

Operations Research

**Skriptum zur Vorlesung im Herbsttrimester 2002 an der
Universität der Bundeswehr München**

Dr. Thomas F. Sturm

Infineon Technologies AG, Corporate Research

Ausgabe: 23

Datum: 19. Dezember 2002

Inhaltsverzeichnis

1	Einführung und Überblick	1
1.1	Der Begriff ‘Operations Research’	1
1.2	Modellierung und Methodik	1
1.3	Mathematische Optimierung	5
1.4	Graphentheorie, Netzplantechnik und Entscheidungsbaumverfahren	6
1.5	Lagerhaltung und Warteschlangen	6
1.6	Simulationsmethoden	6
1.7	Spieltheorie	9
2	Lineare Optimierung	15
2.1	Beispiele und Problemstellung	15
2.2	Eigenschaften der Problemstellung	17
2.3	Die (revidierte) Simplex-Methode	21
2.3.1	Vorbereitungen	21
2.3.2	Bestimmung der Multiplikatoren	22
2.3.3	Eckenaustausch	23
2.3.4	Behandlung entarteter Ecken	25
2.3.5	Zusammenfassung des Simplex-Algorithmus	25
2.3.6	Ermittlung einer zulässigen Startecke (Phase 1)	30
3	Unrestringierte nichtlineare Optimierungsprobleme	33
3.1	Mathematische Problemstellung	33
3.2	Optimalitätskriterien	35
3.3	Numerische Verfahren (Iterationsverfahren)	40
3.3.1	Modellbildung	40
3.3.2	Zweiphasige Optimierung	42
3.3.3	Line-Search-Methoden	42
3.3.4	(Lokale) Konvergenzgeschwindigkeit	45

4	Newton-Methode, Quasi-Newton-Methoden und Newton-ähnliche Methoden	47
4.1	Methode des steilsten Abstiegs (Steepest Descent)	47
4.2	Newton-Methode	49
4.3	Quasi-Newton-Methoden	51
4.4	BFGS-Methode	53
4.5	BNS-Methode	55
4.6	Trust-Region-Methode	59
Literaturverzeichnis		I

Kapitel 1

Einführung und Überblick

1.1 Der Begriff ‘Operations Research’

Mit dem Namen ‘Operational Research’ wurde erstmals 1937 in England eine Gruppe von Wissenschaftlern der britischen Streitkräfte bezeichnet, die den günstigsten Einsatz von Radargeräten für Luftwarnsysteme berechnen sollte. Während des zweiten Weltkriegs wurden Methoden des ‘Operations Research’ (US-amerikanisch) von den Alliierten u.a. für optimale Strategien im Luftkampf und für die optimale Zusammenstellung von Geleitzügen verwendet, [MM73, Nie84, Run86, Zim92].

Nach dem zweiten Weltkrieg wurden mit dem Aufkommen der elektronischen Datenverarbeitung die Verfahren des OR in der Wirtschaft und an Universitäten verfeinert und in zahllosen zivilen Bereichen angewendet. Die Vorstellungen aber darüber, was Operations Research ist, gehen weit auseinander.

Typische Eigenschaften des OR, zu deutsch auch Unternehmensforschung oder Optimalplanung genannt, sind:

1. Es werden Entscheidungen für Entscheidungsprobleme vorbereitet.
2. Dabei handelt es sich um ‘optimale’ Entscheidungen.
3. Es werden mathematische Methoden eingesetzt.

Im weiteren betrachten wir also die Anwendung von mathematischen Methoden zur Vorbereitung optimaler Entscheidungen. In der Regel werden diese Entscheidungen mittels numerischer Programme berechnet.

1.2 Modellierung und Methodik

Eine allgemeine schematische Vorgehensweise des OR ist in Abbildung 1.1 dargestellt. Ein zu lösendes Problem muss dazu in ein mathematisches Problem übertragen werden.

- Das Realproblem ist das zur Entscheidung anstehende Problem der Realität, z.B. Maximierung des Gewinns.

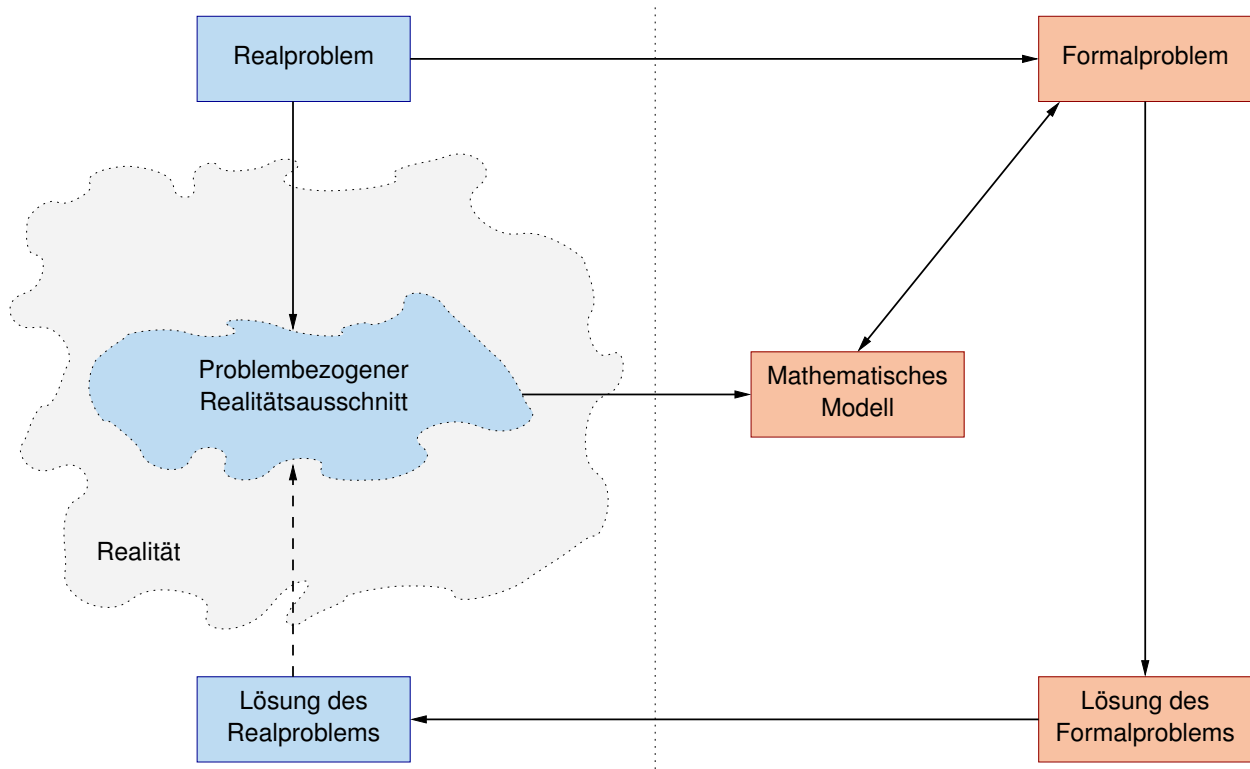


Abbildung 1.1: Schematische Vorgehensweise (nach [MM73])

- Auswahl eines problembezogenen Ausschnitts der Realität, d.h. möglichst aller für das Problem relevanten Realitätsbestände in Abhängigkeit der möglichen **Entscheidungsalternativen**.
- Formulierung eines mathematischen Modells, in dem die problemrelevanten Tatbestände und ihre Zusammenhänge idealisiert beschrieben werden.
- Übertragung der Fragestellung des Realproblems auf das Modell durch Beschreibung als Formalproblem, z.B. Maximierung einer bestimmten Variablen. Zu einem Modell kann es verschiedene Formalprobleme geben (z.B. Maximierung des Gewinns oder des Umsatzes).
- Numerische Berechnung der Lösung(en) des Formalproblems.
- Rückübertragung als Lösung des Realproblems, d.h. Angabe der optimalen Entscheidungen.

Kritisch für den Erfolg der Methodik ist

- die Erfassung aller relevanten Realitätsbestände und ihrer tatsächlichen Zusammenhänge (oft durch ein interdisziplinäres Team) und
- ihre formale Beschreibung in einem mathematischen Modell.

Häufig werden leider Einschränkungen bei der Modellbeschreibung in Kauf genommen, um bekannte Lösungsverfahren anwenden zu können ('Methodenorientierung'), anstatt gegebenenfalls neue Verfahren zu entwickeln ('Problemorientierung').

Beispiel 1.1 (Auslegungsoptimierung (eines Kraftwerks))

Typisch für die **Entwurfsphase** einer industriellen Anlage ist die folgende Problemstellung:

Die Grundstruktur der Anlage ist festgelegt, aber es sind noch Entscheidungen über die Feinstruktur zu treffen, z.B. die Auswahl von Komponenten aus einem Vorrat potentiell einsetzbarer Anlagenkomponenten, und über die Auslegungswerte der technischen Parameter dieser Komponenten.

Im Beispiel der Verschaltungsstruktur des Wärmeschaltplans (Abbildung 1.2) eines fossil befeuerten Kraftwerks sind die wesentlichen Komponenten:

- die **Feuerung**, die überhitzten und unter starkem Druck stehenden Dampf erzeugt ('Frischdampf').
- die **Turbine(n)**, wo sich der Dampf entspannt und dabei die Turbinenschaufeln in schnelle Rotation versetzt. Über eine gemeinsame Achse wird so der Generator angetrieben und elektrischer Strom erzeugt.
- der **Kondensator**, in dem der entspannte Dampf wieder zu Wasser kondensiert wird, das erneut in die Feuerung geleitet wird.

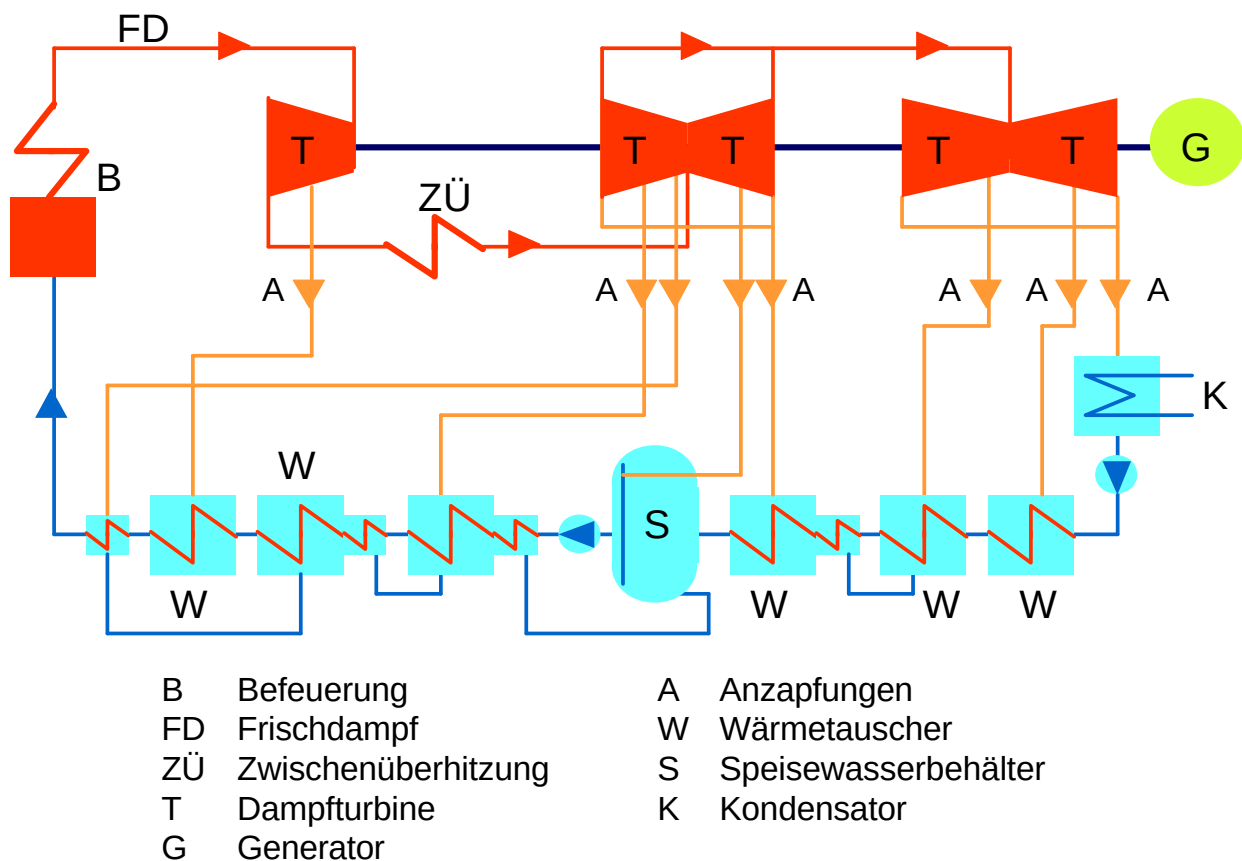


Abbildung 1.2: Optimale Auslegung der Vorwärmstrecke in Kraftwerken

Diese Hauptstruktur des Kraftwerks heißt Wasser-Dampf-Kreislauf. Zur Erhöhung des Wirkungsgrads zapft man zusätzlich an verschiedenen Stellen der Turbine(n) teilentspannten Dampf ab, der über Wärmetauscher zur Vorwärmung des Wassers genutzt wird ('Speisewasser-Vorwärmstrecke'). Dies erhöht zwar den Wirkungsgrad, verursacht aber hohe Investivkosten für Anzapfungen und Wärmetauscher. Der bestmögliche Abgleich solcher konkurrierenden Effekte führt auf ein Optimierungsproblem:

• **(Entscheidungs-)Variable:**

a) Strukturentscheidungen:

- wieviele Anzapfungen?
- welche Subkomponenten pro Anzapfung?
- Subkomponenten als getrennte Apparate?
- wenn getrennt: **wo** eingehängt in der Verschaltung?

Jeder dieser Strukturentscheidungen lässt sich durch eine ganze Zahl codieren; bei k Entscheidungen ergibt sich ein Entscheidungsvektor $d \in \mathbb{Z}^k$.

b) reellwertige (Auslegungs-)Variable:

- pro Anzapfung: Anzapfdruck p_i ist zu wählen.
bei n Anzapfungen: $p \in \mathbb{R}^n$.
- pro Wärmetauscher: Grädigkeit δ_j ist zu wählen.
bei m Wärmetauschern: Grädigkeitsvektor $\delta \in \mathbb{R}^m$.

• **Zielfunktion:**

Die Auslegungsentscheidung soll die (betriebswirtschaftlichen) Gesamtkosten für die Erzeugung einer kWh Strom minimieren ('Stromerzeugungskosten').

$$F(d, p, \delta) = \text{Investkostenanteil} + \text{Brennstoffanteil}$$

$$= \frac{\overbrace{\text{KostenAnlage}(d, p, \delta) \cdot \text{Annuität}}^{\text{auf ein Jahr umgelegte Investkosten}}}{\underbrace{P_{\text{elektr}}(d, p, \delta) \cdot h_{\text{Jahr}}}_{\text{jährliche erzeugte Nettoleistung}}} + \frac{\overbrace{b}^{\text{Brennstoffpreis pro kWh zugeführter Wärme}}}{\underbrace{\eta(d, p, \delta)}_{\text{Wirkungsgrad}}}.$$

• **Nebenbedingungen:**

a) für Strukturentscheidungen:

Sehr viele technische Randbedingungen.

b) für reellwertige Variable:

Aus physikalischen Gründen müssen die Anzapfdrucke streng geordnet sein:

$$p_1 < p_2 < \dots < p_n.$$

Unter Berücksichtigung eines Mindest-Druckabstandes Δp ergeben sich folgende **Ungleichungsnebenbedingungen**:

$$\begin{aligned} p_1 - p_2 + \Delta p &\leq 0, \\ &\vdots \leq \vdots, \\ p_{n-1} - p_n + \Delta p &\leq 0. \end{aligned}$$

Auch die Grädigkeiten δ_j müssen innerhalb technischer Grenzen gewählt werden:

$$\delta_j^{\text{unten}} \leq \delta_j \leq \delta_j^{\text{oben}}.$$

Somit ist das **gemischt diskret-kontinuierliche Optimierungsproblem**

$$\min_{d,p,\delta} \left\{ F(d,p,\delta) \mid \text{Nebenbedingungen an } d,p,\delta \right\}$$

zu behandeln. └─┘

Teilgebiete des Operations Research:

Im beschränkten Rahmen der Vorlesung können viele Bereiche des OR nur gestreift werden. Im Folgenden wird daher eine Auswahl von Themenfeldern kurz vorgestellt. Später werden wir dann insbesondere lineare und nichtlineare Optimierungsprobleme näher betrachten.

1.3 Mathematische Optimierung

Die grundsätzliche Aufgabe der mathematischen Optimierung besteht in der ‘optimalen’ Auswahl eines Elements oder einer Untermenge von Elementen aus einer gegebenen Menge anhand eines oder mehrerer Bewertungskriterien.

Man betrachtet eine Abbildung $F : R \rightarrow \mathbb{R}^k$, $k \in \mathbb{N}$, auf einem **zulässigen Bereich** $R \subseteq \mathbb{R}^n$, $n \in \mathbb{N}$.

$$F(x) = \begin{pmatrix} F_1(x) \\ \vdots \\ F_k(x) \end{pmatrix}, \quad x \in R, \quad F_i : R \rightarrow \mathbb{R}, \quad i = 1, \dots, k.$$

Jede Abbildung F_i , $i = 1, \dots, k$, heißt **Zielfunktion**.

In der **skalaren Optimierung** ($k = 1$) werden Minima von F gesucht, während in der **Mehrzieloptimierung / Vektoroptimierung** ($k > 1$) konkurrierende Zielfunktionen abgeglichen werden müssen (‘Pareto-Optimalität’).

$$\begin{aligned} k = 1 : & \quad \min_x \left\{ F(x) \mid x \in R \right\} \\ k > 1 : & \quad \text{„min“}_x \left\{ F(x) \mid x \in R \right\} \end{aligned}$$

Ist $R = \mathbb{R}^n$, so spricht man von **kontinuierlicher unrestringierter Optimierung (Optimierung ohne Nebenbedingungen)**.

Sind $h_i : \mathbb{R}^n \rightarrow \mathbb{R}$, $i = 1, \dots, m+p$, Funktionen mit

$$R = \left\{ x \in \mathbb{R}^n \mid h_i(x) \leq 0, i = 1, \dots, m, h_i(x) = 0, i = m+1, \dots, m+p \right\} \subseteq \mathbb{R}^n,$$

so heißen die h_i **Nebenbedingungen** und man spricht von **kontinuierlicher restringierter Optimierung (Optimierung mit Nebenbedingungen)**.

In beiden Fällen nennt man R einen **kontinuierlichen zulässigen Bereich**.

Enthält $R \subset \mathbb{R}^n$ nur endlich viele oder abzählbar unendlich viele Elemente, so spricht man von **diskreter Optimierung** oder **kombinatorischer Optimierung**, und R wird **diskreter zulässiger Bereich** genannt.

Im Fall $R = R_1 \times R_2$, wobei $R_1 \subseteq \mathbb{R}^{n_1}$ ein kontinuierlicher zulässiger Bereich ist und $R_2 \subseteq \mathbb{R}^{n_2}$ ein diskreter zulässiger Bereich ist, spricht man von **gemischt diskret-kontinuierlicher Optimierung** (R heißt gemischt diskret-kontinuierlicher zulässiger Bereich), siehe Beispiel 1.1.

Wichtige Spezialfälle: Man betrachte ein skalares kontinuierliches restringiertes Optimierungsproblem mit linearen Nebenbedingungen h_i .

Ist F linear, so spricht man von **linearer Optimierung**.

Ist F quadratisch, so spricht man von **quadratischer Optimierung**.

Diese Spezialfälle treten vor allem bei wirtschaftsmathematischen Fragestellungen auf. In allen anderen Fällen spricht man von **nichtlinearer Optimierung** (z.B. bei Industrieanwendungen und physikalischen Problemen).

In späteren Kapiteln werden wir hauptsächlich die lineare Optimierung und die (skalare kontinuierliche) unrestringierte nichtlineare Optimierung vertiefen.

1.4 Graphentheorie, Netzplantechnik und Entscheidungsbaumverfahren

Bei dieser Gruppe von Problemen und Lösungsmethoden des OR ist die mathematische Theorie der Graphen das benötigte Rüstzeug. Behandelt werden damit Probleme zur Bestimmung kürzester Wege, Transportaufgaben, das Traveling Salesman Problem, Zeit- und Kostenplanungen bei Projekten und vieles mehr.

Diese Thematik wird in der Vorlesung ‘Projektmanagement’ [Sch02] behandelt und kann ausführlich z.B. in [Neu75b], aber auch in [MM73, Ker87, Nie84, Zim92, Run86, Run89] nachgelesen werden.

1.5 Lagerhaltung und Warteschlangen

Bei Lagerhaltungssystemen ist die zentrale Frage, wann und wieviel bestellt werden soll unter Beachtung von Angebot, Nachfrage, Lieferzeit, Lagerkosten, Lagerkapazität, etc. Bei Warteschlangensystemen befasst man sich mit der optimalen Auslegung und Steuerung von Warteprozessen, bei denen unregelmäßig Aufgaben bei einem Verarbeitungssystem eintreffen und sich bis zur Bearbeitung in eine Warteschlange einfügen müssen. Ein Beispiel ist der Bedienungsschalter eines Kundendienstes, bei dem z.B. eine kostenminimale Besetzungszeit berechnet werden soll ohne Bedingungen an die Wartezeit der Kunden zu verletzen. Von großer Bedeutung sind Warteschlangensysteme beim Entwurf von Betriebssystemen (Prozess-Bearbeitung) und anderen Informatik-Aufgaben.

Probleme dieser Art werden i.d.R. durch stochastische Modelle beschrieben und durch stochastische Verfahren behandelt. Hierbei spielt die Theorie der Markov-Ketten eine große Rolle. Zur Vertiefung sei auf [MM73, Nie84, Zim92, Run89, Neu77] verwiesen.

1.6 Simulationenmethoden

Simulationenmethoden lassen sich auffassen als zielgerichtetes Experimentieren an Modellen, wobei insbesondere das Verhalten der Realität (Aktionen und Reaktionen) nachgeahmt wird. Im Allgemeinen kann ein Modell, mit welchem experimentiert wird, auch physischer Art sein (z.B. bei Windkanalversuchen), aber im engeren Sinn werden mathematische Modelle betrachtet, die insbesondere auch als Computerprogramme (Simulatoren) gegeben sein können.

Simulationen lassen sich auch deterministisch durchführen (z.B. beim Bau verschiedenartiger physischer Modelle), aber in der Regel versteht man Simulationen als stochastische Experimente, d.h. zufallsgesteuerte Experimente. Man spricht hier von **Monte-Carlo-Simulationen**.

Bekannte Beispiele für mit Monte-Carlo-Simulationen zu behandelnde Probleme sind

- die Berechnung des Neutronendurchgangs durch eine Platte¹ z.B. für die Dimensionierung des Schutzschildes von Kernkraftwerken,
- Zuverlässigkeitssimulationen von technischen Systemen mittels sog. **beschleunigter Lebensdauerversuche** (Ausfallphysik),
- Lagerhaltungs- und Warteschlangenprobleme (siehe Abschnitt 1.5).

Monte-Carlo-Simulationen lassen sich auch auf Probleme anwenden, die ihrer Natur nach nicht stochastisch sind. Das wichtigste Beispiel ist hierbei die Berechnung bestimmter Integrale, auch **Monte-Carlo-Integration** genannt.

Aufgabe:

Bei einer gegebenen Funktion

$$f : [a, b] \rightarrow \mathbb{R}, \quad a, b \in \mathbb{R}, \quad a < b,$$

ist (im Fall der Existenz) das Integral

$$I = \int_a^b f(x) \, dx$$

gesucht.

Variante 1:

Als ein Beispiel zur Integration mittels der Monte-Carlo-Vorgehensweise betrachten wir eine stetige Zufallsvariable Z , deren Realisierung in $[a, b]$ liegen sollen, und deren Wahrscheinlichkeitsdichtefunktion $g : [a, b] \rightarrow \mathbb{R}$ sei. Im einfachsten Beispiel kann man Z als gleichverteilt auf $[a, b]$ annehmen, so dass dann $g(x) = \frac{1}{b-a}$ für alle $x \in [a, b]$ ist.

Weiter definieren wir die Zufallsvariable Y durch

$$h(x) := \frac{f(x)}{g(x)}, \quad Y := h(Z) = \frac{f(Z)}{g(Z)}.$$

Dann gilt für den Erwartungswert² von Y , dass

$$\mathbf{E}(Y) = \mathbf{E}(h(Z)) = \int_a^b g(x) \cdot h(x) \, dx = \int_a^b g(x) \cdot \frac{f(x)}{g(x)} \, dx = \int_a^b f(x) \, dx = I.$$

¹Tatsächlich wurde die Monte-Carlo-Methode in den vierziger Jahren im Zusammenhang mit Problemen der Neutronenphysik entwickelt. Als Väter der Methode gelten u.a. von Neumann, Ulam und der Physiker Enrico Fermi.

²Lediglich zur Vereinfachung haben wir bei der Umformung das Riemann-Integral verwendet. Zweifelsfälle bei $g(x) = 0$ treten bei Verwendung des Lebesgue-Integrals nicht auf, so dass die Allgemeinheit der Umformung gesichert ist.

Jetzt simulieren wir Y durch Simulation von Z , d.h. wir müssen in der Lage sein, unabhängige Realisierungen $z_1, z_2, \dots, z_n$ (Zufallszahlen) gemäß der Dichte g zu erzeugen. Bei $n \in \mathbb{N}$ Versuchen ist also jeweils

$$y_k := h(z_k) = \frac{f(z_k)}{g(z_k)}, \quad k = 1, \dots, n.$$

Im einfachsten Fall der Gleichverteilung haben wir $y_k = \frac{1}{b-a} f(z_k)$ für $k = 1, \dots, n$.

Als Näherung für I erhalten wir dann

$$I \approx \frac{1}{n} \sum_{k=1}^n y_k.$$

Jetzt stellt sich natürlich die Frage nach der **Genauigkeit** dieser Näherung.

Da die Voraussetzungen des **zentralen Grenzwertsatzes** [SS94] erfüllt sind und damit $\frac{1}{n} \sum_{k=1}^n y_k$ für große n als Realisierung einer normalverteilten Zufallsvariable angesehen werden kann, gilt für große n :

$$P \left(\left| \frac{1}{n} \sum_{k=1}^n y_k - I \right| < 3 \sqrt{\frac{\text{Var}(Y)}{n}} \right) > 0.99$$

Allgemein gilt, dass um die Genauigkeit um den Faktor 10 zu verbessern (d.h. um eine Dezimalstelle zu gewinnen), die Anzahl der Versuche um den Faktor 100 vergrößert werden muss.

Es stellt sich die Frage, ob man durch geeignete Wahl von g die Varianz $\text{Var}(Y)$ möglichst klein machen kann.

Tatsächlich ist

$$g(x) := c \cdot |f(x)|, \quad x \in [a, b], \quad c = \frac{1}{\int_a^b |f(x)| dx}$$

unter allen möglichen Wahlen für g diejenige mit der minimalen Varianz $\text{Var}(Y)$ (Beweis siehe [Buc02]).

Da die Bestimmung des Faktors c nicht einfacher ist als die ursprüngliche Aufgabenstellung, wird man sich darauf beschränken, eine Dichte zu wählen, für die Zufallszahlen einfach auszulösen sind und die $|f|$ möglichst ‘ähnlich’ ist.

Variante 2:

Ein alternative Monte-Carlo-Verfahren zur Lösung betrachtet die Realisierungen einer auf $[a, b]$ gleichverteilten Zufallsvariable U und einer auf $[0, c]$ gleichverteilten Zufallsvariable V , wobei als Zusatzannahme $0 \leq f(x) \leq c$ für $x \in [a, b]$ betrachtet wird.

Die Realisierungen $(u_1, v_1), (u_2, v_2), \dots, (u_n, v_n)$ sind gleichverteilte Zufallspunkte im Quader $[a, b] \times [0, c]$. Sei nun hierfür z die Anzahl aller Punkte (u_k, v_k) für $k = 1, \dots, n$, für die gilt $v_k < f(u_k)$. Dann gilt (vergleiche Abbildung 1.3):

$$I \approx \underbrace{\frac{z}{n}}_{\text{relative Häufigkeit für } v_k < f(u_k)} \underbrace{c(b-a)}_{\text{Volumen des Quaders}}.$$

Auch für diese Lösungsvariante lassen sich Genauigkeitsabschätzungen angeben.

Die genannten Simulationsmethoden lassen sich auch für höherdimensionale Integrale anwenden. Monte-Carlo-Verfahren eignen sich nicht zur Erzeugung hoher Genauigkeiten, aber im Gegensatz zu deterministischen numerischen Verfahren wächst der Aufwand nicht exponentiell mit der Dimension, sondern oft nur polynomial oder sogar nur linear.

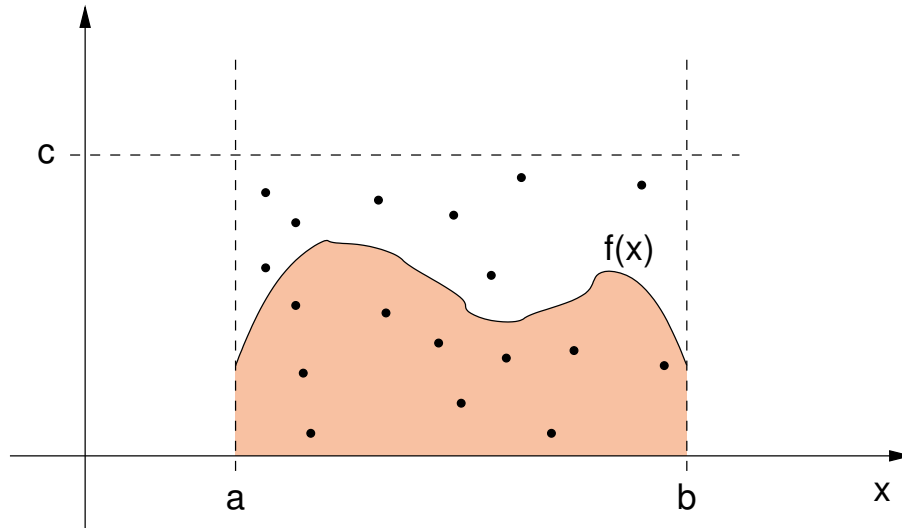


Abbildung 1.3: Monte-Carlo-Integration

Weitere Varianten der Monte-Carlo-Integration und zahlreiche sonstige Simulationen sind z.B. in [Buc02, PZ78] zu finden und werden auch in [Neu77, MM73, Nie84, Run89] behandelt. Die Frage, wie man überhaupt Zufallszahlen erzeugt, ist insbesondere in [Knu97, Zie78] zu finden.

1.7 Spieltheorie

In der Optimierung versucht **ein** Akteur ein oder mehrere Ziele durch optimale Entscheidungen zu erreichen. In der **Spieltheorie** werden Probleme behandelt, bei denen **mehrere** Akteure konkurrierend oder kooperierend Ziele durch optimale Entscheidungen erreichen wollen. Man betrachtet die Akteure als **Spieler** in einem **Spiel**, die i.d.R. ihre jeweiligen Gewinne maximieren wollen.

Die Praxisrelevanz spieltheoretischer Probleme ist umstritten, da ‘Spiele’ in der Wirtschaft oder militärstrategischen Planungen sehr komplex sind und psychologische Einflüsse oder externe Ereignisse die ‘Spielregeln’ variieren oder auf den Kopf stellen. Durch eine präzise Beschreibung des Verhaltens der einzelnen Spieler schult die Spieltheorie allerdings das Verständnis für ökonomische, soziologische und andere Zusammenhänge.

‘Die Spieltheorie ist die Wissenschaft vom Strategischen Denken’.

Das bekannteste (aber längst nicht einzige) Problem der Spieltheorie ist das **Zweipersonen-Nullsummenspiel**. Zwei Personen spielen gegeneinander in einer Folge von Partien, wobei jeweils Spieler 1 eine **Strategie** $x \in R_X$ und Spieler 2 eine Strategie $y \in R_Y$ wählt (R_X und R_Y sind die Mengen der zulässigen Strategien). Über eine Abbildung

$$E : R_X \times R_Y \rightarrow \mathbb{R},$$

ist eine feste **Auszahlungsvorschrift** gegeben, d.h. pro Partie wird $E(x,y)$ von Spieler 2 an Spieler 1 gezahlt ($E(x,y)$ kann natürlich auch negativ sein). Jeder Spieler versucht seine Strategien so zu optimieren, dass er den höchsten Gewinn davonträgt.

Beispiel 1.2 ([Neu75a])

Beim bekannten Papier-Stein-Schere-Knobeln lässt sich die Auszahlungsvorschrift wie folgt angeben:

1 \ 2	Papier	Stein	Schere
Papier	0	1	-1
Stein	-1	0	+1
Scherer	+1	-1	0

Beispiel 1.3

Die beiden Spieler haben je drei verschiedenfarbige Karten, bei denen pro Partie eine aufgedeckt wird. Die Auszahlungsvorschrift sei dabei wie folgt angegeben:

1 \ 2	Grün	Weiß	Lila
Schwarz	3	4	-6
Blau	-1	2	0
Rot	-1	-1	4

Bei dieser Art der Nullsummenspiele ist die Auszahlungsvorschrift durch eine Auszahlungsmatrix $A \in \mathbb{R}^{n,m}$ bestimmt (Matrixspiele).

Im Beispiel 1.3 etwa ist

$$A = \begin{pmatrix} 3 & 4 & -6 \\ -1 & 2 & 0 \\ -1 & -1 & 4 \end{pmatrix}$$

Wählt z.B. Spieler 1 Rot und Spieler 2 Weiß, so beträgt die Auszahlung

$$A_{3,2} = \underbrace{e_3^\top}_{=:x^\top} A \underbrace{e_2}_{=:y}.$$

Die Auszahlung ist also allgemein $E(x,y) := x^\top Ay$.

Man spricht von einer **reinen Strategie**, wenn sich jeder Spieler fest für eine Wahl entscheidet, d.h. $x = e_i$ für ein $i \in \{1, \dots, n\}$ und $y = e_j$ für ein $j \in \{1, \dots, m\}$.

Von einer **gemischten Strategie** wird gesprochen, wenn sich die Spieler jeweils mit einer gewissen Wahrscheinlichkeit für eine Strategie entscheiden, z.B. wenn Spieler 1 mit 40% Wahrscheinlichkeit die Strategie e_1 wählt und mit 60% Wahrscheinlichkeit für die Strategie e_2 , denn dann gilt für die Auszahlung

$$\frac{4}{10} \cdot e_1^\top Ay + \frac{6}{10} e_2^\top Ay = \underbrace{\left(\frac{4}{10}, \frac{6}{10}, 0, \dots, 0 \right)}_{=:x^\top} Ay$$

Allgemein kann also Spieler 1 aus der (gemischten) Strategiemenge

$$R_X := \left\{ x \in \mathbb{R}^n \mid x_i \geq 0, i = 1, \dots, n, \sum_{i=1}^n x_i = 1 \right\} \quad (1.1)$$

wählen und Spieler 2 aus der (gemischten) Strategiemenge

$$R_Y := \left\{ y \in \mathbb{R}^m \mid y_j \geq 0, j = 1, \dots, m, \sum_{j=1}^m y_j = 1 \right\}. \quad (1.2)$$

Bei der Definition **optimaler Strategien** wird davon ausgegangen, dass der jeweilige Gegenspieler seinerseits optimal spielt³. Eine Strategie wird als optimal angesehen, wenn sie den **sicheren Mindestgewinn** maximiert.

Somit heißt $x^* \in R_X$ optimal, wenn gilt

$$\min_{y \in R_Y} E(x^*, y) = \max_{x \in R_X} \underbrace{\min_{y \in R_Y} E(x, y)}_{\text{Mindestgewinn von Spieler 1 bei Wahl von } x} \quad (1.3)$$

$y^* \in R_Y$ heißt optimal, wenn gilt

$$\max_{x \in R_X} E(x, y^*) = \min_{y \in R_Y} \underbrace{\max_{x \in R_X} E(x, y)}_{\text{Mindestgewinn von Spieler 2 bei Wahl von } y} \quad (1.4)$$

Satz 1.4 (Hauptsatz der Spieltheorie f. Zweip.-Nullsummenspiele (Minimax-Theorem))

Für jedes Matrixspiel mit einer Auszahlungsfunktion

$$E : R_X \times R_Y \rightarrow \mathbb{R}, \quad (x, y) \mapsto x^\top A y,$$

mit $A \in \mathbb{R}^{n,m}$, $n, m \in \mathbb{N}$ und R_X wie in (1.1), R_Y wie in (1.2), gilt:

Es gibt ein $x^* \in R_X$ und ein $y^* \in R_Y$ mit

$$E(x^*, y^*) = \max_{x \in R_X} \min_{y \in R_Y} E(x, y) = \min_{y \in R_Y} \max_{x \in R_X} E(x, y). \quad (1.5)$$

$E(x^*, y^*)$ heißt dann auch **Wert des Spiels**. —

Beweis. Ohne Einschränkung der Allgemeinheit können die Komponenten von A als positiv angenommen werden, denn anderenfalls betrachtet man eine Konstante

$$\alpha > - \min_{i=1, \dots, n, j=1, \dots, m} A_{i,j}$$

und definiert eine Matrix $B \in \mathbb{R}^{n,m}$ durch $B_{i,j} := A_{i,j} + \alpha$ für alle $i = 1, \dots, n$, $j = 1, \dots, m$, d.h. alle Komponenten von B sind dann positiv. Definiert man $\hat{E}(x, y) := x^\top B y$ für $x \in \mathbb{R}^n$ und $y \in \mathbb{R}^m$, so folgt

$$\hat{E}(x, y) = \sum_{i=1}^n \sum_{j=1}^m x_i B_{i,j} y_j = \sum_{i=1}^n \sum_{j=1}^m x_i (A_{i,j} + \alpha) y_j = E(x, y) + \alpha \sum_{i=1}^n x_i \sum_{j=1}^m y_j = E(x, y) + \alpha.$$

Da die Verschiebung um eine Konstante an den Aussagen und Strategien nichts ändert, nehmen wir im Folgenden an, dass alle Komponenten von A positiv sind.

Man betrachte nun

$$h(x) := \min_{j \in \{1, \dots, m\}} x^\top A e_j, \quad \text{für } x \in R_X.$$

³Bei einem nicht optimal spielenden Gegenspieler können andere Strategien einen höheren Gewinn versprechen. Insbesondere psychologische Faktoren, die z.B. auch beim Schach eine Rolle spielen, können hier nicht betrachtet werden.

Es gilt $h(x) > 0$ nach Voraussetzung. Weiter gilt

$$\min_{y \in R_Y} E(x, y) = \min_{y \in R_Y} \left(\sum_{k=1}^m x^\top A e_k \cdot y_k \right) \geq \min_{y \in R_Y} \left(\sum_{k=1}^m h(x) y_k \right) = h(x) \cdot \min_{y \in R_Y} \left(\sum_{k=1}^m y_k \right) = h(x)$$

und

$$h(x) = \min_{y \in \{e_1, \dots, e_m\}} x^\top A y \geq \min_{y \in R_Y} x^\top A y = \min_{y \in R_Y} E(x, y)$$

Damit folgt

$$h(x) = \min_{y \in R_Y} E(x, y)$$

und somit gilt:

$$\max_{x \in R_X} \min_{y \in R_Y} E(x, y) = \max_{x \in R_X} h(x). \quad (1.6)$$

Mit der Variablensubstitution $z := h(x)$ lässt sich (1.6) als Maximierung von z umformen, wobei die (zusätzliche) Nebenbedingung $z \leq \min_{j \in \{1, \dots, m\}} x^\top A e_j$ beachtet werden muss, d.h. betrachtet wird

$$\max_{z, x} \left\{ z \in \mathbb{R} \mid z \leq x^\top A e_j, j = 1, \dots, m, x \in \mathbb{R}^n, x_i \geq 0, i = 1, \dots, n, \sum_{i=1}^n x_i = 1 \right\} \quad (1.7)$$

Zur Vereinfachung betrachte die Variablentransformation⁴ $u_i := \frac{x_i}{z}$ für $i = 1, \dots, n$, d.h.

$$\sum_{i=1}^n u_i = \frac{1}{z}.$$

Dann geht (1.7) über in

$$\min_u \left\{ \sum_{i=1}^n u_i \mid u \in \mathbb{R}^n, e_j^\top A^\top u \geq 1, j = 1, \dots, m, u_i \geq 0, i = 1, \dots, n \right\} \quad (1.8)$$

Das ursprüngliche ‘Maxmin-Problem’ ist also jetzt auf ein lineares Optimierungsproblem zurückgeführt worden.

Man sieht sofort, dass die Zielfunktion auf dem zulässigen Bereich nach unten durch 0 beschränkt ist. Aufgrund von $A_{i,j} > 0$ für alle $i = 1, \dots, n, j = 1, \dots, m$, überlegt man sich leicht, dass der zulässige Bereich nicht leer ist. Mit einem Satz der linearen Optimierung (Satz 2.6 auf Seite 18) folgt, dass (1.8) eine optimale Lösung u^* besitzt und es rücktransformiert also ein $x^* \in \mathbb{R}^n$ gibt mit

$$\min_{y \in R_Y} E(x^*, y) = \max_{x \in R_X} \min_{y \in R_Y} E(x, y).$$

Da R_Y ein Kompaktum ist und $E(x^*, y)$ in y linear und damit stetig ist, existiert somit auch ein $\hat{y} \in \mathbb{R}^m$ mit

$$E(x^*, \hat{y}) = \min_{y \in R_Y} E(x^*, y) = \max_{x \in R_X} \min_{y \in R_Y} E(x, y).$$

In völlig analoger Weise führt man das ‘Minimax-Problem’ auf die lineare Optimierungsaufgabe

$$\max_v \left\{ \sum_{j=1}^n v_j \mid v \in \mathbb{R}^m, e_i^\top A v \leq 1, i = 1, \dots, n, v_j \geq 0, j = 1, \dots, m \right\} \quad (1.9)$$

⁴Wegen der Positivitätsanforderungen an die Komponenten von A ist stets $z > 0$.

zurück und folgert die Existenz von $y^* \in \mathbb{R}^m$ und $\hat{x} \in \mathbb{R}^n$, so dass

$$E(\hat{x}, y^*) = \max_{x \in R_X} E(x, y^*) = \min_{y \in R_Y} \max_{x \in R_X} E(x, y).$$

Nun gilt:

$$E(x^*, \hat{y}) = \min_{y \in R_Y} E(x^*, y) \leq E(x^*, y^*) \leq \max_{x \in R_X} E(x, y^*) = E(\hat{x}, y^*).$$

Die Umkehrung dieser Beziehung, also

$$E(\hat{x}, y^*) \leq E(x^*, y^*) \leq E(x^*, \hat{y})$$

würde den Beweis vervollständigen. Da dazu aber noch weiteres theoretisches Rüstzeug aus der linearen Optimierung benötigt wird, sei zur Nachvollziehung auf [Neu75a] verwiesen. $\square$

Die beiden linearen Optimierungsprobleme (1.8) und (1.9) bezeichnet man als zueinander **duale** Probleme.

Der Beweis von Satz 1.4 wurde konstruktiv geführt mit Ausnahme der Lösung der abgeleiteten linearen Minimierungsaufgabe (1.8), so dass sich also ein computergestütztes Verfahren beschreiben lässt. In Kapitel 2 werden wir uns der algorithmischen Lösung linearer Optimierungsprobleme zuwenden.

Man bezeichnet ein Spiel als **fair**, wenn der Wert des Spieles $E(x^*, y^*)$ gleich 0 ist.

Beispiel 1.5 (Fortführung von Beispiel 1.3)

Um zu einer Matrix mit positiven Elementen zu gelangen, erhöhen wir alle Komponenten von A um 7, d.h.

$$B := \begin{pmatrix} 10 & 11 & 1 \\ 6 & 9 & 7 \\ 6 & 6 & 11 \end{pmatrix}.$$

Wie durch (1.8) beschrieben, ist nun folgendes lineares Optimierungsproblem zu lösen: Ermittle das Minimum von

$$u_1 + u_2 + u_3 \tag{1.10}$$

unter den Nebenbedingungen

$$\begin{aligned} 10u_1 + 6u_2 + 6u_3 &\geq 1, \\ 11u_1 + 9u_2 + 6u_3 &\geq 1, \\ u_1 + 7u_2 + 11u_3 &\geq 1, \\ u_1 &\geq 0, \\ u_2 &\geq 0, \\ u_3 &\geq 0. \end{aligned}$$

Sei u^* das optimale Ergebnis dieses Problems. Mit

$$s := u_1^* + u_2^* + u_3^*$$

lautet die optimale Strategie von Spieler 1 dann

$$x_1^* = \frac{u_1^*}{s}, \quad x_2^* = \frac{u_2^*}{s}, \quad x_3^* = \frac{u_3^*}{s},$$

und der Wert des Spiels ist

$$E(x^*, y^*) = \frac{1}{s} - 7.$$

Korollar 1.6 (Sattelpunkt)

Seien die Voraussetzungen von Satz 1.4 gegeben.

$x^* \in R_X$ und $y^* \in R_Y$ sind genau dann optimale Strategien, wenn gilt

$$E(x, y^*) \leq E(x^*, y^*) \leq E(x^*, y), \quad \text{für alle } x \in R_X, y \in R_Y. \quad (1.11)$$

(x^*, y^*) definieren also einen **Sattelpunkt** von E auf $R_X \times R_Y$.

Beweis siehe z.B. [Neu75a].

Weitere Problemstellungen und vertiefende Betrachtungen zur Spieltheorie sind z.B. in [Neu75a, MM73, Zim92] zu finden.

Kapitel 2

Lineare Optimierung

2.1 Beispiele und Problemstellung

In Abschnitt 1.7 wurde das Zweipersonen-Nullsummenspiel auf ein lineares Optimierungsproblem zurückgeführt. Ein Vielzahl von weiteren insbesondere wirtschaftsmathematischen Fragestellungen lassen sich ebenfalls als lineare Optimierungsaufgaben beschreiben z.B. in der Produktionsplanung, bei Investitionsproblemen, bei der Aufstellung von Finanzierungsplänen und der Lösung von Transportproblemen. Die lineare Optimierung (lineares Programmieren) gehört damit zu den klassischen und bedeutendsten Teilgebieten des Operations Research.

Beispiel 2.1 (Transportproblem [RR92])

In einem festen Planungszeitraum stehen in den Angebotsorten $A_1, \dots, A_n$ die Mengen $a_1, \dots, a_n$ eines bestimmten Artikels zur Verfügung. In den Bedarfsorten $B_1, \dots, B_m$ werden die Mengen $b_1, \dots, b_m$ dieses Artikels benötigt. Die Transportkosten je Mengeneinheit vom Ort A_i zum Ort B_j seien $c_{i,j}$ Geldeinheiten ($i = 1, \dots, n$, $j = 1, \dots, m$), wobei die Transportkosten proportional zu den transportierten Mengen angesetzt werden. Gesucht ist ein Transportplan mit minimalen Transportkosten.

Wir bezeichnen mit $x_{i,j}$ die vom Ort A_i zum Ort B_j transportierte Menge. Dann ist die Aufgabe, die gesamten Transportkosten

$$\sum_{i=1}^n \sum_{j=1}^m c_{i,j} x_{i,j}$$

unter den Nebenbedingungen

$$\begin{aligned} \sum_{j=1}^m x_{i,j} &\leq a_i, & i = 1, \dots, n, \\ \sum_{i=1}^n x_{i,j} &\geq b_j, & j = 1, \dots, m, \\ x_{i,j} &\geq 0, & i = 1, \dots, n, j = 1, \dots, m \end{aligned}$$

zu minimieren.

Beispiel 2.2 ([RR92])

Minimiere

$$-3x_1 - 4x_2$$

unter den Nebenbedingungen

$$-x_1 + x_2 \leq 4,$$

$$x_1 + 2x_2 \leq 14,$$

$$4x_1 + 3x_2 \leq 36,$$

$$-x_1 \leq 0,$$

$$-x_2 \leq 0.$$

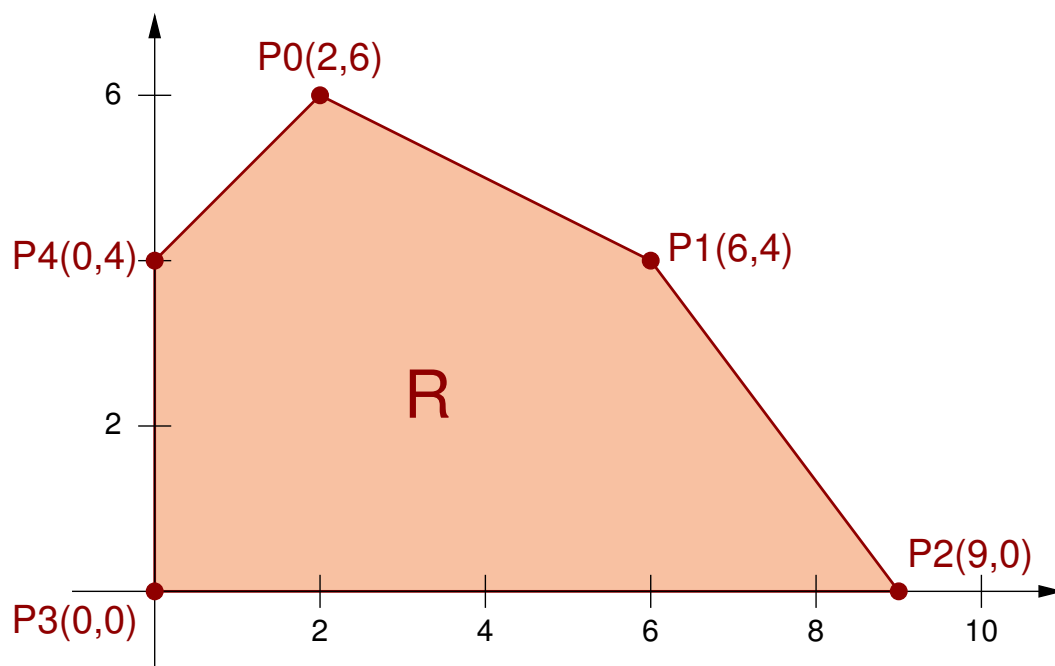


Abbildung 2.1: Zulässiger Bereich von Beispiel 2.2 (nach [RR92])

Wie in Abschnitt 1.3 beschrieben, ist bei linearen Optimierungsproblemen die zu minimierende **Zielfunktion** $F : \mathbb{R}^n \rightarrow \mathbb{R}$ von der Gestalt $F(x) = c^\top x$ für ein $c \in \mathbb{R}^n$. Die **Nebenbedingungen** werden allgemein durch eine Menge linearer (Un-)gleichungen der Form

$$a^\top x \begin{cases} \geq \\ = \\ \leq \end{cases} \beta,$$

beschrieben. Es ist üblich, die Nebenbedingungen in einer von mehreren möglichen Standardformen zu schreiben. Im Beispiel 2.2 folgt die Beschreibung der folgenden Ungleichungs-Standardform

$$\min_x \left\{ c^\top x \mid x \in \mathbb{R}^n, Ax \leq b \right\}, \quad \text{mit } A \in \mathbb{R}^{m,n}, b \in \mathbb{R}^m, c \in \mathbb{R}^n, \quad (2.1)$$

wobei die Ungleichungsbeziehung komponentenweise zu verstehen ist.

Beispiel 2.3

In unserem Beispiel 2.2 gilt

$$n = 2, \quad m = 5, \quad A = \begin{pmatrix} -1 & 1 \\ 1 & 2 \\ 4 & 3 \\ -1 & 0 \\ 0 & -1 \end{pmatrix}, \quad b = \begin{pmatrix} 4 \\ 14 \\ 36 \\ 0 \\ 0 \end{pmatrix}, \quad c = \begin{pmatrix} -3 \\ -4 \end{pmatrix}.$$

Sind $a_j \in \mathbb{R}^n$, $j = 1, \dots, m$, die Zeilen von A , d.h.

$$A = (a_1, \dots, a_m)^\top,$$

so heißt $a_j^\top x \leq b_j$ bzw. $h_j(x) \leq 0$ mit $h_j(x) := a_j^\top x - b_j$ die j -te Nebenbedingung von (2.1) und $a_j = \nabla h_j(x)$ heißt der **Gradient** der j -ten Nebenbedingung. Weiter heißt

$$\begin{aligned} R &:= \left\{ x \in \mathbb{R}^n \mid Ax \leq b \right\} \\ &= \left\{ x \in \mathbb{R}^n \mid a_j^\top x \leq b_j; j = 1, \dots, m \right\} = \left\{ x \in \mathbb{R}^n \mid h_j(x) \leq 0; j = 1, \dots, m \right\} \end{aligned}$$

der **zulässige Bereich** von (2.1). Ein $x \in \mathbb{R}^n$ heißt **zulässig**, wenn $x \in R$, und **unzulässig**, wenn $x \notin R$.

$$\text{Die } j\text{-te Nebenbedingung heißt in } x \begin{cases} \text{inaktiv,} & \text{falls } a_j^\top x < b_j, \\ \text{aktiv,} & \text{falls } a_j^\top x = b_j, \\ \text{verletzt,} & \text{falls } a_j^\top x > b_j. \end{cases}$$

Ein $x_0 \in \mathbb{R}^n$ heißt **optimale Lösung** von (2.1), wenn $x_0 \in R$ und $c^\top x_0 \leq c^\top x$ für alle $x \in R$.

2.2 Eigenschaften der Problemstellung

Auf der Suche nach einem Verfahren zur Bestimmung optimaler Lösungen des Problems (2.1) betrachten wir zunächst den zulässigen Bereich.

Abbildung 2.1 zeigt eine graphische Darstellung des zulässigen Bereichs von Beispiel 2.2, der die Gestalt eines zweidimensionalen Polyeders hat. Wir werden sehen, dass sich diese Beobachtung verallgemeinern läßt.

Die Menge $\left\{ x \in \mathbb{R}^n \mid a_j^\top x = b_j \right\}$ der Punkte x , für die die j -te Nebenbedingung aktiv ist, beschreibt eine Hyperebene in $\mathbb{R}^n$, so dass $\left\{ x \in \mathbb{R}^n \mid a_j^\top x \leq b_j \right\}$ somit alle Punkte sind, die auf einer Seite dieser Hyperebene (inklusive) liegen. Da $\left\{ x \in \mathbb{R}^n \mid a_j^\top x \leq b_j \right\} = h_j^{-1}([-\infty, 0])$, weiß man aus der Topologie sofort, dass diese Menge abgeschlossen ist. Somit ist auch

$$R = \left\{ x \in \mathbb{R}^n \mid a_j^\top x \leq b_j; j = 1, \dots, m \right\} = \bigcap_{j=1, \dots, m} \left\{ x \in \mathbb{R}^n \mid a_j^\top x \leq b_j \right\}$$

abgeschlossen. Falls R auch noch beschränkt ist, dann ist R kompakt.

Definition 2.4 (Konvexität)

Eine Menge $M \subseteq \mathbb{R}^n$ heißt konvex, wenn für alle $x, y \in M$ gilt

$$tx + (1-t)y \in M, \quad t \in [0, 1].$$

Satz 2.5 (Konvexität des zulässigen Bereichs)

Der zulässige Bereich R des linearen Optimierungsproblems (2.1) ist konvex.

Beweis. Seien $x, y \in R$. Dann gilt für alle $j = 1, \dots, m$

$$a_j^\top x \leq b_j, \quad a_j^\top y \leq b_j.$$

Für jedes $t \in [0, 1]$ und alle $j = 1, \dots, m$ folgt

$$a_j^\top (tx + (1-t)y) = ta_j^\top x + (1-t)a_j^\top y \leq tb_j + (1-t)b_j = b_j,$$

und somit gilt $tx + (1-t)y \in R$.

□

Damit können wir die folgende Aussage herleiten.

Satz 2.6 (Existenz einer optimalen Lösung)

Es sei R der zulässige Bereich des linearen Optimierungsproblems (2.1) und R sei nicht leer. Das Optimierungsproblem (2.1) besitzt genau dann (mindestens) eine optimale Lösung, wenn $c^\top x$ auf R nach unten beschränkt ist, d.h. wenn es ein $\gamma \in \mathbb{R}$ gibt mit $c^\top x \geq \gamma$ für alle $x \in R$.

Beweis. Ist der zulässige Bereich R beschränkt, so ist er kompakt. Die stetige Funktion $c^\top x$ nimmt dann auf diesem Kompaktum ihr Minimum an, d.h. es gibt eine optimale Lösung.

Nun betrachten wir den Fall, dass R unbeschränkt ist. Von jedem Punkt $x \in R$ aus gibt es dann ‘Strahlen ins Unendliche’, d.h. es gibt ein $s \in \mathbb{R}^n$, $\|s\| = 1$, und $\{y \in \mathbb{R}^n \mid y = x + \mu s, \mu \geq 0\} \subseteq R$, denn anderenfalls würde das konvexe R durch die Schnittpunkte aller Strahlen mit dem Rand von R beschränkt werden.

Für die Punkte y eines jeden solchen in R liegenden Strahls gilt

$$c^\top y = c^\top (x + \mu s) = c^\top x + \mu \cdot c^\top s = \begin{cases} > c^\top x, & \text{falls } c^\top s > 0, \mu > 0, \\ = c^\top x, & \text{falls } c^\top s = 0, \\ \xrightarrow{\mu \rightarrow \infty} -\infty, & \text{falls } c^\top s < 0. \end{cases}$$

Da aber $c^\top x \geq \gamma$ vorausgesetzt ist, gilt für die Richtungen s aller Strahlen in R , dass $c^\top s \geq 0$ und dass die Zielfunktion bereits am Ausgangspunkt x des Strahl ihren minimalen Wert annimmt.

Da R bzw. sein Rand durch die Schnitte endlich vieler Hyperflächen beschrieben wird, gibt es ein $r > 0$, so dass R die Vereinigung aller Strahlen mit Ausgangspunkten $x \in R$, $\|x\| \leq r$ darstellt. Auf dieser kompakten Menge nimmt $c^\top x$ sein Minimum an und entlang der Strahlen gibt es keinen Punkt, der einen kleineren Zielfunktionswert hätte, d.h. es gibt eine optimale Lösung.

□

Eine wichtige Rolle spielen die Ecken, die durch die Hyperflächenschnitte entstehen.

Definition 2.7 (Ecken des zulässigen Bereichs)

Sei R der zulässige Bereich des linearen Optimierungsproblems (2.1). Für $x_0 \in R$ sei $J \subseteq \{1, \dots, m\}$ die Indexmenge der in x_0 aktiven Nebenbedingungen, d.h.

$$a_j^\top x_0 = b_j, \quad j \in J, \quad a_j^\top x_0 < b_j, \quad j \in \{1, \dots, m\} \setminus J.$$

Sei $J = \{j_1, \dots, j_k\}$. Dann heißt x_0 eine **Ecke** von R , wenn gilt

$$\dim \langle a_{j_1}, \dots, a_{j_k} \rangle = \text{rang}(a_{j_1}, \dots, a_{j_k}) = n,$$

d.h. die a_j , $j \in J$, spannen den $\mathbb{R}^n$ auf.

Eine Ecke x_0 heißt **nichtentartet**, wenn $|J| = n$ ist, d.h. wenn die Gradienten a_j der aktiven Nebenbedingungen linear unabhängig sind. Anderenfalls, also $|J| > n$, heißt eine Ecke x_0 **entartet**. ┐

Wie man sich leicht überlegt, kann ein zulässiger Bereich R maximal $\binom{m}{n}$, also nur endlich viele Ecken besitzen ($m \geq n$).

Im Fall der Beschränktheit läßt sich R vollständig durch seine Ecken beschreiben.

Satz 2.8 (Polyedergestalt des zulässigen Bereichs)

Es sei R der zulässige Bereich des linearen Optimierungsproblems (2.1) und R sei beschränkt¹ und nicht leer. Weiter seien $x^1, \dots, x^M$ ($M \in \mathbb{N}$) die (endlich vielen) Ecken von R . Dann gilt

$$R = \left\{ x \in \mathbb{R}^n \mid x = \sum_{k=1}^M \lambda_k x^k, \lambda_k \geq 0, k = 1, \dots, M, \sum_{k=1}^M \lambda_k = 1 \right\}.$$

Somit ist R ein durch die Punkte $x^1, \dots, x^M$ aufgespanntes **konvexes Polyeder**. ┐

Beweis. siehe z.B. [Neu75a] □

Satz 2.9 (Optimalität einer Ecke des zulässigen Bereichs)

Es sei R der zulässige Bereich des linearen Optimierungsproblems (2.1) und R sei beschränkt und nicht leer. Dann ist eine Ecke von R eine optimale Lösung von (2.1). ┐

Beweis. Aufgrund der Kompaktheit von R besitzt (2.1) optimale Lösungen und $\hat{x}$ sei eine solche optimale Lösung.

Annahme: Es gibt keine Ecke von R , die eine optimale Lösung darstellt, d.h. für alle Ecken x^k von R gilt $c^\top x^k > c^\top \hat{x}$, $k = 1, \dots, M$. Nach Satz 2.8 gibt es aber $\lambda_k \geq 0$, $k = 1, \dots, M$, $\sum_{k=1}^M \lambda_k = 1$, und

$$c^\top \hat{x} = c^\top \left(\sum_{k=1}^M \lambda_k x^k \right) = \sum_{k=1}^M \lambda_k c^\top x^k > \sum_{k=1}^M \lambda_k c^\top \hat{x} = c^\top \hat{x}.$$

Damit ist die Annahme widerlegt, d.h. (mindestens) eine Ecke von R ist eine optimale Lösung von (2.1). □

¹d.h. es gibt ein $r > 0$ mit $\|x\| \leq r$ für alle $x \in R$.

Aufgrund des letzten Satzes wissen wir, dass wir bei der Suche nach einer optimalen Problemlösung nur die endlich vielen Ecken des zulässigen Bereichs betrachten müssen. Ein Lösungsverfahren muss daher ‘nur’ schrittweise die Ecken absuchen. Beim Verfahren sollte nach Möglichkeit die jeweils nächste betrachtete Ecke einen kleineren Zielfunktionswert besitzen als die aktuelle. Als physikalisches Analogon kann man sich einen auf einer schrägen Tischplatte abrollenden Ball vorstellen, der sich entlang einer Begrenzung von Ecke zu Ecke bewegt.

Zur Formulierung des Verfahrens betrachten wir eine zweite Standardform des Problemschreibung, das sogenannte **Simplex-Standardproblem**

$$\min_x \left\{ c^\top x \mid x \in \mathbb{R}^n, Ax = b, x \geq 0 \right\}, \quad \text{mit } A \in \mathbb{R}^{m,n}, b \in \mathbb{R}^m, c \in \mathbb{R}^n, b \geq 0, \operatorname{rg}(A) = m. \quad (2.2)$$

- Man beachte, dass aus der Voraussetzung $\operatorname{rg}(A) = m$ folgt, dass $m \leq n$. Bei nichttrivialen Problemen ist $m < n$.
- Falls der zulässige Bereich nicht leer ist, dann besitzt (2.2) immer eine Ecke.

Es gibt noch eine Reihe weiterer Standardformen, die sich jeweils äquivalent in andere Standardformen umformen lassen.

Umformung der Ungleichungs-Standardform 2.1 in die Simplex-Standardform 2.2:

Sei also $\hat{A} \in \mathbb{R}^{\hat{m},\hat{n}}, \hat{b} \in \mathbb{R}^{\hat{m}}, \hat{c} \in \mathbb{R}^{\hat{n}}$ gegeben und das folgende Problem betrachtet

$$\min_y \left\{ \hat{c}^\top y \mid y \in \mathbb{R}^{\hat{n}}, \hat{A}y \leq \hat{b} \right\}.$$

Formal lässt sich jedes y_i in zwei nichtnegative Anteile aufspalten, d.h. $y_i = x_{2i-1} - x_{2i}$, wobei $x_{2i-1} \geq 0$ und $x_{2i} \geq 0$. Desweiteren kann jede Ungleichung durch Einführung einer sog. **Schlupfvariable** zu einer Gleichung umgeformt werden, d.h. $\hat{a}_j^\top y + x_{2\hat{n}+i} = \hat{b}_i$ und $x_{2\hat{n}+i} \geq 0$. Schließlich müssen gegebenenfalls die Gleichungen mit -1 multipliziert werden, um $b_j \geq 0$ zu sichern.

Beispiel 2.10

Bei praktischen Beispielen lässt sich oftmals geschickter vorgehen. Wir betrachten erneut unser Beispiel 2.2 bzw. Beispiel 2.3 und formen es in die Simplex-Standardform um. Wir erhalten durch die Einführung von drei Schlupfvariablen

$$n = 5, \quad m = 3, \quad A = \begin{pmatrix} -1 & 1 & 1 & 0 & 0 \\ 1 & 2 & 0 & 1 & 0 \\ 4 & 3 & 0 & 0 & 1 \end{pmatrix}, \quad b = \begin{pmatrix} 4 \\ 14 \\ 36 \end{pmatrix}, \quad c = \begin{pmatrix} -3 \\ -4 \\ 0 \\ 0 \\ 0 \end{pmatrix}.$$

Umformung der Simplex-Standardform 2.2 in die Ungleichungs-Standardform 2.1:

Formal ist die Umformung erreicht, indem man aus jeder Gleichung $a_j^\top x = b_j$ zwei Ungleichungen $a_j^\top x \leq b_j$ und $-a_j^\top x \leq -b_j$ bildet und $x \geq 0$ als $-x \leq 0$ schreibt. Auch hier kann man natürlich geschickter vorgehen (siehe obiges Beispiel).

Alle bisherigen Sätze und Aussagen gelten also auch für die Simplex-Standardform 2.2.

Um die Verfahrensidee zu verwirklichen, benötigen wir noch eine ‘griffigere’ Beschreibung von Ecken, die wir aus folgendem Satz erhalten.

Satz 2.11 (Hinreichende Optimalitätsbedingungen)

Man betrachte das Simplex-Standardproblem (2.2) und einen zulässigen Punkt $x_0 \in \mathbb{R}^n$ dieses Standardproblems, d.h.

$$Ax_0 = b, \quad x_0 \geq 0.$$

x_0 ist optimal, wenn es $u \in \mathbb{R}^m$ und $v \in \mathbb{R}^n$ gibt mit

$$A^\top u - v = -c, \quad v \geq 0, \quad v^\top x_0 = 0.$$

Beweis. Zu zeigen ist

$$c^\top x_0 \leq c^\top x, \quad \text{für alle zulässigen } x \in R.$$

Mit den gegebenen Voraussetzungen gilt für alle zulässigen $x \in R$

$$\begin{aligned} -c^\top (x - x_0) &= (A^\top u - v)^\top (x - x_0) \\ &= u^\top Ax - u^\top Ax_0 - v^\top x + v^\top x_0 \\ &= u^\top b - u^\top b - v^\top x = -\underbrace{v^\top}_{\geq 0} \underbrace{x}_{\geq 0} \leq 0 \end{aligned}$$

Und damit ist $c^\top x_0 \leq c^\top x$.

□

Die Komponenten von u und v werden die **Multiplikatoren** zu den Nebenbedingungen genannt. Über die Betrachtung dieser Multiplikatoren werden im Simplex-Verfahren die einzelnen Ecken berechnet.

2.3 Die (revidierte) Simplex-Methode

2.3.1 Vorbereitungen

Mit Hilfe des von Satz 2.11 soll nun die Verfahrensidee, die Ecken des zulässigen Bereichs (Polyeder) abzusuchen, ausformuliert werden. Das resultierende Verfahren geht zurück auf George B. Dantzig, der es in den 1940er Jahren als ‘Simplex-Methode’ entwickelt hat.

Wir betrachten zunächst eine nichtentartete Ecke x zum Simplex-Standardproblem (2.2). Da dort genau n Nebenbedingungen aktiv sind und die m Gleichungsnebenbedingungen immer aktiv sind, müssen also noch $n - m$ Ungleichungen aktiv sein, d.h. für $n - m$ Komponenten von x gilt $x_i = 0$ und für die restlichen Komponenten gilt $x_i > 0$. Wir werden nun allgemein (eindeutige) Multiplikatoren $u \in \mathbb{R}^m$ und $v \in \mathbb{R}^n$ berechnen mit

$$A^\top u - v = -c, \quad v^\top x = 0.$$

Wenn dann zusätzlich $v \geq 0$ erfüllt ist, dann ist x nach Satz 2.11 optimal. Falls nicht, dann werden wir eine aktive Nebenbedingung durch eine andere ersetzen und dadurch zu einer benachbarten Ecke gelangen.

Die Indizes der m nicht-aktiven Ungleichungen sollen $\beta_1, \dots, \beta_m$ heißen, d.h. $x_{\beta_j} > 0$ für $j = 1, \dots, m$. Diese x_{β_j} heißen **Basisvariable** und die $n - m$ restlichen verschwindenden Komponenten von x heißen **Nichtbasisvariable**. Weiter betrachten wir die Indexmengen

$$B := \{\beta_1, \dots, \beta_m\}, \quad N := \{1, \dots, n\} \setminus B,$$

und verwenden die abkürzende Schreibweise

$$x_B := \begin{pmatrix} x_{\beta_1} \\ \vdots \\ x_{\beta_m} \end{pmatrix},$$

und entsprechend x_N usw., wobei die Reihenfolge der Komponenten durch $\beta_1, \dots, \beta_m$ bestimmt ist. Nach Voraussetzung gilt also $x_B > 0$ und $x_N = 0$.

Als nächstes seien $A_1, \dots, A_n \in \mathbb{R}^m$ die Spalten von A , d.h. $A = (A_1, \dots, A_n)$. Auch hier schreiben wir abkürzend

$$A_B := (A_{\beta_1}, \dots, A_{\beta_m}) \in \mathbb{R}^{m,m}.$$

Wir nehmen an, dass A_B vollen Rang hat, also invertierbar ist (wir werden sehen, dass dies immer erfüllbar ist). Zur späteren Vereinfachung definiere

$$D := A_B^{-\top} = (A_B^{-1})^\top = (A_B^\top)^{-1} = (A_{\beta_1}, \dots, A_{\beta_m})^{-\top} =: (d_1, \dots, d_m),$$

d.h. $d_1, \dots, d_m$ sind die Spalten von D .

Mit diesen Abkürzungen schreiben wir $Ax = b$ als

$$Ax = \sum_{j \in B} A_j x_j + \sum_{j \in N} A_j x_j = A_B x_B + A_N x_N = b$$

Da $x_N = 0$, folgt

$$A_B x_B = b, \quad \text{und somit} \quad x_B = A_B^{-1} b = D^\top b.$$

Analog gilt

$$c^\top x = c_B^\top x_B + c_N^\top x_N = c_B^\top x_B.$$

2.3.2 Bestimmung der Multiplikatoren

Nun können wir die Multiplikatoren $u \in \mathbb{R}^m$ und $v \in \mathbb{R}^n$ bestimmen mit dem Ansatz

$$A^\top u - v = -c, \quad v^\top x = 0.$$

Zum einen ist also

$$0 = v^\top x = v_B^\top x_B + v_N^\top x_N = v_B^\top x_B,$$

und da $x_B > 0$ ergibt sich $v_B = 0$.

Zum anderen gilt mit den Partitionierungen

$$\begin{pmatrix} A_B^\top \\ A_N^\top \end{pmatrix} u - \begin{pmatrix} v_B \\ v_N \end{pmatrix} = \begin{pmatrix} -c_B \\ -c_N \end{pmatrix},$$

so dass wir durch komponentenweise Betrachtung erhalten

$$\begin{aligned} A_B^\top u &= -c_B, \\ A_N^\top u - v_N &= -c_N, \end{aligned}$$

und somit

$$\begin{aligned} u &= -\left(A_B^\top\right)^{-1} c_B = -D c_B, \\ v_N &= c_N + A_N^\top u. \end{aligned}$$

Damit haben wir die gesuchten Multiplikatoren berechnet. Falls nun zusätzlich $v_N \geq 0$ gilt, so ist x nach Satz 2.11 eine optimale Lösung des Simplex-Standardproblems (2.2).

2.3.3 Eckenaustausch

Falls nicht gilt $v_N \geq 0$, dann wollen wir eine k -te aktive Nebenbedingung der Ecke x mit $k \in N$ zu einer inaktiven Nebenbedingung machen, d.h. $\hat{x}_k$ zu einer Basisvariable machen, also von der Ecke x zu einer neuen zulässigen Ecke $\hat{x}$ gelangen. Die restlichen aktiven Nebenbedingungen sollen erhalten bleiben.

Sei also nun $k \in N$ und

$$v_k < 0.$$

Der Ansatz zur Bestimmung der nächsten Ecke $\hat{x}$ ist

$$\hat{x} = x - \sigma s, \quad \text{mit } \sigma \geq 0, s \in \mathbb{R}^n,$$

wobei wir s als **Suchrichtung** und σ als **Schrittweite** bezeichnen.

Alle aktiven Nebenbedingungen mit Index aus $N \setminus \{k\}$ sollen aktiv bleiben, d.h. wir setzen $s_N := -(e_k)_N$, wobei $e_k \in \mathbb{R}^n$ der k -te Einheitsvektor ist. Es gilt dann

$$\hat{x}_j = x_j, \text{ für } j \in N \setminus \{k\}, \quad \hat{x}_k = x_k - \sigma \cdot (-1) = \sigma \geq 0.$$

Insbesondere haben wir $\hat{x}_N \geq 0$.

Der neue Punkt $\hat{x}$ muss die Gleichungsnebenbedingungen erfüllen, also

$$b = A\hat{x} = A(x - \sigma s) = A_B x_B - \sigma A_B s_B + A_N x_N - \sigma A_N s_N = b - \sigma A_B s_B + \sigma A_k$$

Für positive σ gilt also

$$s_B = (A_B)^{-1} A_k = D^\top A_k.$$

Weiter muss der neue Punkt $\hat{x}$ auch die Ungleichungsnebenbedingungen erfüllen, also

$$0 \leq \hat{x}_B = x_B - \sigma s_B, \quad \text{also} \quad \sigma s_B \leq x_B.$$

Zusätzlich betrachten wir noch den Wert der Zielfunktion im neuen Punkt, also

$$\begin{aligned} c^\top \hat{x} &= c^\top (x - \sigma s) = c_B^\top (x_B - \sigma s_B) + c_N^\top (x_N - \sigma s_N) \\ &= c_B^\top x_B + \sigma \left(c_k - c_B^\top D^\top A_k \right) = c^\top x + \sigma \left(c_k + A_k^\top u \right) \\ &= c^\top x + \sigma v_k. \end{aligned}$$

Wenn nun $s_B \leq 0$, dann bleibt $x - \sigma s$ zulässig für alle $\sigma \geq 0$ und wegen $v_k < 0$ geht die Zielfunktion dann gegen $-\infty$, d.h. das Problem ist dann nach unten nicht beschränkt und es existiert keine Lösung.

Wenn aber mindestens eine Komponente von s_B positiv ist, dann ist der maximal zulässige Wert für σ gegeben durch

$$\sigma := \min \left\{ \frac{x_{\beta_j}}{s_{\beta_j}} \mid s_{\beta_j} > 0, j = 1, \dots, m \right\}.$$

Da $x_B > 0$, folgt $0 < \sigma < \infty$. Insbesondere gilt

$$c^\top \hat{x} < c^\top x, \quad (2.3)$$

d.h. an der neuen Ecke $\hat{x}$ nimmt die Zielfunktion einen kleineren Wert an als an der alten Ecke x .

Ist auch die neue Ecke $\hat{x}$ nicht entartet, so wird also genau eine weitere Nebenbedingung aktiv, um die aufgegebene k -te zu ersetzen. Sei dazu $l \in \{1, \dots, m\}$ der Index mit

$$\sigma = \frac{x_{\beta_l}}{s_{\beta_l}} = \min \left\{ \frac{x_{\beta_j}}{s_{\beta_j}} \mid s_{\beta_j} > 0, j = 1, \dots, m \right\}.$$

Die Basis-Indexmenge der neuen Ecke lässt sich dann wie folgt beschreiben:

$$\hat{\beta}_j := \beta_j, \quad \text{für } j = 1, \dots, m, j \neq l, \quad \hat{\beta}_l := k,$$

und

$$\hat{B} := \{\hat{\beta}_1, \dots, \hat{\beta}_m\} = (B \setminus \{\beta_l\}) \cup \{k\}, \quad \hat{N} := \{1, \dots, n\} \setminus \hat{B},$$

Nun definiere

$$\begin{aligned} \hat{d}_l &:= \frac{1}{s_{\beta_l}} d_l, \\ \hat{d}_j &:= d_j - s_{\beta_j} \hat{d}_l, \quad \text{für } j = 1, \dots, m, j \neq l, \\ \hat{D} &:= (\hat{d}_1, \dots, \hat{d}_m). \end{aligned}$$

Dann gilt

$$\begin{aligned} A_{\hat{\beta}_i}^\top \hat{d}_j &= A_{\beta_i}^\top \left(d_j - \frac{s_{\beta_j}}{s_{\beta_l}} d_l \right) = 0, \quad \text{für } i, j \in \{1, \dots, m\}, i \neq l, j \neq l, j \neq i, \\ A_{\hat{\beta}_i}^\top \hat{d}_l &= A_{\beta_i}^\top \left(\frac{1}{s_{\beta_l}} d_l \right) = 0, \quad \text{für } i \in \{1, \dots, m\}, i \neq l, \\ A_{\hat{\beta}_i}^\top \hat{d}_i &= A_{\beta_i}^\top \left(d_i - \frac{s_{\beta_i}}{s_{\beta_l}} d_l \right) = A_{\beta_i}^\top d_i = 1, \quad \text{für } i \in \{1, \dots, m\}, i \neq l, \\ A_{\hat{\beta}_l}^\top \hat{d}_j &= A_k^\top \left(d_j - \frac{s_{\beta_j}}{s_{\beta_l}} d_l \right) = s_{\beta_j} - \frac{s_{\beta_j}}{s_{\beta_l}} s_{\beta_l} = 0, \quad \text{für } j \in \{1, \dots, m\}, j \neq l, \\ A_{\hat{\beta}_l}^\top \hat{d}_l &= A_k^\top \left(\frac{1}{s_{\beta_l}} d_l \right) = \frac{1}{s_{\beta_l}} s_{\beta_l} = 1. \end{aligned}$$

Insgesamt haben wir also

$$A_{\hat{B}}^\top \hat{D} = I_m,$$

und damit

$$\hat{D} = A_{\hat{B}}^{-\top}.$$

Jetzt haben wir alle Voraussetzungen, um an der Ecke $\hat{x}$ wieder die Multiplikatoren bestimmen zu können.

2.3.4 Behandlung entarteter Ecken

Bisher haben wir stets nichtentartete Ecken betrachtet. Bei gut gestellten Problemen aus der Praxis kommt der Fall der Entartung sehr selten vor, aber es lassen sich natürlich Beispiele konstruieren, bei denen in einer Ecke mehr als m Nebenbedingungen aktiv sind. Das führt dazu, dass bei einer entarteten Ecke die Basis-Indexmenge nicht mehr eindeutig ist, da x_B nun auch verschwindende Komponenten enthalten muss, die im Prinzip willkürlich aus der Menge aller verschwindenden Komponenten von x zu wählen sind. Ebenso geht auch die Eindeutigkeit der Multiplikatoren verloren.

Der kritische Fall tritt bei der Berechnung der Schrittweite

$$\sigma := \min \left\{ \frac{x_{\beta_j}}{s_{\beta_j}} \mid s_{\beta_j} > 0, j = 1, \dots, m \right\}.$$

ein, da nun auch $\sigma = 0$ eintreten kann. In diesem Fall ist $\hat{x} = x$, d.h. man verläßt die Ecke nicht, sondern wechselt lediglich die Basis. Hierbei ist zu beachten

- sämtliche Möglichkeiten zu prüfen, die aktuelle Ecke zu verlassen,
- beim Wechsel der Basis innerhalb der Ecke nicht eine schon untersuchte zu verwenden.

Ohne strategisches Vorgehen besteht sonst die Gefahr, durch zyklisches Austauschen von Basen innerhalb einer Ecke in einer Endlosschleife zu kreisen. Mögliche Strategien sind

- eine Indexverwaltung, bei der alle Möglichkeit der Reihe nach geprüft werden,
- die Einführung einer lexikographischen Ordnung, die die möglichen Basen einer Ecke in eine Reihenfolge bringt und so einen ‘Abstieg’ innerhalb einer Ecke definiert,
- Monte-Carlo-Wahl an allen Alternativstellen (Wahl der Indizes k und l), so dass der Raum der Möglichkeiten stochastisch exploriert wird,
- simplifizierte Strategien, die nicht alle, aber die wahrscheinlichsten kritischen Fälle vermeiden,
- der Verzicht auf eine spezielle Strategie, so lange ‘alles gut geht’.

Im Weiteren betrachten wir nur den Fall nichtentarteter Ecken, um den Aufwand zu reduzieren. Für eine vertiefte Betrachtung sei auf [CW71, RR92, Neu75a] verwiesen.

2.3.5 Zusammenfassung des Simplex-Algorithmus

Für den Fall nichtentarteter Ecken können wir nun die Erkenntnisse und Herleitungen der vorangegangenen Abschnitte zum **Simplex-Algorithmus** (nach George B. Dantzig) zusammenfassen. In Algorithmus 2.1 wird das Verfahren vollständig beschrieben, wobei vorausgesetzt ist, dass eine Ecke des zulässigen Bereichs als Startpunkt vorliegt.

Im Falle der Nichtentartung aller Ecken sieht man sehr leicht, dass der Algorithmus eine optimale Lösung produziert (falls eine existiert). Wie wir uns schon überlegt hatten, gibt es nur endlich viele

Revidierte Simplex-Methode (RSM): **Eingang** A, b, c, x, β, D ; **Ausgang** x, β, D ;

Problemstellung:

$$\min_x \left\{ c^\top x \mid x \in \mathbb{R}^n, Ax = b, x \geq 0 \right\},$$

wobei $A \in \mathbb{R}^{m,n}, b \in \mathbb{R}^m, c \in \mathbb{R}^n, b \geq 0, \text{rg}(A) = m$.

Schritt 0: Initialisierung:

Gegeben sei eine zulässige Ecke x , Basis-Indizes $\beta_1, \dots, \beta_m$ und $D = (d_1, \dots, d_m) \in \mathbb{R}^{m,m}$.
 $B = \{\beta_1, \dots, \beta_m\}, D = A_B^{-\top}$ und $x_B = D^\top b, x_N = 0$ wird vorausgesetzt.

Gehe zu Schritt 1

Schritt 1: Ermittlung der Multiplikatoren und der Suchrichtung

$$B := \{\beta_1, \dots, \beta_m\}, N := \{1, \dots, n\} \setminus B.$$

$$u := -Dc_B.$$

$$v_N := c_N + A_N^\top u.$$

falls $v_N \geq 0$:

STOP mit optimaler Lösung x , Indizes β und Matrix D .

sonst:

k sei der kleinste Index mit $k \in N$ und $v_k := \min \{v_j \mid j \in N\}$.

$$s_B := D^\top A_k;$$

Gehe zu Schritt 2

Schritt 2: Ermittlung der Schrittweite

falls $s_B \leq 0$:

STOP: Das Problem ist nach unten nicht beschränkt.

sonst:

$$\sigma := \min \left\{ \frac{x_{\beta_j}}{s_{\beta_j}} \mid s_{\beta_j} > 0, j = 1, \dots, m \right\}.$$

l sei der kleinste Index mit $l \in \{1, \dots, m\}$ und $\sigma := \frac{x_{\beta_l}}{s_{\beta_l}}$.

Gehe zu Schritt 3

Schritt 3: Eckenaustausch (Update)

$$x_k := \sigma;$$

$$x_B := x_B - \sigma s_B;$$

Insbesondere $x_{\beta_l} := 0$

$$d_l := \frac{1}{s_{\beta_l}} d_l;$$

$$d_j := d_j - s_{\beta_j} d_l, \quad \text{für } j = 1, \dots, m, j \neq l;$$

$$D := (d_1, \dots, d_m);$$

$$\beta_l := k;$$

Gehe zu Schritt 1

Algorithmus 2.1: Die revidierte Simplex-Methode

mögliche Ecken des zulässigen Bereichs R . Außerdem wissen wir durch (2.3), dass bei Nichtentartung der Zielfunktionswert jeder berechneten Ecke kleiner ist als der der vorhergehenden. Somit wird nach endlich vielen Schritten die Ecke mit dem kleinsten Zielfunktionswert erreicht, die nach

Satz 2.9 auf Seite 19 zugleich die optimale Lösung des Simplex-Standardproblems (2.2) ist.

Beispiel 2.12

Zur Übung werden wir den Simplex-Algorithmus nun auf Beispiel 2.2 anwenden, welches wir in Beispiel 2.10 bereits in die Simplex-Standardform gebracht hatten:

$$n = 5, \quad m = 3, \quad A = \begin{pmatrix} -1 & 1 & 1 & 0 & 0 \\ 1 & 2 & 0 & 1 & 0 \\ 4 & 3 & 0 & 0 & 1 \end{pmatrix}, \quad b = \begin{pmatrix} 4 \\ 14 \\ 36 \end{pmatrix}, \quad c = \begin{pmatrix} -3 \\ -4 \\ 0 \\ 0 \\ 0 \end{pmatrix}.$$

Da A die Einheitsmatrix enthält, lässt sich eine zulässige Startecke sehr einfach aus der Problemstellung ablesen, indem die Variablen, die nicht zu Einheitsvektoren gehören, auf Null gesetzt werden und der Rest die Komponenten von b erhält.

Schritt 0: Auf diese Weise lesen wir die Startecke wie folgt ab:

$$x = \begin{pmatrix} 0 \\ 0 \\ 4 \\ 14 \\ 36 \end{pmatrix}, \quad \beta_1 := 3, \beta_2 := 4, \beta_3 := 5, \quad D := \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}.$$

Schritt 1:

$$B := \{3, 4, 5\}, \quad N := \{1, 2\}, \quad x_B = \begin{pmatrix} 4 \\ 14 \\ 36 \end{pmatrix}, \quad c_B = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix},$$

$$u := -Dc_B = 0, \quad \begin{pmatrix} v_1 \\ v_2 \end{pmatrix} = v_N := c_N + A_N^\top u = c_N = \begin{pmatrix} -3 \\ -4 \end{pmatrix}.$$

$$\text{Damit folgt } k = 2, \quad v_k = -4, \quad s_B := D^\top A_2 = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 \\ 2 \\ 3 \end{pmatrix} = \begin{pmatrix} 1 \\ 2 \\ 3 \end{pmatrix}.$$

Schritt 2:

$$\frac{x_{\beta_1}}{s_{\beta_1}} = \frac{4}{1} = 4, \quad \frac{x_{\beta_2}}{s_{\beta_2}} = \frac{14}{2} = 7, \quad \frac{x_{\beta_3}}{s_{\beta_3}} = \frac{36}{3} = 12.$$

$$\text{Daraus folgt } \sigma = 4, \quad l = 1.$$

Schritt 3:

$$x_2 := \sigma = 4, \quad \begin{pmatrix} x_3 \\ x_4 \\ x_5 \end{pmatrix} = x_B := x_B - \sigma s_B = \begin{pmatrix} 4 \\ 14 \\ 36 \end{pmatrix} - 4 \begin{pmatrix} 1 \\ 2 \\ 3 \end{pmatrix} = \begin{pmatrix} 0 \\ 6 \\ 24 \end{pmatrix}.$$

$$\begin{aligned}
d_1 &:= \frac{1}{s_{\beta_1}} d_1 = \frac{1}{1} \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} \\
d_2 &:= d_2 - s_{\beta_2} d_1 = \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix} - 2 \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} = \begin{pmatrix} -2 \\ 1 \\ 0 \end{pmatrix}, \\
d_3 &:= d_3 - s_{\beta_3} d_1 = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} - 3 \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} = \begin{pmatrix} -3 \\ 0 \\ 1 \end{pmatrix}. \\
D &:= \begin{pmatrix} 1 & -2 & -3 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}, \quad \beta_1 := 2.
\end{aligned}$$

Schritt 1:

$$\begin{aligned}
B &:= \{2, 4, 5\}, \quad N := \{1, 3\}, \quad x_B = \begin{pmatrix} 4 \\ 6 \\ 24 \end{pmatrix}, \quad c_B = \begin{pmatrix} -4 \\ 0 \\ 0 \end{pmatrix}, \\
u &:= -Dc_B = \begin{pmatrix} 1 & -2 & -3 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 4 \\ 0 \\ 0 \end{pmatrix} = \begin{pmatrix} 4 \\ 0 \\ 0 \end{pmatrix}, \\
\begin{pmatrix} v_1 \\ v_3 \end{pmatrix} &= v_N := c_N + A_N^\top u = \begin{pmatrix} -3 \\ 0 \end{pmatrix} + \begin{pmatrix} -1 & 1 & 4 \\ 1 & 0 & 0 \end{pmatrix} \begin{pmatrix} 4 \\ 0 \\ 0 \end{pmatrix} = \begin{pmatrix} -7 \\ 4 \end{pmatrix}.
\end{aligned}$$

$$\text{Damit folgt } k = 1, \quad v_1 = -7, \quad s_B := D^\top A_1 = \begin{pmatrix} 1 & 0 & 0 \\ -2 & 1 & 0 \\ -3 & 0 & 1 \end{pmatrix} \begin{pmatrix} -1 \\ 1 \\ 4 \end{pmatrix} = \begin{pmatrix} -1 \\ 3 \\ 7 \end{pmatrix}.$$

Schritt 2:

$$\begin{aligned}
\frac{x_{\beta_2}}{s_{\beta_2}} &= \frac{6}{3} = 2, \quad \frac{x_{\beta_3}}{s_{\beta_3}} = \frac{24}{7} > 2. \\
\text{Daraus folgt } \sigma &= 2, \quad l = 2.
\end{aligned}$$

Schritt 3:

$$\begin{aligned}
x_1 &:= \sigma = 2, \quad \begin{pmatrix} x_2 \\ x_4 \\ x_5 \end{pmatrix} = x_B := x_B - \sigma s_B = \begin{pmatrix} 4 \\ 6 \\ 24 \end{pmatrix} - 2 \begin{pmatrix} -1 \\ 3 \\ 7 \end{pmatrix} = \begin{pmatrix} 6 \\ 0 \\ 10 \end{pmatrix}. \\
d_2 &:= \frac{1}{s_{\beta_2}} d_2 = \frac{1}{3} \begin{pmatrix} -2 \\ 1 \\ 0 \end{pmatrix} \\
d_1 &:= d_1 - s_{\beta_1} d_2 = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} + \frac{1}{3} \begin{pmatrix} -2 \\ 1 \\ 0 \end{pmatrix} = \frac{1}{3} \begin{pmatrix} 1 \\ 1 \\ 0 \end{pmatrix}, \\
d_3 &:= d_3 - s_{\beta_3} d_2 = \begin{pmatrix} -3 \\ 0 \\ 1 \end{pmatrix} - 7 \cdot \frac{1}{3} \begin{pmatrix} -2 \\ 1 \\ 0 \end{pmatrix} = \frac{1}{3} \begin{pmatrix} 5 \\ -7 \\ 3 \end{pmatrix}.
\end{aligned}$$

$$D := \frac{1}{3} \begin{pmatrix} 1 & -2 & 5 \\ 1 & 1 & -7 \\ 0 & 0 & 3 \end{pmatrix}, \quad \beta_2 := 1.$$

Schritt 1:

$$B := \{2, 1, 5\}, \quad N := \{3, 4\}, \quad x_B = \begin{pmatrix} 6 \\ 2 \\ 10 \end{pmatrix}, \quad c_B = \begin{pmatrix} -4 \\ -3 \\ 0 \end{pmatrix},$$

$$u := -Dc_B = \frac{1}{3} \begin{pmatrix} 1 & -2 & 5 \\ 1 & 1 & -7 \\ 0 & 0 & 3 \end{pmatrix} \begin{pmatrix} 4 \\ 3 \\ 0 \end{pmatrix} = \frac{1}{3} \begin{pmatrix} -2 \\ 7 \\ 0 \end{pmatrix},$$

$$\begin{pmatrix} v_3 \\ v_4 \end{pmatrix} = v_N := c_N + A_N^\top u = \begin{pmatrix} 0 \\ 0 \end{pmatrix} + \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix} \frac{1}{3} \begin{pmatrix} -2 \\ 7 \\ 0 \end{pmatrix} = \frac{1}{3} \begin{pmatrix} -2 \\ 7 \end{pmatrix}.$$

$$\text{Damit folgt } k = 3, \quad v_3 = -\frac{2}{3}, \quad s_B := D^\top A_3 = \frac{1}{3} \begin{pmatrix} 1 & 1 & 0 \\ -2 & 1 & 0 \\ 5 & -7 & 3 \end{pmatrix} \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} = \frac{1}{3} \begin{pmatrix} 1 \\ -2 \\ 5 \end{pmatrix}.$$

Schritt 2:

$$\frac{x_{\beta_1}}{s_{\beta_1}} = 3 \cdot \frac{6}{1} = 18, \quad \frac{x_{\beta_3}}{s_{\beta_3}} = 3 \cdot \frac{10}{5} = 6.$$

$$\text{Daraus folgt } \sigma = 6, \quad l = 3.$$

Schritt 3:

$$x_3 := \sigma = 6, \quad \begin{pmatrix} x_2 \\ x_1 \\ x_5 \end{pmatrix} = x_B := x_B - \sigma s_B = \begin{pmatrix} 6 \\ 2 \\ 10 \end{pmatrix} - \frac{6}{3} \begin{pmatrix} 1 \\ -2 \\ 5 \end{pmatrix} = \begin{pmatrix} 4 \\ 6 \\ 0 \end{pmatrix}.$$

$$d_3 := \frac{1}{s_{\beta_3}} d_3 = \frac{3}{5} \frac{1}{3} \begin{pmatrix} 5 \\ -7 \\ 3 \end{pmatrix} = \frac{1}{5} \begin{pmatrix} 5 \\ -7 \\ 3 \end{pmatrix}$$

$$d_1 := d_1 - s_{\beta_1} d_3 = \frac{1}{3} \begin{pmatrix} 1 \\ 1 \\ 0 \end{pmatrix} - \frac{1}{3} \frac{1}{5} \begin{pmatrix} 5 \\ -7 \\ 3 \end{pmatrix} = \frac{1}{15} \begin{pmatrix} 0 \\ 12 \\ -3 \end{pmatrix} = \frac{1}{5} \begin{pmatrix} 0 \\ 4 \\ -1 \end{pmatrix},$$

$$d_2 := d_2 - s_{\beta_2} d_3 = \frac{1}{3} \begin{pmatrix} -2 \\ 1 \\ 0 \end{pmatrix} + \frac{2}{3} \frac{1}{5} \begin{pmatrix} 5 \\ -7 \\ 3 \end{pmatrix} = \frac{1}{15} \begin{pmatrix} 0 \\ -9 \\ 6 \end{pmatrix} = \frac{1}{5} \begin{pmatrix} 0 \\ -3 \\ 2 \end{pmatrix}.$$

$$D := \frac{1}{5} \begin{pmatrix} 0 & 0 & 5 \\ 4 & -3 & -7 \\ -1 & 2 & 3 \end{pmatrix}, \quad \beta_3 := 3.$$

Schritt 1:

$$B := \{2, 1, 3\}, \quad N := \{4, 5\}, \quad x_B = \begin{pmatrix} 4 \\ 6 \\ 6 \end{pmatrix}, \quad c_B = \begin{pmatrix} -4 \\ -3 \\ 0 \end{pmatrix},$$

$$u := -Dc_B = \frac{1}{5} \begin{pmatrix} 0 & 0 & 5 \\ 4 & -3 & -7 \\ -1 & 2 & 3 \end{pmatrix} \begin{pmatrix} 4 \\ 3 \\ 0 \end{pmatrix} = \frac{1}{5} \begin{pmatrix} 0 \\ 7 \\ 2 \end{pmatrix},$$

$$\begin{pmatrix} v_4 \\ v_5 \end{pmatrix} = v_N := c_N + A_N^\top u = \begin{pmatrix} 0 \\ 0 \end{pmatrix} + \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \frac{1}{5} \begin{pmatrix} 0 \\ 7 \\ 2 \end{pmatrix} = \frac{1}{5} \begin{pmatrix} 7 \\ 2 \end{pmatrix} > 0.$$

Die Optimalitätsbedingungen sind damit erfüllt, d.h.

$$x_{\text{opt}} = \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \end{pmatrix} = \begin{pmatrix} 6 \\ 4 \\ 6 \\ 0 \\ 0 \end{pmatrix}$$

ist die optimale Lösung des Problems von Beispiel 2.10. Die letzten drei Variable waren eingeführte Schlupfvariable, so dass die optimale Lösung von Beispiel 2.3 somit lautet

$$x_1 = 6, \quad x_2 = 4.$$

2.3.6 Ermittlung einer zulässigen Startecke (Phase 1)

Nun ist noch die Frage offen, wie man allgemein zu einer zulässigen Startecke kommt. Wir erreichen dies durch die Formulierung und Lösung eines Hilfsproblems. Zum Simplex-Standardproblem (2.2) betrachten wir das künstliche **Startpunktproblem (Initial Point Problem)**:

$$\min_{x,y} \left\{ e^\top y \mid x \in \mathbb{R}^n, y \in \mathbb{R}^m, Ax + y = b, x \geq 0, y \geq 0 \right\}, \quad \text{mit } e = (1, \dots, 1)^\top \in \mathbb{R}^m. \quad (2.4)$$

Es gilt

- Das Problem (2.4) ist ein lineares Optimierungsproblem in $n + m$ Variablen, die alle eine Nichtnegativitätsbedingung erfüllen müssen.
- Eine zulässige Ecke von (2.4) ist durch $x = 0$ und $y = b$ gegeben².
- Die Zielfunktion $e^\top y$ ist nach unten durch Null beschränkt und besitzt damit nach Satz 2.6 auf Seite 18 eine optimale Lösung.

Jetzt bringen wir Problem (2.2) durch Definition von

$$\hat{A} := (A, I_m), \quad \hat{b} := b, \quad \hat{c} := \begin{pmatrix} 0 \\ e \end{pmatrix}, \quad \hat{x} := \begin{pmatrix} x \\ y \end{pmatrix}$$

noch in die Simplex-Standardform

$$\min_{\hat{x}} \left\{ \hat{c}^\top \hat{x} \mid \hat{x} \in \mathbb{R}^{n+m}, \hat{A}\hat{x} = \hat{b}, \hat{x} \geq 0 \right\}. \quad (2.5)$$

²In (2.2) war $b \geq 0$ Voraussetzung.

Die Initialisierung des Simplex-Algorithmus 2.1 kann durch die zulässige Ecke

$$\hat{x} := \begin{pmatrix} 0 \\ b \end{pmatrix}, \quad \hat{\beta}_1 := n+1, \hat{\beta}_2 := n+2, \dots, \hat{\beta}_m := n+m, \quad \hat{D} := I_m,$$

erfolgen, d.h. Algorithmus 2.1 produziert unter den gegebenen Voraussetzungen eine optimale Lösung $\hat{x}^*$ mit Indizes $\hat{\beta}^*$ und Matrix $\hat{D}^*$ zum Problems (2.5).

Für $\hat{x}^* = \begin{pmatrix} x^* \\ y^* \end{pmatrix}$ sind zwei Fälle zu unterscheiden:

1. $y^* \neq 0$:

In diesem Fall besitzt das Simplex-Problem (2.2) keinen zulässigen Punkt. Wäre nämlich $\tilde{x}$ zulässig für (2.2), dann wäre $\begin{pmatrix} \tilde{x} \\ \tilde{y} \end{pmatrix}$ mit $\tilde{y} = 0$ zulässig für (2.5), aber $e^\top \tilde{y} = 0 < e^\top y^*$ widerspricht der Optimalität von y^* .

2. $y^* = 0$:

Dann ist x^* eine zulässige Ecke des Simplex-Problems (2.2) und die Lösung von (2.2) kann in einer zweiten Phase mit Algorithmus 2.1 mit den Initialwerten $x = x^*$, $\beta = \hat{\beta}^*$ und $D = \hat{D}^*$ berechnet werden. Im Falle der Entartung muss gegebenenfalls noch die Basis gewechselt werden, falls darin noch künstliche Variablen y_j enthalten wären.

Enthält A bereits Einheitsvektoren, z.B. weil A durch Einführung von Schlupfvariablen erzeugt wurde, so kann man sich einige oder alle künstliche Variable sparen (siehe Beispiel 2.12 aus dem vorangegangenen Abschnitt).

Bei der Betrachtung des Zweipersonen-Nullsummenspiels in Abschnitt 1.7 hatten wir zwei zueinander **duale** lineare Optimierungsprobleme erkannt. Solche dualen Aussagen lassen sich für jedes lineare Optimierungsproblem formulieren und führen auf die **Dualitätstheorie**, die wir hier aber nicht weiter behandeln können. Zur Vertiefung sei auf [RR92] oder [CW71, MM73, Neu75a, Run86, Nie84, Zim92, Ker87] verwiesen.

Kapitel 3

Problemstellung und Lösungsansätze unrestringierter nichtlinearer Optimierungsprobleme

3.1 Mathematische Problemstellung

Gegeben ist eine **Zielfunktion**

$$F : \mathbb{R}^n \rightarrow \mathbb{R}$$

(resultierend aus einem Anwendungsproblem).

Gesucht ist ein $x^* \in \mathbb{R}^n$ mit

$$F(x^*) \leq F(x), \quad \text{für alle } x \in \mathbb{R}^n.$$

Die Kurzbezeichnung der Problemstellung lautet

$$\min_{x \in \mathbb{R}^n} \{F(x)\}, \quad \left(\min_x \{F(x) \mid x \in \mathbb{R}^n\} \right) \quad (3.1)$$

‘**Minimierungsproblem ohne Nebenbedingungen**’ bzw. ‘**Unrestringiertes Minimierungsproblem**’.

Ein solches x^* heißt ein **globaler Minimierer** von F .

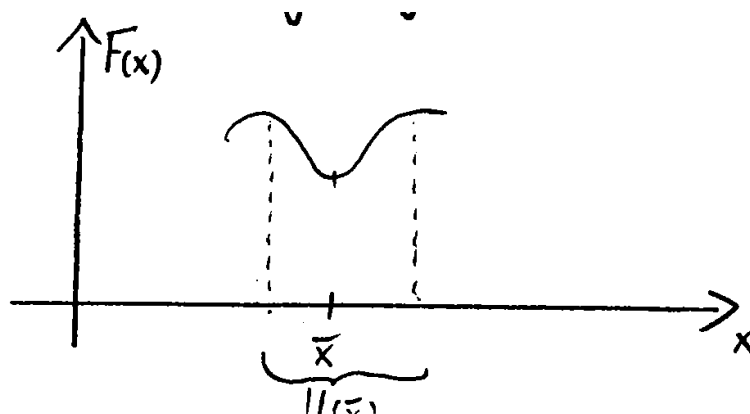


Abbildung 3.1: Offene Umgebung

Die Berechnung eines globalen Minimierers ist i.a. ein sehr kompliziertes Problem, so dass man sich meist auf die Berechnung von **lokalen Minimierern** $\bar{x} \in \mathbb{R}^n$ beschränkt, für die gilt:

$$F(\bar{x}) \leq F(x), \quad \text{für alle } x \in U(\bar{x}),$$

wobei $U(\bar{x})$ eine offene Umgebung von $\bar{x}$ ist, siehe Abbildung 3.1.

Betrachten wir dazu die folgenden Visualisierungsbeispiele mit $n = 1$.

Beispiel 3.1

$$F : \mathbb{R} \rightarrow \mathbb{R}, x \mapsto \exp(x).$$

Die Exponentialfunktion besitzt keine lokalen (und damit auch keine globalen) Minimierer, ist aber dennoch nach unten durch den Wert 0 beschränkt, siehe Abbildung 3.2. Satz 2.6 aus der linearen Optimierung lässt sich also **nicht** übertragen auf nichtlineare Probleme.

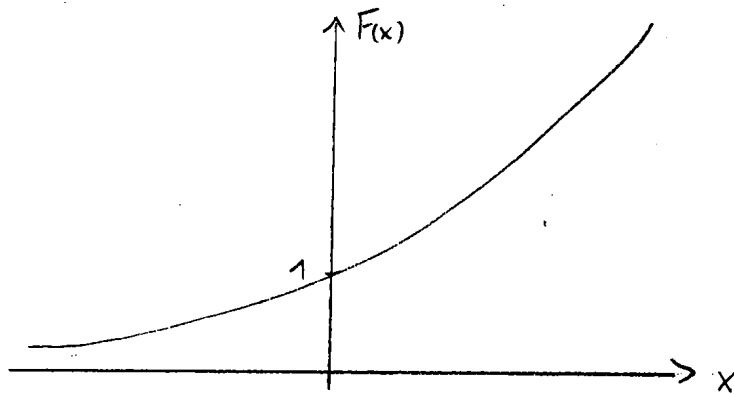


Abbildung 3.2: Exponentialfunktion

Beispiel 3.2

$$F : \mathbb{R} \rightarrow \mathbb{R}, x \mapsto \sin(x).$$

F besitzt unendlich viele lokale Minimierer, siehe Abbildung 3.3. Jeder lokale Minimierer ist zugleich auch ein globaler Minimierer.

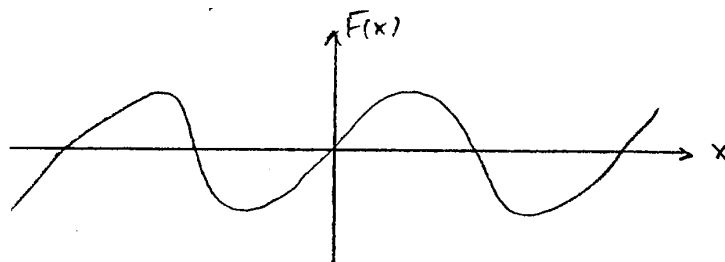


Abbildung 3.3: Sinusfunktion

Beispiel 3.3

$$F : \mathbb{R} \rightarrow \mathbb{R}, x \mapsto \frac{1}{x^4 - 2x^2 + 2}.$$

F besitzt genau einen lokalen Minimierer, aber keinen globalen Minimierer, siehe Abbildung 3.4.

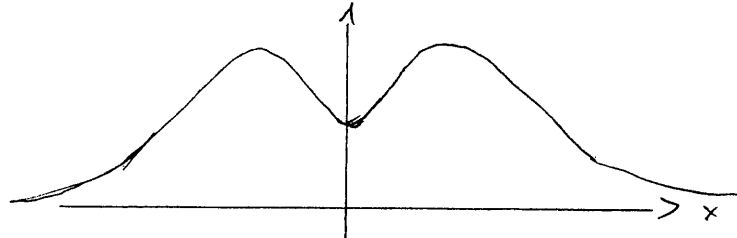


Abbildung 3.4: Lokaler Minimierer ohne globalen Minimierer

3.2 Optimalitätskriterien

Numerische Verfahren zur Berechnung von Minimierern benötigen Kriterien, anhand derer sie entscheiden können, ob ein Punkt $x \in \mathbb{R}^n$ ein lokaler oder globaler Minimierer ist. Die definierenden Eigenschaften in Abschnitt 3.1 sind aber algorithmisch nicht nachprüfbar. Gesucht sind also (mit dem Computer) nachprüfbare Optimalitätsbedingungen, die als Abbruchkriterien für Optimierungsalgorithmen dienen können.

Man unterscheidet zwischen **notwendigen Optimalitätsbedingungen**, die ein Minimierer erfüllen muss, und **hinreichenden Optimalitätsbedingungen**, die garantieren, dass der betrachtete Punkt ein Minimierer ist. Da es keine algorithmisch nachprüfbaren Optimalitätsbedingungen gibt, die speziell globale Minimierer auszeichnen, werden vor allem Verfahren zur lokalen Optimierung entwickelt.

Sofern die Zielfunktion F genügend glatt ist, lassen sich solche Optimalitätsbedingungen mit Hilfe des **Gradienten**

$$\nabla F(x) := \begin{pmatrix} \frac{\partial F(x)}{\partial x_1} \\ \vdots \\ \frac{\partial F(x)}{\partial x_n} \end{pmatrix} \in \mathbb{R}^n,$$

und der **Hessematrix** (Spezialfall der **Jacobimatrix**)

$$\nabla^2 F(x) := \begin{pmatrix} \frac{\partial^2 F(x)}{\partial x_1 \partial x_1} & \cdots & \frac{\partial^2 F(x)}{\partial x_1 \partial x_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial^2 F(x)}{\partial x_n \partial x_1} & \cdots & \frac{\partial^2 F(x)}{\partial x_n \partial x_n} \end{pmatrix} \in \mathbb{R}^{n,n},$$

an Stellen $x \in \mathbb{R}^n$ formulieren.

Durch den Gradienten $\nabla F(x)$ wird die Richtung der stärksten Vergrößerung der Funktionswerte startend an der Stelle x angegeben. Folglich gibt der Vektor $-\nabla F(x)$ die Richtung der stärksten

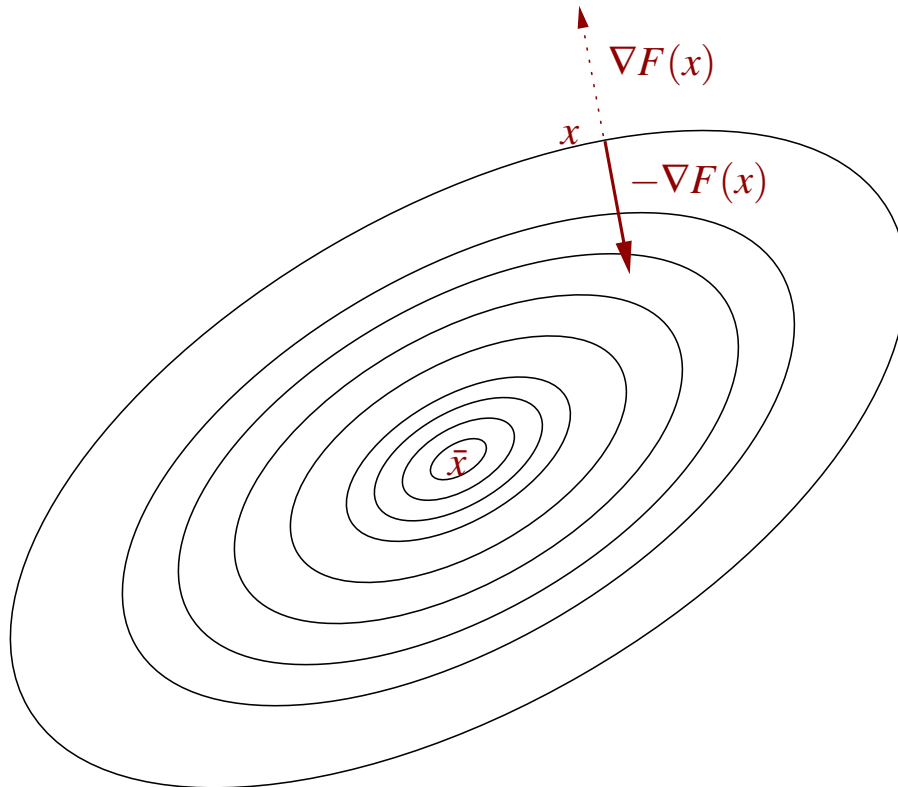


Abbildung 3.5: Richtung des steilsten Abstiegs

Verkleinerung der Funktionswerte von F an der Stelle x an und wird daher auch als **Richtung des steilsten Abstiegs** bezeichnet, siehe Abbildung 3.5.

Ist $\bar{x}$ ein lokaler Minimierer, so erwarten wir also, dass $\nabla F(\bar{x}) = 0$. Dies gilt aber auch für lokale Maximierer.

Satz und Definition 3.4 (Optimalitätsbedingungen)

- (i) Sei $F : \mathbb{R}^n \rightarrow \mathbb{R}$ stetig differenzierbar und sei $\bar{x} \in \mathbb{R}^n$ ein lokaler Minimierer von F . Dann gilt die **notwendige Bedingung erster Ordnung**:

$$\nabla F(\bar{x}) = 0.$$

- (ii) Sei $F : \mathbb{R}^n \rightarrow \mathbb{R}$ zweimal stetig differenzierbar und sei $\bar{x} \in \mathbb{R}^n$ ein lokaler Minimierer von F . Dann gilt die **notwendige Bedingung zweiter Ordnung**:

$$\nabla F(\bar{x}) = 0, \quad \nabla^2 F(\bar{x}) \text{ ist positiv-semidefinit,}$$

$$\text{d.h. } v^\top \nabla^2 F(\bar{x}) v \geq 0 \text{ für alle } v \in \mathbb{R}^n.$$

- (iii) Sei $F : \mathbb{R}^n \rightarrow \mathbb{R}$ zweimal stetig differenzierbar und für $\hat{x} \in \mathbb{R}^n$ gelte die **hinreichende Bedingung zweiter Ordnung**:

$$\nabla F(\hat{x}) = 0, \quad \nabla^2 F(\hat{x}) \text{ ist positiv-definit,}$$

$$\text{d.h. } v^\top \nabla^2 F(\hat{x}) v > 0 \text{ für alle } v \in \mathbb{R}^n \setminus \{0\}.$$

Dann ist $\hat{x}$ ein lokaler Minimierer von F .

Beweis.

Ad (i): $\bar{x}$ ist ein lokaler Minimierer, also $F(\bar{x}) \leq F(x)$ für alle $x \in U(\bar{x})$, wobei $U(\bar{x})$ eine offene Umgebung von $\bar{x}$ ist. Somit gibt es ein $\delta > 0$, so dass

$$\bar{x} + \varepsilon s \in U(\bar{x}), \quad \text{für alle } 0 < \varepsilon < \delta \text{ und alle } s \in S_1(0) = \left\{ x \in \mathbb{R}^n \mid \|x\|_2^2 = 1 \right\}.$$

Wir betrachten nun die **Taylor-Entwicklung** von F bis zum ersten Glied¹, d.h.

$$F(\bar{x} + \varepsilon s) = F(\bar{x}) + \nabla F(\bar{x})^\top \varepsilon s + o(\|\varepsilon s\|),$$

wobei o das **Landau-Symbol** ist mit der Bedeutung

$$\lim_{a \rightarrow 0} \frac{o(a)}{a} = 0.$$

Damit gilt

$$0 \leq F(\bar{x} + \varepsilon s) - F(\bar{x}) = \varepsilon \nabla F(\bar{x})^\top s + o(\varepsilon),$$

und damit

$$0 \leq \nabla F(\bar{x})^\top s + \frac{o(\varepsilon)}{\varepsilon}.$$

Jetzt folgt

$$0 \leq \lim_{\varepsilon \downarrow 0} \left[\nabla F(\bar{x})^\top s + \frac{o(\varepsilon)}{\varepsilon} \right] = \nabla F(\bar{x})^\top s, \quad \text{für alle } s \in S_1(0). \quad (3.2)$$

Wäre $\nabla F(\bar{x}) \neq 0$, so wähle

$$s := -\frac{\nabla F(\bar{x})}{\|\nabla F(\bar{x})\|} \in S_1(0), \quad \text{also} \\ \nabla F(\bar{x})^\top s = -\|\nabla F(\bar{x})\| < 0$$

im Widerspruch zu (3.2).

Ad (ii): Wir betrachten nun die Taylor-Entwicklung bis zum zweiten Glied²

$$0 \leq F(\bar{x} + \varepsilon s) - F(\bar{x}) = \underbrace{\varepsilon \nabla F(\bar{x})^\top s}_{=0} + \frac{1}{2} \varepsilon^2 s^\top \nabla^2 F(\bar{x}) s + o(\varepsilon^2)$$

Daraus folgt

$$0 \leq s^\top \nabla^2 F(\bar{x}) s + 2 \frac{o(\varepsilon^2)}{\varepsilon^2},$$

und wir erhalten

$$0 \leq \lim_{\varepsilon \downarrow 0} \left[s^\top \nabla^2 F(\bar{x}) s + 2 \frac{o(\varepsilon^2)}{\varepsilon^2} \right] = s^\top \nabla^2 F(\bar{x}) s, \quad \text{für alle } s \in S_1(0).$$

¹Allgemein unter geeigneten Voraussetzungen:

$$F(x) = F(y) + \nabla F(y)^\top (x - y) + o(\|x - y\|).$$

²Allgemein unter geeigneten Voraussetzungen:

$$F(x) = F(y) + \nabla F(y)^\top (x - y) + \frac{1}{2} (x - y)^\top \nabla^2 F(y) (x - y) + o(\|x - y\|^2).$$

Somit ist $\nabla^2 F(\bar{x})$ positiv-semidefinit.

Ad (iii): Für $\hat{x}$ gelte $\nabla F(\hat{x}) = 0$ und $\nabla^2 F(\hat{x})$ positiv-definit. Wenn $\lambda_{\min}$ der kleinste Eigenwert von $\nabla^2 F(\hat{x})$ ist, so gilt

$$s^\top \nabla^2 F(\hat{x}) s \geq \lambda_{\min} > 0, \quad \text{für alle } s \in S_1(0).$$

Erneut betrachten wir die Taylor-Entwicklung:

$$F(\hat{x} + \varepsilon s) - F(\hat{x}) = \underbrace{\varepsilon \nabla F(\hat{x})^\top s}_{=0} + \underbrace{\frac{1}{2} \varepsilon^2 s^\top \nabla^2 F(\hat{x}) s}_{\geq \lambda_{\min}} + o(\varepsilon^2) \geq \varepsilon^2 \left(\frac{1}{2} \lambda_{\min} + \underbrace{\frac{o(\varepsilon^2)}{\varepsilon^2}}_{\xrightarrow{\varepsilon \downarrow 0} 0} \right)$$

Somit existiert ein $\delta > 0$ mit

$$\frac{o(\varepsilon^2)}{\varepsilon^2} \leq \frac{1}{4} \lambda_{\min}, \quad \text{für alle } 0 < \varepsilon < \delta,$$

und folglich gilt

$$F(\hat{x} + \varepsilon s) - F(\hat{x}) \geq \varepsilon^2 \left(\frac{1}{2} \lambda_{\min} - \frac{1}{4} \lambda_{\min} \right) = \frac{1}{4} \varepsilon^2 \lambda_{\min} > 0, \quad \text{für alle } s \in S_1(0) \text{ und } 0 < \varepsilon < \delta.$$

Nun ist

$$U(\hat{x}) := \left\{ x \in \mathbb{R}^n \mid x = \hat{x} + y, \|y\| < \delta \right\} = \left\{ x \in \mathbb{R}^n \mid x = \hat{x} + \varepsilon s, 0 \leq \varepsilon < \delta, s \in S_1(0) \right\}$$

eine offene Umgebung von $\hat{x}$ mit $F(\hat{x}) \leq F(x)$ für alle $x \in U(\hat{x})$. Also ist $\hat{x}$ ein lokaler Minimierer von F .

□

Bemerkung: Da $F(\hat{x}) < F(x)$ für alle $x \in U(\hat{x}) \setminus \{\hat{x}\}$, ist $\hat{x}$ sogar ein **isolierter** lokaler Minimierer von F .

Zu den Optimalitätsbedingungen können wir für $n = 1$ wieder Visualisierungsbeispiele betrachten.

Beispiel 3.5

$$F : \mathbb{R} \rightarrow \mathbb{R}, x \mapsto x^2, \quad F'(x) = 2x, \quad F''(x) = 2 > 0.$$

Bei $\bar{x} = 0$ gilt $F'(\bar{x}) = 0$ und $F''(\bar{x}) > 0$, d.h. die notwendigen Bedingungen erster und zweiter Ordnung sowie die hinreichende Bedingung zweiter Ordnung sind erfüllt, also ist $\bar{x}$ ein Minimierer, siehe Abbildung 3.6.

—

Beispiel 3.6

$$F : \mathbb{R} \rightarrow \mathbb{R}, x \mapsto -x^2, \quad F'(x) = -2x, \quad F''(x) = -2 < 0.$$

Bei $\bar{x} = 0$ gilt $F'(\bar{x}) = 0$ und $F''(\bar{x}) < 0$, d.h. die notwendige Bedingung erster Ordnung ist erfüllt, aber die notwendige Bedingung zweiter Ordnung nicht, also ist $\bar{x}$ kein Minimierer, siehe Abbildung 3.7.

—

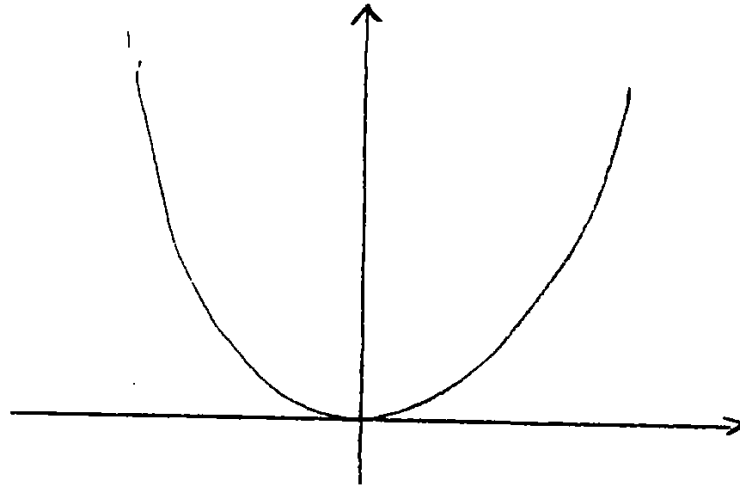


Abbildung 3.6: Notwendige und hinreichende Bedingungen erfüllt

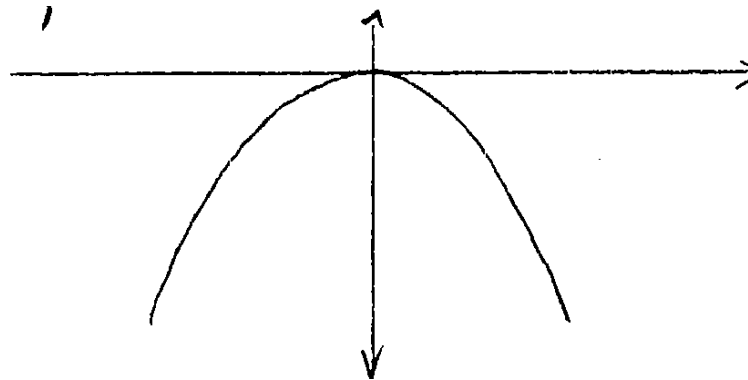


Abbildung 3.7: Notwendige Bedingung zweiter Ordnung nicht erfüllt

Beispiel 3.7

$$F : \mathbb{R} \rightarrow \mathbb{R}, x \mapsto x^3, \quad F'(x) = 3x^2, \quad F''(x) = 6x.$$

Bei $\bar{x} = 0$ gilt $F'(\bar{x}) = 0$ und $F''(\bar{x}) = 0$, d.h. die notwendigen Bedingungen erster und zweiter Ordnung sind erfüllt, aber die hinreichende Bedingung zweiter Ordnung nicht. Tatsächlich ist $\bar{x}$ kein Minimierer, siehe Abbildung 3.8.

Beispiel 3.8

$$F : \mathbb{R} \rightarrow \mathbb{R}, x \mapsto x^4, \quad F'(x) = 4x^3, \quad F''(x) = 12x^2.$$

Bei $\bar{x} = 0$ gilt $F'(\bar{x}) = 0$ und $F''(\bar{x}) = 0$, d.h. die notwendigen Bedingungen erster und zweiter Ordnung sind erfüllt, aber die hinreichende Bedingung zweiter Ordnung nicht. Dennoch ist $\bar{x}$ ein lokaler Minimierer (sogar ein globaler), siehe Abbildung 3.9. In diesem Fall war die hinreichende Bedingung zu scharf, um den Minimierer erkennen zu können (höhere Ableitungen wären hier zur Erkennung notwendig).

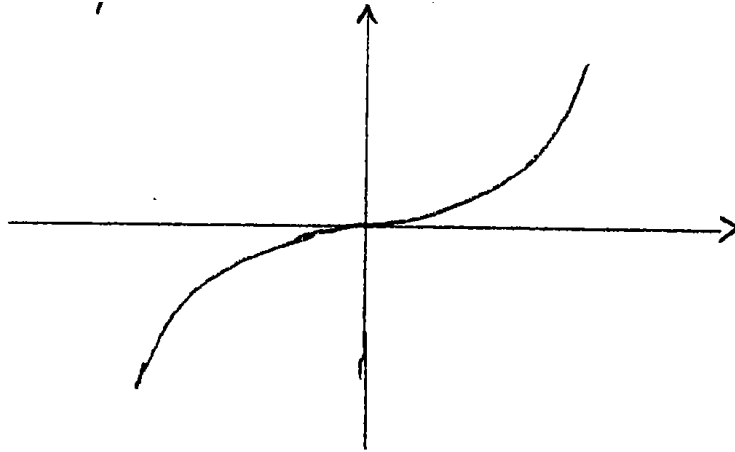


Abbildung 3.8: Notwendige Bedingungen erfüllt, hinreichende nicht, kein Minimierer

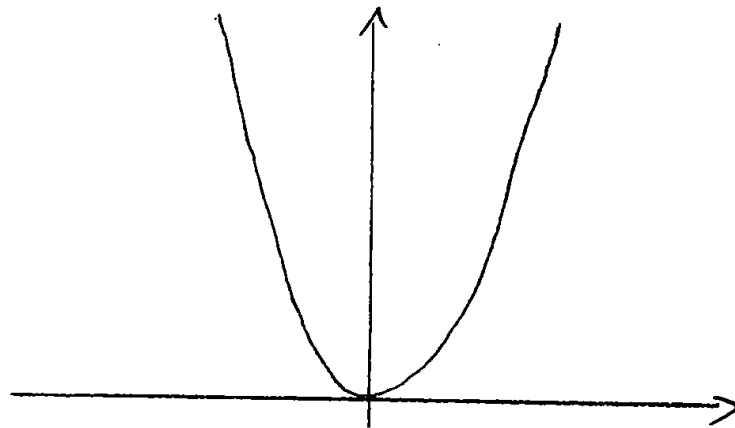


Abbildung 3.9: Notwendige Bedingungen erfüllt, hinreichende nicht, dennoch Minimierer

3.3 Numerische Verfahren (Iterationsverfahren)

3.3.1 Modellbildung

Im allgemeinen kann ein nichtlineares Optimierungsproblem nicht analytisch gelöst werden (‘auf einen Schlag’). Daher konstruiert man sukzessive eine Folge von einfachen **Ersatzproblemen**, für die je ein Minimierer $x_j \in \mathbb{R}^n$, $j \in \mathbb{N}$, bestimmbar ist. In der Regel verwendet man die über die Zielfunktion gesammelten Informationen (Funktionswerte, Gradienten, Hessematrizen, etc.), um ein **lokales Modell** der Zielfunktion zu erstellen, welches dann als Zielfunktion des Ersatzproblems fungiert. Aufgrund ihrer einfachen Struktur sind diese Modelle oft quadratische Polynome, siehe Kapitel 4. Für eine **globale Optimierung** reichen lokale Informationen außer in trivialen Fällen nicht aus, um einen globalen Minimierer zu bestimmen. Daher enthalten globale Verfahren zusätzlich **stochastische Elemente** (**Zufallssuche**, **Monte-Carlo-Verfahren**).

Nichtlineare Optimierungsverfahren erzeugen also iterativ eine Folge von Punkten $x_j \in \mathbb{R}^n$, $j \in \mathbb{N}$, ausgehend von einem gegebenen **Startpunkt** $x_0 \in \mathbb{R}^n$, und sind daher **Iterationsverfahren**, siehe Abbildung 3.10.

Um die **Konvergenz** des Verfahrens zu sichern, sind zwei Aspekte zu betrachten:

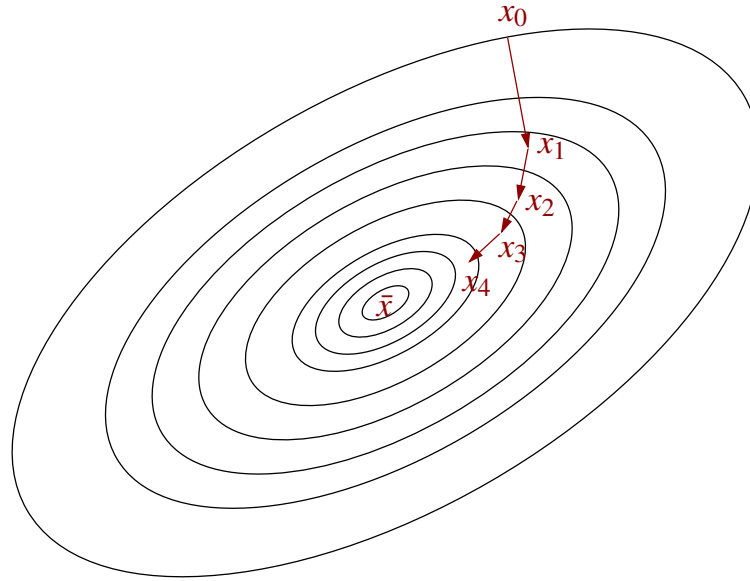


Abbildung 3.10: Punktfolge eines Iterationsverfahrens

1. Die Zielfunktion muss in der Regel mathematische Voraussetzungen erfüllen (Differenzierbarkeit, Verhalten im Unendlichen, etc.), die in Anwendungsfällen oft nicht überprüfbar oder gegeben sind.
2. Die Verfahrensvorschrift muss so konstruiert sein, dass unter gegebenen Voraussetzungen die Folge $\{x_j\}_{j \in \mathbb{N}_0}$ gegen einen Minimierer $\bar{x}$ konvergiert (schwächer: eine Teilfolge; global: eine stochastische Konvergenzaussage).

Konvergenzaussagen gelten in der Regel für unendliche Folgen $\{x_j\}_{j \in \mathbb{N}_0}$, aber praktische numerische Verfahren stoppen natürlich nach endlicher Zeit bei Erreichung eines **Abbruchkriteriums** (etwa: ‘Gradient klein genug’) mit einem Punkt $x_N \in \mathbb{R}^n$ als Approximation des Minimierers.

In [Abbildung 3.11](#) ist das allgemeine Modellierungsschema für nichtlineare Optimierungsprobleme dargestellt. Zu bedenken ist:

- Das mathematische Optimierungsproblem ist selbst ein Modell des eigentlichen Anwendungsproblems, siehe [Abbildung 1.1](#) (‘Ist die mathematische Modellierung adäquat?’).
- Die Konvergenz der Folge $\{x_j\}_{j \in \mathbb{N}_0}$ gegen einen Minimierer ist oft nur unter Voraussetzungen beweisbar, die die Zielfunktion nicht erfüllt (‘Konvergiert das Verfahren auch bei dieser Zielfunktion?’).
- Die approximierte Lösung x_N ist von der tatsächlichen Lösung möglicherweise weit entfernt (‘Gute Approximation erst bei $N = 10^{20}$ statt bei $N = 27$?’).
- Beantwortet x_N die ursprüngliche Fragestellung?

Trotz aller möglichen Probleme und kritischen Fragen sind die Erfolge bei der Optimierung industrieller Anwendungsprobleme hoch. Die verwendeten Verfahren müssen bei schwachen Voraussetzungen an die Zielfunktion robuste Konvergenzeigenschaften aufweisen.

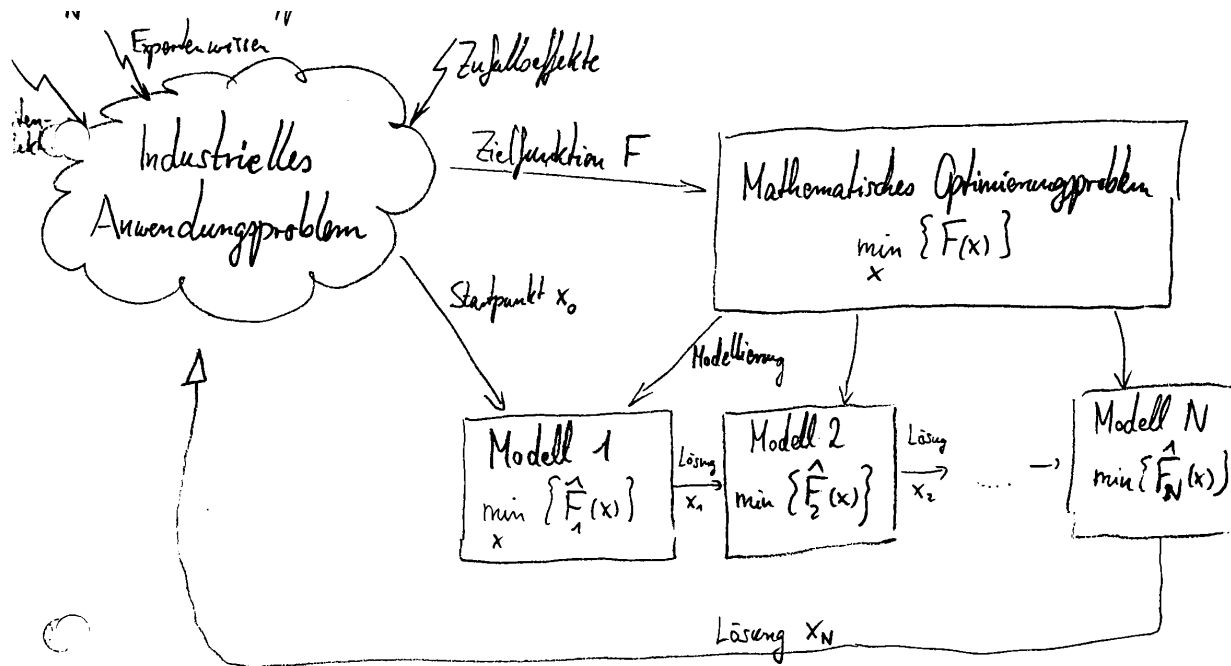


Abbildung 3.11: Modellierung bei nichtlinearen Optimierungsproblemen

3.3.2 Zweiphasige Optimierung

Es existieren zahlreiche nichtlineare Optimierungsverfahren, die lokale Minimierer sehr effizient und schnell berechnen können (siehe Kapitel 4). Man geht von der Annahme aus, dass ein solches Verfahren den zum Startpunkt nächstliegenden lokalen Minimierer bestimmt. Da Methoden, die globale Minimierer suchen, stochastische Anteile enthalten, besitzen sie in der Regel schlechte lokale Konvergenzeigenschaften.

Aus diesen Gründen liegt bei der globalen Optimierung ein hybrider Ansatz nahe, nämlich die **zweiphasige Optimierung**.

Phase I: Eine stochastische Methode sucht global nach Nachbarschaften von lokalen Minimierern.

Phase II: Eine lokale Methode bestimmt die lokalen Minimierer, startend bei den in Phase I erzeugten ‘Nachbarschaftspunkten’.

Dabei können Phase I und Phase II auch verschränkt sein, indem etwa vom stochastischen Verfahren bei jedem ‘interessanten’ neuen Punkt die lokale Methode zur Berechnung des Minimierers gestartet wird. Im primitivsten Fall kann Phase I aus einer ‘Monte-Carlo-Suche’ (zufällige Erzeugung von Punkten) bestehen. Eine weitaus performantere Methode wird bei Schäffler [Sch95] beschrieben. Weitere Phase-I-Verfahren sind ‘Simulated Annealing’ und ‘Genetische Algorithmen’.

3.3.3 Line-Search-Methoden

Die größte und bekannteste Klasse von lokalen Optimierungsverfahren bilden die **Line-Search-Methoden**, die ausgehend von einem Iterationspunkt x_j den nächsten Iterationspunkt entlang einer

Halbgeraden bestimmen:

$$x_{j+1} := x_j - \sigma_j s_j,$$

wobei $F(x_{j+1}) < F(x_j)$, siehe Abbildung 3.12.

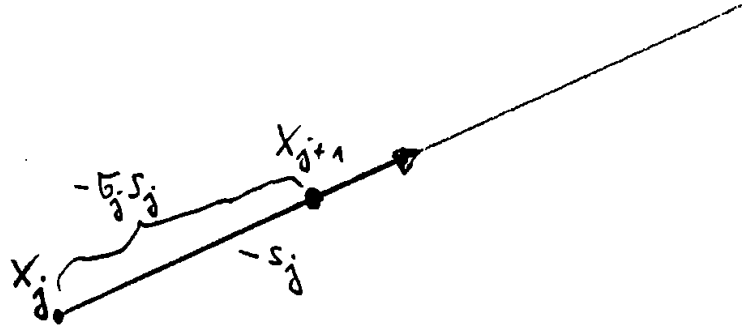


Abbildung 3.12: Prinzip der Line-Search-Methoden

Man wählt dazu eine **Suchrichtung** $s_j \in \mathbb{R}^n$, die die Richtung der Geraden bestimmt, und eine **Schrittweite** $\sigma_j > 0$, die den Abstand von x_{j+1} zu x_j auf der gewählten Halbgeraden bestimmt.

Definiert man bei gegebenen $s_j \in \mathbb{R}^n$ die Abbildung

$$\varphi_j : \mathbb{R}_0^+ \rightarrow \mathbb{R}, \sigma \mapsto \varphi_j(\sigma) := F(x_j - \sigma s_j),$$

so lässt sich immer dann ein $\sigma_j > 0$ finden mit $F(x_{j+1}) = \varphi_j(\sigma_j) < \varphi_j(0) = F(x_j)$, wenn $\varphi_j'(0) < 0$, also

$$0 > \varphi_j'(0)|_{\sigma=0} = \frac{d}{d\sigma} (F(x_j - \sigma s_j))|_{\sigma=0} = \left(-\nabla F(x_j - \sigma s_j)^\top s_j \right)|_{\sigma=0} = -\nabla F(x_j)^\top s_j.$$

Daher fordert man von den Suchrichtungen s_j die Eigenschaft

$$\nabla F(x_j)^\top s_j > 0.$$

s_j heißt dann auch **Abstiegsrichtung**.

Ein allgemeines Line-Search-Verfahren lässt sich durch Algorithmus 3.1 formulieren.

Seien $\{s_j\}_{j \in \mathbb{N}}$ und $\{\sigma_j\}_{j \in \mathbb{N}}$ durch den Algorithmus 3.1 erzeugte Folgen³.

- $\{s_j\}_{j \in \mathbb{N}}$ heißt eine **zulässige** Folge von Suchrichtungen, wenn gilt:

$$\text{Aus } \lim_{j \rightarrow \infty} \frac{\nabla F(x_j)^\top s_j}{\|s_j\|_2} = 0 \quad \text{folgt} \quad \lim_{j \rightarrow \infty} \|\nabla F(x_j)\|_2 = 0. \quad (3.3)$$

- $\{\sigma_j\}_{j \in \mathbb{N}}$ heißt eine **zulässige** Folge von Schrittweiten, wenn gilt:

$$\text{Aus } \lim_{j \rightarrow \infty} (F(x_j - \sigma_j s_j) - F(x_j)) = 0 \quad \text{folgt} \quad \lim_{j \rightarrow \infty} \frac{\nabla F(x_j)^\top s_j}{\|s_j\|_2} = 0. \quad (3.4)$$

³Nur in Ausnahmefällen, die hier nicht betrachtet werden, wird dieser (theoretische) Algorithmus nach endlich vielen Iterationen stoppen.

Line-Search: **Eingang** F, x_0 ; **Ausgang** $\{x_j\}_{j \in \mathbb{N}}, \{s_j\}_{j \in \mathbb{N}}, \{\sigma_j\}_{j \in \mathbb{N}}$;

Schritt 0: Initialisierung:

Wähle $x_0 \in \mathbb{R}^n$, setze $j := 0$ und **gehe** zu Schritt 1.

Schritt 1: Berechnung der Suchrichtung

falls $\nabla F(x_j) = 0$:

STOP mit Ergebnis x_j .

sonst:

Bestimme ein lokales Modell der Zielfunktion und wähle durch dessen Optimierung ein s_j mit $\nabla F(x_j)^\top s_j > 0$.

Gehe zu Schritt 2.

Schritt 2: Berechnung der Schrittweite

Wähle $\sigma_j > 0$ mit $F(x_j - \sigma_j s_j) < F(x_j)$.

Gehe zu Schritt 3.

Schritt 3: Update

Setze $x_{j+1} := x_j - \sigma_j s_j$.

Ersetze j durch $j + 1$ und **gehe** zu Schritt 1.

Algorithmus 3.1: Allgemeines Schema der Line-Search-Methoden

Damit lässt sich nun ein allgemein gültiger Konvergenzsatz für Line-Search-Methoden formulieren.

Satz 3.9 (Konvergenz von Line-Search-Methoden)

Die Zielfunktion $F : \mathbb{R}^n \rightarrow \mathbb{R}$ sei stetig differenzierbar und für ein gegebenes $x_0 \in \mathbb{R}^n$ sei die **Niveaumenge**

$$N_0 := \left\{ x \in \mathbb{R}^n \mid F(x) \leq F(x_0) \right\}$$

kompakt. Die Folgen $\{x_j\}_{j \in \mathbb{N}_0}$, $\{s_j\}_{j \in \mathbb{N}}$ und $\{\sigma_j\}_{j \in \mathbb{N}}$ seien durch Algorithmus 3.1 erzeugt, $\{s_j\}_{j \in \mathbb{N}}$ und $\{\sigma_j\}_{j \in \mathbb{N}}$ seien zulässig und die Abbruchbedingung $\nabla F(x_j) = 0$ werde nie erfüllt. Dann gilt:

(i)

$$F(x_{j+1}) < F(x_j), \quad \text{für alle } j \in \mathbb{N}_0.$$

(ii)

$$\{x_j\}_{j \in \mathbb{N}_0} \subset N_0.$$

(iii)

$$\lim_{j \rightarrow \infty} \|\nabla F(x_j)\|_2 = 0.$$

(iv) Für jeden Häufungspunkt $\bar{x}$ von $\{x_j\}_{j \in \mathbb{N}_0}$ gilt:

$$\nabla F(\bar{x}) = 0.$$

- (v) Wenn $\bar{x}$ ein isolierter Häufungspunkt von $\{x_j\}_{j \in \mathbb{N}_0}$ ist und $\lim_{j \rightarrow \infty} \|x_{j+1} - x_j\| = 0$, dann konvergiert $\{x_j\}_{j \in \mathbb{N}_0}$ gegen $\bar{x}$.

Beweis. Siehe [RK89].

□

Betrachtet man eine Optimierungsmethode, die die Bauart von Algorithmus 3.1 hat, so ist für die obigen Konvergenzaussagen nur die Zulässigkeit der Suchrichtungs- und Schrittweitenfolgen zu zeigen.

3.3.4 (Lokale) Konvergenzgeschwindigkeit

Besitzt ein Verfahren die notwendigen prinzipiellen Konvergenzeigenschaften, so erwünscht man zusätzlich eine möglichst hohe **Konvergenzgeschwindigkeit**. Zur Charakterisierung der Konvergenzgeschwindigkeit betrachte man eine von einem Algorithmus erzeugte Folge $\{x_j\}_{j \in \mathbb{N}_0}$, die gegen einen Minimierer $\bar{x}$ konvergiere. Die Folge $\{x_j\}_{j \in \mathbb{N}_0}$ heißt

- **linear konvergent**, falls es ein $\alpha \in \mathbb{R}$ mit $0 \leq \alpha < 1$ und ein $j_0 \in \mathbb{N}$ gibt, so dass

$$\|x_{j+1} - \bar{x}\| \leq \alpha \|x_j - \bar{x}\|, \quad \text{für alle } j \geq j_0.$$

- **superlinear konvergent**, falls es eine Nullfolge⁴ $\{\alpha_j\}_{j \in \mathbb{N}_0}$ gibt mit $\alpha_j \geq 0$, so dass

$$\|x_{j+1} - \bar{x}\| \leq \alpha_j \|x_j - \bar{x}\|, \quad \text{für alle } j \in \mathbb{N}_0.$$

- **von p -ter Ordnung konvergent** ($p > 1$), falls es ein $\alpha \geq 0$ und ein $j_0 \in \mathbb{N}$ gibt, so dass

$$\|x_{j+1} - \bar{x}\| \leq \alpha \|x_j - \bar{x}\|^p, \quad \text{für alle } j \geq j_0.$$

Ist $p = 2$, so spricht man von **quadratischer Konvergenz**.

Für Line-Search-Methoden existiert die folgende hilfreiche Aussage:

Lemma 3.10 (Dennis-Moré)

Die Zielfunktion $F : \mathbb{R}^n \rightarrow \mathbb{R}$ sei stetig differenzierbar und für ein gegebenes $x_0 \in \mathbb{R}^n$ sei die Niveaumenge $N_0 := \{x \in \mathbb{R}^n \mid F(x) \leq F(x_0)\}$ kompakt. Die Folgen $\{x_j\}_{j \in \mathbb{N}_0}$, $\{s_j\}_{j \in \mathbb{N}}$ und $\{\sigma_j\}_{j \in \mathbb{N}}$ seien durch Algorithmus 3.1 erzeugt, $\{s_j\}_{j \in \mathbb{N}}$ und $\{\sigma_j\}_{j \in \mathbb{N}}$ seien zulässig und die Abbruchbedingung $\nabla F(x_j) = 0$ werde nie erfüllt. $\{x_j\}_{j \in \mathbb{N}_0}$ konvergiere gegen ein $\bar{x} \in \mathbb{R}^n$ und $\nabla^2 F(\bar{x})$ sei positiv-definit⁵. Dann gilt:

Die Folge $\{x_j\}_{j \in \mathbb{N}_0}$ konvergiert superlinear gegen $\bar{x}$ genau dann, wenn

$$\lim_{j \rightarrow \infty} \frac{\|\nabla F(x_j) - \sigma_j \nabla^2 F(\bar{x}) s_j\|}{\|\nabla F(x_j)\|} = 0. \quad (3.5)$$

⁴d.h. $\lim_{j \rightarrow \infty} \alpha_j = 0$.

⁵Damit ist $\bar{x}$ ein isolierte lokaler Minimierer.

Beweis. Siehe [Kos93, RK89].

□

Weiterführende Literatur zur Theorie nichtlinearer Optimierungsprobleme ist z.B. in [Kos93, Lue89, Neu75a, RK89] zu finden.

Kapitel 4

Newton-Methode, Quasi-Newton-Methoden und Newton-ähnliche Methoden

4.1 Methode des steilsten Abstiegs (Steepest Descent)

Da $-\nabla F(x_j)$ die Richtung des steilsten Abstiegs an der Stelle $x_j \in \mathbb{R}^n$ ist, liegt es zunächst sehr nahe, vom Iterationspunkt x_j aus in Richtung $-\nabla F(x_j)$ nach dem nächsten Iterationspunkt x_{j+1} zu suchen. Damit ergibt sich ein Line-Search-Algorithmus mit Suchrichtung

$$s_j := \nabla F(x_j).$$

s_j ist trivialerweise eine Abstiegsrichtung¹ und $\{s_j\}_{j \in \mathbb{N}}$ ist trivialerweise zulässig.

Eine zulässige Folge $\{\sigma_j\}_{j \in \mathbb{N}}$ von Schrittweiten $\sigma_j > 0$ entsteht zum Beispiel durch Verwendung der ‘**Regel der optimalen Schrittweite**’, d.h. σ_j sei ein Minimierer des eindimensionalen Minimierungsproblems

$$\min_{\sigma} \left\{ \varphi_j(\sigma) \mid \sigma > 0 \right\},$$

mit

$$\varphi_j : \mathbb{R}_0^+ \rightarrow \mathbb{R}, \quad \sigma \mapsto \varphi_j(\sigma) = F(x_j - \sigma s_j),$$

siehe Abbildung 4.1.

Da

$$\varphi_j'(0) = -\nabla F(x_j)^\top s_j < 0,$$

fällt die Kurve bei $\sigma = 0$.

Die resultierende **Methode des steilsten Abstiegs** (Steepest Descent) ist allerdings auch mit der optimalen Schrittweite nur **linear konvergent** und für den praktischen Einsatz i.a. nicht geeignet. Der Grund wird offensichtlich, wenn man das lokale Modell Q_j der Zielfunktion betrachtet, auf

1

$$\nabla F(x_j)^\top s_j = \|\nabla F(x_j)\|^2 > 0, \quad \text{für } \nabla F(x_j) \neq 0.$$

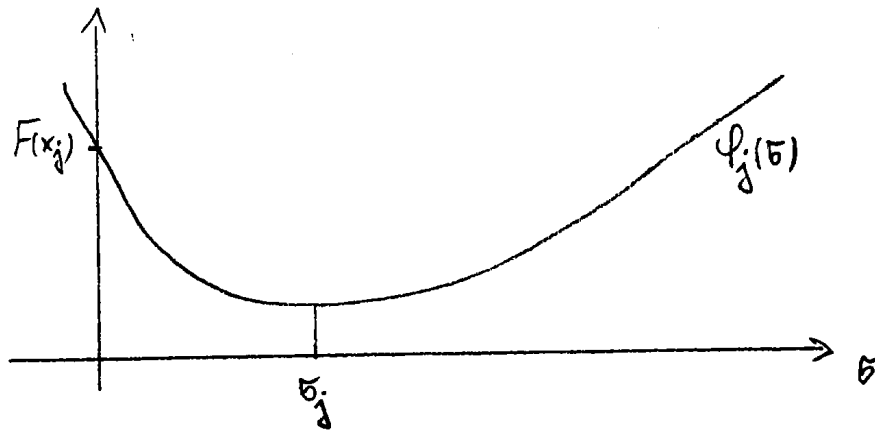


Abbildung 4.1: Optimale Schrittweite

dem die Methode des steilsten Abstiegs beruht:

$$Q_j(x) := F(x_j) + \nabla F(x_j)^\top (x - x_j) + \frac{1}{2} (x - x_j)^\top I_n (x - x_j),$$

$$\nabla Q_j(x) = \nabla F(x_j) + I_n (x - x_j),$$

$$\nabla^2 Q_j(x) = I_n.$$

Der Punkt

$$\hat{x}_{j+1} := x_j - \nabla F(x_j)$$

ist der Minimierer von $\min_{x \in \mathbb{R}^n} \{Q_j(x)\}$ und daher

$$s_j := x_j - \hat{x}_{j+1} = \nabla F(x_j).$$

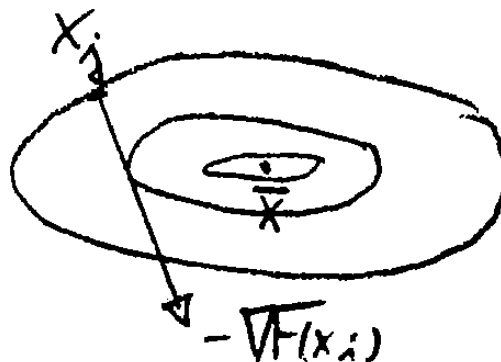


Abbildung 4.2: Steilster Abstieg (Steepest Descent)

Das lokale quadratische Modell Q_j von F bei x_j verwendet keine Krümmungseigenschaften von F , sondern ersetzt diese durch die Einheitsmatrix I_n (vergleiche die Illustration von [Abbildung 4.2](#)).

4.2 Newton-Methode

Unter der Voraussetzung, dass die Zielfunktion F zweimal stetig differenzierbar ist, lässt sich das quadratische Modell der Methode des steilsten Abstiegs durch Verwendung von Krümmungsinformationen entscheidend verbessern:

$$\begin{aligned} Q_j(x) &:= F(x_j) + \nabla F(x_j)^\top (x - x_j) + \frac{1}{2}(x - x_j)^\top \nabla^2 F(x_j)(x - x_j), \\ \nabla Q_j(x) &= \nabla F(x_j) + \nabla^2 F(x_j)(x - x_j), \\ \nabla^2 Q_j(x) &= \nabla^2 F(x_j). \end{aligned}$$

Damit ist Q_j das Taylorpolynom 2ten Grades von F bei x_j .

Ist $\nabla^2 F(x_j) = \nabla^2 Q_j(x_j)$ positiv-definit, so besitzt das Modellproblem $\min_{x \in \mathbb{R}^n} \{Q_j(x)\}$ den eindeutigen globalen Minimierer

$$\hat{x}_{j+1} := x_j - \underbrace{(\nabla^2 F(x_j))^{-1} \nabla F(x_j)}_{=: s_j}.$$

D.h. s_j ist die Lösung des Gleichungssystems

$$\nabla^2 F(x_j)s_j = \nabla F(x_j).$$

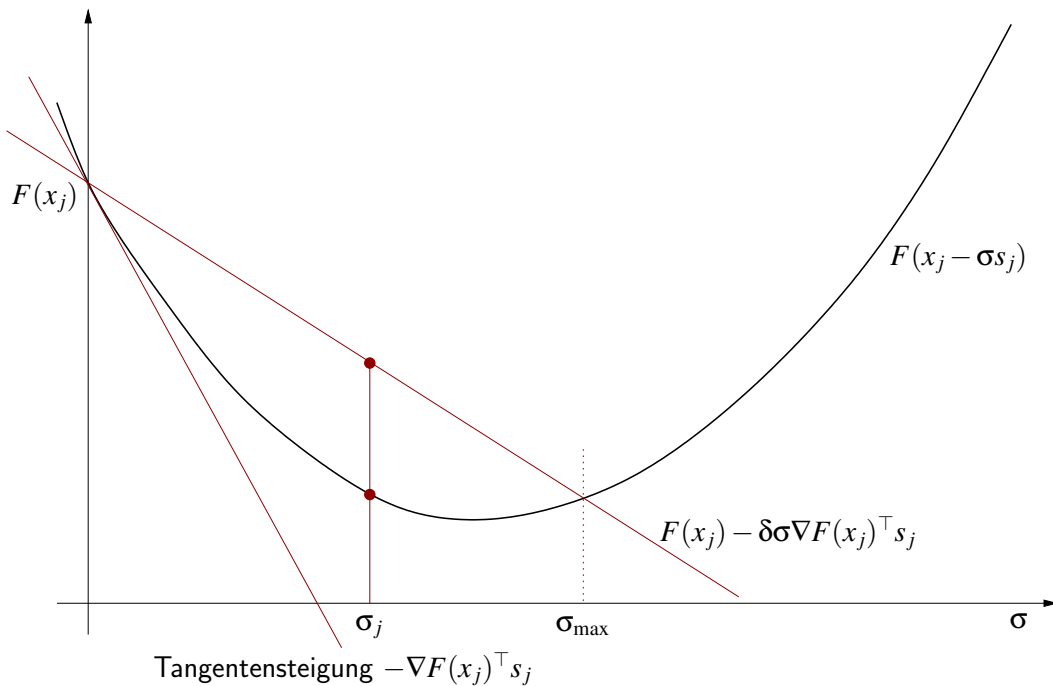


Abbildung 4.3: Schrittweitenwahl nach Armijo-Goldstein

Zur Sicherung der globalen Konvergenz ist wiederum eine Schrittweitenwahl notwendig. Statt der aufwendigen Regel der optimalen Schrittweite kann man für ein fest gewähltes $\delta \in]0, \frac{1}{2}[$ auch die **‘Regel von Armijo-Goldstein’** verwenden:

Wähle $\sigma_j = \left(\frac{1}{2}\right)^{v_j}$ so, dass $v_j \in \mathbb{N}_0$ die kleinste Zahl ist mit

$$F(x_j) - F\left(x_j - \left(\frac{1}{2}\right)^{v_j} s_j\right) \geq \delta \left(\frac{1}{2}\right)^{v_j} \nabla F(x_j)^\top s_j, \quad (4.1)$$

vergleiche Abbildung 4.3. Zur Sicherung der Konvergenz wird also keine beliebig kleine Abnahme der Zielfunktionswerte gefordert, sondern eine Abnahme um mindestens

$$\delta \sigma_j \nabla F(x_j)^\top s_j.$$

Bei einem implementierten Verfahren muss die Iteration nach einer endlichen Anzahl von Schritten beendet werden. Für ein $\varepsilon > 0$ wird daher die Abbruchbedingung $\|\nabla F(x_j)\| = 0$ in $\|\nabla F(x_j)\| < \varepsilon$ modifiziert ('gut genug'). Algorithmus 4.1 beschreibt nun die **Newton-Methode** mit Armijo-Goldstein-Schrittweitenwahl.

Newton-Methode: **Eingang** $F, x_0, \varepsilon, \gamma, \kappa$; **Ausgang** $x_j, F(x_j), \nabla F(x_j), \nabla^2 F(x_j)$;

Schritt 0: Initialisierung:

Wähle $x_0 \in \mathbb{R}^n$, $\varepsilon > 0$, $\gamma > 0$ und $0 < \kappa < 1$.

Setze $j := 0$ und **gehe** zu Schritt 1.

Schritt 1: Berechnung der Suchrichtung

falls $\|\nabla F(x_j)\| < \varepsilon$:

STOP mit Ergebnis x_j .

sonst:

falls das Gleichungssystem $\nabla^2 F(x_j)s = \nabla F(x_j)$ Lösungen besitzt,
dann sei $\hat{s}_j$ eine der Lösungen.

sonst:

$\hat{s}_j := \nabla F(x_j)$.

falls $\|\hat{s}_j\| \geq \gamma \|\nabla F(x_j)\|$ und $\nabla F(x_j)^\top \hat{s}_j \geq \min \{ \kappa, \|\nabla F(x_j)\| \} \|\nabla F(x_j)\| \|\hat{s}_j\|$
 $s := \hat{s}_j$.

sonst:

$\hat{s}_j := \nabla F(x_j)$.

Gehe zu Schritt 2.

Schritt 2: Berechnung der Schrittweite

Wähle σ_j nach der Armijo-Goldstein-Regel (4.1).

Gehe zu Schritt 3.

Schritt 3: Update

Setze $x_{j+1} := x_j - \sigma_j s_j$.

Ersetze j durch $j + 1$ und **gehe** zu Schritt 1.

Algorithmus 4.1: Schema der Newton-Methode

- Bei der Berechnung der Suchrichtung wurden Maßnahmen zur Sicherung der globalen Konvergenz getroffen, die im wesentlichen die **Newton-Richtung** $(\nabla^2 F(x_j))^{-1} \nabla F(x_j)$ durch die Gradienten-Richtung $\nabla F(x_j)$ ersetzt, falls technische Randbedingungen verletzt werden.

Im Einzugsbereich eines lokalen Minimierers $\bar{x}$ mit positiv-definiten Hessematrix $\nabla^2 F(\bar{x})$ wird nach endlich vielen Schritten stets $s_j = (\nabla^2 F(x_j))^{-1} \nabla F(x_j)$ gewählt, falls

$$\gamma < \min \left\{ 1, \frac{2}{3\lambda_{\max}(\nabla^2 F(x_j))} \right\}.$$

Insgesamt erzeugt Algorithmus 4.1 eine zulässige Folge $\{s_j\}_{j \in \mathbb{N}}$ von Suchrichtungen gemäß (3.3).

- Bei Konvergenz gegen einen solchen Minimierer wird nach endlich vielen Schritten stets $\sigma_j = 1$ gewählt (ungedämpftes Newton-Verfahren). Die mit der Armijo-Goldstein-Regel erzeugte Folge $\{\sigma_j\}_{j \in \mathbb{N}}$ von Schrittweiten ist zulässig gemäß (3.4). Algorithmus 4.1 ist eine Spezialisierung von Algorithmus 3.1 und erfüllt die Voraussetzungen von Satz 3.9 auf Seite 44 zur Konvergenz von Line-Search-Verfahren.
- Die Konvergenzgeschwindigkeit ist **superlinear**. Ist die zweite Ableitung von F zudem Lipschitz-stetig auf einer Umgebung von $\bar{x}$ (hinreichend ist die dreimalige stetige Differenzierbarkeit), so konvergiert die Newton-Methode sogar **quadratisch**.
- Es existieren zahlreiche Varianten der vorgestellten Newton-Methode, die oft alternative Strategien zur Behandlung nicht-positiv-definiten Hessematrizen anwenden. **Gill, Murray, Wright** ersetzen etwa $\nabla^2 F(x_j)$ durch eine Matrix $H_j = \nabla^2 F(x_j) + D_j$, wobei D_j eine so gewählte Diagonalmatrix ist, dass H_j positiv-definit ist.
- Ein Nachteil der Newton-Methode ist der hohe numerische Aufwand zur Berechnung der Hessematrix, insbesondere bei hohen Dimensionen n . Falls Ableitungen von F durch numerische Differentiation bestimmt werden müssen, kommt die Unsicherheit der Approximation der Hessematrix hinzu (mit Hilfe des sog. Automatischen Differenzierens lässt sich dieses Problem aber bisweilen umgehen).

4.3 Quasi-Newton-Methoden

Vielfach besteht der Wunsch, den numerischen Aufwand zur Berechnung der Hessematrix beim Newton-Verfahren zu vermeiden, aber dennoch ein Verfahren mit superlinearer Konvergenzgeschwindigkeit zu verwenden. Man versucht dazu die bislang gewonnenen Informationen über die Zielfunktion (Funktionswerte und Gradienten) für eine Approximation der Hessematrix $\nabla^2 F(x_j)$ zu nutzen. Aus praktischen Erwägungen heraus werden Approximationen H_j der invertierten Hessematrix bestimmt, um die Suchrichtung s_j über

$$s_j := H_j \nabla F(x_j)$$

approximativ zur Newton-Richtung $(\nabla^2 F(x_j))^{-1} \nabla F(x_j)$ in einfacher Weise berechnen zu können. An jedem Iterationspunkt x_j wird also das lokale quadratische Modell

$$\begin{aligned} Q_j(x) &:= F(x_j) + \nabla F(x_j)^\top (x - x_j) + \frac{1}{2} (x - x_j)^\top H_j^{-1} (x - x_j), \\ \nabla Q_j(x) &= \nabla F(x_j) + H_j^{-1} (x - x_j), \\ \nabla^2 Q_j(x) &= H_j^{-1}, \end{aligned} \tag{4.2}$$

betrachtet. Da ein quadratisches Modell zwei Gradienten interpolieren kann, fordern die **Sekantenmethoden** neben den Eigenschaften $Q_{j+1}(x_{j+1}) = F(x_{j+1})$ und $\nabla Q_{j+1}(x_{j+1}) = \nabla F(x_{j+1})$ auch die Eigenschaft

$$\nabla F(x_j) \stackrel{!}{=} \nabla Q_{j+1}(x_j) = \nabla F(x_{j+1}) + H_{j+1}^{-1}(x_j - x_{j+1}),$$

also

$$H_{j+1}(\nabla F(x_j) - \nabla F(x_{j+1})) = x_j - x_{j+1}. \quad \textbf{‘Quasi-Newton-Gleichung’} \quad (4.3)$$

Eine **Update-Formel**, die H_{j+1} unter Verwendung von H_j und den neu gewonnenen Informationen über die Zielfunktion berechnet, soll natürlich so konstruiert werden, dass das resultierende Verfahren numerisch stabil ist und eine möglichst hohe Konvergenzgeschwindigkeit besitzt. In der Literatur werden (mindestens) zwei leicht widersprüchliche Definitionen für den Begriff **Quasi-Newton-Methode** gebraucht:

- a) Sekanten-Methoden, die auf obigem quadratischem Modell beruhen und die eine Update-Formel verwenden, so dass die Matrizen H_j die Quasi-Newton-Gleichung (4.3) erfüllen.
- b) Superlinear konvergente Line-Search-Methoden, d.h. Line-Search-Methoden, die die Dennis-Moré-Bedingung (3.5)

$$\lim_{j \rightarrow \infty} \frac{\|\nabla F(x_j) - \sigma_j \nabla^2 F(\bar{x}) s_j\|}{\|\nabla F(x_j)\|} = 0.$$

erfüllen (siehe Lemma 3.10 auf Seite 45).

Als Beispiel für die Konstruktion einer Update-Formel leiten wir eine **Rang-1-Korrekturformel** her, bei der in jedem Iterationsschritt eine symmetrische Rang-1-Matrix zu H_j addiert wird:

$$H_{j+1} := H_j + u_j u_j^\top, \quad \text{mit } u_j \in \mathbb{R}^n. \quad (4.4)$$

Zur Abkürzung (und Normierung) definieren wir

$$p_j := \frac{x_j - x_{j+1}}{\|x_j - x_{j+1}\|} \quad \text{und} \quad d_j := \frac{\nabla F(x_j) - \nabla F(x_{j+1})}{\|x_j - x_{j+1}\|}.$$

Damit lässt sich die Quasi-Newton-Gleichung (4.3) schreiben als

$$H_{j+1} d_j = p_j.$$

Nun wird in den Rang-1-Update-Ansatz (4.4) eingesetzt:

$$p_j = H_{j+1} d_j = H_j d_j + u_j u_j^\top d_j.$$

1.Fall: $p_j = H_j d_j$, d.h. $H_{j+1} := H_j$ erfüllt die QNG (4.3) (und somit $u_j := 0$).

2.Fall: $p_j \neq H_j d_j$. Somit ist

$$u_j = \underbrace{\frac{1}{u_j^\top d_j}}_{=: \alpha_j} (p_j - H_j d_j),$$

d.h. u_j ist ein Vielfaches von $p_j - H_j d_j$. Ersetzt man nun u_j durch $\alpha_j (p_j - H_j d_j)$, so erhält man

$$\alpha_j (p_j - H_j d_j) = \frac{1}{\alpha_j (p_j - H_j d_j)^\top d_j} (p_j - H_j d_j),$$

also

$$\alpha_j^2 = \frac{1}{(p_j - H_j d_j)^\top d_j}.$$

Somit erhalten wir die Update-Formel

$$H_{j+1} = H_j + \frac{(p_j - H_j d_j) (p_j - H_j d_j)^\top}{(p_j - H_j d_j)^\top d_j}. \quad (4.5)$$

Dieser Ansatz führt aber zu numerisch instabilen Updates, falls $(p_j - H_j d_j)^\top d_j$ klein, aber $\|p_j - H_j d_j\|$ groß ist.

Verwendet man daher **Rang-2-Updates**, so ergibt sich eine ganze Klasse (**Broyden-Klasse**) von Quasi-Newton-Verfahren, von denen das nachfolgend beschriebene BFGS-Verfahren das erfolgreichste ist.

4.4 BFGS-Methode

Die BFGS-Methode (nach Broyden, Fletcher, Goldfarb, Shanno - 1971) ist eine spezielle Sekanten-Methode mit Rang-2-Update. Wieder sei

$$p_j := \frac{x_j - x_{j+1}}{\|x_j - x_{j+1}\|} \quad \text{und} \quad d_j := \frac{\nabla F(x_j) - \nabla F(x_{j+1})}{\|x_j - x_{j+1}\|}.$$

Dann lautet die **BFGS-Updateformel**

$$H_{j+1} := H_j + \frac{d_j^\top p_j + d_j^\top H_j d_j}{(d_j^\top p_j)^2} p_j p_j^\top - \frac{p_j d_j^\top H_j + H_j d_j p_j^\top}{d_j^\top p_j}. \quad (4.6)$$

Falls H_j positiv-definit ist und $d_j^\top p_j > 0$, so lässt sich zeigen, dass auch H_{j+1} positiv-definit ist. Somit hat das lokale quadratische Modell (4.2) immer einen eindeutigen Minimierer (im Gegensatz zur Newton-Methode).

Da

$$p_j = \frac{\sigma_j s_j}{\|\sigma_j s_j\|} = \frac{s_j}{\|s_j\|},$$

ist die Forderung $d_j^\top p_j > 0$ äquivalent zur Bedingung

$$(\nabla F(x_j) - \nabla F(x_{j+1}))^\top s_j > 0 \quad \text{bzw.} \quad \nabla F(x_j - \sigma_j s_j)^\top s_j < \nabla F(x_j)^\top s_j. \quad (4.7)$$

Daher muss die Armijo-Goldstein-Regel zu Schrittweitenwahl so modifiziert werden, dass diese Bedingung erfüllt wird.

Man betrachte dazu δ_1 und δ_2 mit $0 < \delta_1 < \frac{1}{2}$ und $\delta_1 < \delta_2 < 1$. Weiter sei die Funktion ϕ_j wieder gegeben durch

$$\phi_j(\sigma) := F(x_j - \sigma s_j), \quad \phi_j'(\sigma) = -\nabla F(x_j - \sigma s_j)^\top s_j.$$

Die Bedingung (4.7) lässt sich verschärfen zu

$$\nabla F(x_j - \sigma s_j)^\top s_j \leq \delta_2 \nabla F(x_j)^\top s_j,$$

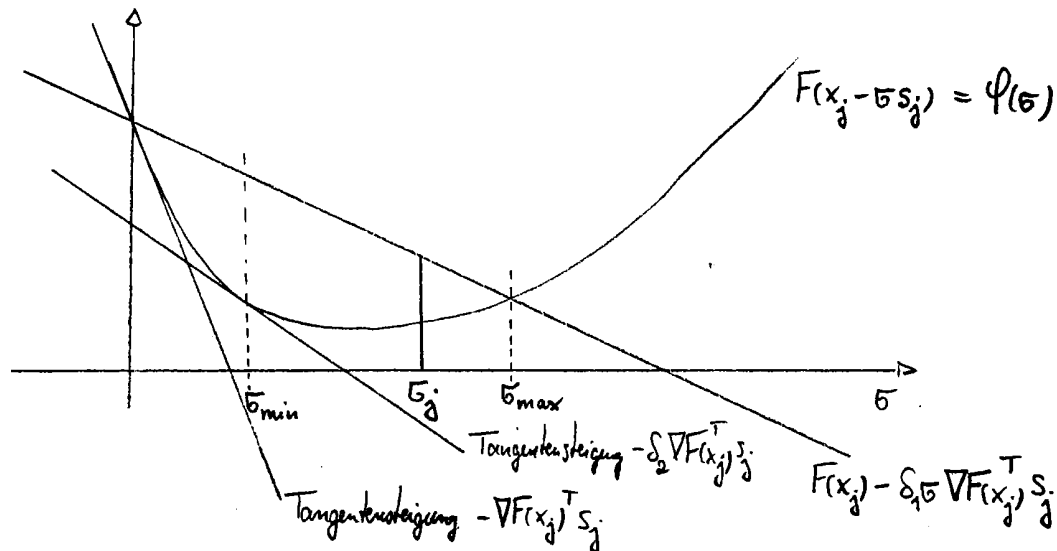


Abbildung 4.4: Schrittweitenwahl für die BFGS-Methode

was bedeutet, dass

$$\phi'_j(\sigma) \geq -\delta_2 \nabla F(x_j)^\top s_j.$$

Wie in Abbildung 4.4 visualisiert, lässt sich stets ein $\sigma_j > 0$ finden, so dass

$$F(x_j) - F(x_j - \sigma_j s_j) \geq \delta_1 \sigma_j \nabla F(x_j)^\top s_j \quad (\text{Armijo-Goldstein})$$

und

$$\nabla F(x_j - \sigma_j s_j)^\top s_j \leq \delta_2 \nabla F(x_j)^\top s_j, \quad (\text{Pos.Def. von } H_{j+1})$$

erfüllt werden.

Die gesamte Vorgehensweise ist in Algorithmus 4.2 zusammengefasst.

- Jede Suchrichtung ist eine Abstiegsrichtung, da H_j positiv-definit konstruiert war. In praktischen Anwendungen wird H_j oft nach einer festen Anzahl von Schritten als Einheitsmatrix gewählt ('Rücksetzen').
- Algorithmus 4.2 ist eine Spezialisierung von Algorithmus 3.1, und die erzeugten Folgen $\{s_j\}_{j \in \mathbb{N}}$ und $\{\sigma_j\}_{j \in \mathbb{N}}$ sind zulässig, d.h. das BFGS-Verfahren erfüllt die Voraussetzungen von Satz 3.9 auf Seite 44 zur Konvergenz von Line-Search-Verfahren.
- Bei Konvergenz gegen einen lokalen Minimierer $\bar{x}$ mit positiv-definiten Hessematrix $\nabla^2 F(\bar{x})$ und Lipschitz-Stetigkeit von F auf einer Umgebung von $\bar{x}$ (hinreichend ist die dreimal stetige Differenzierbarkeit) konvergiert die BFGS-Methode **superlinear** (Das Newton-Verfahren konvergiert unter gleichen Voraussetzungen quadratisch). Allerdings ist im allgemeinen

$$\lim_{j \rightarrow \infty} \|H_j - (\nabla^2 F(\bar{x}))^{-1}\| \neq 0.$$

BFGS-Methode: **Eingang** $F, x_0, \varepsilon, \delta_1, \delta_2$; **Ausgang** $x_j, F(x_j), \nabla F(x_j), H_j$;

Schritt 0: Initialisierung:

Wähle $x_0 \in \mathbb{R}^n$, $\varepsilon > 0$, $0 < \delta_1 < \frac{1}{2}$ und $\delta_1 < \delta_2 < 1$.

falls $\|\nabla F(x_0)\| < \varepsilon$:

STOP mit Ergebnis x_0 .

sonst:

Wähle eine positiv-definite Matrix H_0 .

i.d.R. die Einheitsmatrix

Setze $j := 0$ und **gehe** zu Schritt 1.

Schritt 1: Berechnung der Suchrichtung

Berechne $s_j := H_j \nabla F(x_j)$.

Gehe zu Schritt 2.

Schritt 2: Berechnung der Schrittweite

Berechne σ_j derart, dass gilt

- $F(x_j) - F(x_j - \sigma_j s_j) \geq \delta_1 \sigma_j \nabla F(x_j)^\top s_j$,
- $\nabla F(x_j - \sigma_j s_j)^\top s_j \leq \delta_2 \nabla F(x_j)^\top s_j$,
- $\sigma_j = 1$, falls möglich.

Gehe zu Schritt 3.

Schritt 3: Update

Setze $x_{j+1} := x_j - \sigma_j s_j$.

falls $\|\nabla F(x_{j+1})\| < \varepsilon$:

STOP mit Ergebnis x_{j+1} .

sonst:

$$p_j := \frac{x_j - x_{j+1}}{\|x_j - x_{j+1}\|}.$$

$$d_j := \frac{\nabla F(x_j) - \nabla F(x_{j+1})}{\|x_j - x_{j+1}\|}.$$

$$H_{j+1} := H_j + \frac{d_j^\top p_j + d_j^\top H_j d_j}{(d_j^\top p_j)^2} p_j p_j^\top - \frac{p_j d_j^\top H_j + H_j d_j p_j^\top}{d_j^\top p_j}.$$

Ersetze j durch $j + 1$ und **gehe** zu Schritt 1.

Algorithmus 4.2: Schema der BFGS-Methode

4.5 BNS-Methode

Die verschiedenen Varianten der Newton-Methode unterscheiden sich in der Regel durch die Wahl einer Ersatz-Suchrichtung, falls die Hessematrix der Zielfunktion am aktuellen Iterationspunkt nicht positiv-definit ist. Statt einer Line-Search-Methode wird nun das Konzept einer **Curve-Search-Methode** betrachtet [Stu96].

Sei $F : \mathbb{R}^n \rightarrow \mathbb{R}$ zweimal stetig differenzierbar und $x_0 \in \mathbb{R}^n$. Die **‘Kurve des steilsten Abstiegs’** ist definiert als die eindeutige Lösung $x : \mathbb{R}_0^+ \rightarrow \mathbb{R}^n$ des **Anfangswertproblems**

$$\dot{x}(t) = -\nabla F(x(t)), \quad \text{für } t > 0, \quad x(0) = x_0. \quad (4.8)$$

Falls die Kurve $x(t)$ beschränkt ist, dann ist $\lim_{t \rightarrow \infty} x(t)$ ein lokaler Minimierer von F .

Betrachtet man nun das quadratische Taylor-Polynom Q von F bei x_0 mit den Abkürzungen

$$g_0 := \nabla F(x_0) \quad \text{und} \quad H_0 := \nabla^2 F(x_0),$$

also

$$Q(x) := F(x_0) + g_0^\top (x - x_0) + \frac{1}{2} (x - x_0)^\top H_0 (x - x_0),$$

$$\nabla Q(x) = g_0 + H_0(x - x_0),$$

dann ist die Kurve des steilsten Abstiegs im quadratischen Modell gegeben durch

$$\dot{x}(t) = -H_0 x(t) + H_0 x_0 - g_0 \quad \text{für } t > 0, \quad x(0) = x_0. \quad (4.9)$$

Während die Lösung von (4.8) im allgemeinen nicht berechenbar ist, lautet die Lösung des Ersatzproblems (4.9)

$$x(t) = x_0 - M(t)g_0, \quad \text{für } t \geq 0, \quad (4.10)$$

mit

$$M(t) = \exp(-tH_0) \int_0^t \exp(\tau H_0) d\tau = \exp(-tH_0) \sum_{k=0}^{\infty} \frac{1}{(k+1)!} t^{k+1} H_0^k.$$

Nun betrachte man die Eigenwerte $\lambda_1, \dots, \lambda_n$ von H_0 und das zugehörige orthonormale Eigensystem $v_1, \dots, v_n$. $V := (v_1, \dots, v_n)$ ist dann eine orthogonale Matrix und

$$V^\top H_0 V = \text{diag}(\lambda_1, \dots, \lambda_n).$$

Dann gilt

$$V^\top M(t) V = \text{diag}(\mu(t, \lambda_1), \dots, \mu(t, \lambda_n)), \quad \text{mit} \quad \mu(t, \lambda) := \begin{cases} \frac{1}{\lambda} (1 - \exp(-t\lambda)), & \text{für } \lambda \neq 0, \\ t, & \text{für } \lambda = 0. \end{cases}$$

Definiert man noch $\beta_i := v_i^\top g_0$ für $1 \leq i \leq n$, dann lässt sich (4.10) umformen in

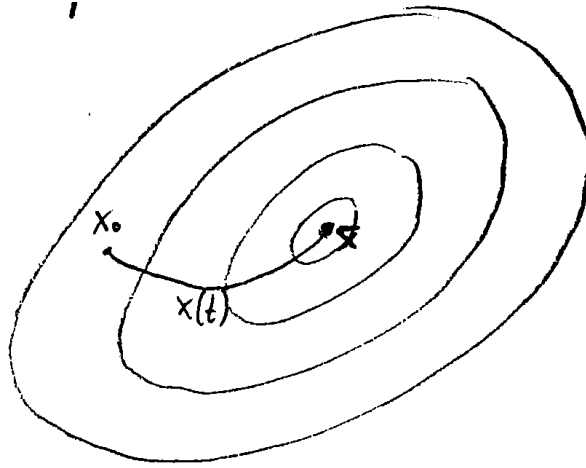
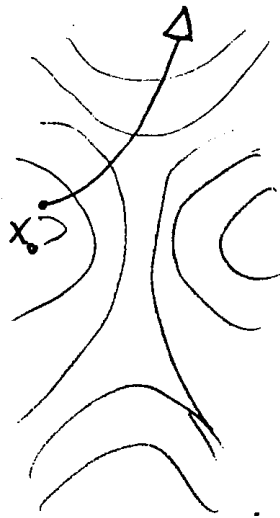
$$x(t) = x_0 - \sum_{i=1}^n \mu(t, \lambda_i) \beta_i v_i, \quad \text{für } t \geq 0, \quad (4.11)$$

Abbildung 4.5 zeigt, wie diese Kurve $x(t)$ in den Minimierer läuft bei positiv-definitem H_0 , während bei mindestens einem negativen Eigenwert diese Kurve ‘im Unendlichen verschwindet’, wie Abbildung 4.6 illustriert.

Das lokalen Modell Q der Zielfunktion approximiert diese Funktion nur in einer Umgebung des Punktes x_0 (allgemein: x_j) ‘hinreichend gut’. Analog zur Vorgehensweise bei den Line-Search-Methoden stellt sich wieder die Frage, wo groß bzw. klein der Steuerparameter t gewählt werden muss, damit $x(t)$ ein guter nächster Iterationspunkt ist. Diese Schrittweitensteuerung beim BNS-Verfahren ist kompliziert und wird im Folgenden konzeptuell vorgestellt.

Zunächst betrachten wir den quadratischen Abstand $a(t)$ von $x(t)$ zu x_0 , d.h.

$$a(t) := \|x(t) - x_0\|_2^2 = g_0^\top M^2(t) g_0 = \sum_{i=1}^n \mu^2(t, \lambda_i) \beta_i^2, \quad \text{für } t \geq 0.$$


 Abbildung 4.5: $x(t)$ für positiv-definites H_0

 Abbildung 4.6: $x(t)$ für H_0 mit mindestens einem negativen Eigenwert

Es lässt sich zeigen, dass $a(t)$ streng monoton steigend ist, d.h. $x(t)$ entfernt sich von x_0 mit wachsendem t ($x(t)$ beschreibt keine Schleifen). Als Maximaldistanz definieren wir

$$s_{\max}(x_0) := \lim_{t \rightarrow \infty} \sqrt{a(t)} \in [0, +\infty].$$

Mit Ausnahme einiger Spezialfälle (wenn einige $\beta_i = 0$ sind) ist $s_{\max}(x_0)$ nur endlich für positiv-definite Hessematrizen H_0 . In diesem Fall gewinnt man den Minimierer von Q natürlich durch einen Newton-Schritt:

$$\lim_{t \rightarrow \infty} x(t) = x_0 - \sum_{i=1}^n \frac{\beta_i}{\lambda_i} v_i = x_0 - H_0^{-1} g_0.$$

Jetzt kann man eine neue Parametrisierung s für die Kurve des steilsten Abstiegs einführen:

$$s: [0, +\infty] \rightarrow [0, s_{\max}(x_0)], \quad t \mapsto s(t) = \sqrt{a(t)}.$$

s ist eine bijektive Abbildung und wird **Abstands-Parametrisierung** genannt. Weiter betrachten wir die Umkehrabbildung τ von s , also

$$\tau: [0, s_{\max}(x_0)] \rightarrow [0, +\infty], \quad s \mapsto \tau(s) = a^{-1}(s^2).$$

mit $\tau(s_{\max}(x_0)) := +\infty$.

Mit Hilfe der neuen Parametrisierung definieren wir die relative Veränderung der Zielfunktion F entlang der Kurve $x(\tau(s))$, startend bei x_0 , als

$$d_{x_0}(s) := F(x_0) - F(x(\tau(s))), \quad \text{für } s \in [0, s_{\max}(x_0)].$$

Analog betrachten wir die Veränderung im quadratischen Modell Q :

$$\hat{d}_{x_0}(s) := F(x_0) - Q(x(\tau(s))), \quad \text{für } s \in [0, s_{\max}(x_0)].$$

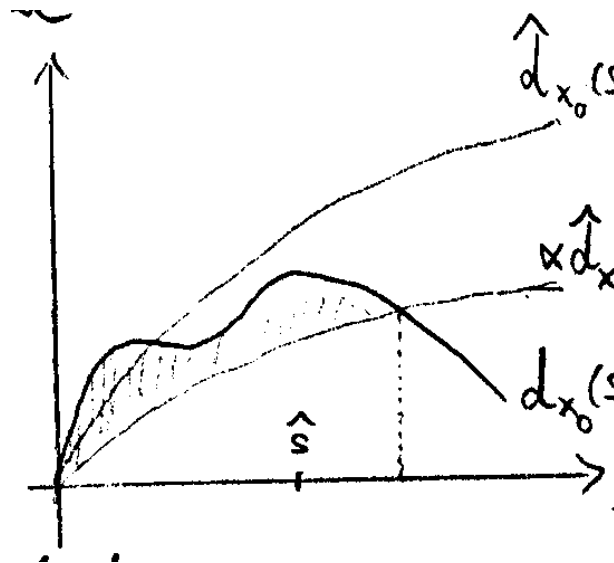


Abbildung 4.7: Schrittweitenwahl für die BNS-Methode

Als Kriterium zur Schrittweitensteuerung vergleicht man nun die ‘tatsächliche Abnahme’ $d_{x_0}(s)$ der Zielfunktion mit der ‘vorhergesagten Abnahme’ $\hat{d}_{x_0}(s)$ im Taylor-Modell Q , siehe Abbildung 4.7. Man akzeptiert eine Schrittweite $\hat{s}$ dann, wenn

$$d_{x_0}(\hat{s}) \geq \alpha \hat{d}_{x_0}(\hat{s})$$

mit einer Konstanten $0 < \alpha < 1$. Der nächste Iterationspunkt ist dann $x(\tau(\hat{s}))$.

- Eine Implementierung muss noch eine Reihe von technischen Randbedingungen berücksichtigen. Die Abbildung τ ist nur implizit gegeben und daher wäre eine (weitgehend) exakte Berechnung von $\tau(s)$ zu einem gegebenen s numerisch zu aufwendig. Für eine Schrittweitensteuerung nach der prinzipiellen Armijo-Goldstein-Idee genügt aber eine numerisch einfache Approximation $\text{Approx}_\gamma(s) > 0$ von $\tau(s)$ mit $0 < \gamma < \frac{1}{3}$ und der Eigenschaft

$$\frac{\sqrt{a(\text{Approx}_\gamma(s))}}{s} \in [1 - \gamma, 1 + \gamma],$$

da dann gilt

$$\text{Approx}_\gamma\left(\frac{s}{2}\right) < \text{Approx}_\gamma(s) < \text{Approx}_\gamma(2s).$$

- Wenn die vom BNS-Algorithmus erzeugte Iterationsfolge $\{x_j\}_{j \in \mathbb{N}_0}$ gegen ein $\bar{x} \in \mathbb{R}^n$ konvergiert mit $\nabla^2 F(\bar{x})$ positiv-definit, dann gibt es ein $M \in \mathbb{N}$ mit

$$x_{j+1} = x_j - (\nabla^2 F(x_j))^{-1} \nabla F(x_j), \quad \text{für alle } j \geq M,$$

d.h. es werden nur noch Newton-Schritte durchgeführt und das BNS-Verfahren konvergiert quadratisch unter geeigneten Voraussetzungen.

- Richtungen negativer Krümmung (Eigenvektoren zu negativen Eigenwerten der Hessematrix) werden vom BNS-Algorithmus durch Verwendung der Kurve des steilsten Abstiegs in natürlicher Weise berücksichtigt.

4.6 Trust-Region-Methode

Die Idee der Trust-Region-Methode ist verwandt mit der BNS-Methode. Man vertraut dem lokalen quadratischen Taylor-Modell Q_j zu einem Iterationspunkt x_j nur auf einer Kugel mit Mittelpunkt x_j und bestimmt auf dieser einen Minimierer, d.h. man berechnet $\hat{x}_{j+1}(\eta)$ als Lösung von

$$\min_{x \in \mathbb{R}^n} \left\{ Q_j(x) \mid \|x - x_j\|_2^2 \leq \eta \right\}, \quad \text{für ein } \eta > 0. \quad (4.12)$$

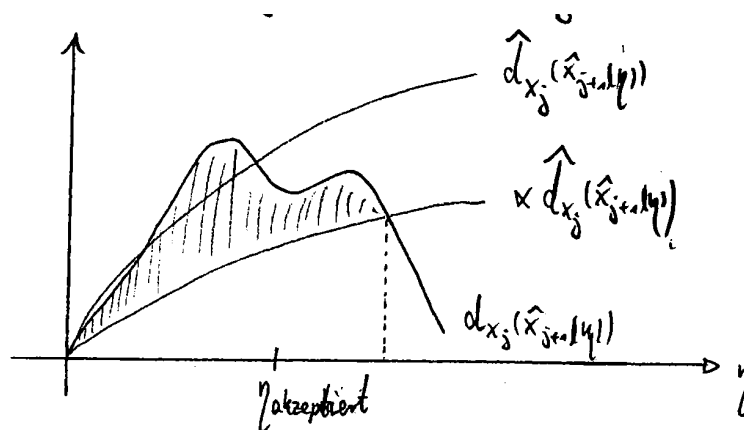


Abbildung 4.8: Schrittweisenwahl für die Trust-Region-Methode

Analog zum BNS-Verfahren betrachtet man die Abbildungen

$$d_{x_j} : \mathbb{R}^n \rightarrow \mathbb{R}, \quad x \mapsto F(x_j) - F(x),$$

$$\hat{d}_{x_j} : \mathbb{R}^n \rightarrow \mathbb{R}, \quad x \mapsto F(x_j) - Q_j(x) = \nabla F(x_j)^\top (x_j - x) - \frac{1}{2}(x_j - x)^\top \nabla^2 F(x_j)(x_j - x),$$

als die ‘tatsächliche Abnahme’ $d_{x_j}(\hat{x}_{j+1}(\eta))$ und die ‘vorhergesagten Abnahme’ $\hat{d}_{x_0}(\hat{x}_{j+1}(\eta))$, siehe Abbildung 4.8.

Der Parameter η (quadrierter Kugelradius) wird nun so gesteuert, dass für den Minimierer $\hat{x}_{j+1}(\eta)$ von (4.12) gilt

$$d_{x_0}(\hat{x}_{i+1}(\eta)) \geq \alpha \hat{d}_{x_0}(\hat{x}_{i+1}(\eta))$$

mit einer Konstanten $0 < \alpha < 1$.

Der nächste Iterationspunkt x_{j+1} wird dann als $\hat{x}_{j+1}(\eta_{\text{akzeptiert}})$ gesetzt und im nächsten Schritt wird zunächst die Akzeptanz des zuletzt bestimmten η getestet.

- Die Methode ist numerisch aufwendig, da in jedem Schritt ein oder mehrere quadratische Optimierungsprobleme mit Nebenbedingungen gelöst werden müssen.
- Unter geeigneten Voraussetzungen werden nach endlich vielen Schritten nur noch Newton-Schritte durchgeführt und das Trust-Region-Verfahren konvergiert quadratisch.

Literaturverzeichnis

- [Buc02] Anton Georg Buchmeier. *Modellierung und Simulation*. Vorlesungs-Skriptum, Technische Universität München, München, Juli 2002.
- [CW71] Lothar Collatz & Wolfgang Wetterling. *Optimierungsaufgaben*. Springer Verlag, Berlin, Heidelberg, New York, Zweite Auflage, 1971.
- [Ker87] Werner Kern. *Operations Research*. C.E.Poeschel Verlag, Stuttgart, 6te Auflage, 1987.
- [Knu97] Donald E. Knuth. *The Art of Computer Programming Vol II: Seminumerical Algorithms*. Addison-Wesley, Reading, Massachusetts, 1997.
- [Kos93] Peter Kosmol. *Methoden zur numerischen Behandlung nichtlinearer Gleichungen und Optimierungsaufgaben*. Teubner Verlag, Stuttgart, Zweite Auflage, 1993.
- [Lue89] David G. Luenberger. *Linear and Nonlinear Programming*. Addison-Wesley, Reading, Massachusetts, Zweite Auflage, 1989.
- [MM73] Heiner Müller-Merbach. *Operations Research*. Verlag Vahlen, München, Dritte Auflage, 1973.
- [Neu75a] Klaus Neumann. *Operations Research I*. Carl Hanser Verlag, München Wien, Erste Auflage, 1975.
- [Neu75b] Klaus Neumann. *Operations Research III*. Carl Hanser Verlag, München Wien, Erste Auflage, 1975.
- [Neu77] Klaus Neumann. *Operations Research II*. Carl Hanser Verlag, München Wien, Erste Auflage, 1977.
- [Nie84] Aribert Nieswandt. *Operations Research*. Verlag Neue Wirtschaftsbrieft, Herne/Berlin, Zweite Auflage, 1984.
- [PZ78] Joachim Piehler & Hans-Ulrich Zschiesche. *Simulationsmethoden*. Verlag Harri Deutsch, Thun, Frankfurt/Main, 1978.
- [RK89] Klaus Ritter & Christian Kredler. *Nichtlineare Optimierung*. Vorlesungs-Skriptum IAMS Nr. 1, Technische Universität München, München, 1989.
- [RR92] Bruno Riedmüller & Klaus Ritter. *Lineare und quadratische Optimierung*. Vorlesungs-Skriptum IAMS Nr. 2, Technische Universität München, München, Mai 1992.
- [Rei95] Christian Reinsch. *Numerische Mathematik 1 und 2*. Vorlesungs-Skriptum TUM-MATH-11-95-00-250/3.-FMI, Technische Universität München, München, November 1995.

- [Run86] Bodo Runzheimer. *Operations Research I*. Gabler Verlag, Wiesbaden, Dritte Auflage, 1986.
- [Run89] Bodo Runzheimer. *Operations Research II*. Gabler Verlag, Wiesbaden, Zweite Auflage, 1989.
- [SS94] Stefan Schäffler & Thomas F. Sturm. *Wahrscheinlichkeitstheorie und Statistik I für Mathematiker*. Vorlesungs-Skriptum IAMS Nr. 5, Technische Universität München, München, Oktober 1994.
- [SS95] Stefan Schäffler & Thomas F. Sturm. *Wahrscheinlichkeitstheorie und Statistik II für Mathematiker*. Vorlesungs-Skriptum IAMS Nr. 6, Technische Universität München, München, März 1995.
- [Sch95] Stefan Schäffler. *Global Optimization Using Stochastic Integration*. Theorie und Forschung. Roderer Verlag, Regensburg, 1995.
- [Sch02] Stefan Schäffler. *Projektmanagement B*. Vorlesungs-Skriptum, Universität der Bundeswehr München, Neubiberg, 2002.
- [Stu96] Thomas F. Sturm. *A Stepsize Control for the Botsaris-Newton Method*. Journal of Mathematical Analysis and Applications, **199**, 59–74, 1996.
- [Zie78] Ryszard Zieliński. *Erzeugung von Zufallszahlen*. Verlag Harri Deutsch, Thun, Frankfurt/Main, 1978.
- [Zim92] Hans-Jürgen Zimmermann. *Operations Research Methoden und Modelle*. Vieweg Verlag, Braunschweig, Wiesbaden, Zweite Auflage, 1992.