

**Skript zur Vorlesung  
Optimierung linearer Modelle  
Gültig ab Sommersemester 2010**

Prof. Dr. S. Dempe



## Inhaltsverzeichnis

Kapitel 0. Einleitung	5
0.1. Historische Entwicklung	5
0.2. Begriff des Operations Research	6
0.3. Einsatz der Methoden des OR	7
Kapitel 1. Lineare Optimierung	9
1.1. Einführung, Modell	9
1.2. Normalform	10
1.3. Graphische Lösung	11
1.4. Simplexalgorithmus für LOA	17
1.4.1. Wiederholung der grundlegenden Begriffe der linearen Algebra	17
1.4.2. Grundlagen des Simplexalgorithmus	19
1.4.3. Idee des Simplexalgorithmus	21
1.4.4. Der Simplexalgorithmus	22
1.5. Berechnung einer ersten Basislösung	24
Kapitel 2. Duale lineare Optimierung	27
2.1. Duale Aufgabe	27
2.2. Dualitätssätze	27
2.3. Interpretation der dualen Aufgabe	28
2.4. Berechnung einer optimalen Lösung der dualen Aufgabe	29
2.5. Dualer Simplexalgorithmus	30
Kapitel 3. Sensitivitätsanalyse	31
3.1. Allgemeine Veränderung der Zielfunktionskoeffizienten	31
3.2. Ein Parameter in der Zielfunktion	32
3.3. Veränderung der rechten Seite	34
3.4. Ein Parameter in der rechten Seite	35
3.5. Aufnahme einer neuen Variablen	36
3.6. Aufnahme einer neuen Nebenbedingung	37
Kapitel 4. Optimierung mit mehreren Zielen	39
4.1. Modell, Aufgabenstellung	39
4.2. Lösungszugang	40
Kapitel 5. Transportoptimierung	41
5.1. Einführung, Modell	41
5.2. Eigenschaften	42
5.3. Konstruktion einer Startbasislösung	44
5.4. Das duale Transportproblem	45
5.5. Verbesserungsschritt	45
5.6. Zweistufige Transportprobleme	46
5.6.1. Charakterisierung des Problems	46

5.6.2. Transformation auf das klassische Transportproblem	48
Kapitel 6. Diskrete Optimierung	49
6.1. Modellierung diskreter Optimierungsaufgaben	49
6.2. Exakte Lösungsverfahren	52
6.2.1. Schnittebenenverfahren	52
6.2.2. Branch-and-bound-Algorithmus	52
6.3. Näherungsverfahren	55
6.3.1. Eröffnungsverfahren	55
6.3.2. Verbesserungsverfahren	56
Kapitel 7. Einige Modelle der Logistik	59
7.1. Tourenprobleme	59
7.1.1. Näherungsverfahren	60
7.1.2. Modell für das Tourenplanungsproblem mit Zeitfenstern	62
7.2. Standortprobleme	63
7.2.1. Das kapazitierte Standortproblem	63
7.2.2. Das unkapazitierte Standortproblem	64
7.2.3. Exakte Lösung des kapazitierten Standortproblems mit branch-and-bound Algorithmus	65
Kapitel 8. Grundlagen zur nichtlinearen Optimierung	67
8.1. Optimierung ohne Nebenbedingungen	67
8.2. Optimierung mit Gleichungsnebenbedingungen	68
8.3. Aufgaben mit Ungleichungsnebenbedingungen	68
Literaturverzeichnis	71

## KAPITEL 0

### Einleitung

#### 0.1. Historische Entwicklung

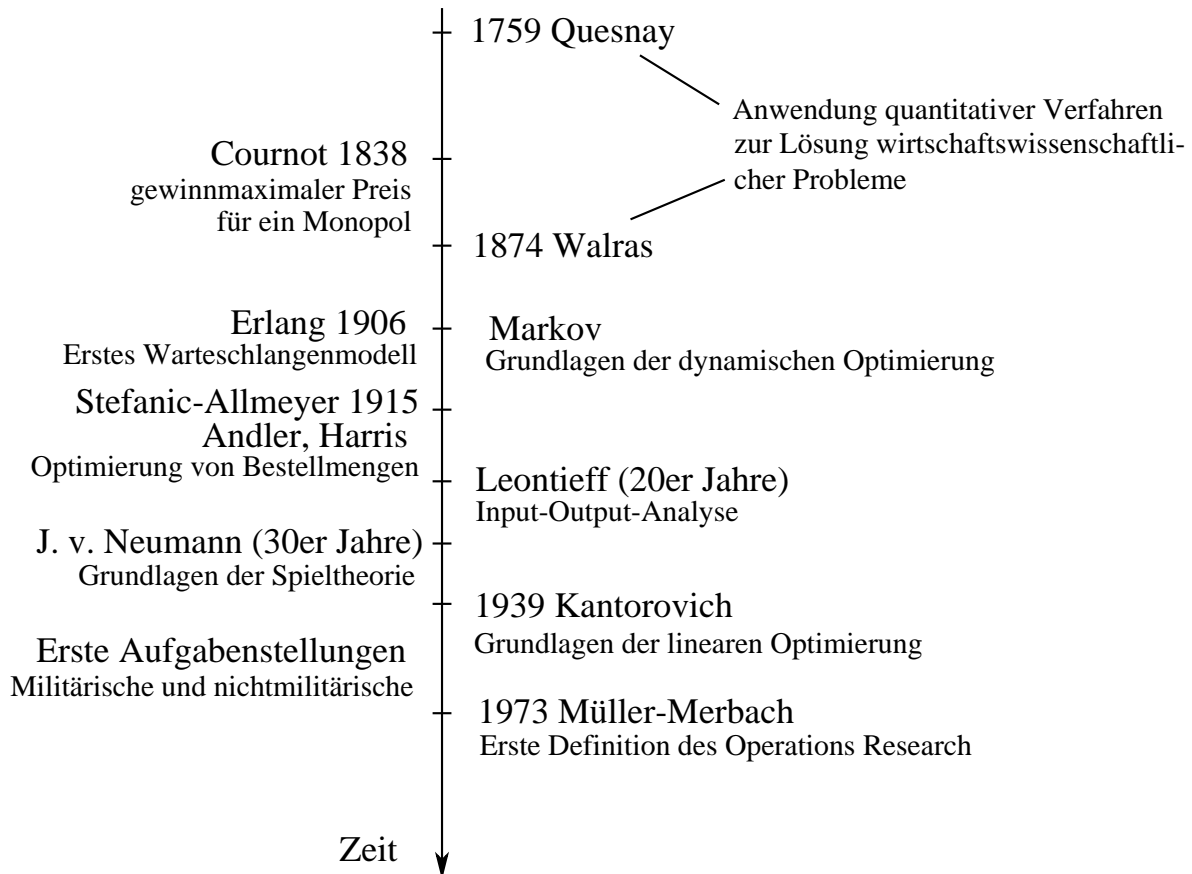


ABBILDUNG 0.1. Historische Entwicklung

#### Erste Aufgabenstellungen des OR

- (1) Militärische Aufgaben im 2. Weltkrieg: Verbesserung der Wirksamkeit militärischer Operationen (Radar, U-Boot-Bekämpfung, Sicherung von Transportschiffen durch Geleitzüge)
- (2) Nichtmilitärische Aufgaben nach dem 2. Weltkrieg:
  - Untersuchung der regionalen Verteilung von Feuerüberwachungsstationen in der British Patrol.
  - Diätenprobleme: Wie ist eine Mahlzeit zusammensetzen, damit sie den Heilungsprozess bestmöglich unterstützt?
  - Verschneiden von Benzin unterschiedlicher Qualität in Ö raffinerien.

## 0.2. Begriff des Operations Research

MÜLLER-MERBACH (1973):

Unter dem Begriff Optimalplanung (Operations Research) wird die Anwendung von mathematischen Methoden zur Vorbereitung optimaler Entscheidungen verstanden.

OPERATIONS RESEARCH SOCIETY OF AMERICA (1976):

Operations Research befasst sich mit wissenschaftlich fundierten Entscheidungen über die beste Gestaltung und Steuerung von Mensch-Maschine-Beziehungen, und zwar zumeist unter der Bedingung, dass die zu verwendenden Mittel knapp sind.

DINKELBACH (1978):

Unternehmensforschung ist die Lehre von den Verfahren zur numerischen Lösung von Entscheidungsmodellen.

JOURNAL OF THE OPERATIONAL RESEARCH SOCIETY (UK):

Operational Research ist die Anwendung wissenschaftlicher Methoden auf komplexe Probleme, die in der Industrie, in der Wirtschaft, in der Verwaltung und in der Verteidigung im Zusammenhang mit der Steuerung und Führung großer Systeme auftreten, in denen Menschen, Maschinen, Material und Geld zusammenwirken. Die charakteristische Vorgehensweise des Operational Research liegt in der Entwicklung eines wissenschaftlichen Modells von dem System, mit dem die Ergebnisse alternativer Entscheidungen, Strategien und Steuerungsmaßnahmen vorhergesagt und verglichen werden können. Diese Modelle umfassen auch Maßzahlen, wie etwa Chance und Risiko einschließlich deren Messungen. Die Modelle dienen dem Zweck, Führungsentscheidungen über Politik und Einzelmaßnahmen wissenschaftlich vorzubereiten.

GAL, HORST, ISERMANN, MÜLLER-MERBACH (1989):

- Operations Research ist eine interdisziplinäre wissenschaftliche Disziplin.
- Operations Research ist eine Modellierungs- und Methodenlehre, die als Sammlung von Methoden (und Strukturierungsverfahren) zwischen der Mathematik, Systemtheorie, Informatik und Entscheidungstheorie steht. Sie kann jedoch zu jedem Sachgebiet zugeordnet werden, sofern sie Sachprobleme dieses Gebietes mit eigenen Methoden löst.
- Die Aufgabe des Operations Research ist es, an der Lösung von Realproblemen mitzuwirken, dabei eigene Methoden und Verfahren zur Strukturierung und zur Lösung der Modelle einzusetzen und bei der Implementierung mitzuwirken. Seine Aufgabe ist es auch, neue Methoden und Verfahren zur Lösung von entsprechenden verallgemeinerten Problemen und die dazugehörige Theorie zu entwickeln.

NEUMANN, MORLOCK:

Operations Research bedeutet die Suche nach einer bestmöglichen (optimalen) Entscheidung unter Berücksichtigung von Nebenbedingungen.

### 0.3. Einsatz der Methoden des OR

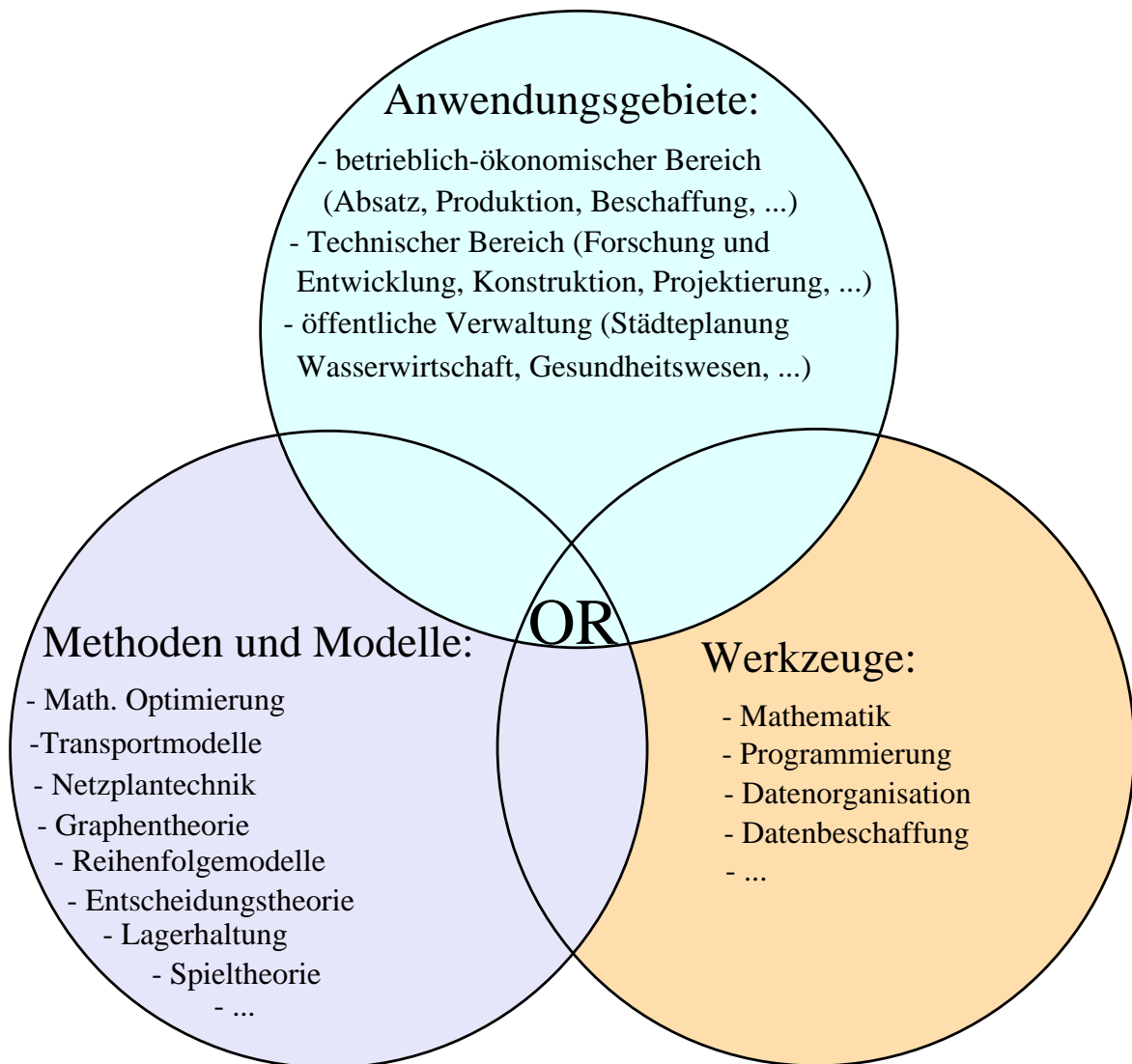
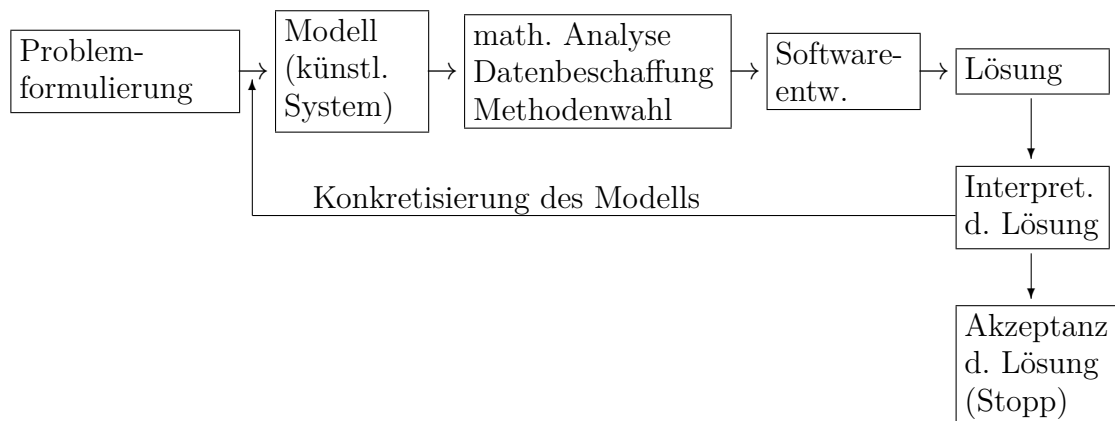


ABBILDUNG 0.2. Operations Research Modell

#### Wesen des Operations Research







## Lineare Optimierung

### 1.1. Einführung, Modell

Aufgabe der Optimierung: Gegeben seien eine Menge  $M$  von Alternativen und ein Zielkriterium. Unter Optimierung versteht man die Suche nach einem gemäß dem Zielkriterium besten Element in der Menge  $M$ .

Unter linearer Optimierung versteht man die Maximierung oder Minimierung einer linearen Funktion, bezeichnet als Zielfunktion, unter der Bedingung, dass die Variablen einem gegebenen System von linearen Ungleichungen oder Gleichungen, bezeichnet als Restriktionen oder Nebenbedingungen, genügen müssen.

#### **Aufgabe der linearen Optimierung (LOA) in Summenschreibweise:**

Maximierung/Minimierung der Funktion (bezeichnet als Zielfunktion)

$$\sum_{j=1}^n c_j x_j \longrightarrow \max / \min$$

unter den Restriktionen (oder Nebenbedingungen)

$$\sum_{j=1}^n a_{ij} x_j \begin{matrix} < \\ = \\ > \end{matrix} b_i \quad \text{für alle } i = 1, 2, \dots, m$$

und unter den Nichtnegativitätsbedingungen

$$x_j \geq 0 \quad \text{für alle } j = 1, 2, \dots, r.$$

Hierbei ist  $r \leq n$  die Anzahl der Variablen mit Nichtnegativitätsbedingung. Unter Verwendung der Matrizenrechnung schreibt man die lineare Optimierungsaufgabe kurz auch als

$$\begin{aligned} c^\top x &= \langle c, x \rangle \rightarrow \min / \max \\ Ax &\begin{cases} \leq \\ = \\ \geq \end{cases} b \\ x_j &\geq 0 \quad \text{für alle } j = 1, 2, \dots, r. \end{aligned}$$

Bezeichnungen:  $A$  Koeffizientenmatrix  
 $b$  Vektor der rechten Seite  
 $c$  Zielfunktionsvektor

Zur Modellierung:

- (1) Wichtigstes ist die korrekte Fixierung der Variablen  $x_j$ ! Als solche kommen nur Größen in Betracht, auf die man direkten Einfluß hat, also z.B. Produktionsmengen in Produktionsplanungsproblemen, zu mischende Mengen in Mischungsproblemen, zu zerschneidende Stangen oder Platten in Zuschnittproblemen usw.

Eine formale Beschreibung dieser Variablen (mit Angabe der Maßeinheit) ist anzugeben.

- (2) Zulässiger Bereich: Er beschreibt die Menge der Alternativen mit Hilfe von Ungleichungen und Gleichungen. In der linearen Optimierung sind das lineare Ungleichungen und Gleichungen. Zu beachten ist dabei das Relationszeichen:
  - (a) „=“: vollständige Ausnutzung gegebener Ressourcen
  - (b) „≤“: Gegebene Ressourcen sind nicht zu überschreiten
  - (c) „≥“: Sicherung gewisser Mindestgrößen, wie z.B. Mindestumsatz, untere Schranke für die Warenproduktion zur Sicherung einer effizienten Produktion usw.
 Die Variablen unterliegen oftmals noch den Nichtnegativitätsbedingungen: Diese dürfen nicht vergessen werden und ergeben sich zumeist aus Vorzeichenbeschränkungen für ökonomische Größen.
- (3) Die Zielfunktion beschreibt das verfolgte Zielkriterium, wie z.B. maximaler Gewinn, minimale Kosten usw. In der linearen Optimierung ist die Zielfunktion eine lineare Funktion.
- (4) Manchmal (z.B. beim Mischungsproblem) sind noch weitere implizit gegebene Nebenbedingungen zu beachten: insgesamt zu mischende Menge.

## 1.2. Normalform

**Normalform linearer Optimierungsaufgaben (NF):**

$$\begin{aligned} \langle c, x \rangle &\rightarrow \min \\ Ax &= b \\ x &\geq 0 \end{aligned} \tag{1.1}$$

Dabei ist  $A$  vom Typ  $(m, n)$ , die Dimensionen der Vektoren  $b, c, x$  sind entsprechend gewählt. Besonderheiten der Normalform:

- (1) Zielfunktion ist immer zu minimieren.
- (2) Es gibt ausschließlich Gleichungsnebenbedingungen.
- (3) Vorhandensein von Nichtnegativitätsbedingungen für alle auftretenden Variablen.
- (4)  $b \geq 0$  (rechte Seite der Beschränkung immer nichtnegativ).

**Definition 1.1** *Betrachtet werde eine lineare Optimierungsaufgabe in Normalform.*

*Ein Punkt  $\bar{x}$  mit  $\bar{x} \geq 0$ ,  $A\bar{x} = b$  heißt zulässige Lösung.*

*Die Menge  $M = \{x \geq 0 : Ax = b\}$  aller zulässigen Lösungen wird als zulässiger Bereich bezeichnet.*

*Eine zulässige Lösung  $x^*$  heißt optimale Lösung der linearen Optimierungsaufgabe, wenn für alle zulässigen Lösungen  $x \in M$  die Bedingung  $\langle c, x \rangle \geq \langle c, x^* \rangle$  erfüllt ist.*

**Überführung von LOA in die Normalform:** Gegeben sei eine lineare Optimierungsaufgabe, die nicht in Normalform gegeben ist. Es gibt also entweder eine Variable ohne Nichtnegativitätsbedingung oder die Zielfunktion ist zu maximieren oder aber eine Nebenbedingung liegt nicht in Gleichungsform vor oder ein Koeffizient der rechten Seite ist negativ. Dann ist die Aufgabe in Normalform zu transformieren, wobei die folgenden Regeln angewendet werden können.

- (1) Fehlende Nichtnegativitätsbedingung für eine Variable. Dann ist diese Variable in der gesamten Aufgabe wie folgt zu ersetzen:

$$x_j = x'_j - x''_j \quad : \quad x'_j, x''_j \geq 0$$

Alternativ kann bei einer fehlenden Nichtnegativitätsbedingung auch eine der folgenden Regeln angewendet werden, wenn in der Ausgangsaufgabe eine Variablenbeschränkung der angegebenen Form vorhanden ist. Eine Anwendung dieser Regeln verringert die Anzahl der neu aufzunehmenden Variablen. Die erste Formel ist dabei eine in der ursprünglichen Aufgabe vorhandene Nebenbedingung und die danach stehenden Ausdrücke geben die Formeln für ihre Substitution an:

$$\begin{aligned} x_j \geq a_j \quad (a_j \neq 0) &\Rightarrow x_j = x'_j + a_j & : \quad x'_j &\geq 0 \\ x_j \leq a_j &\Rightarrow x_j = a_j - x'_j & : \quad x'_j &\geq 0 \end{aligned}$$

(2) Zielfunktion ist zu maximieren

$\Rightarrow$  Multiplikation der Zielfunktion mit  $(-1)$ :

$$\langle c, x \rangle \rightarrow \max \Rightarrow \langle -c, x \rangle \rightarrow \min$$

(3) Ungleichungen als Nebenbedingungen

$\Rightarrow$  Einführung von Schlupfvariablen  $u_i$  :

$$\sum_{j=1}^n a_{ij}x_j \leq b_i \quad \Rightarrow \quad \sum_{j=1}^n a_{ij}x_j + u_i = b_i \quad : \quad u_i \geq 0$$

$$\sum_{j=1}^n a_{ij}x_j \geq b_i \quad \Rightarrow \quad \sum_{j=1}^n a_{ij}x_j - u_i = b_i \quad : \quad u_i \geq 0$$

(4) Bedingung  $b \geq 0$  nicht erfüllt:

$b_i < 0 \Rightarrow$  Multiplikation der  $i$ -ten Nebenbedingung mit  $(-1)$

**Bemerkung 1.2** *Nach der Berechnung der optimalen Lösung müssen die Transformationen unbedingt wieder rückgängig gemacht werden!*

### 1.3. Graphische Lösung

Die graphische Lösung von Problemen der linearen Optimierung ist anwendbar, wenn die Zahl der Variablen  $n \leq 2$  ist (in speziellen Fällen auch bei  $n > 2$ ).

Beispiel:

$$5x_1 + 8x_2 \rightarrow \max$$

$$5x_1 + 2x_2 \leq 24 \tag{1.2a}$$

$$x_1 + 5x_2 \leq 24 \tag{1.2b}$$

$$x_1 + x_2 \leq 6 \tag{1.2c}$$

$$x_1 \geq 0 \tag{1.2d}$$

$$x_2 \geq 0 \tag{1.2e}$$

Lösungsschritte:

(1) Menge der zulässigen Lösungen im Koordinatensystem einzeichnen: Zunächst werden die den zulässigen Bereich begrenzenden Geraden der Reihe nach eingezeichnet.

(a) Einzeichnen einer Geraden. Diese teilt die Ebene in zwei Halbebenen. Der zulässige Bereich liegt in genau einem dieser Halbräume.

- (b) Bestimmung des richtigen Halbraumes durch Einsetzen eines Punktes, der nicht auf der Geraden liegt, in die betrachtete Ungleichung. Erfüllt er diese, so liegt er im richtigen Halbraum, sonst im falschen. Markieren des richtigen Halbraumes.

Der zulässige Bereich ergibt sich als Durchschnitt aller so markierten Halbräume. Wenn der zulässige Bereich leer ist, dann ist die LOA nicht lösbar, stopp. Ansonsten ist der zulässige Bereich zu schraffieren.

- (2) Optimierung: Es wird eine Niveaulinie  $c_1x_1 + c_2x_2 = d$  der Zielfunktion eingezeichnet. Dabei kann eine beliebige Zahl  $d \geq 0$  verwendet werden, da  $d$  den Anstieg der Geraden nicht geändert, sondern lediglich die Lage durch Verschiebung bestimmt („Niveau“). Die Niveaulinie muss durch Parallelverschiebung über dem zulässigen Bereich so weit wie möglich in Optimierungsrichtung verschoben werden. Wenn diese Verschiebung bis in das Unendliche möglich ist, so ist die LOA nicht lösbar, stopp. Ansonsten ergibt sich dabei die Niveaulinie  $c_1x_1 + c_2x_2 = f^*$ . Dabei ist  $f^*$  der optimale Zielfunktionswert. Jeder zulässige Punkt auf dieser Niveaulinie ist optimale Lösung der LOA.

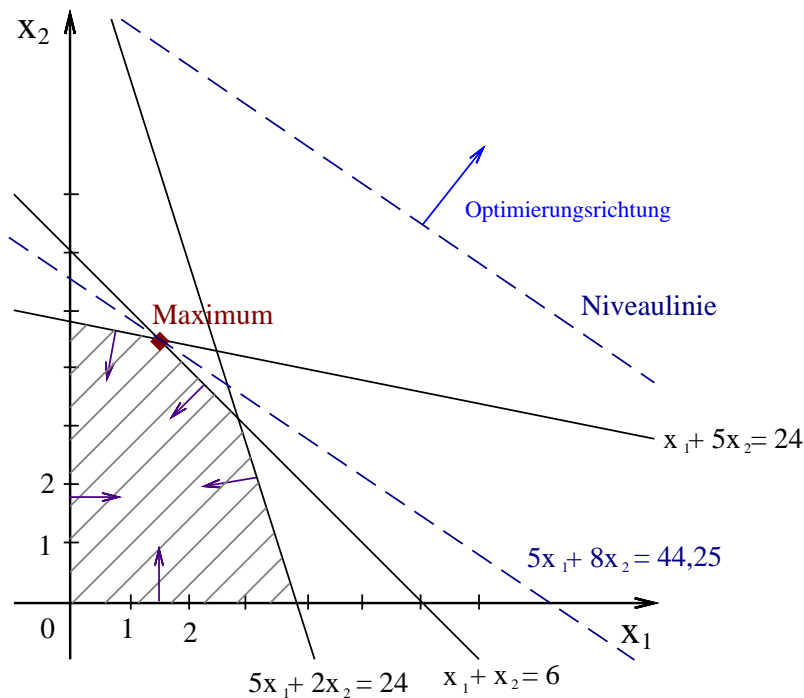


ABBILDUNG 1.1. Graphische Lösung

- (3) Berechnung der optimalen Lösung(en): Hier sind zwei Fälle möglich:
- Die optimale Lösung ist eindeutig. Dann gibt es zwei den zulässigen Bereich begrenzende Geraden, deren Schnittpunkt die optimale Lösung ist. Zur Berechnung der optimalen Lösung ist das durch diese beiden Geraden bestimmte lineare Gleichungssystem zu lösen.
  - Es gibt unendlich viele optimale Lösungen. Dann gibt es auch (mindestens) einen Schnittpunkt von zwei Geraden, der einer optimalen Lösung entspricht. Dieser Schnittpunkt wird berechnet.

Wird die Menge optimaler Lösungen durch zwei solche Schnittpunkte begrenzt, so ist sie gleich der Menge aller konvexen Linearkombinationen dieser zwei Schnittpunkte.

Wenn das nicht der Fall ist, so kann die Menge aller optimalen Lösungen mit Hilfe der Punkt-Richtungsform für die Gerade, auf der die optimalen Lösungen liegen, angegeben werden. Der dabei verwendete Parameter muss im Vorzeichen beschränkt werden.

Im Beispiel sind das die Gleichungen der Nebenbedingungen (1.2b) und (1.2c), die sich im Punkt Maximum schneiden.

$$\begin{aligned}x_1 + 5x_2 &= 24 \\x_1 + x_2 &= 6\end{aligned}$$

Hier ist die optimale Lösung eindeutig:

$$\begin{aligned}\text{optimale Lösung der LOA: } x^* &= \begin{pmatrix} 1,5 \\ 4,5 \end{pmatrix} \\ \text{optimaler Zielfunktionswert: } f^* &= 5 \cdot 1,5 + 8 \cdot 4,5 = 43,5\end{aligned}$$

Folgende Fälle sind möglich:

- (a) Vorhandensein einer eindeutigen optimalen Lösung.
  - (b) Unendlich viele optimale Lösungen.
  - (c) LOA ist nicht lösbar: zulässiger Bereich ist leer.
  - (d) LOA ist nicht lösbar: Unbeschränktheit der Zielfunktion über dem zulässigen Bereich in Optimierungsrichtung.
- (4) Interpretation der Lösung.

**Beispiel 1.3** Professor Milchmann und seine Familie betreiben ein Geschäft, in dem sie Milchprodukte verkaufen, die sie aus der Milch der drei Kühe Daisy, Ermentrude und Florence herstellen. Die drei Kühe geben zusammen pro Woche 22 Fass Milch, aus denen Professor Milchmann und seine Familie Speiseeis und Butter herstellen, die sie in ihrem Geschäft verkaufen. Zur Herstellung von einem Kilogramm Butter werden 2 Fass Milch und zur Produktion von einem Fass Speiseeis werden 3 Fass Milch benötigt. Natürlich können auch beliebige kleinere Teilmengen an Speiseeis und Butter hergestellt werden.

Familie Milchmann besitzt einen sehr großen Kühlschrank, so dass eine beliebig große Menge an Butter aufbewahrt werden kann, jedoch ist der Gefrierschrank wesentlich kleiner und fasst höchstens 6 Fass Speiseeis.

Prof. Milchmanns Familie kann pro Woche höchstens 6 Stunden für die Produktion von Speiseeis und Butter aufwenden. In einer Stunde sind sie in der Lage, entweder 4 Fass Speiseeis oder ein Kilogramm Butter herzustellen.

Prof. Milchmanns Produkte werden so nachgefragt, dass alle hergestellten Produkte zu beliebigen Preisen verkauft werden können. Er bestimmt die Preise so, dass er einen Profit von 5 € pro Fass Speiseeis und 4 € pro Kilogramm Butter macht. Wie viel Speiseeis und wie viel Butter soll er herstellen, so dass sein Gewinn maximal ist?

Um das Problem zu modellieren werden zunächst Variable fixiert:  $x_1$  ist die Menge an produziertem Speiseeis [in Fass] und  $x_2$  die Menge an Butter [in Kilogramm]. Dann ergibt sich die folgende lineare Optimierungsaufgabe:

$$\begin{array}{rcll}
\max & z = 5x_1 & + & 4x_2 \\
\text{u.d.B.} & x_1 & & \leq 6 \\
& 0,25x_1 & + & x_2 \leq 6 \\
& 3x_1 & + & 2x_2 \leq 22 \\
& & & x_1, x_2 \geq 0
\end{array}$$

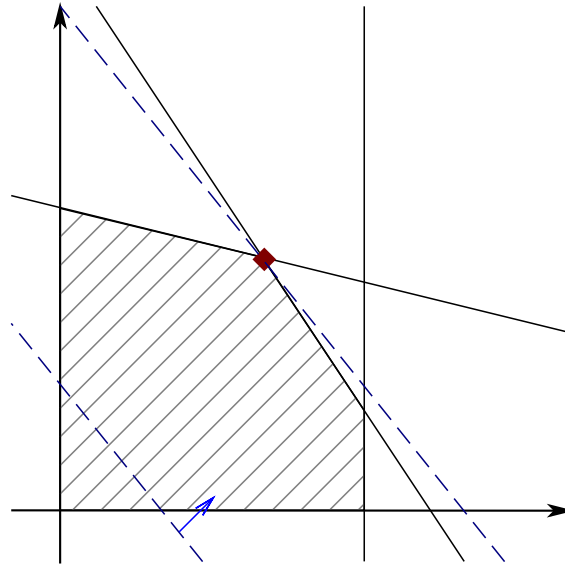


ABBILDUNG 1.2. Graphische Lösung des Beispiels 1.3. Dargestellt sind schraffiert der zulässige Bereich, zwei verschiedene Niveaumengen der Zielfunktion zu den Niveaus  $z = 10$  und  $z = 40$  sowie die optimale Lösung im Punkt  $(4, 5)^T$ .

In der Abbildung 1.2 ist die graphische Lösung der Aufgabe im Beispiel 1.3 abzulesen.

Wenn Prof. Milchmann mit dem erhaltenen Gewinn nicht zufrieden ist, wird er vielleicht den Preis für Speiseeis erhöhen, so dass sich jetzt ein Gewinn von  $5,5 \text{ €}$  pro Fass Speiseeis ergibt. Dann ändert sich die Zielfunktion des Problems von  $z = 5x_1 + 4x_2$  zu  $z = 5,5x_1 + 4x_2$ . In der graphischen Lösung dreht sich die Niveaulinie der Zielfunktion, wie in der Abbildung 1.3 ersichtlich.

Kehren wir nun zum ursprünglichen Gewinn von  $5 \text{ €}$  pro Fass Speiseeis zurück und untersuchen eine etwas andere Fragestellung. Der Nachbar von Prof. Milchmann, Herr Kuhhirt, schlägt Prof. Milchmann vor, ihm zu einem Preis von  $1 \text{ €}$  pro Fass Milch zu verkaufen. Soll Prof. Milchmann auf diesen Handel eingehen und Milch von seinem Nachbarn kaufen und wenn ja, wie viel?

Durch den Zukauf von Milch ändert sich Prof. Milchmanns dritte Nebenbedingung zu  $3x_1 + 2x_2 \leq 22 + c$ , falls er  $c$  Fass Milch von Herrn Kuhhirt kauft. Diesem Zukauf von Milch entspricht eine Parallelverschiebung der Geraden  $3x_1 + 2x_2 \leq 22$  nach oben. Gleichzeitig ändert sich natürlich auch sein Gewinn, da er nun  $c$  Euro von seinem Gewinn abziehen muss. Für fixiertes  $c$  bleibt der Anstieg der Niveaumengen der Zielfunktion gleich, die optimale Lösung liegt also für kleine  $c$  auch weiterhin auf dem Schnittpunkt der Geraden

$$\begin{array}{rcl}
0,25x_1 & + & x_2 = 6 \\
3x_1 & + & 2x_2 = 22 + c.
\end{array}$$

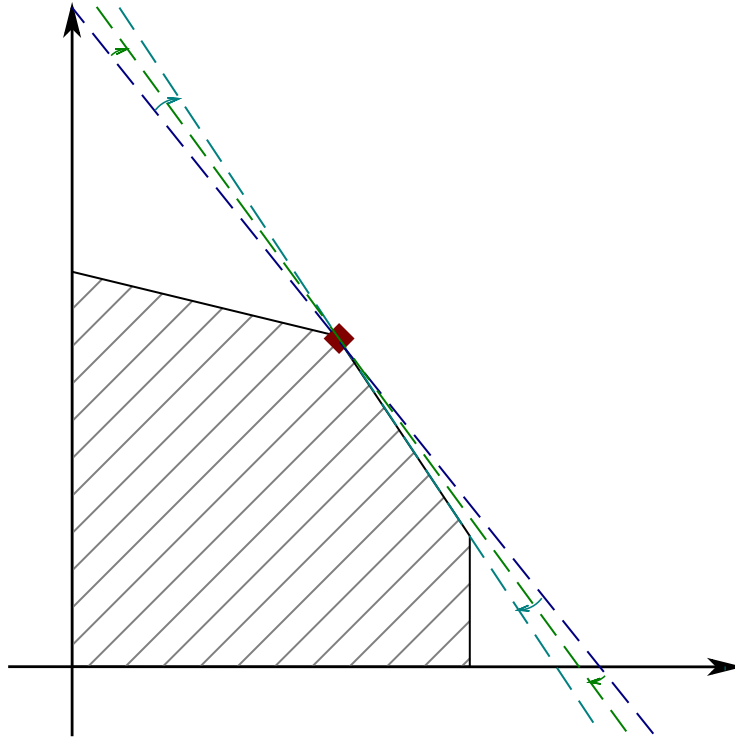


ABBILDUNG 1.3. Durch die Veränderung des ersten Zielfunktionskoeffizienten von  $c_1 = 5$  auf  $c_1 = 5,5$  drehen sich die Niveaumengen der Zielfunktion (die grüne Linie). Die optimale Lösung der Aufgabe bleibt unverändert. Jedoch führt ein erhöhter Speiseeispreis auf 6 € pro Fass zu einer Veränderung der optimalen Lösung. Das Optimum liegt in diesem Fall auf der Strecke zwischen den Punkten:  $(4, 5)^\top$  und  $(6, 2)^\top$ .

Die Lösung ist  $(x_1; x_2)^\top = (4 + 0,4c; 5 - 0,1c)^\top$  mit einem Gewinn von

$$z = 5x_1 + 4x_2 - c = 5(4 + 0,4c) + 4(5 - 0,1c) - c = 40 + 0,6c.$$

Damit ist es sicher für Prof. Milchmann von Vorteil, etwas Milch von seinem Nachbarn zu kaufen, da sein Gewinn um 0,6 € pro gekauftem Fass Milch steigt.

Aus der Abbildung 1.4 ist ersichtlich, dass, wenn  $c$  zu groß wird, die optimale Lösung des Problems von Prof. Milchmann auf dem Schnittpunkt der Geraden

$$\begin{aligned} x_1 &= 6 \\ 0,25x_1 + x_2 &= 6, \end{aligned}$$

also im Punkt  $(x_1; x_2)^\top = (6; 4,5)^\top$  liegt. Dann ist auch der Gewinn von Prof. Milchmann unabhängig von einem weiteren Zukauf von Milch. Prof. Milchmann wird also nur so viel Milch kaufen, bis auch die Gerade  $3x_1 + 2x_2 = 22 + c$  durch den Punkt  $(x_1; x_2)^\top = (6; 4,5)^\top$  verläuft. Das ist für  $c = 5$  der Fall. Prof. Milchmann kauft also 5 Fass Milch und macht anschließend bei einer optimalen Produktion von  $(x_1; x_2)^\top = (6; 4,5)^\top$  einen Gewinn von 43 €.

Betrachten wir jetzt wieder die Ausgangsaufgabe und untersuchen eine weitere Datenveränderung. Prof. Milchman hat zum Geburtstag eine effiziente Eismaschine bekommen, die ein Fass von Speiseeis aus 2,4 Fass Milch produziert. Der Sachverhalt wird in Abbildung 1.5 dargestellt.

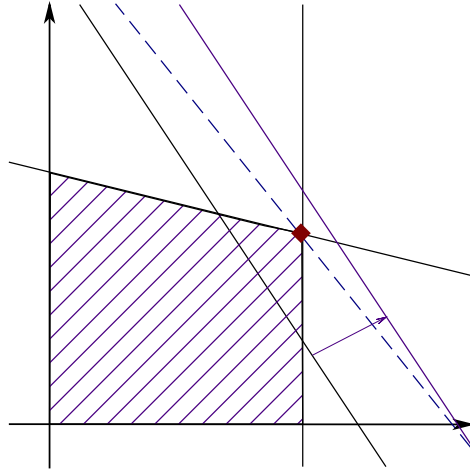


ABBILDUNG 1.4. Durch die Veränderung des Vektors der rechten Seite von  $b_3 = 22$  auf  $b_3 = 22 + c$  verschiebt sich die dritte Nebenbedingung nach rechts und somit verändert sich die optimale Lösung

Die neue optimale Lösung befindet sich jetzt auf dem Schnittpunkt der beiden Geraden

$$\begin{aligned} x_1 &= 6 \\ 2,4x_1 + 2x_2 &= 22. \end{aligned}$$

Die Lösung des Gleichungssystems ist  $(x_1; x_2)^\top = (6; 3,8)^\top$ . Kein Wunder, dass sich Prof. Milchman sehr über das Geschenk gefreut hat, da sein Gewinn von 40€ auf 45,2 € steigt.

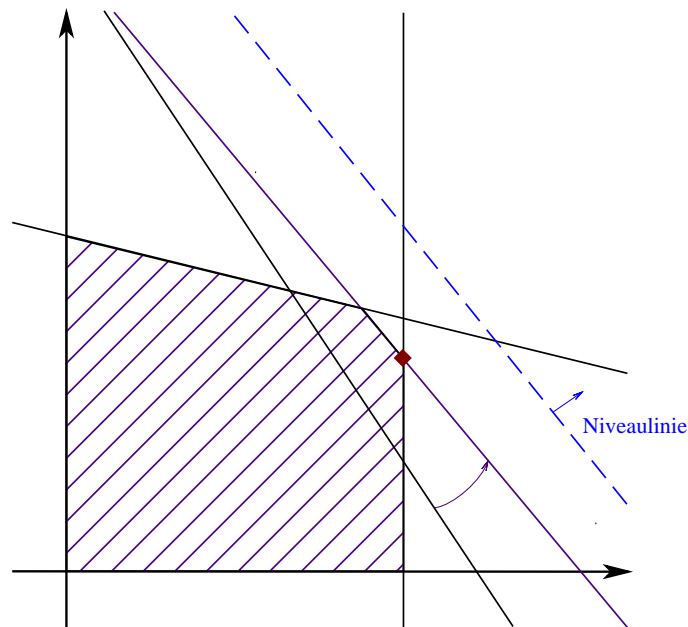


ABBILDUNG 1.5. Durch die Veränderung der Koeffizientenmatrix von  $a_{31} = 3$  auf  $a_{31} = 2,4$  dreht sich die dritte Nebenbedingung und verändert den zulässigen Bereich, sowie die optimale Lösung



## 1.4. Simplexalgorithmus für LOA

### 1.4.1. Wiederholung der grundlegenden Begriffe der linearen Algebra.

**Definition 1.4** Gegeben seien  $r$  Vektoren  $\{x^j\}_{j=1}^r \subset \mathbb{R}^n$ . Dann heißt ein Vektor  $b$  Linearkombination der Vektoren  $\{x^j\}_{j=1}^r$ , wenn reelle Zahlen  $a_1, \dots, a_r$  existieren mit  $b = \sum_{j=1}^r a_j x^j$ .

**Definition 1.5** Gegeben seien  $r$  Vektoren  $\{x^j\}_{j=1}^r \subset \mathbb{R}^n$ . Wenn  $a_1 = a_2 = \dots = a_r = 0$  einzige Lösung des homogenen LGS

$$\sum_{j=1}^r a_j x^j = 0$$

ist, dann heißen die Vektoren  $\{x^j\}_{j=1}^r$  linear unabhängig. Im entgegengesetzten Fall heißen sie linear abhängig.

**Definition 1.6** Eine Menge von linear unabhängigen Vektoren  $\{x^j\}_{j=1}^n \subset \mathbb{R}^n$  heißt Basis des  $\mathbb{R}^n$ . Die Elemente einer Basis heißen Basisvektoren. Die mit den Basisvektoren als Spaltenvektoren gebildete Matrix ist die Basismatrix.

**Bemerkung 1.7** Eine Basismatrix ist quadratisch.

**Definition 1.8** Die maximale Anzahl linear unabhängiger Spalten einer Matrix  $A$  heißt Rang dieser Matrix. Bezeichnung:  $r(A)$ .

**Definition 1.9** Eine quadratische Matrix  $A$  vom Typ  $(n, n)$  heißt regulär, wenn  $r(A) = n$  ist. Sonst heißt sie singulär.

**Definition 1.10** Für eine quadratische reguläre Matrix  $A$  ist die (reguläre quadratische) Matrix  $X$ , für die  $A \cdot X = X \cdot A = E$  gilt, die inverse Matrix zur Matrix  $A$ . Dabei ist  $E$  die Einheitsmatrix.

**Definition 1.11** Eine Menge  $M \subseteq \mathbb{R}^n$  ist konvex, wenn  $\forall x, y \in M, \forall \lambda \in (0, 1)$  auch  $z = \lambda x + (1 - \lambda)y \in M$  ist.

In jedem Schritt des Simplexalgorithmus wird der Gauß-Jordan-Algorithmus benutzt. Die Grundidee des Verfahrens ist die Elimination der Variablen aus den Gleichungen, so dass am Ende in jeder Zeile nur eine Variable bleibt (falls die Matrix quadratisch ist). Mit anderen Worten: Jede Gleichung wird nach einer anderen Variable aufgelöst. Der Algorithmus besteht aus folgenden Schritten:

Gegeben sei ein lineares Gleichungssystem  $Ax = b$  mit  $A \in \mathbb{R}^{m \times n}$  d. h.  $A$  ist eine Matrix vom Typ  $(m, n)$

$$\begin{array}{cccccc} a_{11}x_1 & + & a_{12}x_2 & + & \dots & + & a_{1n}x_n & = & b_1 \\ a_{21}x_1 & + & a_{22}x_2 & + & \dots & + & a_{2n}x_n & = & b_2 \\ \vdots & + & \vdots & + & \ddots & + & \vdots & = & \vdots \\ a_{m1}x_1 & + & a_{m2}x_2 & + & \dots & + & a_{mn}x_n & = & b_m \end{array}$$

Schritt 1: Die erste Gleichung wird durch den Koeffizienten  $a_{11}$  dividiert. Falls  $a_{11} = 0$ , muss eine Zeilenvertauschung durchgeführt werden. Danach wird ein Vielfaches (Multiplikation mit der Zahl  $(-a_{21})$ ) von der ersten Zeile zu der zweiten Zeile addiert, so dass die Variable  $x_1$  aus der zweiten Gleichung eliminiert wird. Um die Variable  $x_1$  aus den weiteren Zeilen zu „entfernen“, wird die erste Gleichung

mit der Zahl  $(-a_{i1})$  multipliziert und zu der  $i$ -ten Zeile addiert. Dies soll für  $i = 3, \dots, m$  wiederholt werden.

Schritt 2: Im zweiten Schritt wird die nach dem ersten Schritt entstandene Matrix  $A^{(1)}$  betrachtet. Die zweite Gleichung wird durch den Koeffizienten  $a_{22}$  dividiert (falls dieser gleich null ist, folgt ein Zeilentausch mit einer der unteren Zeilen, deren Koeffizient  $a_{2j} \neq 0$  ist). Danach wird die Variable  $x_2$  aus der zweiten Spalte wie im Schritt (1) eliminiert. Darauf folgend werden auch über der zweiten Zeile die „nullen“ erzeugt, indem man geeignete Vielfache der zweiten Zeile zu der Ersten addiert. Als Ergebnis bekommt man ein lineares Gleichungssystem der folgenden Gestalt:

$$\begin{array}{cccccccc} x_1 & + & 0 & + & a_{13}^{(2)} x_3 & + & \dots & + & a_{1n}^{(2)} x_n & = & b_1^{(2)} \\ 0 & + & x_2 & + & a_{23}^{(2)} x_3 & + & \dots & + & a_{2n}^{(2)} x_n & = & b_2^{(2)} \\ 0 & + & 0 & + & a_{33}^{(2)} x_3 & + & \dots & + & a_{3n}^{(2)} x_n & = & b_r^{(2)} \\ \vdots & + & \vdots & + & \vdots & + & \ddots & + & \vdots & = & \vdots \\ 0 & + & 0 & + & a_{m3}^{(2)} x_3 & + & \dots & + & a_{mn}^{(2)} x_n & = & b_m^{(2)} \end{array}$$

Schritt 3: Der zweite Schritt wird für weitere Spalten  $j = 3, \dots, r$  wiederholt, wobei  $r$  den Rang der Matrix  $A$  bezeichnet.

Mit Hilfe des Gauß-Jordan-Algorithmus lässt sich jedes lineare Gleichungssystem in die folgende Form bringen:

$$\begin{array}{cccccccccccc} x_1 & + & 0 & + & \dots & + & 0 & + & a_{1r+1} x_{r+1} & + & \dots & + & a_{1n} x_n & = & b_1 \\ 0 & + & x_2 & + & \dots & + & 0 & + & a_{2r+1} x_{r+1} & + & \dots & + & a_{2n} x_n & = & b_2 \\ \vdots & + & \vdots & + & \ddots & + & \vdots & + & \vdots & + & \vdots & + & \vdots & = & \vdots \\ 0 & + & 0 & + & \dots & + & x_r & + & a_{rr+1} x_{r+1} & + & \dots & + & a_{rn} x_n & = & b_r \\ 0 & + & 0 & + & \dots & + & 0 & + & 0 & + & \dots & + & 0 & = & b_{r+1} \\ \vdots & + & \vdots & + & \vdots & + & \vdots & + & \ddots & + & \vdots & + & \vdots & = & \vdots \\ 0 & + & 0 & + & \dots & + & 0 & + & 0 & + & \dots & + & 0 & = & b_m \end{array}$$

**Bemerkung 1.12** Falls eine Zeile der Art  $(0 + 0 + \dots + 0 = b_i)$  mit  $b_i \neq 0$  entsteht, so ist das lineare Gleichungssystem nicht lösbar, weil das ursprüngliche lineare Gleichungssystem widersprüchliche Gleichungen enthielt.

Der Gauß-Jordan-Algorithmus ermöglicht auch die Inverse einer regulären Matrix zu berechnen, indem man die Gleichung  $Ax = E$  löst. Nach der Umformung einer regulären Matrix mit dem Gauß-Jordan-Verfahren entsteht eine Einheitsmatrix ( $E$ ). Das gleiche Ergebnis liefert auch nach der Definition 1.10 die Multiplikation der Gleichung  $Ax = b$  mit der Inversen von  $A$ :

$$Ax = b \xrightarrow{G-J-Alg.} Ex = A^{-1}(Ax) = A^{-1}b.$$

Falls  $A$  nicht quadratisch ist, lässt sich eine quadratische reguläre Teilmatrix  $B$  der Matrix  $A$  finden ( $A = (B N)$ ), dann gilt:

$$Ax = b \Rightarrow (B N)x = b \Rightarrow (B N) \begin{pmatrix} x_B \\ x_N \end{pmatrix} = b \Rightarrow x_B = B^{-1}b - B^{-1}N x_N$$

**Beispiel 1.13** Das folgende Gleichungssystem wird mit dem Gauß-Jordan-Algorithmus gelöst:

Schritt 1:

$$\begin{array}{rcl} -6x_1 & - & 4x_3 = 12 \quad / \div (-6) \\ 3x_1 + 2x_2 + 8x_3 & = & 6 \\ & 2x_2 - 3x_3 & = 15 \end{array}$$

$$\begin{array}{rcl} x_1 & + & \frac{2}{3}x_3 = -2 \\ 3x_1 + 2x_2 + 8x_3 & = & 6 \quad + (-3) \cdot \text{Zeile I} \\ & 2x_2 - 3x_3 & = 15 \end{array}$$

Schritt 2:

$$\begin{array}{rcl} x_1 & + & \frac{2}{3}x_3 = -2 \\ 2x_2 + 6x_3 & = & 0 \quad / \div 2 \\ 2x_2 - 3x_3 & = & 18 \end{array}$$

$$\begin{array}{rcl} x_1 & + & \frac{2}{3}x_3 = -2 \\ x_2 + 3x_3 & = & 0 \\ 2x_2 - 3x_3 & = & 18 \quad + (-2) \cdot \text{Zeile II} \end{array}$$

Schritt 3:

$$\begin{array}{rcl} x_1 & + & \frac{2}{3}x_3 = -2 \\ x_2 + 3x_3 & = & 0 \\ & - & 9x_3 = 18 \quad / \div (-9) \end{array}$$

$$\begin{array}{rcl} x_1 & + & \frac{2}{3}x_3 = -2 \quad + (-\frac{2}{3}) \cdot \text{Zeile III} \\ x_2 + 3x_3 & = & 0 \quad + (-3) \cdot \text{Zeile II} \\ & x_3 & = -2 \end{array}$$

$$\begin{array}{rcl} x_1 & & = -\frac{2}{3} \\ x_2 & & = 6 \\ & x_3 & = -2 \end{array}$$

#### 1.4.2. Grundlagen des Simplexalgorithmus.

Der Simplexalgorithmus löst lineare Optimierungsaufgaben in Normalform.

Wir betrachten den zulässigen Bereich der Aufgabe in Normalform:

$$M = \{x \geq 0 : Ax = b\}.$$

Sei  $B$  eine Basismatrix (reguläre, quadratische Teilmatrix) in der Matrix  $A$  und  $A = (B \ N)$  eine Aufteilung der Matrix  $A$  in den Basis- und den Nichtbasisteil. Mit dem Gauß-Jordan-Algorithmus wird das LGS  $Ax = b$  wie folgt transformiert:

$$(B \ N)x = b \Rightarrow (B \ N) \begin{pmatrix} x_B \\ x_N \end{pmatrix} = b \Rightarrow Bx_B + Nx_N = b$$

$$\xrightarrow{G\text{-}J\text{-}Alg.} x_B + B^{-1}Nx_N = B^{-1}b.$$

Die allgemeine Lösung dieses LGS hat die Gestalt

$$x = \begin{pmatrix} B^{-1}b - B^{-1}Nx_N \\ x_N \end{pmatrix}, \quad x_N \text{ beliebig.} \quad (1.3)$$

**Definition 1.14** Eine spezielle Lösung des LGS  $Ax = b$  der Gestalt

$$x = (x_B, x_N)^\top = (B^{-1}b, 0)^\top$$

heißt Basislösung. Wenn in einer Basislösung  $B^{-1}b \geq 0$  ist, so heißt sie zulässige Basislösung. Eine Basislösung  $\tilde{x}$  heißt zu einer Basislösung  $x$  benachbart, wenn  $\tilde{x}$  aus  $x$  durch Anwendung einer Iteration des Gauß-Algorithmus entsteht.

**Bemerkung 1.15** (1) Im Folgenden sei ohne Einschränkung der Allgemeinheit stets  $r(A) = m \leq n$ .

- (2) In jeder zulässigen Basislösung der LOA gibt es höchstens  $m$  positive Komponenten.
- (3) Zulässige Basislösungen sind Eckpunkte des zulässigen Bereiches.
- (4) Die Zahl der verschiedenen zulässigen Basislösungen ist endlich. Der Simplexalgorithmus geht stets von einer zulässigen Basislösung zu einer benachbarten mit nicht schlechterem Zielfunktionswert über. Damit ist er endlich, falls keine zulässige Basislösung mehrfach betrachtet wird.

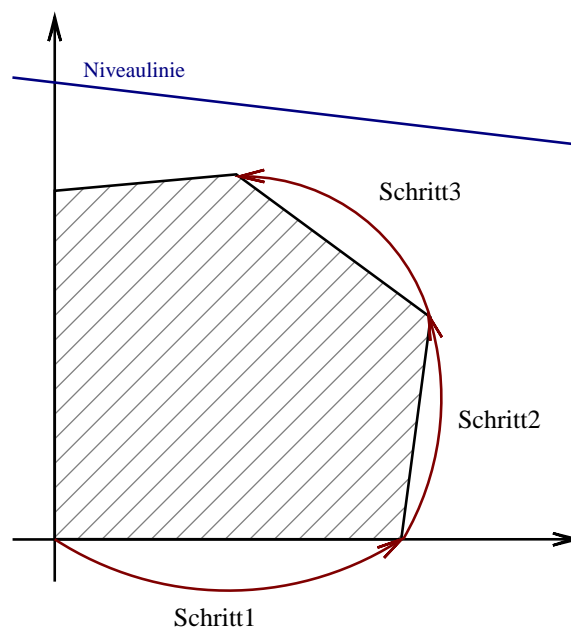


ABBILDUNG 1.6. Die Idee des Simplexalgorithmus für den zweidimensionalen Fall (siehe Bemerkung 1.15)

**Satz 1.16** Wir betrachten die LOA in Normalform (NF). Dann gilt:

- (1) Der zulässige Bereich ist konvex und besitzt mindestens einen Eckpunkt, falls er nicht leer ist. Eckpunkte des zulässigen Bereiches sind zulässige Basislösungen.
- (2) Sei  $c \neq 0$ . Die Optimallösungen einer LOA werden auf dem Rand angenommen, falls die Aufgabe lösbar ist. Unter den optimalen Lösungen gibt es stets auch mindestens eine zulässige Basislösung. Damit reicht es aus, zulässige Basislösungen zu betrachten.
- (3) Eine optimale Lösung existiert, falls der zulässige Bereich nicht leer und die Zielfunktion auf ihm nach unten beschränkt ist. Damit gibt es nur zwei Möglichkeiten für eine Unlösbarkeit der LOA:
  - (a) Der zulässige Bereich ist leer.

- (b) Die Zielfunktion ist auf dem zulässigen Bereich nach unten unbeschränkt.  
 (4) Sind mehrere zulässige Basislösungen (oder auch mehrere zulässige Punkte) optimal, so ist auch jede konvexe Linearkombination dieser Punkte optimal.

### 1.4.3. Idee des Simplexalgorithmus.

Wir betrachten die LOA in Normalform

$$\begin{aligned} \langle c, x \rangle &\rightarrow \min \\ Ax &= b \\ x &\geq 0. \end{aligned} \tag{1.4}$$

Diese wird unter Zuhilfenahme der allgemeinen Lösung des linearen Gleichungssystems  $Ax = b$  in (1.3) umgeformt, d.h.  $x_B$  wird eliminiert durch  $x_B = B^{-1}b - B^{-1}Nx_N$ . Wir betrachten zunächst die Transformation der Zielfunktion:

$$\begin{aligned} c^\top x &= c_B^\top x_B + c_N^\top x_N \\ &= c_B^\top (B^{-1}b - B^{-1}Nx_N) + c_N^\top x_N \\ &= c_B^\top B^{-1}b - c_B^\top B^{-1}Nx_N + c_N^\top x_N \\ &= c_B^\top B^{-1}b - (c_B^\top B^{-1}N - c_N^\top) x_N. \end{aligned}$$

Damit ergibt sich die folgende zu (1.4) äquivalente LOA:

$$\begin{aligned} c_B^\top B^{-1}b - (c_B^\top B^{-1}N - c_N^\top) x_N &\rightarrow \min_{x_N} \\ B^{-1}b - B^{-1}Nx_N &\geq 0 \\ x_N &\geq 0. \end{aligned} \tag{1.5}$$

Aus dieser Darstellung ziehen wir die folgenden Schlüsse:

**Satz 1.17** Sei  $\bar{x} = \begin{pmatrix} \bar{x}_B \\ \bar{x}_N \end{pmatrix}$  eine zulässige Basislösung. Wenn

$$\Delta_j = ((c_B^\top B^{-1}N)^\top - c_N)_j \leq 0 \tag{1.6}$$

ist für alle  $j$ , so ist die vorliegende Basislösung  $\bar{x}$  optimal.

Die Zahlen  $\Delta_j = ((c_B^\top B^{-1}N)^\top - c_N)_j$  heißen Optimalitätsindikatoren (der Nichtbasisvariablen).

**Bemerkung 1.18** Die Optimalitätsindikatoren lassen sich auch für die Basisvariablen analog berechnen, dazu ist die Definition der Optimalitätsindikatoren in Formel (1.6) durch

$$\Delta_j = ((c_B^\top B^{-1}A)^\top - c)_j$$

zu ersetzen. Für die Optimalitätsindikatoren der Basisvariablen ergibt sich dann  $\Delta_j = 0$ .

Sei  $\Delta_{j_0} > 0$ . Dann wird die vorliegende Basislösung nicht als optimal erkannt und gemäß (1.5) wird versucht, die Variable  $x_{j_0}$  zu vergrößern. Wenn alle anderen Nichtbasisvariablen den Wert null behalten, ergibt sich aus den Nebenbedingungen der Aufgabe (1.5) folgendes Ungleichungssystem in der einzigen Variablen  $x_{j_0}$ :

$$(B^{-1}b)_i - (B^{-1}N)_{ij_0} x_{j_0} \geq 0 \quad \forall i.$$

Eine Auflösung dieses Ungleichungssystems ergibt:

$$x_{j_0} \begin{cases} \geq \frac{(B^{-1}b)_i}{(B^{-1}N)_{ij_0}} & \text{für } (B^{-1}N)_{ij_0} < 0 \\ \leq \frac{(B^{-1}b)_i}{(B^{-1}N)_{ij_0}} & \text{für } (B^{-1}N)_{ij_0} > 0 \end{cases}$$

und folglich den maximalen Wert

$$x_{j_0} \leq \min \left\{ \frac{(B^{-1}b)_i}{(B^{-1}N)_{ij_0}} : (B^{-1}N)_{ij_0} > 0 \right\}.$$

Im weiteren sei

$$\theta_i = \frac{(B^{-1}b)_i}{(B^{-1}N)_{ij_0}} \text{ für } (B^{-1}N)_{ij_0} > 0$$

und

$$\theta^* = \min \left\{ \theta_i : (B^{-1}N)_{ij_0} > 0 \right\}.$$

**Satz 1.19** Sei  $\bar{x} = \begin{pmatrix} \bar{x}_B \\ \bar{x}_N \end{pmatrix}$  eine zulässige Basislösung und sei  $\Delta_{j_0} > 0$ . Wenn  $(B^{-1}N)_{ij_0} \leq 0$  ist für alle  $i$ , so ist die lineare Optimierungsaufgabe nicht lösbar, die Zielfunktion ist über dem zulässigen Bereich nach unten unbeschränkt.

**Satz 1.20** Sei  $\bar{x} = \begin{pmatrix} \bar{x}_B \\ \bar{x}_N \end{pmatrix}$  eine zulässige Basislösung und sei  $\Delta_{j_0} > 0$ . Desweiteren existiere mindestens ein  $i$ , so dass  $(B^{-1}N)_{ij_0} > 0$  ist. Sei

$$\theta^* = \theta_{i_0} = \min \left\{ \frac{(B^{-1}b)_i}{(B^{-1}N)_{ij_0}} : (B^{-1}N)_{ij_0} > 0 \right\}.$$

Dann kann durch Aufnahme der Variablen  $x_{j_0}$  in die Basis anstelle der zur  $i_0$ -ten Zeile gehörenden Basisvariablen eine benachbarte zulässige Basislösung berechnet werden, die einen nicht größeren Zielfunktionswert als  $\bar{x}$  besitzt.

**Bemerkung 1.21** Wenn ein Optimalitätsindikator einer Nichtbasisvariable zur optimalen Lösung null ist, so kann es weitere optimale Lösungen geben. Um diese zu errechnen, ist diese Nichtbasisvariable in die Basis aufzunehmen. Geht das nicht, so besitzt die Optimierungsaufgabe eine unbeschränkte Kante optimaler Lösungen. Ist der Basistausch möglich und ergibt die Umrechnung eine andere optimale Lösung, so sind auch alle konvexen Linearkombinationen der berechneten optimalen Lösungen optimal. Im zweidimensionalen Fall befindet sich die Lösung auf einer Strecke zwischen den bestimmten Lösungen.

#### 1.4.4. Der Simplexalgorithmus.

Rechnung unter Verwendung der folgenden Tabelle (Simplextabelle), in der zur Vereinfachung der Schreibweise  $x_B = (x_1, \dots, x_m)^\top$  angenommen wird:

Nr.	$x_B$	$c_B$	$c_1$	$\dots$	$c_m$	$c_{m+1}$	$\dots$	$c_n$	$\bar{b}$	$\theta$
			$x_1$	$\dots$	$x_m$	$x_{m+1}$	$\dots$	$x_n$		
1	$x_1$	$c_1$	1	$\dots$	0	$\bar{a}_{1,m+1}$	$\dots$	$\bar{a}_{1n}$	$\bar{b}_1$	$\theta_1$
2	$x_2$	$c_2$	0	$\dots$	0	$\bar{a}_{2,m+1}$	$\dots$	$\bar{a}_{2n}$	$\bar{b}_2$	$\theta_1$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\vdots$	$\vdots$
m	$x_m$	$c_m$	0	$\dots$	1	$\bar{a}_{m,m+1}$	$\dots$	$\bar{a}_{mn}$	$\bar{b}_m$	$\theta_m$
			$\Delta_1$	$\dots$	$\Delta_m$	$\Delta_{m+1}$	$\dots$	$\Delta_n$	$\langle c_B, x_B \rangle$	

In dieser Tabelle steht  $\bar{a}_{ij}$  für die Elemente der Matrix  $B^{-1}N$  und  $\bar{b}_i$  als Abkürzung für die Komponenten in  $B^{-1}b$ .

**Bemerkung 1.22**

- (1) Basisvariable können auch andere Variable sein.
- (2) Der Wert der Basisvariablen in einer Basislösung steht in der Zeile der Basisvariablen in der Spalte  $\bar{b}$ .
- (3) Nicht unter den Basisvariablen vorkommende Variable sind Nichtbasisvariable. Diese haben in der Lösung den Wert null.
- (4) Der aktuelle Zielfunktionswert ist  $\langle c_B, x_B \rangle = c_B^\top B^{-1}b$ .

Wichtige, aus dem letzten Abschnitt bekannte Rechenformeln:

$$\Delta_j = ((c_B^\top B^{-1}A)^\top - c)_j,$$

$$\theta_i = \frac{(B^{-1}b)_i}{(B^{-1}N)_{ij_0}}.$$

Gegeben sei eine lineare Optimierungsaufgabe in der Normalform (1.4) und es sei einfach eine Basismatrix  $B$  bestimmbar (z. B. mit dem Gauß-Jordan-Algorithmus), die zu einer zulässigen Basislösung führt:

$$B^{-1}b \geq 0.$$

Simplexalgorithmus

Schritt 1: Aufstellen der Simplextabelle und Berechnung der Optimalitätsindikatoren für alle Nichtbasisvariablen ( $\Delta_j$ ).

Schritt 2: Wenn alle  $\Delta_j \leq 0$  sind, dann ist die vorliegende Basislösung optimal. Sonst wähle ein  $j_0$  mit  $\Delta_{j_0} > 0$ .

Schritt 3: Wenn  $(B^{-1}N)_{ij_0} \leq 0$  ist für alle  $i$ , dann ist die LOA nicht lösbar, die Zielfunktion ist über dem zulässigen Bereich nach unten unbeschränkt. Sonst bestimme die Werte  $\theta_i$  für alle  $i$  mit  $(B^{-1}N)_{ij_0} > 0$  sowie

$$\theta_{i_0} = \min\{\theta_i : (B^{-1}N)_{ij_0} > 0\}.$$

Schritt 4: Rechne die gesamte Simplextabelle um: In den Spalten  $x_B$ ,  $c_B$  wird die Basisvariable durch  $x_{j_0}$  und deren Zielfunktionskoeffizient durch  $c_{j_0}$  ersetzt. Der Rest der Tabelle wird mit Hilfe des Gauß-Jordan-Algorithmus umgerechnet, somit wird in der Spalte  $j_0$  eine Einheitsspalte mit einer Eins in der Zeile  $i_0$  geschaffen.

In Bemerkung 1.15 wurde erwähnt, dass der Simplexalgorithmus im Allgemeinen nach endlich vielen Iterationen eine optimale Lösung liefert. Im Ausnahmefall lässt sich ein

Zyklus im Simplexverfahren beobachten. Um dieses zu vermeiden wurden die folgenden Regeln definiert:

**Definition 1.23** *Bland'sche Regeln:*

- (1) Wahl der Spalte  $j_0$  mit  $\Delta_{j_0} > 0$  und dem kleinsten Index  $j_0$  im zweiten Schritt des Simplexalgorithmus.
- (2) Wahl der Zeile  $i_0$  so dass gilt:  $\frac{(B^{-1}b)_{i_0}}{(B^{-1}A)_{i_0j_0}} \leq \frac{(B^{-1}b)_i}{(B^{-1}A)_{ij_0}}$  für alle  $i$  für die  $(B^{-1}A)_{ij_0} > 0$  ist, mit dem kleinsten Index  $i_0$  im dritten Schritt des Algorithmus.

## 1.5. Berechnung einer ersten Basislösung

**Sonderfall:** Ausgangsaufgabe vorgegeben als

$$\begin{aligned} \langle c, x \rangle &\longrightarrow \min \\ Ax &\leq b \\ x &\geq 0 \end{aligned}$$

mit  $b \geq 0$ : Einführung von Schlupfvariablen  $u_i$ , Überführung in Normalform.

Startbasislösung: Basisvariable sind  $u_i$ ,  $i = 1, 2, \dots, m$

Basismatrix: E (Einheitsmatrix)

**Allgemeiner Fall:** Aufgabe in Normalform gegeben

$$\begin{aligned} \langle c, x \rangle &\longrightarrow \min \\ Ax &= b \\ x &\geq 0 \end{aligned}$$

Zur Berechnung einer ersten zulässigen Basislösung wird eine künstliche Aufgabe, die sogenannte Aufgabe der ersten Phase, aufgestellt:

$$\begin{aligned} \sum_{i=1}^m v_i &\longrightarrow \min \\ \sum_{j=1}^n a_{ij}x_j + v_i &= b_i, \text{ für alle } i = 1, 2, \dots, m \\ x, v &\geq 0 \end{aligned}$$

Hier sind  $v_i$  - künstliche Variable.

Diese Aufgabe hat die folgenden Eigenschaften:

- (1) Die Aufgabe besitzt immer eine sofort ablesbare zulässige Basislösung:

$$\bar{x} = (x_B, x_N)^\top \text{ mit } x_B = v = b \text{ und } x_N = x = 0.$$

- (2) Der Zielfunktionswert jeder zulässigen Basislösung ist nicht negativ. Damit ist die Aufgabe der ersten Phase immer lösbar. Sei  $f^*$  der optimale Zielfunktionswert. Es können also nur zwei Fälle auftreten: entweder es ist  $f^* = 0$  oder  $f^* > 0$ .
- (3) Sei  $\bar{x}$  eine zulässige Basislösung für die Aufgabe in Normalform (1.1). Dann ist  $(\bar{x}, 0)$  eine zulässige Basislösung für die Aufgabe der ersten Phase, deren Zielfunktionswert gleich null ist. Damit ist diese Lösung optimal für die Aufgabe der ersten Phase.



- (4) Sei  $(\bar{x}, \bar{v})$  eine optimale Basislösung für die Aufgabe der ersten Phase mit dem Zielfunktionswert null. Dann ist  $\bar{v} = 0$  und  $\bar{x}$  ist eine zulässige Lösung für die Aufgabe in Normalform.

**Satz 1.24** *Die Aufgabe in Normalform hat eine zulässige Basislösung genau dann, wenn die Aufgabe der ersten Phase den optimalen Zielfunktionswert null besitzt. Aus der optimalen Basislösung  $(\bar{x}, \bar{v})$  der Aufgabe der ersten Phase läßt sich dann eine zulässige Basislösung der Aufgabe in Normalform konstruieren.*

**Bemerkung 1.25** *Wenn eine zulässige Basislösung der Aufgabe in Normalform nicht in der optimalen Tabelle der ersten Phase ablesbar ist (d.h. es ist noch eine künstliche Variable in der Basis), so sind die in der Basis verbliebenen künstlichen Variablen durch weitere Austauschschritte aus der Basis zu entfernen. Bei kleinen Aufgaben hilft oftmals eine kritische Betrachtung der optimalen Simplextablelle, um einen Weg zur Realisierung dieser Schritte zu finden.*

Damit ergibt sich der folgende Zwei-Phasen-Algorithmus zur Lösung linearer Optimierungsaufgaben (in diesem Zusammenhang wird die Aufgabe in Normalform auch als Aufgabe der zweiten Phase bezeichnet):

**Zwei-Phasen-Algorithmus:**

Schritt 1: Transformation der gegebenen Aufgabe in Normalform.

Schritt 2: Wenn in  $A$  eine Einheitsmatrix vorhanden ist, gehe zu Schritt 5.

Schritt 3: Konstruiere und löse die Aufgabe der ersten Phase.

- a) Wenn der optimale Zielfunktionswert  $f^*$  der Aufgabe der ersten Phase größer als null ist, stopp, die zu lösende lineare Optimierungsaufgabe hat einen leeren zulässigen Bereich, sie ist nicht lösbar.
- b) Wenn  $f^* = 0$  ist, so gehe zu Schritt 4.

Schritt 4: Berechne die Starttabelle der Aufgabe der zweiten Phase unter Zuhilfenahme der optimalen Tabelle der ersten Phase.

Schritt 5: Löse die Aufgabe der zweiten Phase.

Schritt 6: Rücktransformation der erhaltenen optimalen Lösung in die optimale Lösung der ursprünglichen Aufgabe, falls die Aufgabe der zweiten Phase lösbar war.



## Duale lineare Optimierung

### 2.1. Duale Aufgabe

In der dualen linearen Optimierung werden Paare linearer Optimierungsaufgaben betrachtet. Ausgehend von der Gestalt der primalen linearen Optimierungsaufgabe werden die folgenden Konstruktionsprinzipien zur Konstruktion der dualen Aufgaben verwendet:

Primale Aufgabe	Duale Aufgabe
$\langle c, x \rangle \rightarrow \min$	$\langle b, y \rangle \rightarrow \max$
ZF-Koeffizienten $c_j$	rechte Seite $c_j$
rechte Seite $b_i$	ZF-Koeffizienten $b_i$
Koeffizientenmatrix $A$	Koeffizientenmatrix $A^\top$
Nebenbedingung $\langle a_i, x \rangle \leq b_i$	Vorzeichenbedingung $y_i \leq 0$
Nebenbedingung $\langle a_i, x \rangle \geq b_i$	Vorzeichenbedingung $y_i \geq 0$
Nebenbedingung $\langle a_i, x \rangle = b_i$	keine Vorzeichenbedingung für $y_i$
Vorzeichenbedingung $x_j \geq 0$	Nebenbedingung $\langle a_j, y \rangle \leq c_j$
Vorzeichenbedingung $x_j \leq 0$	Nebenbedingung $\langle a_j, y \rangle \geq c_j$
keine Vorzeichenbedingung für $x_j$	Nebenbedingung $\langle a_j, y \rangle = c_j$

In der Tabelle bezeichnet  $a_i$  die Zeile und  $a_j$  die Spalte der Matrix  $A$ . Durch Anwendung dieser Prinzipien entstehen speziell die folgenden Paare dualer Aufgaben:

$$\left. \begin{array}{l} \langle c, x \rangle \rightarrow \min \\ Ax = b \\ x \geq 0 \end{array} \right\} \quad (\text{NF}) \quad \left. \begin{array}{l} \langle b, y \rangle \rightarrow \max \\ A^\top y \leq c \end{array} \right\} \quad (\text{DNF})$$

sowie

$$\left. \begin{array}{l} \langle c, x \rangle \rightarrow \min \\ Ax \leq b \\ x \geq 0 \end{array} \right\} \quad (\text{PLOA}) \quad \left. \begin{array}{l} \langle b, y \rangle \rightarrow \max \\ A^\top y \leq c \\ y \leq 0 \end{array} \right\} \quad (\text{DLOA})$$

**Definition 2.1** Die Aufgaben (DNF) und (DLOA) heißen *duale lineare Optimierungsaufgaben* zu den primalen linearen Optimierungsaufgaben (NF) beziehungsweise (PLOA).

**Bemerkung 2.2** Wenn die eigentlich untersuchte Aufgabe eine Maximierungsaufgabe ist, wird sie besser als „duale Aufgabe“ betrachtet und mit Hilfe der obigen Konstruktionsregeln um die „primale Aufgabe“ ergänzt.

**Bemerkung 2.3** Die duale Aufgabe zur dualen Optimierungsaufgabe ist wieder die primale Aufgabe.

### 2.2. Dualitätssätze

Die folgenden Sätze sind für das Paar primal-dualer linearer Optimierungsaufgaben (NF), (DNF) aufgeschrieben. Sie gelten analog auch für alle anderen möglichen Paare.

**Satz 2.4** Seien  $\hat{x}$  eine zulässige Lösung für die Aufgabe (NF) und  $\hat{y}$  eine zulässige Lösung für die Aufgabe (DNF). Dann gilt

$$\langle c, \hat{x} \rangle \geq \langle b, \hat{y} \rangle.$$

Die im Satz 2.4 beschriebene Eigenschaft wird als *schwache Dualität* bezeichnet.

**Folgerung 2.5** Für die optimalen Zielfunktionswerte  $f^*$  von (NF) und  $\varphi^*$  von (DNF) gilt folglich ebenfalls  $f^* \geq \varphi^*$ .

**Satz 2.6** Seien  $\hat{x}$  eine zulässige Lösung für die Aufgabe (NF) und  $\hat{y}$  eine zulässige Lösung für die Aufgabe (DNF). Dann sind folgende Aussagen äquivalent:

- (1)  $\hat{x}$  ist optimal für (NF),  $\hat{y}$  ist optimal für (DNF),
- (2)  $\langle c, \hat{x} \rangle = \langle b, \hat{y} \rangle$  (starke Dualität),
- (3)  $\langle \hat{x}, A^\top \hat{y} - c \rangle = 0$ .

**Bemerkung 2.7** Die letzte Bedingung, auch Komplementaritätsbedingung genannt, lautet äquivalent komponentenweise aufgeschrieben:

$$\hat{x}_j (A^\top \hat{y} - c)_j = 0 \quad \forall j = 1, \dots, n.$$

**Satz 2.8** Folgende 5 Aussagen sind äquivalent:

- (1) (NF) ist lösbar,
- (2) (DNF) ist lösbar,
- (3) es gibt  $\hat{x}, \hat{y}$  mit den Eigenschaften  $\hat{x} \geq 0$ ,  $A\hat{x} = b$ ,  $A^\top \hat{y} \leq c$ ,
- (4) es gibt  $\hat{x}, C$  mit den Eigenschaften  $\hat{x} \geq 0$ ,  $C > -\infty$ ,  $A\hat{x} = b$  und es ist  $\langle c, x \rangle \geq C$  für alle  $x \geq 0$ ,  $Ax = b$ ,
- (5) es gibt  $\hat{y}, D$  mit den Eigenschaften  $A^\top \hat{y} \leq c$ ,  $D < \infty$  und es ist  $\langle b, y \rangle \leq D$  für alle  $y$  mit  $A^\top y \leq c$ .

### 2.3. Interpretation der dualen Aufgabe

Wir betrachten das Paar dualer Aufgaben

$$\left. \begin{array}{l} \langle c, x \rangle \rightarrow \max \\ Ax \leq b \\ x \geq 0 \end{array} \right\} \quad (\text{PLOA}) \quad \left. \begin{array}{l} \langle b, y \rangle \rightarrow \min \\ A^\top y \geq c \\ y \geq 0 \end{array} \right\} \quad (\text{DLOA})$$

und verwenden die folgenden Interpretationen der Daten:

- (1)  $x_j$  – hergestellte Mengen an Produkt  $j$
- (2)  $c_j$  – Gewinn pro hergestellter Menge Produkt  $j$
- (3)  $a_{ij}$  – Faktoreinsatzmenge des Faktors  $i$  zur Herstellung einer Mengeneinheit des Produktes  $j$
- (4)  $b_i$  – vorhandene Menge des Faktors  $i$  (sog. Fonds)

Die Gleichheit der Maßeinheiten (und der optimalen Werte) der Zielfunktionen der primalen und der dualen Aufgabe impliziert dann die Maßeinheit  $\left[ \frac{\text{Geldeinheiten}}{\text{Mengeneinheit Faktor } i} \right]$  für die dualen Variablen.

$$\langle c, \hat{x} \rangle = \langle b, \hat{y} \rangle \Rightarrow [\text{€}] = [\text{Mengeneinheit Faktor } i] \left[ \frac{\text{€}}{\text{Mengeneinheit Faktor } i} \right]$$

Diese sind damit finanzielle Bewertungen der vorhandenen (bzw. verwendeten) Mengen des Faktors  $i$ .

**Definition 2.9** Die optimalen Lösungen der dualen Aufgabe heißen Schattenpreise.

Die Schattenpreise stellen einen ideellen Wert (Preis) der Faktoren für die zu lösende Aufgabe dar.

Interpretation der Bedingungen der Aufgabe:

- (1)  $A^T y \geq c$  – Schattenpreis für die verwendeten Faktormengen zur Herstellung einer Mengeneinheit von Produkt  $j$  ist mindestens so groß wie der Gewinn pro hergestellter Mengeneinheit.
- (2) Bedingungen aus dem Satz über die starke Dualität:  
 $x_j(A^T y - c)_j = 0$  – Ist der Schattenpreis für die verwendeten Faktormengen größer als der Gewinn, so wird Produkt  $j$  nicht hergestellt.
- (3) analog:  
 $y_i(Ax - b)_i = 0$  – Werden vorhandene Fonds nicht ausgenutzt, so ist deren Schattenpreis null.

Anwendung bei der Entscheidung über den Zukauf von Faktoren:

- (1) Sei die optimale Lösung  $y^*$  der dualen Aufgabe eindeutig. Werden  $\theta$  ME des Faktors  $i$  hinzugekauft (Vergrößerung der  $b_i$ ), so ändert sich der optimale Zielfunktionswert um etwa  $\theta y_i^*$  Geldeinheiten. Damit wird der Zukauf realisiert, wenn der Schattenpreis  $y_i^*$  größer als der Marktpreis einer Mengeneinheit des Faktors  $i$  ist.  
**Achtung:** Diese Aussage gilt nur für kleine Werte von  $|\theta|$  (so klein, dass es keinen Basiswechsel gibt), allerdings analog auch für den Verkauf von Faktoren.
- (2) Wenn die optimale Lösung der dualen Aufgabe nicht eindeutig ist, so gilt die Aussage für den Zukauf für den kleinsten Wert von  $y_i^*$  unter allen optimalen Lösungen der dualen Aufgabe.

## 2.4. Berechnung einer optimalen Lösung der dualen Aufgabe

Berechnung einer optimalen Lösung der dualen Aufgabe im Spezialfall der Aufgabe:

$$\left. \begin{array}{l} \langle c, x \rangle \rightarrow \min \\ Ax \leq b \\ x \geq 0 \end{array} \right\}$$

mit Hilfe des Simplexalgorithmus. Nach Einführung der Schlupfvariablen ergibt sich die folgende Simplextabelle:

		$c_1$	$c_2$	$\cdots$	$c_n$	0	$\cdots$	0	
$x_B$	$c_B$	$x_1$	$x_2$	$\cdots$	$x_n$	$u_1$	$\cdots$	$u_m$	$\bar{b}$
	$u$	$A$				$E$			$b$
		$-c_1$	$-c_2$	$\cdots$	$-c_n$	0	$\cdots$	0	0

Nach den notwendigen Iterationen erhalten wir:

$$\Downarrow$$

	$B^{-1}A$				$B^{-1}$			$B^{-1}b$
	$\Delta_1$	$\Delta_2$	$\cdots$	$\Delta_n$	$y_1^*$	$\cdots$	$y_m^*$	$f^*$

In der Tabelle sind dann die Inverse der zur optimalen Lösung gehörenden Basismatrix ( $B^{-1}$ ) und die Schattenpreise ( $y_i^*$ ) ablesbar. Im allgemeinen Fall kann durch Aufnahme der Einheitsmatrix in die Tabelle (ohne Beachtung der entsprechenden Elemente in der letzten Zeile im Simplexalgorithmus!) ein analoger Effekt erzielt werden.

## 2.5. Dualer Simplexalgorithmus

Die Idee: Verwende den primalen Simplexalgorithmus zur Lösung der Dualaufgabe. Der Zulässigkeitstest für die duale Aufgabe entspricht dem Optimalitätstest für die primale Aufgabe.

### Dualer Simplexalgorithmus

Schritt 1: Wähle eine Basismatrix  $B$  mit  $\Delta = c^T B^{-1} A - c^T \leq 0$  und stelle die Simplextabelle auf.

Schritt 2: Wenn  $B^{-1}b \geq 0$  in allen Komponenten ist, so ist die vorliegende Lösung optimal - Stopp. Sonst: Wähle ein  $i_0 : (B^{-1}b)_{i_0} < 0$ .

Schritt 3: Falls  $(B^{-1}A)_{i_0j} \geq 0$  für alle  $j$  ist, dann ist die Optimierungsaufgabe nicht lösbar, da der zulässige Bereich leer ist - Stopp. Sonst: Wähle ein  $j_0 : \frac{\Delta_{j_0}}{(B^{-1}A)_{i_0j_0}} \leq \frac{\Delta_{j_0}}{(B^{-1}A)_{i_0j}}$  für alle  $j : (B^{-1}A)_{i_0j} < 0$ .

Schritt 4: Umrechnung der Tabelle:  $x_{j_0}$  wird neue Basisvariable, während  $x_{i_0}$  die Basis verlässt.

Anwendung des dualen Simplexalgorithmus:

- Falls die Bestimmung der Startbasislösung mit geringem Aufwand möglich ist.
- Die optimale Lösung der Aufgabe sowie die optimale Simplextabelle liegt vor, danach wird die rechte Seite  $b$  verändert und eine neue optimale Lösung soll bestimmt werden.
- Nach der Lösung der LOA wird eine neue Nebenbedingung angefügt, danach soll eine neue Lösung bestimmt werden.

## Sensitivitätsanalyse

**Aufgabenstellung:** Sei  $x^*$  optimale Lösung der Aufgabe (NF)

$$\begin{aligned} \langle c, x \rangle &\longrightarrow \min \\ Ax &= b \\ x &\geq 0. \end{aligned}$$

Dann sind die folgenden Fragen von Interesse:

- (1) Was passiert mit der optimalen Lösung  $x^*$  und dem optimalen Zielfunktionswert  $f^*$ , wenn sich ein Zielfunktionskoeffizient oder auch die gesamte Zielfunktion ändert?
- (2) Die gleiche Fragestellung, wenn sich Koeffizienten der rechten Seite ändern.

### 3.1. Allgemeine Veränderung der Zielfunktionskoeffizienten

Anwendung: Änderung von Kosten, Preisen, etc.

Sei  $x^*$  optimale Lösung der Aufgabe (NF). Ist  $x^*$  dann auch optimal für die Aufgabe

$$\begin{aligned} \langle \hat{c}, x \rangle &\longrightarrow \min \\ Ax &= b \\ x &\geq 0 \end{aligned} \tag{3.1}$$

mit  $\hat{c} \neq c$ ?

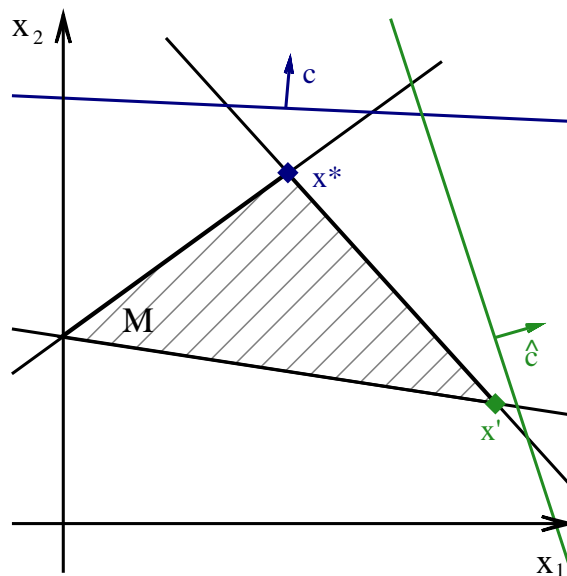


ABBILDUNG 3.1. Sensitivität: Änderung der Zielfunktion

Offensichtlich ist  $x^*$  zulässig, da weiterhin  $Ax^* = b, x^* \geq 0$  gilt. Änderung von  $c$  zu  $\hat{c}$  bewirkt eine Drehung der Zielfunktion, was zu der Veränderung der optimalen Lösung führen kann, wie die Abbildung 3.1 illustriert.

**Satz 3.1** Sei  $x^*$  optimale Basislösung der Aufgabe (NF) zur Basismatrix  $B$ , d.h.  $x^* = (x_B^* \ x_N^*)^\top$  mit  $x_B^* = B^{-1}b, x_N^* = 0$ . Dann ist  $x^*$  auch optimale Basislösung der Aufgabe (3.1), falls alle Optimalitätsindikatoren für diese Aufgabe nichtpositiv sind:

$$\Delta_j = (\hat{c}_B^\top B^{-1}A)_j^\top - \hat{c}_j \leq 0 \quad \forall j = 1, \dots, n.$$

Wenn diese Eigenschaft nicht gilt, so müssen Schritte des Simplexalgorithmus ausgeführt werden, um eine neue optimale Lösung zu berechnen oder die Unlösbarkeit festzustellen.

Falls eine optimale Simplextabelle der Ausgangsaufgabe (NF) vorliegt, berechnet man die Optimalitätsindikatoren  $\Delta_j$  mit den neuen Zielfunktionskoeffizienten um zu überprüfen, ob die Lösung auch für die Aufgabe (3.1) optimal ist.

### 3.2. Ein Parameter in der Zielfunktion

Wir betrachten jetzt die Aufgabe

$$\begin{aligned} \langle c + t\hat{c}, x \rangle &\longrightarrow \min \\ Ax &= b \\ x &\geq 0 \end{aligned} \tag{3.2}$$

mit  $t \in \mathbb{R}$  und wollen alle optimalen Lösungen für diese Aufgabe für alle Werte von dem Parameter  $t$  bestimmen. Für eine Basislösung  $x^*$ , die optimal für den Parameterwert  $t = t^*$  ist, ist es möglich wie im Abschnitt 3.1 vorzugehen, so erhält man aber anstelle der Ungleichungen in Satz 3.1 die folgenden Ungleichungen:

$$\Delta_j(t) = ((c_B + t\hat{c}_B)^\top B^{-1}A)_j^\top - (c + t\hat{c})_j \leq 0 \quad \forall j = 1, \dots, n, \tag{3.3}$$

die jetzt von  $t$  abhängen. Dieses Ungleichungssystem kann wie folgt nach  $t$  aufgelöst werden:

$$\begin{aligned} t &\leq \frac{-(c_B^\top B^{-1}A)_j^\top + c_j}{(\hat{c}_B^\top B^{-1}A)_j^\top - \hat{c}_j}, \quad \text{falls } (\hat{c}_B^\top B^{-1}A)_j^\top - \hat{c}_j > 0, \\ t &\geq \frac{-(c_B^\top B^{-1}A)_j^\top + c_j}{(\hat{c}_B^\top B^{-1}A)_j^\top - \hat{c}_j}, \quad \text{falls } (\hat{c}_B^\top B^{-1}A)_j^\top - \hat{c}_j < 0. \end{aligned}$$

Die Lösungsmenge der Ungleichungen ist ein Intervall  $[\underline{t}, \bar{t}]$  (das nicht leer sein muss), wobei die Fälle  $\underline{t} = -\infty$  beziehungsweise  $\bar{t} = +\infty$  möglich sind. Für  $t = \underline{t}$  oder  $t = \bar{t}$  ist ein Optimalitätsindikator einer Nichtbasisvariablen null, es gibt also eventuell eine alternative optimale Lösung. Ist dies der Fall und wird die Simplextabelle umgerechnet, um diese neue Lösung darzustellen, so ergibt sich nun analog zu obigem (aus den Ungleichungen der Optimalitätsindikatoren) ein neues Intervall und so weiter. Folgende Bezeichnungen werden verwendet:

$$Q = \{t : \text{Problem (3.2) ist lösbar}\}$$

bezeichnet die Menge aller Parameter  $t$ , für die die Aufgabe (3.2) eine optimale Lösung besitzt,

$$\varphi(t) = \min\{\langle c + t\hat{c}, x \rangle : Ax = b, x \geq 0\}$$



bezeichnet den optimalen Zielfunktionswert und

$$\Psi(t) = \{x \geq 0 : Ax = b, \langle c + t\hat{c}, x \rangle = \varphi(t)\}$$

die Menge optimaler Lösungen der Aufgabe (3.2) für einen festen Wert des Parameters  $t \in Q$ . Dann ergibt sich

**Satz 3.2** *Betrachtet werde die Aufgabe (3.2). Sei  $Q \neq \emptyset$ . Dann gibt es Zahlen  $-\infty < t_1 < t_2 < \dots, t_p < \infty$ , so dass folgendes gilt:*

(1)  $\varphi(t)$  ist auf  $[t_i, t_{i+1}]$  (affin-) linear :

$$\varphi(t) = \langle c, \hat{x} \rangle + t\langle \hat{c}, \hat{x} \rangle$$

für eine beliebige Lösung  $\hat{x} \in \Psi(t)$ ,  $t \in (t_i, t_{i+1})$ ,  $i = 1, \dots, p - 1$ .

(2)  $\Psi(t)$  ist auf  $(t_i, t_{i+1})$  konstant,  $i = 1, \dots, p - 1$ . Für  $t = t_i$  ist  $\Psi(t) \subset \Psi(t_i)$  für alle  $t \in (t_{i-1}, t_i) \cup (t_i, t_{i+1})$ ,  $i = 2, \dots, p - 1$ .

(3) Aussagen 1) und 2) gelten analog auch auf  $(-\infty, t_1) \cup (t_p, \infty)$ , falls die Aufgabe dort lösbar ist.

(4)  $\varphi(t)$  ist konkav, d.h. es gilt für alle  $\lambda \in [0, 1]$  und für alle  $t', t'' \in Q$

$$\varphi(\lambda t' + (1 - \lambda)t'') \geq \lambda\varphi(t') + (1 - \lambda)\varphi(t'').$$

(5) Für  $s, t \in (t_i, t_{i+1})$  ist

$$\varphi(t) = \langle c, \hat{x} \rangle + t\langle \hat{c}, \hat{x} \rangle = \langle c, \hat{x} \rangle + s\langle \hat{c}, \hat{x} \rangle + (t - s)\langle \hat{c}, \hat{x} \rangle$$

also

$$\varphi'(s) = \lim_{t \rightarrow s} \frac{\varphi(t) - \varphi(s)}{t - s} = \langle \hat{c}, \hat{x} \rangle$$

und für  $s = t_i$  lässt sich der optimale Zielfunktionswert wie folgt definieren:

$$\varphi(t) = \varphi(t_i) + \min\{(t - t_i)\langle \hat{c}, \hat{x} \rangle : \hat{x} \in \Psi(t_i)\}$$

für  $t \in [t_{i-1}, t_{i+1}]$ .

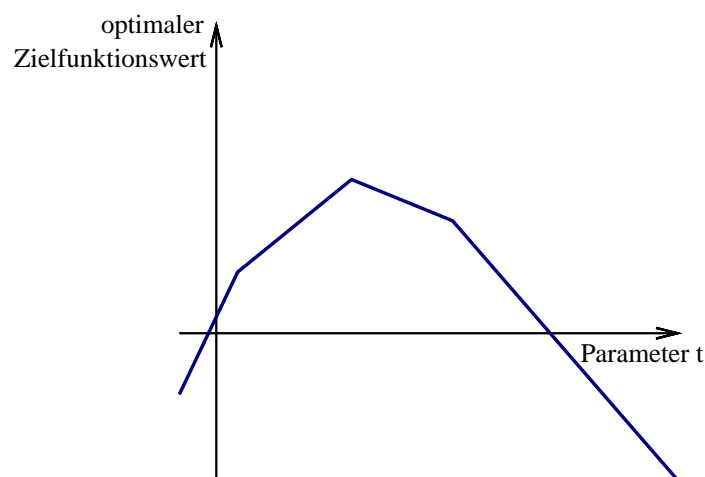


ABBILDUNG 3.2. Sensitivität: Optimalwert-Funktion

Die Funktion  $\varphi(t)$  ist beispielhaft in Abbildung 3.2 dargestellt.

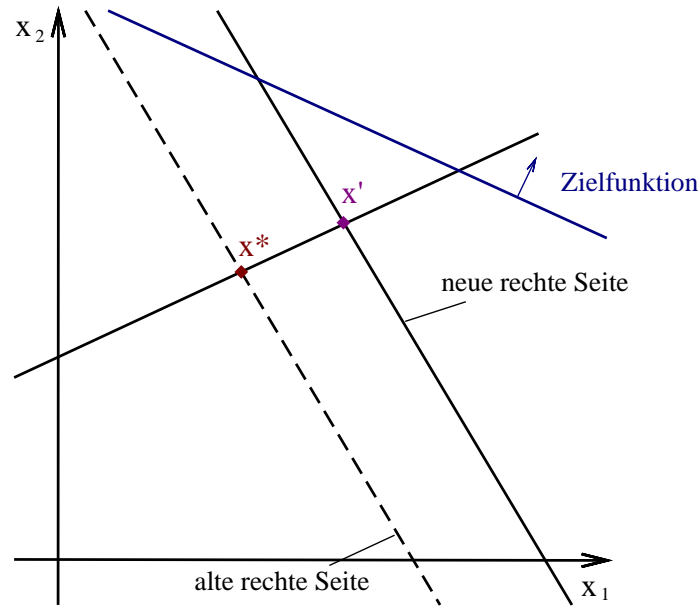


ABBILDUNG 3.3. Sensitivität: Variation rechte Seite

### 3.3. Veränderung der rechten Seite

Sei die Aufgabe (1.1) gelöst worden. Sei  $x^*$  eine optimale Lösung und  $\bar{B}$  die entsprechende Basismatrix. Nach einer Änderung des Vektors der rechten Seite sei die neue Aufgabe

$$\begin{aligned} \langle c, x \rangle &\rightarrow \min \\ Ax &= \hat{b} \\ x &\geq 0 \end{aligned} \quad (3.4)$$

zu lösen. Zur Veranschaulichung der Situation dient Bild 3.3. Die Änderung des Vektors  $b$  impliziert eine Verschiebung der entsprechenden Geraden aus dem zulässigen Bereich.

Dann ist  $x^*$  keine optimale Lösung der Aufgabe (3.4), da  $x^*$  nicht mehr zulässig ist ( $Ax^* \neq \hat{b}$  wenn  $b \neq \hat{b}$ ). Sei

$$\hat{x} = \begin{pmatrix} \hat{x}_B \\ \hat{x}_N \end{pmatrix}, \quad \hat{x}_B = \bar{B}^{-1}\hat{b}, \quad \hat{x}_N = 0. \quad (3.5)$$

Wenn dieser Punkt zulässig ist, d.h. wenn  $\bar{B}^{-1}\hat{b} \geq 0$  ist, so ist  $\hat{x}$  optimal für (3.4), da sich die Optimalitätsindikatoren

$$\Delta_j = (c_B^\top \bar{B}^{-1} A)_j^\top - c_j \leq 0 \quad \forall j = 1, \dots, n$$

nicht geändert haben. Damit gilt

**Satz 3.3** Die Lösung  $\hat{x} = \begin{pmatrix} \hat{x}_B \\ \hat{x}_N \end{pmatrix}$  mit  $\hat{x}_B = \bar{B}^{-1}\hat{b}, \hat{x}_N = 0$  ist optimal für die neue Aufgabe (3.4), wenn sie zulässig ist:  $\bar{B}^{-1}\hat{b} \geq 0$ .

### 3.4. Ein Parameter in der rechten Seite

Sei jetzt die Aufgabe (3.4) konkretisiert zu

$$\begin{aligned} \langle c, x \rangle &\rightarrow \min \\ Ax &= b + t\hat{b} \\ x &\geq 0 \end{aligned} \quad (3.6)$$

für  $t \in \mathbb{R}$ . Dann betrachten wir anstelle von (3.5) die parameterabhängige Lösung

$$\hat{x}(t) = \begin{pmatrix} \hat{x}_B(t) \\ \hat{x}_N \end{pmatrix}, \quad \hat{x}_B(t) = \bar{B}^{-1}(b + t\hat{b}), \quad \hat{x}_N = 0. \quad (3.7)$$

Diese Lösung ist zulässig (und damit optimal), wenn  $\bar{B}^{-1}(b + t\hat{b}) \geq 0$  gilt. Dieses ist ein Ungleichungssystem mit einer Variablen  $t$ , welches wie folgt nach  $t$  aufgelöst werden kann:

$$\begin{aligned} t &\geq -\frac{(\bar{B}^{-1}b)_j}{(\bar{B}^{-1}\hat{b})_j}, \quad \text{falls } (\bar{B}^{-1}\hat{b})_j > 0, \\ t &\leq -\frac{(\bar{B}^{-1}b)_j}{(\bar{B}^{-1}\hat{b})_j}, \quad \text{falls } (\bar{B}^{-1}\hat{b})_j < 0. \end{aligned}$$

Die rechten Seiten dieser Ungleichungen bestimmen Grenzen eines Intervalls, in dem  $t$  variieren kann ohne Änderung der optimalen Basismatrix. Allgemein ergibt sich der folgende Satz, wobei wiederum  $\varphi(t)$  den optimalen Zielfunktionswert

$$\varphi(t) = \min\{\langle c, x \rangle : Ax = b + t\hat{b}, x \geq 0\}$$

und  $\Psi(t)$  die Menge optimaler Lösungen

$$\Psi(t) = \{x \geq 0 : Ax = b + t\hat{b}, \langle c, x \rangle = \varphi(t)\}$$

der Aufgabe (3.6) für ein festes

$$t \in Q = \{t : \text{Aufgabe (3.6) ist lösbar}\}$$

bezeichnet.

**Satz 3.4** *Zu lösen sei die Aufgabe (3.6). Sei  $Q \neq \emptyset$ . Dann gibt es Zahlen  $-\infty < t_1 < t_2 < \dots < t_k < \infty$  mit folgenden Eigenschaften:*

- (1) *Die Funktion  $\varphi(t)$  ist über jedem der Intervalle  $[t_i, t_{i+1}]$ ,  $t = 1, \dots, k-1$ , (affin-) linear:*

$$\varphi(t) = \langle c_B, B_i^{-1}(b + t\hat{b}) \rangle = \langle b + t\hat{b}, y^i \rangle,$$

*wobei  $B_i$  eine Basismatrix und  $y^i$  eine optimale Lösung der dualen Aufgabe sind.*

- (2) *Für jedes Intervall  $(t_i, t_{i+1})$  gibt es eine Basismatrix  $B_i$ , so dass  $\hat{x} = (\hat{x}_B, \hat{x}_N)^\top$  mit  $\hat{x}_B = B_i^{-1}(b + t\hat{b})$ ,  $\hat{x}_N = 0$  für alle  $t \in (t_i, t_{i+1})$  optimal ist. Für  $t = t_i$  gibt es mehrere optimale Basismatrizen.*
- (3) *Wenn die Aufgabe (3.6) über dem Intervall  $(-\infty, t_1)$  bzw.  $(t_k, \infty)$  lösbar ist, dann ist  $\varphi(t)$  auch dort (affin-) linear.*
- (4) *Die Funktion  $\varphi(t)$  ist konvex, d.h. es gilt für alle  $\lambda \in [0, 1]$  und für alle  $t', t'' \in Q$ ,  $|\varphi(t')| < \infty$ ,  $|\varphi(t'')| < \infty$ , stets*

$$\varphi(\lambda t' + (1 - \lambda)t'') \leq \lambda\varphi(t') + (1 - \lambda)\varphi(t'').$$

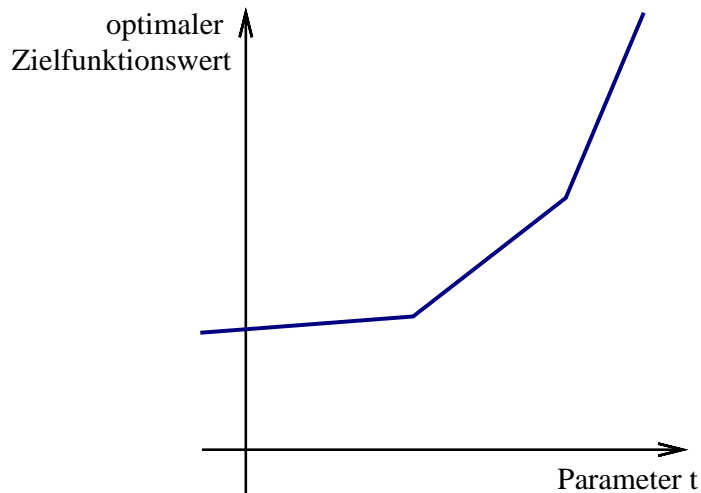


ABBILDUNG 3.4. Sensitivität: Optimalwert-Funktion

(5) Im Intervall  $(t_i, t_{i+1})$  ist die Funktion  $\varphi(t)$  differenzierbar und es gilt

$$\varphi'(t) = \langle \hat{b}, y^i \rangle, \quad t \in (t_i, t_{i+1})$$

In den Punkten  $t_i$  ist

$$\varphi(t) = \varphi(t_i) + \max\{\langle \hat{b}, y^* \rangle(t - t_i) : y^* \text{ löst die duale Aufgabe für } t_i\}$$

für  $t \in [t_{i-1}, t_{i+1}]$ .

Die Funktion  $\varphi(t)$  ist beispielhaft in Abbildung 3.4 dargestellt.

### 3.5. Aufnahme einer neuen Variablen

Sei  $x^* = \begin{pmatrix} x_B^* \\ x_N^* \end{pmatrix}$ ,  $x_B^* = B^{-1}b$ ,  $x_N^* = 0$  eine optimale Basislösung der Aufgabe

$$\begin{aligned} \sum_{j=1}^n c_j x_j &\rightarrow \min \\ \sum_{j=1}^n a_{ij} x_j &= b_i, \quad i = 1, \dots, m \\ x_j &\geq 0, \quad j = 1, \dots, n. \end{aligned} \tag{3.8}$$

Es sei jetzt die neue Aufgabe

$$\begin{aligned} \sum_{j=1}^n c_j x_j + c_{n+1} x_{n+1} &\rightarrow \min \\ \sum_{j=1}^n a_{ij} x_j + a_{i,n+1} x_{n+1} &= b_i, \quad i = 1, \dots, m \\ x_j &\geq 0, \quad j = 1, \dots, n, \quad x_{n+1} \geq 0. \end{aligned} \tag{3.9}$$

zu lösen. Ist dann der Punkt  $\hat{x} = \begin{pmatrix} x^* \\ 0 \end{pmatrix}$  optimal für diese Aufgabe?

Sei dazu

$$\hat{A} = (a_{ij})_{i=1, j=1}^{m, n+1}$$

die aus  $A$  entstehende Matrix mit nun  $n + 1$  Spalten. Diese entsteht aus der Koeffizientenmatrix von (3.8) durch Hinzufügung der zur neuen Variablen  $x_{n+1}$  gehörenden Spalte  $a_{i,n+1}$  in (3.9).

**Satz 3.5** Wenn der  $(n + 1)$ -te Optimalitätsindikator

$$\Delta_{n+1} = (c_B^\top B^{-1} \hat{A})_{n+1} - c_{n+1} \leq 0$$

ist, so ist die Lösung  $\hat{x}$  optimal für die Aufgabe (3.9). Andernfalls ist mit dem Simplexalgorithmus eine optimale Lösung zu berechnen.

Dieser Satz gibt eine untere Schranke für die Größe des Zielfunktionskoeffizienten  $c_{n+1}$  an, bei deren Einhaltung sich eine Aufnahme der Variablen  $x_{n+1}$  in die Basis nicht lohnt. Erst wenn

$$c_{n+1} < (c_B^\top B^{-1} \hat{A})_{n+1} = (y^{*\top} \hat{A})_{n+1}$$

ist, so kann eine Aufnahme der Variablen  $x_{n+1}$  in die Basis einen kleineren optimalen Zielfunktionswert in der Aufgabe (3.8) als  $c^\top \hat{x}$  erzeugen. Dabei ist  $y^*$  die (der Einfachheit halber als eindeutig angenommene; vgl. die Ausführungen im Abschnitt 2.3) optimale Lösung der dualen Aufgabe zu (3.8). Die rechte Seite  $(y^{*\top} \hat{A})_{n+1}$  in dieser Ungleichung kann als Schattenpreis der für die Herstellung einer Mengeneinheit des  $(n + 1)$ -ten Produktes verwendeten Faktormengen interpretiert werden.

### 3.6. Aufnahme einer neuen Nebenbedingung

Sei  $x^* = \begin{pmatrix} x_B^* \\ x_N^* \end{pmatrix}$ ,  $x_B^* = B^{-1}b$ ,  $x_N^* = 0$  eine optimale Basislösung der Aufgabe

$$\begin{aligned} \sum_{j=1}^n c_j x_j &\rightarrow \min \\ \sum_{j=1}^n a_{ij} x_j &= b_i, \quad i = 1, \dots, m \\ x_j &\geq 0, \quad j = 1, \dots, n. \end{aligned} \tag{3.10}$$

Es sei jetzt die neue Aufgabe

$$\begin{aligned} \sum_{j=1}^n c_j x_j &\rightarrow \min \\ \sum_{j=1}^n a_{ij} x_j &= b_i, \quad i = 1, \dots, m + 1 \\ x_j &\geq 0, \quad j = 1, \dots, n. \end{aligned} \tag{3.11}$$

zu lösen. Ist der Punkt  $x^*$  weiterhin optimal für die Aufgabe (3.11)?

**Satz 3.6** Falls die Lösung  $x^*$  für die neue Aufgabe (3.11) zulässig bleibt ( $x^*$  erfüllt die neue Restriktion), so ist die Lösung  $x^*$  für die Aufgabe (3.11) optimal. Andernfalls ist mit dem Dualen Simplexalgorithmus eine optimale Lösung zu bestimmen.



## Optimierung mit mehreren Zielen

Dieses Problem wird oft auch als Vektoroptimierungsaufgabe bezeichnet.

### 4.1. Modell, Aufgabenstellung

Betrachtet sei hier die Aufgabe

$$\left. \begin{array}{l} \langle c^1, x \rangle \\ \langle c^2, x \rangle \\ \vdots \\ \langle c^k, x \rangle \end{array} \right\} \rightarrow \text{„max“} \quad (4.1)$$

$$\begin{array}{l} Ax = b \\ x \geq 0. \end{array}$$

Kurzschreibweise:

$$\begin{array}{l} Cx \rightarrow \max \\ Ax = b \\ x \geq 0 \end{array} \quad (4.2)$$

In dieser Aufgabe geht es darum, eine solche zulässige Lösung  $\bar{x}$  zu finden, die einen besten Kompromiss zwischen den einzelnen Zielfunktionen darstellt.

**Definition 4.1** Ein Punkt  $\bar{x} \geq 0$  mit  $A\bar{x} = b$  heißt Pareto optimal (effizient) für die Aufgabe (4.1), wenn es keinen Punkt  $\hat{x} \geq 0$  mit  $A\hat{x} = b$  gibt, für den  $C\hat{x} \geq C\bar{x}$  und  $C\hat{x} \neq C\bar{x}$  gelten.

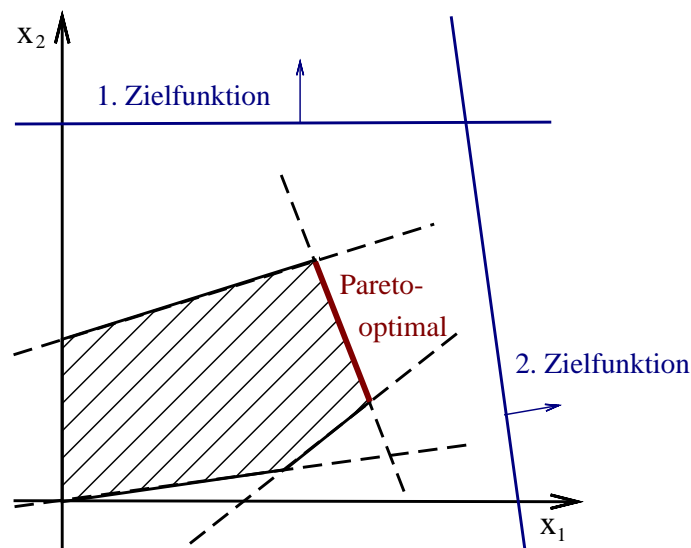


ABBILDUNG 4.1. Lineare Vektoroptimierungsaufgabe

$\bar{x}$  ist Pareto optimal, wenn die Existenz eines zulässigen Punktes  $\hat{x}$  und eines Zeilenindex  $i_0$  mit  $C_{i_0}\hat{x} > C_{i_0}\bar{x}$ , die Existenz eines Zeilenindex  $j_0$  mit  $C_{j_0}\hat{x} < C_{j_0}\bar{x}$  liefert.

Es ergeben sich damit drei Aufgaben:

- (1) Es ist eine Pareto-optimale Lösung zu berechnen.
- (2) Es sind alle Pareto-optimale Lösungen zu berechnen.
- (3) Es ist eine, bezüglich eines bestimmten (weiteren) Kriteriums, beste Pareto-optimale Lösung zu konstruieren. Dazu ist zunächst die Menge aller Pareto-optimale Punkte zu berechnen, aus der dann interaktiv oder durch Anwendung eines Optimierungsalgorithmus eine beste Lösung ausgewählt wird.

## 4.2. Lösungszugang

Zur Lösung der Aufgabe (4.1) wird durch nichtnegative Linearkombination der Zielfunktionen aus (4.1) zu einer einzigen Zielfunktion eine neue lineare Optimierungsaufgabe konstruiert, die dann gelöst werden kann:

$$\begin{aligned} \langle \lambda, Cx \rangle &\rightarrow \max \\ Ax &= b \\ x &\geq 0 \\ \lambda &\geq 0 \end{aligned} \tag{4.3}$$

Dann gilt folgende Aussage:

**Satz 4.2** *Betrachtet seien die Aufgaben (4.1) und (4.3). Dann gilt:*

- (1) *Sei  $\bar{x}$  eine optimale Lösung der Aufgabe (4.3) für einen Vektor  $\lambda$  mit  $\lambda_i > 0, i = 1, \dots, k, \sum_{i=1}^k \lambda_i = 1$ . Dann ist  $\bar{x}$  Pareto-optimal für die Aufgabe (4.1).*
- (2) *Sei  $\bar{x}$  Pareto optimal für die Aufgabe (4.1), dann gibt es einen Vektor  $\lambda$  mit  $\lambda_i > 0, i = 1, \dots, k, \sum_{i=1}^k \lambda_i = 1$ , so dass  $\bar{x}$  eine optimale Lösung der Aufgabe (4.3) ist.*

Damit lassen sich obige Aufgaben umformulieren:

- (1) Wähle ein  $\lambda$  mit  $\lambda_i > 0, i = 1, \dots, k, \sum_{i=1}^k \lambda_i = 1$  und löse die Aufgabe (4.3).
- (2) Bestimme alle optimalen Lösungen der Aufgabe (4.3) für alle  $\lambda_i > 0, i = 1, \dots, k, \sum_{i=1}^k \lambda_i = 1$ .
- (3) Zur Lösung der dritten Aufgabe ist ein zusätzliches Kriterium nötig, womit sich die Güte des Punktes  $x$  ermitteln lässt. Wähle ein  $\lambda_i > 0, i = 1, \dots, k, \sum_{i=1}^k \lambda_i = 1$ , so dass das Kriterium einen bestmöglichen Wert annimmt.

Bei zwei Zielfunktionen ist dabei eine optimale Lösung der Aufgabe

$$\begin{aligned} \lambda \langle c^1, x \rangle + (1 - \lambda) \langle c^2, x \rangle &\rightarrow \max \\ Ax &= b \\ x &\geq 0 \end{aligned} \tag{4.4}$$

für ein  $\lambda \in (0, 1)$  oder für alle  $\lambda \in (0, 1)$  gesucht.



## Transportoptimierung

### 5.1. Einführung, Modell

Gegeben seien  $m$  Angebotsorte eines homogenen Produktes mit den Angebotsmengen  $a_i$ ,  $i = 1, \dots, m$ , welches zu  $n$  Bedarfsorten mit den Bedarfsmengen  $b_j$ ,  $j = 1, \dots, n$  geliefert werden soll. Desweiteren seien die Transportkosten  $c_{ij}$  für den Transport einer Mengeneinheit des Gutes vom Angebotsort  $i$  zum Bedarfsort  $j$  bekannt.

Damit ist folgende Aufgabe zu lösen:

Gesucht sind die Transportmengen  $x_{ij}$  so, dass

- (1) aus den Angebotsorten nicht mehr als  $a_i$  Mengeneinheiten wegtransportiert,
- (2) in die Bedarfsorte nicht weniger als  $b_j$  Mengeneinheiten hineintransportiert werden und
- (3) die gesamten Transportkosten minimal sind.

Eine graphische Darstellung der Transportbeziehungen ist in Bild 5.1 dargestellt.

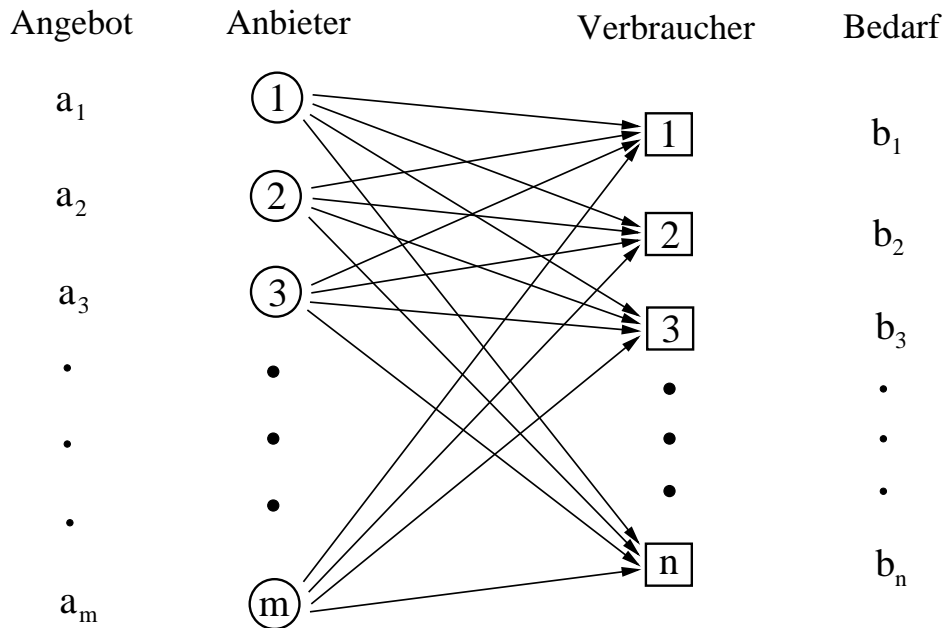


ABBILDUNG 5.1. Graphische Darstellung des Transportproblems

Das Transportproblem (TP) lässt sich wie folgt formulieren:

$$\begin{aligned}
 \sum_{i=1}^m \sum_{j=1}^n c_{ij} x_{ij} &\longrightarrow \min \\
 \sum_{j=1}^n x_{ij} &\leq a_i \quad \text{für alle } i = 1, 2, \dots, m \\
 \sum_{i=1}^m x_{ij} &\geq b_j \quad \text{für alle } j = 1, 2, \dots, n \\
 x_{ij} &\geq 0 \quad \forall i, j
 \end{aligned} \tag{5.1}$$

wobei  $x_{ij}$  die transportierten Mengen vom Angebotsort  $i$  zum Bedarfsort  $j$  bezeichnet.

## 5.2. Eigenschaften

Durch Einsetzen einer zulässigen Lösung der Aufgabe (5.1) erkennt man die folgende Aussage:

**Satz 5.1** *Das Problem (5.1) besitzt zulässige (und bei  $a_i < \infty, i = 1, 2, \dots, n, b_j < \infty, j = 1, 2, \dots, m$  auch optimale) Lösungen genau dann, wenn*

$$\sum_{i=1}^m a_i \geq \sum_{j=1}^n b_j, \quad a_i \geq 0, b_j \geq 0, \quad i = 1, \dots, m, j = 1, \dots, n$$

gelten.

**Satz 5.2** *Seien  $c_{ij} \geq 0$  für alle  $i, j$  und das Problem (5.1) lösbar. Dann gibt es eine optimale Lösung  $\bar{x}$  von (5.1), in der*

$$\sum_{i=1}^m \bar{x}_{ij} = b_j, \quad j = 1, \dots, n$$

gilt.

**Folgerung 5.3** *Wenn o. E. d. A.  $\sum_{i=1}^m x_{ij} = b_j, j = 1, \dots, n$  gefordert wird, so kann die Aufgabe durch Einführung von Schlupfvariablen  $x_{i,n+1}$  in eine lineare Optimierungsaufgabe mit Gleichungsnebenbedingungen überführt werden. Für die rechte Seite  $b_{n+1}$  ist dann*

$$b_{n+1} = \sum_{i=1}^m a_i - \sum_{j=1}^n b_j$$

zu fordern.

Auf diese Weise wird ein fiktiver Verbraucher  $B_{n+1}$  mit dem Bedarf  $b_{n+1}$  eingeführt. Die Transportkosten zwischen  $A_i$  und  $B_{n+1}$  betragen:  $c_{i,n+1} = 0, i = 1, \dots, m$ . Setze nun:  $k = n + 1$  für die Anzahl der Bedarfsorte und  $y_{ij}, i = 1, \dots, m, j = 1, \dots, k$  für die transportierten Mengen vom Angebotsort  $i$  zum Bedarfsort  $j$ . Damit erzeugt man die Normalform

des Transportproblems, das sogenannte Klassische Transportproblem (KTP):

$$\begin{aligned} \sum_{i=1}^m \sum_{j=1}^k c_{ij} y_{ij} &\longrightarrow \min \\ \sum_{j=1}^k y_{ij} &= a_i \quad \text{für alle } i = 1, 2, \dots, m \\ \sum_{i=1}^m y_{ij} &= b_j \quad \text{für alle } j = 1, 2, \dots, k \\ x_{ij} &\geq 0 \quad \forall i, j \end{aligned}$$

Im Folgenden wird  $y_{ij} = x_{ij}$  und  $n = k$  gesetzt. Das Modell kann somit wie folgt dargestellt werden:

$$\begin{aligned} \sum_{i=1}^m \sum_{j=1}^n c_{ij} x_{ij} &\longrightarrow \min \\ \sum_{j=1}^n x_{ij} &= a_i \quad \text{für alle } i = 1, 2, \dots, m \\ \sum_{i=1}^m x_{ij} &= b_j \quad \text{für alle } j = 1, 2, \dots, n \\ x_{ij} &\geq 0 \quad \forall i, j \end{aligned} \tag{5.2}$$

Eine zulässige Lösung des KTP wird Transportplan genannt. Die Nebenbedingungen beschreiben den Abtransport der Vorratsmengen und die Anlieferung der Bedarfsmengen.

**Satz 5.4** *Das klassische Transportproblem (5.2) ist genau dann lösbar, wenn*

$$\sum_{i=1}^m a_i = \sum_{j=1}^n b_j \quad a_i \geq 0, b_j \geq 0, \quad i = 1, \dots, m, j = 1, \dots, n$$

*gilt.*

**Satz 5.5** *Wenn im klassischen Transportproblem (5.2) alle Zahlen  $a_i, i = 1, \dots, m$  und  $b_j, j = 1, \dots, n$  ganzzahlig sind und das Problem lösbar ist, so gibt es eine optimale Lösung mit ganzzahligen Komponenten.*

Damit kann also eine eventuelle Ganzzahligkeitsforderung an die Lösung des Transportproblems vernachlässigt werden.

**Definition 5.6** *Wenn im klassischen Transportproblem alle Zahlen  $a_i = 1, i = 1, \dots, m$  und  $b_j = 1, j = 1, \dots, n$  sind, so nennt man das entstehende Problem auch Lineares Zuordnungsproblem.*

Das lineare Zuordnungsproblem ist lösbar genau dann, wenn  $m = n$  ist. Im weiteren werden wir, insbesondere bei der Lösung des klassischen Transportproblems, auf die Tabellenschreibweise der Daten zurückgreifen:

	Bedarfsort				Angebot
	1	2	...	$n$	
Angebotsort 1	$c_{11}$	$c_{12}$	...	$c_{1n}$	$a_1$
Angebotsort 2	$c_{21}$	$c_{22}$	...	$c_{2n}$	$a_2$
$\vdots$	$\vdots$	$\vdots$		$\vdots$	$\vdots$
Angebotsort $m$	$c_{m1}$	$c_{m2}$	...	$c_{mn}$	$a_m$
	$b_1$	$b_2$	...	$b_n$	

Die Matrix  $C$  ist die Matrix der Transportkosten  $C = (c_{ij})_{i=1, j=1}^{m, n}$ .

### 5.3. Konstruktion einer Startbasislösung

Die Konstruktion einer Startbasislösung kann mit verschiedenen heuristischen Methoden erfolgen. Beschrieben sei nur die Methode des minimalen Elementes.

Algorithmus:

Schritt 1: Alle Zeilen und Spalten der Matrix  $C$  sind ungestrichen.

Schritt 2: Bestimme ein Indexpaar  $(k, l)$  mit minimalem Wert  $c_{kl}$  in allen ungestrichenen Zeilen und Spalten der Matrix  $C$ .

Setze  $x_{kl} = \min\{a_k, b_l\}$ ,  $a_k := a_k - x_{kl}$ ,  $b_l := b_l - x_{kl}$ .

Streiche die Zeile  $k$ , falls  $a_k = 0$  und Zeile  $k$  nicht die letzte ungestrichene Zeile ist bei noch mehr als einer ungestrichenen Spalte. Sonst streiche die Spalte  $l$ .

Falls noch nicht  $m + n - 1$  Variable  $x_{kl}$  fixiert wurden, wiederhole Schritt 2.

Die mit diesem Algorithmus mit Werten belegten  $m + n - 1$  Variable sind die Basisvariablen. Die anderen Variable sind Nichtbasisvariable und erhalten den Wert null.

**Satz 5.7** *Die Methode des minimalen Elementes konstruiert eine erste zulässige Basislösung für das klassische Transportproblem. Dabei werden  $m + n - 1$  Variable mit nichtnegativen Werten belegt.*

Um den Rang der Koeffizientenmatrix zu untersuchen, schreiben wir die Variablen in der Reihenfolge:  $x_{11}, x_{12}, \dots, x_{1n}, x_{21}, x_{22}, \dots, x_{2n}, \dots, x_{m1}, x_{m2}, \dots, x_{mn}$ .

Dann ergibt sich eine Koeffizientenmatrix mit der folgenden Struktur, bei der die Spalten den Variablen in dieser Reihenfolge entsprechen:

$$\left( \begin{array}{cccc|cccc|cc|cccc} 1 & 1 & \dots & 1 & 0 & 0 & \dots & 0 & \dots & \dots & 0 & 0 & \dots & 0 \\ 0 & 0 & \dots & 0 & 1 & 1 & \dots & 1 & \dots & \dots & 0 & 0 & \dots & 0 \\ \vdots & \vdots & & \vdots & \vdots & \vdots & & \vdots & & & \vdots & \vdots & & \vdots \\ 0 & 0 & \dots & 0 & 0 & 0 & \dots & 0 & \dots & \dots & 1 & 1 & \dots & 1 \\ \hline 1 & 0 & \dots & 0 & 1 & 0 & \dots & 0 & \dots & \dots & 1 & 0 & \dots & 0 \\ 0 & 1 & \dots & 0 & 0 & 1 & \dots & 0 & \dots & \dots & 0 & 1 & \dots & 0 \\ \vdots & \vdots & & \vdots & \vdots & \vdots & & \vdots & & & \vdots & \vdots & & \vdots \\ 0 & 0 & \dots & 1 & 0 & 0 & \dots & 1 & \dots & \dots & 0 & 0 & \dots & 1 \end{array} \right)$$

Dabei entsprechen die Zeilen im oberen Teil den Nebenbedingungen

$$\sum_{j=1}^n x_{ij} = a_i \text{ für alle } i = 1, 2, \dots, m$$

und die Zeilen im unteren Teil den Nebenbedingungen

$$\sum_{i=1}^m x_{ij} = b_j \text{ für alle } j = 1, 2, \dots, n.$$

Da die Summe der Zeilen im oberen Teil gleich der Summe der Zeilen im unteren Teil ist, ist der Rang kleiner als  $m + n$ .

**Satz 5.8** Die Koeffizientenmatrix des Transportproblems hat den Rang  $m + n - 1$ .

Es gibt zahlreiche weitere Methoden zur Konstruktion erster zulässiger Basislösungen. Das sind unter anderem: Nord-West-Eckenregel, Methode der minimalen Spaltensummen, Vogelsche Approximationsmethode.

#### 5.4. Das duale Transportproblem

Das duale klassische Transportproblem (DKTP) ist wie folgt definiert:

$$\begin{aligned} \sum_{i=1}^m a_i u_i + \sum_{j=1}^n b_j v_j &\rightarrow \max \\ u_i + v_j &\leq c_{ij} \text{ für alle } i = 1, 2, \dots, n, \quad j = 1, 2, \dots, m. \end{aligned} \quad (5.3)$$

Damit ergibt sich aus der Anwendung der Theorie der dualen linearen Optimierung (Satz 2.6):

**Satz 5.9** Sei  $\bar{x}$  eine zulässige Basislösung der Aufgabe (KTP) und sei  $I$  die Indermenge der Basisvariablen von  $\bar{x}$ . Sei weiter  $\bar{u}, \bar{v}$  eine Lösung des linearen Gleichungssystems

$$u_i + v_j = c_{ij} \text{ für alle } (i, j) \in I. \quad (5.4)$$

Wenn dann die Optimalitätsindikatoren

$$\Delta_{ij} = c_{ij} - \bar{u}_i - \bar{v}_j \quad (5.5)$$

für alle  $i = 1, 2, \dots, n, \quad j = 1, 2, \dots, m$  nichtnegativ sind, so ist  $\bar{x}$  eine optimale Lösung des klassischen Transportproblems (KTP).

Das lineare Gleichungssystem (5.4) besteht aus  $m+n-1$  Gleichungen,  $m+n$  Variablen und hat eine Koeffizientenmatrix vom Rang  $m+n-1$ . Damit existiert in der allgemeinen Lösung des Gleichungssystems genau ein Parameter. Falls dieser fixiert ist, so ist die Lösung eindeutig. Wir fixieren deshalb  $u_1 = 0$  und bestimmen die nun eindeutige Lösung von (5.4) gleich in der Matrix  $C$ .

Danach wird die Matrix der Optimalitätsindikatoren  $\Delta$  nach (5.5) aufgestellt.

#### 5.5. Verbesserungsschritt

Sei einer der Optimalitätsindikatoren  $\Delta_{i_0 j_0} < 0$ , es kann also die Optimalität der vorliegenden Basislösung nicht nachgewiesen werden. Die Variable  $x_{i_0 j_0}$  soll einen positiven Wert  $\theta$  erhalten. Damit die Gleichungsnebenbedingungen des Transportproblems (5.2) auch weiterhin erfüllt sind, muss es eine andere Variable  $x_{i_0 j_1}$  in der  $i_0$ -ten Zeile geben, die um  $\theta$  verringert wird, geben. Aus dem gleichen Grund muss dann in der  $j_1$ -ten Spalte eine Variable  $x_{i_1 j_1}$  um  $\theta$  vergrößert werden und so weiter. Unter Verwendung von Eigenschaften von Bäumen in Graphen kann gezeigt werden, dass dieser Prozeß letztendlich einen eindeutig bestimmten „Kreis (Austauschzyklus)“ von Elementen der Matrix  $X$  erzeugt, deren Elemente abwechselnd um  $\theta$  vergrößert und verkleinert werden. Die Berechnung dieses Kreises erfolgt mit dem folgenden Algorithmus:

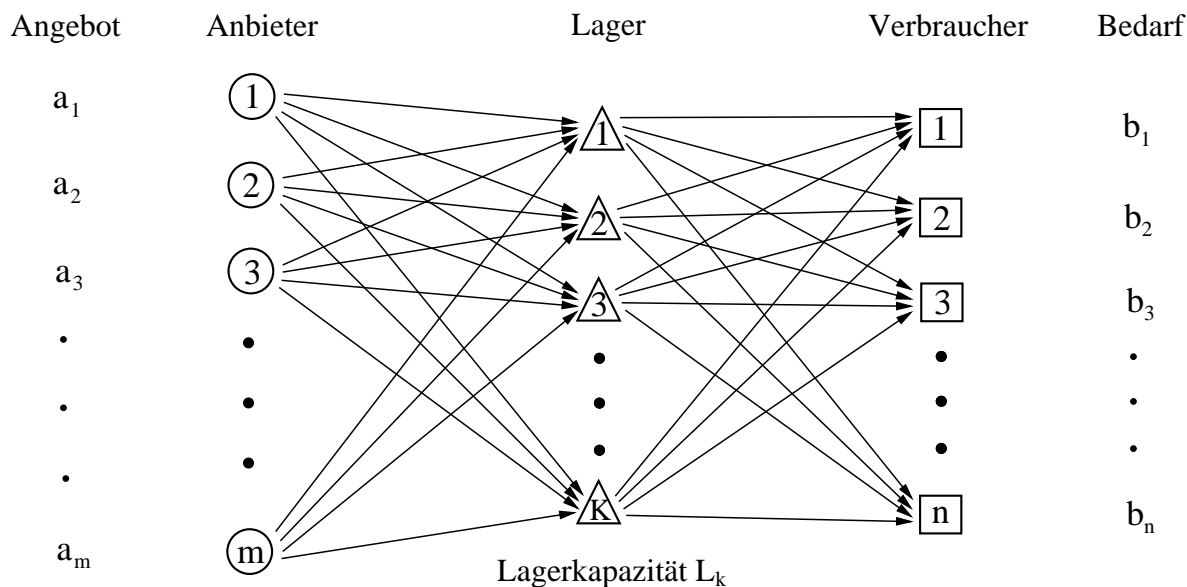


ABBILDUNG 5.2. Graphische Darstellung des zweistufigen Transportproblems

Algorithmus zur Konstruktion des Kreises:

Schritt 1: Wähle ein Element  $\Delta_{i_0j_0}$  mit  $\Delta_{i_0j_0} < 0$  aus. Markiere  $x_{i_0j_0}$  sowie alle Basisvariablen in der aktuellen Lösung.

Schritt 2: Streiche Zeilen bzw. Spalten in der Matrix  $X$ , in denen nur ein markiertes Element existiert, solange dies möglich ist. Übrig bleibt der Kreis.

Der maximale Wert für  $\theta$  ist gleich dem minimalen Wert einer Variable, die um  $\theta$  verkleinert wird. Um diesen Wert zu ermitteln markiert man die Kreiselemente abwechselnd mit „+“ und „-“, wobei  $x_{i_0j_0}$  „+“ erhält. Dann ist

$$\theta = \min\{x_{ij} : x_{ij} \text{ wurde mit „-“ markiert}\}.$$

Die neue zulässige Basislösung ergibt sich nun als

$$x_{ij} = \begin{cases} x_{ij} & ; \text{ falls } x_{ij} \text{ nicht Element des Kreises ist,} \\ x_{ij} + \theta & ; \text{ falls } x_{ij} \text{ mit „+“ markiert wurde,} \\ x_{ij} - \theta & ; \text{ falls } x_{ij} \text{ mit „-“ markiert wurde.} \end{cases}$$

Um die Optimalität dieser Lösung zu überprüfen, führe den Optimalitätstest wiederum durch. Dabei kann die Lösung des linearen Gleichungssystems (5.4) in der Matrix der Optimalitätsindikatoren  $\Delta$  der letzten Iteration berechnet werden.

## 5.6. Zweistufige Transportprobleme

### 5.6.1. Charakterisierung des Problems.

Im Gegensatz zum klassischen Transportproblem erfolgen hier keine Direkttransporte vom Anbieter zum Verbraucher, sondern alle Transporte werden über Zwischenlager realisiert. Damit ergibt sich ein Transportgraph, wie im Bild 5.2 illustriert.

Neben den Angebotsmengen  $a_i$  in den Angebotsorten  $i = 1, \dots, m$  und den Bedarfsmengen  $b_j$  in den Verbrauchsorten  $j = 1, \dots, n$  sind noch  $K$  Lager mit den Lagerkapazitäten  $L_k$ ,  $k = 1, \dots, K$  gegeben. Die Kosten für den Transport einer Mengeneinheit vom Anbieter  $i$  zum  $k$ -ten Lager seien  $c_{ik}$  und die entsprechenden Kosten für den Transport einer

Mengeneinheit vom Lager  $k$  zum Verbraucher  $j$  werden mit  $d_{kj}$  bezeichnet. Es sollen  $x_{ik}$  Mengeneinheiten vom Anbieter  $i$  zum  $k$ -ten Lager und  $y_{kj}$  Mengeneinheiten vom Lager  $k$  zum Verbraucher  $j$  transportiert werden.

Damit ist folgende Aufgabe zu lösen:

Gesucht sind die Transportmengen  $x_{ij}$  so, dass

- (1) die gesamten Transportkosten minimal sind,
- (2) der Bedarf von allen Bedarfsorten exakt gedeckt ist,
- (3) die Angebotsmenge aus allen Angebotsorten vollständig abtransportiert ist,
- (4) die Lagerkapazitäten eingehalten und alle Lager geleert werden.

Das zweistufige Transportproblem (ZTP):

$$\sum_{i=1}^m \sum_{k=1}^K c_{ik} x_{ik} + \sum_{k=1}^K \sum_{j=1}^n d_{kj} y_{kj} \rightarrow \min \quad (\text{kostenminimaler Transport})$$

$$\sum_{k=1}^K y_{kj} = b_j, \quad j = 1, 2, \dots, n \quad (\text{Bedarf})$$

$$\sum_{k=1}^K x_{ik} = a_i, \quad i = 1, 2, \dots, m \quad (\text{Angebot})$$

$$\sum_{i=1}^m x_{ik} = \sum_{j=1}^n y_{kj}, \quad k = 1, 2, \dots, K \quad (\text{Leeren der Lager})$$

$$\sum_{j=1}^n y_{kj} \leq L_k, \quad k = 1, 2, \dots, K \quad (\text{Einhaltung der Lagerkapazität})$$

$$x_{ik}, y_{kj} \geq 0, \quad i = 1, 2, \dots, m, \quad j = 1, 2, \dots, n, \quad k = 1, 2, \dots, K$$

Hier ist wie im klassischen Transportproblem angenommen worden, dass

$$\sum_{i=1}^m a_i = \sum_{j=1}^n b_j$$

gilt.

**Satz 5.10** *Das zweistufige Transportproblem ist lösbar genau dann, wenn*

$$\sum_{i=1}^m a_i = \sum_{j=1}^n b_j \leq \sum_{k=1}^K L_k$$

*ist und  $a_i, b_j, L_k \geq 0$  sind für  $i = 1, \dots, m, j = 1, \dots, n, k = 1, \dots, K$ .*

Wenn die Lagerkapazitäten alle hinreichend groß sind ( $L_k \geq \sum_{i=1}^m a_i$ ), so läßt sich das zweistufige Transportproblem auf ein einstufiges reduzieren: Die Kostenkoeffizienten sind dann  $c_{ij} = \min\{c_{ik} + d_{kj} : k = 1, \dots, K\}$ . Ist  $\sum_{i=1}^m a_i = \sum_{k=1}^K L_k$ , so läßt sich das zweistufige Transportproblem entkoppeln und es sind nur zwei klassische Transportprobleme zu lösen.

Wenn im zweistufigen Transportproblem anstelle der Gleichheiten bei den Angebotsmengen und/oder den Bedarfsmengen Ungleichungen stehen, so läßt sich das Problem durch Einführung von Schlupfvariablen in obiges Modell überführen, wenn geklärt ist, an welchen Orten die überschüssigen Mengen verbleiben sollen und die Zielfunktionskoeffizienten nichtnegativ sind.

### 5.6.2. Transformation auf das klassische Transportproblem.

Das zweistufige Transportproblem kann durch Einführung weiterer Variablen in ein klassisches Transportproblem mit verbotenen Wegen transformiert werden. Die neuen Variablen  $x_{kk}$  stehen für die ungenutzte Lagerkapazität:

$$\sum_{i=1}^m \sum_{k=1}^K c_{ik} x_{ik} + \sum_{k=1}^K \sum_{j=1}^n d_{kj} y_{kj} \rightarrow \min \quad (\text{kostenminimaler Transport})$$

$$\sum_{k=1}^K y_{kj} = b_j, \quad j = 1, 2, \dots, n \quad (\text{Bedarf})$$

$$\sum_{k=1}^K x_{ik} = a_i, \quad i = 1, 2, \dots, m \quad (\text{Angebot})$$

$$\sum_{i=1}^m x_{ik} + x_{kk} = L_k, \quad k = 1, 2, \dots, K \quad (\text{Einhaltung der Lagerkapazität})$$

$$\sum_{j=1}^n y_{kj} + x_{kk} = L_k, \quad k = 1, 2, \dots, K \quad (\text{Einhaltung der Lagerkapazität})$$

$$x_{ik}, y_{kj}, x_{kk} \geq 0, \quad i = 1, 2, \dots, m, \quad j = 1, 2, \dots, n, \quad k = 1, 2, \dots, K$$

Damit kann das Problem mit den oben angeführten Algorithmen gelöst werden. Die Kostenmatrix hat die folgende Gestalt:

$$\left( \begin{array}{cccc|cccc} c_{11} & c_{12} & \dots & c_{1K} & \infty & \infty & \dots & \infty \\ c_{21} & c_{22} & \dots & c_{2K} & \infty & \infty & \dots & \infty \\ \vdots & \vdots & \dots & \vdots & \vdots & \vdots & \dots & \vdots \\ c_{m1} & c_{m2} & \dots & c_{mK} & \infty & \infty & \dots & \infty \\ \hline 0 & \infty & \dots & \infty & d_{11} & d_{12} & \dots & d_{1n} \\ \infty & 0 & \dots & \infty & d_{21} & d_{22} & \dots & d_{2n} \\ \vdots & \vdots & \dots & \vdots & \vdots & \vdots & \dots & \vdots \\ \infty & \infty & \dots & 0 & d_{K1} & d_{K2} & \dots & d_{Kn} \end{array} \right)$$

Zur Berechnung einer zulässigen Startbasislösung kann die Methode des minimalen Elementes oder eine andere Heuristik verwendet werden. Damit dabei keine Variable mit unendlich großem Zielfunktionskoeffizienten belegt werden muss, sind sinnvoller Weise erst die Elemente im linken oberen Block, dann die Elemente im linken unteren Block und zum Schluß die Elemente im rechten unteren Block zu belegen.



## Diskrete Optimierung

### 6.1. Modellierung diskreter Optimierungsaufgaben

Die Modellierung diskreter Optimierungsaufgaben soll am Beispiel des Rundreiseproblems erläutert werden.

Das Rundreiseproblem hat die folgende Interpretation:

Ein Handelsreisender soll nacheinander  $n - 1$  Städte besuchen und erst dann in seine Ausgangsstadt Nr. 1 zurückkehren. Er besucht natürlich jede Stadt genau einmal. Wie lang ist sein Reiseweg mindestens?

Bezeichne  $c_{ij}$  die Entfernung von Stadt  $i$  und Stadt  $j$ . Die Aufgabenstellung soll anhand der Bilder 6.1, 6.2 und 6.3 erläutert werden. Gegeben sind Orte und ihre Verbindungen wie im Bild 6.1. Der Handelsreisende bewegt sich entlang der Linien zwischen den Orten. Bild 6.2 gibt eine der möglichen Fahrtrouten des Handelsreisenden an. Bild 6.3 zeigt keine mögliche Fahrtroute des Handelsreisenden, da er dann beim Start im Ort 1 keinen der Orte 3, 5, 6 und 7 erreicht. Es ist unter allen möglichen Fahrtrouten eine solche mit minimalen Gesamtkosten (Länge, Fahrtzeiten, Reisekosten, etc.) zu suchen.

Das Rundreiseproblem, auch Traveling Salesman Problem genannt, erhielt seinen Namen durch die obige Aufgabenstellung.

Gegeben sind  $n$  Orte.

Gesucht: ein Weg minimaler Länge, der (ausgehend von einem Ort) durch alle Orte genau einmal führt und am Ende den Ausgangspunkt wieder erreicht.

Dieses Problem läßt sich mit unterschiedlichen Mitteln modellieren. Zwei solche Mittel sollen im weiteren vorgestellt werden.

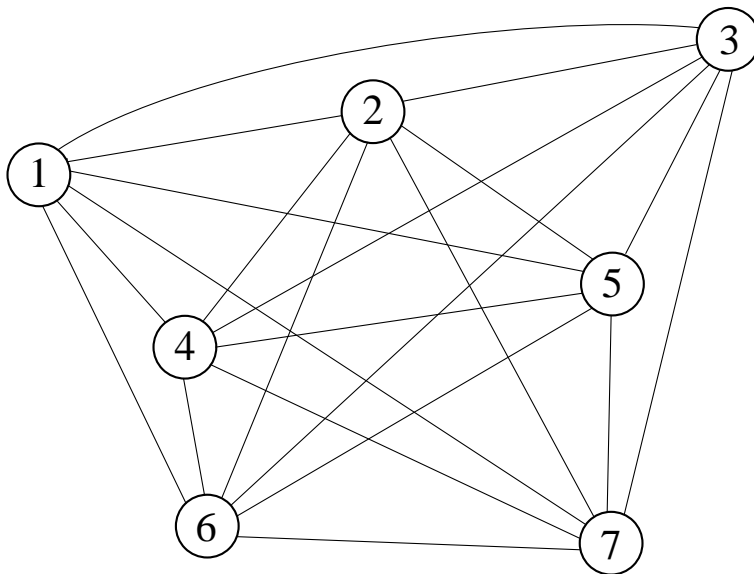


ABBILDUNG 6.1. Aufgabenstellung Rundreiseproblem

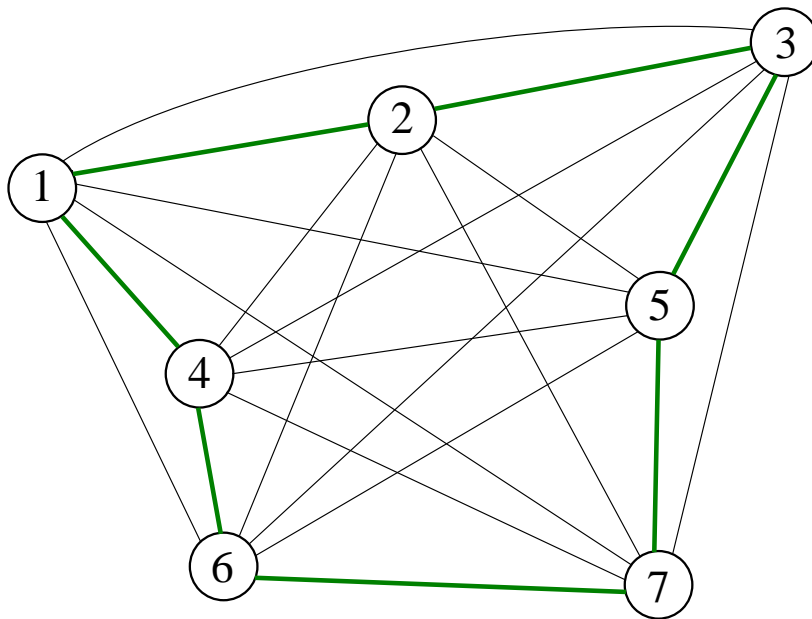


ABBILDUNG 6.2. Mögliche Rundreise

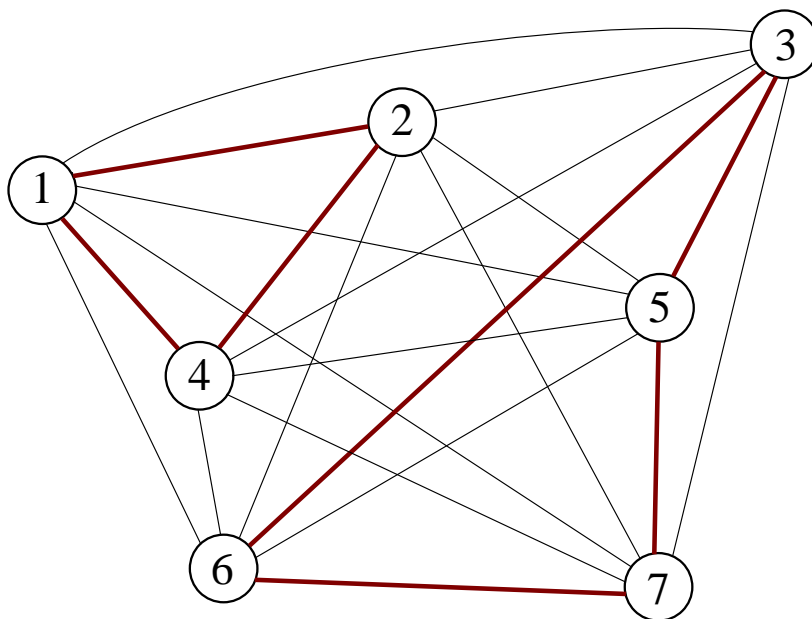


ABBILDUNG 6.3. Keine Rundreise

(1) **Modellierung mit ganzzahligen Variablen:**

Wir führen Variable  $x_{ij}$  ein mit:

$$x_{ij} = \begin{cases} 1, & \text{falls der Ort } j \text{ nach dem Ort } i \text{ besucht wird} \\ 0, & \text{sonst} \end{cases}$$

Mit diesen Variablen ergibt sich das folgende Problem:

$$\sum_{i=1}^n \sum_{j=1}^n c_{ij} x_{ij} \longrightarrow \min \quad (\text{Die Gesamtlänge der Tour soll minimiert werden})$$

$$\sum_{i=1}^n x_{ij} = 1; \quad \forall j = 1, \dots, n \quad (\text{Jeder Ort soll erreicht werden})$$

$$\sum_{j=1}^n x_{ij} = 1; \quad \forall i = 1, \dots, n \quad (\text{Jeder Ort soll verlassen werden})$$

$$x_{ij} \in \{0, 1\} \quad \forall i, j = 1, \dots, n$$

$$\sum_{i \in U} \sum_{j \notin U} x_{ij} \geq 1 \quad \forall U \subseteq \{1, \dots, n\} \text{ mit } 2 \leq |U| \leq n - 2$$

(Kurzzyklenverbote)

Die Notwendigkeit der letzten Ungleichungen sieht man in Bild 6.3 mit der Menge  $U = \{3, 5, 6, 7\}$ .

Problematisch ist hier die Anzahl der Nebenbedingungen, die von der Größenordnung  $O(n!)$  ist.

## (2) Modellierung mit Permutation

Eine Permutation von  $\{1, 2, \dots, n\}$  ist eine bijektive Abbildung dieser Menge auf sich selbst:  $\pi : \{1 \dots n\} \longrightarrow \{1 \dots n\}$

Um Permutationen zur Modellierung des Rundreiseproblems anzuwenden, bezeichne  $\pi(i) = j$ , dass unmittelbar nach der Stadt  $i$  die Stadt  $j$  besucht wird.

Man kann eine Permutation in einer Tabelle schreiben, in deren zweiter Zeile die Funktionswerte von  $\pi(i)$  für die Argumente in der ersten Zeile stehen. Dann ergibt sich für die in der Abbildung 6.2 dargestellte Rundreise die folgende Tabelle:

i	1	2	3	4	5	6	7
$\pi(i)$	2	3	5	1	7	4	6

Zu erkennen ist, dass für eine Permutation jede der Zahlen in  $\{1, 2, \dots, n\}$  genau einmal in der zweiten Zeile der Tabelle steht. Mathematisch kann man das wie folgt ausdrücken:

$$\forall i \in \{1, \dots, n\} \exists j \in \{1, \dots, n\} \text{ mit } \pi(i) = j$$

und

$$\forall j \in \{1, \dots, n\} \exists i \in \{1, \dots, n\} \text{ mit } \pi(i) = j.$$

Unter Verwendung der Permutationen ergibt sich das folgende Modell des Rundreiseproblems:

$$\sum_{i=1}^n c_{i\pi(i)} \rightarrow \min$$

unter den Bedingungen:

$\pi$  ist Permutation

$$\pi(1) \neq 1 \quad (\text{der Nachfolger von 1 ist nicht 1})$$

$$\pi(\pi(1)) = \pi^2(1) \neq 1, \dots, \pi^{n-1}(1) \neq 1$$

$$\pi^n(1) = 1 \quad (\text{der } n\text{-te Nachfolger ist 1})$$

Andere Modellierungszugänge für diskrete Optimierungsaufgaben verwenden graphentheoretische Mittel.

- Bemerkung 6.1** (1) *Es gibt eine große Vielfalt von Problemen der diskreten Optimierung. Die besten Modellierungszugänge sind problemabhängig.*  
 (2) *In der diskreten Optimierung gibt es eine eigenständige Theorie, die sich gerade in den letzten Jahren stürmisch entwickelt hat.*  
 (3) *Sehr viele Anwendungsaufgaben verwenden wesentlich die diskrete Optimierung.*

## 6.2. Exakte Lösungsalgorithmen

### 6.2.1. Schnittebenenalgorithmen.

Hier soll nur die grundlegende Idee der Schnittebenenalgorithmen angegeben werden. Betrachtet werde eine ganzzahlige lineare Optimierungsaufgabe

$$\min\{\langle c, x \rangle : Ax \leq b, x \geq 0, \text{ ganzzahlig}\}. \quad (6.1)$$

Diesem Problem wird eine „einfach“ lösbare Optimierungsaufgabe (eine Relaxation) zugeordnet. Der Einfachheit halber entstehe diese Aufgabe durch Weglassen der Ganzzahligkeitsbedingungen. Die Relaxation ist also die lineare Optimierungsaufgabe

$$\min\{\langle c, x \rangle : Ax \leq b, x \geq 0\}. \quad (6.2)$$

**Schnittebenenalgorithmus:** (vgl. Abbildung 6.4)

Schritt 1: Löse die Aufgabe (6.2). Eine optimale Lösung sei  $\bar{x}$ .

Schritt 2: Wenn  $\bar{x}$  zulässig für die Aufgabe (6.1) ist, so ist sie optimal. Der Algorithmus bricht ab.

Schritt 3: Wenn  $\bar{x}$  nicht zulässig für die Aufgabe (6.1) ist, so bestimmt man eine neue Nebenbedingung

$$\langle a, x \rangle \leq b_1$$

mit folgenden Eigenschaften:

a)  $\langle a, \bar{x} \rangle > b_1$

b) für alle zulässigen Lösungen  $\hat{x}$  der Aufgabe (6.1) gilt die Ungleichung  $\langle a, \hat{x} \rangle \leq b_1$ .

Füge diese Ungleichung zu dem System  $Ax \leq b$  hinzu (das neue System wird wieder mit  $Ax \leq b$  bezeichnet) und gehe zu Schritt 1.

Problematisch ist natürlich der Schritt 3. Ansatzpunkte, um eine solche Ungleichung zu finden, stehen in der Literatur unter den Stichworten „Gomory-Schnitt“, „Chvatal-Gomory-Schnitt“, „valid cut“. Eng verbunden damit sind Untersuchungen zur „Polyedertheorie“. Hier soll nicht weiter darauf eingegangen, sondern nur auf die Literatur verwiesen werden (z.B. G.L. Nemhauser, L.A. Wolsey: Integer and Combinatorial Optimization, Wiley, 1988).

### 6.2.2. Branch-and-bound-Algorithmus.

Der Vollenumerationsalgorithmus zur Lösung des linearen 0-1 Optimierungsproblems

$$\min\{\langle c, x \rangle : Ax \leq b, x_j \in \{0, 1\}, j = 1, \dots, n\} \quad (6.3)$$

berechnet eine optimale Lösung nach folgendem Schema: Es werden alle Vektoren  $x \in \mathbb{R}^n$  mit  $x_i \in \{0, 1\}$ ,  $i = 1, \dots, n$  konstruiert, diese auf Zulässigkeit überprüft und danach eine solche mit einem besten Zielfunktionswert ausgewählt. Zur Konstruktion aller Vektoren  $x \in \mathbb{R}^n$  mit  $x_i \in \{0, 1\}$ ,  $i = 1, \dots, n$  kann man nach dem Schema in Abbildung 6.5 vorgehen, welches diese Punkte in einem Verzweigungsbaum sortiert. Dabei entsprechen

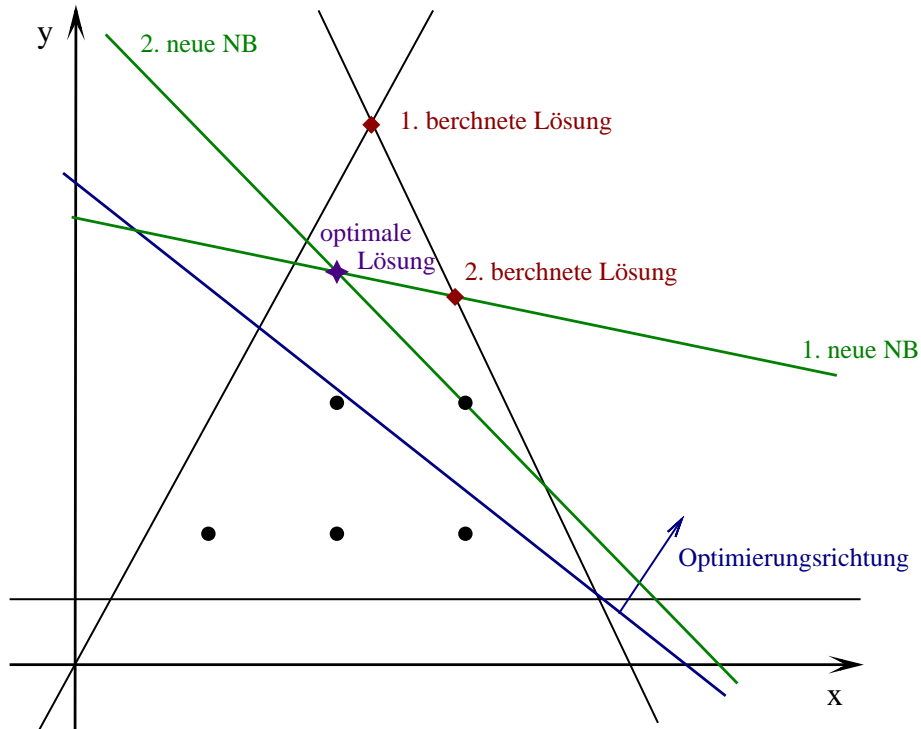


ABBILDUNG 6.4. Der Schnittebenenalgorithmus

die einzelnen Äste des Baumes den Punkten  $x$ , wie es beispielhaft in der Abbildung angegeben ist. Die Punkte im Baum werden Knoten genannt. Jeder Knoten ist durch die auf dem Weg von der Wurzel (Knoten ohne Zahl, der keiner Zuweisung eines Wertes zu einer Variablen entspricht) zu diesem Knoten fixierten Wertzuweisungen zu einigen Variablen eindeutig bestimmt. Diese Knoten sollen dann auch durch diese Wertzuweisungen als  $(I, J)$  bezeichnet werden mit:

$$I = \{i : x_i = 0\}, J = \{i : x_i = 1\},$$

I - Indexmenge der auf 0 fixierten Variablen,  
 J - Indexmenge der auf 1 fixierten Variablen.

Die Vollenumeration ist aufgrund ihres gewaltigen Rechenaufwandes allerdings nicht zur Lösung des Problems (6.3) geeignet. Eine erste Idee, diesen Zugang in einen geeigneten Lösungsalgorithmus zumindest für kleine Aufgaben zu transformieren besteht darin, dass Äste des Verzweigungsbaumes möglichst frühzeitig abgeschnitten werden und somit möglichst frühzeitig erkannt wird, dass aus Knoten  $(I, J)$  nicht verzweigt werden muss. Zur Realisierung dieser Idee dient die Schrankenfunktion  $\varphi(I, J)$ , die den Knoten den optimalen Zielfunktionswert von Schrankenaufgaben zuweist. Diese Schrankenaufgaben

$$\min\{\langle c, x \rangle : Ax \leq b, x_i = 0, i \in I, x_i = 1, i \in J, 0 \leq x_i \leq 1, i \notin I \cup J\} \quad (6.4)$$

sind lineare Relaxationen der in den Knoten zu lösenden Aufgaben

$$\min\{\langle c, x \rangle : Ax \leq b, x_i = 0, i \in I, x_i = 1, i \in J, x_i \in \{0, 1\}, i \notin I \cup J\}. \quad (6.5)$$

Da die zulässigen Bereiche der Schrankenaufgaben durch weitere Fixierung von Variablen immer weiter eingeschränkt werden, gilt:

$$\varphi(I, J) \leq \varphi(I', J') \text{ falls } I \subseteq I', J \subseteq J' \text{ und } I' \cap J' = \emptyset.$$

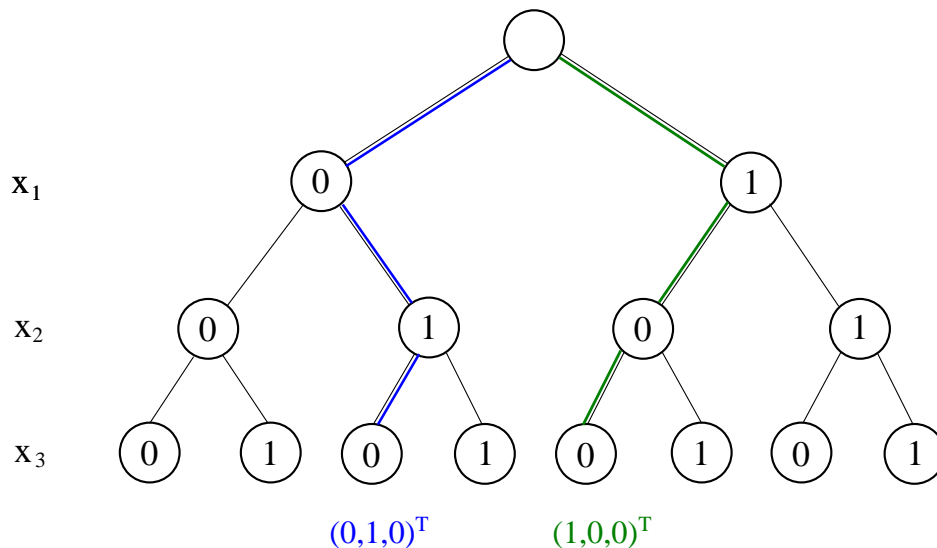


ABBILDUNG 6.5. Verzweigungsbaum, Vollenumeration

Insbesondere gilt diese Ungleichung auch, wenn  $I' \cup J' = \{1, 2, \dots, n\}$  ist. Damit erhalten wir den folgenden Satz:

**Satz 6.2** Sei  $\bar{x}$  eine zulässige Lösung des Problems (6.3) und

$$\varphi(I, J) \geq \langle c, \bar{x} \rangle.$$

Dann ist auch

$$\varphi(I', J') \geq \langle c, \bar{x} \rangle \text{ für alle } I' \subseteq I, J' \subseteq J.$$

Damit muss aus dem Knoten  $(I, J)$  nicht verzweigt werden, die entsprechenden Zweige des Verzweigungsbaumes können gestrichen werden.

Somit ergibt sich die folgende Idee für einen Lösungsalgorithmus für das Problem (6.3).

- Ich beginne mit  $I = \emptyset, J = \emptyset$  und erzeuge und löse nacheinander alle Schranken­aufgaben mit wachsenden Mengen  $I, J$ . (Realisierung durch Verzweigung oder branching)
- Wird einmal (zufällig) eine zulässige Lösung für die Aufgabe (6.3) berechnet, so merke ich mir diese Lösung, wenn sie besser ist als alle schon vorher berechneten zulässigen Lösungen.
- Ist der optimale Zielfunktionswert einer Schrankenaufgabe nicht besser als die bisher beste bekannte zulässige Lösung, so wird diese und alle ihre „Töchter“ nicht weiter betrachtet. (Das ist die Verwendung der Schranken oder bounding)

### Branch-and-bound Algorithmus

Schritt 0 Start mit  $I = \emptyset, J = \emptyset, M = \{(\emptyset, \emptyset)\}$  ( $M$  enthält alle noch zu lösenden Aufgaben (6.4)).

Es sei  $f^* = \langle c, \bar{x} \rangle$  für eine bekannte zulässige Lösung für die Aufgabe (6.3) bzw.  $f^* = \infty$  ( $f^*$  ist der aktuell beste Zielfunktionswert)

Schritt 1 Wenn  $M = \emptyset$ , stopp, Ausgabe der Lösung.

Schritt 2 Wähle  $(I^0, J^0) \in M$ , setze  $M := M \setminus \{(I^0, J^0)\}$  (Streiche das gewählte Paar in der Menge  $M$ .)

Löse die Aufgabe (6.4) mit  $I = I^0, J = J^0$ .

Schritt 3 Wenn  $\varphi(I^0, J^0) \geq f^*$  ist, gehe zu Schritt 1.

Wenn die berechnete optimale Lösung  $\bar{x}$  zulässig ist für die Aufgabe (6.3), merke die Lösung, setze  $f^* := \min \{f^*, \langle c, \bar{x} \rangle\}$  und gehe zu Schritt 1.

Sonst wähle eine Komponente  $\bar{x}_{j_0} \in (0, 1)$  und setze  $M := M \cup \{(I^0 \cup \{j_0\}, J^0), (I^0, J^0 \cup \{j_0\})\}$  und gehe zu Schritt 1.

### 6.3. Näherungsalgorithmen

Näherungsalgorithmen unterscheidet man in Eröffnungs- und Verbesserungsverfahren. Beide sollen am Beispiel des Rundreiseproblems

$$\sum_{i=1}^n c_{i\pi(i)} \rightarrow \min$$

unter den Bedingungen:

$\pi$  ist Permutation

$$\pi(1) \neq 1 \quad (\text{der Nachfolger von 1 ist nicht 1})$$

$$\pi(\pi(1)) = \pi^2(1) \neq 1, \dots, \pi^{n-1}(1) \neq 1$$

$$\pi^n(1) = 1 \quad (\text{der } n\text{-te Nachfolger ist 1})$$

exemplarisch beschrieben werden.

#### 6.3.1. Eröffnungsverfahren.

Eröffnungsverfahren konstruieren zulässige Lösungen, die möglichst gut sein sollen. Sie sollten verbunden werden mit einer Abschätzung ihrer Güte, die wiederum nach den Prinzipien des schlechtesten Falles (worst-case analysis, die angegebene Güteschranke gilt für alle Aufgaben des untersuchten Problems) oder des wahrscheinlichen Falles (average-case analysis, die angegebene Güteschranke gilt im Mittel für alle Aufgaben des untersuchten Problems, wobei eine passende Wahrscheinlichkeitsverteilung der Daten zugrunde gelegt werden muss) berechnet wird.

Für das Rundreiseproblem werde die folgende Heuristik verwendet:

#### Einfügungsalgorithmus

Schritt 1: Setze  $V = \{1, 2, \dots, n\}$  ( $V$  ist die Menge der noch nicht besuchten Orte)

Schritt 2: Wähle ein  $i \in V$ ,  $V := V \setminus \{i\}$  sowie ein  $j \in V$  mit

$$c_{ij} = \max\{c_{ik} : k \in V\}.$$

Setze  $V := V \setminus \{j\}$ ,  $\pi(i) = j$ ,  $\pi(j) = i$ .

Schritt 3: Wähle ein  $q \in V$  mit

$$c_{pq} = \max_{j \in V} \min_{i \notin V} c_{ij}.$$

( $q$  hat größte Entfernung zu den schon besuchten Orten)

Schritt 4: Wähle ein  $i_0 \notin V$  mit

$$i_0 \in \underset{i}{\text{Argmin}} \{c_{iq} + c_{q\pi(i)} - c_{i\pi(i)}\}$$

(kürzester Umweg bei Einfügung des Ortes  $q$  nach dem Ort  $i_0$ )

Schritt 5: Setze

$$\pi(i) = \begin{cases} \pi(i), & \text{falls } i \notin \{i_0, q\} \\ \pi(i_0), & \text{falls } i = q \\ q, & \text{falls } i = i_0 \end{cases}$$

$V := V \setminus \{q\}$ , gehe zu Schritt 3, falls  $V \neq \emptyset$ .

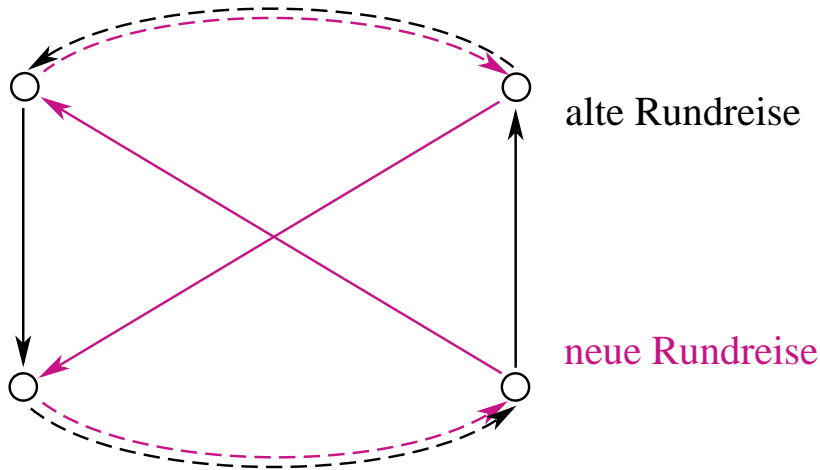


ABBILDUNG 6.6. Rundreisebildung im 2-opt-Verfahren

**Satz 6.3** Sei  $\pi^*$  eine optimale und  $\bar{\pi}$  die mit dem Einsetzungsverfahren berechnete Lösung. Sei weiter

$$c_{ij} + c_{jk} \geq c_{ik} \quad \text{für alle } i, j, k = 1, 2, \dots, n.$$

Dann ist

$$\sum_{i=1}^n c_{i\bar{\pi}(i)} \leq 2 \sum_{i=1}^n c_{i\pi^*(i)}$$

Der Rechenaufwand des Algorithmus ist  $O(n^2)$ .

### 6.3.2. Verbesserungsverfahren.

Verbesserungsverfahren starten mit einer zulässigen Lösung und versuchen, eine bessere zu berechnen. Die meisten dieser Verfahren beruhen auf der Beschreibung von Umgebungen einer zulässigen Lösung. Solche Umgebungen können für das Rundreiseproblem wie folgt beschrieben werden:

- (1) 2-opt: Die Menge der Umgebungen bei diesem Zugang besteht aus allen Rundreisen, die aus der gegebenen Rundreise  $\pi$  durch Ersetzung von genau 2 Nachfolgerbeziehungen durch 2 neue entstehen. Das Prinzip ist in Abbildung 6.6 dargestellt.
- (2) 3-opt: Die Menge der Umgebungen bei diesem Zugang besteht aus allen Rundreisen, die aus der gegebenen Rundreise  $\pi$  durch Ersetzung von 2 oder 3 Nachfolgerbeziehungen durch 2 oder 3 neue entstehen. Das Prinzip ist in Abbildung 6.7 zu erkennen.

Diese Umgebungen können nun zur Konstruktion von Verbesserungsalgorithmen verwendet werden. In diesen Verbesserungsalgorithmen geht man wie folgt vor:

Schritt 1: Start mit einer ersten zulässigen Lösung  $\pi$  und einer entsprechenden Umgebung  $U(\pi)$ .

Schritt 2: Wähle ein  $\bar{\pi} \in U(\pi)$ .

Schritt 3: Ersetze unter gewissen Voraussetzungen  $\pi$  durch  $\bar{\pi}$ . Kehre zu Schritt 2 zurück.

Für die Wahl von  $\bar{\pi} \in U(\pi)$  im Schritt 2 gibt es mehrere Möglichkeiten, u.a.:

- (1) Wahl der besten Lösung  $\bar{\pi}$  in dieser Umgebung.
- (2) Wahl einer besseren Lösung  $\bar{\pi}$  in dieser Umgebung.



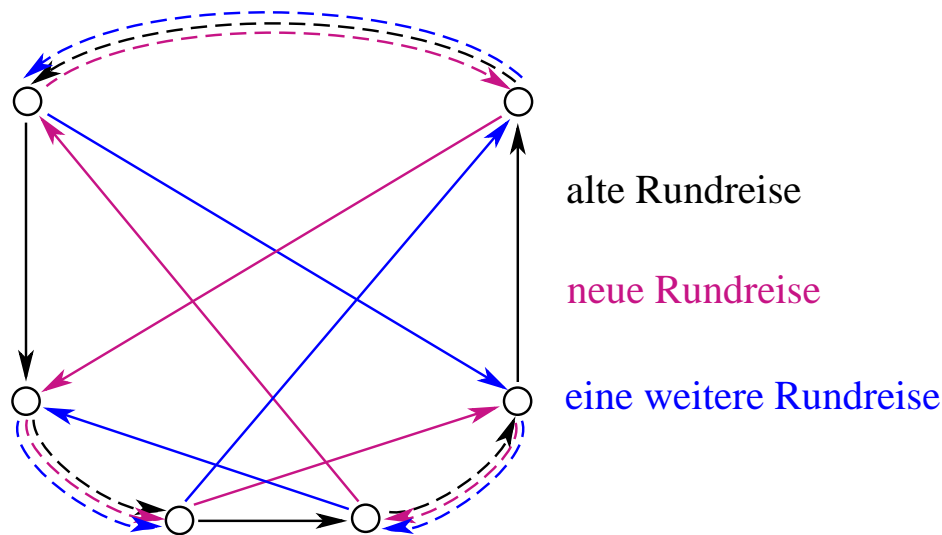


ABBILDUNG 6.7. Rundreisebildung im 3-opt-Verfahren

(3) Wahl einer beliebigen Lösung  $\bar{\pi}$  in dieser Umgebung.

Für die Entscheidung in Schritt 3 des Algorithmus gibt es ebenfalls mehrere Möglichkeiten, u.a.:

- (1) Übergang zu  $\bar{\pi}$ , wenn der Zielfunktionswert von  $\bar{\pi}$  besser als der von  $\pi$  ist.
- (2) Übergang zu  $\bar{\pi}$  auch dann, wenn der Zielfunktionswert von  $\bar{\pi}$  nicht besser als der von  $\pi$  ist. Dann aber nur, wenn der Zielfunktionswert um nicht mehr als einen vorgegebenen Wert schlechter wird, wobei diese Schranke im Laufe der Rechnung immer einschränkender wird.

Mit dieser Herangehensweise kann man zu verschiedenen Algorithmen kommen:

- (1) Methoden des steilsten Abstieges benutzen bei der ersten Entscheidung die erste und bei der zweiten Entscheidung ebenfalls die erste Möglichkeit.
- (2) Lokale Suchmethoden verwenden bei der ersten Entscheidung die erste, zweite oder dritte und bei der zweiten Entscheidung die erste Möglichkeit.
- (3) Monte-Carlo Methoden verwenden bei der ersten Entscheidung die dritte und bei der zweiten Entscheidung die erste Möglichkeit.
- (4) Simulated Annealing verwendet bei der ersten Entscheidung die dritte und bei der zweiten Entscheidung die zweite Möglichkeit.



## Einige Modelle der Logistik

### 7.1. Tourenprobleme

Betrachtet wird zunächst 1 Depot, von dem aus  $n$  Kunden beliefert werden sollen. Kunde  $i$  am Standort  $P_i$  hat einen Bedarf von  $a_i$  Mengeneinheiten, der durch einmalige Anlieferung zu decken ist. Die Frage ist: Wie soll die Belieferung der Kunden mit minimalem Aufwand erfolgen?

Zur Belieferung stehen mehrere (gleichartige) Fahrzeuge zur Verfügung, die alle eine Kapazität von  $b$  Mengeneinheiten besitzen. Gegeben ist auch die Entfernung  $d_{ij}$  zwischen dem Kunden  $i$  am Ort  $P_i$  und dem Kunden  $j$  am Ort  $P_j$  beziehungsweise dem Depot  $j = 0$  am Ort  $P_0$ .

Das Problem soll zunächst modelliert werden. Es sind 2 Fragen zu klären:

- Welche Kunden sollen durch welche Fahrzeuge beliefert werden?  
Einzuhalten ist dabei die Kapazität der Fahrzeuge, d.h. alle Waren aller zu beliefernder Kunden müssen gleichzeitig auf das Fahrzeug passen. Das Ergebnis ist eine Tour.
- In welcher Reihenfolge sollen die Fahrzeuge die Kunden beliefern?  
Jeder Kunde soll genau einmal beliefert werden, die gesamte Fahrstrecke soll minimal sein. Das Ergebnis ist eine Route.

Bei der Bestimmung der Route ist ein Rundreiseproblem zu lösen, welches aus mathematischer Sicht zu den schweren Problemen gehört.

**Ziel:** Bestimmung der Touren und Routen so, dass die insgesamt von allen Fahrzeugen für die Belieferung aller Kunden zurückgelegte Strecke minimal ist.

Mögliche Verallgemeinerungen:

- Betrachtung mehrerer Depots.
- Betrachtung verschiedener Fahrzeuge.
- Mehrere Kapazitätsbeschränkungen der Fahrzeuge: Gewicht, Volumen, etc.
- Zeitfenster.
- Präzedenzbeschränkungen: Reihenfolge der Kunden, pickup and delivery Probleme.
- Anzahl der zur Verfügung stehenden Fahrzeuge ist fixiert oder frei.
- Mögliche Nichtbelieferung von Kunden ist bei Zahlung einer Strafe möglich.
- Einschränkung bei Zuordnung Kunde - Fahrzeug (Milch / Benzin).

Eine der Möglichkeiten zur Modellierung des Problems besteht in der Definition einer Familie:

$$M = \{T_j, j = 1, \dots, J\} \text{ mit } T_j \subset \{1, \dots, n\} \quad (7.1)$$

von Teilmengen der Menge  $\{1, \dots, n\}$  aller Kunden mit

$$\bigcup_{j=1}^J T_j = \{1, \dots, n\}, \quad \sum_{i \in T_j} a_i \leq b, \quad j = 1, \dots, J. \quad (7.2)$$

- Bemerkung 7.1** (1) Letzte Ungleichung fordert, dass die Kapazität der Fahrzeuge eingehalten werden soll.
- (2) Eine einfache (allerdings zu aufwändige) Wahl der Familie  $M$  besteht in der Aufnahme aller möglichen Touren.
- (3) Die Menge  $M$  ist also eine Familie möglicher Touren, so dass jeder Kunde in mindestens einer dieser Touren enthalten ist.

### Tourenproblem

Gegeben sind ein Graph  $G = (V, E)$  mit  $0 \in V$  und einer Kantenbewertung  $d : E \rightarrow \mathbb{R}$ ,  $a_i \geq 0$ ,  $i \in V \setminus \{0\}$ ,  $b > 0$ .

Es seien eine Familie  $M$  gemäß Gleichung (7.1) mit (7.2) bestimmt und die Koeffizienten  $c_j$ ,  $j = 1, \dots, J$ , berechnet. Dabei ist  $c_j$  die Länge der kürzesten Route für die Tour  $j$  und  $a_{ij}$  gibt an, ob der Kunde  $i$  in der Tour  $j$  enthalten ist.

Gesucht ist eine Teilmenge der gewählten Touren in der Menge  $M$ , so dass jeder Kunde in genau einer der gewählten Touren enthalten ist und die Gesamtlänge aller Touren minimal ist. Mit anderen Worten, zu lösen ist:

$$\begin{aligned}
 \sum_{j=1}^J c_j x_j &\rightarrow \min \\
 \sum_{j=1}^J a_{ij} x_j &= 1, \quad \forall i = 1, \dots, n, \\
 \sum_{j=1}^J x_j &= k \\
 x_j &\in \{0, 1\}, \quad \forall j = 1, \dots, J.
 \end{aligned}
 \tag{7.3}$$

#### 7.1.1. Näherungsalgorithmen.

##### Savings-Algorithmus

Gegeben seien die Daten des Tourenproblems.

Gesucht sind Touren zur Belieferung der Kunden.

Schritt 1: Konstruiere  $n$  Pendeltouren  $0 \rightarrow i \rightarrow 0$ ,  $i = 1, \dots, n$ .

Schritt 2: Verbinde, solange dies möglich ist, je zwei Touren zu einer einzigen, wenn dadurch Kosten eingespart werden (Länge der Umwege:  $g_{ij} = d_{0i} + d_{0j} - d_{ij}$  positiv ist) und eine zulässige Tour entsteht.

##### Sweep-Algorithmus

Gegeben seien die Daten des Tourenproblems.

Gesucht sind Touren zur Belieferung der Kunden.

Schritt 1: Sortiere die Kunden in einer gewissen Reihenfolge.

Schritt 2: Bestimme Touren durch Aufnahme der Kunden in der gewählten Reihenfolge solange, wie die entstehenden Touren zulässig sind.

Schritt 3: Verkürze die Touren zum Beispiel durch Lösung des Rundreiseproblems.

Graphische Möglichkeit der Ordnung der Kunden: Sortiere sie entsprechend nicht fallendem Winkel zwischen der positiven  $x$ -Achse und dem Strahl durch Depot und Kundenort.

Überleitung zum nächsten Algorithmus (die Anzahl der zu bildenden Touren ( $k$ ) sei nicht fest vorgegeben):

Ersetze das Mengenaufteilungsproblem

$$\begin{aligned} \sum_{j=1}^J c_j x_j &\rightarrow \min \\ \sum_{j=1}^J a_{ij} x_j &= 1, \quad \forall i = 1, \dots, n, \\ x_j &\in \{0, 1\}, \quad \forall j = 1, \dots, J. \end{aligned} \tag{7.4}$$

durch Anwendung eines Strafzuganges durch das Mengenüberdeckungsproblem

$$\begin{aligned} \sum_{j=1}^J c_j x_j + \alpha \sum_{i=1}^n \left( \sum_{j=1}^J a_{ij} x_j - 1 \right) &\rightarrow \min \\ \sum_{j=1}^J a_{ij} x_j &\geq 1, \quad \forall i = 1, \dots, n, \\ x_j &\in \{0, 1\}, \quad \forall j = 1, \dots, J. \end{aligned} \tag{7.5}$$

mit einem fixierten Strafparameter  $\alpha \geq 0$ .

**Satz 7.2** Wenn das Problem (7.4) lösbar ist und  $\alpha > \sum_{j=1}^J c_j$  gewählt wird, so haben beide Probleme (7.4) und (7.5) die gleichen optimalen Zielfunktionswerte.

Das Problem (7.5) ist äquivalent zu

$$\begin{aligned} \sum_{j=1}^J (c_j + \alpha \sum_{i=1}^n a_{ij}) x_j &\rightarrow \min \\ \sum_{j=1}^J a_{ij} x_j &\geq 1, \quad \forall i = 1, \dots, n, \\ x_j &\in \{0, 1\}, \quad \forall j = 1, \dots, J. \end{aligned} \tag{7.6}$$

Setze  $g_j = c_j + \alpha \sum_{i=1}^n a_{ij}$ . Der folgende Algorithmus wählt sukzessive neue Touren aus der Menge  $M$  aus.

Wenn  $I$  die Menge der noch nicht in den schon ausgewählten Touren enthaltenen Kunden ist, so ist

$$r_j = \frac{g_j}{\sum_{i \in I} a_{ij}}$$

der Anteil der neu entstehenden Kosten für jeden der neu aufgenommenen Kunden, wenn die Route  $T_j$  realisiert wird.

### Greedy-Algorithmus

*Gegeben* seien die Daten des Tourenproblems.

*Gesucht* sind Touren zur Belieferung der Kunden.

Schritt 1: Es sei  $x = 0$ , d.h. keine Tour ist ausgewählt.

Schritt 2: Setze  $J := \{j : x_j = 0\}$ ,  $I = \{i : \sum_{j=1}^J a_{ij} x_j = 0\}$ . Wenn  $I = \emptyset$ , stopp.

Schritt 3: Wähle  $j_0 \in \underset{j}{\operatorname{Argmin}} \{r_j : \sum_{i \in I} a_{ij} > 0\}$ . Setze  $x_{j_0} := 1$  und gehe zu Schritt 2.

### 7.1.2. Modell für das Tourenplanungsproblem mit Zeitfenstern.

Wir beginnen mit dem Rundreiseproblem.

$$\begin{aligned}
 & \sum_{i=1}^m \sum_{j=1}^m c_{ij} x_{ij} \rightarrow \min \\
 & \sum_{i=1}^m x_{ij} = 1 \quad \forall j = 1, \dots, m \\
 & \sum_{j=1}^m x_{ij} = 1 \quad \forall i = 1, \dots, m \\
 & \sum_{i,j \in S} x_{ij} \leq |S| - 1 \quad \forall S \subset \{1, \dots, m\} \\
 & x_{ij} \in \{0, 1\} \quad \forall i, j = 1, \dots, m.
 \end{aligned} \tag{7.7}$$

Beim Tourenproblem haben wir  $k$  Touren und Kanten des Graphen können in der Route nur dann sein, wenn die damit inzidenten Knoten zur Tour gehören. Außerdem ist noch das Depot zu berücksichtigen und die Kapazitätsbeschränkung.

$$\begin{aligned}
 & \sum_{t=1}^J \sum_{i=0}^m \sum_{j=0}^m c_{ij} y_{ijt} \rightarrow \min \\
 & \sum_{i=1}^m a_i x_{it} \leq b \quad \forall t = 1, \dots, J \\
 & \sum_{t=1}^J x_{it} = 1 \quad \forall i = 1, \dots, m \\
 & \sum_{t=1}^J x_{0t} = k \\
 & \sum_{i=1}^m y_{ijt} = x_{jt} \quad \forall j = 0, \dots, m, t = 1, \dots, J \\
 & \sum_{j=1}^m y_{ijt} = x_{it} \quad \forall i = 0, \dots, m, t = 1, \dots, J \\
 & \sum_{i,j \in S} y_{ijt} \leq |S| - 1 \quad \forall S \subset \{1, \dots, m\}, t = 1, \dots, J \\
 & y_{ijt} \in \{0, 1\} \quad \forall i, j = 1, \dots, m, t = 1, \dots, J \\
 & x_{it} \in \{0, 1\} \quad \forall i = 1, \dots, m, t = 1, \dots, J
 \end{aligned} \tag{7.8}$$

Beachtung von Zeitfenstern beim Kunden  $i$ : Kunde  $i$  soll in der Zeit zwischen  $v_i$  und  $w_i$  beliefert werden.

Dann ist also der Zeitpunkt des Erreichens von Ort  $P_i$  zu berechnen. Das ist

$$t_i = t_j + c_{ij},$$

wenn das Fahrzeug direkt vom Kunden  $j$  zum Kunden  $i$  fährt und  $t_i \geq v_i$  ist. Diese Zeitpunkte kann man sukzessive mit berechnen. Fährt das Fahrzeug nicht direkt vom Kunden  $j$  zum Kunden  $i$ , so soll keine Forderung entstehen (durch Abzug einer großen Zahl in der Bedingung).

Der Einfachheit halber wird die Fahrtdauer für eine Kante gleich der Länge dieser Kante gesetzt.

Damit ergibt sich das folgende Modell:

$$\begin{aligned}
& \sum_{t=1}^J \sum_{i=0}^m \sum_{j=0}^m c_{ij} y_{ijt} \rightarrow \min \\
& \sum_{i=1}^m a_i x_{it} \leq b \quad \forall t = 1, \dots, J \\
& \sum_{t=1}^J x_{it} = 1 \quad \forall i = 1, \dots, m \\
& \sum_{t=1}^J x_{0t} = k \\
& \sum_{i=1}^m y_{ijt} = x_{jt} \quad \forall j = 0, \dots, m, t = 1, \dots, J \\
& \sum_{j=1}^m y_{ijt} = x_{it} \quad \forall i = 0, \dots, m, t = 1, \dots, J \\
& \sum_{i,j \in S} y_{ijt} \leq |S| - 1 \quad \forall S \subset \{1, \dots, m\}, t = 1, \dots, J \\
& t_i \geq t_j + c_{ij} - (1 - y_{ijt})T \quad \forall i, j = 1, \dots, m, t = 1, \dots, J \\
& v_i \leq t_i \leq w_i \quad \forall i = 1, \dots, m \\
& y_{ijt} \in \{0, 1\} \quad \forall i, j = 1, \dots, m, t = 1, \dots, J \\
& x_{it} \in \{0, 1\} \quad \forall i = 1, \dots, m, t = 1, \dots, J
\end{aligned} \tag{7.9}$$

## 7.2. Standortprobleme

Ein Unternehmen beliefert  $n$  Kunden mit dem Bedarf  $b_j$  mit einem homogenen Gut. Dieses Unternehmen möchte zur Senkung der Vertriebskosten Warenhäuser errichten. Hierfür stehen  $m$  potentielle Standorte zur Verfügung.

Gegeben: Warenhaus am Standort  $i$  mit Kapazität von  $a_i$  Mengeneinheiten kann errichtet werden für Fixkosten in der Höhe von  $f_i$  Geldeinheiten.

Transportkosten  $c_{ij}$  für die Belieferung des Kunden  $j$  aus dem Warenhaus  $i$  mit einer Mengeneinheit des homogenen Gutes.

Gesucht: Standorte, an denen die Warenhäuser errichtet werden sollen.

Ziel: Minimale Kosten aus dem Transport der Güter und der Errichtung der Warenhäuser.

### Variablen:

- $x_{ij}$ : An den Kunden  $j$  gelieferte Menge an Waren aus dem Warenhaus am Standort  $i$
- $y_i = \begin{cases} 1 & \text{wenn am Standort } i \text{ ein Warenhaus errichtet wird} \\ 0 & \text{sonst} \end{cases}$

### 7.2.1. Das kapazitierte Standortproblem.

Gegeben: Bedarf  $b_j$  von  $n$  Kunden, Kapazität  $a_i$  von Warenhäusern an  $m$  potentiellen Standorten, Fixkosten  $f_i$  für die Errichtung der Warenhäuser und die Transportkosten  $c_{ij}$ .

Gesucht: Standorte der Warenhäuser

Ziel: minimale Gesamtkosten

$$\begin{aligned}
\sum_{i=1}^m \sum_{j=1}^n c_{ij} x_{ij} + \sum_{i=1}^m f_i y_i &\rightarrow \min \\
\sum_{i=1}^m x_{ij} &= b_j, \quad j = 1, \dots, n, \\
\sum_{j=1}^n x_{ij} &\leq a_i y_i, \quad i = 1, \dots, m \\
x_{ij} &\leq b_j y_i, \quad i = 1, \dots, m, \quad j = 1, \dots, n, \\
x_{ij} \geq 0, y_i &\in \{0, 1\}, \quad i = 1, \dots, m, \quad j = 1, \dots, n.
\end{aligned} \tag{7.10}$$

Das Transportproblem kann auch durch ein anderes Problem, zum Beispiel ein Tourenproblem, ersetzt werden. Dann kann auch auf die Forderung, dass ein homogenes Gut geliefert wird, verzichtet werden.

### 7.2.2. Das unkapazitierte Standortproblem.

Gegeben:  $b_j$  von  $n$  Kunden, Fixkosten  $f_i$  für die Errichtung der Warenhäuser an  $m$  potentiellen Standorten und die Transportkosten  $c_{ij}$ .

Gesucht: Standorte der Warenhäuser

Ziel: minimale Gesamtkosten

$$\begin{aligned}
\sum_{i=1}^m \sum_{j=1}^n c_{ij} x_{ij} + \sum_{i=1}^m f_i y_i &\rightarrow \min \\
\sum_{i=1}^m x_{ij} &= 1, \quad j = 1, \dots, n, \\
x_{ij} &\leq y_i, \quad i = 1, \dots, m, \quad j = 1, \dots, n, \\
x_{ij} \geq 0, y_i &\in \{0, 1\}, \quad i = 1, \dots, m, \quad j = 1, \dots, n.
\end{aligned} \tag{7.11}$$

**Satz 7.3** Das unkapazitierte Standortproblem ist  $\mathcal{NP}$ -schwer.

**Satz 7.4** Es sei  $G = (V, E)$  ein Baum, jeder Kunde sein in einem Knoten positioniert, jeder potentielle Standort für ein Warenhaus ebenfalls. Die Transportkosten seien gleich den Längen der (eindeutigen) Wege zwischen den Kundenorten und den potentiellen Standorten. Dann kann das unkapazitierte Standortproblem mit  $O(|E|^3)$  Rechenoperationen gelöst werden, es ist also polynomial lösbar.

Umformulierung des unkapazitierten Standortproblems:

Betrachte das Problem (7.11). Wähle Menge  $S \subseteq \{1, \dots, m\}$  mit

$$y_i = 1 \iff i \in S$$

Dann ergibt sich das äquivalente Problem

$$z(S) := \sum_{j=1}^n \min\{c_{ij} : i \in S\} + \sum_{i \in S} f_i \rightarrow \min_{S \subseteq \{1, \dots, m\}} \tag{7.12}$$

**Definition 7.5** Eine Mengenfunktion  $w : 2^J \rightarrow \mathbb{R}$  heißt *supermodular*, wenn

$$w(S \cup \{k\}) - w(S) \geq w(R \cup \{k\}) - w(R) \quad \forall R \subseteq S \subseteq J \setminus \{k\}.$$

**Satz 7.6** Die Funktion  $z$  in (7.12) ist supermodular.



Lösungsalgorithmus für das unkapazitierte Standortproblem:

**ADD-Algorithmus** (ein greedy-Algorithmus):

Schritt 0: Für  $S \subseteq \{1, \dots, m\}$  setze  $z(S) := \sum_{j=1}^n \min\{c_{ij} : i \in S\} + \sum_{i \in S} f_i$ ,  $S^0 := \emptyset$ ,  $k := 0$ .

Schritt 1: Wähle  $i_k$  so, dass  $z(S^k \cup \{i_k\}) \leq z(S^k \cup \{i\}) \forall i \notin S^k$ .

Schritt 2: Wenn  $z(S^k \cup \{i_k\}) - z(S^k) \geq 0$  ist, stopp.

Sonst setze  $S^{k+1} := S^k \cup \{i_k\}$ ,  $k := k + 1$  und gehe zu Schritt 1.

Eine analoge Strategie ist auch möglich, wenn mit der Menge  $S^0 := \{1, \dots, m\}$  begonnen wird und ein Standort ausgeschlossen wird, wenn dadurch der Zielfunktionswert sinkt (DROP-Algorithmus).

### 7.2.3. Exakte Lösung des kapazitierten Standortproblems mit branch-and-bound Algorithmus.

Schranken Aufgabe: Ersetze  $y_i \in \{0, 1\}$  durch  $0 \leq y_i \leq 1$ . Wenn die redundanten Ungleichungen  $x_{ij} \leq b_j y_i$  fehlen,  $f_i \geq 0$ :

$$\begin{aligned} \sum_{i=1}^m \sum_{j=1}^n \left( c_{ij} + \frac{f_i}{a_i} \right) x_{ij} &\rightarrow \min \\ \sum_{i=1}^m x_{ij} &= b_j, \quad j = 1, \dots, n, \\ \sum_{j=1}^n x_{ij} &\leq a_i, \quad i = 1, \dots, m \\ x_{ij} &\geq 0, \quad i = 1, \dots, m, \quad j = 1, \dots, n. \end{aligned} \tag{7.13}$$

Zur Konstruktion wurde  $\sum_{j=1}^n x_{ij} = a_i y_i$  gesetzt. Die Werte von  $y_i$  sind also aus  $y_i = \frac{1}{a_i} \sum_{j=1}^n x_{ij}$  zu berechnen. Verzweigung erfolgt bezüglich der  $y_i$ .

Schärfere Schranken (und damit weniger Iterationen im branch-and-bound Algorithmus), wenn die redundante Nebenbedingungen  $x_{ij} \leq b_j y_i$  beachtet werden.



## Grundlagen zur nichtlinearen Optimierung

### 8.1. Optimierung ohne Nebenbedingungen

In diesem Abschnitt wird die folgende Optimierungsaufgabe betrachtet:

$$f(x) \longrightarrow \min_{x \in \mathbb{R}^n}$$

wobei  $f : \mathbb{R}^n \rightarrow \mathbb{R}$  eine nichtlineare Funktion bezeichnet.

- Definition 8.1**
- (1) Ein Punkt  $\bar{x} \in \mathbb{R}^n$  heißt globales Minimum der Funktion  $f$ , falls  $f(\bar{x}) \leq f(x)$  für alle  $x \in \mathbb{R}^n$  gilt.
  - (2) Ein Punkt  $\bar{x} \in \mathbb{R}^n$  heißt lokales Minimum der Funktion  $f$ , falls ein  $\epsilon$  existiert, so dass  $f(\bar{x}) \leq f(x)$  für alle  $x$  in einer Umgebung  $U_\epsilon(\bar{x})$  gilt.
  - (3) Ein Punkt  $\bar{x} \in \mathbb{R}^n$  heißt globales Maximum der Funktion  $f$ , falls  $f(\bar{x}) \geq f(x)$  für alle  $x \in \mathbb{R}^n$  gilt.
  - (4) Ein Punkt  $\bar{x} \in \mathbb{R}^n$  heißt lokales Maximum der Funktion  $f$ , falls ein  $\epsilon$  existiert, so dass  $f(\bar{x}) \geq f(x)$  für alle  $x$  in einer Umgebung  $U_\epsilon(\bar{x})$  gilt.

**Satz 8.2** Sei  $f : \mathbb{R}^n \rightarrow \mathbb{R}$ ,  $\bar{x}$  sei ein lokales Minimum oder Maximum und die Funktion  $f$  sei im Punkt  $\bar{x}$  differenzierbar. Dann gilt:

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

**Bemerkung 8.3** Punkte  $\bar{x}$  mit  $\nabla f(\bar{x}) = 0$  heißen stationär.

**Satz 8.4** Sei  $f : \mathbb{R}^n \rightarrow \mathbb{R}$  in  $\bar{x}$  zweimal stetig differenzierbar mit  $\nabla f(\bar{x}) = 0$ .

Wenn  $\nabla^2 f(\bar{x})$  positiv definit ist, so ist  $\bar{x}$  ein lokales Minimum.

Wenn  $\nabla^2 f(\bar{x})$  negativ definit ist, so ist  $\bar{x}$  ein lokales Maximum.

Falls es Vektoren  $d_1, d_2 \in \mathbb{R}^n$  mit  $d_1^\top \nabla^2 f(\bar{x}) d_1 < 0$  und  $d_2^\top \nabla^2 f(\bar{x}) d_2 > 0$  gibt, so ist  $\bar{x}$  kein Extremum.

Dabei ist  $\nabla^2 f(\bar{x})$  die Hesse-Matrix von  $f$  in  $\bar{x}$ .

Die Hesse-Matrix einer zweimal partiell differenzierbaren Funktion  $f$  im Punkt  $x$  ist eine Matrix der Form:

$$\nabla^2 f(x) = \begin{pmatrix} p_{11} & p_{12} & \cdots & p_{1n} \\ p_{21} & p_{22} & \cdots & p_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ p_{n1} & p_{n2} & \cdots & p_{nn} \end{pmatrix}$$

Dabei entspricht  $p_{ij}$  ( $i, j = 1, \dots, n$ ) der partiellen Ableitung zweiter Ordnung:

$$p_{ij} = \frac{\partial^2 f(x)}{\partial x_i \partial x_j}.$$

**Bemerkung 8.5**  $\nabla^2 f(\bar{x})$  ist positiv definit  $\Leftrightarrow d^\top \nabla^2 f(\bar{x}) d > 0 \forall d \neq 0, d \in \mathbb{R}^n$ .  
 $\nabla^2 f(\bar{x})$  ist negativ definit  $\Leftrightarrow -\nabla^2 f(\bar{x})$  positiv definit ist.

## 8.2. Optimierung mit Gleichungsnebenbedingungen

Die in diesem Abschnitt betrachtete Aufgabe lautet:

$$\begin{aligned} f(x) &\rightarrow \min \\ g_1(x) &= 0 \\ &\vdots \\ g_p(x) &= 0 \end{aligned} \tag{8.1}$$

Wir ordnen der Aufgabe (8.1) eine Funktion zu:

$$L(x, u) = f(x) + u_1 g_1(x) + \dots + u_p g_p(x).$$

Die Funktion  $L$  heißt Lagrangefunktion und  $u \in \mathbb{R}^p$  bezeichnet die Lagrangemultiplikatoren.

**Definition 8.6** Sei  $\bar{x}$  zulässig für die Aufgabe (8.1), d.h.  $g_i(\bar{x}) = 0$ ,  $i = 1, \dots, p$ .  $\bar{x}$  ist ein lokales Minimum der Aufgabe (8.1), wenn

$$\exists \epsilon > 0 : f(\hat{x}) \geq f(\bar{x}) \forall \hat{x} : g_i(\hat{x}) = 0, i = 1, \dots, p \text{ und } \|\hat{x} - \bar{x}\| \leq \epsilon.$$

Analog lässt sich ein lokales Maximum für eine Maximierungsaufgabe definieren.

**Satz 8.7** Sei  $\bar{x}$  zulässig für die Aufgabe (8.1):  $g_i(\bar{x}) = 0, i = 1, \dots, p$ ,  $\bar{x}$  sei lokales Minimum der Aufgabe (8.1), die Funktionen  $f, g_i$  stetig differenzierbar und die Gradienten  $\{\nabla g_i(\bar{x}) : i = 1, \dots, p\}$  linear unabhängig. Dann gibt es  $\bar{u}_i, i = 1, \dots, p$  mit

$$\nabla_x L(\bar{x}, \bar{u}) = 0.$$

Hier ist  $\nabla_x L(\bar{x}, \bar{u})$  der Gradient von  $L(x, u)$  im Punkt  $(\bar{x}, \bar{u})$  bezüglich  $x$ .

**Bemerkung 8.8** Punkte, die die in dem Satz 8.7 beschriebenen Bedingungen erfüllen, heißen stationär.

**Bemerkung 8.9** Ohne die lineare Unabhängigkeit der Gradienten  $\{\nabla g_i(\bar{x})\}$  gilt der Satz 8.7 im Allgemeinen nicht.

**Satz 8.10** Sei  $\bar{x}$  stationär für die Aufgabe (8.1), die Funktionen  $f, g_i$  zweimal stetig differenzierbar. Wenn

$$d^\top \nabla_{xx}^2 L(\bar{x}, \bar{u}) d > 0 \forall d \in \mathbb{R}^n : \nabla g_i(\bar{x}) d = 0, i = 1, \dots, p, d \neq 0$$

ist, so ist  $\bar{x}$  ein lokales Minimum der Aufgabe (8.1).

Dabei ist  $\nabla_{xx} L(\bar{x}, \bar{u})$  die Hesse-Matrix von  $L(x, u)$  bezüglich  $x$ .

## 8.3. Aufgaben mit Ungleichungsnebenbedingungen

In diesem Abschnitt wird die folgende Optimierungsaufgabe betrachtet:

$$\begin{aligned} f(x) &\rightarrow \min \\ g_1(x) &\leq 0 \\ &\vdots \\ g_p(x) &\leq 0 \end{aligned} \tag{8.2}$$

**Definition 8.11** Sei  $\bar{x}$  zulässig für die Aufgabe (8.2), d.h.  $g_i(\bar{x}) \leq 0, i = 1, \dots, p$ .  $\bar{x}$  ist ein lokales Minimum der Aufgabe (8.2), wenn

$$\exists \epsilon > 0 : f(\hat{x}) \geq f(\bar{x}) \forall \hat{x} : g_i(\hat{x}) \leq 0, i = 1, \dots, p \text{ und } \|\hat{x} - \bar{x}\| \leq \epsilon.$$

Analog lässt sich ein lokales Maximum für eine Maximierungsaufgabe definieren.

**Satz 8.12** Sei  $\bar{x}$  ein lokales Minimum der Aufgabe (8.2), die Funktionen  $f, g_i$  seien stetig differenzierbar in  $\bar{x}$ , die Gradienten  $\{\nabla g_i(\bar{x}) : \forall i \in I(\bar{x})\}$  linear unabhängig mit  $I(\bar{x}) = \{j : g_j(\bar{x}) = 0\}$ . Dann gilt:  
 $\exists \bar{u}$  mit

$$\nabla f(\bar{x}) + \sum_{i=1}^p \bar{u}_i \nabla g_i(\bar{x}) = 0 \quad (8.3)$$

$$g_i(\bar{x}) \leq 0, \bar{u}_i \geq 0, \bar{u}_i g_i(\bar{x}) = 0 \quad \forall i$$

**Definition 8.13**  $\bar{x}$  mit  $\exists \bar{u}$ , so dass  $(\bar{x}, \bar{u})$  die Bedingungen (8.3) erfüllt, ist stationär.

**Satz 8.14** Sei  $\bar{x}$  stationär,  $(\bar{x}, \bar{u})$  lösen (8.3),  $f, g_i$  in  $\bar{x}$  zweimal stetig differenzierbar. Wenn

$$d^\top \nabla_{xx}^2 L(\bar{x}, \bar{u}) d > 0 \quad \forall d \neq 0 \text{ mit}$$

$$\nabla g_i(\bar{x}) d = 0 \quad \forall i : \bar{u}_i > 0$$

$$\nabla g_i(\bar{x}) d \leq 0 \quad \forall i : i \in I(\bar{x})$$

gilt, so ist  $\bar{x}$  lokal optimal.



## Literaturverzeichnis

- [1] D. Bertsimas and J. Tsitsiklis. *Introduction to Linear Optimization*. Athena Scientific, 1997.
- [2] G. Dantzig and M. Thapa. *Linear Programming: Introduction*. Springer Verlag, New York Berlin Heidelberg, 1997.
- [3] S. Dempe and H. Schreier. *Operations Research. Deterministische Modelle und Methoden*. B. G. Teubner Verlag, Wiesbaden, 2006.
- [4] S. Dempe and T. Unger. *Lineare Optimierung. Modell, Lösung, Anwendung*. Vieweg + Teubner Verlag, Wiesbaden, 2010.
- [5] K. Marti and D. Gröger. *Einführung in die lineare und nichtlineare Optimierung*. Physica-Verlag, Heidelberg, 2000.
- [6] K. Murty. *Operations Research. Deterministic optimization models*. Prentice Hal, Englewood Cliffs, NJ, 1995.
- [7] K. Neumann and M. Morlock. *Operations Research*. Carl Hanser Verlag, München Wien, 2002.