neu, und zwar z. B. mit dem ganzzahligen Ungleichungssystem

$$2^{2\langle A\rangle + \langle b\rangle} Ax \leq 2^{2\langle A\rangle + \langle b\rangle} b + \mathbb{1}. \tag{12.36}$$

Entscheidet die Ellipsoidmethode, dass das zu (12.36) gehörige strikte Ungleichungssystem keine Lösung hat (Abbruch in Schritt (2.a)), so können wir aus (12.35) folgern, dass $P(A, b)$ leer ist. Andernfalls findet die Ellipsoidmethode einen Vektor x', der (12.36) erfüllt. Gilt $x' \in P(A, b)$, haben wir das gewünschte gefunden, falls nicht, kann man mit (einfachen Methoden der linearen Algebra) aus x' einen Punkt $x \in P(A, b)$ konstruieren, siehe (12.35).

Zur Lösung linearer Programme kann man die in Abschnitt 12.2 besprochenen Reduktionen benutzen. Entweder man fasst das lineare Programm (12.6) und das dazu duale (12.7) zu (12.8) zusammen und sucht wie oben angegeben im nicht volldimensionalen Polyeder (12.8) einen Punkt, oder man wendet $\overline{N}$-mal, siehe

(12.18), die Ellipsoidmethode für niederdimensionale Polyeder des Typs P_s aus (12.16) (2) im Rahmen eines binären Suchverfahrens (12.16) an.

In jedem Falle ist das Gesamtverfahren polynomial, wenn die Ellipsoidmethode (12.32) polynomial ist.

12.4 Laufzeit der Ellipsoidmethode

Wir wollen nun die Laufzeit der Ellipsoidmethode untersuchen und auf einige bisher verschwiegene Probleme bei der Ausführung von (12.32) aufmerksam machen. Offenbar ist die maximale Iterationszahl

$$N = 2n((3n+1)\langle A\rangle + (2n+1)\langle b\rangle - n^3)$$

polynomial in der Kodierungslänge von A und b. Also ist das Verfahren (12.32) genau dann polynomial, wenn jede Iteration in polynomialer Zeit ausgeführt werden kann.

Bei der Initialisierung (1) besteht kein Problem. Test (2.a) ist trivial, und die Schritte (2.b) und (2.c) führen wir dadurch aus, dass wir das Zentrum a_k in die Ungleichungen einsetzen und überprüfen, ob die Ungleichungen erfüllt sind oder nicht. Die Anzahl der hierzu benötigten elementaren Rechenschritte ist linear in $\langle A\rangle$ und $\langle b\rangle$. Sie ist also polynomial, wenn die Kodierungslänge des Vektors a_{k+1} polynomial ist.

An dieser Stelle beginnen die Schwierigkeiten. In der Update-Formel (3.a) muss eine Wurzel berechnet werden. I. a. werden also hier irrationale Zahlen auftreten, die natürlich nicht exakt berechnet werden können. Die (möglicherweise) irrationale Zahl $\sqrt{c^T A_k c}$ muss daher zu einer rationalen Zahl gerundet werden. Dadurch wird geometrisch bewirkt, dass der Mittelpunkt des Ellipsoids E_{k+1} ein wenig verschoben wird. Mit Sicherheit enthält das so verschobene Ellipsoid nicht mehr die Menge E'_k (siehe Satz (12.26)) und möglicherweise ist auch $P(A,b)$ nicht mehr in diesem Ellipsoid enthalten. Also bricht unser gesamter Beweis der Korrektheit des Verfahrens zusammen.

Ferner wird beim Update (3.b) durch möglicherweise große Zahlen geteilt, und es ist nicht a priori klar, dass die Kodierungslänge der Elemente von A_{k+1} bei wiederholter Anwendung von (3.b) polynomial in $\langle A\rangle + \langle b\rangle$ bleibt. Also müssen auch die Einträge in A_{k+1} gerundet werden. Dies kann zu folgenden Problemen führen. Die gerundete Matrix, sagen wir A^*_{k+1}, ist nicht mehr positiv definit und das Verfahren wird sinnlos. Oder A^*_{k+1} bleibt positiv definit, aber durch die Rundung hat sich die Form des zugehörigen Ellipsoids, sagen wir E^*_{k+1} so geändert, dass E^*_{k+1} das Polyeder $P(A, b)$ nicht mehr enthält.

Alle diese Klippen kann man mit einem einfachen Trick umschiffen, dessen Korrektheitsbeweis allerdings recht aufwendig ist. Die geometrische Idee hinter diesem Trick ist die folgende. Man nehme die binäre Darstellung der Komponenten des in (3.a) berechneten Vektors und der in (3.b) berechneten Matrix und runde nach p Stellen hinter dem Binärkomma. Dadurch ändert man die Lage des Mittelpunkts und die Form des Ellipsoids ein wenig. Nun bläst man das Ellipsoid ein bisschen auf, d. h. man multipliziert A_{k+1} mit einem Faktor $\zeta > 1$ und zwar so, dass die Menge E'_k beweisbar in dem aufgeblasenen Ellipsoid enthalten ist. Durch die Vergrößerung des Ellipsoids wird natürlich die in (12.30) bestimmte Schrumpfungsrate verschlechtert, was bedeutet, dass man insgesamt mehr Iterationen, sagen wir N', durchführen muss. Daraus folgt, dass der Rundungsparameter p und der Aufblasparameter ζ aufeinander so abgestimmt sein müssen, dass alle gerundeten und mit ζ multiplizierten Matrizen A_k, $1 \leq k \leq N'$ und alle gerundeten Mittelpunkte a_k, $1 \leq k \leq N'$ polynomial in $\langle A \rangle + \langle b \rangle$ berechnet werden können, so dass alle A_k positiv definit sind, $P \cap S(0, R)$ in allen Ellipsoiden E_k enthalten ist und die Iterationszahl N' ebenfalls polynomial in $\langle A \rangle + \langle b \rangle$ ist. Dies kann in der Tat durch Anwendung von Abschätzungstechniken aus der Analysis und linearen Algebra realisiert werden, siehe Grötschel, Lovász & Schrijver (1985).

Die Ellipsoidmethode (mit Rundungsmodifikation) ist also ein polynomialer Algorithmus, der entscheidet, ob ein Polyeder leer ist oder nicht. Mit Hilfe der im Vorhergehenden beschriebenen Reduktion kann sie dazu benutzt werden, lineare Programme in polynomialer Zeit zu lösen.

Wie sich der Leser denken kann, gibt es eine ganze Reihe von Varianten der Ellipsoidmethode, die dazu dienen sollen, schnellere Konvergenz zu erzwingen. Derartige Varianten sind z. B. in Bland, Goldfarb & Todd (1982), Grötschel, Lovász & Schrijver (1985) und M. Akgül, “Topics in Relaxation and Ellipsoidal Methods”, Pitman, Boston, 1984, (80 SK 870 A 315) beschrieben. Aus Platz- und Zeitgründen können wir hier nicht weiter daraus eingehen.

12.5 Ein Beispiel

Wir wollen hier den Ablauf der Ellipsoidmethode anhand eines Beispiels im $\mathbb{R}^2$ geometrisch veranschaulichen. Wir starten mit dem folgenden Polyeder $P(A, b) \subseteq$

$\mathbb{R}^2$ definiert durch

$$\begin{array}{rcr} -\ x_1 -\ \ x_2 & \le & -2 \\ 3x_1\qquad & \le & 4 \\ -2x_1 + 2x_2 & \le & 3 \end{array}$$

Dieses Polyeder P ist in der Kugel (Kreis) um den Nullpunkt mit Radius 7 enthalten. Diese Kugel soll unser Anfangsellipsoid $E_0 = E(A_0, 0)$ sein. Wir führen mit dieser Initialisierung die Schritte (2) und (3) des Ellipsoidverfahrens (12.32) durch. Wir rechnen natürlich nicht mit der eigentlich erforderlichen Genauigkeit sondern benutzen die vorhandene Maschinenpräzision. In unserem Fall ist das Programm in PASCAL geschrieben, und die Rechnungen werden in REAL-Arithmetik durchgeführt, d. h. wir benutzen ein Mantisse von 3 byte und einen Exponenten von 1 byte zur Zahlendarstellung. Das Verfahren führt insgesamt 7 Iterationen durch und endet mit einem zulässigen Punkt. In den nachfolgenden Abbildungen 12.3, . . . ,12.9 sind (im Maßstab 1:2) jeweils die Ellipsoide E_k und E_{k+1} mit ihren Mittelpunkten a_k und a_{k+1}, $k = 0, \ldots, 6$ aufgezeichnet. Man sieht also, wie sich die Form und Lage eines Ellipsoids in einem Iterationsschritt ändert. Das Startellipsoid ist gegeben durch

$$a_0^T = (0, 0), \quad A = \begin{pmatrix} 49 & 0 \\ 0 & 49 \end{pmatrix} .$$

Neben jeder Abbildung sind das neue Zentrum a_{k+1} und die neue Matrix A_{k+1} mit $E_{k+1} = E(A_{k+1}, a_{k+1})$ angegeben. Jede Abbildung enthält die Begrenzungsgeraden des Polytops P, so dass man auf einfache Weise sehen kann, welche Ungleichungen durch den neuen Mittelpunkt verletzt sind. Die Abbildung 12.10 enthält eine “Gesamtschau” des Verfahrens. Hier sind alle während des Verfahrens generierten Ellipsoide (im Maßstab 1:1) gleichzeitig dargestellt. Man sieht hier recht gut, wie die Mittelpunkte “herumhüpfen” und auf den ersten Blick keine “Ordnung” in der Folge der Mittelpunkte zu erkennen ist.

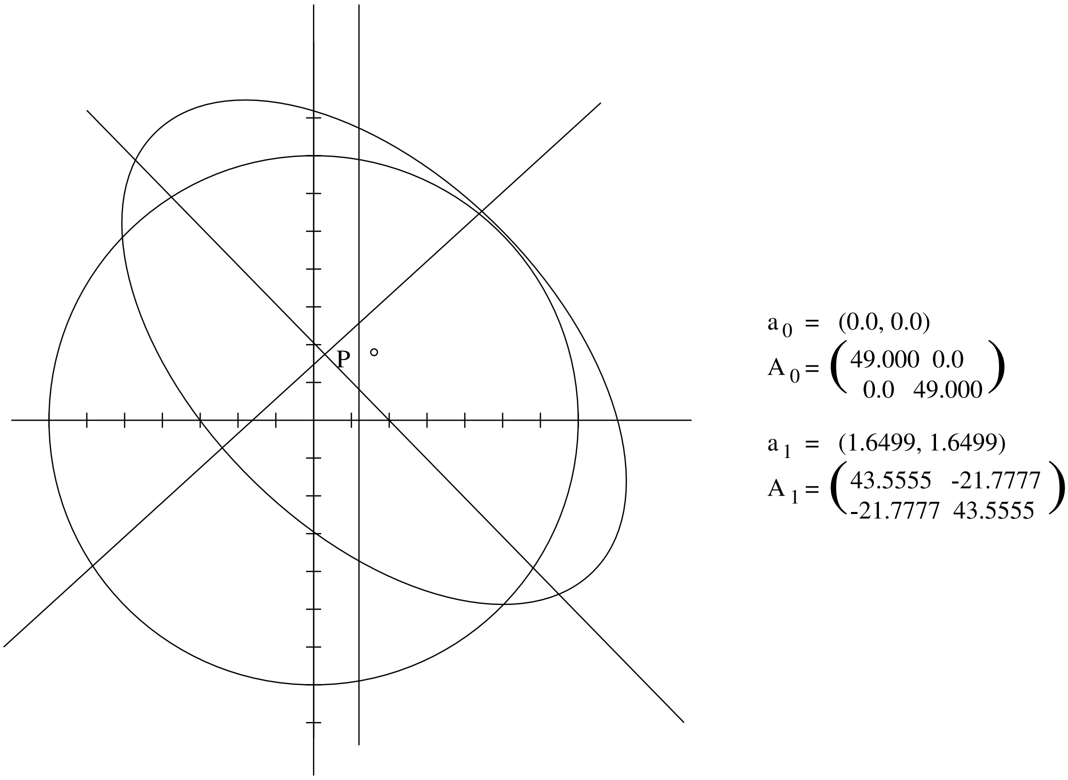
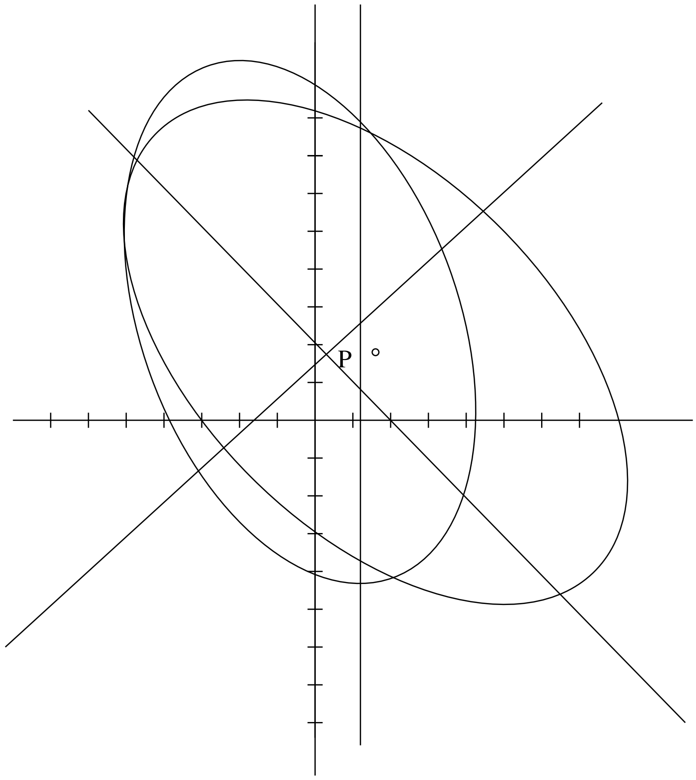

P
a 0 = (0.0, 0.0)
A 0 = (49.000 0.0 / 0.0 49.000)
a 1 = (1.6499, 1.6499)
A 1 = (43.5555 -21.7777 / -21.7777 43.5555)

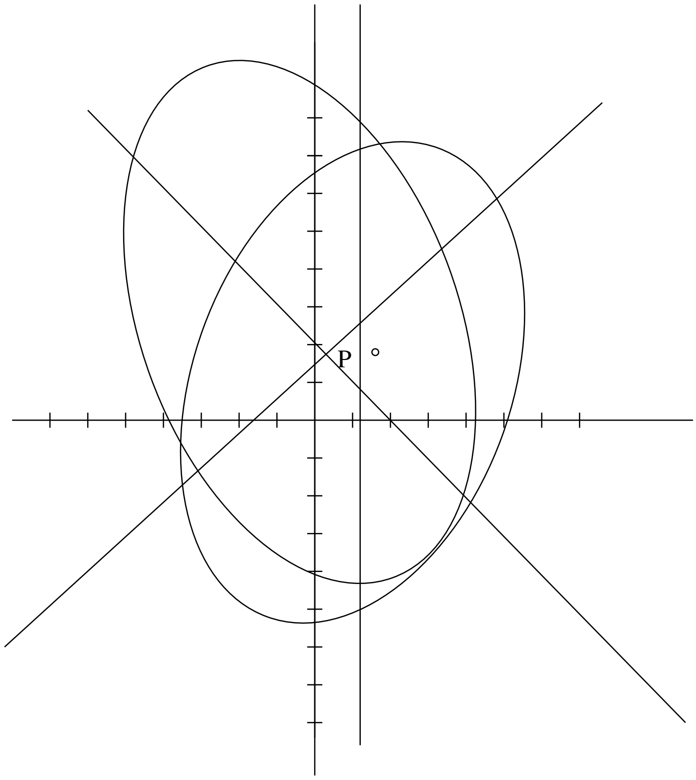
Abb. 12.3



$$a_2 = (-0.5499, 2.7499)$$

$$A_2 = \begin{pmatrix} 19.3580 & -9.6790 \\ -9.6790 & 48.3951 \end{pmatrix}$$

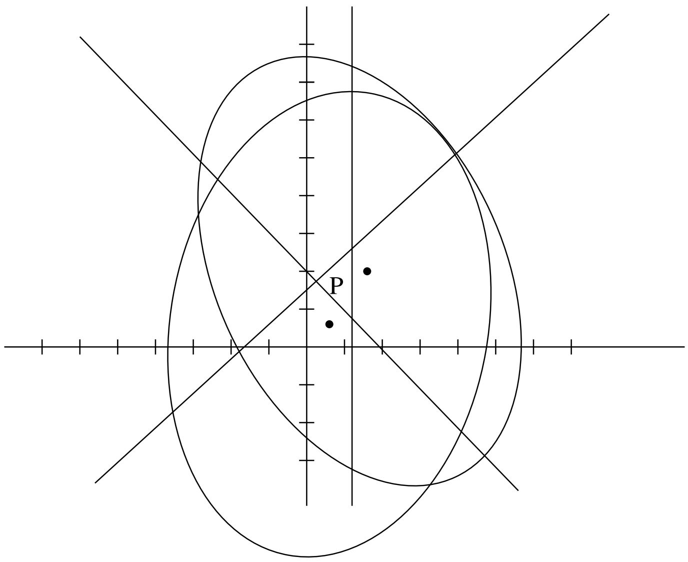
Abb. 12.4



$a_3 = (0.4871, 0.6758)$

$A_3 \begin{pmatrix} 17.2071 & 4.3018 \\ 4.3018 & 30.1125 \end{pmatrix}$

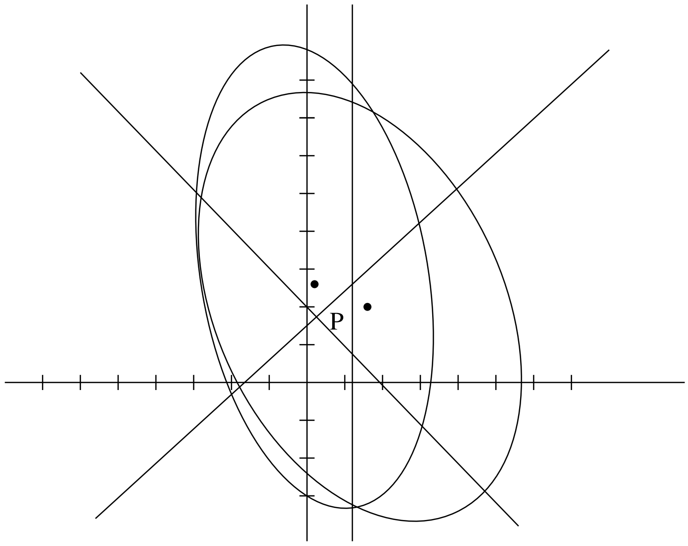
Abb. 12.5



$a_4 = (1.4458, 2.2098)$

$A_4 = \begin{pmatrix} 15.5894 & -6.0299 \\ -6.0299 & 21.351 \end{pmatrix}$

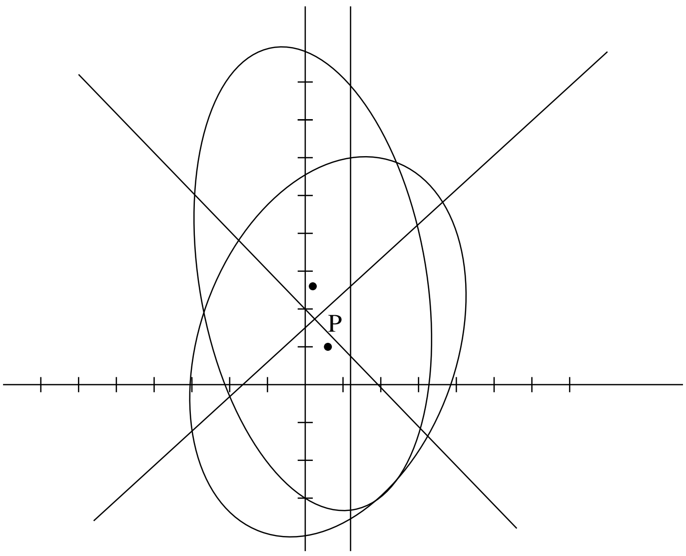
Abb. 12.6



$a_5 = (0.1297, 2.7188)$

$$A_5 = \begin{pmatrix} 6.9286 & -2.6799 \\ -2.6799 & 26.3603 \end{pmatrix}$$

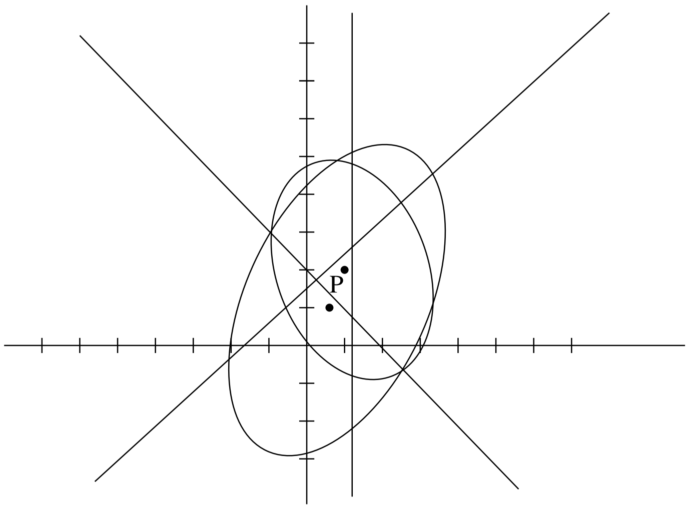
Abb. 12.7



$a_6 = (0.6449. 1\text{-}1618)$

$$A_6 = \begin{pmatrix} 7.1148 & 2.8443 \\ 2.8443 & 15.7511 \end{pmatrix}$$

Abb. 12.8



$a_7 = (1.2661, 2.317)$

$$A_7 = \begin{pmatrix} 6.3988 & -1.9726 \\ -1.9726 & 10.2372 \end{pmatrix}$$

Abb. 12.9

Kapitel 13

Der Karmarkar-Algorithmus

Im Jahre 1984 hat N. Karmarkar (AT&T Bell Laboratories) einen neuen Algorithmus zur Lösung linearer Programme entwickelt. Karmarkar's Aufsatz "A New Polynomial Time Algorithm for Linear Programming" wird in der Zeitschrift Combinatorica veröffentlicht, ist zur Zeit (d. h. im Februar 1985) noch nicht erschienen, zirkuliert aber als Preprint. Im Herbst 1984 hat Karmarkar auf Vorträgen mitgeteilt, dass sein Verfahren nicht nur theoretisch polynomial ist (wie auch die in Kapitel 12 beschriebene Ellipsoidmethode), sondern in der Praxis erheblich rascher als das Simplexverfahren arbeitet. Karmarkar behauptet, dass eine Implementation seines Algorithmus etwa 50 mal schneller ist als MPSX, ein von der Firma IBM angebotenes und auf dem Simplexalgorithmus basierendes Softwarepaket zur Lösung linearer Programme, das als eines der besten LP-Pakete gilt. Die Implementation von AT&T Bell Laboratories soll allerdings auf einigen Modifikationen beruhen, die nicht in Karmarkar's Preprint enthalten sind.

Da bei großen Industriefirmen sehr große LP's täglich in nicht unerheblicher Zahl gelöst werden, ist die Ankündigung Karmarkars auf sehr großes Interesse bei Praktikern gestoßen. Ja, sein Verfahren hat sogar Aufsehen in der Presse verursacht. Zum Beispiel haben New York Times, Science, Time, Der Spiegel (falsch — wie bei Artikeln über Mathematik üblich), Die Zeit, Süddeutsche Zeitung (sehr ordentlich) über Karmarkars Verfahren berichtet, wie immer natürlich unter dem Hauptaspekt, dass dadurch Millionen von Dollars gespart werden können.

Im nachfolgenden wollen wir die Grundidee des Algorithmus von Karmarkar darstellen. A priori ist mir nicht klar, "warum" dieses Verfahren in der Praxis schnell sein soll. Ich selbst verfüge über keinerlei Rechenerfahrung mit diesem Algorithmus, und ich weiß zur Zeit auch nicht, welche Modifikationen bei der Implementation von AT&T Bell Laboratories gemacht wurden. Jedoch sind mir einige der Wissenschaftler, die die Aussagen über die Effizienz des Verfahrens gemacht haben, bekannt, und ich messe ihnen einige Bedeutung bei. Dennoch bedarf die Methode noch weiterer Überprüfungen bevor ein "endgültiges" Urteil abgegeben werden kann. Unabhängig davon ist jedoch die geometrische Idee, die hinter der Methode steckt, sehr interessant.

13.1 13.1 Reduktionen

Wie der Simplexalgorithmus und die Ellipsoidmethode, so ist auch der Karmarkar-Algorithmus auf einen speziellen Problemtyp zugeschneidert und nicht auf allgemeine lineare Programme anwendbar. Wie üblich müssen wir daher zunächst zeigen, dass ein allgemeines LP so transformiert werden kann, dass es mit Karmarkars Algorithmus gelöst werden kann. Die Grundversion des Karmarkar-Algorithmus (und nur die wollen wir hier beschreiben) ist ein Verfahren zur Entscheidung, ob ein Polyeder einer recht speziellen Form einen Punkt enthält oder nicht. Das Problem, das Karmarkar betrachtet ist das folgende.

(13.1) Problem. *Gegeben seien eine Matrix $A \in \mathbb{Q}^{(m,n)}$ und ein Vektor $c \in \mathbb{Q}^n$, entscheide, ob das System*

$$Ax = 0, \quad \mathbb{1}^T x = 1, \quad x \geq 0, \quad c^T x \leq 0$$

eine Lösung hat, und falls das so ist, finde eine.

□

Um den Karmarkar-Algorithmus starten zu können, ist die Kenntnis eines zulässigen Startvektors im relativ Inneren von $\{x \mid Ax = 0, \mathbb{1}^T x = 1, x \geq 0\}$ notwendig. Wir wollen sogar noch mehr fordern, und zwar soll gelten

(13.2) $$\overline{x} := \tfrac{1}{n}\mathbb{1} \quad \text{erfüllt} \quad A\overline{x} = 0.$$

Trivialerweise erfüllt $\overline{x}$ die Bedingungen $\mathbb{1}^T\overline{x} = 1$, $\overline{x} \geq 0$. Gilt $c^T\overline{x} \leq 0$, ist man natürlich fertig. Die eigentliche Aufgabe besteht also darin, aus $\overline{x}$ einen Punkt zu konstruieren, der alle Bedingungen von (13.1) erfüllt.

Wir werden nun, wie bei der Darstellung der Ellipsoidmethode, ein allgemeines LP in ein (bzw. mehrere) Probleme des Typs (13.1) umformen.

Wir wissen bereits, dass jedes LP in ein LP in Standardform (9.1) transformiert werden kann. Gegeben sei also das folgende Problem

$$(13.3)\qquad \begin{array}{l} \max w^T x \\ Bx = b \\ x \geq 0 \end{array}$$

mit $w \in \mathbb{Q}^n$, $B \in \mathbb{Q}^{(m,n)}$, $b \in \mathbb{Q}^m$. Wie in Abschnitt 12.2 gezeigt gibt es zwei prinzipielle Reduktionsmöglichkeiten. Man fasst (13.3) und das dazu duale Problem wie in (12.8) zusammen, oder man führt die in (12.16) beschriebene Binärsuche durch. Wir stellen hier kurz die Reduktion über Binärsuche dar. Daraus ergibt sich auch, wie man die Kombination des primalen mit dem dualen Problem behandeln kann.

Genau wie in (12.16) angegeben (und mit den in den davor angegebenen Sätzen vorgelegten Begründungen) führen wir eine binäre Suche durch über der Menge

$$S := \{\tfrac{p}{q} \in Q \mid |p| \leq n2^{\langle B\rangle+\langle b\rangle+2\langle w\rangle-n^2-n}, 1 \leq q \leq 2^{\langle B\rangle+\langle w\rangle-n^2-n}\}$$

der möglichen optimalen Zielfunktionswerte (falls (13.3) eine endliche Optimallösung hat). Zur Bestimmung einer optimalen Lösung von (13.3) sind nach (12.18) höchstens

$$\overline{N} := 2\langle B\rangle + \langle b\rangle + 3\langle c\rangle - 2n^2 - 2n + \log_2 n + 2$$

Aufrufe eines Algorithmus nötig, der für ein $s \in S$ entscheidet, ob

$$(13.4)\qquad \begin{array}{ll} w^T x & \leq s \\ Bx & = b \\ x & \geq 0 \end{array}$$

eine Lösung hat. Genau dann, wenn wir zeigen können, dass (13.4) in polynomialer Zeit gelöst werden kann, haben wir also ein polynomiales Verfahren zur Lösung von (13.3). Wir wissen aus Satz (12.11), dass wir alle Variablen durch $2^{2\langle B\rangle+\langle b\rangle-2n^2}$ beschränken können. Führen wir für die Ungleichung in (13.4) eine Schlupfvariable x_{n+1} ein, so gilt $0 \leq x_{n+1} \leq |s| \leq n2^{\langle B\rangle+\langle b\rangle+2\langle w\rangle-n^2-n}$. Setzen wir also

$$M := n\big(2^{2\langle B\rangle+\langle b\rangle-2n^2} + 2^{\langle B\rangle+\langle b\rangle+2\langle w\rangle-n^2-n}\big),$$

so ist (13.4) genau dann lösbar, wenn

$$(13.5)\qquad \begin{array}{ll} \begin{pmatrix} w^T & 1 \\ B & 0 \end{pmatrix} \begin{pmatrix} x_1 \\ \vdots \\ x_{n+1} \end{pmatrix} & = \begin{pmatrix} s \\ b \end{pmatrix} \\ \sum_{i=1}^{n+1} x_i & \leq M \\ x_i & \geq 0,\ i = 1, \ldots, n+1 \end{array}$$

lösbar ist. Führen wir eine weitere Schlupfvariable x_{n+2} ein (d. h. es gilt nun $x \in \mathbb{R}^{n+2}$) und skalieren wir die Variablen um, so ist (13.5) genau dann lösbar, wenn

$$\text{(13.6)}\qquad \begin{aligned} &Cx := \begin{pmatrix} w^T & 1 & 0 \\ B & 0 & 0 \end{pmatrix} x &&= \tfrac{1}{M}\begin{pmatrix} s \\ b \end{pmatrix} =: \begin{pmatrix} \overline{c}_0 \\ \vdots \\ \overline{c}_m \end{pmatrix} \\ &\mathbb{1}^T x &&= 1 \\ &x_i &&\geq 0,\ i = 1, \dots, n+2 \end{aligned}$$

lösbar ist. Von der i-ten Zeile der Matrix $C (i = 0, \dots, m)$ ziehen wir nun das $\overline{c}_i$-fache der letzten Gleichung $\mathbb{1}^T x = 1$ ab. Dadurch machen wir die ersten $m + 1$ Gleichungen homogen und erreichen, dass aus dem Ungleichungs- und Gleichungssystem (13.6) ein äquivalentes System der folgenden Form wird:

$$\text{(13.7)}\qquad \begin{aligned} Dx &= 0 \\ \mathbb{1}^T x &= 1 \\ x &\geq 0\,. \end{aligned}$$

Um die Voraussetzung (13.2) herzustellen, führen wir eine weitere Schlupfvariable x_{n+3} wie folgt ein. Wir betrachten

$$\text{(13.8)}\qquad \begin{aligned} Dx - D\mathbb{1}x_{n+3} &= 0 \\ \mathbb{1}^T x + x_{n+3} &= 1 \\ x_i &\geq 0\ , i = 1, \dots, n+3\,. \end{aligned}$$

Offenbar hat (13.7) genau dann eine Lösung, wenn (13.8) eine Lösung mit $x_{n+3} = 0$ hat. Setzen wir

$$A := (D, -D\mathbb{1}) \in \mathbb{Q}^{(m+1,n+3)}, \quad c := (0, , \dots, 0, 1)^T \in \mathbb{Q}^{n+3}$$

und betrachten wir x als Vektor im $\mathbb{R}^{n+3}$, so hat also (13.7) genau dann eine Lösung, wenn

$$\text{(13.9)}\qquad Ax = 0, \quad \mathbb{1}^T x = 1, \quad x \geq 0, \quad c^T x \leq 0$$

eine Lösung hat. Ferner erfüllt nach Konstruktion der Vektor

$$\overline{x}^T := (\tfrac{1}{n+3}, \dots, \tfrac{1}{n+3}) \in \mathbb{Q}^{n+3}$$

bezüglich des Systems (13.9) die Forderung (13.2).

Folglich kann jedes lineare Program durch polynomial viele Aufrufe eines Algorithmus zur Lösung von (13.1) (mit Zusatzvoraussetzung (13.2)) gelöst werden.

13.2 Die Grundversion des Karmarkar-Algorithmus

Wir wollen nun das Karmarkar-Verfahren zur Lösung von (13.1) unter der Zusatzvoraussetzung (13.2) beschreiben. Wie wir am Ende des letzten Abschnittes gesehen haben, können wir jedes LP in eine Folge von Problemen der Form (13.1) transformieren, und wir können sogar das Zentrum des Simplex $\{x \in \mathbb{R}^n \mid \mathbb{1}^T x = 1, x \geq 0\}$ als Startpunkt wählen. Im weiteren betrachten wir also das folgende

(13.10) Problem. *Gegeben seien eine Matrix $A \in \mathbb{Q}^{(m,n)}$ und ein Vektor $c \in \mathbb{Q}^n$. Es seien*

$$\begin{array}{ll} E := \{x \in \mathbb{R}^n \mid \mathbb{1}^T x = 1\} & \textit{(Hyperebene)} \\ \Sigma := E \cap \mathbb{R}^n_+ & \textit{(Simplex)} \\ \Omega := \{x \in \mathbb{R}^n \mid Ax = 0\} & \textit{(linearer Raum).} \end{array}$$

Ferner wird vorausgesetzt, dass der Vektor $\overline{x} := \frac{1}{n}\mathbb{1}$ in $\Sigma \cap \Omega$ enthalten ist. Gesucht ist ein Vektor $x \in \mathbb{Q}^n$ mit

$$x \in P := \Sigma \cap \Omega\,, \quad c^T x \leq 0.$$

□

Problem (13.10) kann sicherlich dadurch gelöst werden, daß eine Optimallösung des folgenden Programms bestimmt wird.

$$(13.11) \qquad \begin{array}{l} \min c^T x \\ Ax = 0 \\ \mathbb{1}^T x = 1 \\ x \geq 0 \end{array} \quad \text{bzw.} \quad \begin{array}{ll} \min c^T x & \\ x \in \Omega & \cap E \cap \mathbb{R}^n_+ \end{array}$$

Offenbar ist der optimale Zielfunktionswert von (13.11) genau dann größer als Null, wenn (13.10) keine Lösung hat. Unser Ziel wird nun sein (13.11) statt (13.10) zu lösen, wobei wir voraussetzen, dass $\frac{1}{n}\mathbb{1}$ für (13.11) zulässig ist.

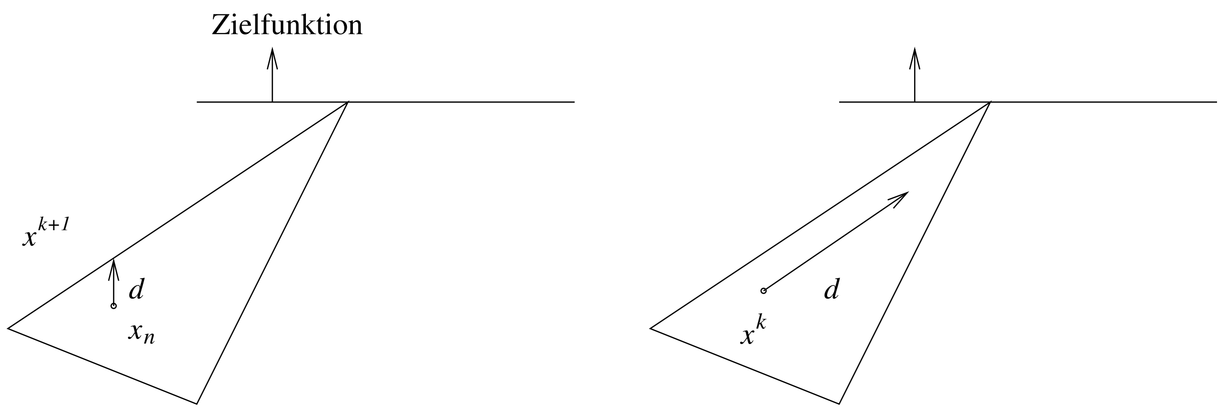
Es erscheint natürlich reichlich umständlich, ein lineares Programm, sagen wir der Form (13.3), durch Binärsuche auf eine Folge von Zulässigkeitsproblemen (13.10) zu reduzieren, die dann wieder durch spezielle lineare Programme der Form (13.11) gelöst werden. Diesen Umweg kann man durch die sogenannte Sliding Objective Function Technique begradigen. Die dabei notwendigen zusätzlichen Überlegungen tragen jedoch nicht zu einem besseren Verständnis der Grundversion des Verfahrens bei, um die es hier geht. Ziel dieses Kapitels ist nicht die Darstellung einer besonders effizienten Version des Karmarkar-Algorithmus sondern dessen prinzipielle Idee.

Geometrische Beschreibung eines Iterationsschrittes

Zur Lösung von (13.11) hat Karmarkar ein Verfahren entworfen, das wie fast alle Optimierungsverfahren (insbesondere die Methoden der nichtlinearen Optimierung, die wir in der Vorlesung Optimierung II kennenlernen werden) auf der folgenden simplen Idee beruht. Angenommen man befinde sich an einem zulässigen Punkt, sagen wir x^k, dann sucht man eine Richtung (also einen Vektor $d \in \mathbb{R}^n$), bezüglich der die Zielfunktion verbessert werden kann. Daraufhin bestimmt man eine Schrittlänge ρ, so dass man von x^k zum nächsten Punkt $x^{k+1} := x^k + \rho d$ gelangt. Der nächste Punkt x^{k+1} soll natürlich auch zulässig sein und einen "wesentlich" besseren Zielfunktionswert haben.

Die Essenz eines jeden solchen Verfahrens steckt natürlich in der Wahl der Richtung und der Schrittlänge. Bei derartigen Verfahren tritt häufig die folgende Situation ein. Man ist in der Lage, eine sehr gute Richtung zu bestimmen (d. h. die Zielfunktion wird in Richtung d stark verbessert), aber man kann in Richtung d nur einen sehr kleinen Schritt ausführen, wenn man die zulässige Menge nicht verlassen will. Trotz guter Richtung kommt man also im Bezug auf eine tatsächliche Verbesserung kaum vorwärts und erhält u. U. global schlechtes Konvergenzverhalten. Man muss sich also bemühen, einen guten Kompromiss zwischen "Qualität der Richtung" und "mögliche Schrittlänge" zu finden, um insgesamt gute Fortschritte zu machen.

Ist man — wie im vorliegenden Fall — im relativen Inneren der Menge P, aber nah am Rand und geht man z. B. in Richtung des Normalenvektors der Zielfunktion, so kann man sehr schnell an den Rand von P gelangen, ohne wirklich weiter gekommen zu sein, siehe Abbildung 13.1.



gute Richtung, geringer Fortschritt

schlechtere Richtung, aber großer Fortschritt

Abb. 13.1

Karmarkars Idee zur Lösung bzw. Umgehung dieser Schwierigkeit ist die folgende. Er führt eine projektive Transformation aus, die den Simplex Σ auf sich selbst, den affinen Teilraum Ω auf einen anderen affinen Teilraum Ω' abbildet und den relativ inneren Punkt x^k auf das Zentrum $\frac{1}{n}\mathbb{1}$ von Σ wirft. P wird dabei auf ein neues Polyeder P_k abgebildet. Offenbar kann man von $\frac{1}{n}\mathbb{1}$ aus recht große Schritte in alle zulässigen Richtungen machen, ohne sofort den zulässigen Bereich P_k zu verlassen. Man bestimmt so durch Festlegung einer Richtung und einer Schrittlänge von $\frac{1}{n}\mathbb{1}$ ausgehend einen Punkt $y^{k+1} \in P_k$ und transformiert diesen zurück, um den nächsten (zulässigen) Iterationspunkt $x^{k+1} \in P$ zu erhalten.

Zur Bestimmung der Richtung macht Karmarkar folgende Überlegung. Ideal wäre es natürlich, direkt das Optimum des transformierten Problems zu bestimmen. Aber die lineare Zielfunktion des ursprünglichen Problems wird durch die projektive Transformation in eine nichtlineare Abbildung übergeführt, so dass dies nicht so einfach zu bewerkstelligen ist. Dennoch kann man diese transformierte Zielfunktion auf natürliche Weise linearisieren. Mit $\overline{c}^T x$ sei die neue (linearisierte) Zielfunktion bezeichnet. Man hat nun ein neues Problem: Es soll eine lineare Zielfunktion über dem Durchschnitt eines Simplex mit einem affinen Raum optimiert werden, wir erhalten

$$(13.12) \qquad \begin{array}{l} \min \overline{c}^T x \\ x \in \Omega' \quad \cap E \cap \mathbb{R}^n_+ \, . \end{array}$$

Dieses Problem ist offenbar wiederum vom Typ unseres Ausgangsproblems (13.11). Aber diese neue Aufgabe ist ja nur ein Hilfsproblem! Möglicherweise ist daher eine gute Approximation der Optimallösung von (13.12) ausreichend für das, was wir bezwecken. Durch die Lösung einer Vereinfachung dieses Problems (13.12) könnten u. U. ein Richtungsvektor und eine Schrittlänge gefunden werden, die von hinreichend guter Qualität bezüglich globaler Konvergenz des Verfahrens sind. Die Vereinfachung, die wir betrachten wollen, geschieht dadurch, dass die bei der Druchschnittsbildung beteiligte Menge $\mathbb{R}^n_+$ durch die größte Kugel, sagen wir K, mit Zentrum $\frac{1}{n}\mathbb{1}$, die im Simplex Σ enthalten ist ersetzt wird. Statt (13.12) wird also die Aufgabe

$$(13.13) \qquad \begin{array}{l} \min \overline{c}^T x \\ x \in \Omega' \quad \cap E \cap K \end{array}$$

betrachtet. Man optimiere eine lineare Zielfunktion über dem Durchschnitt einer Kugel K mit einem affinen Raum $\Omega' \cap E$. Dieses Problem ist ohne Einschaltung eines Algorithmus durch eine explizite Formel lösbar. Durch die Optimallösung von (13.13) werden die gesuchte Richtung und die Schrittlänge explizit geliefert.

Eine Modifikation ist jedoch noch nötig. Das Optimum über $\Omega' \cap E \cap K$ ist i. a. irrational und muss gerundet werden. Dadurch ist es möglich, dass y^{k+1} nicht mehr im relativen Inneren von P_k bzw. der nächste (gerundete) Iterationspunkt x^{k+1} nicht mehr im relativ Inneren von P liegt. Statt K wählt man daher eine kleinere Kugel K' mit dem gleichen Zentrum, so dass auch nach Rundung und Rücktransformation garantiert ist, dass der nächste Iterationspunkt ein relativ innerer Punkt ist.

Analytische Beschreibung eines Iterationsschrittes

Die oben gegebene anschauliche Darstellung wollen wir nun explizit vorführen. Wir starten also mit Problem (13.11). Wir setzen (wie beim Simplexverfahren) voraus, dass A vollen Zeilenrang hat. Außerdem kennen wir mit x^k einen relativ inneren Punkt von P. Ist $D = \operatorname{diag}(x^k)$ die (n,n)-Diagonalmatrix, die die Komponenten $x_1^k, \dots, x_n^k$ von x^k auf der Hauptdiagonalen enthält, so ist durch

(13.14)
$$T_k(x) := \frac{1}{\mathbb{1}^T D^{-1} x} D^{-1} x$$

eine projektive Transformation definiert. ($T_k(x)$ ist natürlich nur dann endlich, wenn $x \notin \overline{H} = \{y \in \mathbb{R}^n \mid \mathbb{1}^T D^{-1} y = 0\}$ gilt. Wir werden beim Rechnen mit T_k diese Einschränkung nicht weiter erwähnen und gehen davon aus, dass klar ist, wie die Formeln, die T_k enthalten, zu interpretieren sind.) Die projektive Transformation T_k hat, wie leicht zu sehen ist, folgende Eingeschaften:

(13.15) Eigenschaften von T_k.

(a) $x \geq 0 \implies T_k(x) \geq 0$.

(b) $\mathbb{1}^T x = 1 \implies \mathbb{1}^T T_k(x) = 1$.

(c) $T_k^{-1}(y) = \frac{1}{\mathbb{1}^T D y} D y$ *für alle* $y \in \Sigma$.

(d) $T_k(\Sigma) = \Sigma$.

(e) $P_k := T_k(P) = T_k(\Sigma \cap \Omega) = \Sigma \cap \Omega_k$ *wobei* $\Omega_k = \{y \in R^n \mid ADy = 0\}$.

(f) $T_k(x^k) = \frac{1}{\mathbb{1}^T \mathbb{1}} D^{-1} x^k = \frac{1}{n} \mathbb{1} \in P_k$.

□

Die Transformation T_k ist also eine bijektive Abbildung $P \longrightarrow P_k$ und $\Sigma \longrightarrow \Sigma$, die den Punkt x^k in das Zentrum $\frac{1}{n}\mathbb{1}$ von Σ abbildet. Aus (13.15) folgt

$$(13.16)\qquad \begin{array}{r} \min c^T x \\ Ax = 0 \\ \mathbb{1}^T x = 1 \\ x \geq 0 \end{array} \;=\; \begin{array}{c} \min c^T x \\ x \in P \\ \\ \\ \end{array} \;=\; \begin{array}{c} \min c^T T_k^{-1}(y) \\ y \in P_k = T_k(P) \\ \\ \\ \end{array} \;=\; \begin{array}{r} \min \frac{1}{\mathbb{1}^T D y} c^T D y \\ A D y = 0 \\ \mathbb{1}^T y = 1 \\ y \geq 0 \end{array}$$

Das letzte Programm in (13.16) ist das in der geometrisch-anschaulichen Erläuterung weiter oben genannte Pogramm mit nichtlinearer Zielfunktion $\frac{1}{\mathbb{1}^T D y} D y$. Wir linearisieren diese durch Weglassen des Nenners wie folgt:

$$(13.17)\qquad \overline{c}^T := c^T D$$

und erhalten das lineare (Hilfs)-Programm

$$(13.18)\qquad \begin{array}{l} \min \overline{c}^T y \\ A D y = 0 \\ \mathbb{1}^T y = 1 \\ y \geq 0 \end{array}$$

Wir vereinfachen (13.18) dadurch, dass wir die Nichtnegativitätsbedingung in (13.18) durch die Bedingung ersetzen, dass y in einer Kugel um $\frac{1}{n}\mathbb{1}$ liegt. Die größte Kugel mit diesem Zentrum, die man dem Simplex Σ einbeschreiben kann, hat den Radius $\frac{1}{\sqrt{n(n-1)}}$. Wie bereits erwähnt, müssen wir aus technischen Gründen eine kleinere Kugel wählen. Es wird sich zeigen, dass man mit der Hälfte dieses optimalen Radius auskommt. Wir betrachten also das Programm

$$(13.19)\qquad \begin{array}{ll} \min \overline{c}^T y & \\ A D y & = 0 \\ \mathbb{1}^T y & = 1 \\ \|y - \frac{1}{n}\mathbb{1}\| & \leq \frac{1}{2}\frac{1}{\sqrt{n(n-1)}} \end{array}$$

Das Minimum von (13.19) kann explizit bestimmt werden.

(13.20) Lemma. *Die Optimallösung von (13.19) ist der Vektor*

$$y^{k+1} := \frac{1}{n}\mathbb{1} - \frac{1}{2}\frac{1}{\sqrt{n(n-1)}} \frac{(I - DA^T(AD^2A^T)^{-1}AD - \frac{1}{n}\mathbb{1}\mathbb{1}^T)Dc}{\|(I - DA^T(AD^2A^T)^{-1}AD - \frac{1}{n}\mathbb{1}\mathbb{1}^T)Dc\|}.$$

Beweis : Zunächst projizieren wir $\overline{c}$ orthogonal auf den linearen Raum $L := \{x \in \mathbb{R}^n \mid ADx = 0, \mathbb{1}^T x = 0\}$. Setzen wir

$$B = \begin{pmatrix} AD \\ \mathbb{1}^T \end{pmatrix},$$

so ist die Projektion von $\overline{c}$ auf L gegeben durch

$$\overline{\overline{c}} = (I - B^T(BB^T)^{-1}B)\overline{c},$$

denn offenbar gilt $B\overline{\overline{c}} = 0$, $(\overline{c} - \overline{\overline{c}})^T\overline{\overline{c}} = 0$. Beachten wir, dass $AD\mathbb{1} = Ax^k = 0$ gilt, so folgt durch einfaches Ausrechnen

$$\begin{aligned}
BB^T &= \begin{pmatrix} AD^2A^T & AD\mathbb{1} \\ (AD\mathbb{1})^T & \mathbb{1}^T\mathbb{1} \end{pmatrix} = \begin{pmatrix} AD^2A^T & 0 \\ 0 & n \end{pmatrix} \\
(BB^T)^{-1} &= \begin{pmatrix} (AD^2A^T)^{-1} & 0 \\ 0 & \frac{1}{n} \end{pmatrix} \\
B^T(BB^T)^{-1}B &= DA^T(AD^2A^T)^{-1}AD + \tfrac{1}{n}\mathbb{1}\mathbb{1}^T,
\end{aligned}$$

und somit

$$\overline{\overline{c}} = (I - DA^T(AD^2A^T)^{-1}AD - \tfrac{1}{n}\mathbb{1}\mathbb{1}^T)\overline{c}.$$

Für $y \in L$ gilt nach Konstruktion $\overline{c}^T y = \overline{\overline{c}}^T y$ und somit ist das Minimum von (13.19) gleich dem Minimum der Funktion $\overline{\overline{c}}^T y$ unter den Nebenbedingungen von (13.19). Aufgrund unserer Konstruktion ist dieses Minimum gleich dem Minimum von

$$\begin{aligned}
&\min \overline{\overline{c}}^T y \\
&\|y - \tfrac{1}{n}\mathbb{1}\| \leq \tfrac{1}{2}\tfrac{1}{\sqrt{n(n-1)}},
\end{aligned}$$

also dem Minimum einer linearen Zielfunktion über einer Kugel. Offenbar wird das Minimum hier durch den Vektor angenommen, den man durch einen Schritt vom Mittelpunkt $\frac{1}{n}\mathbb{1}$ aus in Richtung $-\overline{\overline{c}}$ mit der Länge des Kugelradius erhält, also durch

$$y^{k+1} = \frac{1}{n}\mathbb{1} - \frac{1}{2}\frac{1}{\sqrt{n(n-1)}}\frac{1}{\|\overline{\overline{c}}\|}\overline{\overline{c}}.$$

Hieraus folgt die Behauptung. $\square$

Damit haben wir ein Minimum von (13.19) analytisch bestimmten können und unser vereinfachtes Hilfsproblem gelöst. Den nächsten Iterationspunkt erhält man durch Anwendung der inversen projektiven Transformation T_k^{-1} auf den in (13.20) bestimmten Vektor y^{k+1}:

(13.21) $$x^{k+1} := T_k^{-1}(y^{k+1}) = \frac{1}{\mathbb{1}^T D y^{k+1}} D y^{k+1}.$$

Dem Hilfsprogramm (13.19) kann man übrigens noch eine andere geometrische Interpretation geben. Setzen wir

$$z := Dy \quad \text{bzw.} \quad y := D^{-1}z$$

so kann man (13.19) wie folgt schreiben

$$\text{(13.22)} \qquad \begin{array}{ll} \min c^T z & \\ Az & = 0 \\ \mathbb{1}^T D^{-1} z & = 1 \\ z & \in E(\frac{1}{4n(n-1)} D^2, \frac{1}{n} x^k) \end{array}$$

Denn wegen (13.17) gilt $\overline{c}^T y = c^T z$ und aus $D^{-1}x^k = \mathbb{1}$ folgt:

$$\begin{array}{rcl} \|y - \frac{1}{n}\mathbb{1}\| \le \frac{1}{2}\frac{1}{\sqrt{n(n-1)}} & \Longleftrightarrow & \|D^{-1}z - \frac{1}{n}D^{-1}x^k\| \le \frac{1}{2}\frac{1}{\sqrt{n(n-1)}} \\ & \Longleftrightarrow & \|D^{-1}(z - \frac{1}{n}x^k)\| \le \frac{1}{2}\frac{1}{\sqrt{n(n-1)}} \\ & \Longleftrightarrow & (z - \frac{1}{n}x^k)^T D^{-2}(z - \frac{1}{n}x^k) \le \frac{1}{4n(n-1)} \\ & \Longleftrightarrow & (z - \frac{1}{n}x^k)(4n(n-1))D^{-2}(z - \frac{1}{n}x^k) \le 1 \\ & \overset{(12.20)}{\Longleftrightarrow} & z \in E(\frac{1}{4n(n-1)}D^2, \frac{1}{n}x^k) \end{array}$$

Gehen wir vom Ursprungsproblem (13.11) aus, so bedeutet dies also, dass wir in unserem Hilfsprogramm die Originalzielfunktion c und den linearen Raum Ω unverändert lassen. Wir ersetzen die Hyperebene E durch $E_k = \{z \mid \mathbb{1}^T D^{-1} z = 1\}$ und die Nichtnegativitätsbedingung $x \in \mathbb{R}^n_+$ durch das Ellipsoid $E(\frac{1}{4n(n-1)}D^2, \frac{1}{n}x^k)$ und optimieren hierüber. Aus einer Optimallösung, sagen wir z^{k+1}, von (13.22) erhalten wir eine Optimallösung von (13.19) durch

$$y^{k+1} = D^{-1}z^{k+1}.$$

Wie im Beweis von (13.20) kann man zeigen, dass (13.22) durch eine direkte Formel gelöst werden kann (das folgt natürlich auch aus (13.20) durch die obige Gleichung), und zwar gilt:

(13.23)

$$z^{k+1} = \frac{1}{n}x^k - \frac{1}{2}\frac{1}{\sqrt{n(n-1)}}\frac{D(I - DA^T(AD^2A^T)^{-1}AD - \frac{1}{n}\mathbb{1}\mathbb{1}^T)Dc}{\|(I - DA^T(AD^2A^T)^{-1}AD - \frac{1}{n}\mathbb{1}\mathbb{1}^T)Dc\|}.$$

Hieraus ergibt sich über (13.21) eine neue (einfache) Formel zur Berechnung des nächsten Iterationspunktes.

$$\text{(13.24)} \qquad x^{k+1} := \frac{1}{\mathbb{1}^T D y^{k+1}} D y^{k+1} = \frac{1}{\mathbb{1}^T z^{k+1}} z^{k+1}.$$

Damit haben wir die wesentlichen Bestandteile eines Iterationsschrittes des Karmarkar-Algorithmus beschrieben und können ihn zusammenfassend darstellen, wobei noch das Abbruchkriterium zu begründen ist.

(13.25) Der Karmarkar-Algorithmus.

Input: $A \in \mathbb{Q}^{(m,n)}$ *und* $c \in \mathbb{Q}^n$*. Zusätzlich wird vorausgesetzt, dass* $\frac{1}{n}A\mathbb{1} = 0$ *und* $c^T\mathbb{1} > 0$ *gilt.*

Output: *Ein Vektor* x *mit* $Ax = 0$, $\mathbb{1}^T x = 1$, $x \geq 0$ *und* $c^T x \leq 0$ *oder die Feststellung, dass kein derartiger Vektor existiert.*

(1) **Initialisierung.** *Setze*

$$\begin{array}{ll} x^\circ & := \frac{1}{n}\mathbb{1} \\ k & := 0 \\ N & := 3n(\langle A\rangle + 2\langle c\rangle - n) \end{array}$$

(2) **Abbruchkriterium.**

(2.a) *Gilt* $k = N$*, dann hat* $Ax = 0$, $\mathbb{1}^T x = 1$, $x \geq 0$, $c^T x \leq 0$ *keine Lösung, STOP!*

(2.b) *Gilt* $c^T x^k \leq 2^{-\langle A\rangle - \langle c\rangle}$*, dann ist eine Lösung gefunden. Falls* $c^T x^k \leq 0$*, dann ist* x^k *eine Lösung, andernfalls kann wie bei der Ellipsoidmethode (Satz (12.34)) aus* x^k *ein Vektor* $\overline{x}$ *konstruiert werden mit* $c^T\overline{x} \leq 0$ $A\overline{x} = 0$, $\mathbb{1}^T\overline{x} = 1$, $\overline{x} \geq 0$*, STOP!*

Update.

(3) *(3.a)* $D := \mathrm{diag}(x^k)$

(3.b) $\overline{\overline{c}} := (I - DA^T(AD^2A^T)^{-1}AD - \frac{1}{n}\mathbb{1}\mathbb{1}^T)Dc$ *(siehe Beweis von (13.20))*

(3.c) $y^{k+1} := \frac{1}{n}\mathbb{1} - \frac{1}{2}\frac{1}{\sqrt{n(n-1)}}\frac{1}{\|\overline{\overline{c}}\|}\overline{\overline{c}}$

(3.d) $x^{k+1} := \frac{1}{\mathbb{1}^T D y^{k+1}} D y^{k+1}$

(3.e) $k := k + 1$

Gehe zu (2).

$\square$

Die Geschwindigkeit des Algorithmus hängt natürlich nur von der Implementierung des Schrittes (3) ab. Wir haben hier die aus Lemma (13.20) gewonnene Formel (13.21) für den nächsten Iterationspunkt x^{k+1} gewählt. Man kann x^{k+1} natürlich auch über (13.23), (13.24) bestimmen. Wesentlich ist, dass man eine Update-Formel findet, in der möglichst wenig arithmetische Operationen auftreten.

Es ist offensichtlich, dass die Hauptarbeit von (3) im Schritt (3.b) liegt. Führt man ihn kanonisch aus, so werden $0(n^3)$ Rechenschritte benötigt. Man beachte, dass sich in jedem Schritt nur die Diagonalmatrix D ändert. Diese Tatsache kann man, wie Karmarkar gezeigt hat, ausnutzen, um (3) in $0(n^{2.5})$ Rechenschritten zu erledigen. Die Laufzeit beträgt dann insgesamt $0(n^{2.5}(\langle A\rangle + \langle c\rangle))$.

Wie bei der Ellipsoidmethode können bei der Ausführung des Karmarkar-Algorithmus irrationale Zahlen (durch Wurzelziehen in (3.c)) auftreten. Wir sind also gezwungen, alle Rechnungen approximativ auszuführen. Die dadurch auftretenden Rundungsfehler akkumulieren sich natürlich. Wir müssen also unsere Rundungsvorschriften so einrichten, dass beim Abbruch des Verfahrens eine korrekte Schlussfolgerung gezogen werden kann. Auch das kann man wie bei der Ellipsoidmethode erledigen. Dies sind (auf Abschätzungstechniken der linearen Algebra und Analysis beruhende) Tricks, mit deren Hilfe Verfahren (13.25) in einen polynomialen Algorithmus abgewandelt werden kann. Da insgesamt höchstens N Iterationen durchgeführt werden, folgt

(13.26) Satz. *Die Laufzeit des (geeignet modifizierten) Karmarkar-Algorithmus (13.25) zur Lösung von Problemen des Typs (13.10) ist* $0(n^{3.5}(\langle A\rangle + \langle c\rangle)^2)$.

□

Wir wollen im weiteren die Rundungsfehler vernachlässigen und annehmen, dass wir in perfekter Arithmetik arbeiten. Es bleibt noch zu zeigen, dass Algorithmus (13.25) mit einem korrekten Ergebnis endet.

Zunächst überlegen wir uns, dass alle Punkte x^k im relativ Inneren von $\Omega \cap E \cap \mathbb{R}^n_+$ enthalten sind.

(13.27) Lemma. *Für alle* $k \in \{0, 1, \ldots, N\}$ *gilt* $Ax^k = 0$, $x^k > 0$, $\mathbb{1}^T x^k = 1$.

Beweis : Durch Induktion über k! Für $k = 0$ gilt die Behauptung nach Voraussetzung. Für $k+1$ ist $\overline{\overline{c}}$ die Projektion von c auf $\{x \mid ADx = 0, \mathbb{1}^T x = 0\}$, siehe Beweis von (13.20). Daraus folgt $AD\overline{\overline{c}} = 0$, $\mathbb{1}^T\overline{\overline{c}} = 0$. Mithin ergibt sich

$$\begin{aligned}(\mathbb{1}^T D y^{k+1}) A x^{k+1} &= ADy^{k+1} = AD(\tfrac{1}{n}\mathbb{1} - \tfrac{1}{2}\tfrac{1}{\sqrt{n(n-1)}}\tfrac{1}{\|\overline{\overline{c}}\|}\overline{\overline{c}}) \\ & AD(\tfrac{1}{n}\mathbb{1}) = Ax^k = 0.\end{aligned}$$

Daraus folgt $Ax^{k+1} = 0$.

Um $x^k > 0$ zu zeigen, stellen wir zunächst fest, dass $K := \{y \mid \|y - \frac{1}{n}\mathbb{1}\| \leq \frac{1}{2}\frac{1}{\sqrt{n(n-1)}}\} \subseteq \mathbb{R}^n_+$. Denn gibt es ein $\overline{y} \in K$ mit einer nichtpositiven Komponente, sagen wir $\overline{y}_1 \leq 0$, so gilt $\sum_{i=2}^n (\overline{y}_i - \frac{1}{n})^2 \leq \frac{1}{4n(n-1)} - (\overline{y}_1 - \frac{1}{n})^2 \leq \frac{1}{4n(n-1)} - \frac{1}{n^2} = \frac{-3n+4}{4n^2(n-1)} < 0$, ein Widerspruch. Daraus folgt $y^{k+1} > 0$, und (13.21) ergibt direkt $x^{k+1} > 0$.

Aus (13.21) folgt ebenalls sofort, dass $\mathbb{1}^T x^{k+1} = 1$ gilt. □

Entscheidend für die Abschätzung der Anzahl der Iterationsschritte ist die folgende Beobachtung.

(13.28) Lemma. *Gibt es ein $x \geq 0$ mit $Ax = 0$, $\mathbb{1}^T x = 1$ und $c^T x \leq 0$ dann gilt für alle k*

$$\frac{(c^T x^{k+1})^n}{\prod_{i=1}^n x_i^{k+1}} \leq \frac{2}{e} \frac{(c^T x^k)^n}{\prod_{i=1}^n x_i^k}.$$

□

Wir wollen uns nun überlegen, wieviele Iterationen im Verfahren (13.25) höchstens durchgeführt werden müssen. Der erste Iterationspunkt ist der Vektor

$$x^0 = \tfrac{1}{n}\mathbb{1}.$$

Daraus folgt mit einer groben Abschätzung aus (12.10) (a)

$$c^T x^0 \leq \tfrac{1}{n} \textstyle\sum_{i=1}^n |c_i| \leq \tfrac{1}{n} \sum_{i=1}^n 2^{\langle c_i \rangle - 1} < \tfrac{1}{n} 2^{\langle c \rangle - n}.$$

Aus Lemma (13.28) ergibt sich (unter den dort gemachten Voraussetzungen)

$$\frac{(c^T x^k)^n}{\prod_{i=1}^n x_i^k} \leq \left(\frac{2}{e}\right)^k \frac{(c^T x^0)^n}{\prod_{i=1}^n x_i^0}$$

und somit wegen $\prod_{i=1}^n x_i^k \leq 1$ und $\prod_{i=1}^n x_i^0 = (\frac{1}{n})^n$

(13.29)
$$\begin{aligned} c^T x^k &< \left(\tfrac{2}{e}\right)^{\frac{k}{n}} \left(\tfrac{\Pi x_i^k}{\Pi x_i^0}\right)^{\frac{1}{n}} \tfrac{1}{n} 2^{\langle c \rangle - n} \\ &\leq \left(\tfrac{2}{e}\right)^{\frac{k}{n}} 2^{\langle c \rangle - n}. \end{aligned}$$

In jedem Schritt schrumpft also der Zielfunktionswert um den nur von der Dimension n abhängigen Faktor $\sqrt[n]{\frac{2}{e}}$. Setzen wir

(13.30)
$$N := 3n(\langle A \rangle + 2n\langle c \rangle - n)$$

dann gilt

(13.31) Satz. *Es gibt ein* $x \geq 0$ *mit* $Ax = 0$, $\mathbb{1}^T x = 1$, $c^T x \leq 0$ *genau dann, wenn es ein* $k \in \{0, \dots N\}$ *gibt mit*

$$c^T x^k < 2^{-\langle A \rangle - \langle c \rangle}.$$

Beweis : "$\Longrightarrow$" Angenommen, es gibt ein $x \in P = \Sigma \cap \Omega$ mit $c^T x \leq 0$, dann sind die Voraussetzungen von Lemma (13.28) erfüllt und folglich gilt mit (13.29) und (13.30)

$$c^T x^N < \left(\frac{2}{e}\right)^{\frac{N}{n}} 2^{\langle c \rangle - n} < 2^{-\langle A \rangle - \langle c \rangle},$$

wobei wir die Tatsache ausgenutzt haben, dass $3 > \frac{-1}{\log(\frac{2}{e})}$ gilt.

"$\Longleftarrow$" Angenommen $c^T x^k < 2^{-\langle A \rangle - \langle c \rangle}$ gilt für ein $k \in \{0, \dots, N\}$. Da $P \neq \emptyset$, ein Polytop ist, wird $\min\{c^T x \mid x \in P\}$ in einer Ecke von P angenommen. Nach Satz (12.15) ist der Optimalwert dieses Programms eine rationale Zahl $\frac{p}{q}$ mit

$$1 \leq q \leq 2^{\langle A \rangle + \langle \mathbb{1} \rangle + \langle c \rangle - n^2 - n} < 2^{\langle A \rangle + \langle c \rangle}.$$

Aus $\frac{p}{q} \leq c^T x^k < 2^{-\langle A \rangle - \langle c \rangle}$ folgt dann $\frac{p}{q} \leq 0$. $\square$