OPERATIONS RESEARCH

WS 1998/99

Version v. 14.07.99

Vorlesung: Thomas Hofmeister

Skript: Ingo Wegener
(mit kleinen Änderungen)

Universität Dortmund

Lehrstuhl Informatik 2

D–44221 Dortmund

Vorwort

Ich freue mich über Verbesserungsvorschläge und Hinweise auf Fehler in diesem Skript. Es gibt eine WWW–Seite, auf der es begleitende Materialien, eine Fehlerliste etc. geben wird. Die URL:
`http://Ls2-WWW.cs.uni-dortmund.de/OR9899/`

Thomas Hofmeister, 13.10.1998

Inhaltsverzeichnis

Dieses Skript wurde von Heike Griehl, Matthias Bußmann, Ludwig Voß und Werner Traidl mit dem Textsystem LaTeX geschrieben.

1 Einleitung

Die Entwicklung vieler Wissenschaftszweige ist in den letzten 50 Jahren durch die Mathematisierung der Einzelwissenschaften geprägt. Vorreiter dieser Entwicklung waren die Naturwissenschaften wie Physik, Chemie und später Biologie, noch später folgten Psychologie, Soziologie und die Wirtschaftswissenschaften. Die Informatik, die in vielen Teilbereichen ihre Wurzeln direkt in der Mathematik hat, spielt hierbei eine Sonderrolle. Was ist der Grund für die Mathematisierung der Einzelwissenschaften? Stark verallgemeinernd ist es der Wunsch, qualitative und daher unpräzise Aussagen zu quantifizieren und damit direkt in Handlungsanweisungen münden zu lassen. Wegen der Komplexität der quantitativen Methoden sind diese für Probleme von praktischer Relevanz nur mit Rechnerhilfe einzusetzen, und somit ist die Informatik an der Mathematisierung der Einzelwissenschaften direkt beteiligt.

Bei der Mathematisierung der Betriebswirtschaftslehre entstand das Gebiet Operations Research, das sich heute zwischen Mathematik, BWL, Informatik und einigen Ingenieurwissenschaften ansiedelt. Der Begriff Operations Research soll hier nicht definiert werden, da mir keine wirklich passende Definition bekannt ist. Die folgenden Gebiete sollen jedoch durch quantitative Methoden unterstützt werden: Planung, Entscheidungsvorbereitung, Unternehmensforschung, Marktanalyse, Bewertung von Konfliktsituationen. Es sollte Informatikerinnen und Informatiker daher nicht überraschen, daß auch das Gebiet Operations Research wesentliche Wurzeln in der militärischen Forschung und in militärischen Anwendungen hat, und nach teilweiser geheimer Forschung im Zweiten Weltkrieg seit dem Ende der 40er Jahre öffentlich wissenschaftlich unterstützt und betrieben wird.

Geradezu typisch ist der „dual use“ Charakter der Operations Research Methoden. Für die folgenden Probleme können im wesentlichen die gleichen Methoden verwendet werden: die Suche nach einem abgestürzten Flugzeug in einem unübersichtlichen Dschungelgebiet sowie die Aufspürung eines feindlichen Kriegsschiffes. Die hierfür benutzbaren Methoden wurden eingesetzt, um vor einigen Jahren die Titanic wiederzufinden, 1966 eine von der USA verlorene H-Bombe im Mittelmeer bei Palomares (Spanien) zu bergen und um 1968 „vor den Russen“ das gesunkene US-Atom-U-Boot „Scorpion“ zu entdecken und zu heben. Was unterscheidet die Minimierung des Einsatzes von Dünger oder Insektenvertilgungsmitteln in der Landwirtschaft methodisch von der Planung eines kostengünstigsten Bombenteppichs? Wohl mehr die verwendeten Daten als die verwendeten Methoden. Nichts kann die Informatikerin und den Informatiker von der Verantwortung befreien, vor dem Einsatz von Operations Research Methoden die Entscheidung zu treffen, ob das gestellte Problem überhaupt gelöst werden sollte und ob gegebenenfalls die vorgenommene Modellierung angemessen ist und die benutzte Methode nicht wesentliche Aspekte ausklammert.

Gleiches gilt für betriebswirtschaftliche Anwendungen. So endet der erste Satz in dem Buch „Operations Research — Quantitative Methoden zur Entscheidungsvorbereitung“ (W. Zimmermann, Oldenbourg, 1990): „Da das Ziel jedes unternehmerischen Handelns letztlich in der Maximierung des langfristigen Gesamtgewinns gesehen werden kann, besteht die sogenannte Kunst der Unternehmensführung in der optimalen Kombination der Elementarfaktoren Mensch, Maschine, Material, Money und Management hinsichtlich dieses Ziels.“ Muß nun jemand, der diese zumindest übersimplifizierende Sicht der Welt nicht teilt, andere Operations Research Methoden entwickeln als jemand, der die genannte Weltsicht teilt? Dazu sehen wir uns die drei Phasen an, die beim Einsatz von Operations Research Verfahren auf ein reales Problem durchlaufen werden:

- Umwandlung des realen Problems in ein abstraktes Modell. Beschreibung des angestrebten Ziels und der zu beachtenden Nebenbedingungen.
- Lösung des entstandenen Optimierungsproblemes.
- Interpretation der gefundenen Lösung(en) und Rückübersetzung in einen Handlungsvorschlag für das reale Problem.

Besonders die erste Phase enthält die kritischen Entscheidungen: Soll das Problem überhaupt gelöst werden? Ist eine Quantifizierung der Lösung des Problems angemessen? Inwieweit verändert die abstrakte

Modellierung das Problem? Werden eventuell Lösungsmöglichkeiten ausgeschlossen? Sind alle Nebenbedingungen berücksichtigt und bezogen auf ihre Wichtigkeit bewertet worden? Paßt das Optimalitätskriterium zum Problem? Wessen Interessen werden modelliert? Diese Liste erhebt natürlich keinen Anspruch auf Vollständigkeit. Es soll aber ausdrücklich dazu aufgefordert werden, sich in diese Entscheidungen einzumischen. In der zweiten Phase wird dann ein abstraktes Problem gelöst. In dieser Phase werden Operations Research Verfahren eingesetzt. Nach der Rückübersetzung der Lösung in einen Handlungsvorschlag sollte dieser noch einmal kritisch überprüft werden.

Operations Research Methoden sind also die Methoden, die in der zweiten Phase eingesetzt werden. Dies führt auch in dieser Vorlesung dazu, daß wir abstrakte Methoden behandeln und von den oben andiskutierten Problemen abstrahieren. In konkreten Anwendungen dürfen wir uns aber nicht mit dem Verweis, nur ein abstraktes Problem zu lösen, aus der Verantwortung stehlen.

Gemäß der obigen Beschreibung des Gebietes Operations Research ist klar, daß die Problem- und Methodenvielfalt eine Behandlung aller Teilgebiete des Operations Research ausschließt. Die Auswahl der Themen wurde wesentlich dadurch bestimmt, Überschneidungen mit bereits länger etablierten Stammvorlesungen in Dortmund zu vermeiden. Daher werden die folgenden Bereiche nicht behandelt.

- Kombinatorische Optimierung, Lösung diskreter Optimierungsprobleme wie Traveling Salesman Problem, Rucksackproblem, Flußprobleme, Zuordnungsprobleme (Matching), Kürzeste Wege und Transportprobleme. Diese Probleme werden in der Vorlesung „Effiziente Algorithmen“ behandelt.
- Simulation, da es hierfür eine eigenständige Vorlesung gibt.
- Warteschlangensysteme, die im Rahmen der Vorlesung „Betriebssysteme“ untersucht werden.

Was bleibt für unsere Vorlesung übrig? In Kapitel 2 behandeln wir das Problem der Linearen Optimierung. Einerseits sind lineare Zielfunktionen und lineare Nebenbedingungen die wichtigsten Spezialfälle für Optimierungsprobleme. Andererseits ist der Simplexalgorithmus zur Lösung von linearen Optimierungsproblemen der älteste Operations Research Algorithmus, der noch heute alltägliche Anwendung findet. Bei der Lösung linearer Optimierungsprobleme sind die Lösungen im allgemeinen reellwertig, wobei eine Einschränkung auf rationale, aber nicht auf ganze Zahlen in vielen Praxissituationen möglich ist. Es liegt jedoch in der Natur „vieler Sachen“, daß sie nicht geteilt werden und nur als ganze Einheiten verwendet werden können. Die Einschränkung auf ganzzahlige Lösungen verändert das Problem der Linearen Optimierung grundlegend. Zur Lösung von Problemen der Ganzzahligen Optimierung sind vollständig andere Algorithmen nötig. Derartige Algorithmen behandeln wir in Kapitel 4.

Viele Optimierungsprobleme (auch die Lineare und die Ganzzahlige Optimierung) setzen eine vollständige Kenntnis aller Parameter voraus. Insbesondere ist die Informationsgewinnung vor der Optimierung und vor dem Beginn eigener Handlungen abgeschlossen. Im Gegensatz dazu sehen Planungsprobleme die Planung von Handlungen für einen längeren Zeitraum vor. Wenn wir nun während dieses Zeitraums hinzulernen oder neue Informationen erhalten, sollten wir darauf reagieren können. Kann dennoch ein vollständiger Handlungsplan erstellt werden? In der in Kapitel 8 behandelten Stochastischen Optimierung (manchmal auch Dynamische Optimierung genannt) wird die Kenntnis aller zukünftigen Möglichkeiten vorausgesetzt. Nach einer Handlung von uns am ersten Tag (zum ersten Zeittakt) geschieht etwas, z.B. handeln andere Personen, Firmen oder auch der Zufall. Unsere Handlung am zweiten Tag darf dann von der Situation am Vorabend abhängen. Da natürlich unsere Handlungen die Reaktionen anderer beeinflussen (und umgekehrt), wird im stochastischen Modell angenommen, daß wir zwar nicht genau wissen, was andere tun, aber Wahrscheinlichkeiten für diese Aktionen kennen. Es soll dann ein Handlungsplan (wenn der dies und jenes tut, dann werde ich ...) für einen endlichen oder unendlichen Zeithorizont erstellt und die erwartete Güte maximiert werden.

In den untersuchten Optimierungsproblemen ist die Form der zu optimierenden Funktion eingeschränkt, und die besten Lösungsverfahren sind speziell auf diese Funktionsform zugeschnitten. Neben weiteren speziellen Problemtypen sind natürlich auch allgemeine Methoden für Optimierungsprobleme von Interesse. Derartige Methoden werden in Kapitel 5 vorgestellt.

In Kapitel 6 beschäftigen wir uns mit dem Problem der Aufstellung geeigneter Nutzenfunktionen und mit den Grundzügen der Entscheidungstheorie. Schließlich soll in Kapitel 7 eine Einführung in die Spieltheorie gegeben werden. In den zuvor behandelten Optimierungsproblemen ist die Möglichkeit der Kommunikation und der Erarbeitung von Verhandlungslösungen nicht vorgesehen. Reaktionen der Außenwelt wurden zwar in das Modell eingearbeitet, aber nicht die Erarbeitung einer einvernehmlichen Lösung. Verhandlungen sind jedoch in vielen Bereichen die einzige Möglichkeit, um zu vernünftigen Lösungen zu gelangen. Die Spieltheorie untersucht Konflikte und die Suche nach Verhandlungslösungen. Nur Konflikte zwischen zwei Partnern mit diametral entgegengesetzten Zielen machen Verhandlungen überflüssig. Sie lassen sich mit Methoden der Linearen Optimierung lösen. In allen anderen Problemen ist es wichtig und nützlich, die Interessen der Verhandlungspartner zu berücksichtigen. Vor der Verhandlung sollten die Verhandlungspositionen und -stärken aller Beteiligten analysiert werden. Welche Koalitionen erweisen sich als gewinnträchtig? Wie können gerechte Lösungen aussehen?

Die Vorlesung folgt keinem Lehrbuch, da die Lehrbücher entweder mehr aus dem Blickwinkel der Mathematik oder mehr aus dem Blickwinkel der BWL geschrieben sind. Aus Sicht der Mathematik stehen auch Existenzaussagen im Vordergrund, während es aus Sicht der Informatik darauf ankommt, Algorithmen mit realistischen Rechenzeiten zu erhalten. Die BWL-Lehrbücher über Operations Research beschreiben häufig nur die Verfahren. Dagegen sollten in der Informatik Algorithmuskenntnisse stets auch das Wissen, warum die Algorithmen korrekt sind, und eine Laufzeitanalyse beinhalten. Die vollständige Behandlung von Beispielen führt ab einer bestimmten Größe in Vorlesungen zu Langeweile. BWL-Lehrbücher wie das bereits genannte von Zimmermann können gut als Nachschlagewerk für größere Beispiele benutzt werden.

2 Lineare Optimierung

2.1 Das Problem und eine grundlegende Lösungsidee

Ein Problem der Linearen Optimierung ist das Problem, eine lineare Zielfunktion unter linearen Nebenbedingungen zu minimieren. Es ist gegeben durch die Zahl n der Variablen, die Zahl m der Nebenbedingungen, die Zielfunktion

$$Z(x) := c_1x_1 + \ldots + c_nx_n \to \min,$$

die m Nebenbedingungen

$$a_{i1}x_1 + \ldots + a_{in}x_n \leq b_i \text{ für } 1 \leq i \leq m$$

und die weiteren Bedingungen

$$x_j \geq 0 \quad \text{für } 1 \leq j \leq n.$$

Prinzipiell können die Parameter a_{ij}, b_i und c_j, $1 \leq i \leq m$, $1 \leq j \leq n$, reelle Zahlen sein. Für eine Bearbeitung im Rechner nehmen wir jedoch an, daß die Parameter rationale Zahlen sind. Ansonsten ist die Problemformulierung allgemeiner als sie auf den ersten Blick aussieht. Wenn wir eine lineare Funktion maximieren und nicht minimieren wollen, genügt es, die Zielfunktion mit $\Leftrightarrow$1 zu multiplizieren, um wieder ein Minimierungsproblem zu erhalten. Bedingungen vom Typ $\geq$ werden ebenfalls durch Multiplikation mit $\Leftrightarrow$1 zu Nebenbedingungen vom Typ $\leq$. Nebenbedingungen, die durch Gleichungen gegeben sind, werden in eine $\geq$- und eine $\leq$-Nebenbedingung zerlegt. Schließlich kann eine unbeschränkte Variable x_j durch die Differenz zweier Variablen $x'_j \Leftrightarrow x''_j$ mit $x'_j \geq 0$ und $x''_j \geq 0$ ausgedrückt werden. Nur Nebenbedingungen vom Typ $<$ werden ausgeschlossen. Dies macht aus zwei Gründen Sinn. Aus praktischer Sicht sind $\leq$-Bedingungen (nicht mehr als xDM ausgeben) viel natürlicher als $<$-Bedingungen. Aus theoretischer Sicht führen $<$-Bedingungen zu Problemen ohne Lösungen. Schon die Maximierung von x_1 unter der Nebenbedingung $x_1 < 1$ hat offensichtlich keine Lösung.

Der einfacheren Schreibweise wegen führen wir folgende Notation ein. Die Parameter werden zu Spaltenvektoren c, x und b und zu einer $m \times n$-Matrix A zusammengefaßt. Die Aufgabe besteht dann in der Minimierung von $Z(x) = c^Tx$ (dabei ist c^T der transponierte Vektor, also ein Zeilenvektor) unter den Nebenbedingungen $Ax \leq b$ und $x \geq 0$.

Lineare Funktionen sind, wie wir aus der Schulmathematik wissen, die einfachsten nicht trivialen Funktionen. Darüber hinaus kommen sie aber in der Praxis häufig vor. Der Preis für ein Produkt steigt beispielsweise (von Mengenrabatten abgesehen) linear mit der Menge, die wir von dem Produkt kaufen.

Wir betrachten nun die Zielfunktion und die Nebenbedingungen genauer. Die Bedingung $a_{i1}x_1 + \ldots + a_{in}x_n = b_i$ beschreibt im $\mathbb{R}^n$ eine Hyperebene der Dimension $n \Leftrightarrow 1$: Wenn $n \Leftrightarrow 1$ Variablen festgelegt sind, läßt sich die n-te Variable ausrechnen. Für $n = 2$ beschreiben die Nebenbedingungen Geraden und für $n = 3$ Ebenen. Die Bedingung $a_{i1}x_1 + \ldots + a_{in}x_n \leq b_i$ beschreibt alle Punkte auf der Hyperebene $a_{i1}x_1 + \ldots + a_{in}x_n = b_i$ und alle Punkte auf einer Seite der Hyperebene. Die Menge der Punkte, die diese Nebenbedingung erfüllen, bildet einen Halbraum. Gleiches gilt auch für die Nebenbedingungen $x_i \geq 0$. Alle x-Vektoren, für die die Zielfunktion einen festen Wert z annimmt, bilden ebenfalls eine Hyperebene.

Der Bereich der zulässigen Punkte ergibt sich als Durchschnitt der Mengen der Punkte, die die einzelnen Nebenbedingungen erfüllen. Wenn der Bereich leer ist (z.B. $x_1 \to \min$ mit $x_1 \leq \Leftrightarrow$1 und $x_1 \geq 0$), sprechen wir von einem unzulässigen Problem. Das Problem hat keine Lösung, da die Nebenbedingungen nicht erfüllbar sind. Wenn der Bereich nicht leer und beschränkt ist (z.B. $x_1 \to \min$ mit $x_1 \leq 4$ und $x_1 \geq 0$), gibt es stets eine Lösung, da der Bereich der zulässigen Punkte dann kompakt ist (beschränkt nach Voraussetzung und abgeschlossen als Durchschnitt abgeschlossener Mengen) und da stetige Funktionen auf kompakten Mengen ihr Minimum und Maximum annehmen. Wenn der Bereich zulässiger Punkte unbeschränkt ist, kann es eine Lösung geben (z.B. $x_1 \to \min$ mit $\Leftrightarrow x_1 \leq \Leftrightarrow$1 und $x_1 \geq 0$), oder keine Lösung geben (z.B. $\Leftrightarrow x_1 \to \min$ mit $\Leftrightarrow x_1 \leq \Leftrightarrow$1 und $x_1 \geq 0$). Im zweiten Fall kann der Wert $\Leftrightarrow\infty$ der Zielfunktion angenähert werden. Wenn ein praktisch relevantes Problem richtig modelliert ist, sollte der Bereich zulässiger Punkte nicht leer sein, und es sollten nicht beliebig kleine Werte der Zielfunktion unterschritten werden können.

Wir behandeln nun ein sehr kleines Beispiel ohne praktische Relevanz, um daran die allgemeine Lösungsstrategie zu illustrieren.

Beispiel 2.1.1: Ein Hobbygärtner will die 100 m^2 Nutzfläche seines Gartens optimal mit Blumen und Gemüse bepflanzen. Sein Kapital beträgt DM 720, für den Anbau sind Arbeits- und Materialkosten von 6 DM/m^2 für Gemüse und 9 DM/m^2 für Blumen zu veranschlagen. Der zu erwartende (und zu maximierende) Gewinn beträgt 10 DM/m^2 für Gemüse und 20 DM/m^2 für Blumen. Schließlich sind nur 60 m^2 aufgrund von Sonne und Schatten für den Blumenanbau geeignet. Daraus ergibt sich folgendes Problem der Linearen Optimierung:

$$\Leftrightarrow 20x_1 \Leftrightarrow 10x_2 \to \min, \text{ wobei}$$
$$x_1 + x_2 \leq 100$$
$$9x_1 + 6x_2 \leq 720$$
$$x_1 \leq 60$$
$$x_1 \geq 0, x_2 \geq 0.$$

Hierbei steht x_1 für die für den Blumenanbau vorgesehene Fläche. Da wir nur zwei Variablen haben (in der Praxis sind 100 Variablen nicht viel), läßt sich der Bereich zulässiger Punkte einfach graphisch darstellen, indem wir die fünf Geraden für die Nebenbedingungen und die Halbräume der zulässigen Punkte beschreiben.

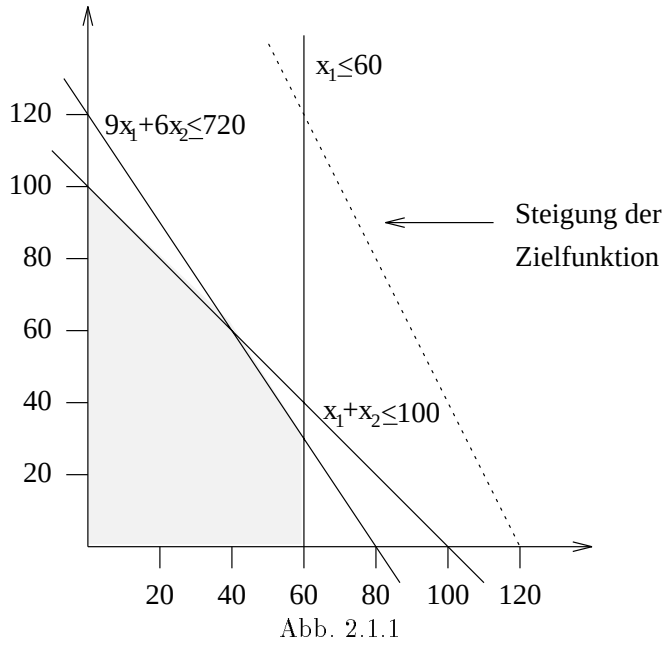


Abb. 2.1.1

Wenn wir für verschiedene z-Werte die Gleichung $\Leftrightarrow 20x_1 \Leftrightarrow 10x_2 = z$ oder $20x_1 + 10x_2 = \Leftrightarrow z$ betrachten, stellen wir fest, daß diese Geraden stets die gleiche Steigung haben. Je kleiner der z-Wert, desto weiter „rechts“ verläuft die Gerade. In unserer Abbildung ist $z = \Leftrightarrow 2400$, und die Gerade schneidet den Bereich der zulässigen Punkte nicht, d.h. der Gewinn von DM 2400 ist nicht realisierbar. Indem wir den gewünschten Gewinn sinken lassen, wird die Zielfunktionsgerade parallel nach links verschoben. Für $z = \Leftrightarrow 1500$ berührt sie den zulässigen Bereich erstmals im Punkt (60,30). Der Gewinn von DM 1500 ist also erreichbar (60 m^2 Blumen und 30 m^2 Gemüse), während ein größerer Gewinn nicht erreichbar ist. Bei dieser Lösung bleiben dem Gärtner sogar 10 m^2 für seinen Liegestuhl.

Im zweidimensionalen Fall erhalten wir als zulässige Menge stets ein konvexes Polygon. Wenn es eine optimale Lösung gibt, wird diese auch an einem Eckpunkt (Schnittpunkt zweier Geraden) angenommen, da bei der Parallelverschiebung der Zielgeraden zunächst eine Ecke erwischt wird. Es kann auch ein ganzes Geradenstück erwischt werden, wenn die Zielgerade parallel zu einer Nebenbedingungsgerade verläuft. Zu diesem Geradenstück gehört dann auch ein Eckpunkt.

Im dreidimensionalen Raum arbeitet unsere geometrische Vorstellung noch. Der zulässige Bereich ist ein konvexes Polyeder, dessen Ecken Schnittpunkte von drei linear unabhängigen Ebenen sind, wobei diese Ebenen sich aus den Nebenbedingungen des Problems ergeben. Auch die Zielfunktion ist für fest vorgegebene Werte eine Ebene, die wir von $z = \Leftrightarrow\infty$ für wachsendes z an den zulässigen Bereich heranwandern lassen. Auch hier ist noch anschaulich klar, daß der zulässige Bereich zuerst in einem Eckpunkt getroffen wird. Es läßt sich vermuten, daß dies im n-dimensionalen Fall ebenfalls gilt. Das müssen wir dann jedoch beweisen.

Was sind allgemein Eckpunkte? Schnittpunkte von n linear unabhängigen Ebenen, die durch die Nebenbedingungen gegeben sind. Wenn unsere Intuition nicht trügt, genügt es also, uns nur die Eckpunkte anzuschauen. Nach unseren Vorbetrachtungen ist die Zahl der Eckpunkte nur durch $\binom{n+m}{n}$ zu beschränken. Diese im allgemeinen in n und m exponentielle Zahl ist aber immerhin endlich. Anstelle der überabzählbar vielen Punkte im zulässigen Bereich können wir uns, wenn unsere Intuition trägt, auf die endlich vielen Eckpunkte beschränken.

Z.B. für $n = m = 50$ gibt es viel zu viele Eckpunkte, um alle zu untersuchen. Wir werden dann eine Greedy-Strategie benutzen. Zunächst suchen wir einen zulässigen Eckpunkt. Von einem Eckpunkt gehen viele, genau n, Kanten aus. Wir wählen dann stets die Kante, auf der die Zielfunktion „am schnellsten“ fällt, wandern diese Kante bis zum nächsten Eckpunkt entlang und fahren dort analog fort. Wenn auf allen Kanten von einem Eckpunkt aus der Wert der Zielfunktion wächst, haben wir zumindest ein lokales Minimum erreicht. Wir müssen theoretisch nachweisen, daß der Eckpunkt dann eine globale Lösung darstellt.

Nach diesen Vorbetrachtungen sollte der Weg zur Lösung ebenso intuitiv klar sein wie die Erkenntnis, daß zur Verifikation des Algorithmus theoretische Arbeit nötig ist.

2.2 Konvexe Mengen

Wir haben schon von konvexen Polygonen und konvexen Polyedern gesprochen, ohne den Begriff konvex zu definieren. In diesem Abschnitt wollen wir die für unsere Zwecke wichtigen Eigenschaften von konvexen Mengen im $\mathbb{R}^n$ herausarbeiten.

Definition 2.2.1:

- Für zwei Punkte $x, y \in \mathbb{R}^n$ bezeichnet die „Verbindungsstrecke zwischen x und y“ die Menge der Punkte $\lambda x + (1 \Leftrightarrow \lambda)y$ mit $0 \leq \lambda \leq 1$.
- Eine Menge $M \subseteq \mathbb{R}^n$ heißt konvex, wenn für alle $x,\ y \in M$ die Verbindungsstrecke zwischen x und y in M liegt.

Es folgt nun sofort, daß der Durchschnitt M beliebig vieler konvexer Mengen $M_i,\ i \in I$, wieder konvex ist. Wenn $x,\ y \in M$ ist, ist $x,\ y \in M_i$ für alle $i \in I$. Also ist die Verbindungsstrecke zwischen x und y in allen M_i und damit in M enthalten. Die Nebenbedingungen in der Linearen Optimierung beschreiben abgeschlossene Halbräume

$$a_{i1}x_1 + \ldots + a_{in}x_n \leq b_i \quad \text{oder} \quad x_i \geq 0.$$

Halbräume sind konvex: Aus $a_i^T x \leq b_i$ und $a_i^T y \leq b_i$ folgt

$$a_i^T(\lambda x + (1 \Leftrightarrow \lambda)y) = \lambda a_i^T x + (1 \Leftrightarrow \lambda)a_i^T y \leq \lambda b_i + (1 \Leftrightarrow \lambda)b_i = b_i.$$

Damit haben wir die zulässigen Mengen von Problemen der Linearen Optimierung charakterisiert.

Satz 2.2.2: *Der zulässige Bereich eines Problems der Linearen Optimierung ist der Durchschnitt von $n+m$ abgeschlossenen Halbräumen und selber konvex und abgeschlossen.*

Wir wollen die in Kap. 2.1 bereits diskutierten Eckpunkte oder allgemeiner Extrempunkte auszeichnen. Dazu diskutieren wir zunächst, was die konvexe Hülle von Mengen in $\mathbb{R}^n$ ist.

Definition 2.2.3: Die konvexe Hülle $KoH\ddot{u}(M)$ von $M \subseteq \mathbb{R}^n$ ist der Durchschnitt aller konvexen Mengen S mit $S \supseteq M$.

Die konvexe Hülle ist also die kleinste konvexe Obermenge der gegebenen Menge. Nach unseren Vorbemerkungen ist sie selber konvex. Da $\mathbb{R}^n$ konvex ist, ist $KoH\ddot{u}(M)$ stets wohldefiniert. Für konvexe Mengen M gilt offensichtlich $KoH\ddot{u}(M) = M$.

Definition 2.2.4: Der Punkt $x_0 \in \mathbb{R}^n$ ist konvexe Linearkombination (kL) von $x_1, \ldots, x_p \in \mathbb{R}^n$, wenn es $\lambda_i \geq 0$ mit $\sum\limits_{1 \leq i \leq p} \lambda_i = 1$ und $x_0 = \sum\limits_{1 \leq i \leq p} \lambda_i x_i$ gibt.

Satz 2.2.5: *Die konvexe Hülle von $M \subseteq \mathbb{R}^n$ ist genau die Menge S aller kL's von Punkten aus M.*

Beweis: Wir zeigen zunächst, daß $S \subseteq KoH\ddot{u}(M)$ gilt. Sei also x_0 eine kL der Punkte $x_1, \ldots, x_p \in M$ zu $\lambda_1, \ldots, \lambda_p$. Wir benutzen Induktion über p. Für $p = 1$ ist $x_0 = 1x_1 = x_1 \in M \subseteq KoH\ddot{u}(M)$. Sei nun $p > 1$. Falls $\lambda_p = 1$, ist die Aussage wieder trivial. Falls $\lambda_p < 1$, kann man wie folgt ausklammern:

$$x_0 = (1 \Leftrightarrow \lambda_p) \underbrace{(\frac{\lambda_1}{1 \Leftrightarrow \lambda_p} x_1 + \ldots + \frac{\lambda_{p-1}}{1 \Leftrightarrow \lambda_p} x_{p-1})}_{x'} + \lambda_p x_p .$$

Nach Induktionsvoraussetzung gehört x' zu $KoH\ddot{u}(M)$ und aus der Konvexität von $KoH\ddot{u}(M)$ folgt $x_0 = (1 \Leftrightarrow \lambda_p)x' + \lambda_p x_p \in KoH\ddot{u}(M)$.

Wir zeigen nun, daß $KoH\ddot{u}(M) \subseteq S$ gilt. Da $x_0 = 1x_0$ eine kL ist, folgt $M \subseteq S$. Es genügt also zu zeigen, daß S konvex ist. Seien also $y, z \in S$, d.h.

$$y = \lambda_1 x_1 + \ldots + \lambda_p x_p \quad \text{mit} \quad \lambda_1 + \ldots + \lambda_p = 1, \quad \lambda_i \geq 0, \quad x_i \in M,$$

$$z = \lambda'_1 x'_1 + \ldots + \lambda'_m x'_m \quad \text{mit} \quad \lambda'_1 + \ldots + \lambda'_m = 1, \quad \lambda'_i \geq 0, \quad x'_i \in M,$$

Dann gilt für $\lambda \in [0, 1]$

$$\lambda y + (1 \Leftrightarrow \lambda) z = \lambda\lambda_1 x_1 + \ldots + \lambda\lambda_p x_p + (1 \Leftrightarrow \lambda)\lambda'_1 x'_1 \ldots + (1 \Leftrightarrow \lambda)\lambda'_m x'_m \text{ mit}$$

$$\lambda\lambda_1 + \ldots + \lambda\lambda_p + (1 \Leftrightarrow \lambda)\lambda'_1 + \ldots + (1 \Leftrightarrow \lambda)\lambda'_m = \lambda + (1 \Leftrightarrow \lambda) = 1,$$

$$\lambda\lambda_i \geq 0, \ (1 \Leftrightarrow \lambda)\lambda'_i \geq 0, \quad x_i \in M, \quad x'_j \in M.$$

Also ist $\lambda y + (1 \Leftrightarrow \lambda) z \in S$. □

Was sollen nun Extrempunkte von konvexen Mengen M sein? Sie sollen nicht im Innern einer Strecke, die ganz in M liegt, liegen. Wir benötigen auch die Definition von Randpunkten.

Definition 2.2.6: Ein Punkt x einer konvexen Menge M heißt

- Randpunkt von M, wenn es keine ε–Kugel ($\varepsilon > 0$) um x gibt, die nur Punkte aus M enthält.
- Extrempunkt von M, wenn aus $x = \lambda y + (1 \Leftrightarrow \lambda) z, \ y, z \in M$ und $\lambda \in (0, 1)$ folgt, daß $x = y = z$ ist.

Man überlege sich als Übungsaufgabe, daß ein Extrempunkt auch definiert werden kann als Punkt x, für den gilt: Es gibt kein y, so daß sowohl $x + y$ als auch $x \Leftrightarrow y$ in M enthalten sind.

Lemma 2.2.7: *Ein Extrempunkt x von M ist auch Randpunkt von M*

Beweis: Angenommen, x wäre nicht Randpunkt von M. Dann gäbe es eine ε–Kugel um x, die nur Punkte aus M enthält. Dann gibt es auch zwei Punkte x' und x'' aus M, die von x verschieden sind, so daß x als kL dieser beiden Punkte geschrieben werden kann. Also muß x Randpunkt von M sein. □

Lemma 2.2.8: *Seien $M_1, \ldots, M_t$ konvexe Mengen. Wenn x Randpunkt von $M := M_1 \cap \cdots \cap M_t$ ist, dann gibt es ein i, so daß x Randpunkt von M_i ist.*

Beweis: Angenommen, x wäre kein Randpunkt von einem M_i. Dann gäbe es für jedes i ein ε_i, so daß alle Punkte in der ε_i–Kugel um x in der Menge M_i wären. Für $\varepsilon := \min_i\{\varepsilon_i\}$ wären somit alle Punkte in der ε–Kugel um x in der Menge M und somit x kein Randpunkt von M. □

Lemma 2.2.9 (von der trennenden Hyperebene): *Sei $M \subseteq \mathbb{R}^n$ konvex und abgeschlossen und x_0 ein Randpunkt von M. Dann gibt es eine Hyperebene $H = \{y \mid uy = ux_0\}$ durch x_0, so daß alle $x \in M$ im gleichen Halbraum bzgl. H liegen, d.h. es ist $ux \geq ux_0$ für $x \in M$.*

Beweis: Da x_0 Randpunkt von M ist, enthält jede ε-Kugel um x_0, $\varepsilon > 0$, Punkte, die nicht in M liegen. Also gibt es eine Folge x_i von Punkten mit $x_i \notin M$ und $x_i \to x_0$. Mit $||x_i||$ bezeichnen wir die Euklidische Länge des Vektors $x_i \in \mathbb{R}^n$. Die Funktion

$$d(y) := ||x_i \Leftrightarrow y||$$

ist auf der abgeschlossenen Menge M stetig und durch 0 nach unten beschränkt. Also nimmt sie ihr Minimum in einem Punkt $z_i \in M$ an, d.h. z_i ist der x_i am nächsten gelegene Punkt in M. Damit liegt im Inneren der Kugel um x_i mit Radius $||x_i \Leftrightarrow z_i||$ kein Punkt von M. Es sei H_i die Hyperebene durch z_i, die tangential an der Kugel um x_i liegt. Mit der Wahl $u_i = \frac{z_i - x_i}{||z_i - x_i||} \in \mathbb{R}^n$ ist dies die Hyperebene

$$H_i = \{y \mid u_i y = u_i z_i\}.$$

Es gilt $||u_i|| = 1$ und $u_i z_i > u_i x_i$. Wir behaupten nun, daß $u_i x \geq u_i z_i$ für alle $x \in M$ gilt, d.h. M liegt ganz in einem durch H_i definierten Halbraum, und zwar in dem Halbraum, der x_i nicht enthält. Dies kann man wie folgt einsehen. Angenommen, es gäbe einen Punkt $x \in M$ mit $u_i x < u_i z_i$. Wir betrachten die Verbindungsstrecke zwischen den beiden, d.h. Punkte $w_\lambda := \lambda x + (1 \Leftrightarrow \lambda) z_i$. Definiere die stetige Funktion

$$f(\lambda) := ||w_\lambda \Leftrightarrow x_i||^2 \Leftrightarrow ||x_i \Leftrightarrow z_i||^2 = (w_\lambda \Leftrightarrow x_i)^2 \Leftrightarrow (x_i \Leftrightarrow z_i)^2 = \ldots = \lambda^2 (x \Leftrightarrow z_i)^2 + 2\lambda(\Leftrightarrow z_i^2 + x z_i \Leftrightarrow x x_i + x_i z_i)$$

Als Ableitung dieser Funktion ergibt sich:

$$f'(\lambda) = 2\lambda(x \Leftrightarrow z_i)^2 + 2(\Leftrightarrow z_i^2 + x z_i \Leftrightarrow x x_i + x_i z_i)$$

und somit $f'(0) = 2(x \Leftrightarrow z_i)(z_i \Leftrightarrow x_i) = c \cdot u_i \cdot (x \Leftrightarrow z_i)$ für eine Zahl $c > 0$. Da $u_i z_i > u_i x_i$, folgt $f'(0) < 0$. Da $f(0) = 0$ und $f'(0) < 0$, gibt es einen Punkt $\lambda_0 > 0$, für den $f(\lambda_0) < 0$ ist. Für den zugehörigen Punkt w_{λ_0} gilt dann, daß (nach Definition der Funktion f) w_{λ_0} näher an x_i ist als an z_i. Wegen der Konvexität von M wäre $w_{\lambda_0} \in M$. Dies ist aber ein Widerspruch zur Eigenschaft von z_i, daß dies der zu x_i nächstgelegene Punkt aus M ist.

Da $||u_i|| = 1$ für alle i, gibt es eine Teilfolge $u_{i(j)}$, die gegen einen Vektor u konvergiert.[1] Da $x_i \to x_0$, gilt auch $z_{i(j)} \to x_0$. Da $u_{i(j)} x \geq u_{i(j)} z_{i(j)}$ für alle $x \in M$, folgt $ux \geq ux_0$ für $x \in M$. Also ist

$$H = \{y \mid uy = ux_0\}$$

die gesuchte Hyperebene. □

[1] Dies ist eine Konsequenz des Satzes von Bolzano–Weierstraß, der besagt, daß jede beschränkte Folge eine konvergente Teilfolge enthält.

Satz 2.2.10 (von Krein-Milman): *Sei $M \subseteq \mathbb{R}^n$ konvex und kompakt (abgeschlossen und beschränkt). Dann läßt sich jeder Punkt $x \in M$ als kL von höchstens $n+1$ Extrempunkten von M darstellen.*

Die Voraussetzung der Beschränktheit von M ist notwendig, da z.B. die konvexe und abgeschlossene Menge $\mathbb{R}^n$ keinen Extrempunkt hat.

Beweis von Satz 2.2.10: Wir führen den Beweis durch Induktion über n. Für $n = 1$ hat jede konvexe und kompakte Menge die Form $[a, b]$ mit $a \leq b$ oder ist leer. Die Behauptung ist trivial.

Für $n > 1$ weisen wir zunächst die Existenz eines Extrempunktes nach und reduzieren das Problem dann um eine Dimension. Wenn M leer ist, ist nichts zu zeigen. Ansonsten sei $x_0 \in M$ beliebig gewählt. Wir betrachten eine beliebige Gerade g durch x_0. Geraden bilden abgeschlossene und konvexe Mengen. Also ist $g \cap M$ kompakt, konvex und eindimensional. Daher ist $g \cap M$ die Verbindungsstrecke zwischen x' und x'' mit $x', x'' \in g \cap M$. Da die Verlängerung dieser Strecke über x' (oder x'') hinaus zwar zu g, aber nicht zu M gehört, sind x' und x'' Randpunkte von M, d.h. es gibt keine ε-Kugeln ($\varepsilon > 0$) um x' oder x'', die ganz zu M gehören. Nach dem Lemma der trennenden Hyperebene existiert eine Hyperebene H, definiert durch $u \cdot y = c$, die x' enthält und für die eine Seite der Hyperebene keinen Punkt von M enthält. Da x' auf der Hyperebene H liegt, ist $ux' = c$ und damit

$$H = \{y \mid uy = ux', y \in \mathbb{R}^n\}.$$

Da M ganz in einem der beiden durch H beschriebenen Halbräume liegt, gilt o.B.d.A. $ux \geq ux'$ für alle $x \in M$.

Die Hyperebene H ist abgeschlossen, konvex und hat Dimension $n \Leftrightarrow 1$. Daher ist $H \cap M$ kompakt, konvex und liegt in einem Raum der Dimension $n \Leftrightarrow 1$. Nach Induktionsvoraussetzung läßt sich x' als kL von höchstens n Extrempunkten von $H \cap M$ darstellen. Wir behaupten, daß die Extrempunkte von $H \cap M$ auch Extrempunkte von M sind. Sei also z ein Extrempunkt von $H \cap M$. Falls z kein Extrempunkt von M ist, gilt $z = \lambda z_1 + (1 \Leftrightarrow \lambda) z_2$ für geeignete $z_1, z_2 \in M \Leftrightarrow \{z\}$ und $\lambda \in (0, 1)$. Da $z \in H$ und $z_1, z_2 \in M$, gilt $uz = ux' \leq uz_i,\ i \in \{1, 2\}$. Andererseits ist

$$uz = u(\lambda z_1 + (1 \Leftrightarrow \lambda) z_2) = \lambda u z_1 + (1 \Leftrightarrow \lambda) u z_2 \geq ux' = uz.$$

Da $0 < \lambda < 1$, muß $uz_i = ux'$ für $i \in \{1, 2\}$ gelten. Damit sind $z_1, z_2 \in H \cap M$. Da z Extrempunkt von $H \cap M$, folgt nach Definition $z = z_1 = z_2$ im Widerspruch zur Wahl von z_1 und z_2.

Insbesondere haben wir die Existenz von Extrempunkten von M nachgewiesen. Damit können wir die zu Beginn gewählte Gerade g so legen, daß x'' Extrempunkt von M ist. Die restlichen Überlegungen bleiben richtig. Es gilt nun für Extrempunkte $x'', x_1 \ldots, x_p$ $(p \leq n)$ von M

$$\begin{aligned} x_0 &= \lambda x' + (1 \Leftrightarrow \lambda) x'' \\ x' &= \lambda_1 x_1 + \ldots + \lambda_p x_p \quad \text{mit} \quad \lambda_1 + \ldots + \lambda_p = 1,\ \lambda_i \geq 0. \end{aligned}$$

Also ist x_0 eine kL der $p + 1 \leq n + 1$ Extrempunkte $x'', x_1, \ldots, x_p$ von M.

$$x_0 = \lambda\lambda_1 x_1 + \ldots + \lambda\lambda_p x_p + (1 \Leftrightarrow \lambda) x''.$$

□

2.3 Grundlegende Eigenschaften linearer Optimierungsprobleme

Wir werden nun die allgemeinen Ergebnisse aus Kap. 2.2 auf lineare Optimierungsprobleme anwenden.

Satz 2.3.1: *Falls die zulässige Menge M eines linearen Optimierungsproblems nicht leer ist, enthält sie mindestens einen Extrempunkt.*

Beweis: Falls M beschränkt ist, ist M nach Satz 2.2.2 konvex und kompakt. Die Behauptung folgt aus dem Satz 2.2.10 von Krein und Milman.

Sei nun M unbeschränkt und $x_0 \in M$. Falls x_0 der Nullvektor ist, ist x_0 auch Extrempunkt wegen der Nebenbedingungen $x_i \geq 0$. Ansonsten betrachten wir die Kugel K mit Radius $(n+2)||x_0||$ um den Nullpunkt. Da $M \cap K$ konvex und kompakt ist, läßt sich x_0 nach dem Satz von Krein und Milman darstellen als

$$x_0 = \sum_{1 \leq i \leq p} \lambda_i x_i, \quad \lambda_1 + \ldots + \lambda_p = 1,\ \lambda_i \geq 0,\ p \leq n+1$$
$$x_i \text{ Extrempunkt von } M \cap K.$$

Entweder gilt $||x_i|| = (n+2)||x_0||$, oder x_i ist bereits Extrempunkt von M. Wir müssen nur widerlegen, daß $||x_i|| = (n+2)||x_0||$ für alle i gelten kann. Wenn dies der Fall ist, wählen wir ein i mit $\lambda_i \geq \frac{1}{n+1}$. Sei $x_j = (x_{j,1}, \ldots, x_{j,n})$. Dann ist wegen der Nebenbedingungen in Problemen der Linearen Optimierung $x_{j,k} \geq 0$. Es folgt

$$||x_0|| = ||\lambda_1 x_1 + \ldots + \lambda_p x_p|| =$$
$$((\lambda_1 x_{1,1} + \ldots + \lambda_p x_{p,1})^2 + \ldots + (\lambda_1 x_{1,n} + \ldots + \lambda_p x_{p,n})^2)^{1/2} \geq$$
$$((\lambda_i x_{i,1})^2 + \ldots + (\lambda_i x_{i,n})^2)^{1/2} = \lambda_i ||x_i|| \geq \frac{n+2}{n+1}||x_0|| > ||x_0||,$$

was ein offensichtlicher Widerspruch ist. □

Satz 2.3.2: *Nimmt die Zielfunktion Z ihr Minimum auf der zulässigen Menge an, dann nimmt Z ihr Minimum auch in einem Extrempunkt der zulässigen Menge an.*

Beweis: Sei x_0 ein Punkt aus M mit $Z(x_0) = \min\{Z(x) | x \in M\}$. Falls $x_0 = \lambda_1 x_1 + \ldots + \lambda_p x_p$ eine kL mit $x_i \in M$ ist, folgt aus der Linearität von Z und der Optimalität von x_0

$$\begin{aligned} Z(x_0) &= \lambda_1 Z(x_1) + \ldots + \lambda_p Z(x_p) \\ &\geq \lambda_1 Z(x_0) + \ldots + \lambda_p Z(x_0) = Z(x_0). \end{aligned}$$

Es gilt also $Z(x_i) = Z(x_0)$, wenn wir o.B.d.A. $\lambda_i > 0$ für alle i annehmen. Wir haben im Beweis von Satz 2.3.1 gezeigt, daß wir x_0 so als kL darstellen können, daß für mindestens ein i der Punkt x_i Extrempunkt von M und $\lambda_i > 0$ ist. Also ist der Extrempunkt x_i auch optimal. □

Satz 2.3.3: *Jeder Extrempunkt der zulässigen Menge M ist Schnittpunkt von n linear unabhängigen Hyperebenen aus der Menge der $n+m$ Hyperebenen $H_1, \ldots H_{m+n}$, die durch die Nebenbedingungen (mit Gleichheit statt $\geq$ bzw. $\leq$) definiert sind. Es gibt maximal $\binom{m+n}{n}$ Extrempunkte der zulässigen Menge.*

Beweis: Sei x_0 ein Extrempunkt von M. x_0 ist damit auch Randpunkt von M und somit Randpunkt mindestens einer der Halbräume, die durch die Hyperebenen bestimmt sind. Somit liegt x_0 mindestens auf einer der Hyperebenen. Wir wählen I so, daß x_0 genau dann auf H_i liegt, wenn $i \in I$ ist. Der Durchschnitt aller $H_i,\ i \in I$, ist nicht leer, da er x_0 enthält. Wir nehmen nun an, daß nur höchstens $n \Leftrightarrow 1$ der Hyperebenen $H_i,\ i \in I$, linear unabhängig sind. Dann hat der Durchschnitt dieser Hyperebenen mindestens Dimension 1 und enthält eine Gerade g durch x_0. Von den anderen (endlich vielen) Hyperebenen $H_i, i \notin I$, hat x_0 einen Abstand, der für ein geeignetes $\varepsilon > 0$ größer als ε ist. Damit liegt die ε-Kugel K um x_0 auf der „richtigen“ Seite aller $H_i,\ i \notin I$. Insbesondere ist $K \cap g$ in M enthalten, und x_0 ist im Widerspruch zu seiner Extrempunkteigenschaft Mittelpunkt einer ganz in M verlaufenden Strecke. Die letzte Behauptung ist nun ein einfaches Korollar. □

Satz 2.3.4: *Jedes lokale Minimum der Zielfunktion Z auf der zulässigen Menge M ist ein globales Minimum.*

Beweis: Die Zielfunktion Z habe auf der zulässigen Menge M ein lokales Minimum in x_0. Wir nehmen an, daß es $x_1 \in M$ mit $Z(x_1) < Z(x_0)$ gibt. Da M konvex ist, liegt die Verbindungsstrecke zwischen x_0 und x_1 ganz in M. Für alle Punkte $\lambda x_0 + (1 \Leftrightarrow \lambda) x_1$ im Inneren dieser Strecke, d.h. $0 < \lambda < 1$, gilt

$$Z(\lambda x_0 + (1 \Leftrightarrow \lambda) x_1) = \lambda Z(x_0) + (1 \Leftrightarrow \lambda) Z(x_1) < Z(x_0).$$

Für $\lambda \to 1$ gilt $\lambda x_0 + (1 \Leftrightarrow \lambda) x_1 \to x_0$ im Widerspruch zur lokalen Minimalität von x_0. □

Wir fassen den Nutzen unserer bisherigen Resultate zusammen. Wir können für alle $\binom{m+n}{n}$ Möglichkeiten, n der gegebenen Hyperebenen auszuwählen, testen, ob diese linear unabhängig sind. Wir erhalten alle Extrempunkte der zulässigen Menge, indem wir die Schnittpunkte von n linear unabhängigen Hyperebenen auf ihre Zulässigkeit prüfen. Wir können dabei auch den Extrempunkt berechnen, der unter allen Extrempunkten die Zielfunktion minimiert. Wenn es keinen Extrempunkt gibt, ist die zulässige Menge leer. Ansonsten haben wir das Optimierungsproblem gelöst, wenn die Zielfunktion ihr Minimum auf der zulässigen Menge annimmt. Wir benötigen noch einen Test, um festzustellen, ob das Optimierungsproblem unbeschränkt ist, d.h. ob keine Lösung angenommen wird.

Darüber hinaus ist der so skizzierte Algorithmus viel zu langsam, um praktische Relevanz zu haben. Dantzig hat 1947 auf dieser Basis einen anderen Weg beschritten. Zunächst soll ein zulässiger Extrempunkt gesucht oder entschieden werden, daß die zulässige Menge leer ist. Vom ersten zulässigen Extrempunkt an wird von den n Hyperebenen, die den Extrempunkt definieren, eine gegen eine neue Hyperebene ausgetauscht, um den nächsten Extrempunkt zu definieren. Dabei wird sichergestellt, daß nur noch zulässige Extrempunkte gebildet werden und der Wert der Zielfunktion auf den Extrempunkten fällt. Dabei wird auch getestet, ob die Zielfunktion beliebig kleine Werte annehmen kann. Wenn ein Extrempunkt lokal optimal ist, können wir in der Gewißheit abbrechen, daß der Extrempunkt global optimal ist.

Wir bemerken noch, daß zulässige Schnittpunkte von n linear unabhängigen Hyperebenen stets Extrempunkte sind. Da n linear unabhängige Hyperebenen genau einen Punkt gemeinsam haben, können sie nämlich kein Geradenstück positiver Länge gemeinsam enthalten.

2.4 Demonstration der Simplex-Methode an Beispielen

Wir wollen die später zu einem Algorithmus ausgebaute Simplex-Methode an zwei Beispielen einführen. Obwohl $n = 2$, wollen wir auf eine graphische Veranschaulichung verzichten, da diese auf nicht verallgemeinerbaren Wegen zum Ziel führt.

Beispiel 2.4.1: $Z = \Leftrightarrow 3x_1 \Leftrightarrow 5x_2 \to \min$, wobei

$$\begin{array}{rrcrcl} \Leftrightarrow & x_1 & + & x_2 & \leq & 2 \\ & 2x_1 & \Leftrightarrow & 3x_2 & \leq & 3 \\ & 2x_1 & + & 3x_2 & \leq & 12 \end{array}$$

$$x_1 \geq 0,\ x_2 \geq 0.$$

Der Vorteil in diesem Beispiel ist, daß die rechten Seiten aller Ungleichungen nicht negativ sind. Wir erhalten daher mit $x_1 = 0$ und $x_2 = 0$ einen zulässigen Extrempunkt mit Wert 0. Wir formulieren nun unser Problem mit 3 (allgemein m) neuen Variablen um, wobei die neuen Variablen den Freiraum in den Ungleichungen ausfüllen. Eine äquivalente Formulierung unseres Problems ist

$$Z \to \min$$

$$Z + 3x_1 + 5x_2 = 0$$

$$\begin{array}{rrcrcrcrcrcl} \Leftrightarrow & x_1 & + & x_2 & + & u_1 & & & & & = & 2 \\ & 2x_1 & \Leftrightarrow & 3x_2 & & & + & u_2 & & & = & 3 \\ & 2x_1 & + & 3x_2 & & & & & + & u_3 & = & 12 \end{array}$$

$$x_1 \geq 0,\ x_2 \geq 0,\ u_1 \geq 0,\ u_2 \geq 0,\ u_3 \geq 0.$$

Unser sogenannter Basispunkt ist $(0, 0, 2, 3, 12)$ mit Z-Wert 0. Eine Basis besteht aus m Basisvariablen, die jeweils in genau einer Gleichung mit Faktor 1 vorkommen, hier u_1, u_2, u_3. Die anderen Variablen, hier x_1, x_2, heißen Nicht-Basisvariablen. Zulässige Basispunkte sind Extrempunkte. Für $x_1 = 0$ und $x_2 = 0$ erhalten wir drei linear unabhängige Gleichungen, da die Lösung des Gleichungssystems eindeutig ist. Wir können nun leicht prüfen, ob wir Z vermindern können. Dies ist durch Erhöhung von x_1 und/oder durch Erhöhung von x_2 möglich. Wir wählen x_2, da x_2 den größeren Vorfaktor hat und wir heuristisch auf möglichst große Verminderung von Z hoffen. Welche Basisvariable soll Platz für x_2 machen? Dazu überlegen wir uns, wie groß x_2 werden kann, wenn $x_1 = 0$ bleibt. Die erste Gleichung erzwingt, da $u_1 \geq 0$, $x_2 \leq 2$. Die zweite Gleichung erlaubt beliebig große Werte für x_2, und die dritte Gleichung erzwingt $x_2 \leq 4$. Zusammen genommen folgt $x_2 \leq 2$. Diese Bedingung folgt aus der ersten Gleichung. Daher soll u_1 die Basis verlassen. Aus dieser Gleichung folgt $x_2 = 2 + x_1 \Leftrightarrow u_1$. Diese Gleichung formen wir so um, daß x_2 mit Faktor 1 mit allen Variablen auf einer Seite steht. In allen anderen Gleichungen ersetzen wir x_2 durch obige Gleichung. Wir erhalten das folgende äquivalente Problem.

$$Z \to \min$$

$$Z + 8x_1 \Leftrightarrow 5u_1 = \Leftrightarrow 10$$

$$\begin{array}{rrrrrrrrrrrl}
\Leftrightarrow & x_1 & + & u_1 & + & x_2 & & & & = & 2 \\
\Leftrightarrow & x_1 & + & 3u_1 & & & + & u_2 & & = & 9 \\
5x_1 & \Leftrightarrow & 3u_1 & & & & & & + \quad u_3 & = & 6
\end{array}$$

$$x_1 \geq 0,\ x_2 \geq 0,\ u_1 \geq 0,\ u_2 \geq 0,\ u_3 \geq 0.$$

Da wir die Gleichung mit der strengsten Beschränkung für x_2 zum Basiswechsel benutzt haben, bleiben die rechten Seiten nicht negativ (allgemeiner Beweis später). Als Basispunkt erhalten wir $(0, 2, 0, 9, 6)$ mit dem besseren Z-Wert $\Leftrightarrow 10$. Eine Vergrößerung von u_1 würde Z vergrößern, aber eine Vergrößerung von x_1 kann Z vermindern. Nun geben, da $u_1 = 0$, die ersten beiden Gleichungen keine Beschränkung für x_1, wohl aber die dritte Gleichung, die $x_1 \leq 6/5$ erzwingt. Würde die dritte Gleichung auch keine Beschränkung liefern, könnte x_1 beliebig erhöht und Z beliebig vermindert werden. Wir ahnen, daß wir ein Kriterium erhalten, um festzustellen, daß ein Optimierungsproblem unbeschränkt ist. In der Basis wird nun u_3 durch x_1 ersetzt. Es gilt

$$x_1 = \tfrac{6}{5} + \tfrac{3}{5}u_1 \Leftrightarrow \tfrac{1}{5}u_3,$$

und wir erhalten

$$Z \to \min$$

$$Z \Leftrightarrow \tfrac{1}{5}u_1 \Leftrightarrow \tfrac{8}{5}u_3 = \Leftrightarrow 19\tfrac{3}{5}$$

$$\begin{array}{rrrrrrrrrrrl}
 & \frac{2}{5}u_1 & + & \frac{1}{5}u_3 & + & x_2 & & & & = & 3\frac{1}{5} \\
 & 2\frac{2}{5}u_1 & + & \frac{1}{5}u_3 & & & + & u_2 & & = & 10\frac{1}{5} \\
\Leftrightarrow & \frac{3}{5}u_1 & + & \frac{1}{5}u_3 & & & & & + \quad x_1 & = & 1\frac{1}{5}
\end{array}$$

$$x_1 \geq 0,\ x_2 \geq 0,\ u_1 \geq 0,\ u_2 \geq 0,\ u_3 \geq 0.$$

Als Basispunkt erhalten wir $(1.2, 3.2, 0, 10.2, 0)$ mit dem besseren Z-Wert $\Leftrightarrow 19.6$. Der Basispunkt ist zulässig. Aus der Gleichung für Z mit den Nebenbedingungen $u_1 \geq 0$ und $u_3 \geq 0$ folgt $Z \geq \Leftrightarrow 19.6$. Also ist $x_1 = 1.2$ und $x_2 = 3.2$ Lösung unseres Optimierungsproblems.

Beispiel 2.4.2: $Z = \Leftrightarrow 5x_1 \Leftrightarrow 2x_2 \to \min$, wobei

$$\begin{array}{rcrcrcl} \Leftrightarrow & 3x_1 & \Leftrightarrow & x_2 & \leq & \Leftrightarrow & 3 \\ \Leftrightarrow & 2x_1 & \Leftrightarrow & 3x_2 & \leq & \Leftrightarrow & 6 \\ & 2x_1 & + & x_2 & \leq & & 4 \end{array}$$

$$x_1 \geq 0,\ x_2 \geq 0.$$

Nach Umformulierung ergibt sich

$$Z \to \min$$

$$Z + 5x_1 + 2x_2 = 0$$

$$\begin{array}{rcrcrcrcrcl} \Leftrightarrow & 3x_1 & \Leftrightarrow & x_2 & + & u_1 & & & = & \Leftrightarrow & 3 \\ \Leftrightarrow & 2x_1 & \Leftrightarrow & 3x_2 & & & + & u_2 & = & \Leftrightarrow & 6 \\ & 2x_1 & + & x_2 & & & & & + \ u_3 \ = & & 4 \end{array}$$

$$x_1 \geq 0,\ x_2 \geq 0,\ u_1 \geq 0,\ u_2 \geq 0,\ u_3 \geq 0.$$

Der zugehörige Basispunkt $(0, 0, \Leftrightarrow 3, \Leftrightarrow 6, 4)$ ist nicht zulässig. Was ist zu tun? Die dritte Gleichung ist in gewünschter Form, da $u_3 = 4 \geq 0$. Die zweite Gleichung ist für $x_1 = x_2 = 0$ mit $u_2 = \Leftrightarrow 6$ nicht zulässig erfüllt. Wären alle Koeffizienten der anderen Variablen positiv, dann wäre die Gleichung nicht zulässig erfüllbar. So können wir prinzipiell nachweisen, daß die zulässige Menge eines Optimierungsproblems leer ist. Hier wählen wir eine Variable, o. B. d. A. x_1, mit negativem Koeffizienten. Die Variable x_1 soll in die Basis aufgenommen werden. Dabei soll die zweite Gleichung zulässig erfüllbar werden, ohne daß diese Eigenschaft der dritten Gleichung verloren geht. Es soll also u_2 oder u_3 die Basis verlassen. Bei Wahl von u_2 erhalten wir, da $x_1 = 3 \Leftrightarrow \frac{3}{2}x_2 + \frac{1}{2}u_2$, aus der zweiten und dritten Gleichung

$$\begin{array}{rcrcrcrcl} \frac{3}{2}x_2 & \Leftrightarrow & \frac{1}{2}u_2 & + & x_1 & & = & & 3 \\ \Leftrightarrow 2x_2 & + & u_2 & & & + & u_3 & = \Leftrightarrow & 2. \end{array}$$

Zwar hat die zweite Gleichung eine gute Form, aber die dritte nicht mehr. Bei Wahl von u_3 erhalten wir, da $x_1 = 2 \Leftrightarrow \frac{1}{2}x_2 \Leftrightarrow \frac{1}{2}u_3$,

$$\begin{array}{rcrcrcrcl} \Leftrightarrow 2x_2 & + & u_3 & + & u_2 & & = & \Leftrightarrow & 2 \\ \frac{1}{2}x_2 & + & \frac{1}{2}u_3 & & & + & x_1 & = & 2. \end{array}$$

Nun bleibt die letzte Gleichung gut, während die zweite Gleichung schlecht bleibt. Später werden wir nachweisen, daß wir für alle Gleichungen, die nicht oberhalb der untersten Gleichung mit negativer rechter Seite stehen, den Quotienten aus der rechten Seite und dem Koeffizienten der neuen Basisvariablen bilden sollten. Hier ergibt sich $\frac{-6}{-2} = 3$ und $\frac{4}{2} = 2$. Unter den positiven Werten sollten wir den kleinsten wählen, die zugehörige Basisvariable, hier u_3, sollte die Basis verlassen. Wir erhalten

$$Z \to \min$$

$$Z \Leftrightarrow \tfrac{1}{2}x_2 \Leftrightarrow \tfrac{5}{2}u_3 = \Leftrightarrow 10$$

$$\begin{array}{rcrcrcrcrcl} \frac{1}{2}x_2 & + & \frac{3}{2}u_3 & + & u_1 & & & & = & & 3 \\ \Leftrightarrow 2x_2 & + & u_3 & & & + & u_2 & & = & \Leftrightarrow & 2 \\ \frac{1}{2}x_2 & + & \frac{1}{2}u_3 & & & & & + \ x_1 \ = & & & 2 \end{array}$$

$$x_1 \geq 0,\ x_2 \geq 0,\ u_1 \geq 0,\ u_2 \geq 0,\ u_3 \geq 0.$$

Der neue Basispunkt $(2,0,3,\Leftrightarrow 2,0)$ ist unzulässig. In der vorletzten Gleichung ist die rechte Seite negativ. Die Variable x_2 wird neue Basisvariable, da sie in der vorletzten Gleichung die einzige mit negativem Vorzeichen ist. Wir bilden die oben beschriebenen Quotienten $\frac{-2}{-2} = 1$ und $\frac{2}{1/2} = 4$. Also verläßt u_2 die Basis. Wir erhalten

$$
\begin{array}{rcl}
 & Z \to \min & \\
 & Z \Leftrightarrow \frac{11}{4}u_3 \Leftrightarrow \frac{1}{4}u_2 = \Leftrightarrow \frac{19}{2} & \\
\frac{7}{4}u_3 + \frac{1}{4}u_2 + u_1 & = & \frac{5}{2} \\
\Leftrightarrow \frac{1}{2}u_3 \Leftrightarrow \frac{1}{2}u_2 + x_2 & = & 1 \\
\frac{3}{4}u_3 + \frac{1}{4}u_2 + x_1 & = & \frac{3}{2}
\end{array}
$$

$$x_1 \geq 0,\ x_2 \geq 0,\ u_1 \geq 0,\ u_2 \geq 0,\ u_3 \geq 0.$$

Der zugehörige Basispunkt $(1.5, 1, 2.5, 0, 0)$ mit Z-Wert $\Leftrightarrow 9.5$ ist zulässig. Da die Nicht-Basisvariablen in der Z-Gleichung negatives Vorzeichen haben, ist dieser Basispunkt Lösung des Optimierungsproblems.

2.5 Das Simplextableau und der Pivot-Schritt

Das Wort Simplex beschreibt in der Mathematik eine konvexe Menge der Dimension n mit genau $n+1$ Extrempunkten, also Dreiecke im $\mathbb{R}^2$, Tetraeder im $\mathbb{R}^3$, usw. Wenn wir hier einen zulässigen Basispunkt im $\mathbb{R}^n$ haben, haben wir n Kandidaten, die in die Basis eintreten können. Jeder Kandidat liefert einen möglichen neuen Basispunkt. Zusammen mit dem alten Basispunkt erhalten wir $n+1$ Punkte, die ein Simplex aufspannen. Wenn der alte Basispunkt für dieses Simplex optimal ist, ist er global optimal. Auf diese Weise hat der Algorithmus von Dantzig seinen Namen erhalten.

Wir bringen nun Probleme der Linearen Optimierung in eine rechnergerechte Form, wobei wir, wenn keine Verwechslungen möglich sind, x und x^T identifizieren. Wir können unsere Optimierungsaufgabe folgendermaßen darstellen.

$$
\begin{array}{rcl}
Z \to & \min & \\
cx + 0 & = & Z \\
Ax \Leftrightarrow b & = & \Leftrightarrow u
\end{array}
$$

$$x \geq 0,\ u \geq 0.$$

Dabei sind die Zeilen $Z \to min$ und $x \geq 0$, $u \geq 0$ redundant.

Definition 2.5.1: Zu einem Problem der Linearen Optimierung gehört das Simplextableau

x_1	x_2	$\cdots$	x_n		
a_{11}	a_{12}	$\cdots$	a_{1n}	$\Leftrightarrow b_1$	$\Leftrightarrow u_1$
a_{21}	a_{22}	$\cdots$	a_{2n}	$\Leftrightarrow b_2$	$\Leftrightarrow u_2$
$\vdots$	$\vdots$	$\cdots$	$\vdots$	$\vdots$	$\vdots$
a_{m1}	a_{m2}	$\cdots$	a_{mn}	$\Leftrightarrow b_m$	$\Leftrightarrow u_m$
c_1	c_2	$\cdots$	c_n	0	Z

Definition 2.5.2: Zwei Simplextableaus heißen äquivalent, wenn die zugehörigen Gleichungssysteme äquivalent sind, d. h. die gleichen Lösungsmengen haben.

Allgemein steht oberhalb der n Spalten und rechts von den m Zeilen eine Permutation der Variablen $(x_1, \ldots, x_n, u_1, \ldots, u_m)$. Die Spaltenvariablen gehören nicht zur Basis, d. h. sie werden im Basispunkt gleich 0 gesetzt, die Zeilenvariablen sind die negierten Basisvariablen. Der Basispunkt entsteht durch Negation der letzten Spalte, wobei in der untersten Zeile dann der zugehörige Z-Wert steht. Ansonsten stehen in der letzten Zeile die Koeffizienten der umgeformten Zielfunktion. Der folgende Satz folgt direkt als Verallgemeinerung unserer Diskussion in Kapitel 2.4.

Satz 2.5.3: *Sind in einem Simplextableau die b-Werte der letzten Spalte nicht negativ, ist der zugehörige Basispunkt zulässig. Sind darüber hinaus die c-Werte der letzten Zeile nicht negativ, ist der Basispunkt optimal.*

Satz 2.5.4: *Wenn ein Simplextableau zur Basis $x_1, \ldots, x_k, u_{k+1}, \ldots, u_m$ äquivalent zu dem zu einem linearen Optimierungsproblem gehörigen Simplextableau ist, sind die Hyperebenen $u_1 = 0, \ldots, u_k = 0$, $x_{k+1} = 0, \ldots, x_n = 0$ linear unabhängig. Der zugehörige Basispunkt ist also, falls er zulässig ist, Extrempunkt der zulässigen Menge.*

Der Satz gilt natürlich allgemeiner, wir haben zur einfacheren Beschreibung nur eine günstige Numerierung gewählt.

Beweis von Satz 2.5.4: Wir schreiben die Hyperebenen als Gleichungen in den ursprünglichen x-Variablen. Aus $u_i = 0$ wird $a_{i1}x_1 + \ldots + a_{in}x_n = b_i$. Die Gleichungen $x_{k+j} = 0$ haben bereits die gewünschte Form. Die Hyperebenen sind genau dann linear unabhängig, wenn die zugehörige Koeffizientenmatrix invertierbar ist, d. h. wenn ihre Determinante von 0 verschieden ist. Die Koeffizientenmatrix hat folgendes Aussehen:

$$\begin{pmatrix} a_{11} & \cdots & a_{1k} & a_{1,k+1} & \cdots & a_{1n} \\ \vdots & & \vdots & \vdots & & \vdots \\ a_{k1} & \cdots & a_{kk} & a_{k,k+1} & \cdots & a_{kn} \\ & & & 1 & \cdots & 0 \\ & 0 & & \vdots & \ddots & \vdots \\ & & & 0 & \cdots & 1 \end{pmatrix}$$

Ihre Determinante ist genau dann von 0 verschieden, wenn dies für die linke obere Untermatrix

$$A^{(k)} = \begin{pmatrix} a_{11} & \cdots & a_{1k} \\ \vdots & & \vdots \\ a_{k1} & \cdots & a_{kk} \end{pmatrix}$$

gilt. Mit dem oberen Index (k) wollen wir andeuten, daß wir von einer Matrix nur die ersten k Zeilen und Spalten betrachten. Es sei A' die Koeffizientenmatrix zum Tableau zur Basis $x_1, \ldots, x_k, u_{k+1}, \ldots, u_m$. Für $x_{k+1} = \ldots = x_n = 0$ erhalten wir hieraus

$$\Leftrightarrow x^{(k)} = A'^{(k)} u^{(k)} \Leftrightarrow b'^{(k)}.$$

Aus dem Originaltableau folgt

$$\Leftrightarrow u^{(k)} = A^{(k)} x^{(k)} \Leftrightarrow b^{(k)}.$$

Nach Einsetzen in die obere Gleichung folgt

$$\Leftrightarrow x^{(k)} = A'^{(k)}(\Leftrightarrow A^{(k)} x^{(k)} + b^{(k)}) \Leftrightarrow b'^{(k)} \text{ und}$$

$$x^{(k)} = A'^{(k)}A^{(k)}x^{(k)} \Leftrightarrow A'^{(k)}b^{(k)} + b'^{(k)}.$$

Da die letzte Gleichung auch für $x^{(k)} = 0^{(k)}$ (Nullvektor aus k Nullen) gilt, ist $\Leftrightarrow A'^{(k)}b^{(k)} + b'^{(k)} = 0^{(k)}$. Also gilt $x^{(k)} = A'^{(k)}A^{(k)}x^{(k)}$ für alle $x^{(k)}$, d. h. $A'^{(k)}$ ist die zu $A^{(k)}$ inverse Matrix. □

Wir wollen nun die Folgen eines Basiswechsels am Simplextableau nachvollziehen. Es seien $r_1, \ldots, r_n$ die Nicht-Basisvariablen und $s_1, \ldots, s_m$ die Basisvariablen. Das Simplextableau habe folgendes Aussehen:

r_1	$\cdots$	r_k	$\cdots$	r_n		
a_{11}	$\cdots$	a_{1k}	$\cdots$	a_{1n}	$\Leftrightarrow b_1$	$\Leftrightarrow s_1$
$\vdots$		$\vdots$		$\vdots$	$\vdots$	$\vdots$
a_{i1}	$\cdots$	a_{ik}	$\cdots$	a_{in}	$\Leftrightarrow b_i$	$\Leftrightarrow s_i$
$\vdots$		$\vdots$		$\vdots$	$\vdots$	$\vdots$
a_{m1}	$\cdots$	a_{mk}	$\cdots$	a_{mn}	$\Leftrightarrow b_m$	$\Leftrightarrow s_m$
c_1	$\cdots$	c_k	$\cdots$	c_n	δ	Z

Definition 2.5.5: Beim Basiswechsel $r_k \leftrightarrow s_i$ heißt das Element a_{ik} Pivotelement.

Im Englischen steht pivot für Drehpunkt. Im Simplextableau ist a_{ik} der Drehpunkt, um den sich der Basiswechsel dreht. Wir führen Basiswechsel nur durch, wenn $a_{ik} \neq 0$ ist. Dann folgt:

$$\Leftrightarrow r_k = \frac{a_{i1}}{a_{ik}}r_1 + \ldots + \frac{a_{i,k-1}}{a_{ik}}r_{k-1} + \frac{1}{a_{ik}}s_i + \frac{a_{i,k+1}}{a_{ik}}r_{k+1} + \ldots + \frac{a_{in}}{a_{ik}}r_n \Leftrightarrow \frac{b_i}{a_{ik}}.$$

In der Matrix wird das Pivotelement a_{ik} durch $\frac{1}{a_{ik}}$ ersetzt. Alle anderen Elemente der Pivotzeile werden durch a_{ik} dividiert. Betrachten wir nun für $j \neq i$ die j-te Zeile des Tableaus:

$$a_{j1}r_1 + \ldots + a_{j,k-1}r_{k-1} + a_{jk}r_k + a_{j,k+1}r_{k+1} + \ldots + a_{jn}r_n \Leftrightarrow b_j = \Leftrightarrow s_j.$$

Wir setzen nun für r_k den obigen Wert ein und fassen zusammen. Es ergibt sich

$$(a_{j1} \Leftrightarrow a_{jk}\frac{a_{i1}}{a_{ik}})r_1 + \ldots + (a_{j,k-1} \Leftrightarrow a_{jk}\frac{a_{i,k-1}}{a_{ik}})r_{k-1} \Leftrightarrow \frac{a_{jk}}{a_{ik}}s_i$$
$$+(a_{j,k+1} \Leftrightarrow a_{jk}\frac{a_{i,k+1}}{a_{ik}})r_{k+1} + \ldots + (a_{jn} \Leftrightarrow a_{jk}\frac{a_{in}}{a_{ik}})r_n \Leftrightarrow (b_j \Leftrightarrow a_{jk}\frac{b_i}{a_{ik}}) = \Leftrightarrow s_j.$$

Die letzte Zeile des Tableaus wird analog umgeformt. Wir fassen die Auswirkungen des Basiswechsels $r_k \leftrightarrow s_i$ mit einem Pivotelement $a_{ik} \neq 0$ zusammen.

Pivotelement: $a_{ik} \rightarrow \frac{1}{a_{ik}}$
Pivotzeile $j \neq k$: $a_{ij} \rightarrow \frac{a_{ij}}{a_{ik}}, \Leftrightarrow b_i \rightarrow \frac{-b_i}{a_{ik}}$.
Pivotspalte $i \neq j$: $a_{jk} \rightarrow \Leftrightarrow\frac{a_{jk}}{a_{ik}}, c_k \rightarrow \Leftrightarrow\frac{c_k}{a_{ik}}$.
Übrige Elemente, $i \neq j$, $k \neq l$: $a_{jl} \rightarrow a_{jl} \Leftrightarrow a_{jk}\frac{a_{il}}{a_{ik}}, \Leftrightarrow b_j \rightarrow \Leftrightarrow b_j \Leftrightarrow a_{jk}\frac{-b_i}{a_{ik}}, c_l \rightarrow c_l \Leftrightarrow c_k\frac{a_{il}}{a_{ik}}, \delta \rightarrow \delta \Leftrightarrow c_k\frac{-b_i}{a_{ik}}$.
Dies ist etwas viel, um sich alles zu merken. Wir können eine kürzere Merkformel angeben. Sei p das Pivotelement, q ein anderes Element der Pivotzeile, r ein anderes Element der Pivotspalte und s das Element, das die Zeile mit r und die Spalte mit q gemeinsam hat. Dann hat der Basiswechsel folgende Wirkung

$$\begin{bmatrix} p & q \\ r & s \end{bmatrix} \rightarrow \begin{bmatrix} \frac{1}{p} & \frac{q}{p} \\ \Leftrightarrow\frac{r}{p} & s \Leftrightarrow \frac{rq}{p} \end{bmatrix}$$

2.6 Der Simplex-Algorithmus

Um den Simplex-Algorithmus zu vervollständigen, müssen wir beschreiben, wie das Pivotelement gewählt wird. Danach sind Korrektheit und Endlichkeit zu zeigen.

Algorithmus 2.6.1 (Simplex-Algorithmus): Wir arbeiten auf dem Simplextableau (s. Def. 2.5.1).

1. Phase
Die letzte Spalte ($\Leftrightarrow b$ Elemente) enthält mindestens eine positive Zahl (d. h. der aktuelle Basispunkt ist nicht zulässig). $G := \{l \mid b_l \geq 0\}$ sei die Menge der „guten" Zeilen und $S := \{l \mid b_l < 0\}$ die Menge der „schlechten" Zeilen. Sei $s := \max S$.

1. Unterfall: $a_{s1}, \ldots, a_{sn} \geq 0$. STOP. Ausgabe: „Die zulässige Menge des Problems ist leer".

2. Unterfall: Es kann ein k_0 mit $a_{sk_0} < 0$ gewählt werden. Wähle die Pivotzeile p so, daß

$$\frac{b_p}{a_{pk_0}} = \min \left\{ \frac{b_j}{a_{jk_0}} \mid j \in G \text{ und } a_{jk_0} > 0 \right\}$$

ist. Falls die betrachtete Menge leer ist, wähle $p = s$. Führe einen Basiswechsel um das Pivotelement a_{pk_0} durch. Fahre mit dem nächsten Simplextableau analog fort.

2. Phase
Die letzte Spalte enthält keine positive Zahl (d. h. der zugehörige Basispunkt ist zulässig). Berechne ein k_0, so daß $c_{k_0} = \min\{c_1, \ldots, c_n\}$.

1. Unterfall: $c_{k_0} \geq 0$. Der aktuelle Basispunkt wird als optimale Lösung ausgegeben. STOP.

2. Unterfall: $c_{k_0} < 0$. Falls $a_{ik_0} \leq 0$ für alle i, kann die Zielfunktion beliebig kleine Werte annehmen und nimmt ihr Minimum auf der zulässigen Menge nicht an. Der Algorithmus meldet dies. STOP.

Ansonsten wähle die Pivotzeile p wie im zweiten Unterfall der ersten Phase. (Hier ist G die Menge aller Zeilen und die Menge, über die das Minimum berechnet wird, nicht leer.) Führe einen Basiswechsel um das Pivotelement a_{pk_0} durch. Fahre mit dem nächsten Simplextableau fort.

Nur der letzte Schritt bedarf eines kurzen Kommentars. Wir haben uns dafür entschieden, die k_0-te Nichtbasisvariable in die Basis aufzunehmen. Alle anderen Nichtbasisvariablen bleiben Nichtbasisvariablen. Für den Basispunkt haben sie den Wert 0. Aus der i-ten Gleichung wird dann

$$a_{ik_0} r_{k_0} + s_i = b_i \text{ mit } \quad b_i \geq 0.$$

Ist $a_{ik_0} \leq 0$, kann r_{k_0} in dieser Gleichung beliebig wachsen. Da $c_{k_0} < 0$, würde dann $Z \to \Leftrightarrow\infty$ konvergieren. Ist $a_{ik_0} > 0$, folgt $r_{k_0} = (b_i \Leftrightarrow s_i)/a_{ik_0} \leq b_i/a_{ik_0}$.

Wir wählen also die stärkste Beschränkung an r_{k_0}. Wir fassen unser Wissen über den Simplex-Algorithmus zusammen.

Satz 2.6.2: *Der Simplex-Algorithmus ist partiell korrekt, d.h. er liefert das korrekte Resultat, wenn er stoppt. Jeder Basiswechsel ist in Zeit $O(nm)$ durchführbar.*

Der Simplex-Algorithmus 2.6.1 ist in der angegebenen Form nicht endlich, da er mit einem Problem nicht umzugehen weiß.

Definition 2.6.3: Ein lineares Optimierungsproblem heißt degeneriert, wenn es unter den $m + n$ zum Problem gehörenden Hyperebenen $n + 1$ Hyperebenen mit nicht leerem Schnitt gibt.

Degenerierte Probleme lassen sich an Simplextableaus erkennen, die in der b-Spalte eine 0 haben. Umgekehrt gibt es für degenerierte Probleme immer ein äquivalentes Tableau, das in der b-Spalte eine 0 hat. Falls $\Leftrightarrow b_i = 0$, liegt der aktuelle Basispunkt auf den $n + 1$ Hyperebenen $r_1 = 0, \ldots, r_n = 0, s_i = 0$.

Umgekehrt muß nur die Basis $r_1, \ldots, r_n$ gewählt werden. Bevor wir diskutieren, wie wir mit degenerierten Problemen umgehen, zeigen wir die Endlichkeit des Simplex-Algorithmus für nicht degenerierte Probleme.

Satz 2.6.4: *Der Simplex-Algorithmus ist für nicht degenerierte Probleme endlich.*

Beweis: Wir zeigen zunächst: Phase 1 findet in endlicher Zeit einen zulässigen Basispunkt oder erkennt, daß die zulässige Menge leer ist. Wenn noch kein zulässiger Basispunkt gefunden worden ist, ist die Menge S der schlechten Zeilen nicht leer. Wir zeigen, daß nach Ausführung eines Basiswechsels für das neue Simplextableau (Elemente a' statt a, usw.) folgendes gilt:

a) eine Zeile g, die vorher gut war, ist immer noch gut, also $b_g \geq 0 \Rightarrow b'_g \geq 0$.

b) für die betrachtete schlechte Zeile s gilt $b'_s > b_s$.

Die Pivotzeile p ist nach dem Basiswechsel auf jeden Fall gut. Nach Wahl von p gilt nämlich (auch, wenn $p = s$ ist): $\Leftarrow b'_p = \frac{-b_p}{a_{pk_0}} \leq 0$. Da das vorliegende Problem nicht degeneriert ist, gilt sogar $\frac{-b_p}{a_{pk_0}} < 0$. Sei $i \neq p$ eine beliebige Zeile. Es gilt

$$\Leftarrow b'_i = \Leftarrow b_i + \underbrace{a_{ik_0} \cdot \frac{b_p}{a_{pk_0}}}_{c_i}.$$

Betrachten wir zunächst das Verhalten der s–ten Zeile. Wenn als Pivotzeile $p = s$ gewählt wird, dann ist mit der Überlegung von gerade die s–te Zeile hinterher auf jeden Fall gut, insbesondere gilt auch Behauptung b). Ansonsten ist $\Leftarrow b'_s = \Leftarrow b_s + c_s$, und da $a_{sk_0} < 0, b_p/a_{pk_0} > 0$ ist, ist $c_s < 0$ und daher $\Leftarrow b'_s < \Leftarrow b_s$ und auch hier gilt Behauptung b). Zum Nachweis von Behauptung a) betrachten wir eine gute Zeile g. Wenn $p = g$ gewählt wurde, ist die Zeile g auch hinterher gut. Ansonsten ist $p \neq g$ und wir unterscheiden zwei Fälle:

1. Fall: $a_{gk_0} \leq 0$. Dann ist $c_g \leq 0$ und somit $\Leftarrow b'_g = \Leftarrow b_g + c_g \leq \Leftarrow b_g \leq 0$.

2. Fall: $a_{gk_0} > 0$. Dann wurde bei der Minimumsbildung zur Berechnung von p auch die gute Zeile g betrachtet und nach Wahl von p ist $\Leftarrow b_g/a_{gk_0} \leq \Leftarrow b_p/a_{pk_0}$. Also ist

$$\Leftarrow b'_g = \Leftarrow b_g + \frac{a_{gk_0} b_p}{a_{pk_0}} = a_{gk_0}(\Leftarrow \frac{b_g}{a_{gk_0}} + \frac{b_p}{a_{pk_0}}) \leq 0.$$

Damit ist Behauptung a) gezeigt.

Da es nur endlich viele Basispunkte gibt, wird nach endlich vielen Schritten die s–te Zeile gut und somit die Menge G der guten Zeilen größer. Also gilt: Entweder stoppt der Algorithmus in der ersten Phase mit dem Nachweis, daß die zulässige Menge leer ist, oder er berechnet einen zulässigen Basispunkt.

Wir kommen zur zweiten Phase. Diese wird gestartet, nachdem der Algorithmus einen zulässigen Basispunkt gefunden hat. Wir zeigen nun, daß in jedem Schritt von Phase 2 zu einem Basispunkt gewechselt wird, der zulässig ist und der einen echt kleineren Z-Wert hat.

In der zweiten Phase sind alle Zeilen gut und ein Basiswechsel wird nur durchgeführt, wenn ein $a_{ik_0} > 0$ ist für ein i. Die betrachtete Menge, über die das Minimum berechnet wird, ist also nicht leer. Nun folgt aber aus Behauptung a), daß der neue Basispunkt zulässig ist, denn alle Zeilen sind gut und bleiben gut. Es bleibt zu zeigen, daß $\delta' < \delta$ ist. Es ist $\delta' = \delta + c_{k_0} b_p / a_{pk_0}$.

Nach Wahl von k_0 ist $c_{k_0} < 0$ und $a_{pk_0} > 0$. Wegen der Nichtdegeneriertheit des Problems ist $b_p > 0$, also ist $\delta' < \delta$.

Da es nur endlich viele Basispunkte gibt, gelangt der Algorithmus nach endlich vielen Schritten zu einer optimalen Lösung oder gibt aus, daß die Zielfunktion auf der zulässigen Menge beliebig kleine Werte annehmen kann. □

Bei degenerierten Problemen können wir Glück haben, und nach endlich vielen Schritten stoppt der Simplex-Algorithmus. Bei etwas Pech können wir in eine unendliche Schleife geraten. Wenn wir bei jedem Basispunkt, den wir wiederholt aufsuchen, die möglichen Aktionen sich abwechseln lassen, sichern wir, daß wir einen optimalen Basispunkt nach endlicher Zeit erreichen. Dieses Vorgehen erfordert einen gewaltigen Overhead an Speicherplatz. In der Praxis wird in degenerierten Basispunkten der nächste Basiswechsel nach der Gleichverteilung ausgewürfelt. Dies sichert, daß die erwartete Rechenzeit endlich ist.

Was gilt für die worst case Rechenzeit selbst bei nicht degenerierten Problemen? Sie ist exponentiell in der Tableaugröße. Es kostet allerdings eine größere Anstrengung (der wir uns nicht unterziehen wollen), um derart schlechte Beispiele zu konstruieren. Der Simplex-Algorithmus ist einer der wenigen („natürlichen") Algorithmen mit exponentieller worst case Rechenzeit, der in der Praxis sehr schnell ist. Er ist bis heute aus praktischer Sicht unübertroffen.

2.7 Heuristische Ideen zur Verbesserung des Simplex-Algorithmus

Wir können das typische Verhalten des Simplex-Algorithmus nicht analysieren. Einerseits ist unklar, was eine typische Eingabe für das Problem der Linearen Optimierung ist, andererseits kennen wir keine Methoden, um Eingaben sehr schnell als einfach oder schwierig zu klassifizieren. Daher sind Verbesserungen stets heuristische Verbesserungen. Wir erwarten oder hoffen, daß sie den Simplex-Algorithmus beschleunigen. Dies kann durch Experimente untermauert werden.

Die erste Verbesserung betrifft Nebenbedingungen, die durch Gleichungen gegeben sind. Bisher haben wir sie durch $\leq$- und $\geq$-Ungleichungen ersetzt, wodurch das Problem automatisch degeneriert wird. Statt dessen können wir die Gleichung nach einer Variablen auflösen, in allen anderen Ungleichungen diese Variable ersetzen und benutzen die Gleichung direkt, d. h. die ausgewählte Variable wird zu Beginn Basisvariable. Gegenüber dem bisherigen Vorgehen sparen wir eine Nebenbedingung, zwei Variablen und verhindern die Entstehung von Degenerationen.

Das zweite Problem ist die Lösung „ähnlicher" Optimierungsprobleme. Wir nehmen an, daß wir das Problem $Z = c^T x \to \min$ mit $Ax = b$ (u-Variablen werden als weitere x-Variablen bezeichnet) und $x \geq 0$ gelöst haben. Die Lösung sei x^* zur Basis B. Wenn wir nun ein neues Problem lösen wollen, das nur eine Variable x_{n+1} und somit eine Spalte im Ausgangstableau mehr hat, sollte es möglich sein, zu vermeiden, ganz von vorne anzufangen. Mit $(x^*, 0)$ (der Wert 0 steht für den Wert von x_{n+1}) erhalten wir sofort einen zulässigen Basispunkt des neuen Problems. Es ist naheliegend zu vermuten, daß $(x^*, 0)$ der bessere Startpunkt als der übliche Basispunkt ist. Wenn wir den Lösungsweg für das alte Problem abgespeichert haben, können wir diesen Weg für die neue Spalte nachvollziehen. Dies bedeutet jedoch eine arge Verschwendung von Speicherplatz.

Besser ist folgendes Vorgehen. Wir kennen für das alte Problem die Startbasis und die Basis für den Lösungsvektor. Der Weg von der Startbasis zur Lösungsbasis ist aber unerheblich. Es genügt nun, einen kürzesten Weg von der Startbasis zur Lösungsbasis nachzuvollziehen. Offensichtlich reichen $\min\{m, n\}$ Basiswechselschritte immer aus. Die Gesamtrechenzeit ist also durch $O(mn \min\{m, n\})$ abschätzbar.

Schließlich können wir aber auch auf den Beweis von Satz 2.5.4 zurückgreifen. O. B. d. A. (Umnumerierung) nehmen wir an, daß $x_1, \ldots, x_k$ gegen $u_1, \ldots, u_k$ ausgetauscht werden müssen. Im Beweis von Satz 2.5.4 haben wir gezeigt, daß die Inverse von $A^{(k)}$ (der oberen linken $k \times k$-Untermatrix des Tableaus) hinterher oben links im Tableau steht. Was geschieht mit der neuen $(n+1)$-ten Spalte? Sei $(a^{n+1})^{(k)}$ der Anfang der Länge k in dieser Spalte. Mit B bezeichnen wir die Matrix aus den Spalten $k+1, \ldots, n$ und Zeilen $1, \ldots, k$ von A. Sei $x' = (x_{k+1}, \ldots, x_n)$. Dann gilt im Starttableau

$$A^{(k)} x^{(k)} + B x' + (a^{n+1})^{(k)} x_{n+1} \Leftrightarrow b^{(k)} = \Leftrightarrow u^{(k)}$$

Nach Multiplikation mit $(A^{(k)})^{-1}$ folgt

$$(A^{(k)})^{-1}u^{(k)} + (A^{(k)})^{-1}Bx' + (A^{(k)})^{-1}(a^{n+1})^{(k)}x_{n+1} \Leftrightarrow (A^{(k)})^{-1}b^{(k)} = \Leftrightarrow x^{(k)}.$$

Dabei können wir $(A^{(k)})^{-1}$ im Lösungstableau für das alte Problem ablesen und müssen nur $(A^{(k)})^{-1}(a^{n+1})^{(k)}$ neu berechnen. Dieses Matrix-Vektorprodukt kann auf die übliche Weise in Zeit $O(k^2)$ berechnet werden.

Es bleibt zu überlegen, wie sich die restlichen Elemente der $(n+1)$-ten Spalte entwickeln, sowohl die a-Elemente als auch c_{n+1}. Diese Werte stehen nie in der Pivotzeile. Sie werden also stets nach der Regel $s \to s \Leftrightarrow rq/p$ behandelt. Für c_{n+1} (analog für die anderen Werte) ergibt sich als neuer Wert

$$c_{n+1} \Leftrightarrow \sum_{1 \le i \le k} (a^{n+1}_{\text{neu}})_i \, c_i.$$

Dies folgt für $k = 1$ direkt und für $k > 1$ durch Induktionsbeweis (Übungsaufgabe). Wir haben also Methoden, um weitere Variablen effizient hinzuzufügen und dann mit einem heuristisch guten Basispunkt den Simplex-Algorithmus starten zu können.

Bisher haben wir unbeschränkte Variablen x als Differenz $y \Leftrightarrow z$ mit $y \ge 0$ und $z \ge 0$ modelliert. Auch diese Erhöhung der Variablenzahl ist nicht notwendig. Wir arbeiten nun auch mit unbeschränkten Variablen. Bei der Wahl der Pivotspalte in der zweiten Phase des Algorithmus 2.6.1 kommen bei den vorzeichenbeschränkten Variablen nur die mit $c_k < 0$ in Frage, bei den unbeschränkten alle mit $c_k \neq 0$. Wir können ja die Variable in die gewünschte Richtung verändern und Z vermindern. Wenn nun keine Variable gewählt werden kann, ist der gefundene Basispunkt optimal. Bei der Wahl der Pivotzeile muß der Fall x_i unbeschränkt und zugehöriger c-Wert positiv nur so verändert werden, daß x_i nun negativ wird, wobei der Basispunkt zulässig bleiben muß (Übungsaufgabe). In der ersten Phase (Suche nach einem zulässigen Basispunkt) können wir das Verfahren im wesentlichen beibehalten, wenn wir beachten, daß manche Variablen auch negativ sein können. Die zugehörigen Gleichungen müssen bei der Wahl der Pivotzeile und Pivotspalte nicht berücksichtigt werden (Übungsaufgabe).

Untere Schranken, $x_i \ge \alpha_i$, können durch Ersetzung von x_i durch $x_i' := x_i \Leftrightarrow \alpha_i$ mit $x_i' \ge 0$ leicht ersetzt werden, wenn $\alpha_i > 0$ ist. Ansonsten kann die Bedingung wegfallen. Müssen wir obere Schranken $x_i \le \alpha_i$ für $\alpha_i > 0$ als Nebenbedingungen verwalten? Auch hier deuten wir nur an, wie wir den Simplex-Algorithmus modifizieren können, so daß er mit Variablen x_i, für die $0 \le x_i \le \alpha_i$ vorausgesetzt ist, direkt fertig wird. Die Ausarbeitung der Ideen wird als Übungsaufgabe gestellt. Wir erweitern den Begriff des Basispunktes. Für die beidseitig beschränkten Variablen, die nicht in der Basis sind, darf der Wert 0 oder der Wert α_i festgesetzt sein. Ein erweiterter Basispunkt ist optimal, wenn der Wert der Zielfunktion durch keine erlaubte Änderung einer Nicht-Basisvariablen vermindert werden kann. Das Problem ist unbeschränkt, wenn eine Variable beliebig wachsen darf und dabei den Wert der Zielfunktion beliebig verkleinert. Daß die zulässige Menge leer ist, kann auf die gewohnte Weise festgestellt werden. Wir kommen nun auf die zwei Phasen des Simplex-Algorithmus zu sprechen.

Bei der Suche nach einem zulässigen erweiterten Basispunkt kann nun ein Basispunkt unzulässig sein, weil eine Variable ihre obere Schranke überschritten hat. Wir konzentrieren uns wieder auf die unterste „schlechte“ Gleichung und „verbessern“ diese, ohne die darunter stehenden Gleichungen unzulässig zu machen.

Bei der Verbesserung von Basispunkten ist zu beachten, in welche Richtung die beidseitig beschränkten Variablen, die in der Basis sind, verändert werden dürfen. Wir führen die Änderung der Variablen, die in die Basis kommt, nur soweit durch, daß die andere Schranke nicht verletzt wird. Als Variable, die die Basis verläßt, müssen wir dabei eine wählen, die die Änderung der neuen Basisvariablen minimiert. So kann gesichert werden, daß der neue erweiterte Basispunkt zulässig ist und sich der Wert der Zielfunktion vermindert.

2.8 Das Dualitätsprinzip der Linearen Optimierung

Aus der Vorlesung „Rechnerstrukturen“ ist die duale Funktion einer Booleschen Funktion bekannt. Sie hat analoge Eigenschaften zu der ursprünglichen, d. h. der primalen Funktion, und die duale Funktion zu der dualen Funktion ist wieder die primale Funktion. Der stets übliche Wechsel zwischen der Untersuchung der primalen und der dualen Funktion erleichtert viele Betrachtungen.

Definition 2.8.1: Das duale Problem zum (primalen) Problem $Z = c^T x \to \min$, $Ax \geq b$, $x \geq 0$ ist das Problem $Z' = b^T y \to \max$, $A^T y \leq c$, $y \geq 0$. (Wir beachten, daß wir im primalen Problem nun die Nebenbedingungen in $\geq$-Form gebracht haben. Der Grund wird im Beweis von Satz 2.8.3 deutlich.)

Zunächst zeigen wir, daß wir die Bezeichnung duales Problem zu recht gewählt haben.

Satz 2.8.2: *Sei P ein primales Problem der Linearen Optimierung, P' dual zu P und P'' dual zu P'. Dann sind P und P'' äquivalent.*

Beweis: P' ist gegeben durch $Z' = b^T y \to \max$, $A^T y \leq c$, $y \geq 0$. Dies ist äquivalent zu $\Leftrightarrow Z' = (\Leftrightarrow b)^T y \to \min$, $(\Leftrightarrow A)^T y \geq \Leftrightarrow c$, $y \geq 0$. Damit ist P'' definiert durch $Z'' = (\Leftrightarrow c)^T z \to \max$, $(\Leftrightarrow A^T)^T z \leq \Leftrightarrow b$, $z \geq 0$. Dies ist äquivalent zu $Z^* = c^T z \to \min$, $\Leftrightarrow Az \leq \Leftrightarrow b$ oder $Az \geq b$, $z \geq 0$, und damit zu P. □

Wir bauen zunächst die Dualitätstheorie für die Lineare Optimierung auf und zeigen dann, welche Anwendungen sie hat. In Zukunft sei P das primale und P' das duale Problem.

Satz 2.8.3: *Sei x_0 zulässig für P und y_0 zulässig für P'. Dann gilt $Z(x_0) \geq Z'(y_0)$.*

Beweis: Wir wenden die Nebenbedingungen für die Probleme P und P' an und erhalten

$$Z(x_0) = c^T x_0 \geq (A^T y_0)^T x_0 = y_0^T A x_0 \geq y_0^T b = Z'(y_0).$$

□

Korollar 2.8.4: *Wenn die Zielfunktion des primalen Problems nicht nach unten beschränkt ist, ist die zulässige Menge des dualen Problems leer.*

Beweis: Jeder zulässige Punkt y_0 für das duale Problem liefert mit $Z'(y_0)$ nach Satz 2.8.3 eine untere Schranke für die Zielfunktion des primalen Problems. □

Satz 2.8.5 (Dualitätstheorem): *P hat genau dann eine Lösung, wenn P' eine Lösung hat. Wenn x_0 Lösung von P und y_0 Lösung von P' ist, gilt $Z(x_0) = Z'(y_0)$. Aus dem Lösungstableau für P kann eine Lösung für P' direkt abgelesen werden (und umgekehrt).*

Beweis: Wir beschreiben P und P' in der für Tableaus üblichen Form.

$$\begin{array}{lllll} P: & Z = c^T x \to \min, & (\Leftrightarrow A)x \leq \Leftrightarrow b, & x \geq 0. \\ P': & \Leftrightarrow Z' = (\Leftrightarrow b)^T y \to \min, & A^T y \leq c, & y \geq 0. \end{array}$$

Die Anfangstableaus haben folgendes Aussehen.

x_1	$\cdots$	x_k	$\cdots$	x_n		
$\Leftrightarrow a_{11}$	$\cdots$	$\Leftrightarrow a_{1k}$	$\cdots$	$\Leftrightarrow a_{1n}$	b_1	$\Leftrightarrow u_1$
$\vdots$		$\vdots$		$\vdots$	$\vdots$	$\vdots$
$\Leftrightarrow a_{i1}$	$\cdots$	$\Leftrightarrow a_{ik}$	$\cdots$	$\Leftrightarrow a_{in}$	b_i	$\Leftrightarrow u_i$
$\vdots$		$\vdots$		$\vdots$	$\vdots$	$\vdots$
$\Leftrightarrow a_{m1}$	$\cdots$	$\Leftrightarrow a_{mk}$	$\cdots$	$\Leftrightarrow a_{mn}$	b_m	$\Leftrightarrow u_m$
c_1	$\cdots$	c_k	$\cdots$	c_n	$\delta = 0$	Z

y_1	$\cdots$	y_i	$\cdots$	y_m		
a_{11}	$\cdots$	a_{i1}	$\cdots$	a_{m1}	$\Leftrightarrow c_1$	$\Leftrightarrow v_1$
$\vdots$		$\vdots$		$\vdots$	$\vdots$	$\vdots$
a_{1k}	$\cdots$	a_{ik}	$\cdots$	a_{mk}	$\Leftrightarrow c_k$	$\Leftrightarrow v_k$
$\vdots$		$\vdots$		$\vdots$	$\vdots$	$\vdots$
a_{1n}	$\cdots$	a_{in}	$\cdots$	a_{mn}	$\Leftrightarrow c_n$	$\Leftrightarrow v_n$
$\Leftrightarrow b_1$	$\cdots$	$\Leftrightarrow b_i$	$\cdots$	$\Leftrightarrow b_m$	$\Delta = 0$	$\Leftrightarrow Z'$

Wir nehmen nun an, daß P eine Lösung hat, und verfolgen den Weg des Simplex-Algorithmus für P. Es sei im ersten Schritt $\Leftrightarrow a_{ik}$ das Pivotelement. Nach den Regeln aus Kapitel 2.5 erhalten wir folgende Tableauwerte:

$$
\begin{aligned}
a'_{ik} &= \Leftrightarrow \frac{1}{a_{ik}} \\
a'_{ij} &= \frac{\Leftrightarrow a_{ij}}{\Leftrightarrow a_{ik}} = \frac{a_{ij}}{a_{ik}} \text{ für } \quad j \neq k \\
b'_i &= \Leftrightarrow \frac{b_i}{a_{ik}} \\
a'_{lk} &= \Leftrightarrow \frac{\Leftrightarrow a_{lk}}{\Leftrightarrow a_{ik}} = \Leftrightarrow \frac{a_{lk}}{a_{ik}} \text{ für } \quad l \neq i \\
c'_k &= \Leftrightarrow \frac{c_k}{\Leftrightarrow a_{ik}} = \frac{c_k}{a_{ik}} \\
a'_{lj} &= \Leftrightarrow a_{lj} \Leftrightarrow (\Leftrightarrow a_{lk} \frac{\Leftrightarrow a_{ij}}{\Leftrightarrow a_{ik}}) = \\
&\quad \Leftrightarrow a_{lj} + a_{lk} \frac{a_{ij}}{a_{ik}} \text{ für } \quad l \neq i \text{ und} j \neq k \\
b'_l &= b_l \Leftrightarrow a_{lk} \frac{b_i}{a_{ik}} \text{ für } \quad l \neq i \\
c'_j &= c_j \Leftrightarrow c_k \frac{a_{ij}}{a_{ik}} \text{ für } \quad j \neq k \\
\delta' &= \delta + c_k \frac{b_i}{a_{ik}}
\end{aligned}
$$

Wir nehmen nun im zweiten Simplextableau einen Basiswechsel zwischen y_i und v_k vor, unabhängig davon, ob der Simplex-Algorithmus dies vorschreibt. Wir erhalten folgende Werte.

$$
\begin{aligned}
a''_{ik} &= \frac{1}{a_{ik}} \\
a''_{ij} &= \Leftrightarrow \frac{a_{ij}}{a_{ik}} \text{ für } \quad j \neq k \\
b''_i &= \frac{b_i}{a_{ik}} \\
a''_{lk} &= \frac{a_{lk}}{a_{ik}} \text{ für } \quad l \neq i \\
c''_k &= \Leftrightarrow \frac{c_k}{a_{ik}} \\
a''_{lj} &= a_{lj} \Leftrightarrow a_{ij} \frac{a_{lk}}{a_{ik}} \text{ für } \quad l \neq i \text{ und } j \neq k
\end{aligned}
$$

$$
\begin{aligned}
b_l'' &= \Leftrightarrow b_l + b_i \frac{a_{lk}}{a_{ik}} \text{ für } \quad l \neq i \\
c_j'' &= \Leftrightarrow c_j + a_{ij} \frac{c_k}{a_{ik}} \text{ für } \quad j \neq k \\
\Delta'' &= \Delta \Leftrightarrow b_i \frac{c_k}{a_{ik}} = \Leftrightarrow \delta \Leftrightarrow b_i \frac{c_k}{a_{ik}},
\end{aligned}
$$

falls $\Delta = \Leftrightarrow \delta$ wie zu Beginn.

Nach diesen Basiswechseln bleiben viele Beziehungen erhalten. So die Beziehung zwischen x_j und v_j sowie zwischen u_l und y_l. Die Matrix des zweiten Tableaus bleibt die negative Transponierte der Matrix des ersten Tableaus. Die letzte Spalte des ersten Tableaus stimmt weiterhin mit der negierten letzten Zeile im zweiten Tableau überein. Ebenso bleibt die letzte Zeile im ersten Tableau gleich der negierten letzten Spalte im zweiten Tableau. Schließlich bleibt die Beziehung $\Delta = \Leftrightarrow \delta$ erhalten. Wenn dies für einen Basiswechsel gilt, dann auch für beliebig viele.

Schließlich stoppt der Simplex-Algorithmus für P. Die Koeffizienten seien mit * versehen. Der optimale Basispunkt $x^* = 0$, $u^* = \Leftrightarrow b^*$ ist zulässig, d. h. $\Leftrightarrow b^* \geq 0$. Da Z nicht verringert werden kann, ist $c^* \geq 0$. Wir untersuchen nun den Basispunkt $y^* = 0$, $v^* = c^*$ für das duale Problem. Der Basispunkt ist zulässig, da $c^* \geq 0$. Die Zielfunktion $\Leftrightarrow Z'$ kann nicht verringert werden, da $\Leftrightarrow b^* \geq 0$ ist. Also kann Z' nicht vergrößert werden, und der betrachtete Basispunkt ist optimal, d. h., P' hat eine Lösung. Der Wert der Lösung für P ist δ^*, und der Wert der Lösung für P' ist $\Leftrightarrow \Delta^* = \delta^*$. Also stimmen die Werte der Lösungen überein. Wir haben auch gezeigt, wie wir die Lösung für P' aus dem Lösungstableau für P ablesen können. □

Praktisch hat dieser Satz folgende Konsequenz. Wenn der erste Basispunkt für P unzulässig und $c \geq 0$ ist, erhalten wir für das duale Problem sofort einen zulässigen Basispunkt. Aus heuristischer Sicht sollte das duale Problem bearbeitet werden.

Satz 2.8.6: *Ein zulässiger Punkt p für das primale Problem ist genau dann optimal, wenn es für das duale Problem einen zulässigen Punkt d gibt, der folgende Bedingungen erfüllt:*

- $p_j > 0 \Rightarrow (\tilde{a}^j)^T d = c_j$ *($(\tilde{a}^j)$ ist die j-te Spalte von A).*
- $(\tilde{a}^j)^T d < c_j \Rightarrow p_j = 0$
- $d_i > 0 \Rightarrow \tilde{a}^i p = b_i$
- $\tilde{a}^i p > b_i \Rightarrow d_i = 0$.

Sind p und d für beide Probleme optimal, gelten die Bedingungen.

Beweis: Falls p für das primale Problem optimal ist, gibt es eine Lösung d für das duale Problem (Dualitätstheorem). Die Werte stimmen überein, d. h.

$$c^T p = b^T d.$$

Andererseits ist $c^T p \geq (A^T d)^T p = d^T A p = p^T A^T d = (Ap)^T d \geq b^T d$.

Aus beiden Beziehungen folgt die Gleichheit in der zweiten Ungleichungskette, d. h. $c^T p = p^T A^T d$ und $b^T d = p^T A^T d$. Die erste Gleichung schreiben wir ausführlich:

$$\sum_{1 \leq j \leq n} c_j p_j = \sum_{1 \leq j \leq n} p_j ((\tilde{a}^j)^T d)$$

Da d zulässig ist, gilt $(\tilde{a}^j)^T d \leq c_j$. Da p zulässig ist, gilt $p_j \geq 0$. Damit folgt $c_j p_j = p_j((\tilde{a}^j)^T d)$ für alle j. Ausführlich erhalten wir die ersten beiden Bedingungen, die anderen beiden Bedingungen folgen auf duale Weise.

Gelten die vier Bedingungen für zulässige Vektoren p und d, folgt mit einer umgekehrten Rechnung $c^T p = p^T A^T d = b^T d$. Aus dem Dualitätstheorem folgt, daß d und p optimal sind. □

Die in obigem Beweis hergeleitete Bedingung $p_j((\tilde{a}^j)^T d \Leftrightarrow c_j) = 0$ und die duale Bedingung $d_i(p^T(a^i)^T \Leftrightarrow b_i) = 0$ heißen auch Optimalitätsbedingungen.

Insgesamt tritt für P und P' stets einer der folgenden vier Fälle ein.

1.) Beide Probleme haben Lösungen, optimale Lösungen haben den gleichen Wert.

2.) Die Zielfunktion von P ist auf der zulässigen Menge nicht nach unten beschränkt, und die zulässige Menge von P' ist leer.

3.) Die Zielfunktion von P' ist auf der zulässigen Menge nicht nach oben beschränkt, und die zulässige Menge von P ist leer.

4.) Die zulässigen Mengen von P und P' sind leer.

Das Dualitätstheorem hat eine interessante ökonomische Interpretation. Das primale Problem $P : Z = g^T x \to \max$, $Ax \leq b$, $x \geq 0$ habe folgende Interpretation. Einem Unternehmen stehen m Produktionsfaktoren $R_1, \ldots, R_m$ in $b_1, \ldots, b_m$ Einheiten zur Verfügung. Es können n verschiedene Produkte $P_1, \ldots, P_n$ hergestellt werden, wobei zur Produktion von einer Einheit von P_j genau a_{ij} Einheiten von R_i $(1 \leq i \leq m)$ benötigt werden. Wenn x_j Einheiten von P_j $(1 \leq j \leq n)$ erstellt werden sollen, ist dies nur möglich, wenn die Nebenbedingungen $Ax \leq b$, d.h. $\sum_j a_{ij} x_j \leq b_i$ für $1 \leq i \leq n$, und $x \geq 0$ erfüllt sind. Der Gewinn beträgt $g^T x$, wenn g_i der Gewinn für eine Einheit von P_i ist. Der Gewinn soll maximiert werden.

Das duale Problem ist $Z' = b^T y \to \min$, $A^T y \geq g$, $y \geq 0$. Für realistische Daten haben die Probleme optimale Lösungen x^* und y^*. Nehmen wir nun an, daß b_i um 1 erhöht wird. Wenn die Lösung des Problems die gleiche Basis hat, erhöht sich der Wert von Z' um y_i^*. Nach dem Dualitätsprinzip ist auch $Z_{\text{opt}}^{\text{neu}} = Z(x^*) + y_i^*$. Wir können nun y_i^* als „Wert“ einer Einheit von R_i bezeichnen, auch „Schattenpreis“ von R_i genannt. Wir kaufen nämlich R_i nur für Preise, die den Schattenpreis nicht übersteigen. Mit dieser Interpretation ist $\sum_i a_{ij} y_i$ der Wert einer Einheit von P_j zum Preisvektor y. Die Nebenbedingung besagt, daß der Gewinn durch den Wert beschränkt sein muß. Unter diesen Nebenbedingungen soll der Wert der eingesetzten Produktionsmittel, nämlich $b^T y$, minimiert werden.

Das Dualitätstheorem sagt nun aus, daß bei optimaler Planung der Gesamtgewinn der produzierten Güter gleich dem Gesamtwert der eingesetzten Produktionsmittel ist. Bei allen suboptimalen Planungen erreicht der Gewinn nicht den Wert.

Auch die Optimalitätsbedingungen lassen sich ökonomisch interpretieren. Falls $a^i x^* < b_i$, wird der Produktionsfaktor R_i nicht voll ausgenutzt. Sein Wert ist 0, und die Optimalitätsbedingung impliziert $y_i^* = 0$. Ist andererseits $y_i^* > 0$, folgt $a^i x^* = b_i$, alle Produktionsfaktoren mit positivem Wert werden aufgebraucht. Wir kommen zu den dualen Bedingungen. Falls $(\tilde{a}^j)^T y^* > g_j$, überschreitet der Wert der Produktionsmittel zur Produktion von P_j den Gewinn. Vernünftigerweise impliziert die Optimalitätsgleichung, daß dieses Produkt nicht hergestellt wird, d.h. $x_j^* = 0$. Produkte, die produziert werden $(x_j^* > 0)$, haben dagegen die Eigenschaft, daß der Wert der eingesetzten Produktionsmittel dem Gewinn entspricht $((\tilde{a}^j)^T y^* = g_j)$.

2.9 Der Algorithmus von Khachiyan

Der Simplex-Algorithmus hat sich in der Praxis bewährt. Dennoch wird seit seiner Erfindung nach besseren Algorithmen gesucht. Auch aus praktischer Sicht ist eine Verringerung der Rechenzeit mehr als wünschenswert. Theoretisch ist interessant, ob es polynomielle Algorithmen für das Problem der Linearen Optimierung gibt.

Die Forschungen gehen in zwei Richtungen. Einerseits wird nach speziellen Algorithmen für spezielle Probleme (wenige Variablen oder wenige Nebenbedingungen) gesucht. Es gibt dann zwar polynomielle Algorithmen, aber die konstant gehaltene Parameterzahl taucht im Exponenten der Laufzeit auf, was für nicht sehr kleine Parameter zu praktisch nicht realisierbaren Laufzeiten führt.

Die andere Richtung betrachtet das allgemeine Problem der Linearen Optimierung. Bevor wir hier einsteigen, wollen wir die Rechenzeitabschätzungen näher diskutieren. Der Simplex-Algorithmus kommt mit $O\left(\binom{m+n}{n} mn\right)$ Rechenschritten aus, wobei die arithmetischen Operationen $+, \Leftrightarrow, *, /$ als Einzelschritte zählen. Die Abschätzung der Rechenzeit ist auch gültig, wenn die Eingaben reellwertig und Operationen über reellen Zahlen zugelassen sind. Bei Eingaben, die aus rationalen oder ganzen Zahlen bestehen, spielt die Bitlänge der gegebenen Zahlen für die Abschätzung der Zahl der arithmetischen Operationen keine Rolle.

Ein stark polynomieller Algorithmus für das Problem der Linearen Optimierung muß für Operationen auf den rationalen Zahlen eine von der Zahlenlänge unabhängige in n und m polynomielle Rechenzeit haben. Derartige Algorithmen sind noch unbekannt.

Bekannt sind Algorithmen, deren Laufzeit polynomiell in n, m und der Bitlänge der Eingabe ist. Wir lassen also nur rationale Eingaben zu. Durch entsprechende Skalierung können wir o. B. d. A. annehmen, daß die Parameter sogar ganzzahlig sind. Diese Algorithmen von Khachiyan (1979) und Karmarkar (1984) haben jeweils große Hoffnungen erregt. In der Praxis hat sich der Simplex-Algorithmus diesen Algorithmen als überlegen erwiesen. Das Gebiet der Linearen Optimierung ist also aus Forschungssicht längst nicht abgeschlossen.

Hier soll der Algorithmus von Khachiyan dargestellt werden. Der ursprüngliche Algorithmus entscheidet nur das Problem, ob das System von Ungleichungen

$$a_{i1}x_1 + \ldots + a_{in}x_n < b_i \quad (1 \leq i \leq m)$$

für $a_{ij}, b_i \in \mathbb{Z}$ durch einen Vektor $(x_1, \ldots, x_n) \in \mathbb{R}^n$ erfüllbar ist. Später werden wir diesen Algorithmus in einen Algorithmus für die Lineare Optimierung umformulieren. Wir nehmen o. B. d. A. an, daß kein Vektor $a_i = (a_{i1}, \ldots, a_{in})$ ein Nullvektor ist. Die zugehörige Ungleichung ist entweder trivialerweise erfüllt $(b_i > 0)$ und kann wegfallen, oder sie ist unerfüllbar $(b_i \leq 0)$, dann ist das Problem bereits gelöst.

Algorithmus 2.9.1 (von Khachiyan):

1.) $k := 0$, $x^{(k)} := 0$ (Vektor der Länge n), I $n \times n$-Einheitsmatrix,

$$L := \sum_{1 \leq i \leq m, 1 \leq j \leq n} \lceil \log(|a_{ij}| + 1) \rceil + \sum_{1 \leq i \leq m} \lceil \log(|b_i| + 1) \rceil + \lceil \log mn \rceil + 1$$

(Bitlänge der Eingabe),
$B^{(k)} = 2^{2L} I$.

2.) Falls $x^{(k)}$ das Ungleichungssystem erfüllt, STOP. Das Problem ist gelöst.
Falls $x^{(k)}$ das Ungleichungssystem nicht erfüllt und $k \geq 4(n+1)^2 L$ ist, STOP. Es wird behauptet, daß das Problem keine Lösung hat.
Ansonsten $k := k + 1$. Sei die i–te Ungleichung eine, die der Punkt $x^{(k-1)}$ nicht erfüllt (d. h. $a_i^T x^{(k-1)} \geq b_i$),

$$x^{(k)} := x^{(k-1)} \Leftrightarrow \frac{1}{n+1} \frac{B^{(k-1)} a_i}{\sqrt{a_i^T B^{(k-1)} a_i}}$$

$$B^{(k)} := \frac{n^2}{n^2 \Leftrightarrow 1} \left[B^{(k-1)} \Leftrightarrow \frac{2}{n+1} \frac{\left(B^{(k-1)} a_i\right) \left(B^{(k-1)} a_i\right)^T}{a_i^T B^{(k-1)} a_i} \right]$$

Weiter in Schritt 2.

Der Algorithmus stoppt spätestens nach $4(n+1)^2 L = O(n^2 L)$ Iterationen. Mit den üblichen Algorithmen

für die Multiplikation von Matrizen mit Vektoren oder Vektoren mit Vektoren folgt, daß jede Iteration in $O(n(n+m))$ Schritten durchführbar ist.

Satz 2.9.2: *Die Rechenzeit des Algorithmus von Khachiyan ist durch $O(n^3(m+n)L)$ Operationen beschränkt.*

Es muß also „nur“ gezeigt werden, daß das Ungleichungssystem keine Lösung hat, wenn nach $4(n+1)^2L$ Iterationen keine Lösung gefunden wurde. Dies ist allerdings harte Arbeit. Bevor wir uns an die Arbeit machen, wollen wir die Idee, die hinter diesem Algorithmus steckt, herausarbeiten. Da wir mit $<$-Ungleichungen arbeiten, ist der Lösungsraum leer oder eine offene Teilmenge des $\mathbb{R}^n$ und muß dann positives Volumen haben. Dieses Volumen kann vorab in Abhängigkeit von m,n und L durch eine positive Zahl abgeschätzt werden. Das gilt sogar für den Teil S des Lösungsraumes, der in der Kugel mit Radius 2^L um den Nullpunkt liegt. Diese Kugel nennen wir $E^{(0)}$, da wir sie als spezielles Ellipsoid auffassen. Allgemein sind Ellipsoide E Mengen, die sich folgendermaßen darstellen lassen. Es gibt eine symmetrische ($b_{ij} = b_{ji}$), positiv definite ($x^T Bx > 0$ für alle $x \neq 0$) $n \times n$-Matrix B über $\mathbb{R}$ und einen Punkt $x' \in \mathbb{R}^n$, so daß gilt

$$E = \left\{ x \mid (x \Leftrightarrow x')^T B^{-1} (x \Leftrightarrow x') \leq 1 \right\}.$$

Kugeln um den Ursprung mit Radius r sind spezielle Ellipsoide, denn sie lassen sich folgendermaßen darstellen

$$\left\{ x \mid x_1^2 + \ldots + x_n^2 \leq r^2 \right\} = \left\{ x \mid (x_1^2 + \ldots + x_n^2)/r^2 \leq 1 \right\} = \left\{ x \mid (x \Leftrightarrow 0)^T (r^2 I)^{-1} (x \Leftrightarrow 0) \leq 1 \right\}.$$

Mit $x^{(0)} = x' = 0$ und $B^{(0)} := 2^{2L} I$ erhalten wir als Ellipsoid $E^{(0)}$ die Kugel mit Radius 2^L um den Ursprung. Nach Definition ist $S \subseteq E^{(0)}$. Das Ellipsoid $E^{(k)}$ wird zu $x^{(k)}$ und $B^{(k)}$ gebildet. Wir haben zu zeigen, daß $B^{(k)}$ symmetrisch und positiv definit ist. Weiterhin zeigen wir, daß einerseits S in allen $E^{(k)}$ enthalten ist und daß andererseits das Volumen der Ellipsoide so schnell fällt, daß das Volumen von $E^{(k)}$ für $k = 4(n+1)^2L$ kleiner als das Volumen von S ist, falls S nicht leer ist. Daraus folgt unmittelbar die Korrektheit des Algorithmus von Khachiyan.

Etwas genauer können wir das Verfahren noch beschreiben. Es sei $S \subseteq E^{(k)}$ und $x^{(k)}$ keine Lösung, da die i-te Ungleichung verletzt ist. Sei weiter

$$H^{(k)} = \{ x \mid a_i^T (x \Leftrightarrow x^{(k)}) \leq 0 \} = \{ x \mid a_i^T x \leq a_i^T x^{(k)} \}$$

der Halbraum, der durch die i-te Nebenbedingung parallel verschoben zum Punkt $x^{(k)}$ beschrieben ist. Wir beobachten, daß er nach Definition alle zulässigen Punkte enthält und somit $S \subseteq E^{(k)} \cap H^{(k)}$ gilt. Wir werden $E^{(k+1)}$ so definieren, daß das Volumen schnell genug fällt und $E^{(k)} \cap H^{(k)} \subseteq E^{(k+1)}$ gilt. Dann folgt $S \subseteq E^{(k+1)}$ trivialerweise. In Kapitel 2.10 beweisen wir die Korrektheit des Algorithmus, und in Kapitel 2.11 wenden wir ihn auf die Lineare Optimierung an. Schon hier ist festzustellen, daß sich der Simplex-Algorithmus und der Algorithmus von Khachiyan grundlegend unterscheiden. Der Simplex-Algorithmus diskretisiert das Problem, indem er es auf die Extrempunkte oder Basispunkte reduziert. Der Algorithmus von Khachiyan arbeitet analytisch. Aus „eigentlich“ viel schöneren Polyedern macht er die „scheinbar“ unhandlicheren Ellipsoide. Diese haben jedoch die angenehme Eigenschaft, daß ihr Volumen wesentlich einfacher handhabbar als das von Polyedern ist.

2.10 Die Korrektheit des Algorithmus von Khachiyan

Wir beginnen mit der Untersuchung des Volumens von S, dem Durchschnitt des Lösungsraumes (unter der Annahme, daß dieser nicht leer ist) mit der Kugel mit Radius 2^L um den Nullpunkt.

Lemma 2.10.1: *Jeder Extrempunkt des Polyeders $Ax \leq b, x \geq 0$ hat rationale Koordinaten. Der Betrag der Koordinaten und der Betrag des Nenners (natürlich in gekürzter Darstellung) ist kleiner als $2^L/mn$.*

Beweis: Wir wissen, daß jeder Extrempunkt n linear unabhängige Gleichungen aus dem System $Ax = b$ und $x = 0$ erfüllt. Das Gleichungssystem von n derartigen Gleichungen sei in Matrixform $Dx = z$. Wegen der linearen Unabhängigkeit der Gleichungen gilt $\det D \neq 0$. Für den Extrempunkt $v = (v_1, \ldots, v_n)$ gilt $v = D^{-1}z$. Nach der Cramer'schen Regel gilt

$$v_k = \frac{1}{\det D} \det D_k \quad \text{mit} \quad D_k := \begin{pmatrix} d_{11} & \cdots & d_{1,k-1} & z_1 & d_{1,k+1} & \cdots & d_{1n} \\ \vdots & & \vdots & \vdots & \vdots & & \vdots \\ d_{n1} & \cdots & d_{n,k-1} & z_n & d_{n,k+1} & \cdots & d_{nn} \end{pmatrix}$$

Es bleibt zu zeigen, daß $|\det D|$ und $|\det D_k|$ kleiner als $2^L/mn$ sind.

Aus der Linearen Algebra sollte bekannt sein, daß das Hyperparallelepiped, das von den Spaltenvektoren $d_1, \ldots, d_n$ von D aufgespannt wird, das n-dimensionale Volumen $|\det D|$ hat. Für $n = 2$ (Parallelogramm) und $n = 3$ (Spat) ist dies leicht nachzurechnen. Das Volumen eines Hyperparallelepipeds mit gegebenen Seitenlängen ist maximal, wenn die Vektoren senkrecht aufeinanderstehen (Rechteck, Quader, ...). Dann ist das Volumen gleich dem Produkt der Länge der Vektoren. Es sei $\|d_j\|$ die Länge des Vektors d_j, d. h.

$$\|d_j\| = \sqrt{d_{1j}^2 + \ldots + d_{nj}^2}$$

Nach den obigen Überlegungen gilt $|\det D| \leq \|d_1\| * \ldots * \|d_n\|$. Aus der trivialen Ungleichung (alle Summanden links kommen rechts vor)

$$y_1^2 + \ldots + y_n^2 \leq (|y_1| + 1)^2 * \ldots * (|y_n| + 1)^2$$

folgt $\|y\| = (y_1^2 + \ldots + y_n^2)^{1/2} \leq \prod_{1 \leq i \leq n} (|y_i| + 1)$.

Also gilt

$$|\det D| \leq \prod_{1 \leq j \leq n} \prod_{1 \leq i \leq n} (|d_{ij}| + 1) \leq 2^{(L - \log mn - 1)} < 2^L/mn.$$

Dabei folgt die vorletzte Ungleichung nach Logarithmieren, da jedes Element aus der Eingabe (A, b) nur höchstens einmal in D vorkommt. Hinzu kommen eventuell aus Gleichungen $x_j = 0$ Einheitsvektoren, die die Determinante nicht vergrößern. Völlig analog folgen die Aussagen über $|\det D_k|$. □

Korollar 2.10.2: *Jeder Extrempunkt des Polyeders $Ax \leq b, x \geq 0, x \leq \lfloor 2^L/n \rfloor e$ mit $e = (1, \ldots, 1)$ hat rationale Koordinaten, deren Nenner betragsmäßig kleiner als $2^L/mn$ sind.*

Beweis: Der Beweis verläuft analog zum Beweis von Lemma 2.10.1, da wir nur die Nenner, also $|\det D|$ betrachten. □

Lemma 2.10.3: *Wenn das Ungleichungssystem $Ax < b$ eine Lösung hat, hat die Menge S ein Volumen von mindestens $2^{-(n+1)L}$.*

Beweis: Es sei y eine Lösung von $Ax < b$. O. B. d. A. nehmen wir $y_i \geq 0$ für alle i an, sonst transformieren wir x_i in $\Leftrightarrow x_i$. Da $Ax < b$ eine offene Lösungsmenge hat, können wir auch $y_i > 0$ annehmen. Damit hat das Polyeder $Ax \leq b, x \geq 0$ einen inneren Punkt und positives Volumen. Da wir uns auf einen Orthanten ($x \geq 0$) beschränken, liegt keine Gerade ganz im Polyeder. Also hat das Polyeder einen Extrempunkt v. Nach Lemma 2.10.1 ist $\|v\|_\infty := \max\{|v_i| \mid 1 \leq i \leq n\} < \lfloor 2^L/n \rfloor$. Es gibt eine ε-Kugel um den Extrempunkt, die ganz in dem Polyeder aller x^* mit $\|x^*\|_\infty < \lfloor 2^L/n \rfloor$ liegt. Damit hat auch das Polyeder $Ax \leq b, x \geq 0, x \leq \lfloor 2^L/n \rfloor e$ einen inneren Punkt und ist echt n-dimensional, d. h. es liegt auf keiner Hyperebene. Da n Extrempunkte nur ein höchstens $(n \Leftrightarrow 1)$-dimensionales Gebilde aufspannen, gibt es $n + 1$ Extrempunkte, die nicht auf einer Hyperebene liegen. Seien $v_0, \ldots, v_n$ diese Extrempunkte, die ein Simplex aufspannen. Das Volumen des Polyeders ist mindestens so groß wie das Volumen dieses Simplexes.

Wir schätzen nun das Volumen des Simplex ab. Es beträgt, wie aus der Linearen Algebra bekannt ist,

$$\left|\det\begin{pmatrix} 1 & \dots & 1 \\ v_0 & \dots & v_n \end{pmatrix}\right| / n! \quad .$$

Nach Korollar 2.10.2 können wir in $\begin{pmatrix} 1 & \dots & 1 \\ v_0 & \dots & v_n \end{pmatrix}$ alle rationalen Einträge durch Nenner, die kleiner als $2^L/n$ sind, ausdrücken. Für jeden Vektor v_i gilt, daß wir für jede seiner Komponenten sogar den gleichen Nenner benutzen können. Dies folgt nach dem Beweis von Lemma 2.10.1. Also ist $v_i = \frac{1}{d_i} u_i$ mit $d_i \in \mathbb{Z},\ |d_i| < 2^L/n,\ u_i \in \mathbb{Z}^n$. Also folgt

$$\left|\det\begin{pmatrix} 1 & \dots & 1 \\ v_0 & \dots & v_n \end{pmatrix}\right| = \left|\det\begin{pmatrix} 1 & \dots & 1 \\ \frac{1}{d_0}u_0 & \dots & \frac{1}{d_n}u_n \end{pmatrix}\right|$$

$$= \frac{1}{|d_0| * \dots * |d_n|} \cdot \left|\det\begin{pmatrix} d_0 & \dots & d_n \\ u_0 & \dots & u_n \end{pmatrix}\right| \geq \frac{1}{|d_0| * \dots * |d_n|} > \left(\frac{2^L}{n}\right)^{-(n+1)} .$$

Die vorletzte Ungleichung folgt, da wir eine ganzzahlige Matrix erhalten haben. Dann ist die Determinante ganzzahlig. Sie kann aber nicht 0 sein, da sonst das Volumen des Simplexes 0 wäre. Die letzte Ungleichung folgt, da $|d_i| < 2^L/n$ für alle i ist.

Also können wir das Simplexvolumen nach unten abschätzen durch

$$\frac{1}{n!}\left|\det\begin{pmatrix} 1 & \dots & 1 \\ v_0 & \dots & v_n \end{pmatrix}\right| > \frac{1}{n!}\left(\frac{2^L}{n}\right)^{-(n+1)} = 2^{-(n+1)L} \cdot \frac{n^{n+1}}{n!} > 2^{-(n+1)L}.$$

□

Im folgenden soll der bereits früher beschriebene Lösungsweg nachvollzogen werden. Hierbei handelt es sich leider um längliche Rechnungen, die keine größeren Ideen enthalten. Zunächst skizzieren wir unser Vorgehen. Lemma 2.10.4 beschreibt eine affine Transformation, mit der das gegebene Ellipsoid in die Einheitskugel transformiert wird; ebenso wird der Halbraum, der der ersten Nebenbedingung entspricht, die von dem untersuchten Vektor $x^{(k)}$ nicht erfüllt wird, auf den „einfachsten" Halbraum $H = \{x_1 \geq 0\}$ transformiert. Ellipsoide werden von affinen Transformationen in Ellipsoide transformiert und Halbräume in Halbräume. Teilmengenbeziehungen bleiben erhalten, und das Volumen aller Ellipsoide wird um den gleichen Faktor verändert. Mit diesem Hilfsmittel müssen die folgenden Lemmas nur noch für die einfacheren Fälle bewiesen werden.

In Lemma 2.10.5 zeigen wir dann, daß $B^{(k+1)}$ symmetrisch und positiv definit ist, wenn dies für $B^{(k)}$ gilt. In Lemma 2.10.6 wird die schon diskutierte Behauptung $E^{(k)} \cap H^{(k)} \subseteq E^{(k+1)}$ nachgewiesen. Schließlich wird in Lemma 2.10.7 das Volumen von $E^{(k+1)}$ mit dem Volumen von $E^{(k)}$ verglichen.

Wir wollen also mit der Einheitskugel $E = \{x \mid x^T x \leq 1\}$ und dem Halbraum $H = \{x \mid x^T v \leq 0\}$ für $v^T = (\Leftrightarrow 1, 0, \dots, 0)$ starten. Für das Ellipsoid E ist also $x' = 0$ und $B = I$. Für das im Algorithmus von Khachiyan definierte Nachfolgeellipsoid gilt für den neuen x'-Vektor x'_{neu}

$$x'_{neu} = x' \Leftrightarrow \frac{1}{n+1}\frac{Iv}{\sqrt{v^T Iv}} = \Leftrightarrow \frac{1}{n+1} \cdot v.$$

Wir setzen $\alpha := \frac{1}{n+1}$. Mit $\beta := \frac{n^2}{n^2-1}$ und $\gamma := \frac{2}{n+1}$ ergibt sich als neue Matrix

$$B_{neu} = \beta(I \Leftrightarrow \gamma v v^T).$$

Das neue Ellipsoid läßt sich beschreiben durch

$$E_{neu} = \left\{x \mid (x \Leftrightarrow x'_{neu})^T B_{neu}^{-1}(x \Leftrightarrow x'_{neu}) \leq 1\right\}$$

Wenn wir nun mit einem beliebigen Ellipsoid

$$\bar{E} = \left\{ x \mid (x \Leftrightarrow \bar{x})^T \bar{B}^{-1} (x \Leftrightarrow \bar{x}) \leq 1 \right\}$$

und dem Halbraum

$$\bar{H} = \left\{ x \mid (x \Leftrightarrow \bar{x})^T a \leq 0 \right\}$$

($a \neq 0$) starten, erhalten wir das neue Ellipsoid

$$\bar{E}_{neu} = \left\{ x \;\middle|\; \left(x \Leftrightarrow \bar{x} + \alpha \frac{\bar{B}a}{\sqrt{a^T \bar{B} a}} \right)^T \frac{1}{\beta} \left(\bar{B} \Leftrightarrow \gamma \frac{(\bar{B}a)(\bar{B}a)^T}{a^T \bar{B} a} \right)^{-1} \left(x \Leftrightarrow \bar{x} + \alpha \frac{\bar{B}a}{\sqrt{a^T \bar{B} a}} \right) \leq 1 \right\}.$$

Lemma 2.10.4: *Es gibt eine affine Transformation T, d.h. $T(x) = Zx + b$, mit $\bar{H} = T(H)$, $\bar{E} = T(E)$ und $\bar{E}_{neu} = T(E_{neu})$.*

Beweis: Aus der Linearen Algebra ist folgendes bekannt. Da $\bar{B}$ symmetrisch und positiv definit ist, gibt es eine reguläre Matrix L mit $\bar{B} = LL^T$. Da $a \neq 0$, ist $L^T a \neq 0$. Daher geht v aus $L^T a$ durch eine Drehung und anschließende Normierung hervor, d.h. es gibt eine Rotationsmatrix Q mit

$$v = \frac{Q^T L^T a}{\|Q^T L^T a\|}.$$

Bekanntlich beschreibt Q^T die Rotation in die Gegenrichtung der Rotation Q, d.h. $QQ^T = I$. Für jeden Vektor x ist $\|x\|^2 = x^T x$ und daher

$$\left\|Q^T L^T a\right\|^2 = (Q^T L^T a)^T Q^T L^T a = a^T L Q Q^T L^T a = a^T L L^T a = a^T \bar{B} a.$$

Wir definieren die affine Abbildung T durch

$$T(x) = LQx + \bar{x}.$$

Die Umkehrabbildung ist

$$T^{-1}(x) = Q^T L^{-1}(x \Leftrightarrow \bar{x}),$$

denn $T^{-1}T(x) = Q^T L^{-1}(LQx + \bar{x} \Leftrightarrow \bar{x}) = Q^T L^{-1} LQx = Q^T Qx = x$. Es ist

$$\begin{aligned}
T(H) &= \{T(x) \mid x \in H\} \\
&= \{x \mid T^{-1}(x) \in H\} \\
&= \{x \mid (Q^T L^{-1}(x \Leftrightarrow \bar{x}))^T v \leq 0\} \\
&= \{x \mid (x \Leftrightarrow \bar{x})^T (L^{-1})^T Q v \leq 0\} \\
&= \{x \mid (x \Leftrightarrow \bar{x})^T (L^{-1})^T Q Q^T L^T a / \left\|Q^T L^T a\right\| \leq 0\} \text{(s. Gleichung für } v) \\
&= \{x \mid (x \Leftrightarrow \bar{x})^T (L^{-1})^T L^T a \leq 0\} \\
&= \{x \mid (x \Leftrightarrow \bar{x})^T a \leq 0\} \\
&= \bar{H},
\end{aligned}$$

$$\begin{aligned}
T(E) &= \{T(x) \mid x \in E\} \\
&= \{x \mid T^{-1}(x) \in E\} \\
&= \{x \mid (T^{-1}(x))^T (T^{-1}(x)) \leq 1\} \\
&= \{x \mid (Q^T L^{-1}(x \Leftrightarrow \bar{x}))^T (Q^T L^{-1}(x \Leftrightarrow \bar{x})) \leq 1\} \\
&= \{x \mid (x \Leftrightarrow \bar{x})^T (L^{-1})^T Q Q^T L^{-1}(x \Leftrightarrow \bar{x}) \leq 1\} \\
&= \{x \mid (x \Leftrightarrow \bar{x})^T (LL^T)^{-1}(x \Leftrightarrow \bar{x}) \leq 1\} \\
&= \{x \mid (x \Leftrightarrow \bar{x})^T \bar{B}^{-1}(x \Leftrightarrow \bar{x}) \leq 1\} \\
&= \bar{E}.
\end{aligned}$$

Nun machen wir uns an die Berechnung von $T(E_{neu})$.

$$
\begin{aligned}
& T(E_{neu}) = \\
= \quad & \{x \mid T^{-1}(x) \in E_{neu}\} \\
= \quad & \{x \mid Q^T L^{-1} \cdot (x \Leftrightarrow \bar{x}) \in E_{neu}\} \\
= \quad & \{x \mid (Q^T L^{-1}(x \Leftrightarrow \bar{x}) + \alpha v)^T \quad \frac{1}{\beta}(I \Leftrightarrow \gamma(vv^T))^{-1} \quad (Q^T L^{-1}(x \Leftrightarrow \bar{x}) + \alpha v) \leq 1\}.
\end{aligned}
$$

Oben hatten wir gezeigt, daß $v = \dfrac{Q^T L^T a}{\|Q^T L^T a\|}$ und $\left\|Q^T L^T a\right\|^2 = a^T \bar{B} a$ gilt. Damit erhalten wir

$$
vv^T = \frac{1}{a^T \bar{B} a} Q^T L^T a a^T L Q \quad \text{und} \quad \alpha v = \alpha \frac{Q^T L^T a}{\|Q^T L^T a\|} = \alpha \frac{Q^T L^{-1} L L^T a}{\sqrt{a^T \bar{B} a}} = Q^T L^{-1} \alpha \frac{\bar{B} a}{\sqrt{a^T \bar{B} a}}.
$$

Also ist

$$
Q^T L^{-1}(x \Leftrightarrow \bar{x}) + \alpha v = Q^T L^{-1} \left(x \Leftrightarrow \bar{x} + \alpha \frac{\bar{B} a}{\sqrt{a^T \bar{B} a}} \right).
$$

Wir fahren in unserer Berechnung von $T(E_{neu})$ fort.

$$
\begin{aligned}
T(E_{neu}) \quad = \quad & \\
= \quad & \left\{ x \;\middle|\; \left(x \Leftrightarrow \bar{x} + \alpha \frac{\bar{B} a}{\sqrt{a^T \bar{B} a}} \right)^T (L^{-1})^T Q \quad \frac{1}{\beta}(I \Leftrightarrow \gamma v v^T)^{-1} \quad Q^T L^{-1} \left(x \Leftrightarrow \bar{x} + \alpha \frac{\bar{B} a}{\sqrt{a^T \bar{B} a}} \right) \leq 1 \right\} \\
= \quad & \left\{ x \;\middle|\; \left(x \Leftrightarrow \bar{x} + \alpha \frac{\bar{B} a}{\sqrt{a^T \bar{B} a}} \right)^T \quad (L^{-1})^T Q \frac{1}{\beta}(I \Leftrightarrow \gamma v v^T)^{-1} Q^T L^{-1} \quad \left(x \Leftrightarrow \bar{x} + \alpha \frac{\bar{B} a}{\sqrt{a^T \bar{B} a}} \right) \leq 1 \right\}
\end{aligned}
$$

Damit $T(E_{neu}) = \bar{E}_{neu}$ ist, genügt es, folgende Gleichung zu beweisen, die wir anschließend äquivalent umformen.

$$
\begin{aligned}
(L^{-1})^T Q \left(I \Leftrightarrow \gamma v v^T\right)^{-1} Q^T L^{-1} \quad &= \quad \left(\bar{B} \Leftrightarrow \gamma \frac{(\bar{B}a)(\bar{B}a)^T}{a^T \bar{B} a} \right)^{-1} \\
\Leftrightarrow \quad L(Q^T)^{-1} \left(I \Leftrightarrow \gamma v v^T\right) Q^{-1} L^T \quad &= \quad \bar{B} \Leftrightarrow \gamma \frac{(\bar{B}a)(\bar{B}a)^T}{a^T \bar{B} a} \\
\Leftrightarrow \quad L(Q^T)^{-1} Q^{-1} L^T \Leftrightarrow \gamma \frac{L(Q^T)^{-1} Q^T L^T a a^T L Q Q^{-1} L^T}{a^T \bar{B} a} \quad &= \quad \bar{B} \Leftrightarrow \gamma \frac{(\bar{B}a)(\bar{B}a)^T}{a^T \bar{B} a}.
\end{aligned}
$$

Die ersten Summanden stimmen überein, da

$$
L(Q^T)^{-1} Q^{-1} L^T = L(QQ^T)^{-1} L^T = L L^T = \bar{B}.
$$

Von den zweiten Summanden muß nur noch die Gleichheit der Zähler gezeigt werden. Auf der linken Seite ergibt sich

$$
L L^T a a^T L L^T = \bar{B} a a^T \bar{B}^T = (\bar{B}a)(\bar{B}a)^T.
$$

□

Lemma 2.10.5: *Falls im Algorithmus von Khachiyan die Matrix $B (= B^{(k-1)})$ symmetrisch und positiv definit ist, gilt dies auch für die Matrix $B_{neu} (= B^{(k)})$. Insbesondere ist $E^{(k)}$ tatsächlich ein Ellipsoid.*

Beweis: Es genügt nach Lemma 2.10.4, den einfachen Fall $x' = 0$, $B = I$ und $a^T = (\Leftrightarrow 1, 0, \ldots, 0)$ zu betrachten. Dann ist

$$
x'_{neu} = 0 \Leftrightarrow \frac{1}{n+1} \frac{a}{\sqrt{a^T a}} = \left(\frac{1}{n+1}, 0, \ldots, 0 \right)^T
$$

und $B_{neu} = \frac{n^2}{n^2 \Leftrightarrow 1}\left(I \Leftrightarrow \frac{2}{n+1}\frac{aa^T}{a^Ta}\right) = \begin{pmatrix} n^2/(n+1)^2 & & & 0 \\ & n^2/(n^2 \Leftrightarrow 1) & & \\ & & \ddots & \\ 0 & & & n^2/(n^2 \Leftrightarrow 1) \end{pmatrix}$

Die letzte Gleichung folgt, da aa^T nur an der Position $(1,1)$ eine 1 hat und sonst aus Nullen besteht, $a^Ta = 1$ ist und nach etwas Bruchrechnung. Somit ist B_{neu} symmetrisch. Wir zeigen nun, daß B_{neu} positiv definit ist. Dazu sei $x \neq 0$. Dann folgt

$$x^T B_{neu} x = \frac{n^2}{(n+1)^2}x_1^2 + \frac{n^2}{n^2 \Leftrightarrow 1}(x_2^2 + \ldots + x_n^2) > 0.$$

□

Lemma 2.10.6: *Im Algorithmus von Khachiyan gilt* $E^{(k)} \cap H^{(k)} \subseteq E^{(k+1)}$.

Beweis: Wieder können wir uns auf den einfachen Fall $x' = 0$, $B = I$ und $a^T = (\Leftrightarrow 1, 0, \ldots, 0)$ beschränken. Es sei also $x^* = (x_1, \ldots, x_n)^T$ ein Punkt aus dem zugehörigen Ellipsoid und dem zugehörigen Halbraum, d. h. $\|x^*\| \leq 1$ und $0 \leq \Leftrightarrow a^T x^* = x_1$. Da $\|x^*\| \leq 1$, ist $x_1 \leq 1$. Wir müssen nun zeigen, daß $x^* \in E_{neu}$ ist, wobei E_{neu} über x'_{neu} und B_{neu} (s. Beweis von Lemma 2.10.5) definiert ist. Da B_{neu} und auch B_{neu}^{-1} Diagonalmatrizen sind, läßt sich das Produkt

$$(x^* \Leftrightarrow x'_{neu})^T B_{neu}^{-1}(x^* \Leftrightarrow x'_{neu})$$

besonders einfach ausrechnen, es ergibt sich nämlich:

$$\begin{aligned} \frac{(n+1)^2}{n^2} \cdot (x_1 \Leftrightarrow \frac{1}{n+1})^2 + \frac{n^2 \Leftrightarrow 1}{n^2} \cdot (x_2^2 + \cdots + x_n^2) &= \frac{(n+1)^2}{n^2} \cdot (x_1 \Leftrightarrow \frac{1}{n+1})^2 + \frac{n^2 \Leftrightarrow 1}{n^2} \cdot (\|x^*\|^2 \Leftrightarrow x_1^2) \\ &\leq \frac{(n+1)^2}{n^2} \cdot (x_1 \Leftrightarrow \frac{1}{n+1})^2 + \frac{n^2 \Leftrightarrow 1}{n^2} \cdot (1 \Leftrightarrow x_1^2), \end{aligned}$$

wobei wir $\|x^*\| \leq 1$ benutzt haben. Wenn man die rechte Seite als Funktion in x_1 betrachtet, dann ist es eine nach oben offene Parabel. Das Maximum der Funktion wird also auf dem Bereich $x_1 \in [0,1]$ für $x_1 = 1$ oder $x_1 = 0$ angenommen. Für $x_1 = 0$ ist die rechte Seite $\frac{1}{n^2} + \frac{n^2-1}{n^2} = 1$, und für $x_1 = 1$ ist die rechte Seite $1 + \frac{n^2-1}{n^2} \cdot (1 \Leftrightarrow 1) = 1$. Damit ist $x^* \in E_{neu}$. □

Lemma 2.10.7: *Der Quotient aus dem Volumen des neuen Ellipsoids und dem Volumen des alten Ellipsoids ist kleiner als* $2^{-1/(2(n+1))}$.

Beweis: Zunächst überlegen wir uns, wie die affine Transformation T aus dem Beweis von Lemma 2.10.4 das Volumen verändert. Sei $\bar{E} = T(E)$. Die Verschiebung verändert das Volumen nicht. Also können wir hier $T(x) = LQx$ setzen. Also gilt

$$vol(\bar{E}) = \det(LQ)vol(E) = \det(L)\det(Q)vol(E).$$

Dabei ist $\det(Q) = 1$, da Q eine Drehung ist und das Volumen gleich läßt. Da $LL^T = \bar{B}$, ist $\det(L) = \det(\bar{B})^{1/2}$ von E unabhängig. Somit können wir uns wieder auf den einfachen Fall zurückziehen.

Dabei ist

$$vol(E_{neu})/vol(E) = \det(B_{neu})^{1/2}.$$

Da B_{neu} eine Diagonalmatrix ist, läßt sich $\det(B_{neu})^{1/2}$ leicht berechnen als

$$\frac{n}{n+1} \cdot \left(\frac{n^2}{n^2 \Leftrightarrow 1}\right)^{(n-1)/2} = (1 \Leftrightarrow \frac{1}{n+1}) \cdot \left(1 + \frac{1}{n^2 \Leftrightarrow 1}\right)^{(n-1)/2}.$$

Wir benutzen die Abschätzung $1+x < e^x$ für alle $x \neq 0$ und erhalten:

$$1 \Leftrightarrow \frac{1}{n+1} < e^{-\frac{1}{n+1}} \quad \text{sowie} \quad 1 + \frac{1}{n^2 \Leftrightarrow 1} < e^{\frac{1}{n^2-1}},$$

also ist der Quotient der beiden Volumina beschränkt durch

$$e^{-\frac{1}{n+1}+\frac{(n-1)/2}{n^2-1}} = e^{-\frac{1}{n+1}+\frac{1}{2\cdot(n+1)}} = e^{-\frac{1}{2\cdot(n+1)}} < 2^{-\frac{1}{2\cdot(n+1)}} (< 1).$$

□

Satz 2.10.8: *Der Algorithmus von Khachiyan ist korrekt.*

Beweis: Es bleibt zu zeigen, daß das gegebene Ungleichungssystem keine Lösung hat, wenn der Algorithmus in $4(n+1)^2 L$ Iterationen keine Lösung gefunden hat. Wir wissen, daß es eine Lösung in $E^{(0)}$ gibt, wenn es überhaupt eine Lösung gibt. Sei S die Menge aller Lösungen in $E^{(0)}$ und $S \neq \emptyset$ angenommen. Nach Lemma 2.10.3 hat S dann ein Volumen von mindestens $2^{-(n+1)L}$. Für alle Iterationen gilt nach der Konstruktion $S \subseteq E^{(k)}$. Wir schneiden stets nur einen Halbraum von Nichtlösungen ab und überdecken den Rest durch das nächste Ellipsoid (Lemma 2.10.5 und Lemma 2.10.6). Nach Lemma 2.10.7 ist

$$vol(E^{(k)}) < 2^{-k/(2(n+1))} vol(E^{(0)}).$$

Das Volumen der Kugel $E^{(0)}$ mit Radius 2^L schätzen wir durch das Volumen eines Würfels mit Kantenlänge $2 \cdot 2^L$ ab, dieses beträgt $2^{n(L+1)}$. Also gilt für $k = 4(n+1)^2 L$

$$vol(E^{(k)}) < 2^{-2(n+1)L+n(L+1)} = 2^{-nL-2L+n}.$$

Da nach Definition von L gilt $n < L$, folgt $vol(E^{(k)}) < 2^{-(n+1)L}$. Dies ist ein Widerspruch zu $S \subseteq E^{(k)}$ und $vol(S) \geq 2^{-(n+1)L}$. □

2.11 Die Anwendung des Algorithmus von Khachiyan auf die Lineare Optimierung

Zunächst reduzieren wir die Lösung von Problemen der Linearen Optimierung auf das Lösen von linearen Ungleichungssystemen mit $\leq$-Ungleichungen. In einem zweiten Schritt setzen wir Ungleichungssysteme mit $\leq$-Ungleichungen mit Ungleichungssystemen mit $<$-Ungleichungen in Beziehung.

Wir starten mit dem Problem $c^T x \rightarrow \max$ mit $Ax \leq b$, $x \geq 0$. Das duale Problem lautet $b^T y \rightarrow \min$ mit $A^T y \geq c$, $y \geq 0$. Nach der Dualitätstheorie gilt für alle zulässigen Lösungen $c^T x \leq b^T y$. Darüber hinaus hat das primale Problem genau dann eine endliche Lösung, wenn es zulässige Lösungen mit $c^T x = b^T y$ gibt. Also ist das Ungleichungssystem

$$c^T x \geq b^T y,\ Ax \leq b,\ x \geq 0,\ A^T y \geq c,\ y \geq 0$$

genau dann erfüllbar, wenn das primale Optimierungsproblem eine endliche Lösung hat. In diesem Fall ist jede Lösung des Ungleichungssystems Lösung des Optimierungsproblems. Wenn wir also allgemein für Ungleichungssysteme $Ax \leq b$ konstruktiv entscheiden können, ob sie lösbar sind, können wir konstruktiv Probleme der Linearen Optimierung lösen, die Optimierung erhalten wir also praktisch gratis.

Satz 2.11.1: *Jeder Algorithmus zur Lösung von Ungleichungssystemen $Ax < b + (2^{-L}, \ldots, 2^{-L})$ kann mit geringem Aufwand in einen Algorithmus zur Lösung von Ungleichungssystemen $Ax \leq b$ umgewandelt werden.*

Wir verzichten in dem Satz auf eine genauere Analyse des Mehraufwandes, da dieser, wie sich im Beweis zeigt, gegenüber dem Aufwand des Algorithmus von Khachiyan überhaupt nicht ins Gewicht fällt. Wir schieben noch den Beweis eines Lemmas ein.

Lemma 2.11.2: *Für $1 \leq i \leq m$ und $x \in \mathbb{R}^n$ sei $\Theta_i(x) := a_i^T x \Leftrightarrow b_i$. Für $x_0 \in \mathbb{R}^n$ läßt sich ein $x_1 \in \mathbb{R}^n$ konstruieren, so daß die folgenden Bedingungen gelten.*

- $\forall_{1 \leq i \leq m}$: $\Theta_i(x_0) \geq 0 \Longrightarrow \Theta_i(x_1) = \Theta_i(x_0)$.
- $\forall_{1 \leq i \leq m}$: $\Theta_i(x_0) < 0 \Longrightarrow \Theta_i(x_1) \leq 0$.
- *Jedes a_j läßt sich als Linearkombination der a_i mit $\Theta_i(x_1) \geq 0$ schreiben.*

Falls x das Ungleichungssystem $Ax \leq b$ erfüllt, gilt $\Theta_i(x) \leq 0$ für alle i. Wenn x_0 das System $Ax < b + (2^{-L}, \ldots, 2^{-L})$ erfüllt, gilt $\Theta_i(x_0) < 2^{-L}$ für alle i. Der neue Vektor x_1 ist keine große Verschlechterung (aus $<$ kann $\leq$ werden), dafür wird Struktur gewonnen, jedes a_j ist Linearkombination der schlechten a_i.

Beweis von Lemma 2.11.2: Wir numerieren die Ungleichungen, so daß $\Theta_1(x_0), \ldots, \Theta_k(x_0) \geq 0$ und $\Theta_{k+1}(x_0), \ldots, \Theta_m(x_0) < 0$. Wir wählen nun a_l mit $l > k$, so daß a_l keine Linearkombination aus $a_1, \ldots, a_k$ ist. Falls l nicht existiert, ist $x_1 := x_0$ als Lösung geeignet. Ansonsten ist, da a_l außerhalb des von $a_1, \ldots, a_k$ aufgespannten Raumes liegt, folgendes Gleichungssystem lösbar.

$$a_i^T y = 0,\ 1 \leq i \leq k, \text{ und } a_l^T y = 1.$$

Sei y_0 eine Lösung dieses Gleichungssystems und

$$\begin{aligned} x_1 &:= x_0 + t y_0 \\ t &:= \min\left\{ \Leftrightarrow\Theta_j(x_0)/a_j^T y_0 \mid a_j^T y_0 > 0 \text{ und } k < j \leq m \right\}. \end{aligned}$$

Für $k < j \leq m$ ist $\Theta_j(x_0) < 0$ nach Definition. Da $a_l^T y_0 = 1 > 0$, folgt $t > 0$ und

$$t \leq \Leftrightarrow\Theta_l(x_0)/a_l^T y_0 = \Leftrightarrow\Theta_l(x_0).$$

Es folgt, daß für alle $1 \leq i \leq m$ gilt:

$$\Theta_i(x_1) = \Theta_i(x_0 + t y_0) = a_i^T (x_0 + t y_0) \Leftrightarrow b_i = t a_i^T y_0 + \Theta_i(x_0).$$

Diese rechte Seite ist für $i \leq k$, da $a_i^T y_0 = 0$, gleich $\Theta_i(x_0)$, und nicht positiv für $i > k$: Dies gilt, falls $a_i^T y_0 > 0$ nach Definition von t, und sonst direkt. Für den Index j, der für die Definition von t entscheidend ist, gilt offensichtlich $\Theta_j(x_1) = 0$. Der Vektor $\Theta(x_1)$ ist also an mindestens einer Stelle mehr als $\Theta(x_0)$ nicht negativ. Vielleicht ist x_1 eine gewünschte Lösung, sonst fahren wir mit x_1 analog zu x_0 fort. Spätestens nach $m \Leftrightarrow k \leq m$ Iterationen sind wir am Ziel, da dann alle $\Theta_j(x_1) \geq 0$ sind. □

Beweis von Satz 2.11.1: Sei nun x_0 eine Lösung von $Ax < b + (2^{-L}, \ldots, 2^{-L})$, d. h. $\Theta_i(x_0) < 2^{-L}$ für alle $i = 1, \ldots, m$. Wir numerieren die Ungleichungen o. B. d. A. so, daß $a_i^T x_0 \geq b_i$ für $1 \leq i \leq k$ und $a_i^T x_0 < b_i$ für $k < i \leq m$ gilt. Außerdem sind $a_1, \ldots, a_r$ linear unabhängig und $a_{r+1}, \ldots, a_m$ Linearkombinationen von $a_1, \ldots, a_r$.

Wir ersetzen nun x_0 durch den im Beweis von Lemma 2.11.2 konstruierten Vektor x_1, benutzen jedoch die alten Bezeichnungen (z. B. wieder x_0 für x_1). Dann gilt zusätzlich, daß sich alle $a_{r+1}, \ldots, a_m$ als Linearkombinationen der Vektoren $a_1, \ldots, a_r$ schreiben lassen.

Wir betrachten eine Lösung z des Gleichungssystems $a_j^T x = b_j$, $1 \leq j \leq r$. Wegen der linearen Unabhängigkeit von $a_1, \ldots, a_r$ existiert z. Wir behaupten, daß $a_i^T z \leq b_i$ für alle i gilt. Falls $r = m$ ist, so ist nichts mehr zu zeigen. Im folgenden können wir also $r \leq m \Leftrightarrow 1$ annehmen.

Wir betrachten nun ein festes i: Es gibt $D_j \in \mathbb{Z}$, $D \in \mathbb{N}$ mit

$$a_i = \sum_{1 \leq j \leq r} \frac{D_j}{D} a_j.$$

Dabei können wir analog zu Lemma 2.10.1 zeigen, daß wir D_j und D dem Betrage nach kleiner als $2^L/mn$ wählen können. Da $D > 0$ ist, können wir $(a_i^T z \Leftrightarrow b_i) \leq 0$ beweisen, indem wir $D(a_i^T z \Leftrightarrow b_i) \leq 0$ beweisen. Hier geht nun der kleine, aber entscheidende Trick ein. Wir zeigen, daß $D(a_i^T z \Leftrightarrow b_i)$ einerseits ganzzahlig und andererseits kleiner als 1 ist. Daraus folgt dann $D(a_i^T z \Leftrightarrow b_i) \leq 0$.

Da $a_j^T z = b_j$ für $1 \leq j \leq r$, folgt

$$D(a_i^T z \Leftrightarrow b_i) = \sum_{1 \leq j \leq r} D_j a_j^T z \Leftrightarrow D b_i = \sum_{1 \leq j \leq r} D_j b_j \Leftrightarrow D b_i \in \mathbb{Z}.$$

Wir formen weiter um:

$$= \sum_{1 \leq j \leq r} D_j \left(a_j^T x_0 \Leftrightarrow \Theta_j(x_0)\right) \Leftrightarrow D \left(a_i^T x_0 \Leftrightarrow \Theta_i(x_0)\right).$$

Dabei gilt

$$\sum_{1 \leq j \leq r} D_j a_j^T x_0 = D \left(\sum_{1 \leq j \leq r} \frac{D_j}{D} a_j^T \right) x_0 = D a_i^T x_0.$$

Also ist

$$D(a_i^T z \Leftrightarrow b_i) = D\Theta_i(x_0) \Leftrightarrow \sum_{1 \leq j \leq r} D_j \Theta_j(x_0).$$

Nun gilt $\Theta_j(x_0) \geq 0$ für $1 \leq j \leq r$, $\Theta_j(x_0) < 2^{-L}$ für alle j und $\Leftrightarrow D_j \leq |D_j|$ für alle j. Damit folgt

$$D(a_i^T z \Leftrightarrow b_i) < D 2^{-L} + \sum_{1 \leq j \leq r} |D_j| \, 2^{-L}$$

$$< \frac{2^L}{mn} 2^{-L} + r \frac{2^L}{mn} 2^{-L} = \frac{r+1}{mn} \leq \frac{m}{mn} < 1.$$

In der letzten Ungleichung haben wir $n \geq 2$ vorausgesetzt, aber sonst ist das Problem trivial. □

Wir sehen, daß wir geschickt die Ganzzahligkeit der Eingabeparameter ausgenutzt haben. Mit reellen Parametern wären die Beweise nicht durchführbar.

Wir fassen zusammen, wie wir Probleme der Linearen Optimierung mit Hilfe des Algorithmus von Khachiyan lösen können. Wir betrachten das Problem $c^T x \to \max$ mit $Ax \leq b$, $x \geq 0$. Zunächst benutzen wir die obigen Konstruktionen und den Algorithmus von Khachiyan, um zu testen, ob $Ax \leq b$, $x \geq 0$ lösbar ist. Damit stellen wir fest, ob die zulässige Menge leer ist. Danach wenden wir den gleichen Ansatz auf das größere System $c^T x \geq b^T y$, $Ax \leq b$, $x \geq 0$, $A^T y \geq c$, $y \geq 0$ an. Entweder wir erhalten eine optimale Lösung für unser Problem, oder unser Problem ist unbeschränkt.

Komplexitätstheoretisch fällt ins Gewicht, daß wir in dem $<$-System die Zahlen 2^{-L} addiert haben. Wir müssen also mit 2^L multiplizieren, um wieder ein ganzzahliges System zu erhalten. Wir arbeiten also mit sehr großen Zahlen. Um aus Lösungen des $<$-Systems Lösungen des $\leq$-Systems zu machen, ist nur geringer Mehraufwand nötig.

Ist der Algorithmus numerisch stabil? Es wird ja nicht nur dividiert, sondern es werden auch Wurzeln gezogen. Wir können immer so runden, daß die Ellipsoide nicht verkleinert werden. Dann genügen $10(n+1)^2 L$ Iterationen bei Zahlen der Länge $300(n+1)L$, wenn die Einträge in B_{neu} jeweils mit $2^{1/(4(n+1)^2)}$ multipliziert werden.

Der Algorithmus von Khachiyan ist hier nur dargestellt worden, um im Vergleich zum Simplex-Algorithmus zu zeigen, daß das gleiche Problem mit völlig verschiedenen Methoden attackiert werden kann. Der Algorithmus von Khachiyan hat keine praktische Bedeutung, während der Algorithmus von Karmarkar bei manchen recht großen Problemen den Simplex-Algorithmus geschlagen hat. Für die Praxis kann aus heutiger Sicht nur der Simplex-Algorithmus empfohlen werden. Allerdings muß die Hoffnung auf bessere Algorithmen nicht aufgegeben werden.

3 Semidefinite Programmierung

In der Informatik sind viele wichtige Probleme *NP*–hart und somit vermutlich nicht auf effiziente Weise lösbar. Oft gibt man sich für solche Probleme dann mit Algorithmen zufrieden, die nicht die optimale Lösung finden, sondern nur eine „gute“ Lösung. Ein Approximationsalgorithmus garantiert, daß die gefundene Lösung nicht viel schlechter ist als die optimale Lösung. Angenommen, wir haben ein Maximierungsproblem. Ein 1/2–Approximationsalgorithmus für dieses Problem ist dann z.B. ein Algorithmus, der Lösungen abliefert, die mindestens halb so groß sind wie das Optimum. In den letzten Jahren (seit 1994) hat es eine interessante Entwicklung gegeben. Es ist den Forschern und Forscherinnen gelungen, bessere Approximationsalgorithmen zu entwerfen, und zwar für eine Reihe von Problemen. Diese Algorithmen basieren auf einer Verallgemeinerung der Linearen Programmierung. Es handelt sich um die sogenannte Semidefinite Programmierung. Wir wollen in diesem Kapitel den Lesern und Leserinnen einen kurzen Eindruck geben, worum es sich bei der Semidefiniten Programmierung handelt und nur erwähnen, daß es Polynomialzeitalgorithmen gibt, die Probleme der Semidefiniten Programmierung lösen können. Diese Algorithmen basieren auf einer Verallgemeinerung von „Innere–Punkt–Methoden“, die ursprünglich für die Lineare Programmierung entwickelt worden sind.

Um ein Beispiel abzuliefern, wie man die Semidefinite Programmierung für in der Informatik auftretende Probleme verwenden kann, schauen wir uns auch das Beispiel an, das gewissermaßen die Forschungslawine in den letzten Jahren ausgelöst hat, nämlich das Problem MAXCUT.

Zunächst benötigen wir aber ein paar Begriffe aus der Matrixtheorie bzw. der linearen Algebra. Diese kann man z.B. auch in einem Buch von G.H. Golub und C.F. van Loan nachlesen: *Matrix Computations, North Oxford Academic, 1986.*

3.1 Grundlagen der Matrixtheorie

Wir gehen in diesem Kapitel davon aus, daß die betrachteten Matrizen nur Einträge aus den reellen Zahlen haben (also nicht etwa komplexe Zahlen).

Eine $n \times m$–Matrix heißt *quadratisch*, wenn $n = m$ ist. Eine Matrix A mit $A = A^T$ heißt *symmetrisch*. Id bezeichnet in diesem Kapitel die Identitätsmatrix.

Definition 3.1.1: Ein *Eigenvektor* einer Matrix A ist ein Vektor $x \neq 0$, für den ein $\lambda \in \mathbb{C}$ existiert mit $A \cdot x = \lambda \cdot x$. Der Wert λ heißt *Eigenwert*.

Eigenwerte lassen sich auch noch anders charakterisieren:

Lemma 3.1.2: *Die Eigenwerte einer Matrix A sind die Nullstellen des Polynoms* $\det(A \Leftrightarrow \lambda \cdot Id)$*, wobei λ die Variable des Polynoms ist.*

Für symmetrische $n \times n$–Matrizen A ist folgendes bekannt: A besitzt n reelle Eigenwerte (die nicht notwendigerweise alle voneinander verschieden sind.) Diese n Eigenwerte werden häufig geschrieben als $\lambda_1, \ldots, \lambda_n$, wobei man eine kanonische Numerierung annimmt, so daß $\lambda_1 \geq \ldots \geq \lambda_n$ gilt.

Wenn ein Eigenwert mehrmals vorkommt, dann wird die Anzahl seines Vorkommens auch *Multiplizität* genannt.

Satz 3.1.3 (Rayleigh-Ritz): *Wenn A eine symmetrische Matrix ist, dann gilt*

$$\lambda_1(A) = \max_{||x||=1} x^T \cdot A \cdot x \qquad \textit{und} \qquad \lambda_n(A) = \min_{||x||=1} x^T \cdot A \cdot x.$$

Definition 3.1.4: Eine quadratische Matrix A heißt

$$\left.\begin{array}{lll} \textit{positiv semidefinit,} & \text{wenn} & x^T \cdot A \cdot x \geq 0 \\ \textit{positiv definit,} & \text{wenn} & x^T \cdot A \cdot x > 0 \end{array}\right\} \text{für alle } x \in \mathbb{R}^n \setminus \{\underline{0}\} \text{ ist.}$$

In diesem Kapitel wollen wir den Begriff „positiv semidefinit“ auch abkürzen als „PSD“.

Als Korollar von Satz 3.1.3 erhalten wir, daß eine symmetrische Matrix PSD ist genau dann, wenn ihr kleinster Eigenwert $\lambda_n \geq 0$ ist.

Wir wollen nun einige Beispiele für Matrizen bringen, die PSD sind. Zunächst sollten wir festhalten, daß eine Matrix, die nur positive Einträge hat, nicht unbedingt PSD ist: Wir betrachten die folgenden beiden Matrizen:

$$A := \begin{pmatrix} 1 & 4 \\ 4 & 1 \end{pmatrix} \qquad B := \begin{pmatrix} 1 & \Leftrightarrow 1 \\ \Leftrightarrow 1 & 4 \end{pmatrix}.$$

Die Matrix A ist nicht PSD. Dies sieht man zum Beispiel daran, daß $(1, \Leftrightarrow 1) \cdot A \cdot (1, \Leftrightarrow 1)^T = \Leftrightarrow 6$ ist. Wenn man die Eigenwerte von A ausrechnet, findet man $\lambda_1 = 5, \lambda_2 = \Leftrightarrow 3$.

Es gilt zum Beispiel $A \cdot (1,1)^T = (5,5)^T$ und $A \cdot (1, \Leftrightarrow 1)^T = (\Leftrightarrow 3, 3)^T$.

Die Matrix B enthält negative Einträge, ist aber „trotzdem“ PSD. Dies liegt daran, daß die beiden Eigenwerte von B ungefähr die Werte 4.30 and 0.697 haben, also beide echt größer als Null sind.

Für eine Diagonalmatrix sind die Eigenwerte identisch mit den Einträgen auf der Diagonale. Eine Diagonalmatrix ist also genau dann PSD, wenn alle ihre Diagonaleinträge nicht kleiner als Null sind.

Aus der Definition der Semidefinitheit folgt auch, daß folgendes gilt: Wenn B und C Matrizen sind, dann ist die Matrix

$$A = \begin{pmatrix} B & 0 \\ 0 & C \end{pmatrix}$$

PSD genau dann, wenn B und C beide PSD sind. Dies bedeutet, daß man eine Bedingung der Form „die Matrizen $A_1, \ldots, A_r$ sollen PSD sein“ auch ausdrücken kann als „die Matrix A^* soll PSD sein“, wenn man die Matrix A^* analog zu gerade aus den A_i zusammensetzt.

Aus der Definition folgt auch: Wenn A PSD ist, dann ist die $k \times k$–Submatrix, die man aus A enthält, indem man die Spalten und Zeilen $k+1$ bis n eliminiert, auch PSD.

Als letztes Beispiel betrachten wir die folgende symmetrische Matrix:

$$\begin{pmatrix} 1 & a & \ldots & a \\ a & 1 & \ldots & a \\ \ldots & \ldots & \ldots & \ldots \\ a & a & \ldots & 1 \end{pmatrix},$$

die Einsen auf der Diagonalen hat und eine Konstante a an allen anderen Stellen. Eine einfache Rechnung (Übungsaufgabe!) zeigt, daß die Eigenwerte dieser Matrix die beiden Zahlen $1+a\cdot(n \Leftrightarrow 1)$ (mit Multiplizität 1) und $1 \Leftrightarrow a$ (mit Multiplizität $n \Leftrightarrow 1$) sind. Die Matrix ist also PSD, wenn wir a so wählen, daß $\Leftrightarrow 1/(n \Leftrightarrow 1) \leq a \leq 1$ gilt.

In diesem Kapitel haben wir meistens mit symmetrischen Matrizen zu tun. Für symmetrische Matrizen gilt die folgende Äquivalenzaussage:

Lemma 3.1.5: *Sei A eine symmetrische $n \times n$–Matrix. Die folgenden Aussagen sind äquivalent:*

- *A ist PSD.*
- *Alle Eigenwerte von A sind nicht–negativ.*
- *Es gibt eine $n \times n$–Matrix B, so daß man A schreiben kann als $A = B^T \cdot B$.*

Die Zerlegung $A = B^T \cdot B$ kann man auch anders lesen: Es gibt n Vektoren $b_1, \ldots, b_n$, so daß jedes Element $A_{i,j} = b_i^T \cdot b_j$ ist. Die Vektoren b_i entsprechen den Spalten in B.

Wenn wir eine symmetrische Matrix A haben, die PSD ist, dann ist die Zerlegung $A = B^T \cdot B$ nicht eindeutig, weil das Skalarprodukt von Vektoren rotationsinvariant ist, also liefert die Rotation der Vektoren,

die durch die Matrix B beschrieben werden, eine neue Zerlegung. Insbesondere kann man B immer so wählen, daß B eine obere (oder untere) Dreiecksmatrix ist.

Wenn man eine symmetrische Matrix A gegeben hat, die PSD ist, dann interessiert man sich eventuell für eine zugehörige Zerlegung. Diese bezeichnet man auch als „(unvollständige) Cholesky–Zerlegung". Es gibt einen Algorithmus, der eine solche Cholesky–Zerlegung in Laufzeit $O(n^3)$ berechnet, man lese zum Beispiel im oben erwähnten Buch von Golub und van Loan nach.

Zum Beispiel hat die folgende Matrix A, die PSD ist, die angegebene Cholesky–Zerlegung:

$$A = \begin{pmatrix} 1 & \Leftrightarrow 1 \\ \Leftrightarrow 1 & 4 \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ \Leftrightarrow 1 & \sqrt{3} \end{pmatrix} \cdot \begin{pmatrix} 1 & \Leftrightarrow 1 \\ 0 & \sqrt{3} \end{pmatrix}.$$

Der erwähnte Algorithmus bemerkt auch, wenn die vorgelegte symmetrische Matrix *nicht* PSD ist.

Wen es nun stört, daß wir zwar für symmetrische Matrizen entscheiden können, ob sie PSD sind oder nicht, aber anscheinend nicht für beliebige Matrizen, der möge durch folgende Überlegung beruhigt werden:

Das Problem, zu entscheiden, ob eine gegebene quadratische Matrix A PSD ist, kann zurückgeführt werden auf das gleiche Problem für eine symmetrische Matrix A'. Wir können nämlich A' wie folgt definieren: $A'_{i,j} := \frac{1}{2} \cdot (A_{i,j} + A_{j,i})$. Da

$$x^T \cdot A \cdot x \;=\; \sum_{i,j} A_{i,j} \cdot x_i \cdot x_j$$

ist, gilt, daß $x^T \cdot A' \cdot x = x^T \cdot A \cdot x$ ist und somit ist die symmetrische Matrix A' genau dann PSD, wenn die Matrix A PSD ist.

3.2 Semidefinite Programme

In der Literatur gibt es verschiedene Definitionen von Semidefiniten Programmen. Wie man jedoch nicht anders erwarten sollte, sind diese Definitionen (oft oder immer?) äquivalent. Wir wählen die folgende Definition:

Definition 3.2.1: Ein *semidefinites Programm* hat die folgende Form: Wir suchen eine Lösung in den reellen Variablen $x_1, \ldots, x_m$. Gegeben ist ein Vektor $c \in \mathbb{R}^m$, und wir wollen die Zielfunktion $c^T \cdot x$ minimieren (oder maximieren). Der zulässige Bereich für die Belegung der x–Variablen ist beschrieben durch eine symmetrische Matrix SP, die als Einträge lineare Funktionen in den x_i–Variablen enthält. Genauer gesagt: Ein Vektor a ist zulässig, wenn die Matrix SP PSD wird sobald wir die Variablen x_i in SP mit den Werten aus a belegen, also $x_i = a_i$ setzen.

Hier ist ein Beispiel eines semidefiniten Programms:

$$\begin{array}{rl} \textbf{max} & x_1 + x_2 + x_3 \\ \textbf{so daß} & \begin{pmatrix} 1 & x_1 \Leftrightarrow 1 & 2x_1 + x_2 \Leftrightarrow 1 \\ x_1 \Leftrightarrow 1 & \Leftrightarrow x_1 & x_2 + 3 \\ 2x_1 + x_2 \Leftrightarrow 1 & x_2 + 3 & 0 \end{pmatrix} \text{ PSD ist.} \end{array}$$

Wie bei der Linearen Programmierung ist es nicht wesentlich, ob wir $x_i \geq 0$ verlangen oder nicht. Wenn wir ein Problem als semidefinites Programm modellieren möchten, muß sich nicht unbedingt eine symmetrische Matrix ergeben. Entsprechend der vorhin gemachten Bemerkungen kann man „unsymmetrische Bedingungen" durch symmetrische Bedingungen ausdrücken.

Die Matrix, die wir erhalten, wenn wir die Variablen x_i gemäß dem Vektor a belegen und in SP einsetzen, bezeichnen wir hier mit $SP(a)$.

Wir bemerken, daß es Methoden gibt, ein semidefinites Programm zu lösen. Genauer: Zu jedem gegebenen $\varepsilon > 0$ und jedem semidefiniten Programm, das eine endliche, polynomiell beschränkte Lösung hat,

kann man das Programm in Polynomialzeit lösen, bis auf einen Fehler (in der Zielfunktion) von ε. Im allgemeinen läßt sich so ein Fehlerterm nicht vermeiden, weil die optimale Lösung eines semidefiniten Programms irrational sein kann.

Es mag auch überraschen, daß man wissen muß, daß das Programm eine polynomiell beschränkte Lösung hat. Warum so etwas aber wichtig ist, sehen wir an folgendem Beispiel: Während wir für ein lineares Programm von vornherein wissen, daß bei gegebener Größe L des linearen Programms ein Lösungspunkt existiert, dessen Koordinaten betragsmäßig durch 2^L beschränkt sind, so gilt dies bei der Semidefiniten Programmierung nicht mehr. Betrachte folgendes Beispiel:

$$\textbf{min } x_n \textbf{ so daß } x_1 = 2 \text{ und } x_i \geq x_{i-1}^2.$$

Die optimale Lösung dieses Programms ist natürlich $2^{2^{n-1}}$ und es handelt sich um ein semidefinites Programm, weil man die Bedingung $x_i \geq x_{i-1}^2$ in einem semidefiniten Programm formulieren kann (Übungsaufgabe). Allein zur Ausgabe der Lösung würden wir aber schon Laufzeit $\Omega(2^n)$ benötigen.

Wir haben erwähnt, daß die Semidefinite Programmierung eine Verallgemeinerung der Linearen Programmierung ist. Dies wollen wir beweisen:

Lemma 3.2.2: *Die Lineare Programmierung ist ein Spezialfall der Semidefiniten Programmierung.*

Beweis: Wir schreiben die Bedingungen des linearen Programms wie folgt auf. Die i–te Bedingung sei durch $IN_i \geq 0$ gegeben, wobei

$$IN_i := a_{i,1}x_1 + \ldots + a_{i,n}x_n + b_i$$

ist, für $1 \leq i \leq r$. Wir definieren die Matrix SP für das semidefinite Programm wie folgt:

$$\begin{pmatrix} IN_1 & 0 & \cdots & 0 \\ 0 & IN_2 & \cdots & 0 \\ \cdots & \cdots & \cdots & \cdots \\ 0 & 0 & \cdots & IN_r \end{pmatrix}.$$

Da eine Diagonalmatrix genau dann PSD ist, wenn alle ihre Diagonalelemente nicht–negativ sind, erhalten wir, daß der zulässige Bereich des linearen Programms und des semidefiniten Programms gleich sind. □

Lineare Programmierung kann also nach diesem Beweis angesehen werden als der Spezialfall der Semidefiniten Programmierung, bei dem die beteiligte Matrix SP eine Diagonalmatrix ist.

Viele Eigenschaften, die für die Lineare Programmierung gelten, lassen sich auf die Semidefinite Programmierung übertragen. Zum Beispiel ist der zulässige Bereich eines semidefiniten Programms auch konvex:

Gegeben zwei zulässige Vektoren a und b und die entsprechenden Matrizen $SP(a)$ und $SP(b)$. Alle Vektoren, die auf der Verbindungsstrecke zwischen a und b liegen, sind dann auch zulässig: Für alle $0 \leq \lambda \leq 1$ gilt nämlich:

$$SP\left((1 \Leftrightarrow \lambda)x + \lambda y\right) = (1 \Leftrightarrow \lambda)SP(x) + \lambda SP(y).$$

Diese Matrix ist auch PSD, wie man wie folgt sehen kann:

$$v^T \cdot SP((1 \Leftrightarrow \lambda)x + \lambda y) \cdot v = v^T \cdot (1 \Leftrightarrow \lambda) \cdot SP(x) \cdot v + v^T \cdot \lambda \cdot SP(y) \cdot v \geq 0.$$

Als eine Konsequenz aus dieser Konvexitätseigenschaft wollen wir hier festhalten, daß man z.B. ein semidefinites Programm nicht benutzen kann, um eine Bedingung wie zum Beispiel „$x \geq 1$ oder $x \leq \Leftrightarrow 1$“ ausdrücken kann, da der zulässige Bereich für diese Menge nicht konvex ist.

Wir wollen an einem Beispiel zeigen, daß man mit Hilfe eines Semidefiniten Programms mehr Bedingungen ausdrücken kann als mit Hilfe eines Linearen Programms.

Wir betrachten zum Beispiel die symmetrische 2×2–Matrix

$$\begin{pmatrix} a & b \\ b & c \end{pmatrix}.$$

Diese Matrix ist genau dann PSD, wenn $a \geq 0, c \geq 0$, und $b^2 \leq ac$ ist. Zum Beispiel beschreibt damit die folgende Matrix

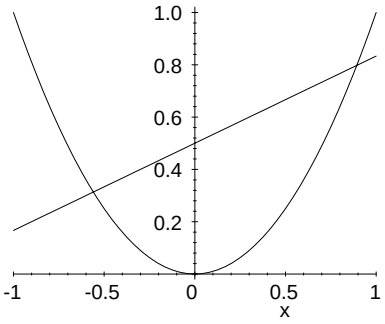
$$\begin{pmatrix} x_1 & 2 \\ 2 & x_2 \end{pmatrix}$$

als zulässigen Bereich denjenigen, wo für die Variablen x_1 und x_2 gilt, daß $x_1 \cdot x_2 \geq 4, x_1 \geq 0, x_2 \geq 0$ ist. Hier ist die Bedingung $x_1 \cdot x_2 \geq 4$ nicht linear.

Ein weiteres Beispiel: Wenn wir im semidefiniten Programm die Matrix

$$\begin{pmatrix} 1 & x & 0 \\ x & y & 0 \\ 0 & 0 & \Leftrightarrow y + \frac{x}{3} + \frac{1}{2} \end{pmatrix}$$

wählen, dann sind genau alle Punkte x, y mit $y \geq 0, y \geq x^2$ und $y \leq x/3 + 1/2$ zulässig. Diese Punkte liegen im folgenden Bild alle zwischen der Parabel und der Geraden. Wir überlassen es dem Leser und der Leserin, die optimale Lösung dieses semidefiniten Programms zu bestimmen, wenn man konkrete Zielfunktionen wählt.



3.3 Eine Anwendung der Semidefiniten Programmierung in der Informatik

Das Problem MAXCUT ist wie folgt definiert:

Problem MAXCUT
Gegeben ein ungerichteter Graph $G = (V, E)$. Finde eine Zerlegung der Knotenmenge $V = V_1 \cup V_2$ (mit $V_1 \cap V_2 = \emptyset$), so daß die Anzahl der Kanten aus E, die zwischen V_1 und V_2 liegen, maximal wird.

MAXCUT ist ein *NP*–hartes Problem, also ist man schon glücklich, wenn man Lösungen, also Zerlegungen berechnen kann, die relativ gut sind, wenn auch nicht optimal.

Für ein $0 \leq c \leq 1$ heiße ein Approximationsalgorithmus A für MAXCUT c–Approximationsalgorithmus, wenn die von A berechnete Lösung mindestens c mal soviele Kanten enthält wie die optimale Lösung. Bis ca. 1994 waren nur 1/2–Approximationsalgorithmen für MAXCUT bekannt. So war es quasi eine Sensation, als es den beiden Forschern Goemans und Williamson gelang, einen Algorithmus zu entwerfen, der ein $0.878\ldots$–Approximationsalgorithmus für MAXCUT ist. Insbesondere haben sie gezeigt, daß die Semidefinite Programmierung eine interessante Methode ist, um zu Approximationsalgorithmen zu gelangen, und entsprechend haben sie eine Lawine ausgelöst, in der viele weitere Probleme daraufhin untersucht wurden, ob sie mit Hilfe der Semidefiniten Programmierung besser gelöst werden können.

Wir beobachten zunächst, daß man die optimale Lösung eines MAXCUT–Problems als Lösung des folgenden mathematischen Programms erhalten kann:

$$\textbf{max} \sum_{\{i,j\}\in E} \frac{1 \Leftrightarrow y_i y_j}{2} \quad \textbf{so daß} \quad y_i \in \{\Leftrightarrow 1, 1\} \text{ ist für alle } 1 \leq i \leq n.$$

Die Idee hinter dieser Formulierung ist, daß man $V_1 = \{i \mid y_i = 1\}$ und $V_2 = \{i \mid y_i = \Leftrightarrow 1\}$ als die beiden Klassen der Zerlegung wählen kann. Ein Term $(1 \Leftrightarrow y_i y_j)/2$ trägt 1 zur Summe bei genau dann, wenn $y_i \neq y_j$ ist und 0 sonst.

Wir wollen dieses mathematische Programm „relaxieren“, d.h. den Raum der zulässigen Lösungen zunächst größer machen. Wir machen den Raum der zulässigen Lösungen dabei so groß, daß wir das neue Programm mit Hilfe der Semidefiniten Programmierung lösen können. Wir relaxieren das Programm dahingehend, daß wir von den Variablen y_i nicht verlangen, daß sie 1–dimensional sind, sondern, wir lassen zu, daß sie n–dimensional sind. Wir erhalten das folgende mathematische Programm:

$$\textbf{max} \sum_{\{i,j\}\in E} \frac{1 \Leftrightarrow \underline{y_i} \underline{y_j}}{2} \quad \textbf{so daß} \quad ||\underline{y_i}|| = 1, \underline{y_i} \in \mathbb{R}^n \text{ ist für alle } 1 \leq i \leq n.$$

(Ausnahmsweise stellen wir in diesem Unterkapitel einmal die Vektoren durch unterstrichene Buchstaben dar, um sie deutlicher von Integervariablen zu unterscheiden.)

Das gerade gezeigte Programm ist noch kein semidefinites Programm, aber indem wir neue Variablen einführen, erhalten wir ein semidefinites Programm:

$$\textbf{max} \sum_{\{i,j\}\in E} \frac{1 \Leftrightarrow y_{i,j}}{2} \quad \textbf{so daß} \quad \begin{pmatrix} 1 & y_{1,2} & y_{1,3} & \cdots & y_{1,n} \\ y_{1,2} & 1 & y_{2,3} & \cdots & y_{2,n} \\ y_{1,3} & y_{2,3} & 1 & \cdots & y_{3,n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ y_{1,n} & y_{2,n} & y_{3,n} & \cdots & 1 \end{pmatrix} \text{ PSD ist.}$$

Der Grund ist der folgende: Wegen der dritten Äquivalenz in Lemma 3.1.5 definiert die Matrix einen zulässigen Bereich, in dem jede Variable $y_{i,j}$ geschrieben werden kann als das Produkt von Vektoren $\underline{v_i} \cdot \underline{v_j}$. Die Diagonale in der Matrix garantiert dabei, daß $||\underline{v_i}|| = 1$ für alle i ist.

Dies ist eine Relaxation des ursprünglichen Problems in einer Dimension, denn jeder zulässige Punkt mit Zielfunktionswert Z für das 1–dimensionale Problem kann als zulässiger Punkt mit Zielfunktionswert Z für das n–dimensionale Problem gesehen werden: Wir füllen die anderen Komponenten einfach mit Nullen.

Wir entwerfen nun folgenden randomisierten Algorithmus, um eine Zerlegung $V_1 \cup V_2$ zu bekommen.

Zunächst wenden wir einen Algorithmus an, um eine Lösung des obigen semidefiniten Programms zu finden, die optimal ist. (Bzw. bis auf einen kleinen additiven Term ε optimal ist).

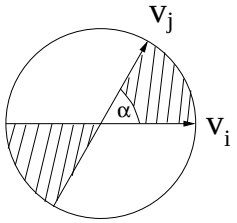
Wir fahren dann wie folgt fort:

- Berechne eine Cholesky–Zerlegung und erhalte so Vektoren $\underline{v_i}$, $i = 1, \ldots, n$, für die $y_{i,j} = \underline{v_i} \cdot \underline{v_j}$ gilt. Dies kann in Laufzeit $O(n^3)$ geschehen.
- Wähle zufällig eine Hyperebene H aus. Dies kann zum Beispiel dadurch geschehen, daß man einen Zufallsvektor $\underline{r}$ als Normale der Hyperebene auswürfelt. Zu diesem Zweck benutzt man eine rotationssymmetrische Verteilung.
- Wir wählen V_1 als die Menge all derjenigen Knoten aus V, deren zugehöriger Vektor auf einer Seite der Hyperebene H liegt. Also: $v_i \in V_1 \Longleftrightarrow \underline{r} \cdot \underline{v_i} \leq 0$. Außerdem wählen wir $V_2 := V \setminus V_1$.

Der Vorgang, aus den Vektoren $\underline{v_i}$ mittels der Wahl einer Hyperebene Elemente aus $\{\Leftrightarrow 1, 1\}$ zu machen, wird auch „Rundungsprozedur“ genannt.

Der gerade beschriebene Algorithmus ist ein randomisierter Algorithmus, er hängt also von Zufallsentscheidungen ab, nämlich von der Wahl der Hyperebene. Die berechnete Zerlegung ist eine Zufallsvariable. Zufallsentscheidungen können auch schlecht ausgehen, das heißt, wir bekommen nicht unbedingt jedes Mal eine gute Zerlegung. Was wir aber zeigen können, ist folgendes: Die erwartete Kantenzahl der zufällig berechneten Zerlegung ist mindestens 87% der Kantenzahl in einer optimalen Zerlegung. Dieser Aufgabe wollen wir uns nun widmen.

Wegen der Linearität des Erwartungswertes brauchen wir nur für zwei gegebene Vektoren $\underline{v_i}$ und $\underline{v_j}$ die Wahrscheinlichkeit auszurechnen, daß sie auf verschiedenen Seiten der gewählten Hyperebene liegen. Da das Produkt $\underline{v_i} \cdot \underline{v_j}$ rotationsinvariant ist, können wir die Ebene betrachten, die von den beiden Vektoren aufgespannt wird:



(Der Winkel $\alpha = \arccos(\underline{v_i} \cdot \underline{v_j})$ zwischen den Vektoren $\underline{v_i}$ und $\underline{v_j}$ ist rotationsinvariant.)

Als erstes beobachten wir, daß die Wahrscheinlichkeit, daß die gewählte Hyperebene H identisch mit der von den Vektoren aufgespannten Ebene ist, gleich Null ist.

In den anderen Fällen schneiden sich die beiden Ebenen in einer Geraden. Wenn die beiden Vektoren auf verschiedenen Seiten von H liegen sollen, dann müssen sie auch auf verschiedenen Seiten der Geraden liegen. Also muß die Gerade „zwischen“ $\underline{v_i}$ und $\underline{v_j}$ liegen, also in den Bereich fallen, der im Bild schraffiert ist. Dies passiert jedoch genau mit Wahrscheinlichkeit

$$\frac{2\alpha}{2\pi} = \frac{\alpha}{\pi} = \frac{\arccos(\underline{v_i} \cdot \underline{v_j})}{\pi}.$$

Also ist die erwartete Zahl an Kanten, die zwischen V_1 und V_2 liegen („Schnittkanten“ genannt) in der zufälligen Zerlegung $V = V_1 \cup V_2$ wie folgt zu berechnen:

$$\mathbb{E}[\text{AnzahlSchnittkanten}] \quad = \quad \sum_{\{i,j\} \in E} \frac{\arccos(\underline{v_i} \cdot \underline{v_j})}{\pi}. \tag{1}$$

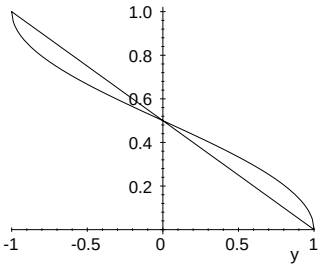
Wenn wir eine konkrete semidefinite Relaxation des Problems MAXCUT berechnen, erhalten wir natürlich eine obere Schranke für die optimale MAXCUT–Lösung. Die Güte der Lösung hängt vom Ausgang der Rundungsprozedur ab, wir berechnen die erwartete Güte. Im konkreten Fall kann man die erhaltene Lösung mit der oberen Schranke vergleichen und so direkt die Güte der erhaltenen Lösung abschätzen. Entsprechend kann man bei Bedarf weitere Zufallsversuche vornehmen.

Als Erfahrungswert sollte man auch im Hinterkopf behalten, daß Experimente zeigen, daß in den „meisten“ Fällen die berechnete Lösung sogar eine Güte hat, die viel besser ist als die von uns bewiesenen $0.878\ldots$.

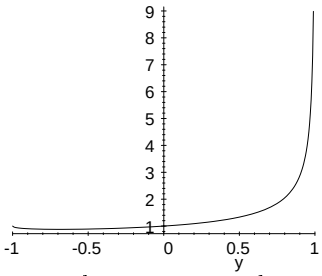
Wir wollen den Erwartungswert in Ausdruck (1) mit dem Wert des relaxierten Programms vergleichen. Eine Möglichkeit, dies zu tun, besteht darin, jeden einzelnen Term $\arccos(\underline{v_i} \cdot \underline{v_j})/\pi$ mit $(1 \Leftrightarrow \underline{v_i} \cdot \underline{v_j})/2$ zu vergleichen. Es ist eine relativ einfache Analysisaufgabe, zu zeigen, daß im Bereich $\Leftrightarrow 1 \leq y \leq 1$

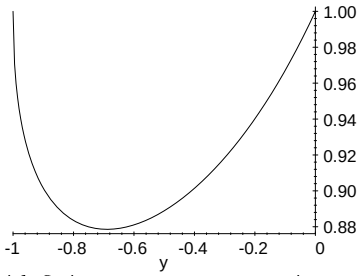
$$\frac{\arccos(y)}{\pi} \geq 0.87856 \cdot \frac{1 \Leftrightarrow y}{2} \quad \text{gilt.}$$

Im folgenden wollen wir die beteiligten Funktionen bildlich darstellen: Wir sehen im folgenden Bild die beiden Funktionen $\arccos(y)/\pi$ und $(1 \Leftrightarrow y)/2$.



Die folgenden beiden Bilder zeigen den Quotienten $\frac{\arccos(y)}{\pi} / \frac{1-y}{2}$ im Bereich $\Leftrightarrow 1 \leq y \leq 1$ (links) und $\Leftrightarrow 1 \leq y \leq 0$ (rechts).





Es bezeichne $Relax_{opt}$ den optimalen Wert der Zielfunktion des Semidefiniten Programms. Die gerade gemachte Analyse zeigt, daß der Erwartungswert der Anzahl der gefundenen Schnittkanten mindestens $0.878 \cdot Relax_{opt}$ ist, also ist die Güte der produzierten Lösung nicht schlechter als $1/0.878 \approx 1.139$.

Zum Schluß wollen wir betonen, daß wir gerade zwar einen randomisierten Algorithmus für die Rundungsprozedur beschrieben haben, daß aber auch deterministische Algorithmen mit der gleichen Güte bekannt sind. Diese sind jedoch wesentlich schwieriger zu beschreiben als die obige Rundungsprozedur.

4 Ganzzahlige Optimierung

4.1 Das Problem und eine komplexitätstheoretische Klassifizierung

Das Problem der Ganzzahligen Optimierung ist wie die Lineare Optimierung das Problem, eine lineare Zielfunktion unter linearen Nebenbedingungen zu minimieren (oder maximieren), wobei allerdings nur Lösungsvektoren aus $\mathbb{Z}^n$ erlaubt sind. In diesem Zusammenhang ist es sinnvoll, auch die Eingabeparameter als ganzzahlig anzunehmen. Wir wählen eine etwas andere Notation als in Kapitel 2, da dies einer einfacheren Beschreibung des Algorithmus dient.

Für $a_{ij} \in \mathbb{Z}$ $(0 \leq i \leq m, 0 \leq j \leq n)$ ist die Funktion $Z(x_1, \ldots, x_n) := \sum_{1 \leq\ j \leq n} a_{0j} x_j \Leftrightarrow a_{00}$ zu minimieren auf der Menge aller $(x_1, \ldots, x_n)$ mit

$$\begin{array}{rcll} \sum_{1 \leq j \leq n} a_{ij} x_j & \leq & a_{i,0} & 1 \leq i \leq m \\ x_1, \ldots, x_n & \geq & 0 & \\ x_1, \ldots, x_n & \in & \mathbb{Z} & \end{array}$$

Im Gegensatz zu Kapitel 2 werden die Parameter der Zielfunktion in die „nullte“ Zeile geschrieben und die Parameter der rechten Seiten der Nebenbedingungen in die nullte Spalte. Daß wir nun einen optimalen Vektor in einer wesentlich kleineren Menge suchen, mag zu der Vermutung Anlaß geben, daß das Problem der Ganzzahligen Optimierung einfacher als das Problem der Linearen Optimierung ist. Dies ist nicht der Fall. Wir wissen, daß es polynomielle (allerdings noch keine stark polynomiellen) Algorithmen für die Lineare Optimierung gibt. Dagegen ist das Problem der Ganzzahligen Optimierung NP-hart, d. h. polynomielle Algorithmen sind nur möglich, wenn die NP≠P-Hypothese falsch ist. Das bekannte Rucksackproblem erweist sich schnell als Spezialfall der Ganzzahligen Optimierung. Beim Rucksackproblem gibt es einen Rucksack mit Gewichtsbeschränkung G. Aus n Objekten mit Gewichten $g_1, \ldots, g_n$ und Nutzenwerten $c_1, \ldots, c_n$ soll eine Auswahl mit maximalem Nutzen getroffen werden, wobei das Gewichtslimit respektiert wird. Dieses Problem läßt sich folgendermaßen formalisieren.

$$\begin{array}{rcll} c_1 x_1 + \ldots + c_n x_n & \rightarrow & \max & \text{mit} \\ g_1 x_1 + \ldots + g_n x_n & \leq & G & \\ x_1, \ldots, x_n & \leq & 1 & \\ x_1, \ldots, x_n & \geq & 0 & \\ x_1, \ldots, x_n & \in & \mathbb{Z}. & \end{array}$$

Die letzten drei Bedingungen sind äquivalent zu $x_1, \ldots, x_n \in \{0, 1\}$. Da die Variablen nur zwei Werte annehmen können, wird dann auch von der Binären Optimierung gesprochen. Das Rucksackproblem ist also das Problem der Binären Optimierung eingeschränkt auf eine Nebenbedingung. Das Rucksackproblem ist NP-äquivalent, d. h. es gibt polynomielle Algorithmen genau dann, wenn NP=P ist. Um zu zeigen, welche Spezialfälle bereits NP-äquivalent sind, schränken wir das Rucksackproblem weiter ein. Zunächst setzen wir $c_i = g_i$ für alle i. Wir können einen letzten Schritt weitergehen und landen beim Partitionsproblem. Gibt es $x_1, \ldots, x_n \in \{0, 1\}$ mit

$$g_1 x_1 + \ldots + g_n x_n = (g_1 + \ldots + g_n)/2?$$

Für das Rucksackproblem gibt es mit Hilfe der Dynamischen Programmierung pseudopolynomielle Algorithmen und effiziente Approximationsschemata. Für eine exakte Lösung stehen Branch-and-Bound Methoden zur Verfügung. Diese Methoden lassen sich teilweise auf die Binäre Optimierung, aber nicht mehr auf die Ganzzahlige Optimierung verallgemeinern. Das Problem der Ganzzahligen Optimierung erweist sich als schwieriger. Andererseits ist die Forderung nach Ganzzahligkeit der Lösungen in vielen Fällen aus praktischer Sicht eine notwendige Forderung.

4.2 Vorbereitende Überlegungen

Wir werden später in einigen Schritten analog zur Simplexmethode vorgehen. Daher sollte die Ganzzahligkeit der Parameter erhalten bleiben. Wenn wir uns die Formeln für den Pivotschritt anschauen, stellen wir fest, daß nur durch das Pivotelement dividiert wird. Wenn wir die Ganzzahligkeit der Parameter erhalten wollen, bleibt uns wohl nichts anderes übrig, als durch Tricks zu erreichen, daß das Pivotelement stets $+1$ oder $\Leftrightarrow 1$ ist.

Was kann denn die Simplexmethode bei der Ganzzahligen Optimierung helfen? Wir nehmen zunächst an, daß der zulässige Bereich (auch ohne die Bedingung $x_1, \ldots, x_n \in \mathbb{Z}$) beschränkt ist. Dann arbeitet die Simplexmethode auf den Schnittpunkten von n linear unabhängigen Hyperebenen und später auf den Extrempunkten der zulässigen Menge. Diese Extrempunkte müssen nicht ganzzahlig sein, auch wenn die Eingabeparameter ganzzahlig sind. Wenn wir einen solchen Extrempunkt gefunden haben, könnte es sinnvoll sein, eine neue Nebenbedingung hinzuzufügen, die diesen Extrempunkt, aber keinen zulässigen ganzzahligen Vektor verbietet. Am Ende könnte die zulässige Menge auf die konvexe Hülle aller zulässigen ganzzahligen Vektoren geschrumpft sein. Auf diesen Bereich könnte die Simplexmethode angewendet werden, um das Problem der Ganzzahligen Optimierung zu lösen. In dieser reinen Form werden jedoch viel zu viele neue Nebenbedingungen gebraucht. Es genügt aber, so viele neue Nebenbedingungen hinzuzufügen, daß die Lösung (ohne die Nebenbedingung $x_1, \ldots, x_n \in \mathbb{Z}$) ganzzahlig ist.

Wir testen zunächst, ob der zulässige Bereich (ohne die Nebenbedingung $x_1, \ldots, x_n \in \mathbb{Z}$) leer oder unbeschränkt ist. Wenn er leer ist, gilt dies auch mit der weiteren Nebenbedingung $x_1, \ldots, x_n \in \mathbb{Z}$, und das Problem ist gelöst. Wenn der zulässige Bereich unbeschränkt ist (was in der Praxis kaum vorkommt), müssen Sondermaßnahmen ergriffen werden, die in den Übungen besprochen werden. Mit dem „normalen" Simplex-Algorithmus können wir ansonsten die Funktion $x_1 + \ldots + x_n$ auf dem beschränkten Bereich maximieren und so eine ganzzahlige obere Schranke M für $x_1 + \ldots + x_n$ erhalten.

Wir nehmen an, daß zu Beginn die Spalten 1 bis n linear unabhängig sind. Falls das nicht der Fall ist, kann man neue Variablen einführen, die den Linearkombinationen von alten Variablen entsprechen und man kann ein neues Tableau mit weniger Spalten aufstellen. Dies kann man so lange durchführen, bis die Spalten 1 bis n linear unabhängig sind. Wir werden das in den Übungen genauer besprechen.

Der von uns vorgestellte „Algorithmus von Gomory" wird auf die duale Simplexmethode zurückgreifen, d.h. er wird dafür sorgen, daß wir stets einen dual zulässigen Basispunkt vorliegen haben. Um zu Beginn einen solchen dual zulässigen Basispunkt zu bekommen, verfahren wir wie folgt: Wenn alle a_{0j} (die Koeffizienten der Zielfunktion) nicht negativ sind, erhalten wir direkt einen dual zulässigen Basispunkt. Ansonsten fügen wir die nicht einschränkende Restriktion $x_1 + \ldots + x_n \leq M$ zum Problem hinzu. Die j-te Spalte wird Pivotspalte, wenn $a_{0j} = \min\{a_{0k} \mid 1 \leq k \leq m\}$ ist. In diesem Fall ist $a_{0j} < 0$. Die neue Restriktion wird Pivotzeile. Damit ist das Pivotelement $+1$ (primal) bzw. $\Leftrightarrow 1$ (dual). Aus den Zahlen a_{0k} werden die Werte a'_{0k}. Dabei ist $a'_{0j} = \Leftrightarrow a_{0j} > 0$. Für $k \neq j$ ist $a'_{0k} = a_{0k} \Leftrightarrow a_{0j} \geq 0$ nach Wahl von j. Nach diesem Basiswechsel erhalten wir sofort einen dual zulässigen Basispunkt. Diese Methode zur Berechnung eines dual zulässigen Basispunktes wird M-Methode genannt.

Wir benötigen zur weiteren Beschreibung des Algorithmus den Begriff der lexikographischen Positivität.

Lexikographisch positive Vektoren

Definition 4.2.1: Ein Vektor $a = (a_1, \ldots, a_t)$ heißt *lexikographisch positiv*, wenn $a \neq \underline{0}$ und der erste von Null verschiedene Eintrag größer als Null ist. Formal: Aus $k = \min\{i \mid a_i \neq 0\}$ folgt $a_k > 0$.

Zum Beispiel ist $(0, 0, 3, \Leftrightarrow 1, 4, \Leftrightarrow 4)$ lexikographisch positiv und $(0, 0, \Leftrightarrow 1, 2, 2)$ nicht. Ein Vektor a heißt *lexikographisch größer* als ein Vektor b, wenn die Differenz $a \Leftrightarrow b$ lexikographisch positiv ist. Naheliegenderweise schreiben wir $a >_{lex} b$, wenn a lexikographisch größer ist als b. Da für zwei Vektoren $a \neq b$ entweder $a \Leftrightarrow b$ oder $b \Leftrightarrow a$ lexikographisch positiv ist, haben wir eine totale Ordnung auf der Menge der Vektoren definiert und in einer Menge von Vektoren kann man also auch den „lexikographisch kleinsten" Vektor auszeichnen. Wir bemerken am Rande, daß ein Vektor lexikographisch positiv ist genau dann, wenn er lexikographisch größer ist als der Nullvektor. Wir zeigen zunächst einfache Eigenschaften:

Lemma 4.2.2: *Die Summe zweier lexikographisch positiver Vektoren a und d ist lexikographisch positiv.*

Beweis: Sei $i_0 := \min\{i \mid a_i > 0 \text{ oder } d_i > 0\}$. Dann ist $a_i + d_i = 0$ für alle $i < i_0$ und $a_{i_0} + d_{i_0} > 0$. □

Wenn wir in den reellen Zahlen für $a, b > 0$ eine Ungleichung der Form $a + c \cdot b > 0$ haben, dann können wir leicht ausrechnen, wie klein wir c wählen können, damit diese Ungleichung wahr wird, es reicht nämlich aus, $c > \Leftrightarrow a/b$ zu wählen. Wie groß bzw. wie klein kann man aber eine Zahl c wählen, so daß für zwei lexikographisch positive Vektoren a und b auch der Vektor $a + c \cdot b$ lexikographisch positiv ist?

Lemma 4.2.3: *Gegeben seien zwei lexikographisch positive Vektoren $a \neq b$ sowie eine Zahl c. Sei $i_a := \min\{i \mid a_i > 0\}$ und $i_b := \min\{i \mid b_i > 0\}$. Wenn eine der folgenden Bedingungen erfüllt ist, dann ist $v := a + c \cdot b$ auch lexikographisch positiv.*

i) $c \geq 0$. *ii)* $i_a < i_b$ *und* c *beliebig.*
iii) $i_a = i_b$ *und* $c > \Leftrightarrow a_{i_a}/b_{i_a}$. *iv)* $a >_{lex} b$ *und* $c \geq \Leftrightarrow 1$.

Beweis: i) Wenn $c = 0$ ist, ist Aussage i) trivial. Für $c > 0$ überlegt man sich leicht, daß $c \cdot b$ auch lexikographisch positiv ist und wendet Lemma 4.2.2 an.

ii) Diese Aussage ist trivial, da für $i < i_a$ die Zahl $v_i = 0$ ist und $v_{i_a} = a_{i_a} > 0$ gilt.

iii) Es ist $v_i = 0$ für alle $i < i_a$ und $v_{i_a} = a_{i_a} + c \cdot b_{i_a} > a_{i_a} \Leftrightarrow a_{i_a} = 0$.

iv) Wenn $i_a < i_b$ ist, so folgt Aussage iv) aus Aussage ii). Wenn $i_a = i_b$ ist, dann ist $\Leftrightarrow a_{i_a}/b_{i_a} \leq \Leftrightarrow 1$. Wenn $c > \Leftrightarrow 1$ ist, folgt dann $c > \Leftrightarrow a_{i_a}/b_{i_a}$ und Aussage iv) folgt aus Aussage iii). Wir müssen nun nur noch den Fall $c = \Leftrightarrow 1$ betrachten. Dann ist $v = a \Leftrightarrow b$ und da $a >_{lex} b$ ist, folgt die lexikographische Positivität von v nach Definition der Relation $>_{lex}$. □

Hinzufügen von neuen Restriktionen

Angenommen, das Simplextableau enthält eine Zeile, die der Gleichung $a_{i,1}x_1 + \cdots + a_{i,n}x_n + u_i = a_{i,0}$ entspricht. Wegen $u_i \geq 0$ folgt, daß für jedes $\lambda > 0$ gilt:

$$\frac{a_{i,0}}{\lambda} \Leftrightarrow \frac{a_{i,1}}{\lambda}x_1 \Leftrightarrow \cdots \Leftrightarrow \frac{a_{i,n}}{\lambda}x_n \geq 0, \quad \text{also} \quad \frac{a_{i,0}}{\lambda} \geq \frac{a_{i,1}}{\lambda}x_1 + \cdots + \frac{a_{i,n}}{\lambda}x_n \quad \text{und somit}$$

$$\frac{a_{i,0}}{\lambda} \geq \lfloor\frac{a_{i,1}}{\lambda}\rfloor x_1 + \cdots + \lfloor\frac{a_{i,n}}{\lambda}\rfloor x_n,$$

da alle x–Variablen nicht negativ sind. Da die rechte Seite dieser Ungleichung für alle zulässigen Variablenbelegungen eine ganze Zahl ist, folgt $\lfloor\frac{a_{i,0}}{\lambda}\rfloor \geq \lfloor\frac{a_{i,1}}{\lambda}\rfloor x_1 + \cdots + \lfloor\frac{a_{i,n}}{\lambda}\rfloor x_n$ und somit können wir für eine neue ganzzahlige Variable $\bar{x} \geq 0$ die Zeile

$$\bar{x} + \lfloor\frac{a_{i,1}}{\lambda}\rfloor x_1 + \cdots + \lfloor\frac{a_{i,n}}{\lambda}\rfloor x_n = \lfloor\frac{a_{i,0}}{\lambda}\rfloor$$

zu unserem ganzzahligen linearen Programm hinzufügen, ohne zulässige Lösungen auszuschließen.

Wir kehren zurück zum Algorithmus von Gomory und dem Simplextableau, das wir in diesem Algorithmus verwalten. Wenn alle Spalten lexikographisch positiv sind, folgt daraus sofort, daß der Basispunkt dual zulässig ist (die Umkehrung gilt nicht). Indem wir $x_1 + \cdots + x_n \leq M$ als erste (!) Nebenbedingung einfügen, erhalten wir, da ja der Basispunkt dual zulässig ist, lexikographisch positive Spalten.

4.3 Der Algorithmus von Gomory

Der Algorithmus von Gomory (1963) für Probleme der Ganzzahligen Optimierung ist ein heuristischer Algorithmus, der (natürlich, siehe Kapitel 4.1) zwar exponentielle worst–case–Rechenzeit hat, der aber in vielen Fällen recht schnell ist. Wir nehmen im folgenden an, daß alle in Kapitel 4.2 diskutierten vorbereitenden Maßnahmen durchgeführt worden sind. Wir stellen den Algorithmus von Gomory zunächst

vor, kommentieren dann die Vorgehensweise und beweisen schließlich Korrektheit und Endlichkeit. An dieser Stelle sei bemerkt, daß die Literatur über den Algorithmus von Gomory voller Fehler steckt.

Während des Algorithmus werden zum Tableau Zeilen hinzugefügt. Die Zahlen m und M sollen uns angeben, wieviele Spalten zu Beginn bzw. aktuell vorhanden sind, d. h. zu Beginn seien die Zeilen 0 bis m vorhanden, während des Algorithmus die Zeilen 0 bis M. Zu Beginn setzen wir also $M := m$.

Algorithmus 4.3.1: Algorithmus von Gomory

1.) Falls $a_{i,0} \geq 0$ für $1 \leq i \leq M$, dann STOP: Der Basispunkt ist optimal.

2.) $i := \min\{i' > 0 \mid a_{i',0} < 0\}$. Die Zeile i wird die Pivotzeile „erzeugen".

3.) $J := \{j \mid 1 \leq j \leq n, a_{i,j} < 0\}$. Falls $J = \emptyset$: STOP, es gibt keinen zulässigen ganzzahligen Vektor.

4.) Wähle aus J die lexikographisch kleinste Spalte s. Spalte s wird Pivotspalte.

5.) Sei $i(j) := \min\{k \mid a_{kj} > 0\}$ für $1 \leq j \leq n$. Es sei $\mu_s := \Leftrightarrow 1$ und für alle $j \in J \setminus \{s\}$ mit $a_{i(j),s} > 0$ sei μ_j eine (möglichst kleine) ganze negative Zahl, so daß (Spalte j) + $\mu_j \cdot$ (Spalte s) lex. positiv ist. Nach Lemma 4.2.3, Teil iv) könnten wir immer $\mu_j = \Leftrightarrow 1$ wählen, aber es ist heuristisch gesehen besser, μ_j kleiner zu wählen, wenn möglich.

6.) $$\lambda := \max\left\{ \frac{a_{i,j}}{\mu_j} \mid \mu_j \text{ ist definiert} \right\}.$$

7.) Die Restriktion $\lfloor a_{i,0}/\lambda \rfloor = \sum_{1 \leq j \leq n} \lfloor a_{ij}/\lambda \rfloor x_j + \bar{x}, \quad \bar{x} \geq 0, \bar{x} \in \mathbb{Z}$ wird als letzte Zeile dem Simplextableau hinzugefügt. Diese Zeile wird Pivotzeile. $M := M+1$.

8.) Basiswechsel anhand der gewählten Pivotzeile und Pivotspalte. Weiter in Schritt 1.

Beim ersten Durchlauf sind die Entscheidungen korrekt, wenn der Algorithmus stoppt. Dies folgt sofort in Analogie zum Simplex-Algorithmus aus der Voraussetzung, daß der Basispunkt zum gegebenen Simplextableau dual zulässig und ganzzahlig ist. Wir müssen also sicherstellen, daß auch in späteren Iterationen der Basispunkt dual zulässig und ganzzahlig ist. Es ist leicht zu sehen, daß jede Iteration des Algorithmus von Gomory in linearer Zeit bezogen auf die Größe des Simplextableaus durchführbar ist.

Zunächst kommentieren wir die einzelnen Schritte.

1.) Wenn die Basispunkte dual zulässig und ganzzahlig sind, treffen wir die richtige Entscheidung, da der Basispunkt ganzzahlig und für den gesamten noch betrachteten Bereich optimal ist.
2.) Dieser Schritt ist analog zum dualen Simplex-Algorithmus angelegt. Es wird die erste störende Zeile gewählt. Diese wird jedoch nicht zur Pivotzeile, da dann die Parameter nicht ganzzahlig bleiben. Sie wird nur die Pivotzeile „erzeugen", indem die neue Restriktion in Schritt 7 aus dieser Zeile konstruiert wird.
3.) Die i-te Zeile ist nicht erfüllbar, daher ist die Entscheidung korrekt.
4.) Bei der Wahl der Pivotspalte müssen wir beachten, daß wir das duale Simplexverfahren nachahmen. Im dualen Simplex-Algorithmus wählen wir eine der Spalten $j \in J$, für die a_{0j}/a_{ij} maximal ist, wobei i die gewählte Pivotzeile ist. Hier wird erst die in Schritt 7 hinzugefügte Restriktion Pivotzeile. Der Zahl a_{ij} entspricht dann $\lfloor a_{ij}/\lambda \rfloor$. Die im dualen Simplex-Algorithmus gewählte Spalte würde in unserer neuen Notation zu J gehören. Wir wählen hier die Pivotspalte mit mehr Sorgfalt. Einerseits müssen wir beachten, daß Probleme degeneriert sein können, andererseits müssen wir erzwingen, daß die lexikographische Positivität der Spalten erhalten bleibt. Entsprechende Eigenschaften zeigen wir später. Hier überzeugen wir uns nur davon, daß die lexikographisch kleinste Spalte eindeutig definiert ist: Zu Beginn sind alle Spalten linear unabhängig, somit können keine zwei Spalten identisch sein und im Laufe des Algorithmus bleibt diese lineare Unabhängigkeit der Spalten erhalten, wie wir gleich sehen werden.

5.) Der Wert von $i(j)$ ist wohldefiniert, da die Spalten lexikographisch positiv sind.

6.) Da $\mu_s = \Leftrightarrow 1$ definiert ist, gilt stets $\lambda \geq \Leftrightarrow a_{i,s} \geq 1 > 0$, also gilt für das Pivotelement $\lfloor a_{i,s}/\lambda \rfloor = \Leftrightarrow 1$.

7.) Da die neue Restriktion außer der neuen Basisvariablen $\bar{x}$ nur Nicht-Basisvariablen enthält, kann sie ohne Umformung in das Simplextableau eingefügt werden. Nach den vorhin gemachten Bemerkungen schließen wir keinen zulässigen ganzzahligen Vektor aus.

Lemma 4.3.2: *Die Einträge im Simplextableau sind stets ganzzahlig.*

Beweis: Wie unter 6.) bemerkt, ist das Pivotelement $\Leftrightarrow 1$. Die Koeffizienten der neuen Restriktion sind offensichtlich ganzzahlig. □

Lemma 4.3.3: *Die Spalten $1, \ldots, n$ des Simplextableaus bleiben linear unabhängig und lexikographisch positiv, auch wenn man sie nach der m-ten Zeile abschneidet.*

Beweis: Wir betrachten zunächst die Pivotspalte. Da die Pivotzeile nicht unter den Zeilen 0 bis m ist und sie somit „abgeschnitten" wird, wird die Pivotspalte nach der Regel „$r \rightarrow \Leftrightarrow r/p$" behandelt. Nach Lemma 4.3.2 ist $p = \Leftrightarrow 1$, d. h. die Pivotspalte wird auf den Zeilen 0 bis m nicht verändert.

Die anderen Spalten ändern sich gemäß der Regel „$s \rightarrow s \Leftrightarrow rq/p = s + rq$". Wir betrachten das Verhalten der Spalte $j \neq s$: Für das neue Element a'_{lj} mit $l \leq m$ gilt

$$a'_{lj} = a_{lj} + \left\lfloor \frac{a_{ij}}{\lambda} \right\rfloor \cdot a_{ls}.$$

Anders gelesen: Für $c := \lfloor \frac{a_{ij}}{\lambda} \rfloor$ ergibt sich die neue j–te Spalte aus der alten j–ten Spalte, indem man das c–fache der alten Spalte s aufaddiert. Damit ist auch klar, daß die lineare Unabhängigkeit der Spalten 1 bis n erhalten bleibt, wenn man die Spalten auf den Zeilen 0 bis m betrachtet.

Wir wollen nachweisen, daß die neue j–te Spalte lexikographisch positiv ist. Dazu wollen wir Lemma 4.2.3 anwenden, denn sowohl die alte j–te Spalte als auch die alte s–te Spalte sind lexikographisch positiv.

1. Fall: $c = \lfloor \frac{a_{ij}}{\lambda} \rfloor \geq 0$. Dann greift Aussage i) von Lemma 4.2.3.

2. Fall: $c < 0$, also $a_{ij} < 0$, also $j \in J$. Dann wissen wir, daß die j–te Spalte lexikographisch größer ist als die s–te Spalte. Dann ist $a_{i(j),s} \geq 0$.

Fall 2a: $a_{i(j),s} = 0$. Es greift Lemma 4.2.3 mit Aussage ii).

Fall 2b: $a_{i(j),s} > 0$. Dann ist μ_j definiert, also $\lambda \geq a_{i,j}/\mu_j$ und, da $a_{i,j} < 0$, $\mu_j < 0$:

$$c = \quad \lfloor a_{i,j}/\lambda \rfloor \geq \lfloor a_{i,j}/(a_{i,j}/\mu_j) \rfloor = \lfloor \mu_j \rfloor = \mu_j.$$

Nach Wahl von μ_j ist die neue Spalte j lexikographisch positiv. □

Wegen der lexikographischen Positivität sind insbesondere alle Basispunkte dual zulässig. Bisher haben wir gezeigt, daß der Algorithmus korrekt entscheidet, wenn er in endlicher Zeit stoppt. Nun zeigen wir, *daß* er stoppt.

Lemma 4.3.4: *Der Algorithmus von Gomory ist endlich.*

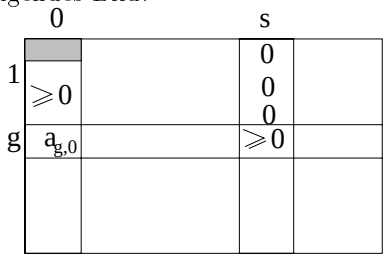
Beweis: Wir nehmen an, daß der Algorithmus unendlich lang läuft und führen diese Annahme zu einem Widerspruch. Wir erhalten eine Folge von Pivotspalten $s_1, s_2, \ldots$. Zunächst wollen wir zeigen, daß es einen Zeitpunkt t^* gibt, ab dem jede jeweilige Pivotspalte s in der 0–ten Zeile nur noch Nullen stehen hat. (Formal würden wir schreiben: $\exists\, t^* \;:\; \forall t \geq t^* : a_{0,s_t} = 0.$)

Zum Beweis beobachten wir, daß sich die Zahl $a_{0,0}$ beim Pivotschritt wie folgt verändert:

$$a'_{0,0} = a_{0,0} + \lfloor a_{i,0}/\lambda \rfloor \cdot a_{0,s}.$$

Wegen der lexikographischen Positivität aller Spalten gilt für alle Pivotspalten $a_{0,s_t} \geq 0$. Wenn a_{0,s_t} unendlich oft größer Null wäre, dann würde wegen $\lfloor a_{i,0}/\lambda \rfloor \leq \Leftrightarrow 1$ die Zahl $a_{0,0}$ unendlich oft kleiner werden, und zwar jeweils mindestens um 1. Aus Dualitätsgründen ist $\Leftrightarrow a_{0,0}$ aber eine untere Schranke für das Minimum der Zielfunktion auf dem zulässigen Bereich. Da wir angenommen haben, daß wir ein beschränktes Problem gegeben haben, gibt es also einen Zeitpunkt t^*, so daß gilt: $\forall t \geq t^* : a_{0,s_t} = 0$.

Nun wollen wir induktiv vorgehen. Angenommen, wir haben schon gezeigt, daß es eine Zahl g und einen Zeitpunkt t^* gibt, ab dem die folgenden Eigenschaften gelten: Jede Pivotspalte s, die noch gewählt wird, hat in den Zeilen 0 bis $g \Leftrightarrow 1$ eine Null und die nullte Spalte hat in den Zeilen 1 bis $g \Leftrightarrow 1$ Zahlen stehen, die größer gleich Null sind. (Wir haben bei der Argumentation über das $a_{0,0}$ gerade den Induktionsanfang für $g=1$ gemacht.) Es ergibt sich folgendes Bild:



Wir wissen $a_{g,s} \geq 0$, da alle Spalten $1, \ldots, n$ lexikographisch positiv sind. Angenommen, es ist $a_{g,0} < 0$. Dann würde der Algorithmus $i=g$ wählen. Da die jeweilige Pivotspalte s aus der Menge J gewählt wird, müßte $a_{g,s} < 0$ sein. Ein Widerspruch. Also kann die g–te Zeile in der nullten Spalte nie negativ werden. Bei einem Pivotschritt verändert sich $a_{g,0}$ wie folgt:

$$a'_{g,0} = a_{g,0} + \lfloor a_{i,0}/\lambda \rfloor \cdot a_{g,s} \leq a_{g,0} \Leftrightarrow a_{g,s}\,.$$

Bei $a_{g,s} > 0$ wird der Wert in Zeile g und Spalte 0 also mindestens um 1 kleiner. Wenn $a_{g,s}$ für alle noch nachfolgenden Pivotspalten s noch unendlich oft größer als Null wäre, dann müßte $a_{g,0}$ aber irgendwann negativ werden, was nicht sein kann, wie wir uns gerade überlegt haben. Also gibt es einen Zeitpunkt t', ab dem gilt: Jede Pivotspalte, die noch gewählt wird, hat in den Zeilen 0 bis g eine Null und die nullte Spalte hat in den Zeilen 1 bis g nur nicht–negative Zahlen stehen.

Induktiv erhält man nun, daß es einen Zeitpunkt gibt, ab dem die gewählte Pivotspalte auf den Zeilen 0 bis m nur Nullen enthält. Dies ist jedoch ein Widerspruch zur Eigenschaft, daß alle Spalten $1, \ldots, n$ lexikographisch positiv sind, auch wenn man sie nach der m–ten Zeile abschneidet. Also wissen wir, daß der Algorithmus nicht unendlich lange läuft. □

Wir halten noch einmal fest, daß die lexikographische Positivität der verkürzten Spalten sichert, daß wir nicht den Wettlauf gegen die Vergrößerung des Tableaus verlieren.

Satz 4.3.5: *Der Algorithmus von Gomory löst das Problem der Ganzzahligen Optimierung.*

Wir haben schon bemerkt, daß der Algorithmus von Gomory ein heuristischer Algorithmus ist. Er hat sich in der Praxis bewährt. Worst case Abschätzungen würden allerdings zeigen, daß der Algorithmus nur in vielen oder den meisten Fällen gut ist. Die Schritte im Algorithmus von Gomory haben wir nachträglich motiviert und bemerken nur noch, daß der Beweis von Lemma 4.3.4 zeigt, daß sich die Koeffizienten im Simplextableau möglichst stark verändern sollten, damit der Algorithmus schnell zum Ziel führt. Da λ im Nenner bei den Änderungen auftritt, sollte λ (unter den Nebenbedingungen) möglichst klein sein. Genauso haben wir λ gewählt.

Weiterhin haben wir gesehen, daß zunächst $a_{0,0}$ seinen festen Wert erhält. Danach kann der Algorithmus noch lange brauchen, bis der Basispunkt auch primal zulässig ist. Wir haben zwar bereits einen sich nicht mehr ändernden Wert der Zielfunktion (ohne dies zu wissen), aber wir müssen noch eventuell viele neue Nebenbedingungen schaffen, um den zulässigen Bereich so zu verändern, daß die ganzzahlige Lösung ein optimaler zulässiger Punkt unter allen (auch nicht ganzzahligen) Vektoren ist.

5 Nichtlineare Optimierung

5.1 Das Problem und eine Klassifizierung der Lösungsansätze

Das allgemeine Problem der Nichtlinearen Optimierung besteht in der Maximierung (oder Minimierung) einer Zielfunktion $f(x_1, \ldots, x_n)$ unter Nebenbedingungen $g_i(x_1, \ldots, x_n) \leq b_i \, (1 \leq i \leq m)$ und $x_j \geq 0 \, (1 \leq j \leq n)$. Im Spezialfall, daß f und alle g_i lineare Funktionen sind, erhalten wir Probleme der Linearen Optimierung. Also wollen wir hier annehmen, daß mindestens eine der beteiligten Funktionen nichtlinear ist. Für das Problem in dieser Allgemeinheit kann es keinen effizienten Lösungsalgorithmus geben. Wir unterscheiden drei Klassen von Lösungsansätzen.

Die analytischen Verfahren versuchen, möglichst große Teilklassen der Nichtlinearen Optimierung mit vertretbarem Aufwand zu lösen. Dazu müssen allerdings starke Voraussetzungen an die Funktionen f und g_i gestellt werden. Die Korrektheit der Verfahren beruht auf Ergebnissen der Analysis, die in dieser Vorlesung nicht vorausgesetzt werden können.

Da hier auch nicht der Platz ist, diese Kenntnisse zu erarbeiten, werden wir in Kap. 5.2 nur einige Methoden vorstellen, ohne ihre Korrektheit zu beweisen.

Im Gegensatz zu analytischen Verfahren, wo eine exakte Lösung angestrebt wird, stehen Verfahren, die sich mit approximativen Lösungen zufrieden geben. Dies ist aufgrund der Schwierigkeit vieler Probleme die einzig verbleibende Möglichkeit. In Kap. 5.3 untersuchen wir die Approximationsmöglichkeiten durch Linearisierung von Zielfunktion und Nebenbedingungen. Die algorithmischen Suchverfahren, die wir in Kap 5.4 andiskutieren, gehen von einer zulässigen Lösung aus und versuchen, die jeweilige Lösung iterativ zu verbessern.

5.2 Analytische Lösungsverfahren

Wir beginnen mit der Optimierung nichtlinearer Zielfunktionen ohne Nebenbedingungen. Aus der Schule wissen wir, daß eine differenzierbare Funktion $f : \mathbb{R} \mapsto \mathbb{R}$ in x_0 nur dann ein lokales Minimum oder Maximum hat, wenn $\partial f(x_0)/\partial x = 0$ ist. Mit dieser Bedingung kann die Zahl der zu untersuchenden Punkte oft auf eine kleine endliche Zahl verringert werden. Wenn es einfache Nebenbedingungen wie $x \in [a, b]$ gibt, müssen bekanntlicherweise die Randpunkte extra betrachtet werden.

Aus der Analysis sollte die Erweiterung dieses Vorgehens auf differenzierbare Funktionen $f : \mathbb{R}^n \mapsto \mathbb{R}$ bekannt sein. Es müssen nicht die Ableitungen in alle möglichen Richtungen betrachtet werden, es ist ausreichend (weil äquivalent) den Vektor der partiellen Ableitungen, den sogenannten Gradienten zu betrachten. Notwendig für ein lokales Optimum in x_0 ist die Bedingung $\operatorname{grad} f(x_0) = 0$. Wenn für die Variablen Randbedingungen wie $x_i \in [a_i, b_i]$ angegeben sind, müssen die Ränder extra behandelt werden. Es sollte auch bekannt sein, wie mit Hilfe der zweiten Ableitung Maxima und Minima unterschieden werden können.

Als erste Verallgemeinerung fügen wir eine Gleichung als Nebenbedingung hinzu

$$\begin{aligned} f(x_1, \ldots, x_n) &\rightarrow \max, \qquad \text{wobei} \\ g(x_1, \ldots, x_n) &= 0. \end{aligned}$$

Hier ist die notwendige Bedingung für ein lokales Optimum in x_0

$$\operatorname{grad} f(x_0) = \lambda \operatorname{grad} g(x_0) \quad \text{für ein } \lambda \in \mathbb{R}.$$

Dabei heißt λ Lagrangescher Multiplikator.

Beispiel 5.2.1: Maximiere $f(x_1, \ldots, x_n) = x_1 \cdots x_n$ unter der Nebenbedingung $x_1 + \ldots + x_n = 1, x_i \geq 0$. An den Rändern wird hier sicherlich das Maximum nicht angenommen. Ebenso ist f als stetige Funktion

auf der zulässigen kompakten Menge beschränkt und nimmt daher ihr Maximum an. Die Methode der Lagrangeschen Multiplikatoren liefert

$$\left(\prod_{i\neq 1} x_i, \prod_{i\neq 2} x_i, \ldots, \prod_{i\neq n} x_i\right) = \lambda(1,1,\ldots,1).$$

Da das Produkt aller x_i, $i \neq 1$, gleich dem Produkt aller x_i, $i \neq 2$ sein muß, folgt $x_1 = x_2$ und analog $x_1 = x_2 = \ldots = x_n$. Aus der Nebenbedingung folgt $x_1 = x_2 = \ldots = x_n = 1/n$.

Beispiel 5.2.2: Maximiere $f(x,y,z) = x^2y^3z$ unter der Nebenbedingung $2x + y^2 + 3z = 1$, $x, y, z \geq 0$. Wir erhalten die Bedingung

$$(2xy^3z, 3x^2y^2z, x^2y^3) = \lambda(2, 2y, 3), \quad \text{also}$$

$$xy^3z = \lambda,\ 3x^2yz = 2\lambda,\ x^2y^3 = 3\lambda.$$

Daraus folgt

$$2xy^3z = 3x^2yz \text{oder} 2y^2 = 3x, \text{ d. h. } x = \frac{2}{3}y^2.$$

Die Bedingungen werden nun zu

$$\frac{2}{3}y^5z = \lambda,\ \frac{4}{3}y^5z = 2\lambda,\ \frac{4}{9}y^7 = 3\lambda.$$

Aus der ersten und letzten Bedingung folgt

$$2y^5z = \frac{4}{9}y^7 \text{oder} z = \frac{2}{9}y^2.$$

Aus der Nebenbedingung wird nun

$$\frac{4}{3}y^2 + y^2 + \frac{2}{3}y^2 = 1, \text{ also } 3y^2 = 1 \text{ und } y = \frac{1}{\sqrt{3}}.$$

Daraus folgt $x = \frac{2}{3}y^2 = \frac{2}{9}$ und $z = \frac{2}{9}y^2 = \frac{2}{27}$.

Wenn die Zahl der Nebenbedingung en wächst, $g_j(x) = 0$, $1 \leq j \leq m$, dann wird die Funktion (Lagrange-Funktion)

$$L(x_1,\ldots,x_n,\lambda_1,\ldots,\lambda_m) := f(x) \Leftrightarrow \sum_{1\leq j\leq m} \lambda_j g_j(x)$$

gebildet. Die notwendige Bedingung für ein lokales Optimum lautet nun

$$\partial L/\partial x_i = 0,\ 1 \leq i \leq n,\ \partial L/\partial \lambda_j = 0, 1 \leq j \leq m.$$

Wir bekommen also ein System von $m+n$ Gleichungen, dessen gemeinsame Lösungen wir suchen müssen. Dies ist in vielen Fällen analytisch nicht möglich.

Die Situation wird noch unangenehmer, wenn die Nebenbedingungen Ungleichungen $g_i(x) \leq b_i$ sind. Es gibt jedoch einen auch praktisch wichtigen Spezialfall, in dem die Analysis Bedingungen für ein globales Maximum liefert. Die zu maximierende Funktion f sei konkav, die Funktionen der Nebenbedingungen dagegen konvex, zusätzlich sei $x_j \geq 0$ gefordert. Dann lauten die Karush-Kuhn-Tucker Bedingungen für die Lagrange-Funktion

$$\begin{array}{lcll} \partial L/\partial x_j & = & 0 & \text{falls } x_{j,0} > 0 \\ \partial L/\partial x_j & \leq & 0 & \text{falls } x_{j,0} = 0 \\ \partial L/\partial \lambda_i & = & 0 & \text{falls } \lambda_{i,0} > 0 \\ \partial L/\partial \lambda_i & \leq & 0 & \text{falls } \lambda_{i,0} = 0 \end{array}$$

für den untersuchten Vektor $(x_{1,0}, \ldots, x_{n,0}, \lambda_{1,0}, \ldots, \lambda_{m,0})$.

Abschließend betrachten wir Methoden, mit denen Nebenbedingungen in die zu optimierende Funktion „eingearbeitet“ werden. Sei das Problem

$$f(x) \to \max$$

$$g_j(x) \leq 0, \quad x_i \geq 0 \quad (1 \leq j \leq m,\ 1 \leq i \leq n)$$

gegeben. Für einen Strafparameter $r > 0$ betrachten wir die Funktion

$$f^*(x, r) := f(x) \Leftrightarrow r \sum_{1 \leq j \leq m} (\max\{0, g_j(x)\})^2$$

Im zulässigen Bereich stimmen f und f^* überein (für jedes r). Für Punkte aus dem unzulässigen Bereich sollte der Strafterm (für große r) verhindern, daß diese Punkte maximal werden. Wenn wir also f^* nur unter den Nebenbedingungen $x_i \geq 0$ maximieren und dabei für irgendein r einen optimalen Punkt erhalten, der die Nebenbedingungen erfüllt, haben wir das ursprüngliche Problem gelöst. Es ist hierbei g_j in f^* quadriert worden, damit die Funktion f^* auch für die Punkte mit $g_j(x) = 0$ für ein j differenzierbar ist, wenn dies für f und alle g_j gilt. Dennoch entsteht das Problem, daß die Ableitungen meistens nur mit Fallunterscheidung $(g_j(x) \geq 0, g_j(x) \leq 0)$ geschrieben werden können.

Dies wird bei der Barriere-Methode vermieden. Dabei wird

$$\begin{aligned} B(x, r) &:= f(x) + \frac{1}{r} b(x) \\ b(x) &= \sum_{1 \leq j \leq m} (\Leftrightarrow g_j(x))^{-1} \text{oder} \\ b(x) &= \sum_{1 \leq j \leq m} \ln(\Leftrightarrow g_j(x)) \end{aligned}$$

maximiert. Die Wirkung der Barrieresummanden ist die folgende. Im Innern des zulässigen Bereiches haben die Funktionen eine geringe Wirkung. Am Rande bekommen sie jedoch den Wert $\Leftrightarrow\infty$, so daß ein Optimum im Innern des zulässigen Bereiches gesucht wird. Wenn nun für $r \to \infty$ (dann verlieren die Barrieresummanden ihre Bedeutung) die Folge optimaler Punkte zulässig bleibt, haben wir das Problem gelöst.

Beispiel 5.2.3:

$$\begin{aligned} f(x, y) &= x^2 + y \ \to \ \max \\ g(x, y) &= x^2 + y^2 \Leftrightarrow 20 \leq 0. \end{aligned}$$

Sei also

$$\begin{aligned} B(x, y, r) &= x^2 + y + \frac{1}{r} \ln(20 \Leftrightarrow x^2 \Leftrightarrow y^2), \\ \partial B/\partial x &= 2x + \frac{1}{r} \frac{\Leftrightarrow 2x}{20 \Leftrightarrow x^2 \Leftrightarrow y^2} = 0, \quad \text{also} \\ r &= \frac{1}{20 \Leftrightarrow x^2 \Leftrightarrow y^2}, \\ \partial B/\partial y &= 1 + \frac{1}{r} \frac{\Leftrightarrow 2y}{20 \Leftrightarrow x^2 \Leftrightarrow y^2} = 0. \end{aligned}$$

Nach Einsetzen von r folgt $1 \Leftrightarrow 2y = 0$, also $y = 1/2$. Nun ist es offensichtlich vernünftig, x so groß wie möglich zu wählen, ohne die Restriktion zu verletzen, d. h. $x = (19,75)^{1/2}$. In diesem Fall ist $r = \infty$ und $f(x, y) = 19,75 + 0,5 = 20,25$.

5.3 Linearisierungsmethoden

Auch hier wollen wir nur zwei Lösungsansätze diskutieren. Im ersten Fall sind die Nebenbedingungen linear, und die Zielfunktion ist separabel, d. h. es gilt

$$f(x_1, \ldots, x_n) = f_1(x_1) + \ldots + f_n(x_n).$$

Dann lassen sich die Funktionen f_i einzeln linear approximieren.

Beispiel 5.3.1:

$$f(x_1, x_2) = \Leftrightarrow(x_1 \Leftrightarrow 3)^2 \Leftrightarrow 2(x_2 \Leftrightarrow 2)^2 \quad \rightarrow \max$$

$$x_1 + x_2 \leq 3,\ x_1 \leq 2,\ x_1 \geq 0,\ x_2 \geq 0$$

Für $f_1(x_1) = \Leftrightarrow(x_1 \Leftrightarrow 3)^2$ und $f_2(x_2) = \Leftrightarrow 2(x_2 \Leftrightarrow 2)^2$ wählen wir jeweils vier Stützstellen 0,1,2 und 3. Wir suchen für f_1 eine stückweise lineare, stetige Funktion g_1, die mit f_1 an diesen vier Stellen übereinstimmt. Es soll also gelten

$$g_1(0) = \Leftrightarrow 9,\ g_1(1) = \Leftrightarrow 4,\ g_1(2) = \Leftrightarrow 1,\ g_1(3) = 0,$$

und g_1 soll auf $[0,1]$, $[1,2]$ und $[2,3]$ linear sein. Wir zerlegen x_1 in $x_1 = x_{11} + x_{12} + x_{13}$ mit $x_{11}, x_{12}, x_{13} \in [0,1]$ und setzen $g_1(x_1) = 5x_{11} + 3x_{12} + 1x_{13} \Leftrightarrow 9$. Wir erhalten die gewünschte Funktion bei folgender Zerlegung

$$\begin{array}{rcl} x_1 \in [0,1] & : & x_{11} = x_1,\ x_{12} = 0,\ x_{13} = 0 \\ x_1 \in [1,2] & : & x_{11} = 1,\ x_{12} = x_1 \Leftrightarrow 1,\ x_{13} = 0 \\ x_1 \in [2,3] & : & x_{11} = 1,\ x_{12} = 1,\ x_{13} = x_1 \Leftrightarrow 2. \end{array}$$

Allerdings erlauben wir auch andere Zerlegungen. Auf analoge Weise erhalten wir

$$g_2(x_2) = 6x_{21} + 2x_{22} \Leftrightarrow 2x_{23} \Leftrightarrow 8.$$

Wir haben also unser Maximierungsproblem linear approximiert durch

$$5x_{11} + 3x_{12} + x_{13} + 6x_{21} + 2x_{22} \Leftrightarrow 2x_{23} \Leftrightarrow 17 \rightarrow \max$$

$$x_{11} + x_{12} + x_{13} + x_{21} + x_{22} + x_{23} \leq 3$$

$$x_{11} + x_{12} + x_{13} \leq 2$$

$$0 \leq x_{ij} \leq 1.$$

Dieses Problem kann mit der Simplex-Methode gelöst werden. Es ergibt sich $x_{11} = 1$, $x_{12} = 1$, $x_{13} = 0$, also $x_1 = 2$, $x_{21} = 1$, $x_{22} = 0$, $x_{23} = 0$, also $x_2 = 1$ und als Wert der Zielfunktion $\Leftrightarrow 3$. Dies ist nicht die Lösung des Ursprungsproblems (wie lautet diese?) Die Qualität der Approximation kann natürlich durch die Wahl engerer Stützstellen verbessert werden. Hier war es auch nicht gerade intelligent, x_1 auf $[0,3]$ zu approximieren, da $x_1 \leq 2$ vorausgesetzt war. Es ist natürlich zu beachten, daß eine Erhöhung der Zahl der Stützstellen das Simplextableau vergrößert und damit Rechenzeit kostet.

Im zweiten Ansatz wird angenommen, daß die nichtlinearen Teile Produkte von zwei verschiedenen Variablen sind. Wir nehmen für diese Variablen x und y an, daß der betrachtete Bereich ein Rechteck $[a,b] \times [c,d]$ ist. Zu Anfang wird dieses Rechteck „groß genug“ gewählt. Wir ersetzen nun x und y durch vier Variablen $\lambda_1, \lambda_2, \lambda_3$ und λ_4. Wir fordern $\lambda_1 + \lambda_2 + \lambda_3 + \lambda_4 = 1$ und $\lambda_i \geq 0$. Die Variablen werden ersetzt durch

$$\begin{array}{rcl} x & \rightarrow & a\lambda_1 + a\lambda_2 + b\lambda_3 + b\lambda_4, \\ y & \rightarrow & c\lambda_1 + d\lambda_2 + c\lambda_3 + d\lambda_4, \\ xy & \rightarrow & ac\lambda_1 + ad\lambda_2 + bc\lambda_3 + bd\lambda_4. \end{array}$$

In den vier Eckpunkten können wir die drei „Variablen“ x, y und xy durch Wahl von $\lambda_i = 1$ für das passende i exakt darstellen. Allgemein wollen wir „Werte“

$$\begin{gathered}\alpha a + (1 \Leftrightarrow \alpha) b \\ \beta c + (1 \Leftrightarrow \beta) d \\ \alpha\beta ac + \alpha(1 \Leftrightarrow \beta) ad + \beta(1 \Leftrightarrow \alpha) bc + (1 \Leftrightarrow \alpha)(1 \Leftrightarrow \beta) bd\end{gathered}$$

darstellen. Das führt zu folgenden Bedingungen.

$$\begin{aligned}\lambda_1 + \lambda_2 + \lambda_3 + \lambda_4 &= 1, \quad \lambda_i \geq 0, \\ \alpha &= \lambda_1 + \lambda_2 \\ \beta &= \lambda_1 + \lambda_3 \\ \alpha\beta &= \lambda_1 \\ \alpha(1 \Leftrightarrow \beta) &= \lambda_2 \\ \beta(1 \Leftrightarrow \alpha) &= \lambda_3 \\ (1 \Leftrightarrow \alpha)(1 \Leftrightarrow \beta) &= \lambda_4 .\end{aligned}$$

Wir bekommen also eine lineare Approximation an die Wirklichkeit. Wenn die Lösung (x^*, y^*) im Innern des Rechtecks liegt, wird das Rechteck um einen Faktor $\gamma \in (0, 1)$, z. B. $\gamma = 1/5$ verkleinert,

$$\begin{aligned}a' &= \gamma a + (1 \Leftrightarrow \gamma) x^* \\ b' &= \gamma b + (1 \Leftrightarrow \gamma) x^* \\ c' &= \gamma c + (1 \Leftrightarrow \gamma) y^* \\ d' &= \gamma d + (1 \Leftrightarrow \gamma) y^* .\end{aligned}$$

Wenn die Lösung auf dem Rand des Rechtecks liegt, wird das Rechteck um die entsprechende Seite geklappt, damit die Lösung „in die gewünschte Richtung“ wandern kann. Wir sehen, daß die Wahl $\gamma = 1/5$ die Rechtecke schnell verkleinert, aber auch dafür sorgt, daß wir maximal viermal in eine Richtung wandern können, ohne das größere Rechteck zu verlassen. Schließlich müssen noch geeignete Abbruchkriterien festgelegt werden.

Diese beiden Lösungsansätze sollten vor allem veranschaulichen, daß es bei den Linearisierungsmethoden erstaunlich viele Freiheiten gibt.

5.4 Algorithmische Suchverfahren

Das Grundprinzip Algorithmischer Suchverfahren ist einfach. Ausgehend von einem (heuristisch guten) zulässigen Punkt wird eine Suchrichtung gewählt. Diese Suchrichtung reduziert n-dimensionale Suchräume auf Geraden. Durch entsprechende Ersetzung der Variablen wird das Optimierungsproblem auf diese Gerade reduziert. Wir erhalten also ein eindimensionales Optimierungsproblem, das meistens direkt lösbar ist. Für das gegebene Problem heißt dies, daß wir den besten Punkt auf einer Geraden gefunden haben. Dieser Punkt ist nun der nächste zulässige Punkt, von dem aus eventuell auf analoge Weise weiter gesucht wird.

Das populärste Verfahren dieser Art ist das Gradientenverfahren.

Der Gradient einer differenzierbaren Funktion gibt die Richtung des steilsten Anstiegs an. Daher gehört das Gradientenverfahren zu den Greedy Methoden. Der Gradient gibt dabei aber nur lokal die Richtung des steilsten Anstiegs an. Für konkave Funktionen ohne Nebenbedingungen konvergiert die Folge der von uns erzeugten Punkte gegen das globale Maximum, in allgemeinen Fällen ist nur sichergestellt, daß die Punktfolge gegen ein lokales Maximum konvergiert. Die Methode ist also nicht empfehlenswert, wenn wir es mit Funktionen mit vielen lokalen Maxima, die wesentlich schlechter als globale Maxima sind, zu

tun haben und wir nicht garantieren können, eine Anfangslösung zu berechnen, die nahe genug an einem globalen Optimum liegt.

Bei Problemen mit Restriktionen müssen wir eventuell an Grenzen stoppen, die durch die Restriktionen vorgegeben sind. Insbesondere gibt es eine Reihe von Regeln zur Wahl der nächsten Suchrichtung, wenn der Gradient in den unzulässigen Bereich zeigt. Eine dieser Regeln lautet beispielsweise: Führt die Richtung des Gradienten in einem Randpunkt des zulässigen Bereichs aus diesem heraus, ohne senkrecht auf dem Rand zu stehen, so wird die Richtung des auf die Grenze des zulässigen Bereichs projizierten Gradienten als weitere Suchrichtung gewählt. Steht der Gradient senkrecht auf dem Rand des zulässigen Bereichs, ist ein lokales Optimum gefunden.

Die iterative Variante des Barriere-Verfahrens aus Kap. 5.2 ist das Gummiwand-Verfahren. Die Funktion

$$B(x,r) := f(x) \Leftrightarrow \frac{1}{r} \sum_{1 \le j \le m} \frac{1}{[\Leftrightarrow g_j(x)]} \quad \text{mit} \quad [\Leftrightarrow g_j(x)] := \begin{cases} \Leftrightarrow g_j(x) & \text{falls } \Leftrightarrow g_j(x) > 0 \\ 0 & \text{sonst.} \end{cases}$$

wird hierbei betrachtet. Jetzt wird jedoch nicht bezüglich einer Richtung optimiert, sondern es werden Schritte einer Schrittlänge Δ in die verschiedenen Koordinatenrichtungen (nach einer bestimmten Reihenfolge) probiert, solange dies den Wert der abgeänderten Zielfunktion verbessert. Im Laufe der Zeit werden r und Δ verringert. Es ist teilweise erheblich besser, auch andere (gemischte) Suchrichtungen zu erlauben.

Diese klassischen Methoden haben den Nachteil, nur Schritte zu gestatten, die den Wert der Zielfunktion verbessern. Wer in einem Gebirge den höchsten Berg besteigen möchte, ist jedoch gezwungen, viele Täler zu durchwandern. Die Methode des Simulated Annealing ermöglicht eine entsprechende Vorgehensweise. Wir wollen hier auf das physikalische Experiment, dem die Methode des Simulated Annealing nachempfunden wurde, nicht näher eingehen. Das Verfahren läßt sich folgendermaßen beschreiben. Wenn wir einen zulässigen Punkt gefunden haben, sind Schritte in die Nachbarschaft möglich. Passende Nachbarschaften müssen definiert werden. Unter diesen Schritten wird einer zufällig ermittelt, wobei häufig eine Gleichverteilung auf der Menge möglicher Schritte benutzt wird, aber eine geschicktere Wahl nicht ausgeschlossen ist. Ein verbessernder Schritt wird stets durchgeführt, ein verschlechternder Schritt wird nur mit einer bestimmten Wahrscheinlichkeit akzeptiert, die von der Größe der Verschlechterung und von der Zahl durchgeführter Iterationen (oder Phasen) abhängt. Die Idee liegt nun darin, zu Beginn auch den Abstieg in viele und tiefe Täler zu akzeptieren, in der Hoffnung, langsam in die Nähe des höchsten Berges zu kommen, den wir dann nur noch mit kleiner Wahrscheinlichkeit wieder verlassen. Es ist Intuition und Erfahrung nötig, um die vielen Freiheiten im Grundalgorithmus des Simulated Annealing für spezielle Probleme gut zu nutzen. Die Erfolge beim Einsatz von Simulated Annealing Algorithmen sind jedoch überzeugend. Genetische und Evolutionäre Strategie werden in vielen Spezialvorlesungen behandelt.

Dieser kurze Ausblick in die Nichtlineare Optimierung soll genügen. Für einige Verfahren würden wir einen tieferen mathematischen Hintergrund brauchen, um ihre Korrektheit unter bestimmten Nebenbedingungen beweisen zu können. Bei den heuristischen Verfahren ist dagegen viel Zeit und praktische Übung nötig, um von dem Grundverfahren aus zu erfolgreichen heuristischen Regeln zu gelangen.

6 Nutzen- und Entscheidungstheorie

6.1 Nutzentheorie

Bei den bisher diskutierten Optimierungsproblemen haben wir angenommen, daß wir wissen, was wir optimieren wollen. Beispiele haben sich mit der Maximierung der Einnahmen bzw. der Minimierung der Kosten befaßt. Geld ist nämlich einfach zu bewerten: Viel Geld gilt als besser als weniger Geld. In vielen Situationen, in denen Entscheidungen gefällt werden müssen, sind jedoch recht unvergleichbare „Güter" abzuwägen. Wie bewerten wir ein finanziell gutes Geschäft, das uns mit einer gewissen Wahrscheinlichkeit einen Freund kostet?

Operations Research Verfahren setzen voraus, daß die Nutzenwerte so gewählt sind, daß eine Maximierung des Nutzens das Gewünschte leistet. Wenn Nutzen nicht zahlenmäßig meßbar ist, können Operations Research Verfahren nicht angewendet werden. Mit der Nutzentheorie wird versucht, für möglichst viele Situationen angemessene Nutzenwerte zu definieren.

Dabei beginnen wir mit einer ordinalen Nutzenbewertung. Menschen sollten für zwei Ereignisse A und B wissen, ob sie A bevorzugen (präferieren), d. h. $A\,p\,B$, B bevorzugen ($B\,p\,A$) oder ob sie der Entscheidung A oder B indifferent gegenüberstehen ($A\,i\,B$). Für die Relationen p und i fordern wir, daß die folgenden Axiome erfüllt sind. Sonst ist eine Nutzenbewertung nicht möglich.

(1) Für je zwei Ereignisse gilt genau eine der drei Beziehungen $A\,p\,B$, $A\,i\,B$ und $B\,p\,A$.

(2) $A\,i\,A$ für alle A.

(3) $A\,i\,B \quad \Leftrightarrow \quad B\,i\,A$.

(4) $A\,i\,B$ und $B\,i\,C \quad \Rightarrow \quad A\,i\,C$.

(5) $A\,p\,B$ und $B\,p\,C \quad \Rightarrow \quad A\,p\,C$.

(6) $A\,p\,B$ und $B\,i\,C \quad \Rightarrow \quad A\,p\,C$.

(7) $A\,i\,B$ und $B\,p\,C \quad \Rightarrow \quad A\,p\,C$.

Mit einer derartigen ordinalen Nutzenskala kann entschieden werden, welche der Möglichkeiten $A_1, \ldots, A_m$ gewählt werden sollte. Spannend wird es bei Entscheidungen unter Risiko. Es gelte $A\,p\,B\,p\,C$. Entscheidung 1 sei die Wahl von B, während Entscheidung 2 mit Wahrscheinlichkeit r das Ereignis A und mit Wahrscheinlichkeit $1 \Leftrightarrow r$ das Ereignis C auslöst. Die zweite Entscheidung nennen wir eine Lotterie $rA + (1 \Leftrightarrow r)C$. Welche Entscheidung soll nun gefällt werden? Wir sehen, daß wir gezwungen sind, Lotterien als Ereignisse zuzulassen, und in unsere Nutzenskala einzufügen. Für Lotterien gelten die folgenden Axiome.

(8) $rA + (1 \Leftrightarrow r)B = (1 \Leftrightarrow r)B + rA$.

(9) $rA + (1 \Leftrightarrow r)[sB + (1 \Leftrightarrow s)C] = rA + (1 \Leftrightarrow r)sB + (1 \Leftrightarrow r)(1 \Leftrightarrow s)C$.

(10) $rA + (1 \Leftrightarrow r)A = A$.

(11) $A\,i\,C$ und $r \in [0,1] \Rightarrow [rA + (1 \Leftrightarrow r)B]\,i\,[rC + (1 \Leftrightarrow r)B]$.

(12) $A\,p\,C$ und $r > 0 \Rightarrow [rA + (1 \Leftrightarrow r)B]\,p\,[rC + (1 \Leftrightarrow r)B]$.

(13) $A\,p\,C\,p\,B$. Dann existiert ein $r \in (0,1)$ mit $[rA + (1 \Leftrightarrow r)B]\,i\,C$.

Wenn wir alle diese Axiome akzeptieren, müssen wir die Folgerungen daraus auch akzeptieren. Diese führen schließlich zu Nutzenfunktionen.

Satz 6.1.1: *Der Wert von r im Stetigkeitsaxiom (13) ist eindeutig bestimmt.*

Beweis: Sei $0 < s < r < 1$ und $[rA + (1 \Leftrightarrow r)B]\, i\, C$ und $[sA + (1 \Leftrightarrow s)B]\, i\, C$. Nach Axiom (10) gilt

$$\frac{r \Leftrightarrow s}{1 \Leftrightarrow s} B + \frac{1 \Leftrightarrow r}{1 \Leftrightarrow s} B = B.$$

Aus $A\, p\, B$ folgt mit Axiom (12)

$$\left[\frac{r \Leftrightarrow s}{1 \Leftrightarrow s} A + \frac{1 \Leftrightarrow r}{1 \Leftrightarrow s} B\right]\, p\, B.$$

Aus Axiom (9) und (10) folgt

$$rA + (1 \Leftrightarrow r)B = sA + (1 \Leftrightarrow s)\left[\frac{r \Leftrightarrow s}{1 \Leftrightarrow s} A + \frac{1 \Leftrightarrow r}{1 \Leftrightarrow s} B\right].$$

Zusammengenommen folgt mit $\left[\frac{r-s}{1-s} A + \frac{1-r}{1-s} B\right]\, p\, B$

$$C\, i\, [rA + (1 \Leftrightarrow r)B]\, p\, [sA + (1 \Leftrightarrow s)B]\, i\, C$$

im Widerspruch zu den grundlegenden Axiomen. □

Satz 6.1.2: *Es existiert eine Funktion u, die die Menge aller Ereignisse in die Menge der reellen Zahlen abbildet, für die $u(A) > u(B)$ genau dann gilt, wenn $A\, p\, B$ ist, und für die die Gleichung*

$$u(rA + (1 \Leftrightarrow r)B) = ru(A) + (1 \Leftrightarrow r)u(B)$$

erfüllt ist. Darüber hinaus ist die Abbildung u bis auf lineare Transformationen eindeutig, d. h. gelten die Bedingungen auch für u', gibt es $\alpha > 0$ und $\beta \in \mathbb{R}$ mit

$$u'(A) = \alpha u(A) + \beta.$$

Dieser Satz besagt, daß aus der Annahme, daß Nutzenfunktionen linear auf Lotterien wirken sollten, die Eindeutigkeit der Nutzenfunktion bis auf „die Wahl der Währung“ folgt. Mit β kann der Nullpunkt der Nutzenfunktion beliebig festgelegt werden. Der Faktor α wählt die „Währung“. Offensichtlich sind u und u' für unsere Zwecke gleich gut geeignet.

Beweis von Satz 6.1.2:: Wenn $A\, i\, B$ für alle Ereignisse gilt, ist die Nutzenfunktion $u(A) = 0$ geeignet und offensichtlich auch bis auf lineare Transformationen eindeutig.

Ansonsten existieren Ereignisse, E_1 und E_0 mit $E_1\, p\, E_0$. Wir definieren $u(E_1) = 1$ und $u(E_0) = 0$. Für jedes andere Ereignis A gibt es fünf Möglichkeiten, für die wir $u(A)$ definieren.

- $A\, p\, E_1\, p\, E_0$. Nach dem Stetigkeitsaxiom existiert ein $r \in (0, 1)$ mit $[rA + (1 \Leftrightarrow r)E_0]\, i\, E_1$. Aus

  $$u(E_1) = u(rA + (1 \Leftrightarrow r)E_0) = ru(A) + (1 \Leftrightarrow r)u(E_0)$$

  folgt $u(A) = 1/r$.
- $A\, i\, E_1$. Dann muß $u(A) = 1$ sein.
- $E_1\, p\, A\, p\, E_0$. Es existiert ein $s \in (0, 1)$ mit $[sE_1 + (1 \Leftrightarrow s)E_0]\, i\, A$. Also setzen wir $u(A) = s$.
- $A\, i\, E_0$. Dann muß $u(A) = 0$ sein.
- $E_1\, p\, E_0\, p\, A$. Es existiert ein $t \in (0, 1)$ mit $[tE_1 + (1 \Leftrightarrow t)A]\, i\, E_0$. Also sollte $tu(E_1) + (1 \Leftrightarrow t)u(A) = u(E_0)$ sein. Daraus folgt

  $$u(A) = \Leftrightarrow t/(1 \Leftrightarrow t).$$

Für die so definierte Funktion u müssen nun allgemein die beiden geforderten Bedingungen nachgewiesen werden.

$$\begin{aligned} u(A) > u(B) &\Leftrightarrow A\, p\, B \\ u(rA + (1 \Leftrightarrow r)B) &= ru(A) + (1 \Leftrightarrow r)u(B). \end{aligned}$$

Der vollständige Beweis ist sehr umfangreich, da A und B je 5 Fälle erfüllen können. Wir wählen nur den Fall

$$E_1\, p\, A\, p\, E_0 \quad \text{und} \quad E_1\, p\, B\, p\, E_0.$$

Die anderen (einige der anderen) Fälle werden als Übungsaufgabe gestellt. Es seien s_A, $s_B \in (0,1)$ mit

$$[s_A E_1 + (1 \Leftrightarrow s_A)E_0]\, i\, A \quad \text{und} \quad [s_B E_1 + (1 \Leftrightarrow s_B)E_0]\, i\, B.$$

Falls $s_A = s_B$, folgt aus Axiom (4), daß $A\, i\, B$ ist. Nach Definition ist

$$u(A) = s_A = s_B = u(B).$$

Falls $s_A > s_B$, folgt wie im Beweis von Satz 6.1.1

$$A\, i\, [s_A E_1 + (1 \Leftrightarrow s_A)E_0]\, p\, [s_B E_1 + (1 \Leftrightarrow s_B)E_0]\, i\, B,$$

also $A\, p\, B$. Analog folgt für $s_A < s_B$, daß $B\, p\, A$ gilt.

Zum Nachweis der Linearität von u setzen wir in $rA + (1 \Leftrightarrow r)B$ die zu A und B indifferenten Ereignisse ein. Es folgt

$$\begin{aligned} &[rA + (1 \Leftrightarrow r)B]\, i\, [r[s_A E_1 + (1 \Leftrightarrow s_A)E_0] + (1 \Leftrightarrow r)[s_B E_1 + (1 \Leftrightarrow s_B)E_0]] \\ &= (rs_A + (1 \Leftrightarrow r)s_B)E_1 + (r(1 \Leftrightarrow s_A) + (1 \Leftrightarrow r)(1 \Leftrightarrow s_B))\, E_0. \end{aligned}$$

Damit gehört $rA + (1 \Leftrightarrow r)B$ auch zum dritten Fall. Nach Definition von u gilt also

$$u(rA + (1 \Leftrightarrow r)B) = rs_A + (1 \Leftrightarrow r)s_B = ru(A) + (1 \Leftrightarrow r)u(B).$$

Abschließend zeigen wir, daß jede andere Nutzenfunktion u', die den Anforderungen entspricht, eine lineare Tranformation von u ist. Da $E_1\, p\, E_0$, folgt $u'(E_1) > u'(E_0)$. Wir setzen

$$\begin{aligned} \beta &= u'(E_0) \text{ und} \\ \alpha &= u'(E_1) \Leftrightarrow u'(E_0) \end{aligned}$$

Dann ist $\alpha > 0$. Wieder betrachten wir nur einen der fünf Fälle. Sei $E_1\, p\, A\, p\, E_0$ und $u(A) = s$, d.h. $A\, i\, [sE_1 + (1 \Leftrightarrow s)E_0]$. Für die Nutzenfunktion u' gilt dann

$$\begin{aligned} u'(A) &= u'(sE_1 + (1 \Leftrightarrow s)E_0) = su'(E_1) + (1 \Leftrightarrow s)u'(E_0) \\ &= s(u'(E_1) \Leftrightarrow u'(E_0)) + u'(E_0) = s\alpha + \beta \\ &= \alpha u(A) + \beta. \end{aligned}$$

□

Der Beweis von Satz 6.1.2 gibt uns auch eine Möglichkeit, wie wir Nutzenfunktionen ermitteln können. Wir entscheiden, ob es überhaupt Ereignisse gibt, gegen die wir nicht indifferent sind. Sei $E_1\, p\, E_0$. Für jedes andere Ereignis A wird ermittelt, zu welchem der fünf Fälle es gehört und was die zugehörige Lotteriewahrscheinlichkeit ist. Daraus kann dann $u(A)$ ermittelt werden. Allerdings soll das Problem, passende Lotteriewahrscheinlichkeiten zu schätzen, nicht verniedlicht werden.

Als Experiment möge jeder Leser und jede Leserin versuchen, den persönlichen Nutzen von Geld zu messen. Der Nutzen von Geld wächst zunächst linear. So sind uns 10 DM im wesentlichen so viel wert wie eine Lotterie

$$\frac{1}{2}(20\,\text{DM}) + \frac{1}{2}(0\,\text{DM}).$$

Hierbei müssen wir verdeutlichen, daß diese Schreibweise eine Lotterie und nicht den Mittelwert 10 DM beschreibt. Allerdings sind den meisten Menschen 1 Million DM lieber als eine Lotterie zu $r = 1/2$ für die Auszahlungen 2 Millionen DM und 0 DM. Der Grenznutzen der meisten Güter fällt mit der Menge des Gutes. Dies erklärt auch, warum Menschen Versicherungen abschließen. Vernünftige Versicherungsverträge führen nämlich stets zu einer negativen erwarteten Auszahlung (die Versicherung will auch leben), aber dennoch zu einem positiven Nutzen, da hohe Kosten (schwere Krankheit, hoher Haftpflichtschaden) in jedem Fall vermieden werden müssen, damit uns finanziell eine Zukunft bleibt.

Nach Wahl einer angemessenen Nutzenfunktion soll der Nutzen einer Lotterie $rA + (1 \Leftrightarrow r)B$ genau $ru(A)+(1\Leftrightarrow r)u(B)$ betragen. Die Ergebnisse aus Kapitel 4 müssen also neu interpretiert werden. Sie dürfen sich nicht stupide auf Geldwerte beziehen, sondern sie müssen auf der Grundlage einer angemessenen Nutzenfunktion gebildet worden sein.

6.2 Grundlagen der Entscheidungstheorie

Entscheidungen unter Gewißheit sind im Prinzip leicht zu treffen. Es muß eine Aktion mit maximalem Nutzen gewählt werden. Im später folgenden Kapitel zur Dynamischen und Stochastischen Optimierung werden wir sehen und auch in früheren Kapiteln dieser Vorlesung haben wir bereits gesehen, daß es schwierig sein kann, eine solche Aktion oder Strategie zu berechnen. Hier wollen wir eine viel einfachere Grundsituation annehmen, nämlich die folgende:

Es gibt nur endlich viele Aktionen und die Parameter zu den Aktionen sind bekannt. Wir müssen jedoch eine Entscheidung unter Ungewißheit treffen. Nicht alle Parameter sind bekannt (Nachfrage, Wetter am nächsten Tag, ...). Uns ist eine Auszahlungsmatrix bekannt, die für jede Aktion und jeden möglichen Parameterwert die Auszahlung angibt. Welche Aktion soll gewählt werden? Es gibt viele Regeln zur Berechnung einer „guten“ Aktion. Wir stellen die wichtigsten vor. Es sei e_{ij} die Auszahlung bei Parameterkonstellation i und Aktion j.

Maximax-Regel: Wähle eine Aktion j, für die

$$\max_i e_{ij}$$

maximal ist. Diese Aktion sichert im Glücksfall die größte Auszahlung. Sie ist nur für extrem risikofreudige Personen angemessen, da sie keinerlei Absicherung gegen widrige Umstände beinhaltet.

Maximin-Regel: Wähle eine Aktion j, für die

$$\min_i e_{ij}$$

maximal ist. Diese Regel ist extrem konfliktscheu, da die Natur als übelwollender Gegner behandelt wird. Die Gefahr, bei einer Aktion einen Minimalnutzen zu verlieren, läßt Aktionen als unattraktiv aussehen, selbst wenn im Glücksfall das Paradies winkt.

Hurwicz-Regel: Wähle ein $\alpha \in [0, 1]$. Wähle eine Aktion, für die

$$\alpha \max_i e_{ij} + (1 \Leftrightarrow \alpha) \min_i e_{ij}$$

maximal ist. Mit einem Parameter α wird also zwischen der Maximax- und der Maximin-Regel gemittelt. Auch hier werden für jede Aktion nur die negativste und die positivste Konstellation betrachtet.

Laplace-Regel: Die Laplace-Verteilung ist die Gleichverteilung. Wähle eine Aktion j, für die

$$\frac{1}{n} \sum_i e_{ij}$$

maximal ist. Hierbei wird über alle Möglichkeiten gemittelt. Wenn keine Kenntnis über die wahren Wahrscheinlichkeiten vorhanden ist, ist die Annahme der Gleichverteilung die einzige vernünftige Annahme.

Minimax-Regret-Regel: Der Regret-Wert einer Aktion bei einer Parameterkonstellation soll messen, wie sehr man den entgangenen Chancen nachtrauert. Es sei

$$r_{ij} = \max_k e_{ik} \Leftrightarrow e_{ij}.$$

Es soll nun eine Aktion j gewählt werden, für die $\frac{1}{n}\sum_i r_{ij}$ minimal ist. Unabhängig von der Parameterkonstellation sind wir durchschnittlich am nächsten am Maximum.

Die beiden folgenden Regeln setzen voraus, daß die Wahrscheinlichkeit p_i für die i-te Parameterkonstellation bekannt ist.

Maximum-Likelihood-Regel: Es wird ein i^* mit $p_{i^*} = \max_i\{p_i\}$ gewählt. Dann wird eine Aktion j gewählt, für die

$$e_{i^*j}$$

maximal ist. Bei vielen möglichen Aktionen und nur kleinem p_{i^*} werden viel zu viele nicht unwahrscheinliche Konstellationen ganz ignoriert.

Bayes–Regel: Es wird eine Aktion j gewählt, für die die erwartete Auszahlung

$$\sum_i p_i e_{ij}$$

maximal ist. Diese Regel werden wir später in der Dynamischen und Stochastischen Optimierung von vornherein zur Richtschnur unseres Handelns machen.

Können wir abschätzen, wann es sich lohnt, sichere Informationen zu beschaffen? Bei sicherem Wissen können wir das nachträgliche Bedauern natürlich auf 0 reduzieren. Der Wert

$$\sum_i p_i r_{ij}$$

gibt den Mittelwert für das nachträgliche Bedauern an. Wenn uns die Beschaffung sicherer Informationen mehr kostet als wir durch dieses Wissen gewinnen, ist es sicherlich vernünftig, auf die Beschaffung sicherer Informationen zu verzichten.

Zunächst wollen wir diskutieren, wie wir unsere a–priori Wahrscheinlichkeiten $p(A_1), \ldots, p(A_n)$ nach einem Experiment, das die Ergebnisse $B_1, \ldots, B_m$ liefern kann, „korrigieren“ müssen. Dazu müssen wir die bedingten Wahrscheinlichkeiten $p(B_j|A_i)$ kennen. So kennen wir eventuell die a-priori Wahrscheinlichkeiten $p(A_1), \ldots, p(A_n)$, daß wir unsere Brille in einem der n Zimmer unserer Wohnung verlegt haben. Wir kennen auch die Wahrscheinlichkeit, die Brille bei einer zehnminütigen Suche im j-ten Zimmer zu finden, wenn die Brille im j-ten Zimmer ist. Nach einer ausgiebigen erfolglosen Suche im j-ten Zimmer sollte die Wahrscheinlichkeit, daß die Brille im j-ten Zimmer ist, gesunken sein, während sie für alle anderen Zimmer gestiegen sein sollte. Wie können wir diese a–posteriori Wahrscheinlichkeiten berechnen?

Satz 6.2.1: Bayessches Gesetz $\quad p(A_i|B_j) = \dfrac{p(B_j|A_i)p(A_i)}{\sum_k p(B_j|A_k)p(A_k)}.$

Beweis: Es gilt $p(A_i|B_j) = \dfrac{p(A_i \cap B_j)}{p(B_j)} = \dfrac{p(A_i \cap B_j)}{\sum_k p(A_k \cap B_j)} = \dfrac{p(B_j|A_i)p(A_i)}{\sum_k p(B_j|A_k)p(A_k)}.$ □

Beispiel 6.2.2: Eine Ölgesellschaft hat die Chance, an einer Stelle nach Öl zu bohren. Die a-priori Wahrscheinlichkeit für das Finden von Öl sei $p(+) := 0.5$. Also ist $p(\Leftrightarrow) = 0.5$. Die Bohrkosten mögen 100.000 DM betragen. Der Gewinn bei Vorhandensein von Öl sei 250.000 DM. Wird nicht nach Öl gebohrt, beträgt die Auszahlung 0 DM. Wird nach Öl gebohrt, beträgt die Auszahlung

$$\Leftrightarrow 100.000 + \frac{1}{2}250.000 + \frac{1}{2}0 = 25.000.$$

Also sollte gebohrt werden. Nun kommt aber ein schlaues Geologieteam und bietet uns eine seismographische Untersuchung für 30.000 DM an. Es ist bekannt, daß die Güte ihrer Arbeit sich folgendermaßen messen läßt. Sie sagen Öl mit Wahrscheinlichkeit 0.9 voraus, wenn auch welches vorhanden ist. Sie sagen allerdings auch mit Wahrscheinlichkeit 0.2 Öl voraus, wenn keines vorhanden ist, d. h.

$$p(+|+) = 0.9 \qquad p(\Leftrightarrow|+) = 0.1$$
$$p(+|\Leftrightarrow) = 0.2 \qquad p(\Leftrightarrow|\Leftrightarrow) = 0.8$$

Sollen wir die Offerte akzeptieren? Die Wahrscheinlichkeit, Öl vorherzusagen, beträgt

$$0.9 \cdot \frac{1}{2} + 0.2 \cdot \frac{1}{2} = 0.55$$

Die a-posteriori Wahrscheinlichkeit für Öl, wenn welches vorhergesagt wurde, beträgt 9/11. Wenn wir also bei positiver Vorhersage nach Öl bohren, beträgt die erwartete Auszahlung

$$\Leftrightarrow 100.000 + \frac{9}{11} 250.000 + \frac{2}{11} \cdot 0 \approx 104.546.$$

Es sollte also nach Öl gebohrt werden. Die a-posteriori Wahrscheinlichkeit für Öl, wenn keines vorhergesagt wurde, beträgt 1/9. Wenn dann nach Öl gebohrt wird, ist die erwartete Auszahlung negativ. Es sollte also nicht gebohrt werden. Wenn wir also die seismographische Untersuchung durchführen lassen, zahlen wir 30.000 DM und erhalten insgesamt die erwartete Auszahlung von $\Leftrightarrow 30.000 + 0.55(\Leftrightarrow 100.000 + \frac{9}{11} 250.000) = 27.500$ DM, die Voruntersuchung lohnt sich also.

In dem behandelten Beispiel hatten wir nur die Möglichkeit, die seismographische Untersuchung durchführen zu lassen oder sie abzulehnen. In anderen Beispielen können wir zwischen zahlreichen Optionen wählen. Es sei für ein Problem der Qualitätskontrolle beispielsweise bekannt, daß 40% aller Sendungen 10% defekte Teile, 30% aller Sendungen 20% defekte Teile und 30% aller Sendungen 30% defekte Teile enthalten. Einkaufs- und Verkaufspreise sind gegeben. Der Händler muß den Kunden defekte Teile ersetzen, die er aber nicht vom Werk ersetzt bekommt, wenn er die Ware akzeptiert hat. Nun kann der Händler Stichproben aus den Sendungen entnehmen und diese Teile testen. Dies kostet natürlich Geld. Nach der Stichprobe kann eine Sendung akzeptiert oder abgelehnt werden. Bei einer abgelehnten Sendung werden alle Teile getestet und defekte Teile vom Hersteller ersetzt. Hier kommt es auch darauf an, eine optimale Stichprobengröße zu berechnen. Die Berechnung der entsprechenden Wahrscheinlichkeiten ist mit den bekannten Methoden möglich. Auch hier wollen wir uns nicht in Details vertiefen, da wir dann Statistik betreiben müßten.

7 Spieltheorie

7.1 Was ist Spieltheorie?

In den bisher behandelten Optimierungsproblemen gab es nur eine Partei, uns selber, die Interessen hatte. Wir wollten Gewinne, Auszahlungen oder Nutzen maximieren oder Kosten minimieren. Die Außenwelt setzte uns teilweise unliebsame Restriktionen, also Nebenbedingungen. Manchmal wurden die zukünftigen Situationen auch durch zufällige Ereignisse beeinflußt. Dies alles ist für viele reale Probleme angemessen. Aber es gibt auch mindestens ebensoviele Probleme, in denen andere Parteien ihre Interessen vertreten. Am klarsten tritt dies in Spielen und Gesellschaftsspielen auf, was der hier betrachteten Theorie ihren unwissenschaftlich klingenden Namen verliehen hat. Gleiches gilt aber im Wirtschaftsleben, wo Firmen um einen Markt konkurrieren, oder in der nationalen und internationalen Politik.

Die klassische Konfliktsituation (Schachspiel) ist die des Zwei-Personen Nullsummenspiels unter vollständiger Information. Diese Spiele lassen sich durch endliche Spielbäume beschreiben, an deren Blättern die Spielausgänge mit ihrem Wert stehen. Der Gewinn des einen Spielers ist unmittelbar der Verlust des anderen. Die Werte in den einzelnen Situationen lassen sich im Spielbaum bottom-up auf triviale Weise berechnen. Das einzige Problem ist eine effiziente Bearbeitung, bei der möglichst große Teilbäume nicht betrachtet werden müssen (α-β-Pruning, siehe Skript DATENSTRUKTUREN). Das Schachspiel ist nur deswegen interessant, weil der Spielbaum so groß ist, daß er bisher nicht analysiert werden konnte. Prinzipiell wissen wir jedoch, wie wir eine optimale Strategie berechnen können.

Prinzipielle Schwierigkeiten treten bereits bei Zwei-Personen-Nullsummenspielen unter unvollständiger Information auf. Beim Stein-Schere-Papier Spiel „ziehen“ die Beteiligten gleichzeitig, beim Poker ist die Kartenverteilung zum Teil unbekannt. Bei Zwei-Personen-Nullsummenspielen ist der Interessengegensatz perfekt. Es gibt also nichts zu verhandeln. Optimale Strategien können wir erstaunlich leicht berechnen, da wir ja die Theorie der Linearen Optimierung und die Dualitätstheorie kennen.

Moderne Spiele ermöglichen Kooperation. Dies macht Spiele spannender, ist menschlicher und der Realität (hoffentlich) angepaßter. Wir behandeln diese Verhandlungssituation zunächst für zwei Beteiligte. Dabei gibt es eine nicht-kooperative Variante, bei der die Beteiligten nicht über die Situation diskutieren dürfen, und eine kooperative Variante. Es ist hierbei nicht das Ziel, das Ergebnis von Verhandlungen vorherzusagen. Dies kann nämlich vom psychologischen Geschick oder auch vom Machtwillen der Beteiligten abhängen. Wir wollen versuchen, Situationen zu bewerten und stabile oder auch gerechte Lösungen zu charakterisieren. Dies kann in der Praxis dazu verhelfen, Situationen realistisch einzuschätzen, d. h. die eigene Verhandlungsstärke weder zu überschätzen noch zu unterschätzen. Dies ist in der Realität übrigens eine der wichtigsten Voraussetzungen für erfolgreiche Verhandlungen. Sich überschätzende Beteiligte sind nicht kompromißfähig, sich unterschätzende Beteiligte erhalten zu wenig, so daß Lösungen in der Zukunft instabil sind. Wir werden stets annehmen, daß jeder Beteiligte nur seinen Nutzen maximieren will. Dies sieht nur auf den ersten Blick egoistisch aus. Wer den Interessen anderer einen positiven Wert beimißt, muß dies in seiner Nutzenfunktion berücksichtigen. Danach ist es ausreichend, den „eigenen“ Nutzen maximieren zu wollen.

Noch spannender wird die Situation bei $n > 2$ Beteiligten. Erstmals sind Koalitionen möglich. Was muß eine Koalition mir bieten, damit ich mich an ihr beteilige? Welche Lösungen können als stabil angesehen werden? Schließlich wollen wir die Macht von Aktienbesitzern oder Parteien in Parlamenten messen. Warum kann eine Erhöhung des Aktienanteils von 20 auf 30% nutzlos sein, während eine Erhöhung auf 31% die Macht vergrößert? Es ist leicht zu erklären (Zünglein an der Waage), warum die FDP in der Bundesrepublik in den Drei-Fraktionen-Bundestagen von 1961–1982 so mächtig war. Wie aber können wir die Macht der Parteien in Italien messen?

Wir wollen versuchen, Antworten (zumindest Teilantworten) auf diese spannenden Fragen zu finden.

7.2 Zwei-Personen-Nullsummenspiele

Wir behandeln hier also zwei sehr wesentliche Einschränkungen. Am Spiel (wir werden diesen Ausdruck benutzen, obwohl wir viel allgemeinere Situationen oder Konflikte meinen) nehmen nur zwei Spieler (Beteiligte, Kontrahenten, Partner) teil, und für jeden Spielausgang gilt, daß Spieler II zahlen muß, was Spieler I erhält. Negative Zahlungen bedeuten natürlich, daß Spieler I an Spieler II zahlt. Die Spieler haben also konträre Interessen. Die Spielregeln und der Ablauf des Spiels interessieren uns weniger. Beide Spieler können sich eine Strategie auswählen, die für jede mögliche Spielsituation und jeden möglichen Wissensstand beschreibt, was der Spieler tun wird. Nach der Wahl der Strategien könnte ein Dritter das Spiel für beide Spieler spielen. In deterministischen Spielen kommt es zu einer festen Auszahlung, bei stochastischen Spielen (zufällige Kartenverteilung) zu einer zufälligen Auszahlung, wobei wir die erwartete Auszahlung als feste Auszahlung interpretieren.

Definition 7.2.1: Ein Zwei-Personen-Nullsummenspiel kann durch eine $n \times m$-Matrix beschrieben werden. Spieler I hat dann n und Spieler II m Strategien zur Auswahl. Bei Wahl der Strategie i durch Spieler I und der Strategie j durch Spieler II muß Spieler II a_{ij} an Spieler I zahlen.

Nach unserer obigen Beschreibung haben wir mit unserem sehr allgemeinen Strategiebegriff wirklich alle Zwei-Personen-Nullsummenspiele erfaßt. Es ist eine gute Übung, sich das Aussehen der Matrix für Spiele wie Schach und Poker zu überlegen. Daß für Gesellschaftsspiele n und m riesig groß sind, muß wohl nicht betont werden. In anderen Konfliktsituationen sind n und m zwar typischerweise groß, aber doch überschaubar (und oft bekannt).

Wann kann die Wahl einer Strategiekombination (i, j) als stabil gelten? Doch nur dann, wenn nicht ein einzelner Spieler durch Wahl einer anderen Strategie sich verbessern würde.

Definition 7.2.2: Eine Strategiekombination heißt Gleichgewichtspaar, wenn kein Spieler durch Änderung seiner Strategie einen Vorteil erhalten würde. Für Zwei-Personen-Nullsummenspiele heißt für die Auszahlungsmatrix $A = (a_{ij})$ die Strategiekombination (i, j) Sattelpunkt, falls a_{ij} das Minimum seiner Zeile und das Maximum seiner Spalte ist.

Offensichtlich sind genau die Sattelpunkte Gleichgewichtspaare.

Beispiel 7.2.3:

$$A = \begin{pmatrix} 5 & 1 & 3 \\ 3 & 2 & 4 \\ \Leftrightarrow 3 & 0 & 1 \end{pmatrix}.$$

Offensichtlich ist nur $a_{22} = 2$ Sattelpunkt. Eine Analyse des Spiels zeigt, daß beide Spieler Strategie 2 benutzen sollten. Hierbei geht natürlich ein, daß niemand den anderen für blöd hält. Spieler I sollte nicht Strategie 1 in der Hoffnung spielen, daß Spieler II seine einzige Gewinnchance wahrnehmen will und auch Strategie 1 spielt. Spielt Spieler II nämlich Strategie 2, bekommt Spieler I nur 1 statt 2 ausgezahlt. Wenn Spieler I erkennt, daß Spieler II Strategie 2 spielen wird, ist es für ihn am besten, Strategie 2 zu spielen. Sollte Spieler II nicht so klug sein, kann Spieler I nur zusätzlich verdienen. Gleiche Überlegungen gelten für Spieler II. Aber was geschieht, wenn es mehrere Gleichgewichtspaare gibt? Bei Zwei-Peronen-Nullsummenspielen sind alle Gleichgewichtspaare äquivalent, wie wir in folgendem Satz zeigen:

Satz 7.2.4: *Sind (i, j) und (i', j') Sattelpunkte in A, dann sind auch (i, j') und (i', j) Sattelpunkte in A, und es gilt $a_{ij} = a_{i'j} = a_{ij'} = a_{i'j'}$.*

Beweis: Aus den Sattelpunkteigenschaften folgt

$$\begin{array}{ccccc} a_{ij} & \leq & a_{ij'} & \leq & a_{i'j'} \text{ und} \\ a_{i'j'} & \leq & a_{i'j} & \leq & a_{ij}. \end{array}$$

Also ist $a_{ij} = a_{i'j} = a_{ij'} = a_{i'j'}$. Da $a_{i'j'}$ Minimum seiner Zeile ist, gilt dies auch für $a_{i'j}$. Da a_{ij} Maximum seiner Spalte ist, gilt dies auch für $a_{i'j}$. Analog kann gezeigt werden, daß (i, j') Sattelpunkt ist. □

Für Zwei-Personen-Nullsummenspiele mit Sattelpunkten in der Auszahlungsmatrix kennen wir somit Lösungen. Betrachten wir aber nun das Spiel Stein-Schere-Papier, wobei wir die Strategien entsprechend durchnumerieren. Es ergibt sich die Auszahlungsmatrix

$$A = \begin{pmatrix} 0 & 1 & \Leftrightarrow 1 \\ \Leftrightarrow 1 & 0 & 1 \\ 1 & \Leftrightarrow 1 & 0 \end{pmatrix}.$$

Diese Matrix hat offensichtlich keinen Sattelpunkt. Gleiches gilt für die Matrix

$$B = \begin{pmatrix} 2 & 0 \\ \Leftrightarrow 1 & 1 \end{pmatrix}.$$

Was kann sich Spieler I sichern? Spielt er allgemein Strategie i, bekommt er mindestens $\min\{a_{ij} | \, 1 \leq j \leq m\}$. Diese Mindestauszahlung möchte er maximieren. Er kann also eine Strategie i wählen, für die

$$v_I^* := \max_i \min_j \{a_{ij}\}$$

sein Maximum annimmt. Spieler II zahlt bei Wahl von Strategie j höchstens $\max_i\{a_{ij}\}$ und möchte dies minimieren. Er kann sich also dagegen wehren, mehr als

$$v_{II}^* := \min_j \max_i \{a_{ij}\}$$

zu zahlen, indem er eine Strategie j wählt, für die v_{II}^* sein Minimum annimmt. Schon aus dieser Interpretation folgt

$$v_I^* \leq v_{II}^*,$$

denn Spieler I kann sich keinen Gewinn sichern, gegen dessen Zahlung Spieler II sich erfolgreich wehren kann. Falls A einen Sattelpunkt (i, j) hat, folgt offensichtlich

$$v_I^* = v_{II}^* = a_{ij}.$$

Im Stein-Schere-Papier Spiel gilt $v_I^* = \Leftrightarrow 1$ und $v_{II}^* = 1$, für die B-Matrix gilt $v_I^* = 0$ und $v_{II}^* = 1$. Für jede Strategienkombination gibt es einen Spieler, der gerne abweichen würde.

Bei Sattelpunkten schadet es nichts, wenn wir die gewählte Strategie bekanntgeben. Beim Stein-Schere-Papier-Spiel besteht der Trick aber gerade darin, das eigene Verhalten nicht preiszugeben und mit psychologischer Raffinesse das Verhalten des Gegenübers zu erraten. Diese Psychologie hat mit den handelnden Personen zu tun und kann nicht allgemein für das Spiel modelliert werden. Wenn wir unsere Absichten verschleiern wollen, müssen wir bereit sein, verschiedene Strategien zu spielen und den Zufall entscheiden zu lassen, was wir tatsächlich tun. Zumindest ist diese Vorstellung zur Analyse des Spiels notwendig.

Definition 7.2.5: Eine gemischte Strategie eines Spielers ist eine Wahrscheinlichkeitsverteilung über seine (sogenannten reinen) Strategien. Wenn die Spieler die gemischten Strategien $(x_1, \ldots, x_n)$ und $(y_1, \ldots, y_m)$ spielen, beträgt die erwartete Auszahlung

$$\sum_{1 \leq i \leq n} \sum_{1 \leq j \leq m} a_{ij} x_i y_j,$$

oder in Matrizenschreibweise $x^T A y$.

Was kann sich Spieler I nun sichern? Spielt er Strategie x und Spieler II „kriegt das raus“, wird Spieler II y so wählen, daß $x^T A y$ minimal ist. Strategie x muß also so gewählt werden, daß

$$v_I(x) := \min\{x^T A y | \, y \text{ Strategie für II}\}$$

maximal ist. Für festes x ist $x^T Ay$ eine lineare Funktion in $(y_1, \ldots, y_m)$. Die Nebenbedingungen $y_j \geq 0$ und $y_1 + \ldots + y_m = 1$ führen zu einem Simplex mit genau m Extrempunkten, nämlich den m Einheitsvektoren $y^{(i)} = (0, \ldots, 0, 1, 0, \ldots, 0)$ mit einer 1 an der i-ten Stelle. Diese Vektoren entsprechen genau den m reinen Strategien. Sei nun $A_{\bullet j}$ die j-te Spalte von A. Dann gilt nach den Ergebnissen der Linearen Optimierung

$$v_I(x) = \min\{x^T A_{\bullet j} \mid 1 \leq j \leq m\}.$$

Mit gemischten Strategien kann sich Spieler I also den Wert v_I, der durch

$$v_I = \max\{\min\{x^T A_{\bullet j} \mid 1 \leq j \leq m\} \mid x \text{ Strategie für } I\}$$

definiert ist, als erwartete Auszahlung sichern. Analog kann sich Spieler II gegen eine höhere erwartete Auszahlung als

$$v_{II} = \min\{\max\{A_{i\bullet} y \mid 1 \leq i \leq n\} \mid y \text{ Strategie für } II\}$$

absichern, wobei $A_{i\bullet}$ die i-te Zeile von A bezeichnet.

Wir können v_I durch ein Problem der Linearen Optimierung beschreiben.

$$\lambda \to \max! \text{ wobei}$$

$$x^T A_{\bullet j} \geq \lambda, \quad 1 \leq j \leq m,$$

$$x_1 + \ldots + x_n = 1$$

$$x_1, \ldots, x_n \geq 0.$$

Wenn $x^T A_{\bullet j} \geq \lambda$ für alle j gilt, dann auch für das Minimum (und umgekehrt). Der Maximalwert von λ entspricht also dem Wert von v_I. Wenn wir das Simplextableau aufschreiben, bemerken wir, daß in der Zielfunktion nur λ den Wert 1 hat, alle anderen Koeffizienten sind 0. In der Gleichung $x_1 + \ldots + x_n$ machen wir x_n zur Schlupfvariablen. In der Spalte für die rechten Seiten der Nebenbedingungen steht in der Darstellungsform von Kap. 2.8 eine $\Leftrightarrow$1 und sonst Nullen. Wenn wir nun das duale Problem bilden, erhalten wir

$$\lambda' \to \min!$$

$$A_{i\bullet} y \leq \lambda', \quad 1 \leq i \leq n$$

$$y_1 + \ldots + y_m = 1$$

$$y_1, \ldots, y_m \geq 0$$

und somit ein Lineares Programm zur Berechnung von v_{II}.

Satz 7.2.6: Minimaxtheorem
Für Zwei-Personen-Nullsummenspiele gilt $v_I = v_{II}$. Diese Werte und Strategien für Spieler I und Spieler II, die Spieler I einen erwarteten Gewinn von v_I sichern bzw. Spieler II gegen eine höhere Zahlung als v_{II} absichern, können mit Hilfe eines Linearen Programms gelöst werden.

Beweis: Das Minimaxtheorem folgt direkt aus dem Dualitätstheorem 2.8.5. □

Zwei-Personen-Nullsummenspiele sind für uns also gelöst. Als Spieler I können wir uns eine erwartete Auszahlung von v_I sichern. Spieler II kann sich gegen eine höhere erwartete Zahlung absichern. Die beiden zugehörigen Strategien sind im Gleichgewicht. Dies gilt für kein Strategienpaar, in dem die erwartete Zahlung nicht $v := v_I = v_{II}$ ist.

7.3 Nicht kooperative Spiele

Wir haben eben gesehen, daß Zwei-Personen-Nullsummenspiele nicht kooperativ sind. Es lohnt sich nicht, mit dem anderen zu verhandeln, da es kein gemeinsames Interesse gibt. Anders ist die Situation bei $n > 2$ Spielern oder bei Spielen, die nicht die Nullsummeneigenschaft haben.

Wir nehmen für Spiele mit n Spielern an, daß es, falls Spieler i genau k_i reine Strategien hat, n-dimensionale Auszahlungsmatrizen $A_1, \ldots, A_n$ gibt, wobei A_i an der Stelle $(l_1, \ldots, l_n)$ mit $1 \leq l_j \leq k_j$ die Auszahlung für Spieler i enthält, wenn Spieler j $(1 \leq j \leq n)$ Strategie l_j benutzt. Spiele heißen nicht kooperativ, wenn die Spieler nicht miteinander reden können, Seitenzahlungen ausgeschlossen sind, usw.

Definition 7.3.1: Ein Strategientupel $(s_1, \ldots, s_n)$ von gemischten Strategien ist für die Auszahlungsmatrizen $A_1, \ldots, A_n$ im Gleichgewicht, wenn für jeden Spieler gilt, daß eine alleinige Strategieänderung seinerseits ihm keinen Vorteil bringt.

Schon hier sollte intuitiv klar sein (Beispiele folgen), daß verschiedene Gleichgewichtsstrategientupel nicht zu gleichen Auszahlungstupeln führen müssen. Wir können aber zeigen, daß es stets Strategientupel gibt, die sich im Gleichgewicht befinden.

Satz 7.3.2: *Jedes nicht kooperative Spiel besitzt mindestens ein Strategientupel, das sich im Gleichgewicht befindet.*

Beweis: Wir beschränken uns auf den Fall $n = 2$, da der allgemeine Fall keine weitere Idee erfordert, aber der Beweis durch den notwendigen Formalismus schwerer lesbar wird. Sei $A = A_1$ und $B = A_2$. Sei (x, y) ein beliebiges Paar (gemischter) Strategien. Sei

$$c_i := \max\{0, A_{i\bullet} y \Leftrightarrow x^T A y\}.$$

Falls $c_i > 0$, sind (x, y) nach Definition nicht im Gleichgewicht. Falls $c_i = 0$ für alle i, ist keine Abweichung zu einer reinen Strategie für Spieler I von Vorteil. Da jede gemischte Strategie eine konvexe Kombination der reinen Strategien ist, kann keine Abweichung für Spieler I von Vorteil sein. Formal gilt

$$x^{*T} A y \Leftrightarrow x^T A y > 0$$

nur, wenn für mindestens ein i

$$A_{i\bullet} y \Leftrightarrow x^T A y > 0$$

ist. Analog sei

$$d_j = \max\{0, x^T B_{\bullet j} \Leftrightarrow x^T B y\}.$$

Für n Spieler müßten wir für jeden Spieler die Abweichung bei einem Wechsel zu einer reinen Strategie messen. Wir definieren nun eine Transformation $T(x, y) = (x', y')$ durch

$$\begin{aligned} x'_i &:= \frac{x_i + c_i}{1 + \sum_k c_k} \\ y'_j &:= \frac{y_j + d_j}{1 + \sum_l d_l}. \end{aligned}$$

Offensichtlich ist (x', y') wieder ein Paar gemischter Strategien. Es gilt

$$T(x, y) = (x, y) \Leftrightarrow \forall i : c_i = 0 \text{ und } \forall j : d_j = 0.$$

Die Richtung „$\Leftarrow$“ ist trivial. Die Richtung „$\Rightarrow$“ folgt, da nicht alle c_i (oder alle d_j) positiv sein können. Eine gemischte Strategie, also eine konvexe Kombination der reinen Strategien, kann nicht schlechter als

jede reine Strategie sein, da die Auszahlungsfunktion linear ist. Mit unserer Diskussion der Bedeutung der c-Werte (d-Werte analog) folgt

$$T(x,y) = (x,y) \Leftrightarrow (x,y) \text{ sind im Gleichgewicht.}$$

Es gibt also genau dann ein Strategienpaar, das sich im Gleichgewicht befindet, wenn die Abbildung T einen Fixpunkt hat. Die Abbildung operiert auf der Menge der Strategienpaare und damit auf einer kompakten und konvexen Teilmenge von $\mathbb{R}^{n+m}$, nämlich

$$\{(x,y) | x_1 + \ldots + x_n = 1, x_1, \ldots, x_n \geq 0,\ y_1 + \ldots + y_m = 1, y_1, \ldots, y_m \geq 0\}.$$

Alle c_i und damit alle x_i' (analog d_j und y_j') hängen stetig von (x,y) ab, also ist T stetig. Nun hat die Mathematik die Arbeit für uns getan. Der seit Jahrzehnten bekannte Fixpunktsatz von Brouwer (auf dessen Beweis wir hier verzichten müssen) besagt, daß jede stetige Funktion $f : S \to S$, wobei S eine konvexe und kompakte Teilmenge des $\mathbb{R}^n$ ist, mindestens einen Fixpunkt hat. □

Es gibt Methoden zur Berechnung bzw. Approximation von Fixpunkten, die wir hier nicht behandeln, da der Gleichgewichtsbegriff (außer für Zwei-Personen-Nullsummenspiele) umstritten ist. Dazu diskutieren wir zwei Beispiele.

Beispiel 7.3.3: Kampf der Geschlechter

$$(A,B) = \begin{pmatrix} (4,1) & (0,0) \\ (0,0) & (1,4) \end{pmatrix}$$

Zumindest unter der angegebenen Bezeichnung ein perfides Spiel. Streiten sich die Geschlechter, geht es allen schlecht (Auszahlung 0). Dominiert ein Geschlecht, geht es diesem gut und dem anderen etwas besser als im Streitfall. Niemand ist gezwungen, diesen letzten Teil der Modellierung als gelungen zu bezeichnen. Das Spiel ist hoffentlich schon deswegen irreal, da eine Kooperation zwischen den Geschlechtern verboten ist. Die reinen Strategien $x = (1,0)$ und $y = (1,0)$ sind im Gleichgewicht mit Auszahlung $(4,1)$, dies gilt aber auch für die Strategien $x' = (0,1)$ und $y' = (0,1)$ mit Auszahlung $(1,4)$. Die Stabilität dieser beiden Gleichgewichtspaare darf wohl angezweifelt werden. Erstaunlicherweise gibt es noch ein Gleichgewichtspaar. Sei $x'' = (4/5, 1/5)$ und $y'' = (1/5, 4/5)$. Mit Wahrscheinlichkeit $4/25$ kommt es zur Auszahlung $(4,1)$, mit Wahrscheinlichkeit $16/25 + 1/25$ kommt es zur Auszahlung $(0,0)$ und mit Wahrscheinlichkeit $4/25$ zur Auszahlung $(1,4)$. Die erwartete Auszahlung für jedes Geschlecht beträgt also sogar nur 0.8. Würde das x-Geschlecht zu $x = (1,0)$ abweichen, erhält die Auszahlung $(4,1)$ die Wahrscheinlichkeit $1/5$, die Auszahlung $(1,4)$ aber die Wahrscheinlichkeit 0. Also bleibt die erwartete Auszahlung 0.8. Würde das x-Geschlecht zu $x' = (0,1)$ abweichen, erhält die Auszahlung $(4,1)$ die Wahrscheinlichkeit 0, die Auszahlung $(1,4)$ aber die Wahrscheinlichkeit $4/5$. Wieder bleibt die erwartete Auszahlung 0.8. Dieser Gleichgewichtspunkt ist also für beide Geschlechter noch schlechter und gewiß nicht stabil.

Hier können wir uns bereits Gedanken machen, was eine Kooperation bewirken kann. Offensichtlich sollten beide Geschlechter eine Auszahlung von 2.5 erhalten. Dazu könnten sie sich auf eine Lotterie, sogar auf einen Münzwurf einigen. Bei Kopf werden die ersten Strategien gespielt, bei Zahl die zweiten. Das liefert eine erwartete Auszahlung von 2.5. Aber wie die Welt so ist, wird sich Geschlecht I denken: Wenn Kopf fällt, prima, wenn aber Zahl fällt, was kümmert es mich. Diese Kooperation wird also erst möglich, wenn sie vertraglich abgesichert werden kann. Selbst dann ist die Lösung nicht voll befriedigend, da jedes Geschlecht nur im Durchschnitt auf die erwartete Auszahlung von 2.5 kommt. Besser ist es, wenn vertraglich Seitenzahlungen vereinbart werden können, die beiden Geschlechtern in jedem Fall die Auszahlung 2.5 zukommen lassen. Geld und Waren lassen auf einfache Weise Seitenzahlungen zu, aber im Kampf der Geschlechter? Die Lösung kann nur ein „Vertrag der Geschlechter“, der echte Gleichberechtigung sichert, sein. Vielleicht ist dann die Auszahlung für beide Geschlechter größer als 2.5, aber dies führt über die Analyse des gegebenen Spiels hinaus.

Beispiel 7.3.4: Gefangenendilemma

$$(A,B) = \begin{pmatrix} (\Leftrightarrow 2, \Leftrightarrow 2) & (\Leftrightarrow 10, 0) \\ (0, \Leftrightarrow 10) & (\Leftrightarrow 5, \Leftrightarrow 5) \end{pmatrix}.$$

Wir befinden uns im Wilden Westen. Der Sheriff hat zwei Banditen verhaftet. Er vermutet, daß sie die Bank ausgeraubt haben, kann dies aber nicht beweisen. Darüber hinaus ist seine Rechtsauffassung klassisch geprägt: „In dubio pro reo". Er sperrt die Gefangenen in Einzelzellen und gibt ihnen bis zur Verhandlung die Chance zu gestehen. Wenn beide nicht gestehen, kommen sie mit zwei Jahren Gefängnis davon (unerlaubter Waffenbesitz und ähnliche Delikte). Wenn einer gesteht, kommt er nach der Kronzeugenregelung frei, während der andere für 10 Jahre sitzen muß. Gestehen beide, gibt es keinen Kronzeugen mehr. Aber geständige Bankräuber müssen nur für 5 Jahre sitzen. Hier geht es also um die Ehre. Während Winnetou und Old Shatterhand den Verlockungen widerstehen würden (in ähnlicher Lage, natürlich sind sie keine Bankräuber), würden typische Banditen gestehen. In der Tat ist dies der einzige Gleichgewichtspunkt, während die Strategie „Klappe halten" für beide besser ist. Dieses Strategienpaar ist aber äußerst instabil. Es läßt sich offensichtlich nur durch besonderes Vertrauen oder durch Verträge stabilisieren.

Das Beispiel ließ sich in der Zeit des Kalten Krieges auf die Rüstungsspirale in der USA und der UdSSR anwenden. Es war stets klar, daß beiderseitiges Abrüsten schön ist. Aber beide Seiten hätten noch lieber heimlich aufgerüstet, während der andere abrüstet. Und so haben sie beide jahrelang aufgerüstet. Diesen beiden „Spielern" waren Verhandlungen nicht verboten, sie konnten Verträge schließen. Nur gab es keine Instanz, die Vertragsverstöße überzeugend sanktioniert hätte. Dadurch war man praktisch im nicht kooperativen Fall.

Ein vernünftiger Einwand ist: Nun, die beiden spielen doch nicht einmal, sondern jeden Monat (oder eine andere Zeitspanne) wieder. Dies soll bedeuten, die gegenseitige Ausspähung war genügend sicher, um Verstöße zu entdecken. Sollte also jemand aufrüsten, könnte der andere mit einer kleinen Zeitverzögerung nachrüsten. Dieses Risiko wäre doch aufgrund der großen Hoffnung tragbar gewesen?! Nehmen wir also an, daß das Spiel k, z. B. $k = 100$ Runden gespielt wird. Nur hat sich leider nichts geändert. Das letzte Spiel ist wie ein einmaliges Spiel, d. h. beide rüsten auf. Wenn das aber klar ist, ist das vorletzte Spiel wie ein einmaliges Spiel, usw. Es ist also wieder nur die Aufrüstungsstrategie im Gleichgewicht.

Was ist aber, wenn die Zahl der Spielrunden nicht feststeht? Eine Strategie könnte folgendermaßen aussehen: Ich rüste ab und vertraue, daß der andere abrüstet. Sollte der andere einmal betrügen, werde ich eine Runde lang aufrüsten und dann zur Abrüstung zurückkehren. Die Drohung verpufft nie, da das Spiel nie zu Ende ist. Wenn beide diese Strategien benutzen, rüsten sie ewig ab. Sollte ein Spieler abweichen, kann er in dieser Runde 2 Einheiten gewinnen (0 statt $\Leftrightarrow 2$). In der nächsten Runde erhält er jedoch bestenfalls $\Leftrightarrow 5$. In den beiden Runden erhält er zusammen höchstens $\Leftrightarrow 5$, während er bei der Abrüstungsstrategie $\Leftrightarrow 4$ erhalten hätte. Die Abrüstungsstrategien verbunden mit den Drohstrategien befinden sich also im Gleichgewicht und führen zur Abrüstung.

Eine Bemerkung am Rande: Breschnew hatte in dem System der UdSSR einen „unendlichen" (besser unbegrenzten) Horizont vor sich, während der Spielraum von Nixon, Carter oder Reagan jeweils auf 4 Jahre beschränkt war.

7.4 Die Nash–Lösung für kooperative Spiele

Wir betrachten nun kooperative Spiele. Wieder beschränken wir uns nur aufgrund der einfacheren Darstellung auf $n = 2$ Spieler. Jeder Spielausgang ergibt einen Punkt $(u, v) \in \mathbb{R}^2$, wobei u die Auszahlung an Spieler I und v die Auszahlung an Spieler II beschreibt. Wir erinnern daran, daß Auszahlung eigentlich den Nutzen des jeweiligen Spielers beschreibt und die Auszahlung nicht in Geldeinheiten erfolgen muß. Da die Spieler nun aber auch Lotterien vertraglich vereinbaren können, sind auch alle konvexen Linearkombinationen der Spielausgänge möglich. Wir haben schon in Kap. 7.3 diskutiert, daß es stabiler ist, Lösungen neu zu konstruieren oder durch Seitenzahlungen zu garantieren, die die gleiche Lösung garantieren. In jedem Fall besteht die Verhandlungsmenge S aus der konvexen Hülle der Punkte, die direkte

Spielausgänge darstellen. Wir verallgemeinern hier sogar etwas, indem wir als Verhandlungsmengen S beliebige konvexe und kompakte Teilmengen des $\mathbb{R}^2$ zulassen. Dies soll bedeuten, daß genau die Nutzenpaare $(u, v) \in S$ realisierbar sind. Welcher Punkt in S stellt nun eine „gerechte" Verhandlungslösung dar? Eine solche Lösung wäre für einen neutralen Schlichter gerecht. Im konkreten Fall ist es nicht verboten, mehr anzustreben, als die gerechte Lösung vorschreibt.

Wie können wir zu einer gerechten Lösung gelangen? Nash ist den Weg gegangen, Axiome, die von allen akzeptiert werden können, aufzustellen und dann aus diesen Axiomen die Verhandlungslösung abzuleiten. Zunächst stellen wir fest, daß uns S als Information nicht genügt. Wir benötigen noch ein Nutzenpaar $(u^*, v^*) \in S$ als Konfliktlösung. Dies sollten Werte sein, die die Spieler ohne Kooperation erreichen können. Für Spieler I ist dies für seine Auszahlungsmatrix A der Wert des Zwei-Personen-Nullsummenspiels mit Auszahlungsmatrix A. Spieler II kann v^* analog zu seiner Auszahlungsmatrix B berechnen.

Definition 7.4.1: Ein Verhandlungsproblem ist gegeben durch die Verhandlungsmenge $S \subseteq \mathbb{R}^n$, die konvex und kompakt sein muß, und durch eine Konfliktlösung $(u_1^*, \ldots, u_n^*) \in S$.

Wir suchen nun eine Abbildung φ, die jedem Verhandlungsproblem $(S, u_1^*, \ldots, u_n^*)$ eine (gerechte) Verhandlungslösung

$$\varphi(S, u_1^*, \ldots, u_n^*) = (\overline{u}_1, \ldots, \overline{u}_n) \in S$$

zuordnet. Nash hat die folgenden Axiome für φ aufgestellt.

Axiom 1: „Individuelle Rationalität".
Für alle i soll $\overline{u}_i \geq u_i^*$ sein. Niemand wird sich in Verhandlungen mit weniger begnügen als er ohne Verhandlungsergebnis erhält.

Axiom 2: „Pareto-Optimalität".
Aus $(u_1, \ldots, u_n) \in S$ und $u_i \geq \overline{u}_i$ für alle i folgt $u_i = \overline{u}_i$ für alle i. Dieses Axiom drückt aus, daß die Spieler nicht im Vergleich gewinnen wollen oder andere bestrafen wollen. Nach geeigneter Wahl ihrer Nutzenfunktionen gönnen sie anderen beliebig viel, wenn sie mit der eigenen Auszahlung zufrieden sind.

Axiom 3: „Unabhängigkeit von irrelevanten Alternativen".
Falls $T \subseteq S$ und $(\overline{u}_1, \ldots, \overline{u}_n) \in T$, ist $\varphi(T, u_1^*, \ldots, u_n^*) = (\overline{u}_1, \ldots, \overline{u}_n)$. Wenn eine gerechte Lösung für ein Verhandlungsproblem anerkannt ist, sollte sich diese Lösung nicht ändern, wenn die Konfliktlösung sich nicht ändert, die Verhandlungslösung zulässig bleibt und nur einige Nutzenkombinationen, die wir vorher nicht gewählt haben, verboten werden. Dieses Axiom hat es in sich. Nehmen wir einmal an, daß alle Auszahlungen (u, v) mit $u \geq 0$, $v \geq 0$ und $u + v \leq 1$ die Verhandlungsmenge bilden und $(u^*, v^*) = (0, 0)$ Konfliktlösung ist. Die Symmetrie legt es nahe (siehe auch Axiom 5), daß $(1/2, 1/2)$ gerechte Verhandlungslösung ist. Wenn wir nun die Verhandlungsmenge durch die Forderung $u \leq 1/2$ einschränken, folgt aus Axiom 3, daß weiterhin $(1/2, 1/2)$ die gerechte Verhandlungslösung ist. Dann bekommt Spieler I sein Maximum, während Spieler II mehr bekommen könnte!

Axiom 4: „Unabhängigkeit von linearen Transformationen".
Es sei T das Bild von S unter den linearen Abbildungen

$$u_i' = \alpha_i u_i + \beta_i,\ \alpha_i > 0,\ \beta_i \in \mathbb{R},\ 1 \leq i \leq n.$$

Dann ist

$$\varphi(T, \alpha_1 u_1^* + \beta_1, \ldots, \alpha_n u_n^* + \beta_n) = (\alpha_1 \overline{u}_1 + \beta_1, \ldots, \alpha_n \overline{u}_n + \beta_n).$$

Dieses Axiom besagt nur, daß ein Spieler, der seine Nutzenfunktion durch eine äquivalente Nutzenfunktion (s. Kap. 6.1) ersetzt, dadurch in der Verhandlungslösung nichts gewinnt oder verliert.

Axiom 5: „Symmetrie".
Wenn S die Eigenschaft hat, daß mit $(u_1, \ldots, u_n)$ auch jede Permutation in S enthalten ist und wenn $u_1^* = \ldots = u_n^*$ ist, dann gilt auch $\overline{u}_1 = \ldots = \overline{u}_n$. Dieses Axiom impliziert, daß alle Spieler prinzipiell gleich „mächtig" sind, es sei denn, das Verhandlungsproblem weist ihnen unterschiedliche Stärken zu. Ein Spieler soll nicht deswegen weniger bekommen, weil z.B. seine Spielernummer ungerade ist.

Sicher läßt sich die Reihe sinnvoller Axiome fortsetzen. Die hier aufgeführten sind jedoch die unumstrittensten Axiome (höchstens Axiom 3 ist angreifbar), und sie sichern bereits die Eindeutigkeit der Verhandlungslösung. Wir werden die folgenden Ergebnisse für beliebiges n formulieren und teilweise nur für $n = 2$ beweisen. Die Beweise für allgemeines n benötigen keine neuen Ideen, sind jedoch komplizierter aufzuschreiben. Wir beginnen mit einer Definition.

Definition 7.4.2: Für eine Verhandlungsmenge S sei

$$\begin{aligned} S_> &:= \{(u_1, \ldots, u_n) \in S \mid \forall i : u_i > u_i^*\}. \\ S_\geq &:= \{(u_1, \ldots, u_n) \in S \mid \forall i : u_i \geq u_i^*\}. \end{aligned}$$

Wir nennen S reduziert, wenn $S = S_\geq$ und $S_> \neq \emptyset$ ist.

Wir können uns bei der Analyse von Verhandlungsmengen auf reduzierte Verhandlungsmengen beschränken, wie die folgenden Überlegungen zeigen:

Ein $u \in S$ mit $u_i < u_i^*$ kann wegen Axiom 1 keine gerechte Verhandlungslösung sein, daher brauchen wir solche Vektoren nicht zu betrachten. Angenommen, für alle i gäbe es ein $u^{(i)} \in S$ mit $u_i^{(i)} > u_i^*$. Dann ist die konvexe Linearkombination $V := \frac{1}{n} \cdot (u^{(1)} + \cdots + u^{(n)})$ in S, da S konvex ist. Wegen „$S = S_\geq$“ und nach Wahl der $u^{(i)}$ ist V in jeder Komponente größer als u^* und somit $S_> \neq \emptyset$. Wenn $S_>$ leer wäre, gäbe es also ein i mit der Eigenschaft, daß alle $u \in S$ $u_i = u_i^*$ haben. Für Spieler i weiß man dann aber auf jeden Fall, daß er in der gerechten Lösung u_i^* bekommt. Man kann das Problem dann durch Entfernen von Spieler i reduzieren.

Lemma 7.4.3: *Wenn S reduziert ist, so gilt: Es existiert ein eindeutig bestimmter Punkt $(u_1', \ldots, u_n')$ aus S, der die folgende Funktion g maximiert:*

$$g(u_1, \ldots, u_n) = (u_1 \Leftrightarrow u_1^*) \cdots (u_n \Leftrightarrow u_n^*).$$

Beweis: Wieder führen wir den Beweis nur für n=2. Da S konvex und kompakt ist, nimmt die stetige Funktion g ihr Maximum auf S an. Sei also $\max := \max_{x \in S} g(x)$. Da S reduziert und somit $S_> \neq \emptyset$ ist, ist $\max > 0$. Nehmen wir nun an, daß es zwei verschiedene Punkte $u', u'' \in S$ gibt mit $g(u') = g(u'') = \max$. Da S konvex ist, gehört $(u' + u'')/2$ zu S. Es gilt:

$$\begin{aligned} g((u' + u'')/2) &= \left(\frac{u_1' + u_1''}{2} \Leftrightarrow u_1^*\right)\left(\frac{u_2' + u_2''}{2} \Leftrightarrow u_2^*\right) \\ &= \frac{(u_1' \Leftrightarrow u_1^*)(u_2' \Leftrightarrow u_2^*)}{2} + \frac{(u_1'' \Leftrightarrow u_1^*)(u_2'' \Leftrightarrow u_2^*)}{2} + \frac{(u_1' \Leftrightarrow u_1'')(u_2'' \Leftrightarrow u_2')}{4}. \\ &= \frac{g(u')}{2} + \frac{g(u'')}{2} + \frac{(u_1' \Leftrightarrow u_1'')(u_2'' \Leftrightarrow u_2')}{4} \\ &= \max + \frac{(u_1' \Leftrightarrow u_1'')(u_2'' \Leftrightarrow u_2')}{4} \end{aligned}$$

Falls $u_1' = u_1''$, folgt aus $g(u') = g(u'') = \max > 0$, daß $u_2' = u_2''$ ist im Widerspruch dazu, daß $u' \neq u''$ ist. Falls $u_1' > u_1''$ ($u_1' < u_1''$), folgt aus $g(u') = g(u'')$, $u_2' < u_2''$ ($u_2' > u_2''$). Also ist der dritte Summand positiv im Widerspruch zur Maximalität von u' und u''. □

Damit ist für eine reduzierte Verhandlungsmenge S der Punkt $(\overline{u}_1, \ldots, \overline{u}_n)$ eindeutig definiert, der die Funktion g auf S maximiert.

Lemma 7.4.4: *Sei $h(u_1, u_2) = (\overline{u}_1 \Leftrightarrow u_1^*)u_2 + (\overline{u}_2 \Leftrightarrow u_2^*)u_1$. Wenn S reduziert ist, so gilt $h(u_1, u_2) \leq h(\overline{u}_1, \overline{u}_2)$ für alle $(u_1, u_2) \in S$.*

Beweis: Wir nehmen an, daß $h(u_1, u_2) > h(\overline{u}_1, \overline{u}_2)$ und $(u_1, u_2) \in S$ ist. Die Strecke zwischen (u_1, u_2) und $(\overline{u}_1, \overline{u}_2)$ liegt wegen der Konvexität von S ganz in S. Da h in den Variablen u_1 und u_2 linear ist, folgt aus $h(u_1, u_2) > h(\overline{u}_1, \overline{u}_2)$

$$h(u_1 \Leftrightarrow \overline{u}_1, u_2 \Leftrightarrow \overline{u}_2) > 0.$$

Wir betrachten den Punkt $(u_1' = \overline{u}_1 + \varepsilon(u_1 \Leftrightarrow \overline{u}_1), u_2' = \overline{u}_2 + \varepsilon(u_2 \Leftrightarrow \overline{u}_2))$ auf der Strecke zwischen (u_1, u_2) und $(\overline{u}_1, \overline{u}_2)$, d. h. $0 < \varepsilon < 1$. Dann gilt

$$\begin{aligned} g(u_1', u_2') &= (u_1' \Leftrightarrow u_1^*)(u_2' \Leftrightarrow u_2^*) \\ &= (\overline{u}_1 \Leftrightarrow u_1^* + \varepsilon(u_1 \Leftrightarrow \overline{u}_1))(\overline{u}_2 \Leftrightarrow u_2^* + \varepsilon(u_2 \Leftrightarrow \overline{u}_2)) \\ &= (\overline{u}_1 \Leftrightarrow u_1^*)(\overline{u}_2 \Leftrightarrow u_2^*) + \varepsilon[(u_1 \Leftrightarrow \overline{u}_1)(\overline{u}_2 \Leftrightarrow u_2^*) + (\overline{u}_1 \Leftrightarrow u_1^*)(u_2 \Leftrightarrow \overline{u}_2)] \\ &\quad + \varepsilon^2(u_1 \Leftrightarrow \overline{u}_1)(u_2 \Leftrightarrow \overline{u}_2) \\ &= g(\overline{u}_1, \overline{u}_2) + \varepsilon h(u_1 \Leftrightarrow \overline{u}_1, u_2 \Leftrightarrow \overline{u}_2) + \varepsilon^2(u_1 \Leftrightarrow \overline{u}_1)(u_2 \Leftrightarrow \overline{u}_2). \end{aligned}$$

Der zweite Summand ist positiv und für $\varepsilon \to 0$ größer als der dritte Summand. Also folgt $g(u_1', u_2') > g(\overline{u}_1, \overline{u}_2)$ für ein geeignet kleines ε im Widerspruch zur Definition von $(\overline{u}_1, \overline{u}_2)$. □

Wir kommen nun zur Definition der Nash-Lösung. Für alle Spieler $i \in I$, für die es kein $(u_1, \ldots, u_n) \in S$ mit $u_i > u_i^*$ gibt, setzen wir $\overline{u}_i = u_i^*$. Dann wird das Spiel auf die anderen Spieler reduziert, d. h. es werden die Spieler $i \notin I$ betrachtet und nur die Verhandlungsergebnisse, die mit den Spielern $i \in I$ um u_i ergänzt werden können. Für diese Spieler wird $\overline{u}_i$, $i \notin I$, als eindeutiger Punkt definiert, der die zugehörige g-Funktion maximiert.

Satz 7.4.5: *Es gibt nur eine Verhandlungslösung, die die Axiome 1 – 5 erfüllt. Diese sogenannte Nash-Lösung ist der Vektor* $(\overline{u}_1, \ldots, \overline{u}_n)$.

Beweis: Für die Spieler $i \in I$ ist nur die Auszahlung $\overline{u}_i = u_i^*$ nach Axiom 1 möglich. Also ist die Reduktion des Spiels korrekt.

Wir nehmen nun an, daß nur noch zwei Spieler betrachtet werden müssen.

Zunächst zeigen wir, daß die genannte Lösung die fünf Axiome erfüllt. Dies gilt für Axiom 1 direkt. Die Funktion g hat nun ein positives Maximum. Dann ist klar, daß das Maximum nur an Pareto-optimalen Punkten angenommen werden kann. Für die Maximierung gilt die Unabhängigkeit von irrelevanten Alternativen offensichtlich. Ein maximaler Punkt bleibt maximal, wenn andere Alternativen verboten werden. Die linearen Transformationen wirken sich auf g folgendermaßen aus. Aus dem Faktor $u_i \Leftrightarrow u_i^*$ wird $\alpha_i u_i + \beta_i \Leftrightarrow (\alpha_i u_i^* + \beta_i) = \alpha_i(u_i \Leftrightarrow u_i^*)$, d. h. wir erhalten die gleiche Funktion mit einem konstanten positiven Faktor. Es werden also auch die maximalen Punkte linear transformiert. Das letzte Axiom ist das Symmetrie-Axiom. Die Funktion g ist ebenfalls symmetrisch. Wenn also $(\overline{u}_1, \overline{u}_2)$ maximaler Punkt ist, ist für symmetrisches S, da $u_1^* = u_2^*$, auch $(\overline{u}_2, \overline{u}_1)$ maximal. Aus der Eindeutigkeit maximaler Punkte folgt also $\overline{u}_1 = \overline{u}_2$.

Es bleibt zu zeigen, daß kein anderer Vektor aus S die Axiome erfüllen kann. Dazu machen wir unser Problem (S, u_1^*, u_2^*) handlicher. Sei

$$U := \{(u_1, u_2) \mid h(u_1, u_2) \leq h(\overline{u}_1, \overline{u}_2)\}.$$

Nach Lemma 7.4.4 ist $S \subseteq U$. Falls also (U, u_1^*, u_2^*) die eindeutige Lösung $(\overline{u}_1, \overline{u}_2)$ hat, gilt dies wegen des Axioms der Unabhängigkeit von irrelevanten Alternativen auch für (S, u_1^*, u_2^*).

Die Menge U hat die angenehme Eigenschaft, daß sie linear in eine symmetrische Menge transformiert werden kann. Sei

$$u_1' := \frac{u_1 \Leftrightarrow u_1^*}{\overline{u}_1 \Leftrightarrow u_1^*} \text{ und } u_2' := \frac{u_2 \Leftrightarrow u_2^*}{\overline{u}_2 \Leftrightarrow u_2^*}$$

die lineare Transformation. Dabei ist zu beachten, daß $\overline{u}_1 \Leftrightarrow u_1^*$ und $\overline{u}_2 \Leftrightarrow u_2^*$ positive konstante Faktoren sind. Wie sieht das Bild T von U unter dieser linearen Transformation aus? Es gilt

$$
\begin{aligned}
(u_1,u_2)\in U &\Leftrightarrow (\overline{u}_1 \Leftrightarrow u_1^*)u_2 + (\overline{u}_2 \Leftrightarrow u_2^*)u_1 \le (\overline{u}_1 \Leftrightarrow u_1^*)\overline{u}_2 + (\overline{u}_2 \Leftrightarrow u_2^*)\overline{u}_1\\
&\Leftrightarrow (\overline{u}_1 \Leftrightarrow u_1^*)u_2 + (\overline{u}_2 \Leftrightarrow u_2^*)u_1 \Leftrightarrow u_1^*(\overline{u}_2 \Leftrightarrow u_2^*) \Leftrightarrow u_2^*(\overline{u}_1 \Leftrightarrow u_1^*)\\
&\quad \le (\overline{u}_1 \Leftrightarrow u_1^*)\overline{u}_2 + (\overline{u}_2 \Leftrightarrow u_2^*)\overline{u}_1 \Leftrightarrow u_1^*(\overline{u}_2 \Leftrightarrow u_2^*) \Leftrightarrow u_2^*(\overline{u}_1 \Leftrightarrow u_1^*)\\
&\Leftrightarrow (\overline{u}_1 \Leftrightarrow u_1^*)(u_2 \Leftrightarrow u_2^*) + (\overline{u}_2 \Leftrightarrow u_2^*)(u_1 \Leftrightarrow u_1^*)\\
&\quad \le (\overline{u}_1 \Leftrightarrow u_1^*)(\overline{u}_2 \Leftrightarrow u_2^*) + (\overline{u}_2 \Leftrightarrow u_2^*)(\overline{u}_1 \Leftrightarrow u_1^*) = 2(\overline{u}_1 \Leftrightarrow u_1^*)(\overline{u}_2 \Leftrightarrow u_2^*)\\
&\Leftrightarrow \frac{u_1 \Leftrightarrow u_1^*}{\overline{u}_1 \Leftrightarrow u_1^*} + \frac{u_2 \Leftrightarrow u_2^*}{\overline{u}_2 \Leftrightarrow u_2^*} \le 2\\
&\Leftrightarrow u_1' + u_2' \le 2.
\end{aligned}
$$

Also ist $T = \{(u_1' + u_2') \,|\, u_1' + u_2' \le 2\}$. Die Konfliktlösung wird offensichtlich auf $P = (0,0)$ transformiert. Die Lösung von $(T,0,0)$ ist mit den Axiomen leicht. Nach dem Symmetrie-Axiom erhalten beide Spieler das gleiche. Aber unter derartigen Punkten ist nur $(1,1)$ Pareto-optimal. Wegen des Axioms der Unabhängigkeit von linearen Transformationen, erhalten wir die eindeutige Lösung von (U, u_1^*, u_2^*) durch Rücktransformation des Punktes $(1,1)$. Daher gilt

$$1 = \frac{u_1 \Leftrightarrow u_1^*}{\overline{u}_1 \Leftrightarrow u_1^*} \quad \text{und} \quad 1 = \frac{u_2 \Leftrightarrow u_2^*}{\overline{u}_2 \Leftrightarrow u_2^*}.$$

Also folgt, daß $(\overline{u}_1, \overline{u}_2)$ die Lösung von (U, u_1^*, u_2^*) und damit auch von (S, u_1^*, u_2^*) ist. □

Für gegebene Verhandlungsprobleme können die Nash–Lösungen zumindest numerisch recht effizient berechnet werden.

Wir wollen den Nutzen der Nash–Theorie an zwei Beispielen erläutern.

Beispiel 7.4.6: Wie teilen Reich und Arm?
Wir nehmen an, daß der Nutzen von Geld logarithmisch wächst, d. h. eine Person, die x DM besitzt und y DM hinzubekommt, erzielt dadurch einen Nutzen von $\log(x+y) \Leftrightarrow \log x = \log(1 + \frac{y}{x})$. Wir nehmen nun an, daß zwei Personen sich 1000 DM teilen dürfen, wenn sie sich über die Aufteilung einigen können. Die eine Person ist reich, sie besitzt 1000000 DM und verschwendet keinen Gedanken an „sozialen Klimbim“. Wenn sie nun u DM erhält, beträgt der Nutzen

$$\log(1 + \frac{u}{1000000}) \approx \frac{u}{1000000}.$$

Da lineare Transformationen keine Rolle spielen, bewerten wir den Nutzen mit u. Die andere Person ist arm, sie besitzt nur 1000 DM. Wenn sie bei der Verhandlung v DM erhält, beträgt der Nutzen

$$\log(1 + \frac{v}{1000}).$$

Da $0 \le v \le 1000$, ist hier die lineare Approximation nicht sehr genau. Die Konfliktlösung ist natürlich $u^* = 0$, $v^* = 0$. Was ist nun eine Verhandlungslösung, die der Verhandlungsstärke entspricht? Von gerecht möchte ich hier nicht reden. Für die Pareto-optimalen Punkte gilt $u + v = 1000$. Also ist die Nashfunktion

$$g(u,v) = u \log(1 + \frac{v}{1000})$$

für $v = 1000 \Leftrightarrow u$ zu maximieren. Wir erhalten die Funktion

$$g^*(u) = u \log(2 \Leftrightarrow \frac{u}{1000}),$$

die auf $[0, 1000]$ zu maximieren ist. An den Rändern sind die Funktionswerte 0. Wir bilden also die 1. Ableitung

$$g^{*\prime}(u) = \log(2 \Leftrightarrow \frac{u}{1000}) \Leftrightarrow u/[(2 \Leftrightarrow \frac{u}{1000})1000 \ln 2]$$

Es gilt $g^{*\prime}(u) = 0$ genau dann, wenn

$$\log\left(2 \Leftrightarrow \frac{u}{1000}\right) = \frac{u}{2000 \Leftrightarrow u} / \ln 2$$

ist. Diese Gleichung kann nur numerisch approximativ gelöst werden. Im Ergebnis erhält der Reiche ca. 545 DM und der Arme nur 455 DM. Die Nash–Theorie ist also in der Lage, Machtverhältnisse zu berücksichtigen.

Beispiel 7.4.7: KSZE.
Anfang der 70er Jahre wurde bereits über eine europäische Sicherheitskonferenz diskutiert. Stattgefunden hat sie dann 1975 unter dem Namen KSZE in Helsinki. Vor solchen Konferenzen sollten die Unterhändler nicht nur wissen, was sie wollen, sondern auch, was die anderen wollen. Nur so können Kompromisse bewertet werden. Nur dies kann zu stabilen Verträgen führen. In den Planungsstäben finden daher Planspiele statt, deren Ergebnisse leider nicht veröffentlicht werden, damit die eigene Verhandlungsposition nicht geschwächt wird.

1971 haben Politologen unter Leitung von Reich ein Planspiel gespielt und das Resultat veröffentlicht. Es wurden die acht „wichtigsten" Nationen USA, GB, FRA und BRD einerseits und UdSSR, POL, CSSR und DDR andererseits simuliert. Die Gruppen, die die Nationen „spielten", erhielten eine lange Liste von Statements, die eventuell Teil des Vertragstextes sein konnten. Sie haben diesen Statements Nutzenwerte zugeordnet. Mit diesen Nutzenwerten wurde eine Nash–Analyse durchgeführt. Die Lösung gab allen Teilnehmern einen positiven Nutzen gegenüber der Konfliktlösung „kein Vertrag". Allerdings waren für die USA und die BRD die Nutzenwerte nur geringfügig über der Konfliktlösung, so daß die Lösung als nicht sehr stabil bezeichnet werden mußte.

Darüber hinaus simulierten die Nationen auch drei Tage lang eine Sicherheitskonferenz. Diese Konferenz endete „nach gutem und konstruktivem" Beginn mit der Konfliktlösung.

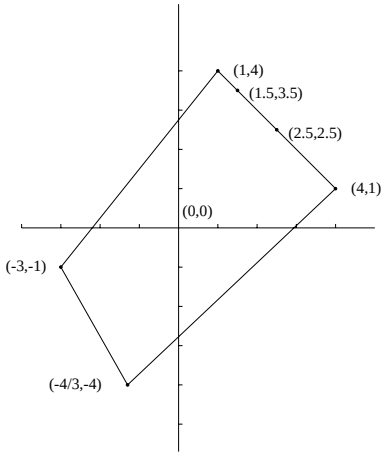
Was hatte dieses Experiment mit der Wirklichkeit zu tun? Es wurden nur Statements über die politische Lage in Europa verhandelt. Im Mittelpunkt der Verhandlungen standen die Unverletzlichkeit von Grenzen, die Anerkennung des Status Quo, usw. An diesen Statements hatten insbesondere die UdSSR und die DDR Interesse, während den USA und der BRD diese Statements eher Probleme bereiteten. So wurde die Idee der europäischen Sicherheitskonferenz auch wesentlich von östlicher Seite betrieben. Als die Konferenz zustande kam, ergab sich eine Verhandlungskrise wie in dem beschriebenen Planspiel. Die USA und die BRD „drohten" mit der Konfliktlösung, die ihnen auch kaum einen Verlust, außer dem Imageverlust, Verursacher des Scheiterns zu sein, bereitet hätte. Was ist nun zu tun, wenn diese Lage allen Beteiligten klar ist? Im Gegensatz zum Planspiel gab es die Möglichkeit, das Spiel zu verändern. Der bisherige Verhandlungsrahmen wurde zum „Korb 1" deklariert, und zwei neue Körbe wurden „erfunden". In Korb 2 lagen die wirtschaftlichen Beziehungen, Statements, über die sich alle einig waren und die die Verhandlungsstimmung positiv beeinflußten, ohne allerdings das Grundproblem zu lösen. In den später berühmt gewordenen Korb 3 wurden die Menschenrechte gepackt. Vertragliche Zusicherungen, die Menschenrechte einzuhalten, bereiteten den östlichen Nationen Probleme. Tatsächlich gründeten sich später Menschenrechtsbewegungen, die sich ausdrücklich auf Helsinki beriefen. Für die USA und die BRD war es dagegen von großem Nutzen, diese Statements vertraglich zu regeln. Nun war eine neue Verhandlungssituation geschaffen. Alle konnten profitieren, und alle mußten Kröten schlucken. In der Summe konnte allerdings jede Nation aus dem Vertrag einen größeren positiven Nutzen schlagen. So wurde ein wichtiger Vertrag tatsächlich beschlossen.

Es konnte nicht überprüft werden, ob der endgültige Vertrag der Nash Lösung in dem neuen erweiterten Verhandlungsspiel nahe kommt, da für die neuen Statements keine Nutzenschätzungen vorlagen. An diesem Beispiel wurde aber deutlich, daß eine Nash–Analyse zur Beurteilung von Verhandlungssituationen beitragen kann.

Beispiel 7.4.8: Drohungen Wir betrachten die folgenden Auszahlungsmatrizen.

$$(A, B) = \begin{pmatrix} (1,4) & (\Leftrightarrow 4/3, \Leftrightarrow 4) \\ (\Leftrightarrow 3, \Leftrightarrow 1) & (4,1) \end{pmatrix}$$

Wenn wir die Konfliktlösung auf die übliche Weise als Maximin-Werte berechnen, ergibt sich $u^* = v^* = 0$.



Ausgehend von der Konfliktlösung $(0,0)$ ergibt sich offensichtlich die Nash-Lösung $(2.5, 2.5)$. Dies gefällt Spieler II überhaupt nicht. Er sagt sich, daß er mehr Macht als Spieler I hat. Er droht, im Konfliktfall Strategie 1 zu spielen. Spieler I kann dann im Konfliktfall zwischen Strategie 1 (Spieler II erhält viel mehr) und Strategie 2 (Spieler II verliert weniger als er) wählen oder eine gemischte Strategie spielen. Den Abstand zu Spieler II minimiert er mit Strategie 2. Konfliktlösung ist dann $(\Leftrightarrow 3, \Leftrightarrow 1)$. Bezogen auf diese Konfliktlösung ändert sich die Nash-Lösung. Die Pareto-optimalen Punkte sind $(u, 5 \Leftrightarrow u)$ mit $(1 \leq u \leq 4)$. Es ergibt sich die Nash–Funktion

$$\begin{aligned} g^*(u) &= (u \Leftrightarrow (\Leftrightarrow 3))(5 \Leftrightarrow u \Leftrightarrow (\Leftrightarrow 1)) \\ &= (u+3)(6 \Leftrightarrow u) = \Leftrightarrow u^2 + 3u + 18. \end{aligned}$$

Es ist $g^{*\prime}(u) = \Leftrightarrow 2u + 3 = 0$ genau dann, wenn $u = 1.5$ ist. Die neue Nash-Lösung ist also $(1.5, 3.5)$ und berücksichtigt das Drohpotential von Spieler II.

Es läßt sich analog zu Kapitel 7.3 zeigen, daß es stets Drohstrategien gibt, die sich im Gleichgewicht befinden, d. h. eine Abweichung eines Spielers würde seine Nash-Lösung nicht verbessern. Aber wir geraten auch in die gleichen Probleme, wie sie allgemein der Gleichgewichtstheorie anhaften.

7.5 Die Bedeutung von Koalitionen in kooperativen Spielen

Die Nash-Lösung ergibt sich für Zwei-Personen-Spiele auf analoge Weise wie für n-Personen-Spiele, da wir davon ausgehen, daß jeder Spieler das beste für sich herausholen will. Für $n > 2$ werden jedoch Koalitionen möglich. Hierbei wird das Stabilitätsproblem noch gravierender.

Beispiel 7.5.1: Drei Spieler stehen vor dem Problem, eine Koalition zu bilden. Gelingt dies nicht, findet keine Auszahlung statt, ebenso bei der Dreier-Koalition. Gelingt eine Zweier-Koalition, muß der Ausgeschlossene je eine Einheit an jeden der beiden Koalitionäre zahlen. Zu den drei Koalitionen gehören die Auszahlungen $(\Leftrightarrow 2, 1, 1)$, $(1, \Leftrightarrow 2, 1)$ und $(1, 1, \Leftrightarrow 2)$, die alle stabil sind. Wir nehmen nun abweichend an, daß beim Zustandekommen der Koalition $\{2,3\}$ der erste Spieler 1.1 Einheiten an Spieler 2 und 0.9 Einheiten an Spieler 3 zahlen muß. Ist das nicht schön für Spieler 2? Nein, denn die von ihm gebildeten Koalitionen werden instabil, da Spieler 1 den Spieler 3 leicht aus der Koalition $\{2,3\}$ locken kann. Auch die Koalition $\{1,2\}$ ist nicht stabil, da Spieler 2 selbst verlockt ist, zu Spieler 3 zu wechseln. Im Endeffekt ist nur die Koalition $\{1,3\}$ stabil und Spieler 2 der Verlierer. Dies kann er ausgleichen, wenn Seitenzahlungen erlaubt sind. Spieler 2 könnte in der Koalition $\{2,3\}$ 0.1 Einheiten seiner Auszahlung an Spieler 3 weitergeben.

Wir nehmen im folgenden an, daß die Nutzenfunktionen so gewählt sind, daß Nutzen direkt (im Nutzenverhältnis 1:1) transferierbar ist. Die nun folgende Theorie ist also im Gegensatz zur Nash–Theorie nicht unabhängig von linearen Transformationen der Nutzenfunktionen.

Definition 7.5.2: Sei $N = \{1, \ldots, n\}$ die Menge der n Spieler. Jede nichtleere Teilmenge von N heißt Koalition.

Eine Koalition S kann sich den Maximinwert des Zwei-Personen-Spiels zwischen dem Superspieler S und dem Superspieler $N \Leftrightarrow S$ sichern. Diesen Wert nennen wir $v(S)$. Zusätzlich definieren wir $v(\emptyset) := 0$. Sind S und T disjunkte Koalitionen, können sie sich natürlich zusammen mindestens so viel sichern, wie sie sich einzeln sichern könnten, d. h. es gilt

$$v(S \cup T) \geq v(S) + v(T), \text{ falls } S \cap T = \emptyset.$$

Wir reduzieren die Spiele nun darauf, daß die Spielregeln gar nicht mehr interessieren, sondern nur noch die Machtstruktur der Koalitionen.

Definition 7.5.3: Ein n-Personen-Spiel in Form der charakteristischen Funktion ist eine Abbildung v auf der Potenzmenge von $N = \{1, \ldots, n\}$, die die Bedingungen $v(\emptyset) = 0$ und $v(S \cup T) \geq v(S) + v(T)$ für $S \cap T = \emptyset$ erfüllt.

Wir suchen nun Auszahlungsvektoren $x = (x_1, \ldots, x_n)$, die bei gegebener charakteristischer Funktion stabil sind. Eine Koalition S kann, da Seitenzahlungen erlaubt sind, den Betrag $v(S)$ beliebig unter sich aufteilen. Wir stellen zunächst zwei notwendige Bedingungen an stabile Auszahlungsvektoren. Kein Spieler i wird akzeptieren, wenn er weniger als $v(\{i\})$, was er sich alleine sichern kann, erhält. Wenn alle zusammen weniger als $v(N)$ bekommen, könnten sich alle einigen und jedem etwas mehr auszahlen. Bei Nichteinigung kann es zwar zu derartigen Auszahlungsvektoren kommen, jedoch ist diese Situation nicht stabil.

Definition 7.5.4: Ein Vektor $x = (x_1, \ldots, x_n)$ heißt Imputation des n-Personen-Spiels v, wenn $x_1 + \ldots + x_n = v(N)$ und $x_i \geq v(\{i\})$ für alle i gelten. Die Menge aller Imputationen wird mit $I(v)$ bezeichnet.

Aus der zweiten Bedingung an charakteristische Funktionen folgt $v(\{1\}) + \ldots + v(\{n\}) \leq v(N)$. Falls hier Gleichheit gilt, gibt es nur die Imputation $(v(\{1\}), \ldots, v(\{n\}))$. Das Spiel ist also von der Koalitionskonstellation uninteressant und heißt unwesentlich. Wir behandeln nur wesentliche Spiele.

Für zwei Imputationen x und y gilt $x_1 + \ldots + x_n = y_1 + \ldots + y_n$. Falls $x \neq y$, gibt es Spieler i, die x bevorzugen, da $x_i > y_i$, und Spieler j, die y bevorzugen, da $y_j > x_j$. Welche dieser Spielergruppen kann und wird sich durchsetzen?

Definition 7.5.5: Seien $x, y \in I(v)$ und S eine Koalition. Die Imputation x dominiert y durch S (Notation $x >_S y$), wenn die folgenden Bedingungen gelten:

$$x_i > y_i \text{ für alle } i \in S \quad \text{sowie} \quad \sum_{i \in S} x_i \leq v(S).$$

„Imputation x dominiert y“, falls es eine Koalition S mit $x >_S y$ gibt.

Für eine dominierte Imputation gilt, daß es eine Koalition gibt, die aus eigener Kraft diese Imputation verhindern kann, wobei alle Teilnehmer dieser Koalition dabei gewinnen. Die Relation $<_S$ ist für festes S transitiv, dies gilt aber nicht für $<$. Aus $x <_S y$ und $y <_{S'} z$ kann nicht die Existenz einer Koalition S'' mit $x <_{S''} z$ gefolgert werden. Es kann nämlich ohne weiteres $x <_S y$ und $y <_{S'} x$ gelten, während $x <_{S''} x$ definitionsgemäß unmöglich ist. In Beispiel 7.5.1 haben Einerkoalitionen den Wert $\Leftrightarrow 2$, Zweierkoalitionen den Wert 2 und die Dreierkoalition den Wert 0. Jede Imputation wird dominiert. Es gibt immer zwei Spieler, die zusammen nicht die Auszahlung 2 erhalten. Sie können sich zusammentun und die Auszahlung 2 so aufteilen, daß jeder mehr bekommt als zuvor.

Wir wollen nun die Menge der Spiele strukturell untersuchen, äquivalente Spiele identifizieren und Spiele in eine Normalform bringen.

Definition 7.5.6: Zwei Spiele u und v heißen äquivalent, wenn es $r \in \mathbb{R}^+$ und $\alpha_1, \ldots, \alpha_n \in \mathbb{R}$ gibt, so daß für alle Koalitionen S gilt

$$v(S) = ru(S) + \sum_{i \in S} \alpha_i.$$

Bemerkung 1: Der Äquivalenzbegriff auf Spielen stellt eine Äquivalenzrelation dar.

Beweis: Natürlich ist v zu sich selber äquivalent ($r = 1$, $\alpha_i = 0$ für alle i). Die Relation ist symmetrisch, denn es gilt

$$u(S) = \frac{1}{r} v(S) \Leftrightarrow \sum_{i \in S} \alpha_i / r.$$

Schließlich ist die Relation transitiv:

Aus $v(S) = r \cdot u(S) + \sum_{i \in S} \alpha_i$ und $w(S) = r' \cdot v(S) + \sum_{i \in S} \alpha'_i$ folgt $w(S) = r'ru(S) + \sum_{i \in S} (r'\alpha_i + \alpha'_i)$. □

Wir charakterisieren nun einen Vertreter jeder Äquivalenzklasse.

Definition 7.5.7: Ein Spiel v heißt auf $(0,1)$ normiert, wenn $v(N) = 1$ und $v(\{i\}) = 0$ für alle i ist.

Satz 7.5.8: *Jedes wesentliche Spiel ist genau zu einem Spiel in* $(0,1)$*-Normierung äquivalent.*

Beweis: Zwei Spiele u und v in $(0,1)$-Normierung können nur äquivalent sein, wenn sie gleich sind. Die Bedingungen für die Koalitionen $\{i\}$ führen zu $\alpha_i = 0$, aus der Bedingung für N folgt $r = 1$. Es bleibt die Existenz eines zu v äquivalenten Spiels in $(0,1)$-Normierung zu zeigen. Dafür haben wir $n+1$ Unbekannte $r, \alpha_1, \ldots, \alpha_n$ und die Gleichungen

$$\begin{aligned} v(\{i\}) &= ru(\{i\}) + \alpha_i = \alpha_i \text{ für } 1 \leq i \leq n \text{ und} \\ v(N) &= ru(N) + \sum_{i \in N} \alpha_i = r + \sum_{i \in N} \alpha_i. \end{aligned}$$

Die ersten Gleichungen definieren die α-Werte und die letzte Gleichung dann r. Dabei ist $r > 0$, da wir über wesentliche Spiele reden. □

Wir können uns also auf Spiele in $(0,1)$-Normierung beschränken. Dann gelten folgende Bedingungen:

$$\begin{aligned} v(\emptyset) &= 0 \\ v(\{i\}) &= 0 \text{ für } i \in \{1, \ldots, n\} \\ v(N) &= 1 \\ v(S \cup T) &\geq v(S) + v(T) \text{ für } S \cap T = \emptyset. \end{aligned}$$

Zwei Klassen von Spielen sind von besonderem Interesse.

Definition 7.5.9: Ein Spiel v heißt symmetrisch, wenn $v(S)$ nur von $|S|$ abhängt. Ein Spiel in $(0,1)$-Normierung heißt einfach, wenn $v(S) \in \{0,1\}$ für alle Koalitionen S ist. Ein Spiel heißt einfach, wenn seine $(0,1)$-Normierung einfach ist.

Wir haben bereits gesehen, daß dominierte Imputationen keine stabilen Lösungen darstellen. Daher bietet es sich an, die Menge der nicht dominierten Imputationen zu untersuchen.

Definition 7.5.10: Die Menge aller nicht dominierten Imputationen eines Spiels v heißt Kern des Spiels.

Satz 7.5.11: *Der Kern des Spiels v enthält genau alle Auszahlungsvektoren x mit*

$$\begin{aligned} \sum_{i \in S} x_i &\geq v(S) \text{ für alle Koalitionen } S \\ \sum_{i \in N} x_i &= v(N). \end{aligned}$$

Beweis: Die zweite Bedingung ist für alle Imputationen erfüllt. Falls die erste Bedingung nicht erfüllt ist, kann die Koalition S, für die die Bedingung nicht erfüllt ist, ihre Gesamtauszahlung verbessern. Dabei kann sie den zusätzlichen Gewinn gleichmäßig auf ihre Mitglieder verteilen. Die gegebene Imputation wird also bzgl. dieser Koalition dominiert.

Nehmen wir nun an, daß die Bedingungen für x erfüllt sind. Falls $x <_S y$ gilt, folgt

$$\sum_{i \in S} y_i > \sum_{i \in S} x_i \geq v(S)$$

im Widerspruch zur Definition von $<_S$. □

Dieser Satz hat als Konsequenz, daß der Kern eine abgeschlossene und konvexe Menge ist, die durch ein System von linearen Ungleichungen beschrieben ist. Damit ist, wie wir in dieser Vorlesung gelernt haben, der Kern eine gut handhabbare Menge. Verschiedene Imputationen im Kern sind alle gleich stabil, da Änderungen von keiner Koalition erzwungen werden können. Den Kern als Lösungsmenge anzusehen, ist also recht vernünftig. Allerdings ist der Kern oft leer.

Definition 7.5.12: Ein Spiel v heißt Konstant-Summen-Spiel, wenn $v(S) + v(N \Leftrightarrow S) = v(N)$ für alle Koalitionen S gilt.

Satz 7.5.13: *Für wesentliche Konstant-Summen-Spiele ist der Kern leer.*

Beweis: Sei angenommen, daß x im Kern liegt. Für jedes i gilt dann

$$\sum_{j \neq i} x_j \geq v(N \Leftrightarrow \{i\}) = v(N) \Leftrightarrow v(\{i\}).$$

Die letzte Gleichung folgt aus der Konstant-Summen-Bedingung. Da $x_i \geq v(\{i\})$ für jede Imputation x gilt, folgt

$$\sum_{j \in N} x_j \geq v(N).$$

Andererseits kann hier nur Gleichheit gelten. Daraus folgt $x_i = v(\{i\})$ für alle i. Dann gilt aber, da das Spiel wesentlich ist,

$$\sum_{j \in N} x_j = \sum_{j \in N} v(\{j\}) < v(N).$$

Mit diesem Widerspruch haben wir gezeigt, daß der Kern leer ist. □

Was ist zu tun, wenn der Kern leer ist? Von Neumann und Morgenstern haben stabile Mengen eingeführt.

Definition 7.5.14: Eine Menge $V \subseteq I(v)$ von Imputationen heißt stabil, wenn die beiden folgenden Bedingungen erfüllt sind:

- keine Imputation $x \in V$ dominiert eine andere Imputation $y \in V$ (innere Stabilität).
- jede Imputation $x \notin V$ wird von einer Imputation $y \in V$ dominiert (äußere Stabilität).

Es gibt auch Spiele ohne stabile Mengen. Diese sind jedoch schwer zu konstruieren. Andererseits haben Spiele teilweise viele stabile Mengen. Es ist dann unklar, welche stabile Menge gewählt wird, und dann, welche Imputation der gewählten stabilen Menge gewählt wird. Wir untersuchen nur ein Beispiel, die auf $(0,1)$ normierte Variante von Beispiel 7.5.1.

Beispiel 7.5.15: $v(S) = 0$, falls $|S| \leq 1$, und $v(S) = 1$, falls $|S| \geq 2$, wobei $n = 3$.

Die Menge $V = \{(1/2, 1/2, 0), (1/2, 0, 1/2), (0, 1/2, 1/2)\}$ ist stabil. Die innere Stabilität ist klar. Sei $x = (x_1, x_2, x_3) \notin V$ eine Imputation, dann kann eine Zweierkoalition x dominieren.

Für $0 \leq c < 1/2$ ist auch

$$V_{3,c} = \{(x_1, 1 \Leftrightarrow c \Leftrightarrow x_1, c) \mid 0 \leq x_1 \leq 1 \Leftrightarrow c\}$$

stabil. Analoges gilt für entsprechende Mengen $V_{1,c}$ und $V_{2,c}$.

Die innere Stabilität von $V_{3,c}$ ist offensichtlich, da sich zwei Imputationen $x, y \in V_{3,c}$ nur darin unterscheiden, wie sich die Spieler 1 und 2 den Rest, nämlich $1 \Leftrightarrow c$, aufteilen. Derartige Imputationen dominieren sich nicht, da Dominanz durch Einerkoalitionen ausgeschlossen ist.

Wir kommen zur äußeren Stabilität. Sei $y \notin V_{3,c}$.

1. Fall: $y_3 > c$. Dann sei $\varepsilon := y_3 \Leftrightarrow c$, $x_1 := y_1 + \varepsilon/2$, $x_2 := y_2 + \varepsilon/2$, $x_3 := c$. Es ist $x \in V_{3,c}$, und x dominiert y bzgl. der Koalition $\{1,2\}$. Beide Partner dieser Koalition bekommen mehr und können dies auch zusammen erzwingen.

2. Fall: $y_3 < c$. Es ist $y_1 \leq 1/2$ oder $y_2 \leq 1/2$, o. B. d. A. $y_1 \leq 1/2$. Sei $x := (1 \Leftrightarrow c, 0, c)$. Dann ist $x \in V_{3,c}$. Nun dominiert x die Imputation y bzgl. der Koalition $\{1,3\}$. Spieler 1 bekommt mehr als zuvor, da $c < 1/2$ vorausgesetzt ist. Spieler 3 bekommt nun c statt vorher y_3. Schließlich kann sich die Koalition $\{1,3\}$ die gemeinsame Auszahlung 1 sichern.

7.6 Der Shapley-Wert

Beim Kern hatten wir das Problem, daß der Kern oft leer ist. Wenn dagegen der Kern groß ist, wissen wir nicht, welche Imputation des Kernes gewählt wird. Ähnlich sieht es bei den stabilen Mengen aus. Shapley ist dagegen einen Weg analog zu dem von Nash gegangen. Er hat Axiome formuliert, aus denen sich ein eindeutiger Auszahlungsvektor für n-Personen-Spiele ableiten läßt. Dieser Vektor soll nicht die wahrscheinlichste oder gerechteste Auszahlung angeben, sondern die Macht, die Stärke oder den Einfluß der Spieler messen. Jeder Spieler kann dann eine Auszahlung oberhalb des Shapley-Wertes als Erfolg feiern.

Da Spiele bei uns charakteristische Funktionen sind, können wir Spiele addieren und mit positiven Faktoren multiplizieren. Wir wollen einfache Spiele sinnvoll lösen und dann zeigen, daß jedes Spiel so aus einfachen Spielen zusammengesetzt ist, daß jede Zerlegung zum gleichen „Wert“ für die einzelnen Spieler führt.

Um die Shapley-Axiome beschreiben zu können, benötigen wir noch zwei Definitionen.

Definition 7.6.1: Eine Koalition T eines Spiels heißt Träger des Spiels, wenn $v(S) = v(S \cap T)$ für alle Koalitionen S gilt, d. h. Spieler außerhalb des Trägers tragen zu keiner Koalition etwas bei.

Definition 7.6.2: Sei v die charakteristische Funktion eines Spiels. Die charakteristische Funktion zu dem Spiel, das aus v nur durch Umnumerierung der Spieler nach der Permutation π entsteht, heißt πv, d. h.

$$\pi v(\{\pi(i_1), \ldots, \pi(i_s)\}) := v(\{i_1, \ldots, i_s\}).$$

Die Shapley–Funktion φ soll jeder charakteristischen Funktion v einen Vektor $\varphi(v) = (\varphi_1(v), \ldots, \varphi_n(v))$ zuordnen, wobei $\varphi_i(v)$ der Shapley-Wert für den Spieler i im Spiel v ist. Die Shapley–Funktion soll die folgenden drei Axiome erfüllen.

Axiom 1: Ist T ein Träger des Spiels v, soll die Summe aller $\varphi_i(v)$, $i \in T$, genau $v(T)$ $(= v(N))$ betragen. Spieler, die zu keiner Koalition etwas beitragen, sind machtlos und erhalten nichts.

Axiom 2: Für alle Permutationen π und Spiele v gilt $\varphi_{\pi(i)}(\pi v) = \varphi_i(v)$. Kein Spieler soll nur aufgrund seiner Nummer etwas erhalten.

Axiom 3: Die Shapley–Funktion soll linear auf Summen wirken, d.h. $\varphi(u+v) = \varphi(u) + \varphi(v)$. Auch dieses Axiom ist plausibel, es hat aber viele Konsequenzen, da sich Spiele auf sehr viele Weisen in Summanden zerlegen lassen.

Lemma 7.6.3: *Das Spiel v_T sei definiert durch $v_T(S) := 0$, falls $S \not\supseteq T$, und $v_T(S) := 1$, falls $S \supseteq T$. Dann ist für $t := |T|$*

$$\begin{aligned} \varphi_i(v_T) &= 1/t \quad \text{, falls } i \in T, \text{ und} \\ \varphi_i(v_T) &= 0 \quad \text{, falls } i \notin T. \end{aligned}$$

Beweis: Aus der Definition des Spiels folgt, daß T Träger des Spiels ist. Da die Spieler des Trägers alles bekommen und sich Einzelspieler 0 sichern können, ist $\varphi_i(v_T) = 0$ für $i \notin T$. Aus Axiom 2 folgt für Permutationen π, die T auf sich abbilden, daß $\pi v_T = v_T$ und $\varphi_i(v_T) = \varphi_{\pi(i)}(\pi v_T) = \varphi_{\pi(i)}(v_T)$ ist. Also erhalten im Shapley-Wert alle Spieler aus T das gleiche. Nach Axiom 1 erhalten sie zusammen 1. Also ist $\varphi_i(v_T) = 1/t$ für $i \in T$. □

Wir können nun zeigen, daß die Spiele v_T, $\emptyset \subsetneq T \subseteq N$, eine Basis für alle Spiele bilden.

Lemma 7.6.4: *Für jedes Spiel v existieren $2^n \Leftrightarrow 1$ reelle Zahlen c_T mit $T \subseteq N$ und $T \neq \emptyset$, so daß v die Summe der Spiele $c_T v_T$ ist (wobei einige der c_T negativ sein können).*

Beweis: Wir setzen im folgenden stets $t = |T|$ und $s = |S|$ und definieren $c_T := \sum_{S \subseteq T} (\Leftrightarrow 1)^{t-s} v(S)$.
Sei nun $U \subseteq N$ eine Koalition. Wir wollen zeigen, daß

$$v(U) = \sum_{T \subseteq N} c_T v_T(U)$$

ist. Dazu benutzen wir die Definition von v_T. Es gilt

$$\begin{aligned} \sum_{T \subseteq N} c_T v_T(U) &= \sum_{T \subseteq U} c_T \\ &= \sum_{T \subseteq U} \sum_{S \subseteq T} (\Leftrightarrow 1)^{t-s} v(S) \\ &= \sum_{S \subseteq U} \left(\sum_{T \subseteq U,\, S \subseteq T} (\Leftrightarrow 1)^{t-s} \right) v(S). \end{aligned}$$

Wir betrachten nun die innere Summe. Es gilt stets $S \subseteq T \subseteq U$. Für jeden Wert t mit $s \leq t \leq u = |U|$ gibt es genau $\binom{u-s}{t-s}$ Mengen T mit t Elementen und $S \subseteq T \subseteq U$. Also ergibt die innere Summe nach dem Binomischen Lehrsatz

$$\sum_{s \leq t \leq u} \binom{u \Leftrightarrow s}{t \Leftrightarrow s} (\Leftrightarrow 1)^{t-s} = \sum_{0 \leq t \leq u-s} \binom{u \Leftrightarrow s}{t} (\Leftrightarrow 1)^t 1^{u-s-t} = (1 \Leftrightarrow 1)^{u-s}.$$

Dieser Wert ist 0 für $s < u$ und 1 für $s = u$. Für $s = u$ ist $S = U$, und der äußere Faktor ist $v(U)$. Also ergibt die Summe wie gewünscht $v(U)$. □

Satz 7.6.5: *Es gibt genau eine Funktion, die die drei Shapley-Axiome erfüllt. Für diese Funktion gilt*

$$\varphi_i(v) = \sum_{T \subseteq N,\, i \in T} \frac{(t \Leftrightarrow 1)!(n \Leftrightarrow t)!}{n!}(v(T) \Leftrightarrow v(T \Leftrightarrow \{i\})).$$

Beweis: Wir leiten die Shapley–Funktion zunächst aus den Axiomen ab. Wir benutzen die Zerlegung des Spiels v in die Summanden $c_T v_T$ aus Lemma 7.6.4. Wir haben nun das Problem, daß manche Koeffizienten c_T negativ sein können. Aus dem Summenaxiom folgt allgemein für Spiele $u \Leftrightarrow v$ und v, daß $\varphi(u \Leftrightarrow v) + \varphi(v) = \varphi(u)$ ist. Dann ist aber auch $\varphi(u \Leftrightarrow v) = \varphi(u) \Leftrightarrow \varphi(v)$. Also können wir das Summenaxiom auf die Summe aller $c_T v_T$ anwenden. Es folgt mit der Lösung der Spiele $c_T v_T$ aus Lemma 7.6.3

$$\varphi_i(v) = \sum_{T \subseteq N} \varphi_i(c_T v_T) = \sum_{T \subseteq N,\, i \in T} c_T / t.$$

Nun können wir die Werte für c_T einsetzen. Also ist

$$\begin{aligned} \varphi_i(v) &= \sum_{T \subseteq N,\, i \in T} \frac{1}{t} \sum_{S \subseteq T} (\Leftrightarrow 1)^{t-s} v(S) \\ &= \sum_{S \subseteq N} \sum_{T \subseteq N,\, S \cup \{i\} \subseteq T} (\Leftrightarrow 1)^{t-s} \frac{1}{t} v(S) \end{aligned}$$

Sei nun

$$\gamma_i(S) = \sum_{T \subseteq N,\, S \cup \{i\} \subseteq T} (\Leftrightarrow 1)^{t-s} \frac{1}{t}.$$

Sei $S = S' \cup \{i\}$, $i \notin S'$. Wir wollen $\gamma_i(S)$ und $\gamma_i(S')$ vergleichen. Es ergeben sich in der Definition die gleichen Summanden, nur ist $s' = s \Leftrightarrow 1$. Also ist $\gamma_i(S) = \Leftrightarrow \gamma_i(S')$. Also ist

$$\varphi_i(v) = \sum_{S \subseteq N,\, i \in S} \gamma_i(S)(v(S) \Leftrightarrow v(S \Leftrightarrow \{i\})).$$

Um die behauptete Form von φ zu erhalten, genügt es,

$$\gamma_i(S) = \frac{(s \Leftrightarrow 1)!(n \Leftrightarrow s)!}{n!} \text{ für } i \in S$$

zu beweisen. Dazu gehen wir einen überraschenden Umweg über Integrale. Sei $i \in S$. Dann gibt es genau $\binom{n-s}{t-s}$ Koalitionen T mit t Elementen und $S \subseteq T$. Also gilt

$$\gamma_i(S) = \sum_{s \le t \le n} (\Leftrightarrow 1)^{t-s} \binom{n \Leftrightarrow s}{t \Leftrightarrow s} \frac{1}{t}.$$

Wir schreiben nun $1/t$ als Integral. Die nächste Gleichheit läßt sich rückwärts trivial verifizieren. Es ist

$$\begin{aligned} \gamma_i(S) &= \sum_{s \le t \le n} (\Leftrightarrow 1)^{t-s} \binom{n \Leftrightarrow s}{t \Leftrightarrow s} \int_0^1 x^{t-1} dx \\ &= \int_0^1 \sum_{s \le t \le n} (\Leftrightarrow 1)^{t-s} \binom{n \Leftrightarrow s}{t \Leftrightarrow s} x^{t-1} dx \\ &= \int_0^1 x^{s-1} \sum_{s \le t \le n} (\Leftrightarrow 1)^{t-s} \binom{n \Leftrightarrow s}{t \Leftrightarrow s} x^{t-s} dx \end{aligned}$$

$$= \int_0^1 x^{s-1} \sum_{0 \le t \le n-s} (\Leftrightarrow x)^t 1^{n-s-t} \binom{n \Leftrightarrow s}{t} dx$$

$$= \int_0^1 x^{s-1} (1 \Leftrightarrow x)^{n-s} dx.$$

Die letzte Gleichung folgt wieder aus dem Binomischen Lehrsatz. Wir haben nun $\gamma_i(S)$ in eine Form gebracht, die in der Analysis in allgemeiner Form studiert wurde. Wir erhalten nämlich ein Integral, das die Betafunktion definiert. Aus der Analysis folgt nun direkt

$$\gamma_i(S) = \frac{(s \Leftrightarrow 1)!(n \Leftrightarrow s)!}{n!}.$$

Wir haben noch zu zeigen, daß die abgeleitete Funktion die Shapley-Axiome erfüllt.

Dazu diskutieren wir folgendes Experiment. Wir betrachten alle $n!$ Permutationen der Spieler. Für jede Permutation lassen wir die Spieler in der zugehörigen Reihenfolge an die Kasse gehen. Wenn vor Spieler i die Spielermenge S' an der Kasse war, erhält Spieler i für $S := S' \cup \{i\}$ die Auszahlung $v(S) \Leftrightarrow v(S \Leftrightarrow \{i\})$. Insgesamt wird für jede Permutation $v(N)$ ausgezahlt, nämlich bei der Reihenfolge $i_1, \ldots, i_n$ die Werte $v(\{i_1\}) \Leftrightarrow v(\emptyset), v(\{i_1, i_2\}) \Leftrightarrow v(\{i_1\}), \ldots, v(N) \Leftrightarrow v(N \Leftrightarrow \{i_n\})$. Jeder Spieler erhält, da für $i \in S$

$$v(S) \ge v(\{i\}) + v(S \Leftrightarrow \{i\})$$

ist, stets mindestens $v(\{i\})$. Also ist die Auszahlung stets eine Imputation. Wenn wir jedem Spieler den Durchschnittswert über alle Permutationen geben, bleibt diese Eigenschaft erhalten. Was erhält Spieler i dann? Es gibt $(s \Leftrightarrow 1)!(n \Leftrightarrow s)!$ Permutationen, in denen vor ihm genau die Spielermenge S' an der Kasse war. Also erhält Spieler i bei diesem Experiment im Durchschnitt genau den Shapley-Wert! Spieler, die nicht im Träger sind, tragen zu keiner Koalition etwas bei und bekommen nie etwas. Also bekommen die Spieler aus dem Träger alles, d. h. Axiom 1 gilt. Eine Umnumerierung der Spieler verändert das Experiment nicht, d. h. Axiom 2 gilt. Auch Axiom 3 folgt sehr einfach. Wenn wir das Experiment nacheinander für u und für v durchführen, kommt das gleiche heraus, als wenn wir das Experiment für $u + v$ durchführen. □

Mit dem zweiten Teil des Beweises haben wir den Shapley-Wert auch gut veranschaulicht. Wir beschließen dieses Kapitel mit zwei Beispielen zum Shapley-Wert.

Beispiel 7.6.6: In einem Parlament sitzen vier Parteien mit 10, 20, 30 und 40 Sitzen (analog Aktienbesitzer einer AG und ihre Anteile). Wenn „wie in modernen Demokratien, jeder mit jedem koalitionsfähig ist", wollen wir mit dem Shapley-Wert die Macht messen, die die Parteien aus ihren Mandaten ziehen können. Es sei $v(S) \in \{0, 1\}$ und $v(S) = 1$ für alle Koalitionen, die mehr als 50% der Mandate haben. Dann gibt es die folgenden Gewinnkoalitionen (nur sie tragen zum Shapley-Wert bei): $\{2, 4\}$, $\{3, 4\}$, $\{1, 2, 3\}$, $\{1, 2, 4\}$, $\{1, 3, 4\}$, $\{2, 3, 4\}$ und $\{1, 2, 3, 4\}$.

Für Spieler 1 ist $\{1, 2, 3\}$ die einzige Koalition, die ohne ihn nicht gewinnt. Also ist

$$\varphi_1(v) = \frac{2!\,1!}{4!} = \frac{1}{12} = 0.0833\ldots$$

Für Spieler 2 sind die Koalitionen $\{2, 4\}$, $\{1, 2, 3\}$ und $\{1, 2, 4\}$ zu betrachten. Alle erhalten den Faktor $2/4!$, also ist $\varphi_2(v) = 3 \cdot \frac{1}{12} = 0.25$.

Für Spieler 3 sind die Koalitionen $\{3, 4\}$, $\{1, 2, 3\}$ und $\{2, 3, 4\}$ zu betrachten. Damit ist auch $\varphi_3(v) = 0.25$.

Für Spieler 4 müssen die Koalitionen $\{2, 4\}$, $\{3, 4\}$, $\{1, 2, 4\}$, $\{1, 3, 4\}$ und $\{2, 3, 4\}$ betrachtet werden. Damit ist $\varphi_4(v) = 5/12 = 0.416\cdots$.

Also haben die Parteien 2 und 4 mehr Macht als Mandate, während das für die Parteien 1 und 3 genau umgekehrt ist.

Beispiel 7.6.7: Wir betrachten wieder ein Parlament, nun sei die Mandatsverteilung 10, 30, 30 und 40. Gewinnkoalitionen sind $\{2,3\}$, $\{2,4\}$, $\{3,4\}$, $\{1,2,3\}$, $\{1,2,4\}$, $\{1,3,4\}$ und $\{1,2,3,4\}$. Es zeigt sich, daß $\{2,3,4\}$ Träger des Spiels ist und daß die vierte Partei von ihren 10 zusätzlichen Mandaten nichts hat. Es ist $\varphi_1(v) = 0$, $\varphi_2(v) = \varphi_3(v) = \varphi_4(v) = 1/3$.

8 Dynamische und Stochastische Optimierung

8.1 Das Problem und einleitende Bemerkungen

Sowohl bei der Linearen Optimierung als auch bei der Ganzzahligen Optimierung werden alle Daten als bekannt vorausgesetzt, und eine bekannte Funktion muß optimiert werden, um eine gegenwärtige Aktion zu planen. Obwohl diese Problemstellungen in vielen Praxissituationen angemessen sind, gibt es doch viel allgemeinere Problemstellungen. Die gegenwärtigen Daten sind bekannt, dann haben wir eine Aktion durchzuführen, gleichzeitig handeln aber auch andere Beteiligte, die Daten einen Zeittakt später hängen nicht nur von unserer Aktion, sondern von allen Aktionen ab, in Abhängigkeit von diesen Daten müssen wir wieder eine Aktion durchführen, usw. Hinzu kommt, daß die Daten zu keinem Zeitpunkt sicher bekannt sind. Wir kennen, aber auch nur vielleicht, unsere eigene Situation genau, der Erfolg unserer Aktionen hängt auch von der Ausgangslage anderer Beteiligter ab. Diese Daten können nur geschätzt werden. Wir modellieren unser Wissen über die Daten durch ein stochastisches Modell, d. h. wir nehmen an, daß wir die Wahrscheinlichkeiten kennen, mit denen bestimmte Parameter gewisse Werte annehmen.

Was ist nun unser Ziel? Wir wissen, daß es im allgemeinen fatal ist, den gegenwärtigen Gewinn zu maximieren und nicht in die Zukunft zu planen. Im Leben von Studierenden führt diese Greedy Strategie dazu, nie Übungsaufgaben zu lösen. Statt dessen suchen wir einen Handlungsplan, eine Strategie, die unser Verhalten in allen gegenwärtigen und zukünftigen Situationen plant und dabei den erwarteten Gewinn maximiert. In einem stochastischen Modell kann natürlich nur der erwartete Gewinn und nicht der Gewinn maximiert werden. Dieses Modell ist auch dann sinnvoll, wenn wir eventuell später zusätzliches gesichertes Wissen erhalten. In einem derartigen Fall muß die Zukunft neu geplant werden. Das gegenwärtige Verhalten kann natürlich nur auf dem Hintergrund des gegenwärtigen Wissens optimiert werden. Dabei stellt auch „stochastisches Wissen", d. h. die Kenntnis von Wahrscheinlichkeiten, mit denen Ereignisse eintreten, Wissen dar.

Wir stellen zunächst allgemeine Resultate für Probleme der Dynamischen und Stochastischen Optimierung vor. Dieses Wissen ist notwendig für die Lösung konkreter Probleme, reicht dafür aber nicht aus. Für konkrete Probleme müssen die allgemeinen Resultate auf die spezielle Situation angewendet werden, was jedes Mal aufwendig sein kann.

Im folgenden diskutieren wir das Grundmodell der Dynamischen und Stochastischen Optimierung.

- S ist eine nicht leere, höchstens abzählbar unendliche Menge (state space), die die möglichen „Zustände" enthält. Der Zustand einer Firma kann sich zusammensetzen aus dem Kontostand, der Auftragslage, der Qualifikation der Beschäftigten, dem Zustand des Maschinenparks, den Beziehungen zu Kunden, Verwaltung und Politikern, ... Auch hier sehen wir die großen Freiheiten bei der Modellbildung. Die einzigen Einschränkungen bestehen darin, daß wir einen diskreten Zustandsraum (höchstens abzählbar unendlich, nicht überabzählbar unendlich wie $\mathbb{R}$) annehmen (dies ist keine wesentliche Einschränkung) und daß wir annehmen, den momentanen Zustand erfassen (oder messen) zu können.

- A ist die Menge der durchführbaren Aktionen (Produktionsentscheidungen, Kauf- oder Verkaufsentscheidungen, Investitionsentscheidungen, Entscheidungen über Einstellungen oder sogenannte Freistellungen, ...).

- $\bar{H}_n$ ist die Menge der Vektoren, die die Vergangenheit (history) zum Zeitpunkt n beschreiben, d. h. für $h_n \in \bar{H}_n$ ist
$$h_n = (s_1, a_1, \ldots, a_{n-1}, s_n) \in S \times A \times S \times \ldots \times A \times S$$
ein Vektor, der die Zustände $s_1, \ldots, s_n$ zu den Zeitpunkten $1, \ldots, n$ und die Aktionen $a_1, \ldots, a_{n-1}$ zu den Zeitpunkten $1, \ldots, n \Leftrightarrow 1$ beschreibt.

  Nicht bei jeder Vergangenheit ist jede Aktion tatsächlich durchführbar. Bei negativem Kontostand erhält man nicht beliebig viel Kredit, auch nicht, wenn man in der Vergangenheit als Kreditschwindler aufgefallen ist.

- $D_n(h) \subseteq A$ beschreibt für $h \in \bar{H}_n$ die Menge der bei dieser Vergangenheit erlaubten Aktionen.
- H_n soll die Vektoren $h \in \bar{H}_n$ enthalten, die tatsächlich möglich sind. Dabei ist $H_1 = \bar{H}_1 = S$, und H_n enthält alle $h_n = (s_1, a_1, \ldots, a_{n-1}, s_n) \in \bar{H}_n$, für die $a_i \in D_i(s_1, a_1, \ldots, s_i)$ für alle i ist. Offensichtlich genügt es, wenn $D_n(h)$ für $h \in H_n$ definiert ist.
- p_0 beschreibt die a priori Wahrscheinlichkeitsverteilung für den Anfangszustand. Daher ist $p_0 : S \to [0,1]$ und $\sum_{s \in S} p_0(s) = 1$.
- p_n beschreibt die Übergangswahrscheinlichkeiten vom Zeitpunkt n zum Zeitpunkt $n+1$. Es soll $p_n(h,a,s)$ für $h \in H_n$, $a \in D_n(h)$ und $s \in S$ die Wahrscheinlichkeit sein, bei Vergangenheit h und gewählter Aktion a in den Zustand s zu gelangen. Daher ist $p_n(h,a,\cdot) : S \to [0,1]$ und $\sum_{s \in S} p_n(h,a,s) = 1$.
- r_n beschreibt die Auszahlung (Belohnung, reward) zum Zeitpunkt n. Für $h \in H_n$ ist $r_n(h,\cdot)$ auf $D_n(h)$ definiert und nimmt Werte in $\mathbb{R}$ an. Negative Belohnungen sind Kosten. Also ist $r_n(h,a)$ die Auszahlung zum Zeitpunkt n bei Historie h und gewählter Aktion a.

Was ist nun ein Handlungsplan? Eine deterministische Strategie f gibt für jede Historie h an, welche Aktion gewählt werden soll. Es soll $f = (f_n)$ mit $f_n : S^n \to A$ sein. Zu $(s_1, \ldots, s_n)$ gehört bei Verwendung von Strategie f sicher die Historie

$$h_n = (s_1, f_1(s_1), s_2, f_2(s_1, s_2), \ldots, f_{n-1}(s_1, \ldots, s_{n-1}), s_n) .$$

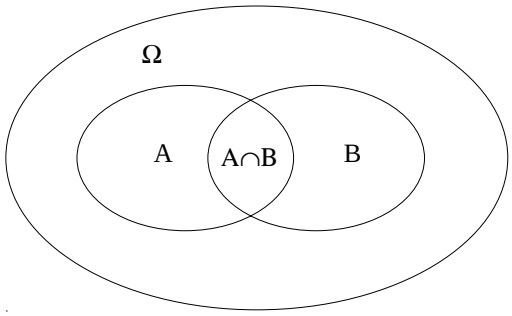
Natürlich verlangen wir, daß $f(h_n) \in D_n(h_n)$ ist. Wir beschränken uns zunächst auf deterministische Strategien. Randomisierte Strategien erlauben, daß wir statt einer Aktion stets eine Wahrscheinlichkeitsverteilung auf der Menge der erlaubten Aktionen wählen. Wir zeigen später, daß diese größere Strategienklasse keine besseren Strategien enthält. Dies mag selbstverständlich klingen, gilt aber nur, weil wir den Rest der Welt stochastisch und nicht reagierend modelliert haben. Wenn ein Unternehmen an einem Markt nur einen kleinen Anteil hat, ist dieses Modell angemessen. Wenn es jedoch wie in einem Oligopol durch seine Aktionen die Bedingungen für die anderen Oligopolisten verändert, reagieren diese, um die Aktionen zu beeinflussen. Diese Situation wird in der Spieltheorie (Kapitel 7) modelliert. Dort haben sich randomisierte Strategien in vielen Fällen als den deterministischen Strategien überlegen erweisen.

Wenn die Strategie f gewählt ist, ergibt sich in unserem Modell folgender stochastischer Prozeß. Gemäß p_0 wird der Startzustand s_1 (Zeitpunkt 1) ausgewählt. Wir reagieren mit Aktion $a_1 = f(s_1)$ und erhalten die Auszahlung $r_1(s_1, a_1)$. Der Nachfolgezustand ergibt sich gemäß der Verteilung $p_1(s_1, a_1, \cdot)$. Wenn der Nachfolgezustand s_2 ist, wählen wir die Aktion $a_2 = f(s_1, s_2)$ und erhalten die Auszahlung $r_2((s_1, a_1, s_2), a_2)$. Der Nachfolgezustand s_3 ergibt sich gemäß der Verteilung $p_2((s_1, a_1, s_2), a_2, \cdot)$, usw.

Wir müssen in Zukunft viel mit bedingten Wahrscheinlichkeiten rechnen. Wir streuen dann die Rechenregeln der Wahrscheinlichkeitstheorie, die eigentlich bekannt sein sollten, wegen der eventuellen Vergeßlichkeit des einen oder der anderen zu gegebener Zeit ein. Es seien P ein Wahrscheinlichkeitsmaß und A und B Ereignisse mit $P(B) > 0$. Dann ist die bedingte Wahrscheinlichkeit von A unter der Bedingung, daß B eingetreten ist, definiert durch

$$P(A|B) := P(A \cap B)/P(B).$$

Die Gleichung $P(A \cap B) = P(A|B)P(B)$ gilt trivialerweise auch für $P(B) = 0$. Diese Definition läßt sich folgendermaßen veranschaulichen, wobei Ω der gesamte Wahrscheinlichkeitsraum ist.



Wahrscheinlichkeiten sind Maße und können in der Abbildung durch Flächenmaße symbolisiert werden. Der gesamte Raum Ω hat Maß 1. $P(A)$ ist der Anteil der Fläche von A an Ω. Was ändert sich, wenn wir die gesicherte Erkenntnis gewinnen, daß das Ereignis B eingetreten ist? Alle Masse konzentriert sich auf B. Der Anteil von A, der außerhalb von B liegt, ist unmöglich geworden. Also ist $P(A|B)$ der Anteil der Fläche von $A \cap B$ an der Fläche von B. Sei $F(X)$ die „Fläche“ von X. Dann ist $P(A \cap B) = F(A \cap B)/F(\Omega)$ und $P(B) = F(B)/F(\Omega)$. Also ist $F(A \cap B)/F(B) = P(A \cap B)/P(B)$ und $P(A|B)$ „vernünftig“ definiert.

Wir wollen die Wahrscheinlichkeit $P_f(s_1, \ldots, s_n)$ berechnen, daß die Zustandsfolge $s_1, \ldots, s_n$ unter der Strategie f entsteht. Das Auftreten der Zustandsfolge ist der Durchschnitt der Ereignisse, daß s_i der i-te Zustand ist. Also folgt

$$\begin{aligned} P_f(s_1, \ldots, s_n) &= P_f(s_2, \ldots, s_n \mid s_1) P_f(s_1) \\ &= P_f(s_3, \ldots, s_n \mid s_1, s_2) P_f(s_2 \mid s_1) P_f(s_1) \\ &= \prod_{1 \le i \le n} P_f(s_i \mid s_1, \ldots, s_{i-1}) \\ &= \prod_{1 \le i \le n} p_{i-1}((s_1, f(s_1), \ldots, s_{i-1}), f(s_1, \ldots, s_{i-1}), s_i). \end{aligned}$$

Für $i = 1$ ergibt sich als Faktor $p_0(s_1)$. Auch an der Abbildung wird deutlich, daß für disjunkte Ereignisse, d. h. $A \cap B = \emptyset$, $P(A \cup B) = P(A) + P(B)$ gilt. Wir wollen die Wahrscheinlichkeit $P_f(s_n)$ berechnen, daß unter Strategie f der n-te Zustand s_n ist. Für verschiedene $(s_1, \ldots, s_{n-1}) \in S^{n-1}$ sind die Ereignisse $(s_1, \ldots, s_{n-1}, s_n)$ disjunkt. Also ist

$$P_f(s_n) = \sum_{(s_1, \ldots, s_{n-1}) \in S^{n-1}} P_f(s_1, \ldots, s_n).$$

Im konkreten Fall müssen wir uns natürlich überlegen, wie wir konkrete Wahrscheinlichkeiten analytisch berechnen. Falls $|S| = \infty$, haben wir es mit unendlichen Reihen zu tun. Falls $|S| = k$, hat die Formel für $P_f(s_n)$ genau k^{n-1} Summanden, die selber Produkte aus n Faktoren sind. Theoretisch erhalten wir zwar manchmal lange, aber stets gut zu manipulierende Formeln.

Jetzt können wir unser Ziel formalisieren. Die Auszahlung zum Zeitpunkt n ist unter Strategie f, falls die Zustandsfolge $(s_1, \ldots, s_n)$ eingetreten ist, $r_n((s_1, f(s_1), \ldots, s_n), f(s_1, \ldots, s_n))$ oder abgekürzt $r_{n,f}(s_1, \ldots, s_n)$. Die erwartete Auszahlung zum Zeitpunkt n beträgt

$$R_{n,f} = \sum_{(s_1, \ldots, s_n) \in S^n} P_f(s_1, \ldots, s_n) r_{n,f}(s_1, \ldots, s_n).$$

Die erwartete Auszahlung unter Strategie f ist also

$$G_f := R_f = \sum_{1 \le n < \infty} R_{n,f}.$$

Es sei

$$G := \sup\{G_f \mid f \text{ zulässige Strategie}\}.$$

Eine Strategie heißt genau dann optimal, wenn $G_f = G$ ist, und ε-optimal, wenn $G_f \geq G \Leftrightarrow \varepsilon$ ist. Unser Ziel ist also die Berechnung einer optimalen Strategie oder ersatzweise einer ε-optimalen Strategie. Optimale Strategien müssen natürlich nicht existieren (Bsp.: Auszahlung nur zum Zeitpunkt 1 von 0 verschieden, Aktion a ($\in \mathbb{N}$) ergibt Auszahlung $1 \Leftrightarrow 1/a$), da G als Supremum nicht angenommen werden muß. Dann müssen wir uns mit ε-optimalen Strategien zufriedengeben. Selbst wenn optimale Strategien existieren, ist die Berechnung ε-optimaler Strategien teilweise vorzuziehen, da diese sich effizienter berechnen lassen (Abschätzung unendlicher Reihen, usw.).

Eigentlich sind wir zu schnell vorgeprescht, da die Reihen, die $R_{n,f}$ oder G_f definieren, gar nicht konvergieren müssen. In einem Modell, wo wir nur eine Strategie f haben und zu ungeraden Zeitpunkten 1 DM kassieren und zu geraden Zeitpunkten 1 DM bezahlen müssen, ist G_f undefiniert. Es bleibt uns also nichts übrig, als die Konvergenz zu erzwingen. Für Funktionen $g : \mathrm{Def}(g) \to \mathbb{R}$ sei

$$\begin{aligned} \|g\| &= \sup\{|g(x)| \mid x \in \mathrm{Def}(g)\} \\ \|g^+\| &= \sup\{\max\{0, g(x)\} \mid x \in \mathrm{Def}(g)\} \\ \|g^-\| &= \sup\{\Leftrightarrow\min\{0, g(x)\} \mid x \in \mathrm{Def}(g)\}. \end{aligned}$$

Wir machen die generelle Annahme, daß stets eine der folgenden Voraussetzungen erfüllt ist.

(EN) essentially negative case: $\sum_{1 \leq n < \infty} \|r_n^+\| < \infty$, die Auszahlungen sind im wesentlichen negativ.

(EP) essentially positive case: $\sum_{1 \leq n < \infty} \|r_n^-\| < \infty$, die Auszahlungen sind im wesentlichen positiv.

(C) convergent case : $\sum_{1 \leq n < \infty} \|r_n\| < \infty$, d. h. (EN) und (EP). Die erwarteten Auszahlungen konvergieren nach dem Majoritätsprinzip für jede Strategie.

Auch in den Fällen (EN) und (EP) konvergieren die erwarteten Auszahlungen, wenn wir die Konvergenzpunkte $\Leftrightarrow\infty$ bzw. $+\infty$ zulassen.

Im allgemeinen sind die Funktionen r_n durch eine gemeinsame Konstante beschränkt. Da unsere Schätzungen für die Zukunft kaum fehlerfrei sind, ist es oft sinnvoll, die Auszahlungen zum Zeitpunkt n mit einem Diskontfaktor α^n für ein $\alpha < 1$ zu versehen. Dann sind wir automatisch im Fall (C).

Wir beschließen das einführende Kapitel mit einigen Bemerkungen.

1.) Die Fälle (EN) und (EP) sind nicht dual zueinander, da wir in beiden Fällen die erwartete Auszahlung maximieren wollen. Die Ergebnisse für die Fälle (EN) und (EP) werden sich stark unterscheiden.
2.) Die Bedingungen an die Fälle (EN), (EP) und (C) können abgeschwächt werden. Es genügt natürlich, daß die zugehörigen Reihen für alle Strategien konvergieren. In Stopproblemen entstehen beispielsweise solange Kosten, bis wir die Stopaktion wählen und eine Auszahlung erhalten. Danach entstehen weder Kosten noch Auszahlungen. Wenn $||r_n|| \leq c$ für eine Konstante c und alle n, konvergieren die betrachteten Reihen.
3.) Ist es nicht allgemeiner, wenn die Auszahlungen auch noch vom erreichten Zustand abhängen dürfen? Nein, denn dies ist in unserem Modell implizit enthalten, da wir die Auszahlungen um einen Zeitpunkt verschieben können. Dann ist $r_1 \equiv 0$, und $r_n(h, a)$ hängt nicht von a ab.
4.) Der Spezialfall $r_n \equiv 0$ für $n > N$ führt zu Problemen mit endlichem Horizont.
5.) Der Spezialfall, daß $p_n(h, a, s) = 1$ für ein $s = s(h, a)$ ist, führt zu Problemen der deterministischen Dynamischen Programmierung.

Probleme, die wir in Vorlesungen wie Datenstrukturen, GTI oder Effiziente Algorithmen mit Dynamischer Programmierung lösen, sind Fälle von deterministischen Problemen mit endlichem Horizont. Insgesamt haben wir ein sehr allgemeines Modell, unter dem sich eine Vielzahl sehr verschiedenartiger wichtiger Probleme wiederfindet.

8.2 Das Bellman'sche Optimalitätsprinzip

Bellman(1957): „An optimal policy has the property that whatever the initial state and initial decisions are, the remaining decisions must constitute an optimal policy with regard to the state resulting from

the first decision".

Die zitierte Aussage von Bellman scheint trivial zu sein. Sie ist es aber nicht, wie folgendes Beispiel zeigt.

Beispiel 8.2.1: $S = \{1,2\}$, $A = \{W,S\}$ ($W \stackrel{\wedge}{=}$ weiter, $S \stackrel{\wedge}{=}$ stop}, $p_0(1) = 1$, im Zustand 1 bleiben wir, solange wir Aktion W wählen, bei Aktion S wechseln wir in Zustand 2, wo wir stets bleiben. Im Zustand 2 sind die Auszahlungen 0, in Zustand 1 bei Aktion W ebenfalls, nur in Zustand 1 gibt es bei Aktion S die Auszahlung $1 \Leftrightarrow 1/n$, wobei n der zugehörige Zeitpunkt ist. Es ist niemals optimal, die Aktion S zu wählen, da es besser ist, die Aktion S einen Zeitpunkt später zu wählen. Zu jedem Zeitpunkt ist es also optimal, die Aktion W zu wählen, was zu der schlechtestmöglichen Auszahlung 0 führt.

Der Haken an diesem für viele „Qual der Wahl"-Situationen des alltäglichen Lebens typischen Problem ist die Nichtexistenz optimaler Strategien. Bellman setzt aber klugerweise die Existenz einer optimalen Strategie voraus. Wir werden sein Prinzip formalisieren und dann verifizieren.

Definition 8.2.2: Für $y = (s_1, \ldots, s_n) \in S^n$ und eine deterministische Strategie f sei $G_{nf}(y)$ die erwartete Auszahlung ab Zeitpunkt n unter der Bedingung, daß die ersten n Zustände $s_1, \ldots, s_n$ sind. Damit ist $G_{nf}(y)$ nur wohldefiniert, falls $P_f(y) > 0$. Falls $P_f(y) = 0$, sei $G_{nf}(y) := 0$.

Wir geben eine Formel für $G_{nf}(y)$ an, wobei wir wie auch im folgenden $P(A|B) = 0$ setzen, falls $P(B) = 0$ ist. Die erwartete Auszahlung zum Zeitpunkt $m \geq n$ beträgt

$$R_{mf}(y) := \sum_{s_{n+1},\ldots,s_m} P_f(s_{n+1}, \ldots, s_m|y)\ r_m(s_1, f(s_1), \ldots, f(s_1, \ldots, s_m))$$

Also ist

$$G_{nf}(y) = \sum_{n \leq m < \infty} R_{mf}(y).$$

Definition 8.2.3: Für eine Historie $h = (s_1, a_1, \ldots, s_n)$ sei $\Delta(h)$ die Menge der Strategien f, die die Historie h möglich machen, für die also $f(s_1, \ldots s_k) = a_k$ für $k < n$ gilt. Mit

$$G_n(h) := \sup\{G_{nf}(y)|f \in \Delta(h)\}$$

bezeichnen wir den maximal zu erwartenden Gewinn bei Historie h. Für $y = (s_1, \ldots, s_n)$ bezeichnen wir mit $h_{nf}(y)$ die Historie, die aus y und f bis zum Zeitpunkt n entsteht, d.h. $a_i = f(s_1, \ldots, s_i)$ für $i < n$. Damit sind die Notationen $p_{nf}(y,s)$ und $r_{nf}(y)$ für die Übergangswahrscheinlichkeiten und Auszahlungen zu y und f und dem Zeitpunkt n wohldefiniert.

Mit diesen Notationen gilt trivialerweise

$$G_{nf}(y) \leq G_n(h_{nf}(y)).$$

Wir müssen die Ungleichung nur ausführlich „lesen". Links steht die erwartete Auszahlung ab Zeitpunkt n zur bzgl. y und f gebildeten Historie, wenn Strategie f weiter benutzt wird. Rechts steht die erwartete Auszahlung ab Zeitpunkt n zur bzgl. y und f gebildeten Historie, wenn wir über alle Strategien (also auch f) das Supremum bilden. Auch in Zukunft müssen wir formale Aussagen auf diese Weise lesen. Das folgende Hilfsmittel der Wahrscheinlichkeitstheorie ist trivial zu beweisen, aber dennoch der Schlüssel vieler Umformungen.

Satz 8.2.4 (Satz von der bedingten Wahrscheinlichkeit): *Es sei $B_1, \ldots, B_m$ eine disjunkte Zerlegung des zugrundeliegenden Wahrscheinlichkeitsraumes Ω, Dann gilt*

$$P(A) = \sum_{1 \leq i \leq m} P(A|B_i)\, P(B_i).$$

Beweis: Es gilt

$$\begin{aligned} P(A) &= P(A \cap \Omega) = P(A \cap \bigcup_{1 \le i \le m} B_i) \\ &= \sum_{1 \le i \le m} P(A \cap B_i) \qquad \text{(Disjunktheit der } B_i\text{)} \\ &= \sum_{1 \le i \le m} P(A|B_i) P(B_i) \qquad \text{(Definition } P(A|B_i)\text{)}. \end{aligned}$$

□

Wir zerlegen nun G_{nf} mit Hilfe des Satzes von der bedingten Wahrscheinlichkeit in die erwartete Auszahlung zum Zeitpunkt n und die erwartete Auszahlung ab Zeitpunkt $n+1$.

Lemma 8.2.5: *Sei f eine deterministische Strategie. Dann gilt*

$$\begin{aligned} G_f &= \sum_{s \in S} p_0(s)\ G_{1f}(s) \\ G_{nf}(y) &= r_{nf}(y) + \sum_{s \in S} p_{nf}(y,s)\ G_{n+1,f}(y,s). \end{aligned}$$

Beweis: Die erste Aussage kann als Spezialfall der zweiten Aussage für den Zeitpunkt $n = 0$ aufgefaßt werden. Es gibt dann keine Historie, und die einzige Aktion ist Warten. Außerdem ist $r_0 \equiv 0$, der Zustand s zum Zeitpunkt 1 wird mit Wahrscheinlichkeit $p_0(s)$ erreicht.

Wir zeigen daher nur die zweite Aussage. Es ist

$$G_{nf}(y) = \sum_{n \le m < \infty} R_{mf}(y)$$

für $y = (s_1, \ldots, s_n)$. Da mit f und y die Historie bis zur n-ten Aktion feststeht, ist $R_{nf}(y) = r_{nf}(y)$ die deterministische Auszahlung zum Zeitpunkt n.

Für $m > n$ ist

$$\begin{aligned} R_{mf}(y) &= \sum_{s_{n+1}, \ldots, s_m} P_f(s_{n+1}, \ldots, s_m | y)\ r_{mf}(y, s_{n+1}, \ldots, s_m) \qquad \text{(Definition)} \\ &= \sum_{s_{n+1}} \sum_{s_{n+2}, \ldots, s_m} P_f(s_{n+1}|y) P_f(s_{n+2}, \ldots, s_m | y, s_{n+1})\ r_{mf}(y, s_{n+1}, \ldots, s_m) \quad \text{(bed. W.keit)} \\ &= \sum_s p_{nf}(y,s) \sum_{s_{n+2}, \ldots, s_m} P_f(s_{n+2}, \ldots s_m | (y,s))\ r_{mf}((y,s), s_{n+2}, \ldots, s_m) \\ &= \sum_s p_{nf}(y,s) R_{mf}(y,s). \end{aligned}$$

Also ist

$$\begin{aligned} G_{nf}(y) &= r_{nf}(y) + \sum_{n < m < \infty} \sum_s p_{nf}(y,s)\ R_{mf}(y,s) \\ &= r_{nf}(y) + \sum_s p_{nf}(y,s)\ G_{n+1,f}(y,s). \end{aligned}$$

□

Um einigermaßen überschaubare Formeln zu behalten, benötigen wir Abkürzungen, die wir dann aber auch wie Vokabeln lernen müssen, um uns verständigen zu können.

Definition 8.2.6: Für Funktionen $v : S^{n+1} \to \overline{\mathbb{R}} := \mathbb{R} \cup \{-\infty, +\infty\}$ sei $\Lambda_{nf} v : S^n \to \overline{\mathbb{R}}$ definiert durch

$$\Lambda_{nf} v(y) := r_{nf}(y) + \sum_s p_{nf}(y,s) v(y,s).$$

Da Λ_{nf} auf Funktionen angewendet wird, bezeichnen wir Λ_{nf} als Operator. Wir wenden den Operator nur in Situationen an, in denen die rechte Seite konvergiert. Um mit dem Operator vernünftig arbeiten zu können, müssen wir ihn richtig lesen. Gegeben sind wieder n, y und f. Wir erhalten zum Zeitpunkt n die normale Auszahlung r_{nf} und zum Zeitpunkt $n+1$ in Zustand s die Endauszahlung $v(y,s)$. In dieser Situation beträgt die erwartete Auszahlung $\Lambda_{nf} v(y)$. Wenn $v = G_{n+1,f}$ ist, erhalten wir als Endauszahlung die „richtige" erwartete Auszahlung. Lemma 8.2.5 kann daher verkürzt geschrieben werden.

Lemma 8.2.7: $G_{nf} = \Lambda_{nf} G_{n+1,f}$.

Mit Operatoren können wir nun abstrakt arbeiten. Wir werden häufiger die folgenden beiden trivial aus der Definition folgenden Eigenschaften anwenden:

$$v \le w \Rightarrow \Lambda_{nf} v \le \Lambda_{nf} w$$

$$\Lambda_{nf}(v + c) = \Lambda_{nf} v + c \quad \text{für } c \text{ konstant.}$$

Lemma 8.2.8: $G_{nf}(y) = \lim_{k\to\infty} \Lambda_{nf} \Lambda_{n+1,f} \ldots \Lambda_{n+k,f} 0(y)$.

Auf der linken Seite steht die erwartete Auszahlung ab Zeitpunkt n. Auf der rechten Seite wenden wir Operatoren auf die Nullfunktion an. Nach unserer Interpretation erhalten wir die korrekte Auszahlung für die Zeitpunkte $n, \ldots, n+k$ und dann die Schlußzahlung 0. Für $k \to \infty$ sollte dies die korrekte Auszahlung bei Strategie f sein. Auch im folgenden sollten uns die Aussagen schon klar sein, bevor wir sie beweisen.

Beweis von Lemma 8.2.8: Für alle konvergenten Reihen gilt

$$G_{nf}(y) = \sum_{n \le m < \infty} R_{mf}(y) = \lim_{k\to\infty} \sum_{n \le m \le n+k} R_{mf}(y).$$

Den letzten Ausdruck hinter dem lim-Zeichen bezeichnen wir mit $E_{nk}(y)$. Dann genügt es zu zeigen, daß

$$E_{nk}(y) = \Lambda_{nf} \ldots \Lambda_{n+k,f} 0(y)$$

ist. Dies zeigen wir durch Induktion über k. Für $k = 0$ ist

$$E_{n0}(y) = R_{nf}(y) = r_{nf}(y) = \Lambda_{nf} 0(y).$$

Im Induktionsschritt wenden wir die Induktionsvoraussetzung für $k-1$ und $n+1$ an. Es gilt

$$\begin{aligned}
E_{nk}(y) &= \sum_{n \le m \le n+k} R_{mf}(y) \\
&= R_{nf}(y) + \sum_{n < m \le n+k} \sum_s p_{nf}(y,s) R_{mf}(y,s) \quad \text{(Satz 8.2.4, „bedingte W'keit")} \\
&= r_{nf}(y) + \sum_s p_{nf}(y,s) \sum_{n < m \le n+k} R_{mf}(y,s) \\
&= r_{nf}(y) + \sum_s p_{nf}(y,s) \Lambda_{n+1,f} \ldots \Lambda_{n+k,f} 0(y,s) \quad \text{(Ind.vor.)} \\
&= \Lambda_{nf} \ \Lambda_{n+1,f} \ldots \Lambda_{n+k,f} \ 0(y,s) \quad \text{(Def. von } \Lambda_{nf}\text{)}.
\end{aligned}$$

□

Nun kommen wir zur Formalisierung des Optimalitätsprinzips. Da wir für Zustandsfolgen mit Wahrscheinlichkeit 0 keine Aussage treffen können, bezeichnen wir mit dem Träger T_{nf} die Menge aller $y \in S^n$ mit $P_f(y) > 0$. Außerdem sei $T_1 = \{s \mid p_0(s) > 0\}$.

Satz 8.2.9 (Bellman'sches Optimalitätsprinzip):

Falls $\Leftrightarrow\infty < G < \infty$, *sind die folgenden drei Aussagen äquivalent.*

(1) f *ist eine optimale Strategie.*

(2) $G_{1f}(s) = G_1(s)$ *für alle* $s \in T_1$.

(3) $G_{nf}(y) = G_n(h_{nf}(y))$ *für alle* $y \in T_{nf}$ *und alle* $n \in \mathbb{N}$.

Beweis: (1) $\Rightarrow$ (2) Sei f optimal, aber $G_{1f}(s^*) < G_1(s^*)$ für ein $s^* \in T_1$. Nach Definition von G_1 gibt es eine Strategie g mit $G_{1g}(s^*) > G_{1f}(s^*)$. Wir definieren die Strategie h durch

$$\begin{aligned} h(s_1, \ldots, s_n) &:= f(s_1, \ldots, s_n), \text{ falls } s_1 \neq s^*, \text{ und} \\ h(s_1, \ldots, s_n) &:= g(s_1, \ldots, s_n), \text{ falls } s_1 = s^*. \end{aligned}$$

Offensichtlich ist $G_{1h}(s) = G_{1f}(s)$, falls $s_1 \neq s^*$ und $G_{1h}(s) = G_{1g}(s)$, falls $s_1 = s^*$. Da $s^* \in T_1$, ist $p_0(s^*) > 0$. Da $G \in \mathbb{R}$ folgt

$$\begin{aligned} G = G_f &= \sum_s p_0(s)\ G_{1f}(s) \quad \text{(Lemma 8.2.5)} \\ &= \sum_{s \neq s^*} p_0(s)\ G_{1f}(s) + p_0(s^*)\ G_{1f}(s^*) \\ &< \sum_{s \neq s^*} p_0(s)\ G_{1f}(s) + p_0(s^*)\ G_{1g}(s^*) \quad (G \in \mathbb{R}) \\ &= \sum_s p_0(s)\ G_{1h}(s) = G_h \leq G \quad \text{(Lemma 8.2.5)}, \end{aligned}$$

offensichtlich ein Widerspruch.

(2) $\Rightarrow$ (1) Sei g eine beliebige Strategie, dann ist für $s \in T_1$: $G_{1f} = G_1(s) \geq G_{1g}(s)$.

Es folgt: $G_f = \sum_s p_0(s)\ G_{1f}(s) \geq \sum_s p_0(s) G_{1g}(s) = G_g$, und damit ist f optimal.

(2) $\Rightarrow$ (3) Wir beweisen die Behauptung mit Induktion über n. Für $n = 1$ ist die Behauptung genau die Voraussetzung. Für den Induktionsschluß nehmen wir die Existenz eines Paares $(y, s^*) \in T_{n+1,f}$ mit $y \in S^n$, $s^* \in S$ und $G_{n+1,f}(y, s^*) < G_{n+1}(h_{n+1,f}(y, s^*))$ an. Nach Definition von G_{n+1} existiert eine Strategie g mit $G_{n+1,f}(y, s^*) < G_{n+1,g}(y, s^*)$, wobei $g \in \Delta(h_{n+1,f}(y, s^*))$ ist, d.h. g stimmt in der Vergangenheit auf Teilfolgen von y mit f überein. Wir versuchen, f zu verbessern und setzen

$$\begin{aligned} h(s_1, \ldots, s_m) &:= f(s_1, \ldots, s_m), \text{ falls } m \leq n \text{ oder } (s_1, \ldots, s_{n+1}) \neq (y, s^*), \\ h(s_1, \ldots, s_m) &:= g(s_1, \ldots, s_m), \text{ sonst.} \end{aligned}$$

Da h und f bis zum Zeitpunkt n übereinstimmen, ist $h \in \Delta(h_{nf}(y))$. Es folgt

$$
\begin{aligned}
G_{nf}(y) &= \Lambda_{nf}\ G_{n+1,f}(y) && \text{(Lemma 8.2.7)}\\
&= r_{nf}(y) + \sum_s p_{nf}(y,s)\ G_{n+1,f}(y,s) && \text{(Definition } \Lambda_{nf})\\
&= r_{nf}(y) + \sum_{s\neq s^*} p_{nf}(y,s)\ G_{n+1,f}(y,s) + p_{nf}(y,s^*)\ G_{n+1,f}(y,s^*)\\
&< r_{nf}(y) + \sum_{s\neq s^*} p_{nf}(y,s)\ G_{n+1,f}(y,s) + p_{nf}(y,s^*)\ G_{n+1,g}(y,s^*) && \text{(da } p_{nf}(y,s^*)>0)\\
&= r_{nh}(y) + \textstyle\sum_{s\neq s^*} p_{nh}(y,s)\ G_{n+1,h}(y,s)\ + p_{nh}(y,s^*)\ G_{n+1,h}(y,s^*)\\
&= \Lambda_{nh}\ G_{n+1,h}(y)\ =\ G_{nh}(y) \leq G_n(h_{nf}(y))
\end{aligned}
$$

nach Definition von G_n. Dies ist ein Widerspruch zur Induktionsvoraussetzung.

(3) $\Rightarrow$ (2) Dies ist trivial, da die Voraussetzung für $n = 1$ die Behauptung beinhaltet. □

Uns ist also die Formalisierung und Verifikation des Bellman'schen Optimalitätsprinzips geglückt.

8.3 Die Bellman'sche Optimalitätsgleichung

Wir stellen zunächst das Hauptresultat dar.

Satz 8.3.1 (Bellman'sche Optimalitätsgleichung):

$$G = \sum_s p_0(s)\ G_1(s)$$

$$G_n(h) = \sup\{r_n(h,a) + \sum_s p_n(h,a,s)\ G_{n+1}(h,a,s)\ \mid a \in D_n(h)\}$$

Wieder läßt sich festhalten, daß wir nur die zweite Gleichung beweisen müssen, da die erste Gleichung den Spezialfall $n = 0$, einzige Aktion zum Zeitpunkt 0 „Warten", Auszahlung 0 zum Zeitpunkt 0, beschreibt. Die Optimalitätsgleichung gilt, auch wenn keine optimale Strategie existiert. Wenn wir G_{n+1} kennen (z.B. ist bei endlichem Horizont $G_{n+1} \equiv 0$), können wir G_n durch Lösung einer einfachen Maximierungsaufgabe berechnen. Dies mag schwierig sein, wenn der Aktionenraum unendlich oder komplex ist. Es gibt jedoch viele Anwendungen mit endlichem kleinem Aktionenraum. Aus *sup* wird *max*, und wir können die endlich vielen Werte rechts ausrechnen und das Maximum bilden. Allgemein ist die Interpretation der Optimalitätsgleichung einfach. Wenn wir die Aktion $a \in D_n(h)$ wählen, kommen wir mit Wahrscheinlichkeit $p_n(h,a,s)$ in den Zustand s. Das Supremum der zu erwartenden Gewinne ist dann $G_{n+1}(h,a,s)$. Da wir zum Zeitpunkt n zusätzlich $r_n(h,a)$ erhalten, gibt der Klammerausdruck das Supremum der zu erwartenden Auszahlungen bei Aktion a an. Hierüber das Supremum über alle Aktionen, sollte das Bestmögliche sein. Der Operator $\mathcal{L}_n$ wird hier die Rolle spielen, die der Operator Λ_{nf} in Kapitel 8.2 hatte.

Definition 8.3.2: Es sei $K_n := \{(h,a) | h \in H_n, a \in D_n(h)\}$. Für $w : H_{n+1} \to \overline{\mathbb{R}}$ sei $\mathcal{L}_n w : K_n \to \overline{\mathbb{R}}$ definiert durch

$$(\mathcal{L}_n w)(h,a) := r_n(h,a) + \sum_s p_n(h,a,s)\ w(h,a,s).$$

Wir wollen eine Beziehung zwischen $\mathcal{L}_n$ und Λ_{nf} herstellen. Sei $f \in \Delta(h)$ und y die Zustandsfolge in h. Dann ist

$$(\mathcal{L}_n\ G_{n+1})(h,f(y)) = r_n(h,f(y)) + \sum_s p_n(h,f(y),s)\ G_{n+1}(h,f(y),s).$$

Sei nun $h_{n+1,f}(y)$ die Historie, die f bei anfänglicher Historie y erzeugt. Dann ist

$$
\begin{aligned}
(\Lambda_{nf}(G_{n+1} \circ h_{n+1,f}))(y) &= r_{nf}(y) + \sum_s p_{nf}(y,s)\ G_{n+1}(h_{n+1,f}(y,s))\\
&= (\mathcal{L}_n\ G_{n+1})(h,f(y)).
\end{aligned}
$$

Beweis von Satz 8.3.1: Die Behauptung läßt sich mit Hilfe von $\mathcal{L}_n$ neu formulieren:

$$G_n(h) = \sup\{(\mathcal{L}_n\ G_{n+1})(h,a) \mid a \in D_n(h)\}.$$

„$\leq$“ Falls „$>$“ gilt, gibt es eine Strategie $f \in \Delta(h)$, für die $G_{nf}(h)$ größer als die rechte Seite der Behauptung ist. Nach Lemma 8.2.5 ist

$$G_{nf}(h) = r_{nf}(y) + \sum_s p_{nf}(y,s)\ G_{n+1,f}(y,s).$$

Da $f \in \Delta(h)$, ist

$$G_{n+1,f}(y,s) \leq G_{n+1}(h,f(y),s).$$

Also ist

$$\begin{aligned} G_{nf}(h) &\leq r_n(h,f(y)) + \sum_s p_n(h,f(y),s)\ G_{n+1}(h,f(y),s) \\ &\leq \sup\{r_n(h,a) + \sum_s p_n(h,a,s)\ G_{n+1}(h,a,s) | a \in D_n(h)\}, \end{aligned}$$

im Widerspruch zur Annahme.

„$\geq$“ Zunächst nehmen wir an, daß $G_{n+1}(h,a,s) < \infty$ für alle $s \in S$ mit $p_n(h,a,s) > 0$ und ein festes $(h,a) \in K_n$ ist. Die Zustandsfolge in h sei x. Für jedes $\varepsilon > 0$ gibt es nach Definition von G_{n+1} eine Strategie $f^{(s)} \in \Delta(h,a,s)$ mit

$$G_{n+1,f^{(s)}}(x,s) \geq G_{n+1}(h,a,s) \Leftrightarrow \varepsilon.$$

Sei nun $f \in \Delta(h,a,s)$ beliebig. Wir setzen nun f^* aus den guten Strategien $f^{(s)}$ und der Dummy-Strategie f für die Zustände, die mit Wahrscheinlichkeit 0 erreicht werden, zusammen.

$$\begin{aligned} f^*(y_m) &:= f^{(s)}(y_m), \quad \text{falls } m \geq n+1 \text{ und } y_{n+1} = (x,s), \\ f^*(y_m) &:= f(y_m) \quad \text{sonst}. \end{aligned}$$

Damit ist $f^* \in \Delta(h,a,s)$ für alle s. Falls $p_n(h,a,s) > 0$, ist

$$G_{n+1,f^*}(x,s) = G_{n+1,f^{(s)}}(x,s) \geq G_{n+1}(h,a,s) \Leftrightarrow \varepsilon.$$

Also ist

$$\begin{aligned} G_n(h) &\geq G_{nf^*}(x) = r_{nf^*}(x) + \sum_s p_{nf^*}(x,s)\ G_{n+1,f^*}(x,s) \\ &\geq r_n(h,a) + \sum_s p_n(h,a,s)\ (G_{n+1}(h,a,s) \Leftrightarrow \varepsilon) \\ &= (\mathcal{L}_n\ G_{n+1})(h,a) \Leftrightarrow \varepsilon. \end{aligned}$$

Da diese Aussage für alle $\varepsilon > 0$ gilt, folgt $G_n(h) \geq (\mathcal{L}_n\ G_{n+1})(h,a)$ für alle $a \in D_n(h)$ und damit die Behauptung.

Falls $p_n(h,a,s^*) > 0$ und $G_{n+1}(h,a,s^*) = \infty$, sind wir im Fall (EP). Jetzt gibt es für $k \in \mathbb{N}$ eine Strategie $f^{(k)} \in \Delta(h,a,s^*)$ mit $G_{n+1,f^{(k)}}(x,s^*) \geq k$. Dann folgt für beliebige $k \in \mathbb{N}$

$$\begin{aligned} G_n(h) &\geq G_{n,f^{(k)}}(x) = r_n(h,a) + \sum_s p_n(h,a,s)\ G_{n+1,f^{(k)}}(x,s) \\ &\geq r_n(h,a) + k \cdot p_n(h,a,s^*) \Leftrightarrow \sum_{n+1 \leq m < \infty} ||r_m^-||. \end{aligned}$$

Die letzte Reihe konvergiert im Fall (EP). Da wir k beliebig groß wählen können, folgt $G_n(h) = \infty$ und damit die Behauptung. □

Was nützt uns die Optimalitätsgleichung? Bei Problemen mit endlichem Horizont können wir daraus G und jeweils eine optimale Aktion (bei endlichem Aktionsraum) ausrechnen. Dies wird in vielen Algorithmen nach der Dynamischen Programmierung ausgenutzt. Als Beispiele seien genannt: Optimale Klammerung bei Multiplikation vieler Matrizen, kürzeste Wege in Graphen, Rucksackproblem (Datenstrukturen), Cocke-Younger-Kasami Algorithmus zur Syntaxanalyse für kontextfreie Sprachen, Umwandlung regulärer Sprachen in reguläre Ausdrücke (GTI), Rucksackproblem, Traveling Salesman Problem (Effiziente Algorithmen). Die Liste kann erweitert werden durch die Konstruktion optimaler Codes, die Konstruktion optimaler Suchbäume, usw.

Und bei Problemen mit unendlichem Horizont? Wir können auf analoge Weise ε-optimale Strategien berechnen, indem wir künstlich einen endlichen Horizont N (N groß) schaffen, wobei N so groß ist, daß der Fehler ε nicht übersteigt. Es gibt jedoch Möglichkeiten, Funktionalgleichungen wie die Bellman'sche Optimalitätsgleichung zu lösen oder auch (mit Intuition für das spezielle Problem) eine Lösung zu raten, z.B. indem G_{nf} für eine vermutete optimale Strategie berechnet wird. Aber nicht jede Lösung der Funktionalgleichung muß mit G_n übereinstimmen! Wir wissen nur, daß G_n die Funktionalgleichung erfüllt. In den Fällen (EP) und (C) können wir G_n als eine spezielle Lösung der Bellman'schen Optimalitätsgleichung charakterisieren.

Wieder arbeiten wir mit geeigneten Operatoren.

Definition 8.3.3: Für $w : H_{n+1} \to \overline{\mathbb{R}}$ sei $\mathcal{U}_n w : H_n \to \overline{\mathbb{R}}$ definiert durch

$$\mathcal{U}_n w(h) := \sup\{(\mathcal{L}_n w)(h,a) \mid a \in D_n(h)\}.$$

Die Bellman'sche Optimalitätsgleichung lautet nun $G_n = \mathcal{U}_n\, G_{n+1}$.

Satz 8.3.4: *Im Fall (EP) ist (G_n) die punktweise kleinste aller Lösungen (w_n) der Optimalitätsgleichung unter der Nebenbedingung*

$$w_n(h) \geq \Leftrightarrow \sum_{m=n}^{\infty} ||r_m^-|| \text{ für alle } n \text{ und } h.$$

Beweis: (G_n) genügt offensichtlich selber der Nebenbedingung. Sei nun (w_n) eine beliebige Lösung der Optimalitätsgleichung, die der Nebenbedingung genügt. Wir müssen $G_n(h) \leq w_n(h)$ für alle $h \in H_n$ beweisen.

Sei $R_n := \sum_{n \leq m < \infty} ||r_m^-||$. Im Fall (EP) ist R_n eine Nullfolge. Da w_n der Nebenbedingung genügt, ist $w_n(h) + R_n \geq 0$ für alle $n \in \mathbb{N}$ und $h \in H_n$. Sei nun $G_{n0} :\equiv 0$ und $G_{nk} := \mathcal{U}_n\, G_{n+1,k-1}$. Wir greifen nun auf Satz 8.4.3 vor, der besagt, daß

$$\lim_{k \to \infty} G_{nk} = G_n$$

ist. Wir zeigen hier $G_{nk} \leq w_n + R_{n+k}$. Für $k \to \infty$ folgt daraus die Behauptung. Für $k = 0$ ist

$$G_{n0} \equiv 0 \leq w_n + R_{n+0}.$$

Für den Induktionsschritt können wir $G_{n+1,k} \leq w_{n+1} + R_{n+k+1}$ voraussetzen. Also ist

$$\begin{aligned} G_{n,k+1} &= \mathcal{U}_n\, G_{n+1,k} \leq \mathcal{U}_n(w_{n+1} + R_{n+k+1}) \\ &= \mathcal{U}_n w_{n+1} + R_{n+k+1} = w_n + R_{n+k+1}. \end{aligned}$$

Hierbei haben wir ausgenutzt, daß $\mathcal{U}_n$ monoton ist und additive Konstanten herausgezogen werden können. □

Satz 8.3.5: *Im Fall (C) ist (G_n) die einzige Lösung (w_n) der Optimalitätsgleichung unter der Nebenbedingung*

$$||w_n|| \leq \sum_{n \leq m < \infty} ||r_m||.$$

Beweis: Wieder ist klar, daß (G_n) selber die Nebenbedingung erfüllt. Sei (w_n) eine Lösung der Optimalitätsgleichung, die die Nebenbedingung erfüllt. Wir wollen $G_n = w_n$ oder, was äquivalent ist, $||G_n \Leftrightarrow w_n|| = 0$ beweisen. Da (G_n) und (w_n) die Optimalitätsgleichung erfüllen, gilt

$$\begin{aligned} |G_n(h) \Leftrightarrow w_n(h)| &= |\sup\{(\mathcal{L}_n G_{n+1})(h,a) \,|\, a \in D_n(h)\} \Leftrightarrow \sup\{(\mathcal{L}_n w_{n+1})(h,a) \,|\, a \in D_n(h)\}| \\ &\leq \sup\{|(\mathcal{L}_n G_{n+1})(h,a) \Leftrightarrow (\mathcal{L}_n w_{n+1})(h,a)| \,|\, a \in D_n(h)\}. \end{aligned}$$

Diese letzte Ungleichung gilt allgemein für die Vertauschung von $|\cdot|$ und sup (Übungsaufgabe). Nach Definition von $\mathcal{L}_n$ folgt

$$\begin{aligned} |G_n(h) \Leftrightarrow w_n(h)| &\leq \sup\{|r_n(h,a) + \sum_s p_n(h,a,s) G_{n+1}(h,a,s) \Leftrightarrow r_n(h,a) \\ &\quad \Leftrightarrow \sum_s p_n(h,a,s) w_{n+1}(h,a,s)| \,|\, a \in D_n(h)\} \\ &\leq \sup\{\sum_s p_n(h,a,s) |G_{n+1}(h,a,s) \Leftrightarrow w_{n+1}(h,a,s)| \,|\, a \in D_n(h)\} \\ &\leq \sup\{\sum_s p_n(h,a,s) ||G_{n+1} \Leftrightarrow w_{n+1}|| \,|\, a \in D_n(h)\} \\ &= ||G_{n+1} \Leftrightarrow w_{n+1}||. \end{aligned}$$

Da diese Ungleichung für alle $h \in H_n$ gilt, folgt $||G_n \Leftrightarrow w_n|| \leq ||G_{n+1} \Leftrightarrow w_{n+1}||$ und nach k-facher Iteration

$$\begin{aligned} ||G_n \Leftrightarrow w_n|| &\leq ||G_{n+k} \Leftrightarrow w_{n+k}|| \leq ||G_{n+k}|| + ||w_{n+k}|| \\ &\leq 2 \sum_{n+k \leq m < \infty} ||r_m|| \to 0 \quad \text{für } k \to \infty \text{ in Fall (C)}. \end{aligned}$$

Also ist $||G_n \Leftrightarrow w_n|| = 0$ und $G_n \equiv w_n$. □

Wir werden später in konkreten Fällen sehen, daß uns diese abstrakten Charakterisierungen in den Anwendungen tatsächlich helfen.

8.4 Eine Methode für die Approximation des optimalen Gewinns

Wir arbeiten mit den bereits in Kapitel 8.3 vorgestellten Funktionen G_{nk}.

Definition 8.4.1: $G_{nk} : H_n \to \overline{\mathbb{R}}$ ist definiert durch $G_{n0} :\equiv 0$ und $G_{nk} := \mathcal{U}_n G_{n+1,k-1}$ für $n, k \in \mathbb{N}$.

Lemma 8.4.2: $G_{nk}(h) = \sup\{\sum_{m=n}^{n+k-1} R_{mf}(y) \,|\, f \in \Delta(h)\}$.

Mit dieser Aussage wird die formale Definition von G_{nk} belebt. $G_{nk}(h)$ ist der maximal innerhalb von k Zeittakten bei Historie h erreichbare Gewinn. Wenn das so ist, sollte $G_{nk} \to G_n$ für $k \to \infty$ gelten. Zunächst beweisen wir jedoch die angegebene Charakterisierung von G_{nk}.

Beweis von Lemma 8.4.2: Wir bezeichnen die rechte Seite der Behauptung mit $V_{nk}(h)$. Wir zeigen die Behauptung $G_{nk} = V_{nk}$, indem wir zeigen, daß V_{nk} wie G_{nk} definiert ist, d. h. $V_{n0} \equiv 0$ und $V_{nk} = \mathcal{U}_n V_{n+1,k-1}$. Die Aussage $V_{n0} \equiv 0$ gilt offensichtlich, da dann die betrachtete Summe leer ist. Parameter, die einen oberen Index (m) erhalten, beziehen sich auf das abgeänderte Problem, in dem die Belohnungen ab Zeitpunkt m auf 0 gesetzt worden sind. Da in der Definition von $V_{nk}(h)$ nur Zeitpunkte bis $n+k \Leftrightarrow 1$ vorkommen, können wir R_{mf} durch $R_{mf}^{(n+k)}$ ersetzen. Also ist nach Definition von G_n

$$V_{nk}(h) = \sup\{\sum_{n \le m < \infty} R_{mf}^{(n+k)}(y) |\, f \in \Delta(h)\} = G_n^{(n+k)}(h).$$

Es folgt mit dieser Gleichung für (n,k) und $(n+1, k \Leftrightarrow 1)$ und der Optimalitätsgleichung

$$V_{nk} = G_n^{(n+k)} = \mathcal{U}_n G_{n+1}^{(n+k)} = \mathcal{U}_n V_{n+1,k-1}.$$

□

Satz 8.4.3: *Im Fall (EP) gilt* $G_n = \lim_{k\to\infty} G_{n,k}$ *für alle* $n \in \mathbb{N}$.

Beweis: Wir nehmen zunächst $r_n \ge 0$ an. Dann gilt

$$\sum_{n \le m \le n+k-1} R_{mf}(y) \to \sum_{n \le m < \infty} R_{mf}(y) \text{ für } k \to \infty$$

für alle $f \in \Delta(h)$, und der linke Ausdruck ist monoton wachsend in k. Für monoton wachsende Folgen können lim und sup vertauscht werden. Daher folgt

$$\begin{aligned} G_n(h) &= \sup_{f\in\Delta(h)} \{\sum_{m=n}^{\infty} R_{mf}(y)\} \\ &= \sup_{f\in\Delta(h)} \{\lim_{k\to\infty} \sum_{m=n}^{n+k-1} R_{mf}(y)\} \\ &= \lim_{k\to\infty} \sup_{f\in\Delta(h)} \{\sum_{m=n}^{n+k-1} R_{mf}(y)\} \\ &= \lim_{k\to\infty} G_{nk}(h), \end{aligned}$$

wobei die letzte Gleichung aus Lemma 8.4.2 folgt.

Im allgemeinen (EP)-Fall setzen wir $r'_n := r_n + ||r_n^-||$. Dann ist $r'_n \ge 0$. Für diese Auszahlungsfunktion ist die Behauptung richtig. Nach Lemma 8.4.2 ist

$$G'_{nk} = G_{nk} + \sum_{n \le m \le n+k-1} ||r_m^-||,$$

da die Summanden $||r_m^-||$ sichere Auszahlungen sind. Daher ist auch

$$G'_n = G_n + \sum_{n \le m < \infty} ||r_m^-||.$$

Nach dem ersten Teil des Beweises gilt $G'_{nk} \to G'_n$. Da die $||r_m^-||$-Reihe im Fall (EP) konvergiert, folgt $G_{nk} \to G_n$. □

Wir haben die Aussagen von Satz 8.4.3 allgemein motiviert, nun aber nur im Fall (EP) bewiesen. Das hat seinen guten Grund, wie folgendes Beispiel zeigt.

Beispiel 8.4.4: Im Fall (EN) gilt nicht immer $G_{nk} \to G_n$. Sei $S := \mathbb{N}_0$, $A := \{3,4,5,\ldots\}$, $D_n(h) := A$ für alle h und n, $p_0(1) := 1$, $p_n(h,a,j) := p(s_n,a,j)$ mit $p(0,a,0) := 1$, $p(1,a,a) := 1$, $p(2,a,0) := 1/a$, $p(2,a,2) := 1 \Leftrightarrow 1/a$ und $p(s,a,s \Leftrightarrow 1) := 1$ für $s > 2$.

Im Zustand 0 bleibt man für alle Ewigkeiten sitzen. Im Zustand 1, dem Startzustand, kann man wählen, in welchen Zustand $s \geq 3$ man wechselt. Aus den Zuständen $s \geq 3$ steigt man schrittweise in den Zustand 2 ab. Im Zustand 2 bleibt man eine Weile, bis man schließlich in den Zustand 0 fällt. Die Verweildauer (erwartete Verweildauer) im Zustand 2 kann durch die Aktionen beeinflußt werden. Nun zu den Auszahlungen. Es sei $r(2,a) := \Leftrightarrow 1/a$, $r(3,a) := \Leftrightarrow 1$ und $r(s,a) := 0$ für $s \notin \{2,3\}$.

Für $s_n \in \{3,\ldots,k+2\}$ ist $G_{nk}(h) = \Leftrightarrow 1$, da wir in $k \Leftrightarrow 1$ Schritten sicher Zustand 3 erreichen, und in Zustand 2 durch Wahl geeigneter Aktionen die Kosten beliebig klein machen können. Für alle anderen Zustände s_n ist $G_{nk}(h) = 0$, da wir für $s_n = 1$ die Aktion $a = k+2$ wählen können. Insbesondere ist $G_{1k}(1) = 0$ für alle k. Für jede Strategie f ist $f(1) = a \in A$. Nach $a \Leftrightarrow 1$ Schritten erreichen wir Zustand 3 und zahlen eine Einheit. Also ist $G_{1f}(1) \leq \Leftrightarrow 1$ für alle f und $G_1(1) \leq \Leftrightarrow 1$, d. h. $G_{1k}(1) \not\to G_1(1)$. Was gilt für $G_{1k}(2)$ und $G_1(2)$? (Übungsaufgabe).

Das Beispiel basiert darauf, daß wir notwendige Kosten beliebig weit in die Zukunft vertagen können, ohne ihnen aber endgültig ausweichen zu können.

8.5 Die Existenz optimaler Strategien und das Optimalitätsprinzip

Wir haben Kenntnisse über Eigenschaften von G_n erarbeitet. Wann und wie können wir diese Maximalgewinne realisieren? Eine natürliche Bedingung scheint zu sein, zu jedem Zeitpunkt eine optimale Aktion zu wählen. Dieses Kriterium greift allerdings nur im Fall (EN).

Satz 8.5.1 (Optimalitätsprinzip): *Es sei der Fall (EN) gegeben und $G > \Leftrightarrow\infty$. Dann ist die Strategie f genau dann optimal, wenn für alle $n \in \mathbb{N}$ und $y \in T_{nf}$ die Aktion $f(y)$ die Funktion $(\mathcal{L}_n G_{n+1})(h_{nf}(y),\cdot)$ (s. Definition 8.3.2) auf $D_n(h_{nf}(y))$ maximiert.*

Beweis: „$\Rightarrow$“ Für diesen Beweisteil brauchen wir die Voraussetzung (EN) nicht. Es genügt $\Leftrightarrow\infty < G < \infty$ zur Anwendung des Bellman'schen Optimalitätsprinzips (B. O.). Sei also f optimal, $n \in \mathbb{N}$, $y \in T_{nf}$ und $h := h_{nf}(y)$. Dann ist

$$\begin{aligned}
(\mathcal{L}_n G_{n+1})(h, f(y)) &= (\Lambda_{nf}(G_{n+1} \circ h_{n+1,f}))(y) && \text{(s. vor Beweis Satz 8.3.1)}\\
&= \Lambda_{nf} G_{n+1,f}(y) && \text{(B. O.)}\\
&= G_{nf}(y) && \text{(Lemma 8.2.5)}\\
&= G_n(h) && \text{(B. O.)}\\
&= \sup\{(\mathcal{L}_n G_{n+1})(h,a) \,|\, a \in D_n(h)\} && \text{(Optimalitätsgleichung).}
\end{aligned}$$

Also maximiert $f(y)$ die Funktion $(\mathcal{L}_n G_{n+1})(h,\cdot)$ auf $D_n(h)$.

„$\Leftarrow$“ Für $n \in \mathbb{N}$ und $y \in T_{nf}$ ist

$$\begin{aligned}
(\Lambda_{nf}(G_{n+1} \circ h_{n+1,f}))(y) &= (\mathcal{L}_n G_{n+1})(h, f(y)) && \text{(s. vor Beweis Satz 8.3.1)}\\
&= \sup\{(\mathcal{L}_n G_{n+1})(h,a) \mid a \in D_n(h)\} && \text{(Voraussetzung)}\\
&= G_n(h) = G_n(h_{nf}(y)) && \text{(Optimalitätsgleichung).}
\end{aligned}$$

Es folgt induktiv für alle $n \in \mathbb{N}$ und alle $s \in T_1$:

$$(\Lambda_{1f} \ldots \Lambda_{nf}(G_{n+1} \circ h_{n+1,f}))(s) = G_1(s).$$

Des weiteren gilt

$$\begin{aligned}
(\Lambda_{nf}(G_{n+1} \circ h_{n+1,f}))(y) &= (\mathcal{L}_n G_{n+1})(h, f(y))\\
&= r_{nf}(y) + \sum_s p_{nf}(y,s) G_{n+1}(h, f(y), s) \quad \text{(Definition } \mathcal{L}_n)\\
&\leq r_{nf}(y) + \sum_{n+1 \leq m < \infty} ||r_m^+||\\
&= (\Lambda_{nf} 0)(y) + R_n,
\end{aligned}$$

wobei wir R_n als die letzte Summe in der vorletzten Gleichung definieren.

Mit Hilfe der Monotonie der Λ-Operatoren folgt

$$\begin{aligned} G_1(s) &= (\Lambda_{1f} \dots \Lambda_{nf}(G_{n+1} \circ h_{n+1,f}))(s) \\ &\leq \Lambda_{1f} \dots \Lambda_{n-1,f}((\Lambda_{nf}0)(y) + R_n) \\ &= (\Lambda_{1f} \dots \Lambda_{nf}0)(s) + R_n. \end{aligned}$$

Im Fall (EN) gilt $R_n \to 0$ für $n \to \infty$. Nach Lemma 8.2.8 ist

$$G_{1f}(s) = \lim_{k\to\infty} (\Lambda_{1f} \dots \Lambda_{kf}0)(s).$$

Zusammen folgt nun $G_1(s) \leq G_{1f}(s)$. Da aber stets gilt $G_{1f}(s) \leq G_1(s)$, ist $G_1 = G_{1f}$ auf T_1. Nach dem Bellman'schen Optimalitätsprinzip ist f optimal. □

Dieser Beweis zeigt auch deutlich, daß wir bereits eine mächtige Theorie aufgebaut haben. Wir haben im wesentlichen nur alte Ergebnisse gut zusammengeklebt.

Korollar 8.5.2: *Im Fall (EN) und $G > \Leftrightarrow\infty$ ist f genau dann optimal, wenn*

$$G_n(h_{nf}(y)) = (\mathcal{L}_n G_{n+1})(h_{nf}(y), f(y)) \text{ für alle } n \in \mathbb{N} \text{ und } y \in T_{nf} \text{ ist.}$$

Beweis: Das Korollar folgt aus der Definition von $\mathcal{L}_n$ und Satz 8.5.1. □

Beispiel 8.5.3: Im Fall (EP) und $G < \infty$ ist die in Satz 8.5.1 gegebene Bedingung nicht hinreichend für die Optimalität einer Strategie.

Sei $S := \mathbb{N}$ und $A := \{\text{STOP}, \text{WEITER}\}$. Die D_n-Mengen sind so definiert, daß die Aktion STOP genau dann durchgeführt werden darf, wenn sie noch nicht durchgeführt wurde, während die Aktion WEITER immer ausgeführt werden darf. Start ist im Zustand 1, von Zustand s wechselt man mit Sicherheit in den Zustand $s+1$. Die Belohnung ist bei Aktion STOP im Zustand s gerade $1 \Leftrightarrow 1/s$ und sonst 0. Dieses Beispiel ist eine Formalisierung unserer schon diskutierten „Qual der Wahl" Situation.

Es ist $G_n(h) = 1$ für alle Historien h, die Aktion STOP noch nicht enthalten. Wir können $1\Leftrightarrow\varepsilon$ übertreffen, wenn wir lange genug warten, bevor wir Aktion STOP wählen. Es ist $G_n(h) = 0$ für alle Historien h, die Aktion STOP enthalten. Die Strategie f, die stets Aktion WEITER wählt, ist die schlechteste, aber sie genügt dem Optimalitätsprinzip, wie wir jetzt zeigen.

Es sei $h_n = (1, \text{WEITER}, 2, \text{WEITER}, \dots, \text{WEITER}, n)$ die einzige Historie mit positiver Wahrscheinlichkeit unter f. Dann ist (s. o.) $G_n(h_{nf}(y)) = 1$ für $y = (1, 2, \dots, n)$. Andererseits ist

$$\begin{aligned} (\mathcal{L}_n G_{n+1})(h_{nf}(y), f(y)) &= \max\{r_{nf}(1) + G_{n+1}(h_n, \text{STOP}, n+1),\ r_{nf}(2) + G_{n+1}(h_n, \text{WEITER}, n+1)\} \\ &= \max\{1 \Leftrightarrow 1/n + 0, 0 + 1\} = 1. \end{aligned}$$

Also ist die Bedingung aus Korollar 8.5.2 erfüllt.

Warten ist zwar zu jedem Zeitpunkt besser als Zugreifen. Wer aber nie zugreift, erhält gar nichts.

Satz 8.5.4: *Im Fall (EN) existiert eine optimale Strategie, wenn für alle $n \in \mathbb{N}$ und alle $h \in H_n$ die Funktion $(\mathcal{L}_n G_{n+1})(h, \cdot)$ auf $D_n(h)$ ihr Supremum annimmt.*

Beweis: Für $G = \Leftrightarrow\infty$ ist der Satz wahr, weil dann jede Strategie optimal ist.

Sei also $G > \Leftrightarrow\infty$. Wir definieren f induktiv. Für $h \in H_n$ sei $M_n(h)$ die Menge der Maximumpunkte für $(\mathcal{L}_n G_{n+1})(h, \cdot)$. Nach Voraussetzung ist $M_n(h) \neq \emptyset$. Wir wählen $f(s_1)$ beliebig in $M_1(s_1)$. Für $y \in S^n$ wählen wir $f(y)$ beliebig in $M_n(h_{nf}(y))$. Damit ist f nach Korollar 8.5.2 optimal. □

Wir bemerken nur, daß die Bedingung in Satz 8.5.4 nicht notwendig für die Existenz einer optimalen Strategie ist. Sie kann natürlich ohne weiteres für Historien, die bei optimalen Strategien nicht erreicht werden, verletzt sein.

Korollar 8.5.5: *Im Fall (EN) existiert eine optimale Strategie, wenn für alle $n \in \mathbb{N}$ und alle $h \in H_n$ die Menge $D_n(h)$ endlich ist.*

Beweis: Dies folgt unmittelbar aus Satz 8.5.4. □

Die Mathematische Analysis stellt uns eine Fülle von Bedingungen zur Verfügung, die mit Satz 8.5.4 hinreichend für die Existenz optimaler Strategien sind. Die Kenntnis der Existenz optimaler Strategien kann manchmal auch beim Nachweis der Optimalität einer Strategie helfen. Wenn es uns gelingt, für alle bis auf eine Strategie zu zeigen, daß sie nicht optimal sind, haben wir die übrig gebliebene Strategie nur dann als optimal nachgewiesen, wenn wir wissen, daß es eine optimale Strategie gibt.

8.6 Markoffsche Modelle, stationäre Modelle

Definition 8.6.1: Ein Problem heißt Markoff–Problem, wenn D_n, p_n und r_n nur noch vom letzten erreichten Zustand (und von n) abhängen, d.h.,

(1) $D_n(h) = D'_n(s_n)$.

(2) $p_n(h, a, s) = p'_n(s_n, a, s)$.

(3) $r_n(h, a) = r'_n(s_n, a)$.

Eine Strategie heißt Markoff-Strategie, wenn $f_n(h)$ nur von s_n abhängt.

Man beachte, daß eine Markoff-Strategie $f = (f_n)$ nur aus Abbildungen $f_n : S \to A$ besteht.

Satz 8.6.2: *Bei Markoff-Problemen hängt G_n nur von s_n ab, d.h. wir können $G_n(s_n) := G_n(h)$ für h mit n-tem Zustand s_n definieren. Die Optimalitätsgleichung für Markoff-Probleme lautet dann*

$$G_n(s) = \sup_{a \in D_n(s)} \{r_n(s, a) + \sum_j p_n(s, a, j) G_{n+1}(j)\}.$$

Hierbei haben wir schon die Notation vereinfacht, also G_n als Funktion auf S bezeichnet, analog für r_n, D_n und p_n.

Beweis von Satz 8.6.2: Wenn G_n nur vom letzten Zustand abhängt, folgt daraus sofort die angegebene Form der Optimalitätsgleichung. Bleibt also zu zeigen, *daß* G_n nur vom letzten Zustand abhängt.

Sei nun $f \in \Delta(h)$. Dann kann $f_m(s_1, \ldots, \ldots, s_m)$ von $s_1, \ldots, s_m$ abhängen.

Wir betrachten die Werte $G_{mf}(s_1, \ldots, s_m)$ für festes s_m. Wenn in der Menge dieser Werte ein maximaler Wert $G_{mf}(s_1, \ldots, s_n, s'_{n+1}, \ldots, s'_{m-1}, s_m)$ ist, setzen wir $f^*_m(s_m) = f(s_1, \ldots, s_n, s'_{n+1}, \ldots, s'_{m-1}, s_m)$. Wenn die Menge beschränkt ist, wählen wir einen Wert, der sich um höchstens $\varepsilon/2^m$ vom Supremum unterscheidet und definieren $f^*_m(s_m)$ entsprechend. Falls die Menge unbeschränkt ist, wählen wir einen Wert, der größer als eine vorgegebene Konstante ist. Mit dieser Markoff-Strategie f^* können wir die Güte von f „fast“ erreichen. Dies läßt sich für $G < \infty$ folgendermaßen zeigen. Wenn wir f^* nur zum Zeitpunkt n und dann f anwenden, verlieren wir gegenüber f höchstens einen Gewinn von $\varepsilon/2^m$. Wenn wir f^* nur zu den Zeitpunkten $n, \ldots, m$ und dann f anwenden, ist der Verlust gegenüber f durch $\varepsilon/2^n + \ldots + \varepsilon/2^m$ beschränkt. Also ist der Verlust von f^* gegenüber f durch ε beschränkt. Damit ist das Supremum aller erwarteten Gewinne von Markoff-Strategien nicht kleiner als das Supremum der erwarteten Gewinne beliebiger Strategien. Dies folgt auch für den Fall $G = \infty$. □

Satz 8.6.3: *Für Markoff-Probleme vom Typ (EN) existiert eine optimale Markoff-Strategie, wenn eine optimale Strategie existiert.*

Beweis: Für $G = \Leftrightarrow\infty$ ist die Aussage trivial. Sei nun $G > \Leftrightarrow\infty$ und f optimal. Nach Korollar 8.5.2 gilt

$$G_n(h_{nf}(y)) = \mathcal{L}_n G_{n+1}(h_{nf}(y), f_n(y))$$

für $n \in \mathbb{N}$ und $y \in T_{nf}$. Nach Satz 8.6.2 können wir dies umschreiben zu

$$G_n(s_n) = \mathcal{L}_n G_{n+1}(s_n, f_n(y)),$$

wobei s_n ein n-ter Zustand mit positiver Wahrscheinlichkeit unter f ist. Wir wollen nun eine optimale Markoff-Strategie g entwerfen. Da es zum Zeitpunkt 1 noch keine Vergangenheit gibt, ist $g_1 := f_1$ geeignet. Wir nehmen an, daß $g_1, \ldots, g_{n-1}$ so definiert sind, daß nur solche n-ten Zustände positive Wahrscheinlichkeit haben, die unter f positive Wahrscheinlichkeit haben. Sei s_n so ein Zustand. Da es nur abzählbar unendlich viele Zustandsfolgen $y = (s_1, \ldots, s_n)$ mit n-tem Zustand s_n gibt, muß eine davon unter f positive Wahrscheinlichkeit haben. Wir setzen $g_n(s_n) := f_n(y)$. Dann gilt

$$G_n(s_n) = \mathcal{L}_n G_{n+1}(s_n, g_n(s_n)),$$

und g ist eine optimale Markoff-Strategie. □

Das folgende Beispiel soll zeigen, daß ein analoges Vorgehen im Fall (EP) nicht zum Ziel führt.

Beispiel 8.6.4: $S = \mathbb{N}_0$, $A = \mathbb{N}$, $D_n(s) = A$, $p_0(0) = 0$, $p_0(i) = 2^{-i}$, $p_n(i, \cdot, 0) = 1$, $r_2(s_2, a_2) = 2^{a_2}$, $r_n \equiv 0$ für $n \neq 2$. Wir starten irgendwo, wechseln dann in den Zustand 0, wo wir verbleiben. Eine Auszahlung gibt es nur zum Zeitpunkt 2.

Folgende Strategie ist optimal, da sie die erwartete Auszahlung ∞ ergibt. Nur die zweite Aktion ist wichtig. Wir setzen $f_2(s_1, a_1, s_2) := s_1$. Die erwartete Auszahlung beträgt

$$\sum_{1 \leq i < \infty} 2^{-i} 2^i = \infty.$$

Für jede Markoff-Strategie beträgt die Auszahlung $2^{f_2(0)} < \infty$. Wenn wir bei der Markoff-Strategie würfeln dürfen, können wir auch die erwartete Auszahlung ∞ erzielen, indem wir $f_2(0) := a$ mit Wahrscheinlichkeit 2^{-a} wählen. Die Nicht-Markoff-Strategie war deterministisch, sie hat den Anfangspunkt, der von der „Natur“ ausgewürfelt wird, als Wahrscheinlichkeitsexperiment benutzt. Dies ist einer Markoff-Strategie verboten, da das „Würfeln“ der Natur zu lange zurück lag.

Eine weitere Spezialisierung von Markoff-Problemen, bei denen wir die Abhängigkeit der Parameter von der Zeit ausschließen, führt zu stationären Problemen.

Definition 8.6.5: Ein Markoff-Problem heißt stationär, wenn D_n und p_n nicht von n abhängen und es eine Funktion $r : S \times A \to \mathbb{R}$ mit $r_n = \beta^{n-1} r$ für ein $\beta \in (0, 1]$ gibt.

Die Unterfälle (EN), (EP) und (C) erhalten dann folgendes Aussehen.

(EN): $r \leq 0$ oder ($\beta < 1$ und r nach oben beschränkt).
(EP): $r \geq 0$ oder ($\beta < 1$ und r nach unten beschränkt).
(C): $r \equiv 0$ oder ($\beta < 1$ und r beschränkt).

Definition 8.6.6: Eine Markoff-Strategie heißt stationär, wenn f_n nicht von n abhängt.

Satz 8.6.7: *Sei ein stationäres Markoff–Problem gegeben. Es gilt:*

a) Für stationäre Strategien f ist $G_{nf} = \beta^{n-1} G_{1f}$.

b) Sei s_n der letzte Zustand von $h \in H_n$. Dann ist $G_n(h) = \beta^{n-1} G_1(s_n)$.

c) $G_1 = \mathcal{U}_1 \beta G_1$.

Beweis: a) Nach Voraussetzung ist $R_{mf} = \beta^{n-1} R_{m-n+1,f}$, da nur $r_k = \beta^{k-1} r$ über den Diskontfaktor β von der Zeit abhängt. Also ist

$$\begin{aligned} G_{nf}(h) &= G_{nf}(s_n) = \sum_{n \le m < \infty} R_{mf}(s_n) = \sum_{n \le m < \infty} \beta^{n-1} R_{m-n+1,f}(s_n) \\ &= \beta^{n-1} \sum_{1 \le m < \infty} R_{mf}(s_n) = \beta^{n-1} G_{1f}(s_n). \end{aligned}$$

b) Wir wissen, daß für Markoff-Probleme $G_n(h)$ nur vom letzten Zustand s_n von h abhängt. Nun folgt die Behauptung analog zu Teil a).

c) Die Optimalitätsgleichung läßt sich nun schreiben als

$$G_1(s) = \sup_{a \in D(s)} \{r(s,a) + \beta \sum_j p(s,a,j) G_1(j)\}.$$

Also ist $G_1 = \mathcal{U}_1 G_2 = \mathcal{U}_1 \beta G_1$ nach Teil b). □

Satz 8.6.8: *Für stationäre Markoff-Probleme vom Typ (EN) gibt es eine optimale stationäre Strategie, wenn es eine optimale Strategie gibt.*

Beweis: Wir wissen, daß es eine optimale Markoff-Strategie $g = (g_n)$ gibt, die mit ihren Aktionen $g_n(s)$ den Ausdruck $\mathcal{L}_n G_{n+1}(s, \cdot)$ maximiert. Nach Satz 8.6.7 maximiert $g_n(s)$ auch den Ausdruck $\mathcal{L}_n \beta^n G_1(s, \cdot)$ für alle s, die mit positiver Wahrscheinlichkeit als n–ter Zustand auftreten. Wir werden nun eine stationäre Strategie f entwerfen, die so beschaffen ist, daß nur Zustände mit positiver Wahrscheinlichkeit erreicht werden, die auch unter g mit positiver Wahrscheinlichkeit erreicht werden. Wenn wir auf solchen Zuständen s das Verhalten von g_n simulieren, erhalten wir nach der Optimalitätsgleichung eine optimale stationäre Strategie, wenn unter g der Zustand s zum Zeitpunkt n mit positiver Wahrscheinlichkeit erreicht wird. Zunächst definieren wir $f(s) = g_1(s)$ für alle s mit $p_0(s) > 0$. Damit erreichen wir als zweite Zustände nur solche Zustände mit positiver Wahrscheinlichkeit, die wir auch unter g mit positiver Wahrscheinlichkeit erreichen. Für solche Zustände s, für die $f(s)$ noch nicht definiert ist, setzen wir $f(s) = g_2(s)$. Wir fahren analog fort. Insgesamt erreichen wir unter f höchstens weniger Zustände mit positiver Wahrscheinlichkeit als unter g. Wenn f von g abweicht, ahmt f ein früheres Verhalten von g nach und erreicht also nur Zustände mit positiver Wahrscheinlichkeit, die unter g mit positiver Wahrscheinlichkeit erreicht werden. Für Zustände s, für die $f(s)$ durch obiges Verfahren nicht definiert wurde, kann $f(s)$ beliebig definiert werden. □

8.7 Übersicht über die Anwendungen

Wir beginnen mit einer ganzen Problemklasse, dem optimalen Stoppen einer Markoffkette, d.h. wir beobachten eine Markoffkette und dürfen uns einmal entscheiden, die Kette zu stoppen. Dies ist bei Kaufentscheidungen häufig der Fall. Wir bekommen eine Reihe von Angeboten, wobei wir auch langsam lernen, wie die Angebotspalette aussieht. Leider dürfen wir auf alte Angebote nicht zurückgreifen (andere haben das Haus gekauft, den Job genommen, ...). Wann entscheiden wir uns zuzugreifen?

Wir wenden uns dann speziell Optimierungsproblemen mit diskontierten Kosten zu. Beispielprobleme sind

- Maschinenerneuerung
- Qualitätskontrolle
- Hausverkauf.

In diesem Zusammenhang befassen wir uns auch mit der effizienten Berechnung optimaler Strategien.

8.8 Optimales Stoppen einer Markoffkette

Wir betrachten nun den Spezialfall, daß wir in einem Markoff-Problem nur die Aktionen „Stoppen“ und „Warten“ zur Verfügung haben. Wenn wir stoppen, erhalten wir eine Auszahlung, die vom gegenwärtigen Zustand abhängt. Da wir die Auszahlungsfunktion hier mehr als die zentrale Funktion ansehen, bezeichnen wir sie mit f. Nachdem wir gestoppt haben, stirbt die Markoffkette. Formal wechseln wir in einen Zustand, in dem nichts mehr passiert. Wenn wir warten, erhalten wir keine Auszahlung. Die Übergangswahrscheinlichkeit von Zustand x in Zustand y wird mit $p(x,y)$ bezeichnet. Des weiteren setzen wir voraus, daß f beschränkt ist. O. B. d. A. können wir annehmen, daß $f \geq 0$ ist (bei negativer Auszahlung stoppen wir sowieso nicht) und daß $f \not\equiv 0$ ist (sonst ist das Problem trivial). Wir befinden uns also im Fall (EP), und die Existenz optimaler Strategien ist nicht gesichert. Die Optimalitätsgleichung lautet

$$G(x) = \max\{\sum_y p(x,y)G(y), f(x)\},$$

da wir nur zwei Aktionen betrachten müssen. Außerdem ist G die punktweise kleinste aller Lösungen der Optimalitätsgleichung. Die zusätzliche Bedingung würde hier $G \geq 0$ lauten und kann wegfallen, da $f \geq 0$ ist.

In der Praxis wird der Zustandsraum häufig endlich sein, nehmen wir also $S = \{1, \ldots, n\}$ an. Für jede Lösung w der Optimalitätsgleichung gilt (da G die punktweise kleinste Lösung ist):

$$w(x) \geq G(x) \text{ für } x \in S.$$

Falls $w \neq G$, ist $w(y) > G(y)$ für mindestens ein y. Also gilt sogar $\sum_{x\in S} w(x) > \sum_{x\in S} G(x)$.

Damit hat G unter allen Lösungen der Optimalitätsgleichung die kleinste Summe der Funktionswerte. Damit können wir G mit Hilfe der folgenden Linearen Optimierungsaufgabe berechnen. Die Variablen sind $w(x)$, $x \in S$.

$$\begin{array}{lll} \textbf{min} \sum_{x\in S} w(x) & \text{unter den Nebenbedingungen} & \\ w(x) \geq \sum_{y\in S} p(x,y)w(y) & & x \in S, \\ w(x) \geq f(x) & & x \in S, \end{array}$$

Wir können also nun die Funktion G (bei endlichen Zustandsräumen) mit der Linearen Programmierung berechnen. Daher setzen wir in Zukunft die Kenntnis von G voraus.

Definition 8.8.1: Die Menge der Zustände x mit $G(x) = f(x)$ heißt Stützmenge und wird mit S^* bezeichnet.

Offensichtlich ist sofortiges Stoppen in $x \in S^*$ optimal. Stoppen in $x \notin S^*$ ist nicht optimal. Daraus folgt aber nicht, daß Warten optimal ist. Wir wissen ja nicht, ob stets eine optimale Strategie existiert.

Beispiel 8.8.2: $S = \mathbb{N}$, $f(n) = 1 \Leftrightarrow 1/n$, $p(n, n+1) = 1$, Start in 1. Offensichtlich ist $G \equiv 1$, und die Stützmenge leer. Würden wir nur in der Stützmenge stoppen, wäre die Auszahlung 0.

Wir können aber zeigen, daß solche Beispiele nur für unendliche Zustandsräume möglich sind.

Satz 8.8.3: *Die Strategie, beim ersten Eintreten in die Stützmenge zu stoppen, ist für endliche Zustandsräume S optimal.*

Dem Beweis dieser Aussage widmen wir uns nun.

Nach Satz 8.4.3 gilt für die Funktion G im Fall (EP), daß $G(x) = \lim_{k\to\infty} G_k(x)$, wobei G_k der maximal in k Zeiteinheiten erzielbare erwartete Gewinn ist. Es sollte klar sein, daß für den Markoffkettenfall $G_1(x) = f(x)$ ist und $G_{i+1}(x) = \max\{\sum_y p(x,y) \cdot G_i(y), f(x)\}$.

Für den Operator γ^* mit

$$(\gamma^* v)(x) := \max\{\sum_y p(x,y) \cdot v(y), f(x)\}$$

gilt also $G_{i+1} = \gamma^* G_i$.

Operator γ^* ist monoton. Da $G_2 \geq G_1$ ist, wie man leicht verifizieren kann, folgt induktiv $G_{i+1} \geq G_i$ für alle i. Also gilt für den Operator γ mit

$$(\gamma v)(x) := \max\{\sum_y p(x,y) \cdot v(y), v(x)\}$$

auch $G_{i+1} = \gamma G_i$.

Wir wollen nun eine optimale Strategie τ^* beschreiben. Sei $ST := \{s \in S \mid G(s) = f(s)\}$ die sogenannte „Stützmenge“. Strategie τ^* sei wie folgt definiert: Stoppe sofort, wenn sich die Markoffkette in einem Zustand aus ST befindet, sonst mache weiter. Wir behaupten, daß die Strategie τ^* optimal ist. Um dies zu zeigen, müssen wir ein bißchen arbeiten.

Analog zu Satz 8.4.3 gilt im Falle (EP) natürlich für jede Strategie τ, daß man G^τ berechnen kann als Limes der in k Zeiteinheiten erwarteten Auszahlungen für die Strategie τ. Also ist

$$G_1^{\tau^*}(x) = \begin{cases} f(x), & \text{falls } x \in ST, \\ 0, & \text{falls } x \notin ST. \end{cases}$$

Außerdem

$$G_{i+1}^{\tau^*}(x) = \begin{cases} f(x), & \text{falls } x \in ST, \\ \sum_y p(x,y) \cdot G_i^{\tau^*}(y), & \text{falls } x \notin ST, \end{cases}$$

bzw. (für einen entsprechend definierten Operator γ^{**}):

$$G_{i+1}^{\tau^*} = \gamma^{**} G_i^{\tau^*}.$$

Zunächst wollen wir beobachten, daß wir äquivalenterweise auch $G_{i+1}^{\tau^*} = \gamma G_i^{\tau^*}$ schreiben können. Der Grund dafür ist der folgende:

Der γ^{**}–Operator ist monoton und es gilt $G_2^{\tau^*} \geq G_1^{\tau^*}$, wie man leicht verifizieren kann. Dadurch gilt für alle i, daß $G_i^{\tau^*} \geq G_{i-1}^{\tau^*}$ ist. Also kann man in der Definition von $G_{i+1}^{\tau^*}(x)$ für $x \notin ST$ ruhig $G_{i+1}^{\tau^*}(x) = \max\{\sum_y p(x,y) \cdot G_i^{\tau^*}(y), G_i^{\tau^*}(x)\ \}$ schreiben, ohne die Folge zu verändern.

Es ist $G_i^{\tau^*} \leq G_i$. (τ^* ist eine feste Strategie und G_i ist das Supremum über alle Strategien.) Für $x \in ST$ beobachten wir:

$$f(x) = G_{i+1}^{\tau^*}(x) = (\gamma^{**} G_i^{\tau^*})(x) \leq (\gamma G_i)(x) = G_{i+1}(x) = f(x),$$

also $(\gamma^{**} G_i^{\tau^*})(x) = (\gamma G_i)(x)$.

Also können wir feststellen, daß auch die $G_i^{\tau^*}$–Folge mit Hilfe des γ–Operators geschrieben werden kann.

Grund genug, das Verhalten einer beliebigen Funktion v unter dem iterierten γ–Operator zu analysieren. Man stellt sich das v nun besser als Array $v(1), \ldots, v(s)$ vor.

Zu einem solchen Array definieren wir eine Folge von Arrays v_i wie folgt: $v_1 := v$ sowie $v_{i+1} := \gamma v_i$. Sei $v^{max} := \max_i v(i)$. Nach Definition gilt für jedes x und alle i die folgende Aussage:

$$v(x) \leq v_i(x) \leq v^{max}.$$

Dies kann man induktiv nachweisen, denn $\sum p(x,y) v_i(x) \leq \sum p(x,y) v^{max} = v^{max}$.

Damit gilt für alle x, daß die Folge $v_i(x)$ monoton wachsend und beschränkt ist, somit konvergiert sie gegen eine Zahl. Damit konvergiert die Folge der Arrays gegen das Array

$$(\lim_{i\to\infty} v_i(1), \ldots, \lim_{i\to\infty} v_i(s)).$$

Dieses Konvergenzarray bezeichnen wir mit $\mathcal{L}im\ v$. Natürlich gilt $\mathcal{L}im\ v \geq v$.

Als Stützstellen bezeichnen wir alle i, für die $(\mathcal{L}im\ v)_i = v_i$ gilt. Das sind also alle Arraypositionen, auf denen sich in der Folge nie etwas verändert.

Wir halten zunächst folgende einfache Monotonieeigenschaft fest: Falls $v^* \leq v^{**}$ (komponentenweises Kleinergleich) gilt, so ist auch $\mathcal{L}im\ v^* \leq \mathcal{L}im\ v^{**}$. Dies gilt wegen der Monotonie des γ–Operators. Außerdem gilt natürlich auch $\mathcal{L}im\ \mathcal{L}im\ v = \mathcal{L}im\ v$.

Satz 8.8.4: *Sei $w \leq v$ ein Array, das an den Stützstellen von v die gleichen Werte hat wie v. Dann gilt $\mathcal{L}im\ w = \mathcal{L}im\ v$.*

Beweis: Sei $V := \mathcal{L}im\ v$ und $ST = \{i \mid V_i = v_i\}$ bezeichne die Menge der Stützstellen. $\overline{ST}$ bezeichne die Menge der Nichtstützstellen. Falls $\overline{ST} = \emptyset$ ist, gilt $v = w$, und es ist nichts zu zeigen. Sei nun also $\overline{ST}$ nicht leer.

Wir sagen, daß ein Vektor g die Eigenschaft $E(\Delta)$ (für ein $\Delta > 0$) hat, wenn folgende drei Eigenschaften gelten:

a) g stimmt an den Stützstellen mit v (und V) überein.
b) g ist an allen Nichtstützstellen mindestens um Δ kleiner als V.
c) $\mathcal{L}im\ g = V$.

Offensichtlich hat der Vektor v die Eigenschaft $E(\delta)$ für $\delta := \min_{i\in\overline{ST}}\{V_i \Leftrightarrow v_i\}$ mit $\delta > 0$.

Wir zeigen nun: wenn es einen Vektor g mit der Eigenschaft $E(\Delta)$ gibt, dann gibt es auch einen Vektor $g' \leq g$ mit der Eigenschaft $E(2\Delta)$.

Sei g' der Vektor, der aus g hervorgeht, indem wir in jeder Nichtstützstelle Δ subtrahieren. Damit treffen Eigenschaft a) und b) auf g' zu (und zwar mit 2Δ).

$\mathcal{L}im\ g'$ stimmt an den Stützstellen mit V überein. (Grund: $\mathcal{L}im\ g' \geq g'$. Außerdem $g' \leq V$, also $\mathcal{L}im\ g' \leq \mathcal{L}im\ V = V$. V und g' stimmen aber an den Stützstellen mit v überein.)

An den Nichtstützstellen ist $\mathcal{L}im\ g'$ höchstens um Δ kleiner als $\mathcal{L}im\ g = V$. Dies folgt aus der Definition der Arrayfolge (man kann es wieder induktiv für jedes Element der Folge nachweisen).

Damit ist $\mathcal{L}im\ g'$ ein Array, das $\geq g$ ist. Also gilt $\mathcal{L}im\ g' \geq \mathcal{L}im\ g = V$. Wegen $g' \leq V$ ist auch $\mathcal{L}im\ g' \leq V$ und somit $\mathcal{L}im\ g' = V$. Also gilt auch Aussage c).

Nun gibt es induktiv für jedes $\Delta > 0$ einen Vektor, der die Eigenschaft $E(\Delta)$ hat. Wähle $\Delta := \max_i(V_i \Leftrightarrow w_i)$, dann gibt es einen Vektor w' mit der Eigenschaft $E(\Delta)$, also $w' \leq w$. Für diesen gilt wegen Eigenschaft c) $\mathcal{L}im\ w' = V$ wegen $w' \leq w \leq v$ also auch $\mathcal{L}im\ w = V$. □

Damit sollte nun klar sein, daß wir die Optimalität der Strategie τ^* nachgewiesen haben, denn man kann G^{τ^*} als $\mathcal{L}im\ G_1^{\tau^*}$ schreiben und G als $\mathcal{L}im\ G_1 = \mathcal{L}im\ f$. Da $G_1^{\tau^*} \leq G_1$ gilt und $G_1^{\tau^*}$ an der Stützmenge, also an der Menge der Stützstellen, mit G_1 übereinstimmt, besagt der gerade bewiesene Satz, daß $\mathcal{L}im\ G_1^{\tau^*} = \mathcal{L}im\ G_1$, also $G = G^{\tau^*}$ ist.

Für Zweifler schieben wir noch den Beweis des folgenden Satzes nach:

Satz 8.8.5: *$\mathcal{L}im\ \mathcal{L}im\ w = \mathcal{L}im\ w$ für alle w.*

Beweis: $\underline{\varepsilon}$ bezeichne das Array, das s mal den Eintrag ε hat. Wir beobachten, daß für alle $w_1 \leq w_2$ mit $||w_2 \Leftrightarrow w_1|| \leq \varepsilon$ gilt, daß

$$\mathcal{L}im\ w_1 \leq \mathcal{L}im\ w_2 \leq \underline{\varepsilon} + \mathcal{L}im\ w_1 \text{ ist.}$$

In der Arrayfolge zu einem w gibt es nach Definition des Limes zu jedem $\varepsilon > 0$ ein w_n, so daß $||\mathcal{L}im\ w \Leftrightarrow w_n|| \leq \varepsilon$ ist. Da w_n ein Element der Folge ist, ist $\mathcal{L}im\ w_n = \mathcal{L}im\ w$.

Nun ist nach der gerade gemachten Bemerkung

$$0 \leq \mathcal{L}im\ \mathcal{L}im\ w \Leftrightarrow \mathcal{L}im\ w \leq \underline{\varepsilon}.$$

Da diese Aussage für alle $\varepsilon > 0$ gilt, gilt Gleichheit. □

Für endliche Zustandsräume haben wir das Problem des optimalen Stoppens von Markoffketten gelöst. Wenn wir den Beweis noch einmal rekapitulieren, haben wir nur an einer einzigen Stelle ausgenutzt, daß die Zustandsmenge endlich ist, nämlich an der Stelle, wo wir $\delta > 0$ geschrieben haben für $\delta = \min_{i \in \overline{ST}}\{V_i \Leftrightarrow v_i\}$, denn $\overline{ST}$ ist endlich, wenn S endlich ist.

Schließlich beweisen wir noch ein Lemma, das in der Praxis manchmal Anwendung findet, weil man die Stützmenge besonders einfach berechnen kann.

Lemma 8.8.6: *Sei S endlich und $B := \{s \mid (\gamma f)(s) = f(s)\}$. Wenn $p(b, y) = 0$ ist für alle $b \in B$ und $y \notin B$, dann ist B die Stützmenge.*

Beweis: Wir wissen $G = \mathcal{L}im\ f$ und $G \geq \gamma f \geq f$. Falls $s \notin B$, dann folgt $\gamma f(s) > f(s)$, also $G(s) > f(s)$ und s ist nicht in der Stützmenge.

Falls $s \in B$, dann beobachten wir

$$(\gamma^2 f)(s) = \max\{\sum_{y \in S} p(s,y) \cdot (\gamma f)(y), f(s)\} = \max\{\sum_{y \in B} p(s,y) \cdot (\gamma f)(y), f(s)\}$$

$$= \max\{\sum_{y \in B} p(s,y) \cdot f(y), f(s)\}(\gamma f)(s)$$

und somit induktiv $(\mathcal{L}im\ f)(s) = f(s)$, also ist s in der Stützmenge. □

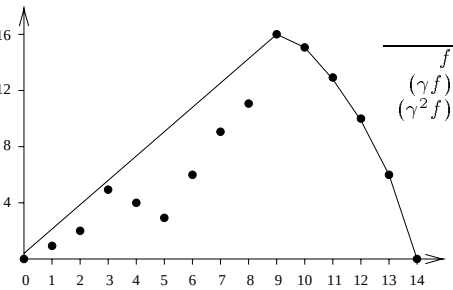
Übrigens kann man für unendliche Zustandsräume analog zu Satz 8.8.3 folgendes Resultat beweisen.

Satz 8.8.7: *Sei $\varepsilon > 0$ und S_ε^* die Menge der Zustände x mit $G(x) \Leftrightarrow f(x) \leq \varepsilon$. Die Strategie τ_ε^*, beim ersten Eintritt in S_ε^* zu stoppen, ist ε-optimal.*

8.9 Beispiele für das optimale Stoppen von Markoffketten

Wir betrachten zwei Beispiele, in denen wir die Lösung ohne Rechnerhilfe berechnen können.

Wir beginnen mit einer eindimensionalen Irrfahrt mit Absorption in den Endpunkten, d. h. $S = \{0, \ldots, n\}$, $p(0,0) = p(n,n) = 1$ und $p(x, x+1) = p(x, x \Leftrightarrow 1) = 1/2$ für $0 < x < n$.



x	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
$f(x)$	0	1	2	5	4	3	6	9	11	16	15	13	10	6	0
$(\gamma f)(x)$	0	1	3	5	4	5	6	9	12.5	16	15	13	10	6	0
$(\gamma^2 f)(x)$	0	1.5	3	5	5	5	7	9.25	12.5	16	15	13	10	6	0

Trivialerweise gilt $G(0) = f(0) = 0$, $G(14) = f(14) = 0$, $G(9) = f(9) = 16$, da f dort sein Maximum hat. Auch gilt offensichtlich $G(5) > f(5)$, da einmaliges Warten besser als Stoppen ist. Sollen wir im Zustand 3 stoppen? Einmaliges Warten verschlechtert die Auszahlung. Ist aber die Hoffnung, in den Zustand 9 zu gelangen, verlockend genug, um der Gefahr zu begegnen, im Zustand 0 „verschluckt“ zu werden? Wir wissen, was für G gelten muß, nämlich

$$g(0) = f(0), g(n) = f(n),$$
$$g(x) \geq \frac{1}{2}g(x \Leftrightarrow 1) + \frac{1}{2}g(x+1), \quad \text{für } 0 < x < n,$$

und außerdem muß g die punktweise kleinste solche Lösung g sein.

Also ist G die kleinste konkave Majorante von f. In unserem Fall ist G linear auf $[0, 9]$, d. h. $G(x) = \frac{16}{9}x$ für $x \in \{0, \ldots, 9\}$, und $G(x) = f(x)$ für $x \in \{9, \ldots, 14\}$. Die Stützmenge S^* enthält die Zustände 0, 9, 10, 11, 12, 13, 14. Es ist optimal, beim ersten Eintreten in S^* zu stoppen. Im Zustand 3 muß also gewartet werden.

Das zweite Beispiel ist das Problem der besten Wahl. Wir stellen uns einen typischen Autofahrer vor. Er fährt, ohne vorher zu planen, wo er übernachten soll. Natürlich wird er auf keinen Fall umkehren oder die Autobahn verlassen. Da er zwischen 20 und 22 Uhr die Fahrt unterbrechen will, kann er die Zahl n der Hotels, an denen er vorbeikommt, abschätzen. Er möchte die Wahrscheinlichkeit, das beste Hotel zu erwischen, maximieren. Wenn er das i-te Hotel gesehen hat, kann er die ersten i Hotels ihrer Güte nach klassifizieren. A priori seien alle Reihenfolgen gleichwahrscheinlich, d. h. jedes Objekt (Hotel) ist mit Wahrscheinlichkeit $1/n$ das beste. Es gibt sicher praxisrelevantere Situationen, die auf gleiche Weise modelliert werden können.

Es sei $G(n)$ die Erfolgswahrscheinlichkeit einer optimalen Strategie. Erstaunlicherweise liefert bereits eine sehr einfache Strategie eine Erfolgswahrscheinlichkeit von mehr als 1/4. Sei n gerade. Wir warten $n/2$ Objekte lang ab und wählen das erste Objekt, das besser als die ersten $n/2$ Objekte ist. Bei dieser Strategie erhalten wir das beste Objekt zum Beispiel dann, wenn das zweitbeste Objekt in der ersten und das beste Objekt in der zweiten Hälfte ist. Die Wahrscheinlichkeit dafür beträgt ($H_1 = 1$. Hälfte, $H_2 =$ 2. Hälfte)

$$\begin{aligned} P(\text{best } \in H_2,\ \text{2-best } \in H_1) &= P(\text{best } \in H_2 |\ \text{2-best } \in H_1) P(\text{2-best } \in H_1) \\ &= \frac{n/2}{n \Leftrightarrow 1} \cdot \frac{n/2}{n} = \frac{1}{4} \cdot \frac{n}{n \Leftrightarrow 1} > \frac{1}{4}. \end{aligned}$$

Wir wollen das Problem der besten Wahl nun als Problem des Stoppens einer Markoffkette modellieren. Es sei $S = \{1, \ldots, n\}$, und wir wollen Zustand k genau dann erreichen, wenn k das beste der ersten k Objekte ist und nicht vorher gestoppt wurde. Dann wird sicher im Zustand 1 gestartet, da 1 das beste der ersten 1 Objekte ist. Für die Übergangswahrscheinlichkeit $p(k, l)$ gilt nun sicher $p(k, l) = 0$ für $l \leq k$. Es sei B die Menge der Zustände k, die die besten der ersten k Objekte sind. Dann gilt für $l > k$

$$\begin{aligned} p(k,l) &= P\{l \in B,\ l \Leftrightarrow 1,\ \ldots,\ k+1 \notin B\} \\ &= P\{l \in B |\ l \Leftrightarrow 1, \ldots,\ k+1 \notin B\} P\{l \Leftrightarrow 1 \notin B |\ l \Leftrightarrow 2, \ldots,\ k+1 \notin B\} \cdot \ldots \cdot \\ &\quad P\{k+2 \notin B | k+1 \notin B\} P\{k+1 \notin B\} \\ &= P\{l \in B\}\ P\{l \Leftrightarrow 1 \notin B\} \cdot \ldots \cdot P\{k+2 \notin B\}\ P\{k+1 \notin B\}. \end{aligned}$$

Die letzte Gleichung folgt aus der Gleichwahrscheinlichkeit aller Permutationen. Damit hängt die Zugehörigkeit von j zu B nicht von der Rangordnung der ersten $j \Leftrightarrow 1$ Objekte ab.

Es folgt

$$p(k,l) = \frac{1}{l} \cdot \frac{l \Leftrightarrow 2}{l \Leftrightarrow 1} \cdot \frac{l \Leftrightarrow 3}{l \Leftrightarrow 2} \cdot \ldots \cdot \frac{k+1}{k+2} \cdot \frac{k}{k+1} = \frac{k}{l \cdot (l \Leftrightarrow 1)}.$$

Formal brauchen wir noch einen Zustand, in den die Markoffkette gerät, wenn kein folgendes Objekt besser ist. In dem Zustand bleiben wir und erhalten nichts. Die Auszahlungsfunktion $f(k)$ ist die Wahrscheinlichkeit, daß $k+1,\ldots,n \notin B$ sind. Also ist

$$\begin{aligned} f(k) &= 1 \Leftrightarrow \sum_{k+1\le l\le n} p(k,l) = 1 \Leftrightarrow \sum_{k+1\le l\le n} \frac{k}{l(l \Leftrightarrow 1)} \\ &= 1 \Leftrightarrow k \sum_{k+1\le l\le n} \left(\frac{1}{l \Leftrightarrow 1} \Leftrightarrow \frac{1}{l}\right) = 1 \Leftrightarrow k\left(\frac{1}{k} \Leftrightarrow \frac{1}{n}\right) = \frac{k}{n}. \end{aligned}$$

Wir greifen auf die Lineare Optimierung zur Berechnung der Gewinnfunktion G zurück.

$$\begin{aligned} \sum_{1\le k\le n} \quad & G(k) \to \min \quad \text{mit} \\ & G(k) \ge k/n \qquad 1 \le k \le n, \\ & G(k) \ge \sum_{k+1\le l\le n} \frac{k}{l(l \Leftrightarrow 1)} G(l) \quad 1 \le k \le n. \end{aligned}$$

Wir werden dieses Lineare Programm von Hand lösen. Wenn wir $G(k+1),\ldots,G(n)$ kennen, gibt es zwei Nebenbedingungen für $G(k)$, z.B. $G(k) \ge c_1$, $G(k) \ge c_2$, d.h. $G(k) \ge \max\{c_1, c_2\}$. Aus der Form der Nebenbedingungen folgt, daß wir alle $G(k)$ einzeln minimieren können.

Für $G(n)$ sind die Beschränkungen 1 und 0, also ist $G(n) = 1$. Für $G(n \Leftrightarrow 1)$ sind die Bedingungen $\frac{n-1}{n}$ und $\frac{1}{n}$. Da $n \ge 2$ (sonst gibt es den Zustand $n \Leftrightarrow 1$ nicht), ist $G(n \Leftrightarrow 1) = \frac{n-1}{n}$. Wir nehmen an, daß $G(l) = l/n$ für $k < l \le n$ gilt. Dann sind die Beschränkungen für $G(k)$

$$\begin{aligned} G(k) &\ge \frac{k}{n} \\ G(k) &\ge \sum_{k+1\le l\le n} \frac{k}{l(l \Leftrightarrow 1)} \frac{l}{n} = \frac{k}{n}\left(\frac{1}{k} + \ldots + \frac{1}{n \Leftrightarrow 1}\right). \end{aligned}$$

Es gilt also $G(k) = \frac{k}{n}$ genau dann, wenn $\frac{1}{k} + \cdots + \frac{1}{n-1} \le 1$ ist, und $G(k) > \frac{k}{n}$ sonst. Sei nun

$$k(n) := \max\left\{ k \,\middle|\, \frac{1}{k} + \ldots + \frac{1}{n \Leftrightarrow 1} > 1 \right\}.$$

Für $k \le k(n)$ gilt dann als dritte Bedingung

$$G(k) \ge \sum_{k+1\le l\le n} \frac{k}{l(l \Leftrightarrow 1)} G(l) \ge \sum_{k+1\le l\le n} \frac{k}{l(l \Leftrightarrow 1)} \frac{l}{n} \ge \frac{k}{n}\left(\frac{1}{k(n)} + \ldots + \frac{1}{n \Leftrightarrow 1}\right) > \frac{k}{n}.$$

Die Stützmenge ist also $\{k(n)+1,\ldots,n\}$, d.h. eine optimale Strategie schreibt vor, die ersten $k(n)$ Objekte nicht zu wählen und dann das erste Objekt, das besser als alle vorherigen ist, zu wählen. Die Erfolgswahrscheinlichkeit $p(n)$ dieser Strategie ist

$$\begin{aligned} p(n) &= \sum_{k(n)+1\le l\le n} P\{l \in B,\ l \Leftrightarrow 1 \notin B, \ldots,\ k(n)+1 \notin B\} \frac{l}{n} \\ &= \sum_{k(n)+1\le l\le n} \frac{k(n)}{l(l \Leftrightarrow 1)} \frac{l}{n} = \frac{k(n)}{n}\left(\frac{1}{k(n)} + \ldots + \frac{1}{n \Leftrightarrow 1}\right). \end{aligned}$$

Für jedes n kann $k(n)$ in linearer Zeit ausgerechnet werden. Wir wollen $k(n)$ und $p(n)$ für große n approximieren. Es gilt

$$\ln(m+1) \Leftrightarrow \ln(m) = \int_{m}^{m+1} \frac{1}{x} dx < \frac{1}{m} < \int_{m-1}^{m} \frac{1}{x} dx = \ln(m) \Leftrightarrow \ln(m \Leftrightarrow 1).$$

Wir summieren diese Ungleichungen von $k, \ldots, n \Leftrightarrow 1$ auf und erhalten

$$\ln(n) \Leftrightarrow \ln(k) < \frac{1}{k} + \ldots + \frac{1}{n \Leftrightarrow 1} < \ln(n \Leftrightarrow 1) \Leftrightarrow \ln(k \Leftrightarrow 1).$$

Der Parameter $k(n)$ ist durch

$$\frac{1}{k(n)+1} + \ldots + \frac{1}{n \Leftrightarrow 1} \leq 1 < \frac{1}{k(n)} + \ldots + \frac{1}{n \Leftrightarrow 1}$$

definiert. Also gilt

$$\ln(n) \Leftrightarrow \ln(k(n)+1) < 1 < \ln(n \Leftrightarrow 1) \Leftrightarrow \ln(k(n) \Leftrightarrow 1).$$

Wir leiten hieraus eine obere und untere Schranke für $k(n)$ ab. Es gilt

$$\ln(n) \Leftrightarrow 1 < \ln(k(n)+1), \quad \text{also } \frac{n}{e} \Leftrightarrow 1 < k(n), \text{ und}$$

$$\ln(k(n) \Leftrightarrow 1) < \ln(n \Leftrightarrow 1) \Leftrightarrow 1, \quad \text{also } k(n) < \frac{n \Leftrightarrow 1}{e} + 1 = \frac{n}{e} + \left(1 \Leftrightarrow \frac{1}{e}\right).$$

Insgesamt gilt

$$k(n) \in \left(\frac{n}{e} \Leftrightarrow 1, \frac{n}{e} + 1 \Leftrightarrow \frac{1}{e}\right),$$

dies ist ein Intervall, das maximal zwei ganze Zahlen enthält. Z.B. für $n = 800$ haben wir noch mehr Glück. Es gilt

$$k(800) \in (293.30\ldots, 294.94),$$

also $k(800) = 294$.

Wir kommen zur Abschätzung von $p(n)$. Es ist

$$\frac{k(n)}{n} \in \left(e^{-1} \Leftrightarrow \frac{1}{n}, e^{-1} + \frac{1 \Leftrightarrow e^{-1}}{n}\right), \text{ also}$$

$$\lim_{n\to\infty} \frac{k(n)}{n} = e^{-1}.$$

Dazu ist

$$\frac{1}{k(n)+1} + \ldots + \frac{1}{n \Leftrightarrow 1} \leq 1 < \frac{1}{k(n)} + \cdots + \frac{1}{n \Leftrightarrow 1}.$$

Da $k(n) \to \infty$, folgt

$$\lim_{n\to\infty} p(n) = \lim_{n\to\infty} \frac{k(n)}{n} \left(\frac{1}{k(n)} + \ldots + \frac{1}{n \Leftrightarrow 1}\right) = e^{-1} \approx 0.368.$$

8.10 Optimales Stoppen bei Strafkosten für das Warten

In vielen Problemen der Praxis entstehen beim Warten Kosten. Beim Stoppen in Zustand i erhalten wir die Belohnung $f(i) \geq 0$, beim Warten in Zustand i entstehen Kosten $C(i) \geq 0$, nur die Übergangswahrscheinlichkeiten $p(i,j)$ beim Warten sind interessant. In der momentanen Situation haben wir weder ein (EN)- noch ein (EP)-Problem. Wir nehmen an, daß $f^{(max)} := \sup\{f(i) \mid i \in S\}$ endlich ist. Wir können nur einmal eine Belohnung von höchstens $f^{(max)}$ erhalten. Wir ersetzen nun die Auszahlung beim Stoppen durch $f(i) \Leftrightarrow f^{(max)}$, dadurch sind nun alle Auszahlungen kleiner gleich 0. Für alle Strategien, die mit Wahrscheinlichkeit 1 stoppen, haben wir die Kosten um den konstanten Summanden $f^{(max)}$ verändert, d.h. die gleichen Strategien wie zuvor sind optimal. Was ist mit Strategien, die nicht mit Sicherheit stoppen? Unter der Annahme, daß $C := \inf\{C(i) \mid i \in S\} > 0$ ist, haben diese Strategien unendliche erwartete Kosten und sind niemals optimal. Nun sind wir offensichtlich im Fall (EN), d.h. die Optimalitätsgleichung gilt. Dies bedeutet

$$G(x) = \max\{\sum_y p(x,y) \cdot G(y) \Leftrightarrow C(x), f(x)\}.$$

Da wir im Fall (EN) sind, können wir folgendes beobachten:

1.) Der Aktionenraum bei unserem Problem ist endlich, also ist die Existenz einer optimalen Strategie garantiert.

2.) Da unser Problem ein stationäres Markoff–Problem ist, existiert eine optimale stationäre Markoff–Strategie.

Man überlegt sich leicht, daß jede stationäre Markoff–Strategie τ in unserem Problem durch eine Teilmenge $T \subseteq S$ beschrieben ist, für die gilt

$$\tau(s) = \tau_T(s) := \begin{cases} \text{STOP}, & \text{falls } s \in T, \\ \text{WEITER} & \text{sonst.} \end{cases}$$

Analog wie im Fall mit Wartekosten 0 kann man nun einen Operator γ definieren als

$$(\gamma v)(s) = \max\{\sum_y p(s,y) \cdot v(y) \Leftrightarrow C(s), f(s)\}.$$

Die Optimalitätsgleichung besagt $\gamma G = G$.

Wieder gilt, daß γ monoton ist und wieder können wir zu γ einen zugehörigen $\mathcal{L}im_\gamma$–Operator definieren.

Wir sind im Fall (EN) und daher gilt nun nicht unbedingt, daß wir G als Limes der $G_{n,k}$ berechnen können. Aber in unserem Fall können wir beobachten, daß folgendes gilt: Wenn wir den γ–Operator iteriert auf 0 anwenden, erhalten wir eine monoton fallende Folge von Vektoren. Wenn wir den γ–Operator iteriert auf f anwenden, erhalten wir eine monoton wachsende Folge von Vektoren. Dies kann man wieder induktiv nachweisen, auch daß der Limes existiert. Es gilt nun $G \leq \mathcal{L}im_\gamma 0$. Der Grund dafür ist der folgende. Wir wissen, daß $G \leq 0$ ist, somit ist $\gamma G \leq \gamma 0$, also $G \leq \gamma 0$. Es folgt induktiv $G \leq \mathcal{L}im_\gamma 0$. Analog kann man aus $G \geq f$ folgern, daß $G \geq \mathcal{L}im_\gamma f$ ist.

Wieder definieren wir als Stützmenge die Menge ST aller Zustände s, für die $G(s) = f(s)$ gilt. Wenn eine Strategie auf einem Zustand außerhalb der Stützmenge stoppt, so ist sie offensichtlich nicht optimal. Wir müssen nun also nur noch alle Strategien τ_T betrachten, die auf einer Teilmenge $T \subseteq ST$ der Stützmenge stoppen. Für eine Strategie τ_T definieren wir einen γ_T–Operator, um die erwartete Auszahlung G_{τ_T} zu beschreiben:

$$(\gamma_T v)(s) = \begin{cases} f(s), & \text{falls } s \in T, \\ \sum_y p(s,y) \cdot v(y) \Leftrightarrow C(s) & \text{sonst.} \end{cases}$$

Nach Lemma 8.2.8 errechnet sich G_{τ_T} als $\mathcal{L}im_{\gamma_T} 0$.

Wir halten auch fest, daß nach Definition für alle v die Eigenschaft $\gamma_T v \leq \gamma v$ gilt.

Lemma 8.10.1: *Sei $T_1 \subseteq T_2 \subseteq ST$. Für alle $i \geq 0$ gilt*

$$(\gamma_{T_1}^i)0 \leq (\gamma_{T_2}^i)0.$$

Beweis: Für $i = 0$ gilt es trivialerweise. Gelte es nun für alle $i \leq k$.

Sei $v := \gamma_{T_1}^k 0$ und $w := \gamma_{T_2}^k 0$. Wir wissen $v \leq w$ nach Induktionsvoraussetzung und $w \leq \gamma^k 0 \leq G$.

Für $s \notin ST$ gilt $(\gamma_{T_1} v)(s) \leq (\gamma_{T_1} w)(s) = (\gamma_{T_2} w)(s)$ nach Definition der Operatoren, ebenso für $s \in T_1$. Für $s \in T_2 \setminus T_1$ (also ist $s \in ST$) beobachten wir, daß $(\gamma_{T_2} w)(s) = f(s)$ und andererseits, daß $(\gamma_{T_1} w)(s) \leq (\gamma w)(s) \leq G(s) = f(s)$ ist. □

Damit ist die Strategie τ_{ST} als optimal nachgewiesen, denn mit einer Übertragung der Aussage des Lemmas auf die Limiten folgt $G_{\tau_{ST}} \geq G_{\tau_T}$ für alle $T \subseteq ST$. Wir bemerken am Rande, daß wir hier nicht die Endlichkeit der Zustandsmenge benötigt haben.

Man muß nun nur noch die Stützmenge berechnen. Wir betrachten einen Spezialfall, wo man das einfach tun kann.

Satz 8.10.2: *Es sei $B := \{b \mid (\gamma f)(b) = f(b)\}$. Ist $p(i,j) = 0$ für alle $i \in B$ und $j \notin B$, dann ist B die Stützmenge.*

Beweis: ... □

Wir wollen ein Beispiel diskutieren, in dem Satz 8.10.2 angewendet werden kann.

Beispiel 8.10.3: Wir wollen ein Haus verkaufen, für das wir täglich (im Durchschnitt) ein Angebot erhalten. Die Angebote sind identisch und unabhängig verteilt. Die Angebotshöhe ist $j \in \{0, \ldots, N\}$ mit Wahrscheinlichkeit p_j. Mit der Angebotshöhe 0 haben wir auch modelliert, kein Angebot zu bekommen. Wir wollen nun annehmen, daß wir tägliche Unterhaltskosten von c für das Haus haben und auf frühere Angebote zurückgreifen können. Es gilt also $R(i) = i$ und

$$\begin{aligned} p(i,j) &= 0 \text{ für } j < i \\ p(i,j) &= p_0 + \ldots + p_i \text{ für } j = i \text{ und} \\ p(i,j) &= p_j \text{ für } j > i. \end{aligned}$$

Nach Definition ist

$$\begin{aligned} B &= \left\{ i \,\middle|\, \Leftrightarrow i \leq c \Leftrightarrow i(p_0 + \ldots + p_i) \Leftrightarrow \sum_{j>i} j p_j \right\} \\ &= \left\{ i \,\middle|\, \sum_{j>i} j p_j \Leftrightarrow i \sum_{j>i} p_j \leq c \right\} = \left\{ i \,\middle|\, \sum_{j>i} (j \Leftrightarrow i) p_j \leq c \right\}. \end{aligned}$$

Da $\sum_{j>i} (j \Leftrightarrow i) p_j$ für wachsendes i fällt, ist $B = \{i^*, i^* + 1, \ldots, N\}$ für

$$i^* := \min \left\{ i \,\middle|\, \sum_{j>i} (j \Leftrightarrow i) p_j \leq c \right\}.$$

Also ist nach Definition ($p(i,j) > 0$ nur für $j \geq i$) $p(i,j) = 0$ für $i \in B$ und $j \notin B$. Also ist die Strategie, beim ersten Angebot, das mindestens i^* beträgt, zu akzeptieren, optimal. Dabei ist i^* effizient berechenbar.

Können wir ähnliche Beispiele erzielen, wenn wir nicht auf alte Angebote zurückgreifen dürfen? Die Menge B' für dieses Problem erfüllt im allgemeinen nicht die Voraussetzungen von Satz 8.10.2. Da unser Zustand i' nie größer ist als der erreichte Zustand i, falls auf alte Angebote zurückgegriffen werden kann, gilt für die minimalen erwarteten Kosten $G \leq G'$. Andererseits ist die Strategie, für das berechnete i^* das erste Angebot von mindestens i^* zu akzeptieren, weiterhin zulässig und erzielt weiterhin erwartete Kosten von G. D.h. $G = G'$, und in beiden Fällen ist die gleiche Strategie optimal.

In diesem Beispiel erkennen wir, mit welchem Feingefühl wir unsere Methoden anwenden müssen. Das zweite Problem können wir nicht direkt lösen. Wir lösen ein anderes Problem, nämlich das erste, das sich auf den ersten Blick entscheidend vom zweiten Problem unterscheidet, und übertragen dann die Lösung auf das zweite Problem.

8.11 Beispiele und Lösungsansätze für stationäre Modelle

Ein Problem in der Praxis bei stationären Problemen besteht in der Wahl des passenden Diskontfaktors β. Mit dieser Wahl wird die Gewichtung der Gegenwart, naher Zukunft und ferner Zukunft vorgenommen. Wir nehmen hier an, daß $\beta \in (0,1)$ gegeben ist und die Kostenfunktion C beschränkt ist. Auch hier ist es in vielen Beispielen üblicher, über Kosten statt über Belohnungen zu reden, $C(i,a)$ sind die Kosten in Zustand i bei Wahl der Aktion a. Aus der Beschränktheit von C folgt unmittelbar, daß wir im Fall (C) sind. Wir erinnern an Notationen. Dabei beschreibt $p(i,a,j)$ die zugehörige Übergangswahrscheinlichkeit, $G_\beta(i)$ die minimalen erwarteten Kosten bei Start in Zustand i und $G_f(i)$ die erwarteten Kosten bei Strategie f und Start in i. Wir können uns auf stationäre Strategien beschränken. Wenn der Aktionenraum endlich ist, lautet die Optimalitätsgleichung

$$G_\beta(i) = \min \left\{ C(i,a) + \beta \sum_{j \in S} p(i,a,j)\ G_\beta(j) \,\middle|\, a \in A \right\}.$$

Wir wissen, daß G_β die einzige Lösung der Optimalitätsgleichung ist, die der Kostenbeschränkung genügt. Bei Wahl einer Aktion, die die rechte Seite der Optimalitätsgleichung minimiert, erhalten wir eine optimale Strategie. Die Funktion G_β kann durch Wertiteration approximiert werden. Bevor wir zu konkreten Beispielen kommen, wollen wir ein Verfahren zur Verbesserung von Strategien diskutieren.

Zu stationären Strategien f betrachten wir den Operator T_f auf der Menge der beschränkten Funktionen $u : S \to \mathbb{R}$. Der Operator ist definiert durch

$$T_f u(i) := C(i, f(i)) + \beta \sum_{j \in S} p(i, f(i), j)\ u(j)$$

und beschreibt die erwarteten Kosten, wenn im ersten Schritt f angewendet wird und es danach zu den Schlußkosten u kommt. Die folgenden Eigenschaften von T_f sind einfach zu zeigen:

$$\begin{aligned} u \leq v &\Rightarrow T_f u \leq T_f v, \\ T_f\ G_f &= G_f, \\ T_f^n u &\to G_f \text{ für } n \to \infty. \end{aligned}$$

Definition 8.11.1: Zu einer stationären Strategie f sei f^* die Strategie, die im Zustand i eine Aktion a wählt, die den Ausdruck $C(i,a) + \beta \sum_{j \in S} p(i,a,j)\ G_f(j)$ minimiert.

Satz 8.11.2: *Die Strategie f^* ist mindestens so gut wie f. Wenn f^* nicht für irgendeinen Zustand besser als f ist, sind f und f^* optimal.*

Beweis: Nach Definition von f^* gilt

$$\begin{aligned} T_{f^*}\ G_f(i) &= C(i, f^*(i)) + \beta \sum_{j \in S} p(i, f^*(i), j)\ G_f(j) \\ &\leq C(i, f(i)) + \beta \sum_{j \in S} p(i, f(i), j)\ G_f(j) = G_f(i). \end{aligned}$$

Aus der Monotonie des T-Operators folgt induktiv

$$T^n_{f^*}\ G_f(i) \leq G_f(i).$$

Da die linke Seite gegen G_{f^*} konvergiert, ist f^* nicht schlechter als f.
Falls $G_{f^*} = G_f$ ist, folgt nach Definition von f^*

$$G_f(i) = G_{f^*}(i) = \min \left\{ C(i,a) + \beta \sum_{j \in S} p(i,a,j) G_f(j) \,\middle|\, a \in A \right\}.$$

Also ist G_f eine Lösung der Optimalitätsgleichung, die die übliche Nebenbedingung erfüllt. Damit sind f und f^* optimal. □

Satz 8.11.2 gibt einen heuristischen Algorithmus zur Konstruktion von guten oder sogar optimalen Strategien an. Wenn Zustands- und Aktionsraum endlich sind, ist die Menge der stationären Strategien und damit auch der Algorithmus endlich.

Unser erstes Beispiel beschäftigt sich mit dem besten Zeitpunkt der Erneuerung einer Maschine. Es sei $S = \{0, \ldots, N\}$ oder $S = \mathbb{N}_0$, die Menge der möglichen Zustände der betrachteten Maschine, wobei der Zustand $i+1$ „schlechter" als der Zustand i ist. Jeden Morgen wird der Zustand der Maschine festgestellt. Dann kann Aktion 1 (Erneuerung der Maschine) oder Aktion 2 (Beibehaltung der Maschine) durchgeführt werden. Da Zustand 0 der Zustand einer neuen Maschine ist, gilt für die Kosten D einer neuen Maschine

$$C(i,1) = D + c(0), \quad C(i,2) = c(i),$$

wobei $c(i)$ die Unterhaltkosten für eine Maschine im Zustand i angibt. Die Übergangswahrscheinlichkeiten kürzen wir ab durch $p(i,2,j) = p(i,j)$ und $p(i,1,j) = p(0,j)$. Die beiden folgenden Annahmen sind nach der Interpretation sinnvoll.

- Die Funktion c ist monoton wachsend und beschränkt.
- Die Funktion $i \to \sum_{k \leq j < \infty} p(i,j)$ ist für festes k wachsend in i. Je schlechter der Zustand, desto größer die Wahrscheinlichkeit, in die Klasse der schlechtesten Zustände zu kommen.

Zur Lösung dieses Problems benötigen wir ein technisches Lemma, dessen Beweis wir uns hier nicht ansehen wollen.

Lemma 8.11.3: *Sei $h : \mathbb{N}_0 \to \mathbb{R}$ monoton wachsend und beschränkt. Dann ist auch g monoton wachsend, wobei g definiert ist durch*

$$g(i) := \sum_{0 \leq j < \infty} p(i,j) h(j).$$

Nun können wir eine sehr intuitive Aussage beweisen. Je schlechter der Startzustand einer Maschine ist, desto höher die erwarteten Kosten.

Lemma 8.11.4: *$G_\beta(i)$ ist monoton wachsend in i.*

Beweis: Sei

$$\begin{aligned} G_\beta(i,1) &= \min\{D + c(0), c(i)\} \\ G_\beta(i,n) &= \min\left\{ D + c(0) + \beta \sum_{0 \le j < \infty} p(0,j)\ G_\beta(j, n \Leftrightarrow 1), \right. \\ &\qquad \left. c(i) + \beta \sum_{0 \le j < \infty} p(i,j)\ G_\beta(j, n \Leftrightarrow 1) \right\}. \end{aligned}$$

Nach Annahme 1 ist $G_\beta(i,1)$ monoton wachsend. Der erste Term auf der rechten Seite der Definition von $G_\beta(i,n)$ ist von i unabhängig, im zweiten Term ist der erste Summand nach Voraussetzung monoton wachsend, der zweite Summand ist nach Lemma 8.11.3 monoton wachsend, da $G_\beta(j, n \Leftrightarrow 1)$ nach Induktionsvoraussetzung monoton wächst. Damit ist $G_\beta(i,n)$ monoton wachsend. Da wir im Fall (C) auch im Fall (EP) sind, gelten die Aussagen über die Approximierbarkeit der Funktion G_β, d.h. $G_\beta(i,n) \to G_\beta(i)$ für $n \to \infty$. Also ist $G_\beta(i)$ monoton wachsend. □

Satz 8.11.5: *Es gibt ein $i^* \in \mathbb{N}_0 \cup \{\infty\}$, so daß die Strategie, die Maschine genau in den Zuständen $i > i^*$ zu erneuern, optimal ist.*

Beweis: Die Optimalitätsgleichung lautet

$$G_\beta(i) = \min\left\{ D + c(0) + \beta \sum_{0 \le j < \infty} p(0,j)\ G_\beta(j),\ c(i) + \beta \sum_{0 \le j < \infty} p(i,j)\ G_\beta(j) \right\}.$$

Der erste Term ist unabhängig von i, der zweite wiederum nach Lemma 8.11.3 monoton wachsend in i. Also ist die Menge der Zustände, für die der zweite Term kleiner als der erste Term ist, vom Typ $\{0, \ldots, i^*\}$ für $i^* \in \mathbb{N}_0 \cup \{\infty\}$. Da wir auch im Fall (EN) sind, ist die beschriebene Strategie optimal. □

Bei endlich vielen Zuständen, genügt es also, $|S|+1$ Strategien zu vergleichen. Zusätzlich kann die Methode der Strategieverbesserung eingesetzt werden.

In dem zweiten Beispiel der Qualitätskontrolle geben wir uns mit der Modellierung des Problems und der Aufstellung der Optimalitätsgleichung zufrieden. Die Maschine kann in Ordnung sein und produziert dann fehlerfreie Ware (bis auf den normalen Ausschuß), und die Maschine kann defekt sein und produziert dann fehlerhafte Ware. Durch die Kontrolle der Ware kann (bis auf einen zu vernachlässigenden statistischen Fehler) festgestellt werden, in welchem Zustand die Maschine ist. Die Kontrolle kostet I, die Kosten für die Erneuerung der Maschine betragen D, und der Verlust durch die Produktion schlechter Ware ist C. Aktion 1 sei die Kontrollaktion und Aktion 2 die Aktion, nicht zu kontrollieren. Der Zustand q gibt die Wahrscheinlichkeit an, mit der die Maschine defekt ist, und γ die Wahrscheinlichkeit, mit der eine funktionsfähige Maschine in einem Tag defekt wird. Der Zustandsraum ist zwar überabzählbar, aber es werden nur abzählbar viele Zustände angenommen. Wir beschreiben nun diese Parameter formal.

$C(q,1) = I + q(C+D)$, $C(q,2) = qC$, $p(q,1,\gamma) = 1$ (neue Maschine nach einem Tag wieder defekt), $p(q,2,q + (1 \Leftrightarrow q)\gamma) = 1$.

Die letzte Gleichung folgt nach dem Satz von der bedingten Wahrscheinlichkeit. Es gilt nämlich

$$\begin{aligned} P\{\text{defekt zu } t+1\} &= P\{\text{defekt zu } t+1 | \text{defekt zu } t\} P\{\text{defekt zu } t\} + \\ &\quad P\{\text{defekt zu } t+1 | \text{nicht defekt zu } t\} P\{\text{nicht defekt zu } t\} \\ &= 1 \cdot q + \gamma(1 \Leftrightarrow q). \end{aligned}$$

Damit lautet die Optimalitätsgleichung

$$G_\beta(q) = \min\left\{I + q(C+D) + \beta G_\beta(\gamma),\ qC + \beta G_\beta(q + (1 \Leftrightarrow q)\gamma)\right\}.$$

Jetzt kann die Methode der Strategieverbesserung angewendet werden.

Wir greifen nun auf das Beispiel des Hausverkaufs zurück (Beispiel 8.10.3). Wir nehmen an, daß nicht auf alte Angebote zurückgegriffen werden darf. Analoge Resultate lassen sich jedoch auch im anderen Fall erzielen. Die Optimalitätsgleichung lautet

$$G_\beta(i) = \min\left\{\Leftrightarrow i, c + \beta \sum_{0 \le j \le N} p_j G_\beta(j)\right\}.$$

Hier ist der zweite Term unabhängig von i. Die Strategie, alle Angebote, die mindestens

$$i^* := \min\left\{i \,\middle|\, \Leftrightarrow i < c + \beta \sum_{0 \le j \le N} p_j G_\beta(j)\right\}$$

betragen, zu akzeptieren, ist also optimal. Um i^* zu berechnen, berechnen wir für die Strategien f_i, alle Angebote von mindestens i zu akzeptieren, die erwarteten Kosten. Wenn f_i das Akzeptieren eines Angebots vorschreibt, beträgt das Angebot mindestens i. Die bedingte Wahrscheinlichkeit, daß die Angebotshöhe j ist, beträgt also $p_j/(p_i + \ldots + p_N)$ für $j \ge i$. Wenn zum Zeitpunkt T gestoppt wird, betragen die erwarteten Kosten also

$$c + \beta c + \ldots + \beta^{T-1} c \Leftrightarrow \beta^T \sum_{i \le j \le N} j p_j / \sum_{i \le j \le N} p_j.$$

Die ersten Summanden ergeben $c(1 \Leftrightarrow \beta^T)/(1 \Leftrightarrow \beta)$. Es sei $q := p_0 + \ldots + p_{i-1}$ die Wahrscheinlichkeit, nicht zu stoppen. Dann beträgt die Wahrscheinlichkeit, zum Zeitpunkt T zu stoppen, genau $q^T(1 \Leftrightarrow q)$, da zu den Zeitpunkten $0, \ldots, T \Leftrightarrow 1$ nicht gestoppt wird. Es gilt

$$E(\beta^T) = \sum_{0 \le t < \infty} \beta^t q^t (1 \Leftrightarrow q) = (1 \Leftrightarrow q)/(1 \Leftrightarrow \beta q).$$

Also ist

$$G_{f_i} = c\frac{1 \Leftrightarrow \frac{1-q}{1-\beta q}}{1 \Leftrightarrow \beta} \Leftrightarrow \frac{1 \Leftrightarrow q}{1 \Leftrightarrow \beta q}\frac{\sum_{i \le j \le N} j p_j}{1 \Leftrightarrow q} = \frac{1}{1 \Leftrightarrow \beta q}(cq \Leftrightarrow \sum_{i \le j \le N} j p_j) = \frac{c \sum_{0 \le j < i} p_j \Leftrightarrow \sum_{i \le j \le N} j p_j}{1 \Leftrightarrow \beta \sum_{0 \le j < i} p_j}.$$

Dieser Ausdruck ist bzgl. i zu minimieren.

Wir kehren zur allgemeinen Methode der Strategieverbesserung zurück. Innerhalb dieses Algorithmus muß G_f ausgerechnet werden. Für endliche Zustandsräume $S = \{0, \ldots, N\}$ gilt

$$G_f(i) = C(i, f(i)) + \beta \sum_{0 \le j \le N} p(i, f(i), j) G_f(j).$$

Analog zu der entsprechenden Aussage für G_β kann gezeigt werden, daß G_f die einzige Lösung dieser Gleichung ist. Also kann G_f durch Lösung dieses Linearen Gleichungssystems berechnet werden. Wenn wir G_β vorab kennen, können wir den Algorithmus der Strategieverbesserung vorzeitig abbrechen, wenn $||G_f \Leftrightarrow G_\beta||$ klein genug ist. Wir zeigen, wie wir G_β mit Hilfe der Linearen Optimierung berechnen können. Für beschränkte Abbildungen $u : S \to \mathbb{R}$ betrachten wir den Operator T_β definiert durch

$$T_\beta u(i) := \min\left\{C(i,a) + \beta \sum_{j \in S} p(i,a,j) u(j) \,\middle|\, a \in A\right\}$$

Aus der Optimalitätsgleichung folgt $T_\beta G_\beta = G_\beta$. Im Fall (C) folgt aus der Methode der Approximation von G_β, da $\beta^n ||u|| \to 0$ für $n \to \infty$, daß $T_\beta^n u \to G_\beta$ für $n \to \infty$ konvergiert.

Lemma 8.11.6: *Falls $T_\beta u \geq u$ ist, gilt $G_\beta \geq u$.*

Beweis: Da der Operator T_β monoton ist, folgt aus der Voraussetzung $T_\beta^n u \geq u$. Da $T_\beta^n u \to G_\beta$ für $n \to \infty$, folgt $G_\beta \geq u$. □

Da, wie oben gesehen, $T_\beta G_\beta = G_\beta$ ist, ist G_β die punktweise größte Funktion, die das Ungleichungssystem $T_\beta u \geq u$ erfüllt. Da wir in $T_\beta u(i) \geq u(i)$ die Minimumbedingung durch Bedingungen für alle Aktionen ersetzen können, erhalten wir folgendes Lineare Optimierungsproblem auf den Variablen $u(j),\ j \in S$, dessen Lösung G_β ist. Dabei ist $S = \{0, \ldots N\}$.

$$\sum_{0 \leq j \leq N} u(j) \to \text{ max, wobei}$$

$$C(i,a) + \beta \sum_{0 \leq j \leq N} p(i,a,j) u(j) \geq u(i) \quad \text{für } 0 \leq i \leq N,\ a \in A,$$

$$u(j) \leq \frac{1}{1 \Leftrightarrow \beta} ||C|| \quad \text{für } 0 \leq j \leq N.$$

Die letzte Nebenbedingung haben wir eingefügt, um für die Variable $u'(j) := u(j) \Leftrightarrow ||C||/(1 \Leftrightarrow \beta)$ die übliche Bedingung $u'(j) \leq 0$ zu erhalten. Die Bedingung ist korrekt, da die Kosten zum Zeitpunkt t durch $\beta^t ||C||$ nach oben beschränkt sind. Wir erhalten ein Lineares Programm mit $|S|$ Variablen und $|S|(|A|+1)$ Nebenbedingungen. Spätestens jetzt sieht man ein, daß 100 Nebenbedingungen keine utopische Zahl darstellen.

Literatur:

Aspvall, B. und Stone, R.E.: *Khachiyan's linear programming algorithm*, Journal of Algorithms 1, 1-13, 1980.

Burkard, R.E.: *Methoden der ganzzahligen Optimierung*, Springer, 1972

Dynkin, E.B. und Juschkewitsch, A.A.: *Sätze und Aufgaben über Markoffsche Prozesse*, Springer, 1969.

Hinderer K.: *Foundations of non-stationary dynamic programming with discrete time parameter*, Lecture Notes in Operations Research and Mathematical Systems 33, Springer, 1970.

Hu, T.C.: *Ganzzahlige Programmierung und Netzwerkflüsse*, Oldenbourg, 1972.

Muth, J.F. und Thompson, G.L.: *Industrial scheduling*, Prentice Hall, 1963.

Neumann: *Operations Research Verfahren* Bd I, II, III, Carl Hanser Verlag.

Owen, G.: *Spieltheorie*, Springer, 1971.

Reich, U.P.: *Die europäische Sicherheitskonferenz. Ein spieltheoretisches Experiment.* Carl Hanser Verlag, 1971.

Ross, S.M.: *Applied probability models with optimization applications*, Holden-Day, 1970.

Zimmermann, W.: *Operations Research*, Oldenbourg, 1990.