

# Vorlesungsmitschrift Operations Research I und II

**Bemerkung:** Dies ist eine Überarbeitung der beiden Skripte [1] und [2] zu den oben genannten Vorlesungen von Prof. Sebastian, ergänzt um Anmerkungen, die während der Vorlesungen im WS 02/03 und SS 03 gegeben wurden. Zusätzlich wurden einige Bemerkungen aus [3] eingefügt. Auf die Abbildungen aus den beiden Skripten wurde vollständig verzichtet.

geT<sub>E</sub>Xt von Volker Peters<sup>1</sup>

---

<sup>1</sup>Volker.Peters@rwth-aachen.de

## Inhaltsverzeichnis

<b>0 Was ist Operations Research?</b>	<b>1</b>
<b>1 Typische OR-Anwendungen</b>	<b>1</b>
1.1 Produktionsplanung . . . . .	1
1.2 Transportproblem . . . . .	2
1.3 Zuordnungsproblem . . . . .	2
1.4 Weitere Probleme . . . . .	3
<b>2 Das Simplexverfahren</b>	<b>3</b>
2.1 Einführende Beispiele — Modellierungsgrundsätze . . . . .	3
2.2 Mathematische Grundlagen der Linearen Programmierung (LP) . . . . .	5
2.3 Theorie der LP . . . . .	7
2.4 Die Simplexmethode — Rechenverfahren . . . . .	12
2.4.1 Ausgangsbasislösung . . . . .	13
2.4.2 Aufnahme­regel . . . . .	13
2.4.3 Eliminations­regel . . . . .	14
2.4.4 Umformung . . . . .	15
2.5 Erweiterungen (Zweiphasenwechsel) . . . . .	16
<b>3 Dualität in der LP</b>	<b>17</b>
3.1 Duales Modell . . . . .	17
3.2 Ökonomische Interpretation der dualen Lösung . . . . .	20
3.3 Dualer Simplexalgorithmus . . . . .	21
3.3.1 Ausgangsbasislösung . . . . .	21
3.3.2 Eliminations­regel . . . . .	21
3.3.3 Aufnahme­regel . . . . .	21
3.3.4 Abbruchregel (Optimalitätskriterium) . . . . .	22
<b>4 Postoptimale Analysen</b>	<b>22</b>
4.1 Sensitivitätsanalyse . . . . .	22
4.2 Postoptimale Modifikation . . . . .	23
4.2.1 Änderung der ZF-Koeffizienten . . . . .	23
4.2.2 Änderung der rechten Seite . . . . .	24
4.2.3 Hinzufügen von Variablen . . . . .	24
4.2.4 Hinzufügen von weiteren $\leq$ -Restriktionen . . . . .	24
4.3 Idee der Parametrischen Programmierung . . . . .	24
<b>5 Ganzzahlige LP</b>	<b>25</b>
5.1 Problemstellung, Beispiele . . . . .	25
5.2 Das Schnittebenenverfahren von Gomory . . . . .	25
<b>6 Revidierter Simplexalgorithmus</b>	<b>28</b>
6.1 Grundlagen . . . . .	28
6.2 Produktform der Inversen . . . . .	28
6.3 Basistransformation des revidierten Simplexalgorithmus . . . . .	28
<b>7 Projektplanung</b>	<b>28</b>

<b>8</b>	<b>Entwicklung und Grundlagen der Netzplantechnik</b>	<b>29</b>
8.1	Begriffe der Netzplantechnik . . . . .	29
8.2	CPM . . . . .	32
8.3	PERT (bei Ereignisknotennetzen) . . . . .	34
8.4	MPM . . . . .	36
8.5	Exkurs: Gantt-Chart (Balkendiagramm) . . . . .	36
<b>9</b>	<b>Zuordnungsproblem</b>	<b>37</b>
9.1	Anwendungsbeispiel . . . . .	37
9.2	Ein Heuristisches Eröffnungsverfahren . . . . .	38
9.3	Die Ungarische Methode . . . . .	39
<b>10</b>	<b>Transport- und Umladeproblem</b>	<b>41</b>
10.1	Das Transportproblem . . . . .	42
10.1.1	Modellierung . . . . .	42
10.1.2	Eröffnungsheuristiken . . . . .	43
10.1.3	Die MODI-Methode . . . . .	45
10.2	Das Umladeproblem . . . . .	49
<b>11</b>	<b>Das Rucksackproblem</b>	<b>51</b>
11.1	Anwendungsbeispiel . . . . .	51
11.2	Zwei Branch and Bound Algorithmen für das Rucksackproblem . . . . .	52
<b>12</b>	<b>Tourenplanung</b>	<b>56</b>
12.1	Das Traveling Salesman Problem (TSP) . . . . .	56
12.1.1	Eine allgemeine Heuristik . . . . .	56
12.1.2	Graphentheoretische Formulierung . . . . .	56
12.1.3	Beschreibung als Zuordnungsproblem . . . . .	57
12.1.4	Lösung mit Branch and Bound . . . . .	57
12.2	Das Vehicle Routing Problem (VRP) . . . . .	59
12.2.1	Modell zum VRP . . . . .	59
12.2.2	Eröffnungsheuristiken (Savings- und Sweep-Verfahren) . . . . .	60
12.2.3	Verbesserungsverfahren für TSP und VRP . . . . .	62
<b>13</b>	<b>Gemischt-ganzzahlige Programmierung, Branch and Bound</b>	<b>63</b>
13.1	Beispiel eines gemischt-ganzzahligen LP's . . . . .	63
13.2	Branch and Bound: Ansatz von Dakin . . . . .	65
13.3	Allgemeine Darstellung des Prinzips „Branch and Bound“ . . . . .	66
<b>14</b>	<b>Dynamische Programmierung</b>	<b>68</b>
14.1	Optimalitätsprinzip und Bellmannsche Gleichungen . . . . .	69
14.2	Anwendung: Lagerhaltungsmodelle . . . . .	72
14.2.1	Klassische Losgrößenmodelle . . . . .	73
14.2.2	Dynamisches Deterministisches Modell . . . . .	76
14.2.3	Beispiel eines 3-stufigen Lagerhaltungsmodells mit Dynamischer Pro- grammierung . . . . .	78
14.2.4	Das Verfahren von Wagner und Whitin . . . . .	79
14.3	Anwendung: Marketing-Beispiel . . . . .	81

<b>15 Nichtlineare Optimierung</b>	<b>82</b>
15.1 Grundlagen . . . . .	83
15.2 Konvexe Kuhn-Tucker-Theorie . . . . .	83
15.3 Kuhn-Tucker-Theorie Zugang über Lagrangefunktion . . . . .	86
15.4 Quadratisches Programmieren . . . . .	87
15.5 Verfahren der NLP . . . . .	88
15.5.1 Minimierung ohne Restiktionen . . . . .	88
15.5.2 Mehrdimensionales Suche (unrestringiert) — Differenzierbarkeit vor- ausgesetzt . . . . .	88
15.5.3 Verfahren zulässiger Richtungen (restringiert) . . . . .	89
15.5.4 Minimierung mit Nebenbedingungen . . . . .	91
<b>Literaturverzeichnis</b>	<b>92</b>

## 0 Was ist Operations Research?

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

- quantifizierbare Probleme
- Entscheidungsmöglichkeiten
- beste Alternative unter allen möglichen aussuchen
- Gütekriterium

Begriff OR? Wirksamkeit militärischer Operationen (zu Beginn des 2. Weltkrieges in England) mit Hilfe mathematischer Verfahren, Operational Research genannt. In den USA wurde von Operations Research gesprochen. Keine Übersetzung hat sich durchgesetzt. Gegen Ende der vierziger Jahre fing die Entwicklung leistungsfähiger Verfahren für die vielfältigen Planungsprobleme in zivilen Unternehmen an.

## 1 Typische OR-Anwendungen

OR-Probleme können aus verschiedensten Bereichen stammen. Das Gemeinsame all dieser Probleme besteht darin, dass ein Gütekriterium (Zielfunktion) einen möglichst kleinen oder großen Wert annehmen soll. Die zu treffenden Entscheidungen werden durch Entscheidungsvariablen beschrieben. Es sind gewissen Nebenbedingungen (Beschränkungen, Restriktionen) zu beachten.

### 1.1 Produktionsplanung

Auf einer Produktionsanlage können  $n$  Güter hergestellt werden, wobei  $m$  verschiedene Rohstoffe benötigt werden. Ein optimaler Produktionsplan besteht darin zu entscheiden, welche Mengen  $x_1, \dots, x_n$  von der Gütern  $1, \dots, n$  zu produzieren sind, wobei folgendes zu berücksichtigen ist:

- (a) Erzielter Gesamtgewinn soll maximiert werden. Gewinne  $g_j$  pro produzierter Mengeneinheit (ME) von Gut  $j = 1, \dots, n$  sind bekannt.
- (b) Güter konkurrieren hinsichtlich der zur Produktion benötigten Rohstoffe.  $b_i$  für  $i = 1, \dots, m$  ist die verfügbare Menge von Rohstoff  $i$ .
- (c) „Faktor-Einsatzmenge“ für Produktion einer ME eines jeden Gutes.  $a_{ij}$  benötigte Menge von Rohstoff  $i$  zur Produktion von 1 ME von Gut  $j$ .

⇒ Optimierungsproblem:  
Maximiere

$$g_1x_1 + \dots + g_nx_n \quad (\text{Zielfunktion})$$

unter den  $m + n$  Nebenbedingungen

$$\begin{aligned} a_{i1}x_1 + \dots + a_{in}x_n &\leq b_i \quad (i = 1, \dots, m) \\ x_1 &\geq 0, \dots, x_n \geq 0 \end{aligned}$$

In den Nebenbedingungen können auch Gleichungen auftreten (z.B. Materialbilanz-Gleichungen). Es kommen nur lineare Funktionen vor. Zielfunktion ist lineare Funktion der Entscheidungsvariablen und Nebenbedingungen liegen in Form linearer Gleichungen bzw. Ungleichungen vor  $\Rightarrow$  lineares Optimierungsproblem (in Abschnitt 15 auch nichtlineare Optimierungsprobleme)

Anderes typisches Beispiel: Rohölverarbeitung in Raffinerien [1].

## 1.2 Transportproblem

Ein Gut, das in  $m$  Lagern vorhanden ist, soll zu  $n$  Kunden transportiert werden. Zu entscheiden ist, welche Menge  $x_{ij}$  des Gutes vom Lager  $i$  zum Kunden  $j$  zu transportieren ist ( $i = 1, \dots, m; j = 1, \dots, n$ ). Hierbei ist zu beachten:

- Transportkosten sind proportional zur transportierten Menge:  $c_{ij}x_{ij}$ .
- Die Summe der gesamten Transportkosten ist zu minimieren.
- Vorhandene Menge des Gutes  $a_i$  sowie der Bedarf  $b_j$  sind bekannt.
- Bedarf jedes Kunden muss gedeckt werden.
- Negative Transportmenge (Rücktransport) sind nicht zugelassen.

Minimiere

$$\sum_{i=1}^m \sum_{j=1}^n c_{ij}x_{ij} \quad (\text{Gesamtkosten})$$

unter den Nebenbedingungen

$$\begin{aligned} \sum_{j=1}^n x_{ij} &\leq a_i \quad (\text{für feste Lager } i = 1, \dots, m) && (\text{Lagerkapazitäten}) \\ \sum_{i=1}^m x_{ij} &= b_j \quad (\text{für festen Kunden } j = 1, \dots, n) && (\text{Bedarfsbefriedigung}) \\ x_{ij} &\geq 0 \quad (i = 1, \dots, m; j = 1, \dots, n) \end{aligned}$$

Besondere Nebenbedingungen: Koeffizienten der Variablen sind alle 1.

Umladeproblem: Transport eines Gutes von mehreren Produzenten über Umladeorte zu den Verbrauchern (siehe Abschnitt 10.2).

## 1.3 Zuordnungsproblem

Es sind 3 offene Stellen ( $j = 1, 2, 3$ ) durch 5 qualifizierte Bewerber ( $i = 1, \dots, 5$ ) zu besetzen, wobei die Einarbeitungszeiten  $c_{ij}$  des Bewerbers  $i$  für die Stelle  $j$  bekannt sind. Die Einstellung soll so erfolgen, dass die Summe der Einarbeitungszeiten minimal ist.

$$x_{ij} = \begin{cases} 1 & \text{Bewerber } i \text{ wird für Stelle } j \text{ eingestellt} \\ 0 & \text{sonst} \end{cases}$$

Min.

$$\sum_{i=1}^5 \sum_{j=1}^3 c_{ij}x_{ij}$$

u.d.N.

$$\sum_{j=1}^3 x_{ij} \leq 1 \quad (i = 1, \dots, 5) \quad (\text{Jeder Bewerber steht maximal einmal zur Verfügung.})$$

$$\sum_{i=1}^5 x_{ij} = 1 \quad (j = 1, 2, 3) \quad (\text{Jede Stelle kann nur einmal besetzt werden.})$$

$$x_{ij} \in \{0, 1\}$$

## 1.4 Weitere Probleme

- Energieflussproblem
- Kürzeste Wege in Verkehrsnetzen
- Auslieferung von Gütern (Tourenplanungsproblem)
- Verschnittproblem
- Nichtlineare Produktionsplanung
- Investitionsentscheidungen
- Lagerhaltung
- Maschinenbelegungsplanung
- Warteschlangen

## 2 Das Simplexverfahren

### 2.1 Einführende Beispiele — Modellierungsgrundsätze

Beispiel:

Ein Betrieb produziere zwei Produkte, die drei Fertigungsstufen,

- Teilfertigung (TF) 80 Maschinenstunden (MStd)
- Vormontage (VM) 100 Maschinenstunden
- Endmontage (EM) 75 Maschinenstunden

zu durchlaufen haben. Die drei Abteilungen werden durch die beiden Produkte wie folgt belastet:

	Produkt 1	Produkt 2
TF:	4 MStd/Stk.	2 MStd/Stk.
VM:	2 MStd/Stk.	3 MStd/Stk.
EM:	5 MStd/Stk.	1 MStd/Stk.
Erlös:	12 €/Stk.	8 €/Stk.

Gesucht ist das optimale Produktionsprogramm, d.h. wie viel soll von beiden Produkten hergestellt werden, sodass der Gesamterlös maximiert wird und die vorhandenen Kapazitäten nicht überfordert werden.

Modell:

Entscheidungsvariablen (EV)  $x_i$ : Herzustellende Menge von Produkt  $i$  ( $i = 1, 2$ )

Max.

$$z = 12x_1 + 8x_2$$

u.d.N.

$$\begin{array}{ll} \text{I} & 4x_1 + 2x_2 \leq 80 \quad (x_1 = 0, x_2 = 40), (x_1 = 20, x_2 = 0) \\ \text{II} & 2x_1 + 3x_2 \leq 100 \quad (x_1 = 0, x_2 = \frac{100}{3}), (x_1 = 50, x_2 = 0) \\ \text{III} & 5x_1 + 1x_2 \leq 75 \quad (x_1 = 0, x_2 = 75), (x_1 = 15, x_2 = 0) \\ & x_1, x_2 \geq 0 \quad (\text{eigentlich sogar ganzzahlig}) \end{array}$$

Die Isogewinngerade mit maximalem Gewinn schneidet den Punkt  $A = (x_1 = 5, x_2 = 30)$ , d.h.  $x_1 = 5, x_2 = 30$  ist die optimale Lösung (die Lösung ist zulässig und der Gewinn ist maximal). Die optimale Ecke ist diejenige, in der der *Gradient* (die partielle Ableitung) der Zielfunktion im Kegel der Gradienten der Nebenbedingungen liegt.

Lösungsraum  $M$  (zulässiger Bereich) ist die Menge aller zulässigen Produktkombinationen, d.h. aller Punkte der  $(x_1, x_2)$ -Ebene, die allen Nebenbedingungen genügen. Hier ist das der Durchschnitt aller fünf Ungleichungen. Der maximale Wert der Zielfunktion auf  $M$  ist gleich dem größten Wert, für den die Gerade (Isogewinngerade) noch mindestens einen Punkt von  $M$  enthält. Man spricht von der *optimalen Lösung*, d.h. die Lösung ist zulässig und liefert den maximalen Gewinn. Eine graphische Lösung ist im zweidimensionalen Fall einfach und im dreidimensionalen Fall möglich. Höhere Dimensionen erfordern Algorithmen.

Typische Eigenschaften linearer Optimierungsprobleme:

1. Der zulässige Bereich ist ein konvexes Polyeder (Vieleck).
2. Lösungen mit gleichem Zielfunktionswert liegen auf Hyperebenen (Geraden im zweidimensionalen Fall).
3. Hyperebenen verschiedener Zielfunktionswerte sind parallel zueinander.
4. Das Optimum (falls existent) wird auf dem Rand (in einer Ecke oder zwischen optimalen Ecken) von  $M$  angenommen.



5. Mindestens eine Ecke des zulässigen Bereichs stellt eine optimale Lösung dar.

Das heißt, im zweidimensionalen Fall reicht es aus, die Ecken des zulässigen Bereichs zu untersuchen. Bei komplexeren Problemen bietet sich die folgende Vorgehensweise an:

- (1) Bestimme als Startpunkt eine Ecke  $x^0$  von  $M$  (etwa den Ursprung).
- (2) Gehe anschließend zu einer „benachbarten“ Ecke  $x^1$  über (Richtung ist egal), für die (bei einer Maximierungsaufgabe)  $z(x^1) > z(x^0)$  gilt.
- (3) Wiederhole Schritt (2) bis es keine „benachbarte“ Ecke mit einem größeren Zielfunktionswert  $z$  mehr gibt.

Weitere Beispiele:

- Landwirt, der zwei Produkte anbauen kann/will.
- Maschinenbelegungsplan
- Begrenzung der Luftverschmutzung

## 2.2 Mathematische Grundlagen der Linearen Programmierung (LP)

Im diesem Abschnitt werden die zuvor eingeführten Begriffe präzise definiert.

### Definition 1

Punkte  $P(x_1, \dots, x_n)$  des Raumes  $\mathbb{R}^n$  werden als *n-dimensionale Spaltenvektoren*

$$x = \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix}, x = (x_1, x_2, \dots, x_n)^T$$

mit den Komponenten  $x_j \in \mathbb{R}$ ,  $j \in \{1, \dots, n\}$  aufgefasst.

### Definition 2

Ein Raum  $R^*$  heißt *linearer Raum*, wenn gilt:

1.  $x, y \in R^* \Rightarrow z = x + y \in R^*$  (komponentenweise Addition)
2.  $x \in R^*, \lambda \in \mathbb{R} \Rightarrow \lambda \cdot x \in R^*$  (skalare Multiplikation)

$\Rightarrow \mathbb{R}^n$  ist ein linearer Raum.

### Definition 3

Ein Raum  $R^*$  heißt *n-dimensional*, wenn er von  $n$  linear unabhängigen Vektoren (durch komponentenweise Addition und skalarer Multiplikation) erzeugt werden kann.

$\Rightarrow \mathbb{R}^n$  ist ein *n-dimensionaler* Raum, da er durch die Vektoren

$$x_1 = \begin{pmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{pmatrix}, \dots, x_n = \begin{pmatrix} 0 \\ \vdots \\ 0 \\ 1 \end{pmatrix} \text{ erzeugt werden kann.}$$

**Definition 4**

$z$  heißt *lineare Funktion*, wenn gilt:

$$\begin{aligned} z : \mathbb{R}^n &\rightarrow \mathbb{R} \\ \text{mit } z &= c_1x_1 + \dots + c_nx_n + a = c^T \cdot x + a, \end{aligned}$$

wobei  $c, x \in \mathbb{R}^n$  und  $a \in \mathbb{R}$ .

**Definition 5**

Ein *lineares Ungleichungssystem mit Nichtnegativitätsbedingungen* (NNB) hat folgende Form:

$$\begin{aligned} a_{11}x_1 + a_{12}x_2 + \dots + a_{1n}x_n &\leq b_1 \\ a_{21}x_1 + a_{22}x_2 + \dots + a_{2n}x_n &\leq b_2 \\ \vdots & \\ a_{m1}x_1 + a_{m2}x_2 + \dots + a_{mn}x_n &\leq b_m \\ x_j &\geq 0 \quad (\text{NNB}) \end{aligned} \tag{1}$$

mit den Konstanten:  $a_{ij}, b_i; \quad i \in \{1, \dots, m\}, j \in \{1, \dots, n\}$   
und den Entscheidungsvariablen (EV):  $x_j; \quad j \in \{1, \dots, n\}$ .

Äquivalent gilt:

$$\begin{aligned} Ax &\leq b \\ x &\geq 0 \end{aligned} \tag{2}$$

mit

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

und  $x \in \mathbb{R}^n, b \in \mathbb{R}^m$ .

**Definition 6**

Jeder nichtnegative Vektor  $x$  für den  $Ax \leq b$  gilt heißt *zulässige Lösung* des Systems.

**Definition 7**

Die Menge  $X = \{x : \text{zulässige Lösung}\}$  aller solcher Lösungen heißt *Lösungsbereich*.

Sei  $M \subseteq \mathbb{R}^n$  eine Punktmenge des  $\mathbb{R}^n$ .

$\Rightarrow$  Jeder Punkt  $P_i$  von  $M$  wird durch einen  $n$ -dimensionalen Vektor  $x^{(i)}$  beschrieben.

**Definition 8**

Eine Menge  $M$  heißt *konvex*, genau dann wenn  $\forall P_1, P_2 \in M$  gilt:

Alle durch  $x = \lambda \cdot x^{(1)} + (1 - \lambda) \cdot x^{(2)}$ ,  $0 \leq \lambda \leq 1$  dargestellten Punkte  $P$  gehören zu  $M$ .  $x$  wird *konvexe Linearkombination* von  $x^{(1)}$  und  $x^{(2)}$  genannt. Die Verbindungsstrecke zwischen  $x^{(1)}$  und  $x^{(2)}$  gehört zu  $M$ .

Gilt  $A \cap B$  konvex  $\Rightarrow A \cap B$  konvex (aber  $A \cup B$  nicht notwendig konvex).

**Definition 9**

Ein Punkt  $x$  einer konvexen Menge  $M$  heißt *Extrempunkt* (Ecke) von  $M$ , wenn  $x$  nicht als echte Konvexkombination zweier beliebiger Punkte aus  $M$  darstellbar ist (d.h.  $\lambda = 0 \vee \lambda = 1$ ).

**Definition 10**

Eine konvexe Menge  $M$  heißt *konvexes Polyeder*, wenn sie endlich viele Eckpunkte besitzt.

**Definition 11**

Eine beliebige Ungleichung  $a_{i1}x_1 + \dots + a_{in}x_n \leq b_i$  aus dem Ungleichungssystem (1) definiert einen *konvexen Halbraum* im  $\mathbb{R}^n$  mit begrenzender Hyperebene  $a_{i1}x_1 + \dots + a_{in}x_n = b_i$  (im  $\mathbb{R}^2$  eine Gerade, im  $\mathbb{R}^3$  eine Ebene).

**Satz 1**

Als Durchschnitt von  $m+n$  Halbräumen im  $\mathbb{R}^n$  ist die Lösungsmenge von (2) ein konvexes Polyeder.

## 2.3 Theorie der LP

**Grundmodell (GM):**

$$\begin{aligned} \max \quad & z = c^T x \\ \text{sodass} \quad & Ax \leq b \\ & x \geq 0 \end{aligned} \tag{3}$$

$$c, x \in \mathbb{R}^n, b \in \mathbb{R}^m, A = A_{m,n} \quad (\text{Grundmodell: } b \geq 0)$$

mit:

- $z = c^T x$  : Zielfunktion
- $c$  : Vektor der Zielfunktionskoeffizienten
- $A = A_{m,n}$ : Koeffizientenmatrix
- $b$  : Kapazitätsvektor („rechte Seite“)
- $x \geq 0$  : Nichtnegativitätsbedingung (NNB)

Dies ist bereits eine gewisse Normalform.

### Definition 12 (Normalform)

Die lineare Funktion  $z = z(x_1, \dots, x_{n+m}) = c_1 x_1 + \dots + c_{n+m} x_{n+m}$  ist unter Berücksichtigung der Gleichungen

$$\begin{array}{rcl} a_{11} x_1 + \dots + a_{1n} x_n & + x_{n+1} & = b_1 \\ \vdots & & \\ a_{m1} x_1 + \dots + a_{mn} x_n & + x_{n+m} & = b_m \end{array}$$

zu maximieren.

In Matrixschreibweise:

$$\begin{array}{l} z = c^T x \rightarrow \max, \\ Ax = b, \quad x \geq 0, \quad b \geq 0 \end{array}$$

mit

$$A = (A', I) = \begin{pmatrix} a_{11} & \cdots & a_{1n} & 1 & 0 & \cdots & 0 \\ a_{21} & \cdots & a_{2n} & 0 & 1 & \cdots & 0 \\ \vdots & & \vdots & \vdots & \vdots & & \vdots \\ a_{m1} & \cdots & a_{mn} & 0 & 0 & \cdots & 1 \end{pmatrix}$$

$$\begin{array}{l} x^T = (x_1, \dots, x_{n+m}) \\ c^T = (c_1, \dots, c_{n+m}) \\ b^T = (b_1, \dots, b_m) \end{array}$$

### Satz 2

Jedes lineare Optimierungsproblem (LOP) lässt sich in die angegebene NF überführen.

Dies soll anhand des folgenden Beispiels verdeutlicht werden.

$$\begin{array}{ll} \min & z_1 = -2x_1 + 4x_2 \\ \text{s.d.} & 2x_1 + 3x_2 \leq -1 \\ & x_1 - x_2 \leq 2 \\ & -6x_1 + 2x_2 = -4 \\ & x_1 \geq 0 \end{array}$$

weicht erheblich von der NF ab.

### Schritt 1: Nichtnegativitätsbedingung

Regel: Falls  $\exists i$  mit  $x_i$  kann negativ sein, dann ersetze  $x_i$  durch  $x_i^* - x_i^{**}$  mit  $x_i^*, x_i^{**} \geq 0$ .

Beispiel:

$$x_2 = x_2^* - x_2^{**}$$

$$\begin{aligned} \Rightarrow \min \quad & z_2 = -2x_1 + 4x_2^* - 4x_2^{**} \\ \text{s.d.} \quad & 2x_1 + 3x_2^* - 3x_2^{**} \leq -1 \\ & x_1 - x_2^* + x_2^{**} \leq 2 \\ & -6x_1 + 2x_2^* - 2x_2^{**} = -4 \\ & x_1, x_2^*, x_2^{**} \geq 0 \end{aligned}$$

**Schritt 2: Überführen in eine Maximumsaufgabe**

Regel: Äquivalent zur Minimierung von:  $z = c^T x$   
ist die Maximierung von:  $z' = -z = -c^T x$ .

Beispiel:

$$\Rightarrow \max z_3 = -z_2 = 2x_1 - 4x_2^* + 4x_2^{**}$$

**Schritt 3: Nichtnegative rechte Seite der Nebenbedingungen**

Regel: Multipliziere die Nebenbedingungen, deren rechte Seiten negativ sind mit  $-1$ .

Beispiel:

$$\begin{aligned} \Rightarrow \max \quad & z_3 = 2x_1 - 4x_2^* + 4x_2^{**} \\ \text{s.d.} \quad & -2x_1 - 3x_2^* + 3x_2^{**} \geq 1 \\ & x_1 - x_2^* + x_2^{**} \leq 2 \\ & 6x_1 - 2x_2^* + 2x_2^{**} = 4 \end{aligned}$$

**Schritt 4: Überführung in Gleichungen**

Regel 1: Die Ungleichungsnebenbedingung (UNB)  $a_1x_1 + \dots + a_nx_n \leq b$   
ist äquivalent zu der Gleichungsnebenbedingung (GNB)  $a_1x_1 + \dots + a_nx_n + x_s = b$   
mit  $x_s \geq 0$ .

Regel 2: Die UNB  $a_1x_1 + \dots + a_nx_n \geq b$  ( $b \geq 0$ )  
ist äquivalent zu der GNB  $a_1x_1 + \dots + a_nx_n - x_s = b$  mit  $x_s \geq 0$ .

**Definition 13**

$x_s$  wird *Schlupfvariable* genannt, welche eine nicht vollständig benutzte Ressource, sofern  $> 0$ , darstellt. (Sie gibt an, um wie viel die Ressource nicht ausgenutzt wird.)

Beispiel:

$$\begin{aligned} \Rightarrow \max \quad & z_3 = 2x_1 - 4x_2^* + 4x_2^{**} \\ \text{s.d.} \quad & -2x_1 - 3x_2^* + 3x_2^{**} - x_{s_1} = 1 \\ & x_1 - x_2^* + x_2^{**} + x_{s_2} = 2 \\ & 6x_1 - 2x_2^* + 2x_2^{**} = 4 \\ & x_1, x_2^*, x_2^{**}, x_{s_1}, x_{s_2} \geq 0 \end{aligned}$$

**Schritt 5: Einführung von Hilfsvariablen (künstlichen Variablen)**

Regel: Man führe in die ZF und in die GNB künstliche Variablen  $x_{n+1}, \dots, x_{n+m} \geq 0$  ein  
(wo nicht bereits Beitrag zu  $I$  vorhanden), sodass:

$$\begin{aligned} a_{11}x_1 + \dots + a_{1n}x_n + x_{n+1} & = b_1 \\ \vdots & \\ a_{m1}x_1 + \dots + a_{mn}x_n + x_{n+m} & = b_m \end{aligned}$$

$$z' = c_1x_1 + \dots + c_nx_n - Mx_{n+1} - \dots - Mx_{n+m} \rightarrow \max$$

$M \gg 0$ , alle  $x_i, x_{n+j}, b_j \geq 0$

(Entspricht der NF, falls dort  $c_{n+1} = \dots = c_{n+j} = -M$  gilt.)

Die Lösungen dieses Modells und der NF sind äquivalent, denn für die Optimallösung muss wegen  $M \gg 0$

$$\hat{x}_{n+1} = \dots = \hat{x}_{n+m} = 0$$

gelten.

Beispiel:

$$\begin{aligned} \Rightarrow \max \quad & z_4 = 2x_1 - 4x_2^* + 4x_2^{**} - M \cdot x_{k_1} - M \cdot x_{k_2} \\ \text{s.d.} \quad & -2x_1 - 3x_2^* + 3x_2^{**} - x_{s_1} + x_{k_1} = 1 \\ & x_1 - x_2^* + x_2^{**} + x_{s_2} = 2 \\ & 6x_1 - 2x_2^* + 2x_2^{**} + x_{k_2} = 4 \\ & x_1, x_2^*, x_2^{**}, x_{s_1}, x_{s_2}, x_{k_1}, x_{k_2} \geq 0 \end{aligned}$$

### Schritt 6: Umnummerierung von Variablen

Setze:  $x_1 = x_1$ ,  $x_2 = x_2^*$ ,  $x_3 = x_2^{**}$ ,  $x_4 = x_{s_1}$ ,  $x_5 = x_{k_1}$ ,  $x_6 = x_{s_2}$ ,  $x_7 = x_{k_2}$   
 $\Rightarrow$  Modell ist in NF

Beispiel:

$$\begin{aligned} \max \quad & z = 2x_1 - 4x_2 + 4x_3 - M \cdot x_5 - M \cdot x_7 \\ \text{s.d.} \quad & -2x_1 - 3x_2 + 3x_3 - x_4 + x_5 = 1 \\ & x_1 - x_2 + x_3 + x_6 = 2 \\ & 6x_1 - 2x_2 + 2x_3 + x_7 = 4 \\ & x_i \geq 0 \quad \text{für } i = 1, \dots, 7 \end{aligned}$$

Vorteil: Eine erste Lösung ist sofort ablesbar!

$x_1 = x_2 = x_3 = x_4 = 0$ ;  $x_5 = 1$ ;  $x_6 = 2$ ;  $x_7 = 4$

mit einem extrem schlechten ZF-Wert von  $z = -M \cdot 1 - M \cdot 4 = -5M$ .

Preis für die schnelle erste Lösung: Einführung zusätzlicher Variablen.

### Grundmodell mit GNB:

$$\begin{aligned} \max \quad & z = c^T x + 0^T s \\ \text{sodass} \quad & Ax + Is = b \\ & x, s \geq 0 \end{aligned} \tag{4}$$

$$c, x \in \mathbb{R}^n, b, s \in \mathbb{R}^m, b \geq 0, A = A_{m,n}, I = I_m$$

GM (3) und GM (4) mit Schlupfvariablen sind äquivalent:

Falls  $(\hat{x}, \hat{s})$  optimale Lösung des GM (4) mit Schlupfvariablen ist, dann ist  $\hat{x}$  optimale Lösung des GM (3).

Falls  $\hat{x}$  optimale Lösung des GM (3) ist, dann existiert ein  $\hat{s}$  mit der optimalen Lösung  $(\hat{x}, \hat{s})$  für das GM (4) mit Schlupfvariablen.

### andere Schreibweise des GM (4):

$$\begin{aligned} \max \quad & z = c^T x \\ \text{s.d.} \quad & Dx = b \\ & x \geq 0 \end{aligned} \tag{5}$$

$$x := \begin{pmatrix} x \\ s \end{pmatrix}, \quad c := (c, 0), \quad D = (A, I_m)$$

**Definition 14 (Basis)**

Jede nichtsinguläre  $m \times m$ -Teilmatrix  $B$  (d.h. linear unabhängige Spalten,  $\text{rg}B = m$ ,  $\det B \neq 0$ ) von  $D$  heißt *Basis* des Modells (5).

Bemerkungen:

- Existieren  $m + n$  Spalten, dann existieren höchstens  $\binom{m+n}{m}$  verschiedene Basen, denn für eine Basis müssen aus einer Menge von  $m + n$  Spalten genau  $m$  ausgewählt werden (Ziehen ohne Wiederholung/ohne Reihenfolge). Da nicht alle Kombinationen von Spalten linear unabhängig sein müssen, existieren also höchstens so viele Basen.
- $B = I_m$  ist eine Basis.

**Definition 15 (zulässige Basis)**

Ist  $B$  eine Basis, dann nennt man  $B$  eine *zulässige Basis*, wenn  $B^{-1}b \geq 0$  ist.

**Definition 16 (Basisdarstellung)**

Sei eine Basis  $B$  gewählt. Dann bezeichnet  $N$  den aus den „Nichtbasisvariablen“ bestehenden Rest von Matrix  $D$  und  $x_B, x_N, c_B, c_N$  sind die entsprechend aufgeteilten Vektoren  $x$  und  $c$ .

Das Modell (5) lässt sich jetzt schreiben als:

$$\begin{aligned} \max \quad & z = (c_B^T, c_N^T) \cdot \begin{pmatrix} x_B \\ x_N \end{pmatrix} \\ \text{s.d.} \quad & (B, N) \cdot \begin{pmatrix} x_B \\ x_N \end{pmatrix} = b \\ & x_B, x_N \geq 0 \end{aligned} \tag{6}$$

(Wobei o.B.d.A.  $B$  aus den ersten  $m$  Spalten von  $D$  bestehe.)

Multipliziert man nun das lineare Gleichungssystem mit der Basisinversen (die bei zulässiger Basis existiert):

$$\begin{aligned} & B^{-1}(B, N) \begin{pmatrix} x_B \\ x_N \end{pmatrix} = B^{-1}b \\ \Leftrightarrow & (I_m, B^{-1}N) \begin{pmatrix} x_B \\ x_N \end{pmatrix} = B^{-1}b \\ \Leftrightarrow & I_m x_B + B^{-1}N x_N = B^{-1}b \\ \Rightarrow & x_B = B^{-1}b - B^{-1}N x_N \end{aligned}$$

so entsteht die *Basisdarstellung* (BD) aller Lösungen, die  $Dx = b$  genügen (5), bezogen auf die gewählte Basis  $B$ .

Wird die Abhängigkeit des Vektors  $x_B$  von  $x_N$  untersucht, so schreibt man  $x_B = x_B(x_N)$ .

Komponentenschreibweise:  $x_{B_i} = b_i^* - \sum_{j=m+1}^{m+n} a_{ij}^* x_j, \quad i = 1, \dots, m$

mit:  $b_i^*$   $i$ -ten Komponente des Vektors  $b^* = B^{-1}b$ , sowie  $a_{ij}^*$  Elemente der  $i$ -ten Zeile und  $j$ -ten Spalte von  $B^{-1}N$ .

**Definition 17 ((zulässige) Basislösung)**

Der Vektor  $x = \begin{pmatrix} x_B(0) \\ 0 \end{pmatrix} = \begin{pmatrix} B^{-1}b \\ 0 \end{pmatrix} =: x_B$

heißt *Basislösung* (BL) des Modells (5) bzgl.  $B$  (alle Nichtbasisvariablen sind 0).

Gilt  $x_B = x_B(0) = B^{-1}b \geq 0$ , d.h. ist die Basislösung nichtnegativ, so heißt  $x_B$  *zulässige Basislösung* (ZBL) des Modells (5).

**Satz 3**

Eine zulässige Lösung  $x$  von  $Dx = b$ ,  $x \geq 0$  ist genau dann eine Ecke des Lösungsbereichs (LB), wenn  $x$  ZBL ist.

**Satz 4 (Simplextheorem)**

Ein LOP  $Dx = b$ ,  $x \geq 0$ ,  $\max z = c^T x$  nimmt, sofern es lösbar ist, sein Optimum bei einer ZBL (also einer Ecke) an.

Bemerkungen:

- Wenn eine optimale Lösung existiert, dann existiert eine Ecke des LB, die optimale Lösung ist.  
 $\Rightarrow$  Es müssen nur endlich viele ZBL überprüft werden.
- Existiert eine ZBL mit weniger als  $m$  BV, so heißt diese (oder auch die zugehörige Ecke) entartet (degeneriert). (Einige BV sind 0). Beispiel im  $\mathbb{R}^2$ : Ein Punkt, in dem sich mehr als zwei restriktive Geraden schneiden.
- Man kann jedes LOP in ein nichtentartetes LOP überführen.
- Sind  $x^{(1)}$  und  $x^{(2)}$  optimale Lösungen, dann auch alle Punkte auf der konvexen Hülle (im  $\mathbb{R}^2$  Geraden).
- Nicht alle optimalen Lösungen müssen ZBL sein (wenn eine Restriktion parallel zur Zielfunktion verläuft).

**2.4 Die Simplexmethode — Rechenverfahren**

Das Simplexverfahren besteht im Prinzip darin, eine Folge von Ecken des LB (mit steigendem ZF-Wert) zu konstruieren, wobei jede Ecke durch eine Menge linear unabhängiger Spaltenvektoren beschrieben wird. Zu jeder solchen Menge linear unabhängiger Spaltenvektoren gehört eine Lösung des Gleichungssystems  $Dx = b$ .

Vorraussetzung: Es existiert eine ZBL des Modells (5).

Grundidee:

- Tauschen von BV gegen NBV (Basistausch). Erzeugen einer endlichen Folge von ZBL (wobei sich der ZF-Wert jeweils nicht verschlechtert).
- Dabei gelten folgende Regeln
  - Aufnahme-Regel: Zur Bestimmung der neu in die Basis aufzunehmenden Variablen mit verbessertem ZF-Wert.



- Eliminations-Regel: Zur Bestimmung der aus der Basis zu eliminierenden Variablen, sodass die neue BL zulässig ist.

Warum wird ein Basistausch vorgenommen? Man will eine benachbarte Ecke des LB finden, dessen ZF-Wert größer ist. Dazu wird eine neue Basisvariable und eine neue Nichtbasisvariable gewählt. D.h. einer der bisher nicht in der Basis befindlichen Spaltenvektoren der Matrix  $D$  wird neu in die Basis aufgenommen und dafür einer der bisherigen Basisvektoren aus der Basis entfernt. Wenn die neue Basisvariable von 0 an wächst, ändern auch die bisherigen Basisvariablen ihre Werte. Eine Umformung ist notwendig, damit jede Gleichung genau eine der Basisvariablen enthält (mit dem Koeffizienten 1), und diese Variable in keiner anderen Gleichung auftritt.

### 2.4.1 Ausgangsbasislösung

Für das bekannte Modell (5) in der Schreibweise

$$\begin{aligned} Ax + I_m s &= b \\ x, s &\geq 0 \\ \text{mit } b &\geq 0 \end{aligned}$$

wähle die Spalten der Schlupfvariablen als Ausgangsbasis  $B$ , d.h.  $I_m$  als Ausgangsbasis  $B$   
 $\Rightarrow$  ZBL:  $(x^T, s^T) = (0^T, b^T)$

### 2.4.2 Aufnahme-Regel

Sei  $B$  die vorliegende Basis und  $B'$  die neue Basis.

- Es soll gelten:  $z = c_B^T \cdot x_B < c_{B'}^T \cdot x_{B'}$  (ZF-Wert soll sich vergrößern)  
 mit Basisdarstellung:  $x_B = B^{-1}b - B^{-1}N \cdot x_N$   
 $\Rightarrow$  (Einsetzen in die ZF):

$$\begin{aligned} z &= (c_B^T, c_N^T) \cdot \begin{pmatrix} x_B \\ x_N \end{pmatrix} \\ &= c_B^T x_B + c_N^T x_N \\ &= c_B^T B^{-1}b - c_B^T B^{-1}N x_N + c_N^T x_N \\ &= c_B^T B^{-1}b - (-c_N^T + c_B^T B^{-1}N) \cdot x_N \end{aligned}$$

- In Komponentenschreibweise:

$$\begin{aligned} z &= \underbrace{\sum_{i=1}^m c_{B_i} \cdot b_i^*}_{z^*} - \sum_{j=m+1}^{m+n} \underbrace{\left( -c_j + \sum_{i=1}^m c_{B_i} \cdot a_{ij}^* \right)}_{\Delta z_j} \cdot x_j \\ &= z^* - \sum_{j=m+1}^{m+n} \Delta z_j \cdot x_j \end{aligned}$$

wobei  $z^*$  = ZF-Wert der aktuellen BL  $x_B$  und  
 $\Delta z_j$  = „Kriteriumselement“ (reduzierten Kosten)  
 der bisherigen NBV  $x_j$ ,  $j = m + 1, \dots, m + n$ .

- $\Delta z_j$  gibt an, um wie viel sich  $z$  verschlechtert, wenn  $x_j$  um 1 erhöht wird (falls  $\Delta z_j > 0$ ).

- Folglich ist eine Verbesserung des ZF-Wertes  $z$  nur für solche NBV möglich, für die  $\Delta z_j < 0$  ist.
- Existieren mehrere NBV  $x_j$  mit  $\Delta z_j < 0$ , dann wähle diejenige, die die maximale Verbesserung verspricht (Heuristik):

$$\Delta z_l = \min_j \{ \Delta z_j \mid \Delta z_j < 0 \}$$

(Dies muss nicht zwangsläufig die beste Verbesserung sein, da  $z$  auch noch von  $x_j$  abhängt.)

$\Rightarrow$  neue Basis  $B'$ :  $m - 1$  Spalten von  $B$  und bisherige Nichtbasisspalte, die zu  $x_l$  gehört.

### 2.4.3 Eliminationsregel

Auf welche Spalte der alten Basis wird verzichtet?

Sei  $B$  nichtentartete zulässige Basis von Modell (5).

O.B.d.A. enthalte  $B$  die erste  $m$  Spalten von  $A$ .

$$\Rightarrow x_{B_i} = b_i^* - \sum_{j=m+1}^{m+n} a_{ij}^* \cdot x_j, \quad i = 1, \dots, m$$

Sei  $x_l$  die NBV, die Basis werden soll:

$$x_l = \Theta > 0 \text{ und } x_j = 0 \quad \forall j \neq l \text{ (alle anderen NBV)}$$

$$\Rightarrow x_{B_i} = b_i^* - a_{il}^* \Theta, \quad i = 1, \dots, m$$

Umrechnung der  $x_{B_i}$  (bisher =  $b_i^*$  in ZBL).

Aus den NNB  $\Rightarrow x_{B_i} = b_i^* - a_{il}^* \Theta \geq 0$ .

- Ist  $a_{il}^* \leq 0$  kann für das betreffende  $i$  bzgl. der NNB nicht passieren. (Dann kann aber  $\Theta$  beliebig groß sein  $\Rightarrow x_{B_i}$  ist nicht beschränkt.)
- Für  $i$  mit  $a_{il}^* > 0$  muss  $0 \leq \frac{b_i^*}{a_{il}^*}$  gelten.
- Sind alle  $a_{il}^* \leq 0$ , so existiert keine optimale Lösung ( $\Theta$  ist dann beliebig groß, ohne dass Zulässigkeitsprobleme entstehen).
- Falls  $a_{il}^* > 0 \Rightarrow$  Es existiert ein  $i_0 \in \{1, \dots, m\}$   
mit

$$\Theta := \min_i \left\{ \frac{b_i^*}{a_{il}^*} \mid i = 1, \dots, m, a_{il}^* > 0 \right\} = \frac{b_{i_0}^*}{a_{i_0 l}^*}$$

(Somit hat  $\Theta$  den größtmöglichen Wert.)

$$\text{Für } i = i_0 \text{ gilt dann: } x_{i_0} = \overbrace{b_{i_0}^* - a_{i_0 l}^* \cdot \Theta}^{b_{i_0}^*} = 0$$

D.h. die bisherige BV  $x_{i_0}$  hat nun den Wert 0 und fällt aus der Basis raus.

Hierbei müssen zwei Fälle unterschieden werden:

1.  $i_0$  ist eindeutig  $\Rightarrow$  Index der zu eliminierenden BV ist eindeutig bestimmt. Die nächsten BV sind zulässig.

2.  $i_0$  ist nicht eindeutig  $\Rightarrow$  Index  $i_0$  zufällig wählen (wähle den kleinsten Wert (Kleinste-Kosten-Regel) bzw. bei mehreren unter diesen dasjenige mit dem kleinsten Index (Kleinste-Index-Regel)). In der nächsten ZBL gibt es mindestens eine BV = 0 (da  $i_0$  nicht eindeutig ist und für die BV  $x_{i_0} = b_{i_0} - a_{i_0 l}^* \cdot \Theta = 0$  gilt). Es handelt sich um eine entartete (degenerierte) BL, evtl. wird der ZF-Wert nicht besser sondern bleibt gleich. (In diesem Fall nach der Kleinste-Index-Regel vorgehen, um Zyklen zu vermeiden.)

### 2.4.4 Umformung

Umformung von  $(I_m, B^{-1}N) \begin{pmatrix} x_B \\ x_N \end{pmatrix} = b^* [= B^{-1}b]$  mittels Simplextableau (Basistausch).

#### 1. Basistausch

- Zeile der zu eliminierenden Variablen  $k$ : Pivotzeile
- Spalte der aufzunehmenden Variablen  $l$ : Pivotspalte
- $a_{kl}^*$ : Pivotelement  
 $a_{ij}^*$ : Element der  $i$ -ten Zeile und der  $j$ -ten Spalte  
 $a_{il}^*$ :  $i$ -tes Element der Pivotspalte  
 $a_{kj}^*$ :  $j$ -tes Element der Pivotzeile  
 $b_i^*$ :  $i$ -tes Element der rechten Seite

	$c_j$	$c_1 \cdots c_k \cdots c_m$	$c_{m+1} \cdots c_l \cdots c_{m+n}$		
$c_{B_i}$	$x_{B_i} \setminus x_j$	$x_1 \cdots x_k \cdots x_m$	$x_{m+1} \cdots x_l \cdots x_{m+n}$	$b_i^*$	$\Theta_i$
$c_1$	$x_1$	1	$a_{1,m+1}^* \cdots a_{1,l}^* \cdots a_{1,m+n}^*$	$b_1^*$	$\frac{b_1^*}{a_{1,l}^*}$
$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\vdots$	$\vdots$
$c_k$	$x_k$	1	$a_{k,m+1}^* \cdots a_{k,l}^* \cdots a_{k,m+n}^*$	$b_k^*$	$\frac{b_k^*}{a_{k,l}^*}$
$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\vdots$	$\vdots$
$c_m$	$x_m$	1	$a_{m,m+1}^* \cdots a_{m,l}^* \cdots a_{m,m+n}^*$	$b_m^*$	$\frac{b_m^*}{a_{m,l}^*}$
	$\Delta z_j$		$\Delta z_{m+1} \cdots \Delta z_l \cdots \Delta z_{m+n}$	$z^*$	

Vorteil: In der Kopfzeile müssen keine Variablen ausgetauscht werden. Basisvariablen erkennt man an der darunterliegenden Spalte  $(0, 0, 1, 0, \dots, 0)^T$ .

Nachteil: Das Tableau ist unnötig groß.

#### 2. Transformation der Elemente des Tableaus

1. Dividiere die Pivotzeile durch das Pivotelement. (Dadurch ist die  $x_k$ -Spalte nicht mehr Basisspalte.)
2. Pivotspalte wird im neuen Tableau Einheitsvektor  $e_k$ .
3. Elementweise Transformation (Kreisregel) oder Zeilentransformation (Pivotisieren).

In Formeln:

$$\begin{aligned}\bar{a}_{ij}^* &= a_{ij}^* - \frac{a_{kj}^*}{a_{kl}^*} \cdot a_{il}^*, & i = 1, \dots, m; i \neq k, & \quad j = 1, \dots, m+n; j \neq l \\ \bar{b}_i^* &= b_i^* - \frac{a_{il}^*}{a_{kl}^*} \cdot b_k^*, & i = 1, \dots, m; i \neq k \\ \Delta \bar{z}_j &= \Delta z_j - \frac{a_{kj}^*}{a_{kl}^*} \cdot \Delta z_l, & j = 1, \dots, m+n; j \neq l \\ \bar{z}^* &= z^* - \frac{b_k^*}{a_{kl}^*} \cdot \Delta z_l\end{aligned}$$

### 3. Abbruch-Simplexkriterium

Eine optimale BL (optimale Lösung) ist gefunden, wenn für alle NBV gilt:

$$\Delta z_j \geq 0, \quad j = m+1, \dots, m+n.$$

## 2.5 Erweiterungen (Zweiphasenwechsel)

Bekanntes Modell:

$$\begin{aligned}\max \quad & z = c^T x \quad c, x \in \mathbb{R}^n \\ \text{s.d.} \quad & Ax = b \quad b \in \mathbb{R}^m, b \geq 0 \\ & x \geq 0\end{aligned} \tag{7}$$

Wie sieht die Ausgangsbasislösung aus, wenn  $A$  keine Einheitsmatrix als Teilmatrix enthält, weil z.B.  $\geq$ -Nebenbedingung zu einer Schlupfvariablen mit negativem Vorzeichen geführt hat.

Ergänze Modell (7) durch:

$$\begin{aligned}\max \quad & z = (c^T, c_H^T) \cdot \begin{pmatrix} x \\ x_H \end{pmatrix} \\ \text{s.d.} \quad & (A, I_m) \cdot \begin{pmatrix} x \\ x_H \end{pmatrix} = b \\ & x, x_H, b \geq 0 \\ & c, x \in \mathbb{R}^n; c_H, x_H \in \mathbb{R}^m\end{aligned} \tag{8}$$

$x_H$  heißt *Hilfs-* (bzw. *künstliche*) *Variable*.

Eigenschaften:

- Die Modelle sind identisch, wenn in Modell (8)  $x_H = 0$  ist.
- Vorteil: Erste ZBL (d.h. Ausgangslösung)  $x_H = b \geq 0$ .
- Anforderungen an den Algorithmus:
  - $x_H$  eliminieren
  - Sicherstellen, dass keine Komponente von  $x_H$  in eine Basis aufgenommen werden kann (außer in Ausgangsbasis) nachdem sie eliminiert wurde.  
Deshalb:  $c_H = -M$  mit  $M \gg \max_j c_j$  wählen!
- Gelingt es nicht  $x_H = 0$  durch Elimination der  $x_{H_i}$  aus der Basis zu erreichen, dann existiert keine zulässige Lösung des Modells (7) (d.h. die Restriktionen sind widersprüchlich), falls keine Entartung vorliegt.

Abschließende Hinweise:

- Verbessert sich der ZF-Wert nicht  $\Rightarrow$  Entartung. Dabei führt der Austauschschritt nur zu einer anderen Basis der selben entarteten Ecke (mehr als zwei Restriktionen schneiden sich in einem Punkt (im  $\mathbb{R}^2$ )).
- Ist der Wert  $b_i^*$  einer Schlupfvariablen im optimalen Endtableau  $> 0$ , so wird die entsprechende Restriktion nicht voll ausgeschöpft.
- Eine additive Konstante in der ZF muss bei der Lösung der Optimierungsaufgabe nicht berücksichtigt werden (nur bei der abschließenden Angabe des optimalen ZF-Wertes).

### 3 Dualität in der LP

#### 3.1 Duales Modell

Aussage über ein Paar von Modellen, deren Beziehungen zueinander in eindeutiger Weise definiert sind,

- um alternative Algorithmen für LP-Modelle zu konstruieren,
- zur Verringerung des Lösungsaufwandes,
- zur Interpretation gewisser Eigenschaften von LP-Modellen und deren optimalen Endtableaus (direkte Anwendung in Abschnitt 5.2).

#### Definition 18

(Grundmodell ohne  $b \geq 0$  als Basis)

Für das *Primale LP-Modell*:

$$\begin{aligned} \max \quad & z = c^T x \quad c, x \in \mathbb{R}^n \\ \text{s.d.} \quad & Ax \leq b, \quad A = A_{m,n}, b \in \mathbb{R}^m \\ & x \geq 0 \end{aligned}$$

lautet das *Duale LP-Modell*:

$$\begin{aligned} \min \quad & Z = b^T y, \quad b, y \in \mathbb{R}^m \\ \text{s.d.} \quad & A^T y \geq c, \quad A^T = A_{n,m}^T, c \in \mathbb{R}^n \\ & y \geq 0 \end{aligned}$$

Zuordnungsregeln vom dualen zum primalen Modell:

1. Die  $n$ -dimensionalen Vektoren  $x$  des primalen Lösungsbereichs werden in die  $m$ -dimensionalen Vektoren  $y$  des dualen Lösungsbereichs überführt. (Nach der Transformation ist aufgrund der Dimensionsänderung evtl. auch eine graphische Lösung möglich.)
2. Der Zielvektor  $c$  des primalen Modells wird zum Kapazitätsvektor des dualen Modells.
3. Der Kapazitätsvektor  $b$  der primalen Modells wird zum Zielvektor des dualen Modells.

4. Bei primaler Maximierung hat man die dualen ZF zu minimieren.
5. Die transponierte primale Koeffizientenmatrix ist die Koeffizientenmatrix des dualen Problems.
6. Bei  $\leq$ -Beschränkungen des primalen Modells sind die dualen Variablen vorzeichenbeschränkt.
7. Bei primalen Gleichbedingungen sind die entsprechenden dualen Variablen vorzeichenunbeschränkt (d.h. es gibt keine NNB).

Es gelten die folgenden Eigenschaften:

**Satz 5 (Dualität „zueinander“)**

Das dualen Modell (DM) des dualen Modells ist das primale Modell (PM).

Beweis:

$$\begin{array}{llll}
 \text{(PM)} \quad \max & z = c^T x & \text{Dualisieren} & \text{(DM)} \quad \min & Z = b^T y \\
 \text{s.d.} & Ax \leq b & \longrightarrow & \text{s.d.} & A^T y \geq c \\
 & x \geq 0 & & & y \geq 0 \\
 \uparrow \text{NF} & & & \downarrow \text{NF} & \\
 \min & -z = -c^T x & \text{Dualisieren} & \max & -Z = -b^T y \\
 \text{s.d.} & (-A^T)^T x \geq -b & \longleftarrow & \text{s.d.} & -A^T y \leq -c \\
 & x \geq 0 & & & y \geq 0
 \end{array}$$

□

**Satz 6 (Schwache Dualität)**

Ist  $x$  zulässige Lösung für das PM und ist  $y$  zulässige Lösung für das DM, dann gilt:

$$z = c^T x \leq b^T y = Z \quad (z(x) \leq Z(y))$$

Beweis:

$$\begin{array}{l}
 \text{Es gilt } Ax \leq b, x \geq 0 \\
 A^T y \geq c, y \geq 0 \\
 (1) \quad x^T A^T y \geq x^T c = c^T x = z \\
 (2) \quad Ax \leq b \quad \Leftrightarrow (Ax)^T = x^T A^T \leq b \\
 (3) \quad x^T A^T y \leq b^T y = Z \quad \Rightarrow z = c^T x \leq x^T A^T y \leq b^T y = Z
 \end{array}$$

□

**Satz 7 (Starke Dualität)**

Gilt für eine Paar  $(x_0, y_0)$  zulässiger Lösungen des PM bzw. DM die Gleichung:

$$c^T x_0 = b^T y_0$$

so ist  $(x_0, y_0)$  ein Paar optimaler Lösungen des PM bzw. DM, d.h.

$$\begin{array}{l}
 c^T x_0 = \max \{c^T x \mid Ax \leq b, x \geq 0\} \\
 b^T y_0 = \min \{b^T y \mid A^T y \geq c, y \geq 0\}
 \end{array}$$

Die Umkehrung gilt ebenfalls, d.h. für ein Paar optimaler Lösungen gilt  $c^T x_0 = b^T y_0$ .

Beweis:

Da  $y_0$  zulässige Lösung ist  $\Rightarrow$  Satz 6

$$c^T x \leq b^T y_0 \stackrel{\text{Vor.}}{=} c^T x_0 \Rightarrow c^T x_0 = \max\{\dots\}$$

Da  $x_0$  zulässige Lösung ist  $\Rightarrow$  Satz 6

$$b^T y \geq c^T x_0 \stackrel{\text{Vor.}}{=} b^T y_0 \Rightarrow b^T y_0 = \min\{\dots\} \quad \square$$

**Satz 8 (Complementary-Slackness-Relation, Dualitätssatz der LP)**

Seien  $x_s, y_s$  die Vektoren der Schlupfvariablen zu den Lösungsvektoren  $x, y$  und seien  $(x, x_s), (y, y_s)$  zulässige Lösungen des dualen Paares aus PM und DM.

Dann gilt  $(x, x_s), (y, y_s)$  sind genau dann optimale Lösungen, wenn die Komplementaritätsbedingung

$$\begin{aligned} x^T y_s + x_s^T y &= 0 \\ \text{also } (x^T = 0 \vee y_s = 0) \wedge (x_s^T = 0 \vee y = 0) \end{aligned}$$

gilt.

Bemerkung: Ist  $x$  eine ZBL (d.h.  $x_s = 0$ ,  $x$  besteht nur aus Strukturvariablen), dann gilt  $x^T y_s = 0$ . Ist  $y$  eine ZBL (d.h.  $y_s = 0$ ), dann gilt  $x_s^T y = 0$ . Falls eine primale Schlupfvariable Schlupf hat ( $b_i^* > 0$ ) dann hat die zugehörige duale Strukturvariable Schlupf 0.

Beweis:

$(x, x_s), (y, y_s)$  sind zulässige Lösungen

$$\begin{aligned} \Rightarrow \quad Ax + I_m x_s &= b, & x, x_s &\geq 0 \\ \wedge \quad A^T y - I_n y_s &= c, & y, y_s &\geq 0 \\ \Leftrightarrow \quad x^T A^T + x_s^T I_m &= b^T \\ \wedge \quad y^T A - y_s^T I_n &= c^T \\ \Leftrightarrow \quad x^T A^T y - x_s^T I_m y &= b^T y \quad (1) \\ \wedge \quad y^T A x - y_s^T I_n x &= c^T x \quad (2) \end{aligned}$$

Da  $x^T A^T y = y^T A x \stackrel{(1)-(2)}{\Rightarrow} x_s^T y + y_s^T x = b^T y - c^T x \stackrel{\text{Satz 7}}{\Rightarrow} x_s^T y + y_s^T x = 0$  gdw.  $b^T y = c^T x$   
 $\Rightarrow (x, y)$  ist ein Paar optimaler Lösungen.  $\square$

**Satz 9**

Besitzt eines der zueinander dualen Modelle PM und DM eine optimale Lösung, so hat auch das dazu duale Modell eine optimale Lösung und die optimalen ZF-Werte sind gleich.

**Satz 10**

Hat das PM keine zulässige Lösung, so hat das DM keine optimale Lösung (und umgekehrt).

**Satz 11**

Im primalen optimalen Endtableau sind die  $\Delta z_j$  unter den primalen Schlupfvariablen die Werte der Strukturvariablen der optimalen dualen Lösung. Die entsprechenden Werte der dualen Schlupfvariablen finden sich in der  $\Delta z_j$ -Zeile unter den primalen Strukturvariablen. (D.h. aus dem primalen Endtableau ist die optimale duale Lösung ablesbar, siehe Beispiel in [1])

Bemerkungen:

- Ein Endtableau ist optimal, wenn es sowohl primal als auch dual zulässig ist.

- Wenn eines zweier zueinander dualen LOP wesentlich einfacher als das andere zu lösen ist, löse dieses mit dem Simplexverfahren und bestimme die Lösung des anderen durch Ablesen.
- Das DM hat genau dann eine optimale Lösung, wenn das PM eine nichtentartete optimale Lösung besitzt.

### 3.2 Ökonomische Interpretation der dualen Lösung

#### (1) Dimensionsbetrachtung am Beispiel Produktionsplanung

1. Dimension der ZF-Koeffizienten  $c_i$  sei €/Stk.
2. Dimension der EV  $x_i$  sei Stk.
3. Dimension der verfügbare Kapazitäten  $b_i$  sei MStd.
4. Dimension der technologischen Koeffizienten  $a_{ij}$  sei MStd./Stk.  
(Zur Herstellung eines Stk. von Produkt  $j$  sind  $a_{ij}$  MStd. der Maschine  $i$  nötig.)

Wegen  $A^T y \geq c$  gilt für die Dimension der dualen Variable  $y_i$

$$[y_i] = \left[ \frac{c_j}{a_{ij}} \right] = \left[ \frac{\text{€/Stk.}}{\text{MStd./Stk.}} \right] = \left[ \frac{\text{€}}{\text{MStd.}} \right]$$

- (2) Duale Strukturvariablen werden auch Schattenpreise oder Opportunitätskosten genannt. Sie geben an, um wie viel sich der optimale ZF-Wert ändert, wenn diejenige Komponente des  $b$ -Vektors marginal (d.h. um eine Einheit) geändert wird, der die entsprechende duale Strukturvariable zugeordnet ist. (Z.B. „Wert“ einer ME eines Produktionsfaktors oder „Preis“ den das Unternehmen bereit ist für eine ME eines Produktionsfaktors zu zahlen.)

Beispiel:

$$\begin{aligned} a_{i1}x_1 + \dots + a_{in}x_n &\leq b_i \\ y_i \quad i &= 1, \dots, m \end{aligned}$$

Im dualen Endtableau steht z.B.  $y_i = 6$ , d.h. wenn  $b_i$  von 80 auf 81 geändert wird, so steigt der optimale ZF-Wert von 300 GE auf 306 GE.

- Interpretation beruht auf der Annahme, die Knappheit der jeweils zugeordneten Restriktion (Ressource) anzugeben.
- Je größer der Schattenpreis, desto „knapper“ die Ressource, d.h. desto mehr könnte der Optimalwert  $z^*$  unter Berücksichtigung aller Zusammenhänge durch Lockerung der Restriktion erhöht werden.
- Ist der duale Preis = 0, so ist durch eine Lockerung der Restriktion  $y_i = 0$  eine ZF-Verbesserung nicht möglich. Die durch die entsprechende Restriktion modellierte Ressource ist kein Engpass.
- Die primale Restriktion wird von der optimalen primalen Lösung nicht als Gleichung erfüllt und ihre Schlupfvariable ist demzufolge positiv.

Beachte: Der Schattenpreis bleibt nur solange gültig, wie durch Änderung der Komponente  $b$  im PM die Basis ihre primale und duale Zulässigkeit behält.



## (3) Komplementärer Schlupf

Wird die verfügbare Menge eines Faktor nicht ausgenutzt (<-Zeichen), so bringt eine Erhöhung der verfügbaren Menge keinen zusätzlichen Gewinn  $\Rightarrow$  Schattenpreis = 0 (entsprechendes  $y_i = 0$ ).

Ist der zusätzliche Gewinn  $> 0$ , dann ist der entsprechende Faktor knapp, wird also bei optimaler Planung vollständig verbraucht.

Für ein optimales Lösungspaar gilt:

$$(\text{primale Schlupfvariable}) \cdot (\text{entspr. duale Strukturvariable}) = 0$$

Also ist entweder die primale Schlupfvariable = 0 (Ressourcen-Engpass) oder der Schattenpreis = 0 (kein Ressourcen-Engpass) oder beide sind null.

### 3.3 Dualer Simplexalgorithmus

Der duale Simplexalgorithmus ist nützlich für weitergehende Problemstellungen, wie z.B. Ganzzahligkeitsbedingungen oder Binär-Bedingungen. Die „duale“ Vorgehensweise beim dualen Simplexalgorithmus drückt sich darin aus, dass Zeilen- und Spaltenoperationen vertauscht werden. Beim dualen Simplex wird zuerst die Pivotzeile und anschließend die Pivotspalte bestimmt, das Pivotelement ist stets negativ.

#### 3.3.1 Ausgangsbasislösung

Starte mit einer dual zulässigen und primal unzulässigen Lösung, d.h. einer Basislösung für die gilt:

$$\Delta z_j \geq 0, \quad j = 1, \dots, m + n \quad (9)$$

und es gibt Komponenten des Lösungsvektors mit

$$x_{B_i} = b_i^* < 0 \quad (\text{rechte Seite teilweise negativ}) \quad (10)$$

#### 3.3.2 Eliminationsregel

Aus der Menge der dual zulässigen Lösungen ist eine primal zulässige zu bestimmen. Also ist eine BV zu eliminieren, für die (10) gilt.

Trifft dies auf mehrere  $i$  zu, so wähle eine Variable, für die

$$x_k = \min \{b_i^* | b_i^* < 0\} \quad (\text{Minimum der negativen } b_i^*)$$

gilt.

#### 3.3.3 Aufnahme-regel

Duale Zulässigkeit soll bei der Suche nach einer primal zulässige BL erhalten bleiben, d.h. (9) soll auch für die nächste BL gelten.

- Da in der  $k$ -ten Zeile (Pivotzeile) nach Umrechnung (Pivotisierung) nur dann ein  $b_k^* > 0$  entstehen kann, wenn das Pivotelement  $a_{kl}^* < 0$  ist  $\Rightarrow$  Ein dualer Simplexschritt kann nur durchgeführt werden, wenn es in der Pivotzeile negative Koeffizienten  $a_{kj}^*$  gibt.
- Andernfalls: Abbruch, da es keine primal und dual zulässige (also nach Satz 11 keine optimale) Lösung gibt.

- Die aufzunehmende Variable  $x_l$  wird so bestimmt, dass

$$\zeta := \frac{\Delta z_l}{a_{kl}^*} = \max \left\{ \frac{\Delta z_j}{a_{kj}^*} \mid j = 1, \dots, +n, a_{kj}^* < 0 \right\}$$

Begründung:  $\Delta \bar{z}_j = \Delta z_j - \frac{a_{kj}^*}{a_{kl}^*} \cdot \Delta z_l$ ,  $j = 1, \dots, m+n$ ,  $j \neq l$

Wegen  $a_{kl}^* < 0$  und  $\Delta z_j, \Delta z_l \geq 0$  gilt  $\Delta \bar{z}_j \geq 0$  gdw.

$$\frac{\Delta z_j}{a_{kj}^*} \leq \frac{\Delta z_l}{a_{kl}^*} \text{ für } j = 1, \dots, m+n, a_{kj}^* < 0$$

### 3.3.4 Abbruchregel (Optimalitätskriterium)

Der Algorithmus bricht ab, wenn

- primale Zulässigkeit erreicht ist (alle  $b_i \geq 0$ ) oder
- keine weiteren Iterationen existieren, da für alle  $j = 1, \dots, m+n$   $a_{kj}^* \geq 0$  ist.

## 4 Postoptimale Analysen

Was ist, wenn die Daten des Modells nur zum Teil bekannt bzw. nicht unveränderlich sind (z.B. Messfehler, Plandaten die auf Schätzung beruhen)? Zunächst werden die Daten als fix angesehen und eine optimale Lösung berechnet. Anschließend (ex post) wird untersucht, wie sich die optimale Lösung gegenüber veränderten Eingabedaten verhält.

### 4.1 Sensitivitätsanalyse

Es werden nur solche Änderungen der Daten berücksichtigt, bei denen sich die optimale Lösung „qualitativ“ nicht ändert, d.h. trotz modifizierter Daten bleibt die Basis der optimalen Lösung (und damit die optimale Ecke) gleich. Die optimale Lösung und der zugehörige ZF-Wert (zulässiger Bereich und Höhenlinien der ZF) können sich jedoch ändern.

Bisher waren  $c, b, A$  fest.

Wie dürfen diese Größen um ihren angegebenen Wert schwanken, ohne dass die bekannte optimale Lösung ihre Gültigkeit verliert? Es können primale oder duale Zulässigkeit einer BL verletzt werden.

- Duale Zulässigkeit besteht, falls  $\Delta z_j \geq 0$  für  $j = 1, \dots, m+n$ .
- Primale Zulässigkeit besteht, falls  $x_B(0) = B^{-1}b \geq 0$ .

Betrachte folgendes Beispiel:

Veränderung von  $c$ :

$c_1 = 12$  wird ersetzt durch  $c'_1 = c_1 + \lambda$

Die  $\Delta z_j$ -Zeile der Simplextableaus hat dann folgende Gestalt:

	$x_1$	$x_2$	$x_3$	$x_4$	$x_5$	$x_6$	$x_7$	$x_8$
$\Delta z_j$				$(\frac{12}{7} - \frac{\lambda}{7})$			$M$	$(\frac{20}{7} + M + \frac{3}{7}\lambda)$

- Da  $M \gg 0 \Rightarrow \Delta z_8 > 0 \quad \forall \lambda$
- $\Delta z_4 = \frac{12}{7} - \frac{\lambda}{7} \geq 0 \Leftrightarrow \lambda \leq 12$

D.h. für  $\lambda > 12$  ist die duale Zulässigkeit verletzt.

$\Rightarrow$  Basis mit den BV  $x_1, x_2, x_3, x_5, x_6$  ist primal und dual zulässig für  $-\infty < \lambda \leq 12$ ,  
d.h.  $c'_1 = 12 + \lambda \leq 24$  (für jede Zahl  $\leq 24$  bleibt die Lösung optimal).

Entsprechende Analysen sind für jeden ZF-Koeffizienten durchführbar.

Eine Änderung der ZF-Koeffizienten bewirkt im  $\mathbb{R}^2$  eine Änderung der Steigung der Höhenlinien.

Veränderung von  $b$ :

- Duale Zulässigkeit wird nicht beeinflusst.
- Primale Zulässigkeit wird verletzt, wenn die Komponente von  $x_B(0) = B^{-1}b$  negativ wird.

$b_2$  wird ersetzt durch  $b'_2 = b_2 + \lambda$

Die Basislösung ergibt sich aus  $x_B(0) = B^{-1}b'_2 =$

$$\begin{bmatrix} 1 & 2 & 0 & 0 & -\frac{8}{3} \\ 0 & \frac{3}{7} & 0 & 0 & -\frac{2}{7} \\ 0 & \frac{2}{7} & 1 & 0 & -\frac{13}{7} \\ 0 & -\frac{1}{7} & 0 & 0 & \frac{2}{7} \\ 0 & \frac{2}{7} & 0 & -1 & \frac{1}{7} \end{bmatrix} \cdot \begin{bmatrix} 80 \\ 100 + \lambda \\ 75 \\ 10 \\ 45 \end{bmatrix} = \begin{array}{l} 160 + 2\lambda < 0 \quad \text{für } \lambda < -80 \\ 30 + \frac{3}{7}\lambda < 0 \quad \text{für } \lambda < -70 \\ 20 + \frac{2}{7}\lambda < 0 \quad \text{für } \lambda < -70 \\ 5 - \frac{1}{7}\lambda < 0 \quad \text{für } \lambda > 35 \\ 25 + \frac{2}{7}\lambda < 0 \quad \text{für } \lambda < -\frac{175}{2} \end{array}$$

Wegen primaler Zulässigkeit muss die rechte Seite  $\geq 0$  sein. Die im optimalen Tableau enthaltene BL bleibt für  $-70 \leq \lambda \leq 35$  (Durchschnitt aller Bedingungen), d.h.  $30 \leq b_2 \leq 135$  (statt ehemals  $b_2 = 100$ ) auch primal zulässig und damit optimal.

Eine Änderung eines  $b_i$  bewirkt im  $\mathbb{R}^2$ , dass die entsprechende restringierende Gerade parallel verschoben wird.

Veränderung eines Koeffizienten  $a_{ij}$  bewirkt im  $\mathbb{R}^2$  eine Drehung der entsprechenden Geraden.

In der Praxis treten diese drei Fälle natürlich nicht isoliert auf.

## 4.2 Postoptimale Modifikation

Die Postoptimale Modifikation bildet in gewissem Sinne eine Verallgemeinerung von Abschnitt 4.1.

### 4.2.1 Änderung der ZF-Koeffizienten

$$\begin{array}{ll} \max & z = c^T x \\ \text{s.d.} & Ax \leq b \\ & x \geq 0 \end{array}$$

Ersetze  $c$  durch  $c' = c + \lambda v$  ( $v$  Vektor;  $\lambda$  Skalar)

$$\Rightarrow \Delta z'_j = \Delta z_j + \lambda(v_B \cdot B^{-1}a_j - v_j)$$

wobei  $v_B$  die Komponenten von  $v$  sind, die der Basis entsprechen (d.h. Basisvariablen der optimalen Lösung),  $B^{-1}a_j$  die  $j$ -te Spalte der Koeffizientenmatrix ist und  $v_j$  die  $j$ -te Komponente von  $v$  ist.

Gilt  $v_B \cdot B^{-1}a_j - v_j \geq 0$ , dann ist für beliebig große  $\lambda \geq 0$  die duale Zulässigkeit nicht verletzt.

Ist sie jedoch verletzt rechne mit primaler Simplexmethode bis zum Optimum.

#### 4.2.2 Änderung der rechten Seite

Ersetze  $b$  durch  $b' = b + \lambda v$

Die duale Zulässigkeit bleibt bestehen. Für die primale Zulässigkeit gilt:

$$\begin{aligned} x'_B &= B^{-1}b' = B^{-1}b + \lambda B^{-1}v && \stackrel{?}{\geq} 0 \\ x'_B &= x_B + \lambda B^{-1}v \end{aligned}$$

Ist die primale Zulässigkeit verletzt transformiere mit dualer Simplexmethode.

#### 4.2.3 Hinzufügen von Variablen

Falls sich das zu untersuchende Problem ändert, kann es nötig sein Variablen hinzuzufügen. Variable  $x_{n+1}$  (mit Vektor in  $A$  und ZF-Koeffizient  $c_{n+1}$ ) soll hinzugefügt werden.

Ist  $B$  optimale Basis des Ausgangsproblem, dann nehme zusätzliche Spalte mit

$$B^{-1}a_{n+1} \quad \text{und} \quad \Delta z_{n+1} = c_B B^{-1}a_{n+1} - c_{n+1}$$

hinzu.

Ist  $\Delta z_{n+1} \geq 0$ , dann ist die Lösung weiterhin optimal, falls nicht verwende primalen Simplexalgorithmus.

#### 4.2.4 Hinzufügen von weiteren $\leq$ -Restriktionen

Falls sich eine neue Restriktion für das zu untersuchenden Problem ergibt, muss sowohl eine Zeile als auch eine Spalte in das Simplextableau eingefügt werden.

### 4.3 Idee der Parametrischen Programmierung

Die Änderung der Ausgangsdaten in Abhängigkeit von einem Parameter können so groß werden, dass sich die optimale Lösung auch qualitativ ändert (Übergang zu einer anderen Basis). Änderung der Eingabedaten verhalten sich proportional zu einem reellwertigen Parameter (1-parametrische lineare Optimierung). Hier werden nun auch „benachbarte“ Basislösungen und ihre „Gültigkeitsbereiche“ (kritische Bereiche) betrachtet. In Abhängigkeit von  $\lambda$  alle optimalen Lösungen im gültigen Parameterbereich berechnen.

Koeffizienten der rechten Seite bzw. der Zielfunktion:

- Funktion eines Parameters (hier) oder
- Funktion mehrere Parameter

Modell 1:

$$\begin{aligned} \max \quad & z = (c^T + v^T \lambda) x, \quad c, x, v \in \mathbb{R}^n, \lambda \in \mathbb{R} \\ \text{s.d.} \quad & Ax = b, \quad b \in \mathbb{R}^m, A = A_{m,n} \\ & x \geq 0 \end{aligned}$$

bzw. Modell 2:

$$\begin{aligned} \max \quad & z = c^T x, \quad c, x \in \mathbb{R}^n \\ \text{s.d.} \quad & Ax = b' = (b + v\lambda), \quad b, v \in \mathbb{R}^m, \lambda \in \mathbb{R}, A = A_{m,n} \\ & x \geq 0 \end{aligned}$$

## 5 Ganzzahlige LP

Ganzzahlige EV und damit ganzzahlige Optimierungsprobleme treten immer dann auf, wenn kein Kontinuum von Alternativen vorliegt. Ganzzahlige LOPs sind im Allgemeinen schwer, d.h. nur mit exponentiellem Rechenaufwand zu lösen. Ist der zulässige Bereich allerdings eine endliche Menge, so spricht man von *kombinatorischen Optimierungsproblemen*. In der Regel lassen sich die Nebenbedingungen einer kombinatorischen Optimierungsaufgabe wieder durch lineare Gleichungen bzw. Ungleichungen für ganzzahlige EV beschreiben. Der zulässige Bereich ist dann die Menge aller ganzzahligen Gitterpunkte eines konvexen Polytops im  $\mathbb{R}^n$ .

### 5.1 Problemstellung, Beispiele

Forderung nach ganzzahligen Werten für gewissen (evtl. alle) EV.  
Ganzzahlige LP

- Rein ganzzahlige LP (alle EV müssen ganzzahlig sein)
- Gemischt ganzzahlige LP (MIP = mixed integer programming)

In der Praxis sind reellwertige EV oft nicht gegeben:

- Keine beliebige Teilbarkeit, der durch die EV modellierten Ressourcen (z.B. Menschen, Gebäude, Maschinen, Stückgüter).
- Logische Bedingungen: binäre Variablen, z.B. Multiple-Choice-Bedingungen. Genau eine von  $n$  Alternativen wählen:

$$\sum_{j=1}^n y_j = 1 \quad \wedge \quad y_j = \begin{cases} 1, & \text{falls } A_j \text{ gewählt} \\ 0, & \text{sonst} \end{cases}$$

### 5.2 Das Schnittebenenverfahren von Gomory

Mit dem Verfahren von Gomory lassen sich gewöhnliche ganzzahlige LOPs lösen.  
Grundmodell:

$$\begin{aligned} \max \quad & z = c^T x, \quad c, x \in \mathbb{R}^n \\ \text{s.d.} \quad & Ax = b, \quad b \in \mathbb{R}^m \\ & x \geq 0 \end{aligned}$$

(Schlupfvariablen sind ggf. schon in  $x$  enthalten.)

Es liege eine primal und dual zulässige Basis vor.

O.B.d.A. sei die Indizierung so, dass

- die ersten  $m$  Variablen  $i = 1, \dots, m$  die BV sind und
- die letzten  $j = m + 1, \dots, m + n$  Variablen die NBV sind,

sodass die Basisdarstellung der  $i$ -ten BV

$$x_{B_i} = b_i^* - \sum_{j=m+1}^{m+n} a_{ij}^* \cdot x_j, \quad i = 1, \dots, m \text{ ist.} \quad (11)$$

Aufspaltung der nichtganzzahligen Anteile von  $b_i^*$  und  $a_{ij}^*$  in *ganzzahlige* und *nichtnegative gebrochene Anteile*.

$$\begin{aligned} b_i^* &= \lfloor b_i^* \rfloor + h_i \\ a_{ij}^* &= \lfloor a_{ij}^* \rfloor + h_{ij} \\ \lfloor b_i^* \rfloor, \lfloor a_{ij}^* \rfloor &\in \mathbb{Z}, \quad 0 \leq h_i, h_{ij} < 1 \end{aligned} \quad (12)$$

Einsetzen von (12) in (11) ergibt:

$$\begin{aligned} x_{B_i} &= \lfloor b_i^* \rfloor + h_i - \sum_{j=m+1}^{m+n} (\lfloor a_{ij}^* \rfloor + h_{ij}) \cdot x_j \\ &= \underbrace{\lfloor b_i^* \rfloor + \sum_{j=m+1}^{m+n} \lfloor a_{ij}^* \rfloor \cdot x_j}_{(I)} + \underbrace{h_i - \sum_{j=m+1}^{m+n} h_{ij} \cdot x_j}_{(II)} \end{aligned} \quad (13)$$

(I) ist nach Definition ganzzahlig (falls  $x_j$  ganzzahlig),

(II) ganzzahlig, ist notwendige und hinreichende Bedingung für die Ganzzahligkeit von  $x_{B_i}$ . Damit die Summe von (I) und (II) ganzzahlig ist, muss (II) ganzzahlig sein (da (I) schon ganzzahlig ist).

### Herleitung der Gomory-Restriktion

Es gilt:  $h_{ij} \cdot x_j \geq 0$ , da  $h_{ij} \geq 0 \wedge x_j \geq 0$ .

$$\begin{aligned} \Rightarrow \underbrace{h_i - \sum_{j=m+1}^{m+n} h_{ij} \cdot x_j}_{\geq 0} &\leq \underbrace{h_i}_{\text{da nicht ganzzahliger Teil}} < 1 \\ &\text{soll ganzzahlig sein} \\ \Rightarrow h_i - \sum_{j=m+1}^{m+n} h_{ij} \cdot x_j &\stackrel{!}{\leq} 0 \\ &\text{muss } \leq 0 \text{ sein, da } < 1 \text{ und Ganzzahligkeit gelten soll.} \\ \Leftrightarrow - \sum_{j=m+1}^{m+n} h_{ij} \cdot x_j &\leq -h_i \end{aligned}$$

ist eine *notwendige* Bedingung für die Ganzzahligkeit von (II).

Fügt man eine Gomory-Variable als Schlupfvariable hinzu, entsteht die *Gomory-Restriktion*

$$x_G - \sum_{j=m+1}^{m+n} h_{ij} \cdot x_j = -h_i \quad (14)$$

die eine Schnittebene darstellt (und daher ein Stück vom zulässigen Bereich abschneidet.) Da in BL (12) alle NBV  $j = m + 1, \dots, n$  gleich 0 sind, gilt in (14):

$$x_G = -h_i$$

### Definition 19

Man spricht von einem *relaxierten Problem* oder einer *Relaxation* eines Optimierungsproblems, wenn einige (oder alle) Nebenbedingungen des Problems weggelassen oder „gelockert“ werden.

Das bekannte LOP ist also eine Relaxation der ganzzahligen Optimierungsaufgabe.

### Modifikation der Simplexmethode

- Dem optimalen Simplextableau des relaxierten LP die Gomory-Restriktion hinzufügen!
- ⇒
- Die duale Zulässigkeit der BL bleibt bestehen.
  - Wegen  $-h_i < 0$ , wird die primale Zulässigkeit verletzt.
- Deshalb: Eine oder mehrere duale Simplexiteration(en) mit Gomory-Restriktion als Pivotzeile durchführen.  
Es ergibt sich:
    1. Die primale Unzulässigkeit der hinzugefügten Zeile wird beseitigt.
    2. Die Gomory-Variable wird NBV.
  - Ist duale Simplexiteration nicht möglich, so ist der Lösungsraum leer, d.h. im kontinuierlichen Lösungsraum gibt es keine ganzzahligen Lösungen.

### Schnittebenen-Verfahren von Gomory

1. Löse das Grundmodell (LP-Relaxation) mit der Simplexmethode.
2. Sind in der optimalen BL alle Ganzzahligkeitsbedingungen gefunden (STOP), sonst:
3. Spalte die Komponenten  $b_i^*$  der rechten Seite (d.h. der BL) im optimalen Tableau für die ganzzahlig geforderten Variablen gemäß (12) auf. Der Zeilenindex  $t$ , für den die Gomory-Restriktion gebildet wird, wird aus  $h_t = \max_i \{h_i | h_i \text{ aus Aufspaltung (12)}\}$  ermittelt. Ist  $t$  nicht eindeutig, verwende Kleinste-Index-Regel (Heuristik).
4. Spalte die  $a_{tj}^*$  der  $t$ -Zeile des Tableaus nach (12) auf:

$$a_{tj}^* = \lfloor a_{tj}^* \rfloor + h_{tj}, \quad a_{tj}^* \text{ ganzzahlig, } 0 \leq h_{tj} < 1$$

Bilde die Gomory-Restriktion nach (14).

5. Füge dem aktuelle Tableau die Gomory-Restriktion hinzu und iteriere mit der dualen Simplexmethode. Ist derartige Iteration nicht möglich, dann gibt es keine ganzzahlige Lösung (STOP), sonst gehe zu Schritt 2.

## 6 Revidierter Simplexalgorithmus

Beim revidierten Simplexalgorithmus handelt es sich um eine besondere Form der Simplexmethode, die speziell für große LOPs entwickelt wurde und der gewöhnlichen Simplexmethode bei „dünn besetzten“ Koeffizientenmatrizen rechentechnisch überlegen ist. In der Vorlesung wurde der revidierte Simplexalgorithmus allerdings nicht besprochen.

### 6.1 Grundlagen

entfällt

### 6.2 Produktform der Inversen

entfällt

### 6.3 Basistransformation des revidierten Simplexalgorithmus

entfällt

## 7 Projektplanung

### Definition 20 (Projekt)

Ein *Projekt* ist ein Vorhaben mit einem definierten Anfang und einem definierten Ende, charakterisiert durch „Einmaligkeit der Bedingungen in ihrer Gesamtheit“, z.B.

- Zielvorgaben
- zeitliche, finanzielle, personelle oder andere Restriktionen
- Abgrenzung gegenüber anderen Vorhaben (beispielsweise Forschungsprojekt)
- projektspezifische Organisation

weitere Eigenschaften

- Neuartigkeit
- Komplexität

Projektergebnisse sind z.B.

- ein neues Produkt
- eine neue Anlage (Bauten, Systeme)
- veränderte Fertigungsprozesse (Technologien)

*Projektplanung* ist ein wesentlicher Teil des über den gesamten Projektzeitraum reichende „Projektmanagement“.

Teilaufgaben der Projektplanung sind:

- Planung von Teil- oder Unterzielen
- Ableitung von Teilaufgaben des Projekts



- Planung der Ablauforganisation (Netzplanung)
- Planung des Aufwandes für Personal, Finanzen, Sachmittel, Raumbedarf (Ressourceneinsatzplanung)
- Planung des Projekt-Informations- und Dokumentationssystems

Focus hier: Techniken und Systeme zur Unterstützung der Projektplanung (quantitative Methoden)

## 8 Entwicklung und Grundlagen der Netzplantechnik

Die Netzplantechnik dient der optimalen Planung und Überwachung von Projekten.

### 8.1 Begriffe der Netzplantechnik

1. CPM: „Critical Path Method“
2. PERT: „Program Evaluation and Review Technique“
3. MPM: „Metra-Potential-Methode“

Idee:

Zerlegung der zu planenden oder zu steuernden Komplexe in Elemente, um sie darzustellen, analysieren und steuern zu können.

Die Gesamtheit des zu Planenden heißt Projekt (siehe Abschnitt 7, dieses zerlegt man in zwei Arten von Elementen:

- Vorgänge = Tätigkeit, deren Ausführung Zeit erfordert
- Ereignisse = Abschluss gewisser (Mengen) von Vorgänge(n)

#### Definition 21

Ein *Graph*  $G = (V, E)$  ist eine endliche nicht leere Menge  $V \neq \emptyset$  von Knoten (vertices) mit einer Menge  $E \subseteq \{M \subseteq V \mid |M| = 2\}$  von Kanten (edges). D.h. auf  $E$  ist eine Abbildung  $\Psi$  definiert, die jedem  $e \in E$  eine Menge von zwei Elementen  $v, v' \in V$  zuordnet ( $v = v'$  möglich, dann Schleife). Ist das zu jedem  $e \in E$  zugeordnete Paar von Elementen aus  $V$  geordnet, so sind die Kanten gerichtet ( $\rightarrow$ ) und man spricht von einem *gerichteten Graphen*, sonst von einem *ungerichteten Graphen*.

Grundbegriffe:

*Vorgänge*: Zeitbeanspruchende Elemente eines Projekts (Projektteile mit definiertem zeitlichen Anfang und Ende).

*Ereignisse*: Definierte Zustände im Projektablauf (Zeitpunkte, an denen Projektteile beginnen/enden).

*Netzplan*: (spezieller Graph) bestehend aus Knoten und Kanten.

*Netz*: Zyklensfreier Digraph (endlich, gerichtet, schlicht (d.h. keine Schlingen oder parallele Kanten)) mit genau einer Quelle und genau einer Senke.

**Netzwerk:** Ordnet man durch eine Abbildung von  $E$  in den  $\mathbb{R}^n$  den Pfeilen eines Netzes Werte zu, so nennt man das bewertete Netz ein Netzwerk.

Es existieren zwei Darstellungsarten:

- *Pfeildarstellung:* Vorgänge werden durch Pfeile, Ereignisse durch Knoten dargestellt.
- *Kreisdarstellung:* Vorgänge werden durch Knoten, Ereignisse nicht explizit dargestellt. Pfeile deuten den Zusammenhang zwischen Vorgängen an.

Im [1] folgen an dieser Stelle eine Reihe von Definitionen, die hier übersprungen werden.

*Vorgangsorientierter Netzplan:* Vorgänge werden bzgl. Inhalt und Zeitdauer beschrieben.

⇒ Ereignisse: Anfangs- und Endzeitpunkt der Vorgänge.

(beide Darstellungsarten möglich)

*Ereignisorientierter Netzplan:* Beschreibung der Ereignisse inhaltlich als Kontrollpunkt im Projektverlauf.

⇒ Vorgänge im Wesentlichen undefiniert.

(Pfeildarstellung, da Vorgänge undefiniert)

Für die drei Netzplantypen gilt:

1. CPM: Vorgangspfeilnetzwerk (vorgangsorientierter Netzplan in Pfeildarstellung mit Ende-Start-Verknüpfung)
2. MPM: Vorgangsknotennetz (vorgangsorientierter Netzplan in Kreisdarstellung mit Start-Start-Verknüpfung)
3. PERT: Ereignisknotennetz (ereignisorientiertes Netz in Pfeildarstellung, d.h. Ereignisse als Knoten, die Dauer zwischen zwei Ereignissen ist nicht determiniert)

Der Netzplangentwurf und die Berechnung erfolgen unabhängig von der Art des Netzes in vier Stufen:

1. *Strukturanalyse:* Vorgänge, Ereignisse definieren, Darstellung der Abhängigkeitsbeziehungen der Vorgänge. Modellierungsarbeit: Erstellung einer Vorgangsliste, Entwurf eines Netzplans, Kontrolle bzgl. Grundregeln.
2. *Zeitanalyse:* Bestimmung frühester und spätester Anfangs- und Endzeiten der Vorgänge, der Gesamtprojektdauer und der verschiedenen Pufferzeiten.
3. *Kapazitätsanalyse (Ressourcen):* Beschränkte Ressourcen zur Durchführung der Vorgänge bewirken Engpässe: ⇒ Bestimmung optimaler Reihenfolgen von Vorgängen bei Beachtung der vorhandenen Kapazitäten und technologisch bedingten Reihenfolgebeziehungen. Einhaltung von Kapazitätsgrenzen und Terminen.
4. *Kostenanalyse:* Kann man die Dauer der Vorgänge beeinflussen, so kann man bei fester Projektdauer die Vorgangsdauer bestimmen, die minimale Gesamtkosten erfordern.

Die wesentlichen Entscheidungen fallen bereits in Stufe 1, Optimierung ist vor allem in Stufe 3 und 4 möglich.

Im weiteren Verlauf dieses Abschnitts soll die in der Netzplantechnik übliche graphische Darstellung erläutert werden, die zum einen einer visuellen Veranschaulichung des Projektes dient und zum anderen die Bestimmung der längster Wege für die Zeitplanung erlaubt.

Ein Netzplanelement besteht aus einem Vorgang und dessen Anfangs- und Endereignis

Ein Vorgang wird charakterisiert durch einen Namen und das Paar  $\langle i, j \rangle$ , wobei  $i, j$  Ereignisnummern sind.

Regeln:

1. Jeder Vorgang beginnt mit einem Ereignis und endet mit einem nachfolgenden Ereignis.
2. Müssen ein oder mehrere Vorgänge beendet sein, bevor ein weiterer Vorgang beginnen kann, so enden diese alle im Anfangsereignis des nachfolgenden Vorgangs.
3. Können mehrere Vorgänge beginnen, nachdem ein vorausgegangener beendet ist, so beginnen diese alle im Endereignis des vorangegangenen Vorgangs.
4. Haben zwei Vorgänge gemeinsame Anfangs- und Endereignisse, dann wird eine eindeutige Charakterisierung durch einen Scheinvorgang hergestellt.

Scheinvorgang S wird benötigt, besitzt Dauer 0 und hat keine Ressourcen

5. Enden und beginnen in einem Ereignis mehrere Vorgänge, die nicht alle voneinander abhängig sind, stellt man den richtigen Ablauf mit einem Scheinvorgang her.

C kann erst nach Abschluss von A und B beginnen, D kann nach Abschluss von B beginnen (muss aber nicht auf A warten).

6. Kann ein Vorgang beginnen, bevor der vorhergehende vollständig beendet ist, kann der vorhergehende entsprechend unterteilt werden.

7. In einem Netzplan dürfen keine Zyklen auftreten.

Eindeutige Darstellung erfordert Scheinvorgänge.

Wegen der Einführung von Scheinvorgängen ist ein Netzplan im Allgemeinen nicht eindeutig.

## 8.2 CPM

Die Strukturanalyse bei CPM ist analog zu der in Abschnitt 8.1. Die Zeitanalyse besteht bei CPM aus:

- Bestimmung der Vorgangsdauer
- Progressive (Vorwärtsrechnung) und retrograde (Rückwärtsrechnung) Zeitanalyse
- Ermittlung des kritischen Weges und der Zeitreserven

Zweistufige Berechnung:

- a) Berechnung von je zwei Ereigniszeiten
- b) Berechnung von je vier Zeiten pro Vorgang  
(Zuerst a) und daraus dann b) ableiten)

Zu a):

1. Berechnung des *frühestmöglichen* Zeitpunktes  $FZ_i$  für das Ereignis  $i$ .

Frühestmögliche Zeitpunkte der Knotenereignisse sind gleich der Dauer des zeitlängsten Weges vom Startknoten zum Knoten  $i$ . Addition der Dauern  $D_{k,l}$  der Vorgänge auf dem zeitlängsten Weg vom Start zum Ereignis  $i$  (immer das Maximum wählen).

2. Berechnung des *spätest erlaubten* Zeitpunktes für Ereignis  $i$ :  $SZ_i$ , zeitlängster Weg vom Ziel zurück zum Ereignis (immer das Minimum wählen).

Sei  $d_j$  der längste Zeitabstand der Ereignisses  $j$  zum Projektende, und  $\lambda$  die minimale Gesamtprojektdauer (also zuerst alle  $FZ_i$  berechnen).  $\Rightarrow SZ_j = \lambda - d_j$

Zu b):

Sind die Strukturanalyse und a) abgeschlossen, lassen sich aus diesem CPM-Netzplan folgende Ergebnisdaten berechnen:

- Frühestmögliche Beginn- und Abschlusspunkte der einzelnen Vorgänge und damit die minimale Gesamtdauer des Projekts (Vorwärtsrechnung).
- Spätest zulässige Beginn- und Abschlusspunkte der einzelnen Vorgänge unter der Voraussetzung, dass der frühest mögliche Endtermin des Projekt auch eingehalten wird (Rückwärtsrechnung).
- Drei Arten von Puffer- oder Schlupfzeiten der einzelnen Vorgänge (Verschiebbarkeit der Vorgänge):
  - gesamter Puffer,
  - freier Puffer,
  - unabhängiger Puffer

⇒ Vier elementare Zeitpunktgrößen eines Vorgangs

1. Frühestmögliche Anfangszeit  $FA_{i,j}$
2. Spätesterlaubte Anfangszeit  $SA_{i,j}$
3. Frühestmögliches Ende  $FE_{i,j}$
4. Spätesterlaubte Ende  $SE_{i,j}$

Die Vorgangszeiten werden wie folgt aus Ereigniszeiten berechnet

- Frühestmögliche Anfangszeit:  $FA_{i,j} = FZ_i$
- Frühestmögliches Ende:  $FE_{i,j} = FZ_i + D_{i,j}$
- Spätesterlaute (mögliche) Anfangszeit :  $SA_{i,j} = SZ_j - D_{i,j}$
- Spätesterlautes (mögliches) Ende:  $SE_{i,j} = SZ_j$
- maximal verfügbare Zeit:  $SZ_j - FZ_i$

Kritischer Pfad (Weg)

- Vorgang  $\langle i, j \rangle$  mit maximal verfügbarer Zeit ist gleich der Dauer  $D_{i,j}$  heißt kritisch:  $SZ_j - FZ_i = D_{i,j}$   
Eine Verlängerung eines kritischen Vorgangs bewirkt gleich eine Verlängerung der kürzesten Projektdauer. Die kürzeste Projektdauer ist die kürzeste Zeitspanne, in der alle zum Projekt gehörigen Vorgänge ausgeführt werden können und damit gleich der Länge eines längsten Weges von der Quelle zur Senke.
- Ununterbrochene Folge kritischer Vorgänge vom Beginn (Quelle) zum Endereignis (Senke) eines Projektes heißt kritischer Pfad.
- Jede Verlängerung eines kritischen Vorgangs verlängert die Gesamtprojektdauer.
- Für kritische Vorgänge  $\langle i, j \rangle$  gilt:  $FZ_i = SZ_i$ ;  $FZ_j = SZ_j$

Zeitreserven/Pufferzeiten eines Vorgangs. Als Pufferzeit bezeichnet man die maximale Zeitspanne, um die der Vorgang hinausgeschoben werden kann, ohne gewisse Restriktionen zu verletzen.

- gesamter Puffer:  $GP_{i,j} = SA_{i,j} - FA_{i,j} = SE_{i,j} - FE_{i,j} = SZ_j - (FZ_i + D_{i,j})$   
Maximale Zeitspanne, um die der Beginn von Vorgang  $\langle i, j \rangle$  hinausgeschoben werden kann, ohne den Projektendtermin zu gefährden.
- freier Puffer:  $FP_{i,j} = FZ_j - FE_{i,j} = FZ_j - (FZ_i + D_{i,j}) \leq GP_{i,j}$   
Maximale Zeitspanne, um die der Beginn von Vorgang  $\langle i, j \rangle$  hinausgeschoben werden kann, ohne dass der frühestmögliche Anfang aller nachfolgenden Vorgänge beeinflusst wird.
- unabhängiger Puffer:  $UP_{i,j} = \begin{cases} FZ_j - (SZ_i + D_{i,j}), & \text{falls } > 0 \\ 0, & \text{sonst} \end{cases}$   
Maximale Zeitspanne, sodass alle vorhergehenden Vorgänge zum spätesten erlaubten Ende enden und alle nachfolgenden Vorgänge zum frühestmöglichen Anfang beginnen.

Eigenschaften von CPM-Netzplänen:

Jeder CPM-Netzplan besitzt genau eine Quelle und eine Senke, die den Ereignissen Beginn bzw. Abschluss des Projekts entsprechen. Ein CPM-Netzplan ist schwach zusammenhängend und zyklonfrei, denn das Auftreten eines Zyklus würde bedeuten, dass die Beendigung aller den Pfeilen des Zyklus entsprechenden Vorgänge dem Beginn dieser Vorgänge voranginge. Ein Vorgang ist kritisch, wenn die Gesamtpufferzeit für ihn ihren kleinstmöglichen Wert annimmt, welcher gleich  $SZ_1$  ist für das Ereignis 1 als Quelle.

### 8.3 PERT (bei Ereignisknotennetzen)

In der Praxis ist die Schätzung der Vorgangsdauern eines Projekts oft mit großen Unsicherheiten verbunden. In diesem Fall ist es zweckmäßig, die Vorgangsdauern nicht mehr als deterministische Größen, sondern als Zufallsgrößen anzusehen. Hierzu ist die Netzplantechnik PERT entwickelt worden, die im Folgenden skizziert wird.

Wie bei CPM werden Ereignisse durch Knoten modelliert, allerdings liegt der Fokus hier auf den Ereignissen, daher auch Ereignisknotennetz. Die Vorgänge werden nicht explizit definiert, sodass die Pfeile nur die Reihenfolge zwischen Ereignissen abbilden. Bei der Strukturanalyse wird gemäß der Reihenfolgevorschriften der Netzplan konstruiert. Der Hauptunterschied zu CPM besteht in der Zeitplanung. Es werden stochastische Zeiten benutzt, wobei die Schätzungen von voraussichtlichen Zeiten ohne statistische Begründung ausgehen. Die „Vorgangsdauern“ (Vorgänge sind nicht explizit definiert) bilden die Mindestspanne zwischen den Ereignissen und werden bei PERT als Zufallsvariable interpretiert.

Die zugrundeliegende Verteilung der Vorgangsdauer sollte

1. durch wenige Parameter bestimmt sein,
2. unsymmetrisch und nach links schief sein (d.h. Maximum der Funktion liegt links neben dem Erwartungswert),
3. Realisierungen in einem endlichen Intervall besitzen,
4. die Parameter seien möglichst einfach berechenbar oder approximierbar.

Die Vorgangsdauern werden bei PERT als betaverteilte Zufallsgrößen angesehen.

Zeiten  $t$ :

- $t_{i,j}$  : Zeit zwischen Ereignis  $i$  ( $E_i$ ) und Ereignis  $j$  ( $E_j$ )
- $D_{i,j}$  : die entsprechende Zufallsvariable
- $t_{F(i)}$  : früheste Zeit von  $E_i$
- $t_{S(i)}$  : späteste Zeit von  $E_i$

Für die Dichte der  $\beta$ -Verteilung gilt:

$$f(t) = \begin{cases} k \cdot (t-a)^\alpha \cdot (b-t)^\beta & \text{für } a \leq t \leq b \\ 0 & \text{sonst} \end{cases}$$

mit  $b > a$ ,  $\alpha, \beta > -1$ , in der Praxis  $\alpha, \beta > 0$ .

$k$  ist ein Normierungsfaktor, damit  $\int_{-\infty}^{\infty} f(t) dt = 1 \Rightarrow k = \frac{1}{\Gamma(\alpha+1, \beta+1)}$

$m$  : Maximum von  $f(t)$

$t_e$  : Erwartungswert von  $D$ ,  $t_e = \int_a^b t \cdot f(t) dt$

Mit Hilfe der nachfolgenden drei Zeitschätzungen werden die Parameter der Betaverteilung für eine Vorgangsdauer  $D_{i,j}$  geschätzt.

$a = OD$  Optimistische Dauer (unter besonders günstigen Bedingungen erreichbar)

$m = HD$  „Häufigste“ (wahrscheinlichste) Dauer (unter normalen Bedingungen erreichbar)

$b = PD$  Pessimistische Dauer (bei besonders ungünstigen Bedingungen)

Damit folgt für den Erwartungswert und die Varianz

$$t_e = \frac{OD + 4HD + PD}{6} \quad \sigma^2 = \frac{(PD - OD)^2}{36}, \text{ für } \alpha = \beta = 4$$

Bisher sind  $\alpha$  und  $\beta$  noch beliebig, benutze nun die heuristische Regel  $\alpha = \beta = 4$  und fasse  $m$  als zu schätzenden Parameter auf.

$$\Rightarrow t_e = \frac{a + 4m + b}{6} \quad \sigma^2 = \left( \frac{b-a}{6} \right)^2$$

$$\text{exakt für: } \left( \frac{a+b}{2} - m \right)^2 = \frac{1}{8}(b-a)^2$$

D.h.  $t_e$  und  $\sigma^2$  sind mit  $a, b, m$  eindeutig bestimmt unter der Voraussetzung  $\alpha + \beta = 4$  ( $\alpha, \beta$  werden nicht explizit gebraucht).

Die eigentliche Zeitanalyse läuft nun wie folgt ab:

- Bestimmung der frühesten und spätesten Ereigniszeiten:  $t_{F_i}, t_{S_i}$ .  
Als Basis dienen die errechneten  $t_e$  für alle  $D_{i,j}$  wie bei CPM mit Vorwärts- und Rückwärtsrechnung.
- Bestimmung „kritischer Ereignisse“:  
Ereignisse, deren früheste Ereigniszeit = späteste Ereigniszeit (Ereignispuffer = 0)
- Kritischer Pfad: Folge von kritischen Ereignissen von der Quelle zur Senke.

Interpretation: Zentraler Grenzwertsatz (Summe beliebig verteilter Zufallsverteilungen ist asymptotisch normalverteilt)

Die Summe von unabhängigen Zufallsgrößen ist, wenn die Anzahl der Summanden groß ist, näherungsweise normalverteilt. In Anlehnung daran werden bei PERT die Zufallsgrößen (also die Ereigniszeiten)  $t_{F(i)}$  und  $t_{S(i)}$  als normalverteilt angenommen (obwohl die einzelnen Vorgänge in der Regel nicht voneinander unabhängig sind und nicht nur jeweils ein Weg diese Zeitpunkte bestimmt). Insbesondere ist dann die kürzeste Projektdauer mit  $t_e$  und  $\sigma^2$  normalverteilt und lässt sich einfach bestimmen. Mit Hilfe dieser Wahrscheinlichkeitsverteilung kann man die Wahrscheinlichkeit von Terminüberschreitungen ermitteln.

In [1] findet sich ein Rechenbeispiel zu PERT. Da etliche im Rahmen der Methode PERT ausgeführte Rechnungen zu fehlerhaften Ergebnissen führen (da Erwartungswertbildung und Maximierung/Minimierung vertauscht werden), sollte PERT nur dann benutzt werden, wenn eine „deterministische Projektplanung“ tatsächlich unangebracht ist (z.B. bei weitstreuenden Vorgangsdauerschätzungen).

#### 8.4 MPM

Wurde in der Vorlesung nicht behandelt und entfällt daher.

#### 8.5 Exkurs: Gantt-Chart (Balkendiagramm)

Ein Nachteil der Netzpläne, die zuvor behandelt wurden ist, dass diese keine gute Übersicht über den zeitlichen Ablauf eines Projektes geben. Für diesen Zweck empfehlen sich *Balken-* oder *Gantt*diagramme (beliebt zur Darstellung von Ablaufplänen).

Im Balkendiagramm entspricht jedem realen Vorgang  $\langle i, j \rangle$  ein Balken über der Zeitachse, der vom Zeitpunkt  $FA_{i,j}$  bis zum Zeitpunkt  $FE_{i,j}$  reicht, dessen Länge also gleich der Vorgangsdauer  $D_{i,j}$  ist. An diesen Balken wird noch ein Stück der Länge  $GP_{i,j}$  angehängt.

Vorteile:

- Länge der Balken steht in Beziehung zur Zeit
- Diagramm wird zur Aufzeichnung des Arbeitsfortschrittes benutzt
- Auch für Nicht-Spezialisten informativ, Soll-Ist-Vergleich, breiter Anwendungsbereich

Nachteile:

- Simultane Struktur- und Zeitplanung



- Geringe Aussagefähigkeit bzgl. Reihenfolge und Verknüpfung der Vorgänge
- Feste Terminierung aller Vorgängen die zu einem Projekt gehören
- Geringe Aussagefähigkeit zur Verbesserung von Abläufen
- Begrenzte Anzahl der darstellbaren Vorgänge (Übersichtlichkeit)

## 9 Zuordnungsproblem

Vergleiche einführendes Beispiel in Abschnitt 1.3. Wie später zu sehen ist, ist das Zuordnungsproblem ein Spezialfall des in Abschnitt 10.2 behandelten Umladeproblems.

### 9.1 Anwendungsbeispiel

Eine Omnibusgesellschaft bieten einen Liniendienst von Mexico City nach Acapulco. Die Fahrt dauert 6 Stunden. Die Besatzung eines jeden Bussen besteht aus zwei Personen. Die Fahrzeuge fahren nach folgendem Fahrplan:

Mexico-Acapulco			Acapulco-Mexico		
Abfahrt	Linie	Ankunft	Ankunft	Linie	Abfahrt
06:00	$\xrightarrow{a}$	12:00	11:30	$\xleftarrow{1}$	05:30
07:30	$\xrightarrow{b}$	13:30	15:30	$\xleftarrow{2}$	09:30
11:30	$\xrightarrow{c}$	17:30	21:00	$\xleftarrow{3}$	15:00
19:00	$\xrightarrow{d}$	01:00	00:30	$\xleftarrow{4}$	18:30
00:30	$\xrightarrow{e}$	06:30	06:00	$\xleftarrow{5}$	00:00

Problem: Wohnorte der Bus-Crews

Einschränkungen: Abwesenheit der Crews vom Wohnort soll möglichst klein gehalten werden. Eine Crew muss zwischen zwei Fahrten mehr als 4 Stunden Pause haben, soll jedoch nicht länger als 24 Stunden pausieren.

Untersucht man die Fahrpläne, so treten die nachfolgenden Wartezeiten auf:

Crews, die in Mexico-City wohnen						Crews, die in Acapulco wohnen					
	1	2	3	4	5		1	2	3	4	5
a	17,5	21	3	6,5	12	a	18,5	15	9	5,5	0
b	16	19,5	1,5	5	10,5	b	20	16,5	10,5	7	1,5
c	12	15,5	21,5	1	6,5	c	0	20,5	14,5	11	5,5
d	4,5	8	14	17,5	23	d	7,5	4	22	18,5	13
e	23	2,5	8,5	12	17,5	e	13	9,5	3,5	0	18,5

Die 4 Stunden Mindestwartezeit sind in den Tabellen noch nicht berücksichtigt. Es lässt sich nun eine dritte Tabellen aufstellen (links), die jeweils die kleinere der beiden Wartezeiten enthält. Um die Mindestwerte für Pausen von mehr als 4 Stunden zu berücksichtigen, werden die „Verlustzeiten“ von weniger als 4 Stunden unberücksichtigt gelassen.

	1	2	3	4	5
a	17,5	15	9	5,5	12
b	16	16,5	10,5	5	10,5
c	12	15,5	14,5	11	5,5
d	4,5	8	14	17,5	13
e	13	9,5	8,5	12	17,5

	1	2	3	4	5
a	0	1	0	0	0
b	1	0	0	0	0
c	0	0	0	0	1
d	0	0	1	0	0
e	0	0	0	1	0

Die Wahl des Einsatzplanes besteht nun aus der Bestimmung von Paaren von Hin- und Rückfahrten, sodass alle 5 Crews eingesetzt sind. Eine Möglichkeit wäre die oben abgebildete rechte Tabelle. Sie entspricht dem Fahrplan 2a, b1, 5c, d3, e4. Hierbei würden drei Crews in Mexico-City und zwei in Acapulco wohnen und die Gesamtwartezeit wäre:  $15 + 16 + 5,5 + 14 + 12 = 62,5$  Stunden.

Für die vorliegenden Problemgröße gibt es  $5! = 120$  Einsatzpläne, bei 6 Linien bereits 720 und bei 20 sind es  $2.4329 \cdot 10^{18}$ .

Die optimale Lösung sieht folgendermaßen aus:

	1	2	3	4	5	Routen	Wohnort	Wartezeit
a	11	5	0	0	4,5	d-1	Mexico-City	4,5 Std.
b	10	7	2	0	3,5	2-e	Acapulco	2,5 Std.
c	7,5	7,5	7,5	7,5	0	3-a	Acapulco	9 Std.
d	0	0	7	14	7,5	b-4	Mexico-City	5 Std.
e	7	0	0	7	10,5	5-c	Acapulco	5,5 Std.

Die Gesamtwartezeit beträgt 33,5 Std. Die ungünstigste Lösung führt zu einer Wartezeit von 83,5 Stunden, dazwischen liegen 118 weitere Lösungen.

## 9.2 Ein Heuristisches Eröffnungsverfahren

Bewertungsmatrix eines  $5 \times 5$  Zuordnungsproblems

$$C = \begin{pmatrix} 4 & 9 & 18 & 6 & 5 \\ 23 & 6 & 2 & 3 & 3 \\ 7 & 2 & 0 & 21 & 3 \\ 4 & 8 & 7 & 6 & 28 \\ 4 & 9 & 18 & 3 & 2 \end{pmatrix}, C = (c_{ij})_{m,n}$$

Problem: In  $C$  sind  $n$  Elemente so auszuzeichnen, dass in jeder Zeile und in jeder Spalte genau eine Auszeichnung besteht. Die Summe der Bewertungen der ausgezeichneten Elemente sei möglichst klein.

Modell:

$$C \text{ und } x_{ij} = \begin{cases} 1, & \text{Element } (i, j) \text{ wird ausgezeichnet} \\ 0, & \text{sonst} \end{cases}$$

$$\text{Zeilenbedingung: } \sum_{j=1}^n x_{ij} = 1 \text{ für jedes } i = 1, \dots, n$$

$$\text{Spaltenbedingung: } \sum_{i=1}^n x_{ij} = 1 \text{ für jedes } j = 1, \dots, n$$

$$\text{Zielfunktion: } z = \sum_{i,j} c_{ij} \cdot x_{ij} \rightarrow \min$$

Greedy-Heuristik: Man zeichne nacheinander das kleinste jeweils noch mögliche (zulässige) Element aus.

Für das vorherige Beispiel gilt dann:  $x_{33} = x_{55} = x_{24} = x_{11} = x_{42} = 1$ , restlichen  $x_{ij} = 0$  mit den „Kosten“  $0 + 2 + 3 + 4 + 8 = 17$ .

### 9.3 Die Ungarische Methode

#### Satz 12

Die optimale Zuordnung ändert sich nicht, wenn zu eine Zeile oder Spalte der Bewertungsmatrix  $C$  eine Konstante addiert wird.

⇒ Optimale Zuordnung unverändert, wenn

- in jeder Zeile das Zeilenminimum (Zeilenreduktion) und anschließend
- in jeder Spalte das Spaltenminimum (Spaltenreduktion) subtrahiert wird.

⇒ Matrix  $\bar{C} \geq 0$  in der in jeder Spalte und in jeder Zeile mindestens eine Null steht.

Beispiel:

$$C = \begin{pmatrix} 4 & 9 & 18 & 6 & 5 \\ 23 & 6 & 2 & 3 & 3 \\ 7 & 2 & 0 & 21 & 3 \\ 4 & 8 & 7 & 6 & 28 \\ 4 & 9 & 18 & 3 & 2 \end{pmatrix} \xrightarrow{\text{ZR}} \begin{pmatrix} 0 & 5 & 14 & 2 & 1 \\ 21 & 4 & 0 & 1 & 1 \\ 7 & 2 & 0 & 21 & 3 \\ 0 & 4 & 3 & 2 & 24 \\ 2 & 7 & 16 & 1 & 0 \end{pmatrix} \xrightarrow{\text{SR}} \begin{pmatrix} [0] & 3 & 14 & 1 & 1 \\ 21 & 2 & 0 & [0] & 1 \\ 7 & 0 & [0] & 20 & 3 \\ 0 & 2 & 3 & 1 & 24 \\ 2 & 5 & 16 & 0 & [0] \end{pmatrix} = \bar{C}$$

Gelingt eine vollständige Zuordnung, bei der nur Nullen zugeordnet sind, ist die gefundene Lösung offenbar die optimale Lösung. Die Auszeichnung im Beispiel entspricht keiner vollständigen Zuordnung, und lässt sich auch nicht vervollständigen.

Heuristischer Ausweg:

„Nächstbestes“ Element auszeichnen, hier  $\bar{c}_{42} = 2$ , liefert einen ZF-Wert von 17 (nicht optimal). Gesucht ist eine Methode, die das Finden einer Optimal-Lösung garantiert.

Ansatz: Von vornherein andere Nullen auszeichnen, d.h. anders verzweigen, durch Aufnahme von Variablen.

$$\begin{pmatrix} [0] & a & 0 \\ b & [0] & c \\ 0 & d & e \end{pmatrix} \rightarrow \begin{pmatrix} 0 & a & [0] \\ b & [0] & c \\ [0] & d & e \end{pmatrix} \text{ vollständige Auszeichnung}$$

Entwicklung eines Algorithmuses, der diese Modifikation fasst.

Begriffe:

- *freie Null*: nicht ausgezeichnete Null
- *alternierender Pfad*: Kantenzug mit

- abwechselnd senkrechten und waagerechten Kanten
- in Ecken stehen abwechselnd freie und ausgezeichnete Nullen
- Beginn mit freier Null in Zeile ohne Auszeichnung

3 Typen des Abbruchs eines solchen Pfades

- a) Pfad endet mit ausgezeichneter Null, weil in der Zeile keine freie Null existiert (Sackgassenpfad).  
Sackgasse  $\hat{=}$  Es ist keine Vervollständigung der Zuordnung erreichbar.
- b) Pfad enthält einen Zyklus (Sackgassenpfad).
- c) Pfad endet in einer freien Null, da in der Spalte keine ausgezeichnete Null existiert.  
Eine Zuordnung wird durch Vertauschen der Rollen auf dem Pfad erreicht.

**Prozedur:** *Alternierende Wege*

Solange es einen alternierenden Weg gibt, der mit einer Null in einer Zeile ohne Auszeichnung beginnt und bei einer freien Null endet, vertausche die Rollen von freien und ausgezeichneten Nullen auf diesem Pfad.

Im  $3 \times 3$  Beispiel führt dies zur optimalen Lösung, im  $5 \times 5$  Beispiel existiert nur ein alternierender Pfad, der vom Typ a) ist.

Ausweg: Zu Zeilen und Spalten von  $\bar{C}$  Konstanten derart addiert, dass die Matrix nicht-negativ bleibt, aber mindestens eine neue Null entsteht.

Dies ist wegen des folgenden Satzes möglich.

**Satz 13**

Lassen sich in einer  $n \times n$  Bewertungsmatrix (auch nach der Prozedur Alternierende Wege) nur  $K < n$  Nullen auszeichnen, dann gibt es  $K$  Linien durch Spalten und Zeilen, die alle Nullen überdecken.

**Prozedur:** *Überdeckung*

Streiche alle Spalten mit einer ausgezeichneten Null, die auf einem Sackgassenpfad liegen. Streiche alle Zeilen, in denen eine ausgezeichnete Null noch nicht durchgestrichen ist.

Diese Prozedur erzeugt genau  $K < n$  Linien und jede ausgezeichnete Null ist durchgestrichen.

$$\begin{pmatrix} [0] & 3 & 14 & 1 & 1 \\ 21 & 2 & 0 & [0] & 1 \\ 7 & 0 & [0] & 20 & 3 \\ 0 & 2 & 3 & 1 & 24 \\ 2 & 5 & 16 & 0 & [0] \end{pmatrix}$$

Regel: Sei  $m$  das Minimum aller nicht überdeckten Zahlen (hier:  $m = 1$ ). Dann subtrahiere  $m$  von allen nicht überdeckten Elementen und addiere  $m$  zu allen Kreuzungspunkten ( $\Leftrightarrow$  zu allen gestrichenen Zeilen und Spalten  $\frac{m}{2}$  addieren und von allen nicht gestrichenen Zeilen und Spalten  $\frac{m}{2}$  subtrahieren).

Damit entsteht ein Problem, für das nach Satz 12 die gleiche Zuordnung optimal ist und das mindestens eine neue Null besitzt. Nun kann nach einer neuen vollständigen Zuordnung, d.h. nach alternierenden Pfaden gesucht werden.

Für das Beispiel gilt:

$$\begin{pmatrix} [0] & 2 & 13 & 0 & 0 \\ 22 & 2 & 0 & [0] & 1 \\ 8 & 0 & [0] & 20 & 3 \\ 0 & 1 & 2 & 0 & 23 \\ 3 & 5 & 16 & 0 & [0] \end{pmatrix} \rightarrow \begin{pmatrix} 0 & 2 & 13 & [0] & 0 \\ 22 & 2 & [0] & 0 & 1 \\ 8 & [0] & 0 & 20 & 3 \\ [0] & 1 & 2 & 0 & 24 \\ 3 & 5 & 16 & 0 & [0] \end{pmatrix}$$

Die optimale volle Zuordnung ergibt sich durch Rollentausch der Nullen auf dem alternierenden Pfad. Der ZF-Wert ist nun  $4 + 2 + 2 + 6 + 2 = 16$ .

Erstauszeichnung möglichst geschickt vornehmen:

- zuerst nur die Zeile und Spalten mit nur einer Null
- dann jeweils möglichst eine Zeile oder Spalte auszeichnen, für die eine Auszeichnung zwingend ist.

**Algorithmus:** *Ungarische Methode*

1. Zeile- und Spaltenreduktion;
2. Möglichst weitgehende Auszeichnung von Nullen;
3. CALL Alternierende Wege;  
WHILE noch keine vollständige Zuordnung DO  
CALL Überdeckung;  
 $m :=$  Minimum aller nicht gestrichenen Zahlen;  
subtrahiere  $m$  von allen nicht gestrichenen Zahlen;  
addiere  $m$  zu allen Kreuzungspunkten;  
nimm eine möglichst weitgehende Auszeichnung von Nullen vor;  
CALL Alternierende Wege;

Der Algorithmus bricht nach endlich vielen Schritten mit einer Optimallösung ab. Der Rechenaufwand beträgt  $O(n^3)$ .

## 10 Transport- und Umladeproblem

Vergleiche einführendes Beispiel in Abschnitt 1.2.

## 10.1 Das Transportproblem

Beim Transportproblem handelt es sich um einen Spezialfall des Umladeproblems. Es ist nur ein direkter Transport des Gutes von den Angebots- zu den Nachfrageorten möglich (es gibt also insbesondere keine Umladeknoten) und die transportierten Mengen sind nicht nach oben beschränkt.

### 10.1.1 Modellierung

Problem:

Es gibt  $m$  Ausgangsorte  $A_i$ ,  $i = 1, \dots, m$  (Lager, Produzenten, Firmen), in denen ein Erzeugnis (homogenes Gut) zur Verfügung steht. Die in  $A_i$  verfügbare Menge sei  $a_i > 0$  Mengeneinheiten.

Es gibt  $n$  Bestimmungsorte  $B_j$ ,  $j = 1, \dots, n$  (Kunden, Verbraucher, Lager), die einen Bedarf von  $b_j > 0$  Mengeneinheiten an diesem Erzeugnis haben.

Die Belieferung erfolgt jeweils direkt von einem  $A_i$  zu einem  $B_j$ . Dabei fallen für jede Einheit des transportierten Gutes Transportkosten (z.B. bzgl. der Entfernung) in Höhe von  $c_{ij}$  an. Die  $m \times n$ -Matrix  $C$  mit den Elementen  $c_{ij}$  wird Kostenmatrix genannt. Ist ein Transport von  $A_i$  nach  $B_j$  nicht möglich oder erwünscht, so kann man dies durch die Wahl eines genügend großen Kostenkoeffizienten  $c_{ij}$  erzwingen. Der Gesamtbedarf ist gleich dem Gesamtaufkommen, d.h.  $\sum_{i=1}^m a_i = \sum_{j=1}^n b_j$  (Ausgeglichenheit).

Die Aufgabe besteht darin, das Aufkommen an jedem Ausgangsort so zu den Bedarfsorten zu transportieren, dass der Bedarf jeweils komplett gedeckt wird und der Transportplan minimale Kosten besitzt.

Modell:

Sei  $x_{ij}$  die transportierte Menge des Erzeugnisses von  $A_i$  nach  $B_j$  (in Mengeneinheiten).

$$\begin{aligned}
 \min \quad & z = \sum_{i=1}^m \sum_{j=1}^n c_{ij} \cdot x_{ij} \\
 \text{u.d.N.} \quad & \sum_{j=1}^n x_{ij} = a_i, \quad \text{für } i = 1, \dots, m \\
 & \sum_{i=1}^m x_{ij} = b_j, \quad \text{für } j = 1, \dots, n \\
 & x_{ij} \geq 0, \quad \text{für } i = 1, \dots, m; j = 1, \dots, n \\
 & \text{bzw. } \in \mathbb{R}^{\geq 0}
 \end{aligned} \tag{15}$$

Die Ausgeglichenheit des Problems, also  $\sum_{i=1}^m a_i = \sum_{j=1}^n b_j$ , gilt o.B.d.A. da dies durch die Einführung eines fiktiven Ausgangs- bzw. Bestimmungsortes erreicht werden kann.

Beispiel:

	$B_1$	$B_2$	$B_3$	$B_4$
$A_1$	1	8	4	7
$A_2$	9	0	5	7
$A_3$	3	6	8	1

sei die gegebene Kostenmatrix  $C$ . Dann ist

	$B_1$	$B_2$	$B_3$	$B_4$	$a_i$
$A_1$	$x_{11}$	$x_{12}$	$x_{13}$	$x_{14}$	20
$A_2$	$x_{21}$	$x_{22}$	$x_{23}$	$x_{24}$	25
$A_3$	$x_{31}$	$x_{32}$	$x_{33}$	$x_{34}$	40
$b_j$	10	25	15	35	85

die Matrix der Variablen und des Aufkommens/Bedarfs.

Wegen  $0 \leq x_{ij} \leq \min(a_i, b_j)$  für  $i = 1, \dots, m$ ;  $j = 1, \dots, n$  ist der zulässiger Bereich des Transportproblems ein nichtleeres konvexes Polytop und somit besitzt das Transportproblem stets eine optimale Lösung.

Setzt man im Transportproblem  $m = n$  und  $a_i = b_j = 1$  ( $i, j = 1, \dots, n$ ), so erhält man das Zuordnungsproblem aus Abschnitt 9.

### 10.1.2 Eröffnungsheuristiken

#### 1. Greedy (Matrixminimumregel)

Belege minimales Element maximal, d.h. transportiere soviel wie möglich über den billigsten Weg.

	$B_1$	$B_2$	$B_3$	$B_4$	$a_i$
$A_1$	10	–	10	–	20
$A_2$	–	25	–	–	25
$A_3$	–	–	5	35	40
$b_j$	10	25	15	35	85

$m + n = 7$  ( $m + n - 1$  BV), hier 5 Variablen  $> 0$ .

Zielfunktion:  $10 \cdot 1 + 10 \cdot 4 + 25 \cdot 0 + 5 \cdot 8 + 1 \cdot 35 = 125$

Greedy hat keine Optimalitätsgarantie.

#### 2. NordWestEcken-Regel

- Beginne in der NWE und setze  $x_{11} = \min(a_1, b_1)$ .
- Falls  $x_{11} = a_1$ , setze  $x_{21} = \min(a_2, b_1 - x_{11})$ .  
Falls  $x_{11} = b_1$ , setze  $x_{12} = \min(a_1 - x_{11}, b_2)$ .  
Ist  $x_{11} = a_1 = b_1$ , setze zusätzlich  $x_{12} = 0$  oder  $x_{21} = 0$  als Basiseintrag („Basis-Null“).
- Iteriere bis alle  $x_{mn}$  festgelegt sind. Die entsprechende Lösung ist eine ZBL für das Transportmodell.

	$B_1$	$B_2$	$B_3$	$B_4$	$a_i$
$A_1$	10	10	–	–	20
$A_2$	–	15	10	–	25
$A_3$	–	–	5	35	40
$b_j$	10	25	15	35	85

- $x_{11} = \min(a_1, b_1) = \min(20, 10) = 10 = b_1$

b)  $x_{12} = \min(a_1 - x_{11}, b_2) = \min(10, 25) = 10$   
usw.

ZF-Wert =  $10 \cdot 1 + 10 \cdot 8 + 5 \cdot 10 + 5 \cdot 8 + 1 \cdot 35 = 215$

Ist für dieses Beispiel schlechter als Greedy, es werden bei der NordWestEcken-Regel die Kosten nicht berücksichtigt. Eine Verallgemeinerung ist allerdings nicht möglich.

### 3. VAM-Heuristik (Vogel'sche Approximationsmethode)

$I$ : Indexmenge der noch nicht erfüllten Zeilenbedingungen.

$J$ : Indexmenge der noch nicht erfüllten Spaltenbedingungen.

- Bestimme für alle noch nicht erfüllten Zeilen und Spalten die Opportunitätskosten, die gleich der Differenz ( $D_i$  bzw.  $\bar{D}_j$ ) zwischen den zweitniedrigsten und den niedrigsten Einheitstransportkosten  $c_{ij}$  ( $i \in I, j \in J$ ) sind. (Bei der statischen VAM-Heuristik werden die Opportunitätskosten nur einmal zu Beginn berechnet.)
- Bestimme in der Zeile oder Spalte mit der maximalen Differenz das Feld  $(i, j)$  mit dem kleinsten  $c_{ij}$ . Durch diese Wahl wird nur wenig kaputt gemacht (siehe Beispiel). Falls die maximale Differenz nicht eindeutig ist, so wähle eine dieser Zeilen oder Spalten beliebig aus.
- Im Feld  $(i, j)$  setze  $x_{ij} = \min(a_i, b_j)$ , sodass eine Spalten- oder Zeilenbedingung erfüllt ist. Streiche die erfüllte Zeile oder Spalte und reduziere die nicht erfüllte Forderung ( $a_i$  oder  $b_j$ ) um den Wert  $x_{ij}$ . Falls beide Bedingungen erfüllt sind, streiche entweder die Zeile oder die Spalte (in der übrig gebliebenen Zeile oder Spalte entsteht eine  $BV=0$ ).
- Überprüfe, ob alle Zeilen- oder Spaltennebenbedingungen erfüllt sind. Wenn ja „STOP“, sonst „gehe zu a)“.

Beispiel:

	$B_1$	$B_2$	$B_3$	$B_4$
$A_1$	1	8	4	7
$A_2$	9	0	5	7
$A_3$	3	6	8	1

- a) Bestimmung der Opportunitätskosten  $D_i, \bar{D}_j$  für alle Zeile  $i \in I$  und alle Spalten  $j \in J$ .

	$B_1$	$B_2$	$B_3$	$B_4$	$a_i$	$D_i$
$A_1$	1	8	4	7	20	3
$A_2$	9	0	5	7	25	5
$A_3$	3	6	8	1	40	2
$b_j$	10	25	15	35	85	
$\bar{D}_j$	2	6	1	6		

- b) Höchste Opportunitätskosten sind  $\bar{D}_j = \bar{D}_2 = \bar{D}_4 = 6$ .  
Nicht eindeutig: Spalte 2 wird gewählt.  
Feld  $(2, 2)$  führt zu minimalen  $c_{ij} = 0 = c_{22}$ .

- c)  $x_{22} = \min(25, 25) = 25$



Beide Bedingungen sind erfüllt. Wir streichen Spalte, Index  $j = 2$  aus  $J$  und setzen  $a_2^{\text{neu}} = a_2^{\text{alt}} = x_{22} = 0$  (Basisnull).

Zurück zu a)

a)  $I = \{1, 2, 3\}$ ,  $J = \{1, 3, 4\}$

	$B_1$	$B_2$	$B_3$	$B_4$	$a_i$	$D_i$
$A_1$	1	–	4	7	20	3
$A_2$	9	$(25)^1$	5	7	0	2
$A_3$	3	–	8	1	40	2
$b_j$	10	0	15	35	60	
$\bar{D}_j^{(2)}$	2	–	1	6		

b)  $\bar{D}_j^{(2)} = 6$  größte Opportunitätskosten  
Feld (3, 4) kleinstes  $c_{ij} = 1$

c) Zuweisung:  $x_{34} = 35$ ,  $a_3^{\text{neu}} = 40 - 35 = 5$   
Index  $j = 4$  aus  $J$  streichen. Zurück zu a)

a)  $I = \{1, 2, 3\}$ ,  $J = \{1, 3\}$

	$B_1$	$B_2$	$B_3$	$B_4$	$a_i$	$D_i$
$A_1$	1	–	4	–	20	3
$A_2$	9	$(25)^1$	5	–	0	4
$A_3$	3	–	8	$(35)^2$	5	5
$b_j$	10	0	15	0	25	
$\bar{D}_j^{(3)}$	2	–	1	–		

b)  $\max_{i,j} (D_i^{(3)}, \bar{D}_j^{(3)}) = 5 = D_3^{(3)}$   
Feld (3, 1) kleinstes  $c_{ij} = 3$

c) Zuweisung:  $x_{31} = 5$ ,  $b_1^{\text{neu}} = 10 - 5 = 5$   
Zurück zu a) usw.

5. Iteration liefert eine Lösung (hier sogar die optimale, obwohl nur Eröffnungsverfahren).

### 10.1.3 Die MODI-Methode

Die MODI-Methode (modifizierte Distributionsmethode) ist eine abgeänderte Version der Netzwerk-Simplexmethode.

O.B.d.A. gilt  $\sum_{i=1}^m a_i = \sum_{j=1}^n b_j$  das ausgeglichene Problem und  $a_i, b_j \geq 0$ .

Darstellung als LP

		$x_{11}$	$x_{12}$	$\dots$	$x_{1n}$	$x_{21}$	$x_{22}$	$\dots$	$x_{2n}$	$\dots$	$x_{m1}$	$x_{m2}$	$\dots$	$x_{mn}$	RS
Zeile	1	1	1	$\dots$	1	0	0	$\dots$	0	$\dots$	0	0	$\dots$	0	$a_1$
	2	0	0	$\dots$	0	1	1	$\dots$	1	$\dots$	0	0	$\dots$	0	$a_2$
	$\dots$														
	$m$	0	0	$\dots$	0	0	0	$\dots$	0	$\dots$	1	1	$\dots$	1	$a_m$
Zeile	1	1	0	$\dots$	0	1	0	$\dots$	0	$\dots$	1	0	$\dots$	0	$b_1$
	2	0	1	$\dots$	0	0	1	$\dots$	0	$\dots$	0	1	$\dots$	0	$b_2$
	$\dots$														
	$n$	0	0	$\dots$	1	0	0	$\dots$	1	$\dots$	0	0	$\dots$	1	$b_n$

Für die Teile 1) und 2) gilt

- 1)  $\sum_{j=1}^n x_{ij} = a_i$  für  $i = 1, \dots, m$
- 2)  $\sum_{i=1}^m x_{ij} = b_j$  für  $j = 1, \dots, n$

Das Tableau besteht aus  $m \cdot n$  Spalten und  $m + n$  Zeilen. Eine Lösung mit dem Simplexverfahren hätte daher als Basis eine  $(m + n) \times (m + n)$ -Teilmatrix vom Rang  $m + n - 1$ . Da die Summe der Zeilen 1 bis  $n$  gleich der Summe der Zeilen  $m + 1$  bis  $m + n$  ist (Ausgeglichenheit), sind die  $m + n$  Zeilen linear abhängig; streicht man die erste Zeile, so ist die Determinante der aus den Spalten mit den Nummern  $1, \dots, n, n + 1, 2n + 1, \dots, (m - 1)n + 1$  gebildeten Untermatrix  $\neq 0$ . Wegen der Ausgeglichenheit hat auch die um die rechte Seite erweiterte Matrix den Rang  $m + n - 1$ . Daher ist genau eine Gleichung redundant, es genügen also  $m + n - 1$  BV.

Beispiel:

$$\begin{array}{cccccccccccc}
 x_{11}+ & x_{12}+ & x_{13}+ & x_{14} & & & & & & & & & & & & & = 20 \\
 & & & & & & x_{21}+ & x_{22}+ & x_{23}+ & x_{24} & & & & & & & = 25 \\
 & & & & & & & & & & & & x_{31}+ & x_{32}+ & x_{33}+ & x_{34} & = 40 \\
 x_{11}+ & & & & & & x_{21}+ & & & & & & x_{31} & & & & = 10 \\
 & x_{12}+ & & & & & & x_{22}+ & & & & & & x_{32} & & & = 25 \\
 & & x_{13}+ & & & & & & x_{23}+ & & & & & & x_{33} & & = 15 \\
 & & & x_{14}+ & & & & & & x_{24}+ & & & & & & x_{34} & = 35 \\
 \uparrow & & \uparrow & & & & \uparrow & \uparrow & & & & & \uparrow & \uparrow & & & \\
 1 & & 2 & & & & 3 & 4 & & & & & 5 & 6 & & & 
 \end{array}$$
  

$$\begin{array}{cccccccc}
 x_{11}+ & x_{13} & & & & & & = 20 \\
 & x_{13}+ & & & x_{23}+ & x_{33} & & = 15 \\
 & & & x_{22}+ & x_{23} & & & = 25 \\
 & & & & & & x_{33}+ & x_{34} & = 40 \\
 & x_{11} & & & & & & = 10 \\
 & & & x_{22}+ & & & & = 25 \\
 & & & & & & & x_{34} & = 35
 \end{array}$$

alle NBV = 0

$$x_{11} = 10, x_{34} = 35, x_{13} = 10, x_{23} = 0, x_{22} = 25, x_{33} = 5$$

$4 + 3 - 1 = 6$  BV wegen Ausgeglichenheit.

Aus der speziellen Struktur der Koeffizientenmatrix  $A$  (nur 0'en und 1'en, speziell angeordnet) können vereinfachende Schlussfolgerungen gezogen werden.

1.  $B^{-1}$  enthält ausschließlich  $+1, -1, 0$ .
2.  $B^{-1} \cdot b = x_B$  ganzzahlig, falls  $b$  ganzzahlig (d.h. Ganzzahligkeit muss nicht explizit gefordert werden).
3. Ein vereinfachtes Tableau ist möglich, da die Struktur von  $A, B, B^{-1}$  bekannt ist.
4. Es genügt, auf jeder Iteration die Basis, die Basislösung und die  $\Delta z_j$  der NBV zu bestimmen!

### Transporttableau

	$B_1$	$B_2$	$\cdots$	$B_n$	$a_i$
$A_1$	$x_{11}$	$x_{12}$	$\cdots$	$x_{1n}$	$a_1$
$A_2$	$x_{21}$	$x_{22}$	$\cdots$	$x_{2n}$	$a_2$
$\cdots$	$\cdots$	$\cdots$		$\cdots$	$\cdots$
$A_m$	$x_{m1}$	$x_{m2}$	$\cdots$	$x_{mn}$	$a_m$
$b_i$	$b_1$	$b_2$	$\cdots$	$b_n$	$\sum a_i = \sum b_j$

Zulässige Ausgangsbasislösung: NordWestEcken-Regel

Verbesserungsverfahren: MODI (Modified Distribution Method)

Grundidee: Verwendung des DM zu Modell (15)

$$\begin{aligned} \max \quad Z &= \sum_{i=1}^m a_i u_i + \sum_{j=1}^n b_j v_j \\ u_i + v_j &\leq c_{ij}, \quad \text{für } i = 1, \dots, m; \quad j = 1, \dots, n \end{aligned} \quad (16)$$

Wobei  $u_i, v_j$  die dualen Variablen der Restriktionen und  $a_i, b_j$  die ZF-Koeffizienten sind. Die dualen Strukturvariablen sind wegen der Gleichungsrestriktionen von (15) unbeschränkt, d.h. keine NNB für  $u_i, v_j$ , da im PM nur Gleichheitsnebenbedingungen, also  $u_i, v_j \in \mathbb{R}$ . Weil das Gleichungssystem des PM unterbestimmt ist, kann einer dualen Variablen ein beliebiger Wert zugeordnet werden.

Wegen Satz 8 (Complementary Slackness Condition) gilt für die BV:  $u_i + v_j = c_{ij}$ .

Da die Basis  $m + n - 1$  BV enthält, ist das Gleichungssystem  $u_i + v_j = c_{ij}$  unterbestimmt. Man legt  $u_1 = 0$  (oder  $v_1 = 0$ ) fest und löst danach das Gleichungssystem für alle BV.

Für die aufzunehmende NBV benötigt man  $\Delta z_{ij}$ , die hier gleich  $\Delta z_{ij} = c_{ij} - (u_i + v_j)$  sind.

Optimalitätsbedingung:  $\Delta z_{ij} \geq 0$  für alle NBV

Aufnahme-Regel: Diejenige NBV  $x_{rs}$  in die Basis aufnehmen, für die  $\Delta z_{rs} = \min_{i,j} \Delta z_{ij} < 0$  gilt (Kleinste-Kosten-Regel).

Basistausch (zu eliminierende BV): Im Transporttableau Pfad aus denjenigen BV bestimmen, deren Wert sich bei Erhöhung des Wertes der aufzunehmenden NBV ändert, damit die Restriktionen erhalten bleiben (Stepping-Stone-Path, SPP). Dieser Pfad ist eindeutig. Die entsprechenden BV seien  $x_{ij}^B$ . Durch Hinzufügen der NBV  $x_{rs}$  entsteht eine Schleife. Dann gilt:

1. Die Werte der Variablen  $x_{ij}^B$  im SSP ändern sich entweder um  $+x_{rs}$  oder  $-x_{rs}$ .
2. Das Vorzeichen alterniert (also reicht eine Markierung mit + bzw. -).

Sobald bei Erhöhung der NBV eine mit  $-$  markierte ursprünglichen BV einen Wert  $< 0$  annimmt, würde Unzulässigkeit eintreten.

$\Rightarrow$  ausscheidende BV:  $x_{kl}^B$  gemäß  $\Theta = \min_{i,j} x_{ij}^B = x_{kl}^B$  bestimmen,

wobei nur mit  $-$  markierte  $i, j$  betrachtete werden, da nichts negatives produziert werden kann.  $\Theta$  gibt auch den Wert der aufzunehmenden NBV an (Pivotisierungsprozess).

Beispiel:

Die NordwestEcken-Regel liefere die folgende Ausgangsbasislösung

	$B_1$	$B_2$	$B_3$	$B_4$	$a_i$
$A_1$	10	10			20
$A_2$		15	10		25
$A_3$			5	35	40
$b_j$	10	25	15	35	85

NBV = 0 (leeren Felder).

Nun wird für alle BV die Gleichung  $u_i + v_j = c_{ij}$  mit  $v_1 = 0$  gelöst:

$$\begin{aligned}
 u_1 + v_1 = c_{11} = 1 \text{ wegen } v_1 = 0 &\rightarrow u_1 = 1 \\
 u_1 + v_2 = c_{12} = 8 &\rightarrow v_2 = 7 \\
 u_2 + v_2 = c_{22} = 0 &\rightarrow u_2 = -7 \\
 u_2 + v_3 = c_{23} = 5 &\rightarrow v_3 = 12 \\
 u_3 + v_3 = c_{33} = 8 &\rightarrow u_3 = -4 \\
 u_3 + v_4 = c_{34} = 1 &\rightarrow v_4 = 5
 \end{aligned}$$

Berechnung der  $\Delta z_{ij} = c_{ij} - (u_i + v_j)$  für alle NBV.

Im Schema gilt:

- in Klammern stehen die  $c_{ij}$
- alle BV entsprechenden Felder mit  $\Delta z_{ij} = 0$  belegen
- $u_i$  und  $v_j$  Spalte bzw. Zeile anfügen

$\rightarrow \Delta z_{ij}$  für NBV berechnen

	$B_1$	$B_2$	$B_3$	$B_4$	$u_i$		$B_1$	$B_2$	$B_3$	$B_4$	$a_i$	
$A_1$	(1) 0	(8) 0	(4) -9	(7) 1	1		$A_1$	10	10		20	
$A_2$	(9) 16	(0) 0	(5) 0	(7) 9	-7		$A_2$		15	10	25	
$A_3$	(3) 7	(6) 3	(8) 0	(1) 0	-4		$A_3$			5	35	40
$v_j$	0	7	12	5			$b_j$	10	25	15	35	85

Wegen  $\Delta z_{13} = -9 < 0$  ist die BL nicht optimal.  $x_{13}$  in die BL aufnehmen! Die  $x_{13}$  als NBV enthaltene Schleife zur BL ist im rechten Transporttableau gezeigt.

$$\Theta = \min x_{ij}^B = 10$$

Es entsteht die Schleife für  $x_{12} = x_{23} = 10$ , wähle  $x_{12}$ .

Pivotisierung:

Ändern der Elemente der Schleife im Sinne der Markierung um 10 (dadurch entsteht die nächste zulässige BL).

	$B_1$	$B_2$	$B_3$	$B_4$	$a_i$
$A_1$	10		10		20
$A_2$		25	0		25
$A_3$			5	35	40
$b_j$	10	25	15	35	85

Es entsteht eine entartete BL, da weniger als  $m + n - 1 = 6$  BV  $> 0$  sind. Frei entscheiden, ob  $x_{12}$  oder  $x_{23}$  als BV (Basisnull) behandelt wird.

Nach einer weiteren Iteration folgt die optimale Lösung:

$$x_{11} = 5, x_{13} = 15, x_{22} = 25, x_{23} = 0, x_{31} = 5, x_{34} = 35$$

$$\text{mit } Z = 1 \cdot 5 + 4 \cdot 15 + 3 \cdot 5 + 1 \cdot 35 = 115$$

Diese Lösung wird bei diesem Beispiel auch durch die VAM-Heuristik geliefert.

## 10.2 Das Umladeproblem

Beim Umladeproblem ist der kostengünstigste Transport eines Gutes von gewissen Angebotsorten (Anbietern oder Produzenten eines Gutes) über sogenannte Umladeorte (z.B. Zwischenlager oder Montageorte) zu Nachfrageorten (Verbrauchern des Gutes) gesucht. Beispiel: Wasserversorgungsnetz mit Wasserwerk, Pumpstationen (Verzweigungen) und Verbrauchern.

Formulierung des Umladeproblems als LP auf einem Netzwerk  $N$  mit Kapazitäten und Kosten:

Netzwerk  $N = (V, E)$

- Jedem Ort (Angebots-, Umlade-, Nachfrageort) entspricht ein Knoten  $i$ . Knotenmenge  $V = \{1, \dots, i, \dots, n\}$ .
- Ist ein direkter Transport des Gutes von Ort  $i$  zu Ort  $j$  möglich wird ein Pfeil  $(i, j)$  eingeführt. Pfeilmenge  $E = \{(i, j) | i, j \in V, \text{ direkter Transport von } i \text{ nach } j \text{ möglich}\}$ .
- Das Netzwerk  $N = (V, E)$  sei schwach zusammenhängend (sonst läge für jede schwache Zusammenhangskomponente ein Umladeproblem vor).

Jedem Knoten  $i$  entspricht eine Menge  $a_i$ :

„Vom Ort  $i$  abtransportierte Nettomenge (abtransportierte - hintransportierte Menge)“

Dann gilt für:

Angebotsknoten  $i$ :  $a_i > 0$

Umladeknoten  $i$ :  $a_i = 0$

Nachfrageknoten  $i$ :  $a_i < 0$  ( $-a_i > 0$  Bedarf am Ort  $i$ )

Beispiel: Netzwerk  $N$

Angebotsorte:  $i = 1, 2$

Umladeorte:  $i = 3, 4, 5$

Nachfrageorte:  $i = 6, 7, 8, 9$

Von einem Nachfrageknoten kann etwas abtransportiert, zu einem Angebotsknoten etwas hintransportiert werden (vgl. Wassernetzwerk).

Variablen:

Flusswert (-variable)  $x_{ij}$ : auf dem Pfeil  $(i, j)$  transportierte Menge.

Jeder Pfeil besitzt eine Maximalkapazität  $\kappa_{i,j}$  zur Aufnahme des Mengenflusses. Ist die transportierte Menge nach oben nicht beschränkt, setze  $\kappa_{ij} = \infty$ . Da die transportierte Menge von  $i$  nach  $j$  nichtnegativ ist, gelten die folgenden Nebenbedingungen

$$0 \leq x_{ij} \leq \kappa_{ij} \quad \text{für alle } (i, j) \in E$$

Als Flussbedingungen (-gleichungen) gelten:

$$\sum_{j \in S(i)} x_{ij} - \sum_{k \in P(i)} x_{ki} = a_i, \quad i = 1, \dots, n$$

Hierbei ist  $S(i)$  die Menge aller Nachfolgerknoten von  $i$ , d.h.  $j \in S(i)$ , falls  $(i, j) \in E$  und  $P(i)$  die Menge aller Vorgängerknoten von  $i$ , d.h.  $k \in P(i)$ , falls  $(k, i) \in E$ .

Für den Knoten 4 im Beispielnetzwerk gilt:

$$(x_{43} + x_{47} + x_{46} + x_{48} + x_{49} + x_{45}) - (x_{14} + x_{24}) = a_4 = 0 \quad (\text{Umladeknoten})$$

Die Kosten für den Transport einer ME des Gutes auf dem Pfeil  $(i, j)$  seien  $c_{ij}$ .

Damit folgt das *Umladeproblem* als:

$$\begin{aligned} \min & \quad \sum_{(i,j) \in E} c_{ij} \cdot x_{ij} \\ \text{u.d.N.} & \quad \sum_{j \in S(i)} x_{ij} - \sum_{k \in P(i)} x_{ki} = a_i, \quad i = 1, \dots, n \\ & \quad 0 \leq x_{ij} \leq \kappa_{ij}, \quad \text{für alle } (i, j) \in E \\ \text{o.B.d.A.} & \quad \sum_{i=1}^n a_i = 0, \text{ ansonsten fiktiven Nachfrage- bzw. Angebotsort einführen} \end{aligned}$$

Das Umladeproblem kann mit der sogenannten Netzwerksimplexmethode [3] gelöst werden.

## 11 Das Rucksackproblem

Dem Rucksackproblem liegt folgende anschauliche Aufgabenstellung zugrunde: Ein Wanderer kann in seinem Rucksack verschiedene Gegenstände einpacken, wobei die einzelnen Gegenstände unterschiedliche Gewichte und „Werte“ (ein großer Wert entspricht einem großen Nutzen) haben. Gesucht ist die optimale Rucksackfüllung, d.h. eine Füllung mit maximalem Gesamtwert, wobei ein vorgegebenes Gesamtgewicht des Rucksacks nicht überschritten werden darf.

Gewissen Frachtladeprobleme lassen sich als Rucksackproblem formulieren, sowie Optimierungsprobleme bei der Programmplanung in der Einzelproduktion, wenn nur eine einzige Faktorbeschränkung im Produktions- oder Beschaffungsbereich auftritt. In diesem Fall entsprechen die Gegenstände den einzelnen Aufträgen, Wert und Gewicht eines Gegenstandes dem Deckungsbeitrag bzw. der Faktorinanspruchnahme des Engpass-Produktionsfaktors für den entsprechenden Auftrag und das Gesamtgewicht des Rucksacks der verfügbaren Kapazität des Engpass-Produktionsfaktors.

### 11.1 Anwendungsbeispiel

Man verfüge über 37.000€ (Gesamtgewicht), die man in 5 verschiedene Projekte (Gegenstände) investieren könne. Jedes Projekt könne entweder realisiert oder abgelehnt werden (also nicht mehrfach oder teilweise realisiert, Gegenstand einpacken oder nicht einpacken). Budgetreste können nicht zinsbringend angelegt werden.

Projekt (Gegenstand)	Investition € (Gewicht)	Jahresgewinn € (Wert)
1	17.000	11.000
2	16.000	14.000
3	21.000	16.000
4	8.000	8.000
5	12.000	7.000

Welche Projekte sind zu realisieren, um den jährlichen Gewinn zu maximieren?

Mathematisches Modell:

$$\begin{aligned} \max \quad & Z = 11x_1 + 14x_2 + 16x_3 + 8x_4 + 7x_5 \\ \text{s.d.} \quad & 17x_1 + 16x_2 + 21x_3 + 8x_4 + 12x_5 \leq 37 \end{aligned}$$

Allgemein:

$$\begin{aligned} \sum c_j x_j &\rightarrow \max \\ \sum a_j x_j &\leq b \\ x_j &\in \{0, 1\} \text{ für } j = 1, \dots, n \\ x_j &= \begin{cases} 1, & \text{falls in Projekt } j \text{ investiert wird} \\ 0, & \text{sonst} \end{cases} \end{aligned}$$

Alle  $x_j = 0$ , (leerer Rucksack) ist eine zulässige Lösung. Weitere Lösungen existieren genau dann, wenn der schwerste Gegenstand nicht schon das Gesamtgewicht des Rucksacks übersteigt. Überschreitet das Gewicht aller Gegenstände das Gesamtgewicht des Rucksacks nicht, ist die einzige optimale Lösung, alle Gegenstände in den Rucksack zu packen. Es kommt nur eine einzige Restriktion vor (d.h. Lösungsbereich ist eine konvexes Polyeder), aber die  $x_j$  sind aber nicht kontinuierlich sondern binär. Lösung des Problems mit dem

Simplexalgorithmus, wenn das Problem modifiziert/relaxiert wird.

Relaxiertes Problem:

$$\begin{aligned} \max \quad & Z = 11x_1 + 14x_2 + 16x_3 + 8x_4 + 7x_5 \\ \text{s.d} \quad & 17x_1 + 16x_2 + 21x_3 + 8x_4 + 12x_5 \leq 37 \\ & 0 \leq x_j \leq 1, \text{ für } j = 1, \dots, 5 \end{aligned}$$

Der Simplexalgorithmus liefert für dieses Problem die optimale Lösung:

$$\hat{x}_1 = \hat{x}_5 = 0, \hat{x}_2 = \hat{x}_4 = 1, \hat{x}_3 = 0, 81$$

Aufrunden:  $x_2 = x_4 = x_3 = 1$  ist nicht zulässig, da  $45 \not\leq 37$ .

Abrunden:  $x_2 = x_4 = 1, x_3 = 0$  ist nicht optimal.

Optimal ist:  $x_2 = x_3 = 1$ , ist durch Runden nicht zu erhalten.

Heuristische Lösung:

Konzept: Ordne Projekte nach fallender „Rentabilität“ und teile die Mittel zu, soweit sie reichen. D.h. die Gegenstände werden so durchnummeriert, dass  $\frac{c_1}{a_1} \geq \frac{c_2}{a_2} \geq \dots \geq \frac{c_n}{a_n}$  gilt, die Folge der „relativen Werte“ der Gegenstände also monoton fallend ist.

Projekt	$\frac{\text{Jahresgewinn}}{\text{Investition}}$	übrige Mittel
4	$\frac{8.000}{8.000} = 1,000$	29.000
2	$\frac{14.000}{16.000} = 0,875$	13.000
3	$\frac{16.000}{21.000} = 0,762$	-3.000
1	$\frac{11.000}{17.000} = 0,647$	
5	$\frac{7.000}{12.000} = 0,580$	

Der Jahresgewinn beläuft sich also auf  $8.000 + 14.000 = 22.000$  und es bleiben Investitionsmittel von 13.000 übrig. Dies entspricht der abgerundeten Lösung. Diese suboptimale Lösung lässt sich noch verbessern, wenn man unter den übrigen Gegenständen einige findet, deren Gesamtgewicht das noch verfügbare „Restgewicht“ (übrige Mittel) nicht überschreitet (hier  $x_5$ ).

Idee: Insgesamt existieren  $2^5$  Lösungen (sowohl zulässige als auch unzulässige). Die Lösungen lassen sich als binärer Baum darstellen. Eine Lösung besteht dann aus einem Weg von der Wurzel zu einem Blattknoten (hier  $2^5 = 32$  mögliche Wege). Das Verfahren ist jedoch ineffizient, bei 50 Gegenständen existieren bereits  $2^{50}$  potenzielle Lösungen.

## 11.2 Zwei Branch and Bound Algorithmen für das Rucksackproblem

Verzweigen und Abbrechen (Begrenzen)

### 1. Algorithmus:

Idee: Teilbaum generieren, Knoten so früh wie möglich „als nicht optimal“ erkennen und



terminieren (d.h. auf die Erzeugung ganzer Teilbäume verzichten), Optimalitätgarantie (es wird nichts abgebrochen, was nicht noch optimal werden kann).

Bound (Beschränkungsschritt, Schranke):

- a) noch erreichbarer Gewinn  $b_i$
- b) verfügbares Kapital  $f_i$

Branching-Regel (Verzweigung): Knoten mit höchster Schranke  $b_i$  zuerst entwickeln, da Gewinn maximiert werden soll.

Terminierung:

- a) verfügbares Kapital reicht für kein weiteres Projekt (Knoten des Teilbaums sind nicht zulässig)
- b) erreichter Gewinn ist niedriger als bereits realisierte Gewinn (eine bessere Lösung ist bereits bekannt)

Erläuterungen zum Baum:

1. Wurzelknoten 0 wird generiert  
 $b_0 = 56$  (alle Variablen = 1) theoretisches Maximum (Relaxation)  
 $f_0 = 37$  noch keine Variable ist festgelegt
2.  $x_1 = 1 \rightarrow b = 56, f = 20$  (56 noch erreichbar)  
 $x_1 = 0 \rightarrow b = 45, f = 37$  (45 noch maximal erreichbar)  
 Beide Knoten werden generiert und erhalten die Nummern 1 und 2.  
 Nach Branching-Regel wird Knoten 2 zuerst entwickelt.
3. 3 mit  $\binom{42}{20}$  und 4 mit  $\binom{56}{4}$  erzeugen, zuerst 4 weiterentwickeln.
4. 5 und 6 erzeugen, 6 kann terminiert werden, da im Teilbaum mit Wurzel 6 keine zulässige Lösung liegen kann.

5. 5 mit  $\binom{40}{4}$ , weiterhin „aktiv“ (d.h. noch nicht terminiert bzw. entwickelt) ist 1 mit  $\binom{45}{37}$ . Wegen Brachning-Strategie zuerst 1 weiterentwickeln, 5 bleibt „aktiv“.

usw.

Knoten 17 entspricht der optimalen Lösung

$$b_{17} = 30, b_{17} = 0$$

$$x_1 = x_4 = x_5 = 0 \quad x_2 = x_3 = 1$$

Aber zu diesem Zeitpunkt ist noch nicht bekannt, das 17 die optimale Lösung ist, denn zum Erzeugungszeitpunkt von 17 sind noch die Knoten 15 mit  $\binom{32}{4}$  und 7 mit  $\binom{31}{37}$  aktiv. Zur Bestätigung der Optimalität des Knotens 17 müssen noch die Knoten 19 und 26 erzeugt und bewertet werden.

Die optimale Lösung ist in diesem Beispiel die erste erzeugte vollständige Lösung (Blattknoten). Alle anderen Lösungsversuche werden „zurückgestellt“ oder als „unzulässig“ abgebrochen.

## 2. Algorithmus:

LP (P) mit nur einer  $\leq$ -Restriktion, nichtnegative Koeffizienten und 0-1 Variablen und das zugehörige relaxierte Problem (R)

$$\begin{aligned} \max z = & 20x_1 + 12x_2 + 8x_3 + 15x_4 \\ & 7x_1 + 9x_2 + 5x_3 + 8x_4 \leq 14 & \text{(P)} \\ & 0 \leq x_i \leq 1, \text{ ganzzahlig } i = 1, \dots, 4 \end{aligned}$$

$$\begin{aligned} \max z = & 20x_1 + 12x_2 + 8x_3 + 15x_4 \\ & 7x_1 + 9x_2 + 5x_3 + 8x_4 \leq 14 & \text{(R)} \\ & 0 \leq x_i \leq 1, i = 1, \dots, 4 \end{aligned}$$

Lösung von (R):

1. Variablen nach fallenden Quotienten (ZF-Koeffizient/Koeffizient in der Restriktion) umordnen.
2. In der Reihenfolge dieser Ordnung jede Variable auf den größten zulässigen Wert setzen. Wegen  $\frac{20}{7} \geq \frac{15}{8} \geq \frac{8}{5} \geq \frac{12}{9}$  also

$$\begin{aligned} \max z = & 20x_1 + 15x_4 + 8x_3 + 12x_2 \\ & 7x_1 + 8x_4 + 5x_3 + 9x_2 \leq 14 \\ & 0 \leq x_i \leq 1, i = 1, \dots, 4 \end{aligned}$$

Lösung:  $x_1^* = 1, x_4^* = \frac{7}{8}, x_3^* = x_2^* = 0, z = 33\frac{1}{8}$

Bei ganzzahligen Koeffizienten ist die obere Schranke  $\bar{z} = 33$ , d.h. mehr ist auf keinen Fall erreichbar.

$x^*$  ist keine zulässige Lösung von (P)

→ Aufspaltung/Verzweigung von (R) in (R1) und (R2), Restriktionen:  $x_4 = 0, x_4 = 1$ .

$$\begin{aligned} \text{(R1)} \quad \max z = & 20x_1 + 8x_3 + 12x_2 \\ & 7x_1 + 5x_3 + 9x_2 \leq 14 \\ & x_4 = 0 \\ & 0 \leq x_i \leq 1, i = 1, 2, 3 \end{aligned}$$

Lösung:  $x_4^* = 0$ ,  $x_1^* = x_3^* = 1$ ,  $x_2^* = \frac{2}{9}$ ;  $z^* = 30\frac{2}{3}$  ist keine Lösung von (P).

$$(R2) \quad \begin{aligned} \max z = & 15 + 20x_1 + 8x_3 + 12x_2 \\ & 7x_1 + 5x_3 + 9x_2 \leq 6 \\ & x_4 = 1 \\ & 0 \leq x_i \leq 1, \quad i = 1, 2, 3 \end{aligned}$$

Lösung: wie bisher:  $x_1^* = \frac{6}{7} \Rightarrow x_3^* = x_2^* = 0$   
 oder das Verfahren erkennt, dass in jeder Lösung von (R2), die auch Lösung von (P) ist,  $x_1 = x_2 = 0$  sein muss (da die jeweiligen Koeffizienten größer als die rechte Seite sind).  
 $\rightarrow$  Einsetzen einer Prüfroutine, die den Verzweigungsprozess wesentlich abkürzen kann.

$$(R2') \quad \begin{aligned} \max z = & 15 + 8x_3 \\ & 5x_3 \leq 6 \\ & x_1 = 0, \quad x_2 = 0 \\ & x_4 = 1 \\ & 0 \leq x_3 \leq 1 \end{aligned}$$

Lösung:  $x_3^* = x_4^* = 1$ ,  $x_1^* = x_2^* = 0$ ,  $z^* = 23$  (auch zulässige Lösung für (P))  
 $\Rightarrow \underline{z} = c_u = 23$  (untere Schranke, d.h. Teilbäume mit  $z^* < \underline{z}$  nicht weiterentwickeln)

(R1)  $z^* = 30\frac{2}{3} \Rightarrow c_o = \bar{z} = 30$  ist obere Schranke, d.h. bei weiterer Verzweigung kann der ZF-Wert nicht wachsen.

Gäbe es einen Knoten mit  $z^* = c_o \leq 23$  könnten man auf dessen Weiterverzweigung verzichten.

Nun also (R1) bzgl.  $x_2$  weiterentwickeln.

$$(R3) \quad \begin{aligned} \max z = & 20x_1 + 8x_3 \\ & 7x_1 + 5x_3 \leq 14 \\ & x_2 = 0, \quad x_4 = 0 \\ & 0 \leq x_1, x_3 \leq 1 \end{aligned}$$

Lösung:  $x_2^* = x_4^* = 0$ ,  $x_1^* = x_3^* = 1$ ,  $z^* = 28$ ,  $\underline{z} = c_u = 28$

$$(R4) \quad \begin{aligned} \max z = & 12 + 20x_1 + 8x_3 \\ & 7x_1 + 5x_3 \leq 5 \\ & x_2 = 1, \quad x_4 = 0 \\ & 0 \leq x_1, x_3 \leq 1 \end{aligned}$$

Mit Prüfroutine folgt  $x_1 = 0$ , also

$$(R4') \quad \begin{aligned} \max z = & 12 + 8x_3 \\ & 5x_3 \leq 5 \\ & x_2 = 1, \quad x_1 = x_4 = 0 \\ & 0 \leq x_3 \leq 1 \end{aligned}$$

Lösung:  $x_1^* = x_4^* = 0$ ,  $x_2^* = x_3^* = 1$ ,  $z^* = 20$ ,  $\underline{z} = c_u = 20$  (auch Lösung für (P))

Für alle Blätter dieses Baumes gilt: „Die optimale Lösung des entsprechenden Teilproblems liegt in S.“

Abbruch des Verzweigungsprozesses:

Optimale Lösung für (P) ist optimale Lösung für (R3).

$$x_2^* = x_4^* = 0, \quad x_1^* = x_3^* = 1, \quad z^* = 28$$

## 12 Tourenplanung

### 12.1 Das Traveling Salesman Problem (TSP)

Beim TSP sucht man eine optimale Reiseroute von einem Startort über eine Anzahl vorgegebener Orte zurück zum Startort. In der Praxis treten TSPs z.B. bei der Planung kürzester oder kostengünstigster Fahrtrouten zur Belieferung von Kunden, Verdratung von Schaltungen oder Festlegung der Bearbeitungsreihenfolge mehrerer Aufträge auf einer Maschine auf, sodass die Summe der Umrüstkosten oder -zeiten minimal wird (Aufträge entsprechen dann den Orten und die Umrüstkosten/-zeiten von Auftrag zu Auftrag den Entfernungen zwischen den entsprechenden Orten).

gegeben:

- Menge von Punkten (Knoten)
- Entfernung zwischen den Punkten

gesucht: kürzeste Rundreise durch alle Punkte

Da das TSP ein NP-schweres kombinatorisches Optimierungsproblem ist, werden bei großen Instanzen Heuristiken angewandt, die zwar in der Regel nur suboptimale Lösungen liefern, diese jedoch mit vertretbarem Rechenaufwand.

#### 12.1.1 Eine allgemeine Heuristik

Algorithmus des „nächsten“ Nachbarn. Man wähle in jedem Schritt die lokal beste Alternative.

Anwendung auf das TSP:

1. Man wähle irgendeine Stadt als Anfangsstadt.
2. Um nächste Stadt auszuwählen, betrachte alle noch nicht besuchten Städte. Wähle die nächstgelegene aus. Gehe zur nächstgelegenen Stadt.
3. Solange nicht alle Städte besucht wurden, wiederhole Schritt 2.

#### 12.1.2 Graphentheoretische Formulierung

Gegeben:

Gerichteter Graph  $G = (V, E)$  mit  $V$  Knotenmenge,  $E$  Pfeilmenge (Kantenmenge),  $E \subset V \times V$ , jedem Pfeil ist eine nichtnegative Bewertung zugeordnet (parallele Pfeile sind nicht erlaubt).

Es ist eine Rundtour mit minimaler Bewertungssumme gesucht, die jeden Knoten genau einmal durchläuft.

Zulässige Lösung z.B.:

1-2-6-5-3-4-1

Nicht zulässige Lösungen z.B.:

1-3-5-4-6-1 (2 ist nicht enthalten und  $(1,3) \notin E$ )

1-3-4-1-2-5-6 (1 doppelt besucht und  $(1,3) \notin E$ )

### 12.1.3 Beschreibung als Zuordnungsproblem

Bewertungsmatrix  $C = (c_{ij})$ , falls  $(i, j) \notin E$ , setze  $c_{ij} = +\infty$ .

Für das TSP gilt:

- (1)  $x_{ij} \in \{0, 1\}$ ,  $\forall i, j \in V$   
 EV  $x_{ij} = \begin{cases} 1, & \text{falls von „Stadt } i \text{ nach Stadt } j\text{“} \\ 0, & \text{sonst} \end{cases}$
- (2)  $\sum_j x_{ij} = 1$ ,  $\forall i \in V$  (Von jedem Ort aus gibt es genau einen Nachfolgerort in der Tour.)
- (3)  $\sum_i x_{ij} = 1$ ,  $\forall j \in V$  (Jeder Ort hat genau einen Vorgängerort in der Tour.)
- (4)  $\sum_{i \in W} \sum_{j \notin W} x_{ij} \geq 1$ ,  $\forall W \subset V$ ,  $W \neq \emptyset$ ,  $W \neq V$  (Aus jeder echten Teilmenge  $W$  von  $V$  führt mindestens ein Pfeil hinaus — keine Kurzzyklen)  
 Es gibt insgesamt  $2^n - 2$  verschiedenen Menge  $W$ , d.h. die Zahl der Restriktionen wächst exponentiell.
- (5) ZF:  $\min \sum_i \sum_j c_{ij} x_{ij}$

Die Teile (1),(2),(3),(5) entsprechen dem Zuordnungsproblem. Ohne Teil (4) ist das Auftreten von Kurzzyklen nicht ausgeschlossen. Ein System mit Kurzzyklen liegt vor, wenn jeder Zyklus des Systems nur wenige der  $|V|$  Knoten von  $G$  enthält, alle Zyklen zusammen aber jeden der Knoten genau einmal enthalten. Um Kurzzyklen auszuschließen wird Teil (4) (Zyklusbedingungen) ergänzt.

### 12.1.4 Lösung mit Branch and Bound

Problem (R): Relaxation = Zuordnungsproblem ohne Teil (4) (ordne jedem Knoten einen Nachfolgerknoten zu)

Verzweigung: Falls Kurzzyklen entstehen, spalte man in Teilprobleme auf. Dabei wird jeweils mindestens ein Pfeil des Graphen verboten ( $c_{ij} = +\infty$  garantiert Endlichkeit).

Illustration zur Verzweigung:

Lösung von (R) möge folgende Graphen liefern:

Durch (P1) wird z.B. erzwungen, in 4,5,6 oder 7 zu gehen und in 2 von rechts zu verzweigen.

Verzweigungsregel:

- Man wähle zunächst einen möglichst „kurzen“ Zyklus mit  $z$ -Pfeilen ( $z = 3$ ).  
 $\Rightarrow$  In keiner Lösung des TSP sind alle  $z$ -Pfeile vorhanden (ansonsten Zyklus). Deshalb in  $z$  Teilprobleme verzweigen, wobei jeweils einer der Pfeile verboten wird.
- Man kann mehr Pfeile verbieten (kleinerer Baum), falls man wie folgt schließt:  
 Von der linken Komponente des Graphen muss mindestens ein Pfeil in die rechte Komponente führen.

Fall 1: Es geht ein Pfeil von der ersten der  $z$  Knoten zu einer anderen Komponente des Graphen (evtl. auch von weiteren).

Fall  $i$ : Es geht ein Pfeil von der  $i$ -ten der  $z$  Knoten zu einer anderen Komponente des Graphen, aber nicht von den Kanten 1 bis  $i - 1$ . (Fall  $i$  schließt alle Lösungen des TSP aus, die bereits in einem der Fälle 1 bis  $i - 1$  zulässig sind.)

- Man definiere zu jedem Fall  $i$  ( $i = 1, \dots, z$ ) ein Teilproblem (R $i$ )

Im Problem (R $i$ ) können folgende Pfeile verboten werden (weil gemäß Definition R $i$  der eine mögliche Pfeil schon „festgelegt“ ist):

- Alle Pfeile des  $i$ -ten Knoten zu den übrigen  $z - 1$  Knoten dieser Komponente.
- Alle Pfeile von den ersten  $i - 1$  Knoten der Komponente zu irgendeiner anderen Komponente.

Beispiel:

(R1): verbiete (1, 2) und (1, 3) in (R)

(R2): verbiete (2, 1), (2, 3), (1, 4), (1, 5), (1, 6), (1, 7) in (R)

(R3): verbiete (3, 1), (3, 2), (1, 4), (1, 5), (1, 6), (1, 7), (2, 4), (2, 5), (2, 6), (2, 7) in (R)

Nun alle  $z$ -Probleme als Zuordnungsproblem lösen.

Beispiel:

$$C = \begin{pmatrix} 0 & 9 & 18 & 16 & 5 \\ 23 & 0 & 2 & 3 & 3 \\ 7 & 2 & 0 & 21 & 3 \\ 4 & 8 & 7 & 0 & 28 \\ 4 & 9 & 18 & 3 & 0 \end{pmatrix}$$

Direkte Schleifen, als Extremfall von Kurzzyklen werden ausgeschlossen, indem alle diagonalen Zuordnungen verboten werden. Danach wird Zuordnungsproblem (R) gelöst.

$$C = \begin{pmatrix} - & 9 & 18 & 16 & 5 \\ 23 & - & 2 & 3 & 3 \\ 7 & 2 & - & 21 & 3 \\ 4 & 8 & 7 & - & 28 \\ 4 & 9 & 18 & 3 & - \end{pmatrix} \xrightarrow{\text{Zuordnungsproblem}} \dots \xrightarrow{\text{Zuordnungsproblem}} \begin{pmatrix} - & 4 & 13 & 1 & [0] \\ 21 & - & [0] & 1 & 1 \\ 5 & [0] & - & 19 & 1 \\ [0] & 4 & 3 & - & 24 \\ 1 & 6 & 15 & [0] & - \end{pmatrix}$$

Es gibt zwei Kurzzyklen:

Wert der Zuordnung  $f = 5 + 2 + 2 + 4 + 3 = 16$  ist untere Schranke für TSP.

Kurzzyklus  $z = 2$  betrachten:

Fall 1:  $\Rightarrow$  (R2): verbiete  $(3, 2)$  und  $(2, 1), (2, 4), (2, 5)$  in (R)

Fall 2:  $\Rightarrow$  (R1): verbiete  $(2, 3)$  in (R)

1) Ausgehend von der Lösung (R) löse (R2)

$$\begin{pmatrix} - & 4 & 13 & 1 & 0 \\ - & - & 0 & - & - \\ 5 & - & - & 19 & 1 \\ 0 & 4 & 3 & - & 24 \\ 1 & 6 & 15 & 0 & - \end{pmatrix} \text{ Zuordnungsproblem} \xrightarrow{\dots} \begin{pmatrix} - & [0] & 13 & 1 & 0 \\ - & - & [0] & - & - \\ 4 & - & - & 18 & [0] \\ [0] & 0 & 3 & - & 24 \\ 1 & 2 & 15 & [0] & - \end{pmatrix}$$

Rundtour (Lösung des TSP):  $1 \rightarrow 2 \rightarrow 3 \rightarrow 5 \rightarrow 4 \rightarrow 1$

Obere Schranke für Optimallösung  $\bar{f} = 9 + 2 + 3 + 4 + 3 = 21$ .

2) Ausgehend von (R) löse (R1)

$$\begin{pmatrix} - & 4 & 13 & 1 & 0 \\ 21 & - & - & 1 & 1 \\ 5 & 0 & - & 19 & 1 \\ 0 & 4 & 3 & - & 24 \\ 1 & 6 & 15 & 0 & - \end{pmatrix} \text{ Zuordnungsproblem} \xrightarrow{\dots} \begin{pmatrix} - & 4 & 10 & 1 & 0 \\ 20 & - & - & [0] & 0 \\ 5 & [0] & - & 19 & 1 \\ [0] & 4 & 0 & - & 24 \\ 1 & 6 & 12 & 0 & - \end{pmatrix}$$

Minimum der nichtgetrichenen Elemente ist 1.

$$\begin{pmatrix} - & 3 & 9 & 1 & [0] \\ 19 & - & - & [0] & 0 \\ 5 & [0] & - & 20 & 2 \\ 0 & 4 & [0] & - & 25 \\ [0] & 5 & 11 & 0 & - \end{pmatrix} \text{ Zielfunktionswert } f = 21 \text{ (untere Schranke)}$$

Es gibt zwei andere Kurzzyklen:

Ist keine zulässige Lösung des TSP. Der ZF-Wert 21 ist nicht besser (kleiner) als die aktuelle obere Schranke  $\bar{f} = 21$ .  $\Rightarrow$  Keine weitere Verzweigung von (R1) notwendig.

Optimallösung von (R2) ist auch Optimallösung des TSP.

## 12.2 Das Vehicle Routing Problem (VRP)

### 12.2.1 Modell zum VRP

Graph  $G = (V, E, C)$

- Knotenmenge  $V = \{1, \dots, n\}$ 
  - Depots  $R = \{1, \dots, r\}$ ,  $r < n$
  - Kunden  $i \in V \setminus R = \{r + 1, \dots, n\}$  mit Nachfrage  $d_i$  (pro Tour entweder Abholen oder Liefern)

- Kantenmenge  $E$ : gerichtete oder ungerichtete Kanten
- Kostenmatrix  $C = (c_{ij})$ 
  - $C$  symmetrisch:  $c_{ij} = c_{ji}, \forall i, j \in V$   
bzw. asymmetrisch
  - $C$  euklidisch:  $c_{ij} + c_{jk} \geq c_{ik}, \forall i, j, k \in V$

Nebebedingungen für das VRP:

- Kapazitätsbeschränkungen der Fahrzeuge:  $D$  (homogen) oder  $D_k, k = 1, \dots, m$  (in-homogen)  
Nachfrage in den Knoten:  $d_i \geq 0, \forall i \in V, d_i = 0, \forall i \in R$  (Depotknoten)  
 $\sum_{i \in V \setminus R} d_i \geq D$  (mehrere Routen nötig, sonst TSP)
- Anzahl der Orte auf einer Route
- Zeitbeschränkungen: Reisezeit und Wartezeit
- Zeitfenster: Ort  $i$  muss im Intervall  $[a_i, b_i]$  angefahren werden
- Reihenfolge der zu bedienenden Orte:  $i$  vor  $j$

Modelleigenschaften des VRP:

- Fahrzeugflotte:  $m$  Fahrzeuge in den Depots  $R$ 
  - $m$  fest vorgegeben oder variabel:  $\underline{m} \leq m \leq \overline{m}$
  - Anzahl der Fahrzeuge pro Depot
  - Fixkosten für die Benutzung eines Depots oder Fahrzeugs
- Routenaufbau: Kostenminimale Fahrzeugrouten
  - Besuche jeden Ort aus  $V \setminus R$  (Kunden) genau einmal mit einem Fahrzeug
  - Jedes Fahrzeug kehrt an seinen Ausgangspunkt zurück
  - Weitere Nebenbedingungen

Gesucht ist eine Menge von Fahrten mit minimaler Gesamtdauer. Im Rahmen eines VRPs sind also zwei Teilprobleme zu lösen: Eine Art Zuordnungsproblem, das festlegt, welche Kunden auf den einzelnen Fahrten bedient werden, und ein Reihenfolgeproblem, das bestimmt, in welcher Reihenfolge die Kunden einer Fahrt zu beliefern sind.

### 12.2.2 Eröffnungsheuristiken (Savings- und Sweep-Verfahren)

#### 1. Savings-Verfahren

Beim Savings-Verfahren werden das Zuordnungs- und das Reihenfolgeproblem, näherungsweise parallel gelöst werden (Parallelverfahren).

Sei  $t_{ij}$  die nichtnegative Fahrzeit von Kunde  $i$  (bzw. Depot 0) zu Kunde  $j$  (bzw. Depot 0) für  $i, j = 0, 1, \dots, n; i \neq j$ . Für die Fahrzeiten gelte die Dreiecksungleichung, d.h.  $t_{ij} \leq t_{ik} + t_{kj}$  ( $0 \leq i, j \leq n, 1 \leq k \leq n, i \neq j, i \neq k, k \neq j$ )

Anstatt der Fahrzeiten werden analog die allgemeinen Kosten  $c_{ij}$  verwendet.



Die geordnete Menge der Kunden einer Fahrt zusammen mit dem Depot, bezeichnet man als Tour. Eine *Pendeltour* ist eine Tour, bei der Anfangs- und Endkunde zusammenfallen, also  $(0, i, 0)$ ,  $1 \leq i \leq n$ .

Das Savingsverfahren läuft wie folgt ab:

1. Berechne alle positiven Savings (Einsparungen)  $s_{ij} := t_{i0} + t_{0j} - t_{ij}$  (für alle Paare von Knoten).
2. Sortiere Saving in nicht aufsteigender Reihenfolge.
3. Starte mit Pendeltouren Depot-Kunde  $i$ -Depot, also  $(0, i, 0)$  für alle Kunden, d.h. füge alle Kanten  $\{0, i\}$  zweimal zur Lösung hinzu.
4. Bestimme das erste Saving  $s_{ij}$  der Liste mit
  - (i)  $i$  und  $j$  sind verschiedene Touren .
  - (ii)  $i$  und  $j$  sind entweder erster oder letzter Kunde auf der Tour.
  - (iii) die Summe der Bedarfe beider Touren hält die Kapazitätsbeschränkung ein.

Füge beide Touren zusammen, d.h. füge Kante  $\{i, j\}$  zur Lösung hinzu und entferne je eine Kante  $\{0, i\}$  und  $\{j, 0\}$ .
5. Wiederhole 4. bis kein passendes Saving mehr vorhanden.

## 2. Sweep-Verfahren

Beim Sweep-Verfahren wird zunächst die Menge der Kunden unter Beachtung der Kapazitäts- und Zeitrestriktionen in Teilmengen zerlegt, die für Touren in Frage kommen (Zuordnungsproblem näherungsweise lösen). Danach bestimmt man die Reihenfolge der Kunden in jeder Teilmenge (Reihenfolgeproblem). Das Sweep-Verfahren ist also ein Sukzessiverfahren, es läuft wie folgt ab:

1. Polarkoordinaten allen Kunden berechnen, wobei das Depot das Zentrum bildet und ein beliebiger Kunde Winkel  $0^\circ$  hat.
2. Starte mit dem Kunden  $i$  (mit dem kleinsten Winkel) und konstruiere ein neues Cluster durch „Abtasten“ (sweep) aufeinander folgender Kunden  $i + 1, i + 2, \dots$  bis die Fahrzeugkapazität erschöpft ist.
3. Wiederhole 2. bis allen Kunden geclustert sind.
4. Löse das TSP für jedes Cluster und das Depot.

Bemerkung: Paralleles Savings-Verfahren benötigt mehr Fahrzeuge als sequenzielles Sweep-Verfahren.

### 12.2.3 Verbesserungsverfahren für TSP und VRP

Verbesserungsverfahren starten mit einem Hamiltonschen Kreis  $K$ , der etwa mit einem Eröffungsverfahren des Abschnitts 12.2.2 bestimmt worden ist, und versuchen die Anfangslösung  $K$  zu verbessern, d.h. sie betrachten Hamiltonsche Kreise, die in einer gewissen Umgebung von  $K$  liegen und wähle unter diesen einen kürzesten aus.

#### **k-opt**

Bei den sogenannten  $k$ -opt Verfahren werden genau  $k$  Kanten des aktuellen Hamiltonschen Kreise  $K$  gegen  $k$  andere Kanten (die nicht notwendig allen von den ersten Kanten verschieden sein müssen) des zugrundeliegenden Graphen  $G$  ausgetauscht (erst löschen, dann einfügen). Der kürzere der beiden Kreise wird als neuer aktueller Kreis verwendet. Das Verfahren bricht ab, wenn alle möglichen Vertauschungen untersucht worden sind. Da der Rechenaufwand von  $k$ -opt mindestens  $O(n^k)$  ist, die Güte der besten Näherungslösung für größere  $k$  aber im Allgemeinen nur unwesentlich ansteigt, verwendet man in der Praxis meistens nur die Verfahren 2-opt und 3-opt.

Man erkennt, dass durch die Ersetzung der alten durch die neuen Kanten einige der Knoten in umgekehrter Reihenfolge durchlaufen werden, weshalb  $k$ -opt für Probleme mit Zeitfenstern wenig geeignet ist.

#### **Or-opt**

Or-opt ist eine Modifikation von 3-opt bei der die Anzahl der Kantenvertauschungen wesentlich gegenüber 3-opt reduziert wird, die Güte der erhaltenen Näherungslösung aber kaum schlechter ist. Bei diesem Verfahren wird für je  $s$  benachbarte Knoten (zunächst  $s = 3$ , dann  $s = 2$  und schließlich  $s = 1$ ) des aktuellen Hamiltonschen Kreises geprüft, ob diese Knoten zwischen zwei anderen Knoten so eingefügt werden können, dass sich ein kürzerer Hamiltonscher Kreis ergibt.

Für Probleme mit Zeitfenstern gut geeignet, da die Orientierung nicht verändert wird.

#### **2-opt\***

Orientierungen von Teiltouren werden nicht geändert, Kunden am Touranfang bleiben am Anfang, Kunden am Tourende bleiben am Tourende.

Ebenfalls für Problem mit Zeitfenstern gut geeignet.

### $\lambda$ -Interchange

Bei dieser Methode werden maximal  $\lambda$  viele Knoten aus einer Tour in eine andere verschoben.  $\lambda = 2$ : 0,1 oder 2 Knoten aus Tour 1 in Tour 2 schieben und gleichzeitig 0,1 oder 2 Knoten aus Tour 2 in Tour 1 schieben.

Im Beispiel werden ein Knoten aus Tour 1 in Tour 2 und zwei Knoten aus Tour 2 in Tour 1 geschoben.

Als Vergleich der Verbesserungsverfahren diene die nachfolgenden Tabelle.

Verfahren	Nachbarschaft	Bemerkung
$k$ -opt	$k$ Kantentausche	
Or-opt	Verschieben von 1–3 benachbarten Kanten	Teilmenge von 3-opt
2-opt*	Austausch von Tourenanfangs- bzw. -endstücken zwischen 2 Touren	Teilmenge von 2-opt, nur für je 2 Touren
$\lambda$ -Interchange	Austausch von Knotenteilmengen von maximal $\lambda$ Knoten zwischen zwei Touren	enthält die Or-opt-Nachbarschaft

## 13 Gemischt-ganzzahlige Programmierung, Branch and Bound

### 13.1 Beispiel eines gemischt-ganzzahligen LP's

$$\begin{aligned}
 \max \quad & z = x_1 + 4x_2 \\
 \text{s.d.} \quad & 5x_1 + 8x_2 \leq 40 \\
 & -2x_1 + 3x_2 \leq 9 \\
 & x_1, x_2 \geq 0 \text{ und ganzzahlig}
 \end{aligned}$$

Die Formeln sind zwar linear, aber durch die Ganzzahligkeit geht die Linearität verloren.

#### 1. Initialisierung

$P_1$ ; LP-Relaxation, aktuelle untere Schranke  $z = -\infty$   
optimale Lösung von  $P_1$ :  $x_1^0 = (1,55; 4,03)$ ;  $z = 17,67$   
obere Schranke  $\bar{z}_1 = z_1 = 17,67$   
 $x_1^0$  ist keine zulässige Lösung für das ganzzahlige Model.  
 $P_1$  aktiv und wird Verzweigungsknoten.

#### 2. Verzweigung

Verzweigung zunächst nach  $x_1$  (Prioritätsregel), d.h. zusätzliche Restriktionen  
 $x_1 \leq 1$                     oder     $x_1 \geq 2$   
( $P_2$ -Knoten)                    ( $P_3$ -Knoten)

$P_2$  und  $P_3$  heißen aktiv.

### 3. Lösung von $P_2$

optimale Lösung  $x_2^o = (1; 3, 67)$ ,  $z_2 = 15, 67 = \bar{z}_2$

Da  $x_2^o$  nicht ganzzahlig ist, heißt  $P_2$  weiterhin aktiv.

→ nach LIFO-Regel

### 4. Knoten $P_2$ verzweigen

$P_2$ :  $x_2 \leq 3$  (Knoten  $P_4$ ),  $x_2 \geq 4$  (Knoten  $P_5$ )

$P_4$  lösen:

→ ganzzahlige optimale Lösung von  $P_4$

$x_4^o = (1; 3)$ ,  $z = 13 = \bar{z}_4$  (für den Teilbaum unter  $P_4$ )

$x_4^o$  ist zulässig für das Problem  $P$ , da ganzzahlig

$P_4$ :  $x_1 \leq 1$ ,  $x_2 \leq 3$  und andere Restriktionen

→ Knoten  $P_4$  heißt terminiert (nicht mehr aktiv)

$\wedge z_4$  ist untere Schranke für den Optimalwert von  $P$  (zulässige Lösung gefunden)

$\underline{z} := z_4 = 13$

### 5. Backtracking über $P_2$ nach $P_5$

$x_2 \geq 4$  zum optimalen Tableau von  $P_2$  hinzufügen

→ dualer Simplexschritt ist nicht möglich ( $x_2 \geq 4 \wedge x_1 \leq 1$  kollidiert mit  $-2x_1 + 3x_2 \leq 9$ )

d.h.  $P_5$  besitzt keine zulässige Lösung

→ Backtracking über  $P_2$  nach  $P_1$  führt zur „Auswahl“ von  $P_3$

### 6. Lösung von $P_3$

optimale, nicht-ganzzahlige Lösung:  $x_3^o = (2; 3, 75)$ ,  $z_3 = 17 = \bar{z}_3$

Da  $\bar{z}_3 = 17 > \underline{z} = 13 \Rightarrow P_3$  ist weiter zu verzweigen, kein Bounding.

### 7. Verzweigung von $P_3$ nach $x_2$

→  $x_2 \leq 3$  (Knoten  $P_6$ ),  $x_2 \geq 4$  (Knoten  $P_7$ )

optimale Lösung von  $P_6$ :  $x_6^o = (3, 2; 3)$ ,  $z_6 = 15 = \bar{z}_6$

(man beachte:  $x_1$  ist nun wieder nicht-ganzzahlig)

### 8. LIFO-Strategie — von $P_6$ aus weiter verzweigen

→  $x_1 \leq 3$  (Knoten  $P_8$ ),  $x_1 \geq 4$  (Knoten  $P_9$ )

$P_8$  besitzt eine optimale ganzzahlige Lösung:  $x_8^o = (3; 3)$ ,  $z_8 = 15 = \bar{z}_8$

Damit ist eine bessere, insgesamt (also für  $P$ ) zulässige Lösung gefunden worden, als bei  $P_4$ .

→ Aktualisierung der unteren Schranke gemäß  $\underline{z} = z_8 = 15$

( $P_8$  liefert die optimale Lösung, dies ist aber zu diesem Zeitpunkt noch nicht klar.)

### 9. Backtracking zu $P_6$ , von $P_6$ nach $P_9$

$P_9$  liefert einen optimalen ZF-Wert  $z_9 = 14 \leq 15 = \underline{z}$

(schlechter als die schon bekannte untere Schranke)

$P_9$  wird terminiert

### 10. Backtracking von $P_6$ über $P_3$ nach $P_7$

$P_7$  besitzt keine zulässige Lösung.

Kein offener (aktiver) Knoten mehr vorhanden.

⇒ Optimal ist die beste bekannte Lösung, d.h. die aktuelle untere Schranke: Lösung von  $P_8$ .

### 13.2 Branch and Bound: Ansatz von Dakin

Problem: LP-Relaxation

$$\begin{array}{ll} \max & z = c^T x \\ \text{s.d.} & Ax \leq b \\ & x \geq 0 \end{array}$$

$x_j$  ganzzahlig für  $j = 1, \dots, p$  (o.B.d.A)

$x^T = (x_1, \dots, x_p, x_{p+1}, \dots, x_n)$  gemischt ganzzahliges Problem (mixed integer Problem (MIP))

#### Ansatz von Dakin (1965)

1. LP-Relaxation
2. Ist eine ganzzahlige geforderte Variable in einer optimalen Lösung der LP-Relaxation nicht ganzzahlig, dann
3. ist ihre Ganzzahligkeit in den optimalen Lösungen der Teilmodelle (Folgeknoten) durch geeignete ganzzahlige Schranken zu erreichen.

LP-Relaxation

Verletzung der Ganzzahligkeit:

Problem  $P_k$ , repräsentiert von Knoten  $k$

$x_k^o$  optimale Lösung von  $k$

Die Komponente  $x_{k,j}^o$  ( $1 \leq j \leq p$ ) von  $x_k^o$  erfülle die Ganzzahligkeitsbedingung nicht.

$k$  wird unterdrückt, d.h.  $x_j^o = \lfloor x_j^0 \rfloor + f_j$ ,  $0 < f_j < 1$

Verzweigung wie folgt:

- obere Schranke  $\lfloor x_j^o \rfloor$  hinzufügen
- untere Schranke  $\lfloor x_j^o \rfloor + 1$  hinzufügen

Eigenschaften der Verzweigung:

- In jedem Knoten wird nach einer Verzweigungsvariablen in zwei Teilmodelle verzweigt.
- Damit entsteht eine Folge von Teilmodellen  $P_k$ ,  $k = 1, 2, \dots$
- Die Lösungsräume  $X_k$  für  $P_k$  sind LP-Relaxationen mit zusätzlich eingeführten ganzzahligen Bounds für einige oder alle der Variablen, für die Ganzzahligkeit gefordert ist.
- $X_k$  sind dennoch Relaxationen bzgl. der Ganzzahligkeitsforderungen  
 ⇒ Ihre optimalen ZF-Werte  $z_k^o$  sind obere Schranken für die entsprechenden gemischt-ganzzahligen Teilmodelle.

Technische Details:

- Wenn das optimale Tableau für das vorangehende Modell (übergeordnete Baumebene) vorliegt  
 → analog zum Gomory-Verfahren (Abschnitt 5.2) vorgehen.
- Falls mehrere Variablen Ganzzahligkeitsforderungen nicht erfüllen, Konfliktlösung nach folgenden Kriterien (nach welcher Variablen verzweigen?)
  1. die zu erwartende ZF-Wertverschlechterung (z.B. beim ersten dualen Schritt für jedes  $x_j$ )
  2. extern vorgegebene Prioritäten
  3. Abweichung einer Variablen vom nächsten ganzzahligen Wert (ganzzahlige Unzulässigkeit).

### 13.3 Allgemeine Darstellung des Prinzips „Branch and Bound“

Problem      P:  $\max f(x)$ ,  $x \in S$

Eine Möglichkeit das Problem  $P$  zu lösen ist die *vollständige Enumeration*, bei der alle möglichen (und zulässigen) Belegungen des Vektors  $x$  nacheinander erzeugt und die ZF-Werte verglichen werden. Diese Vorgehensweise ist nur bei geringen Dimensionen geeignet.

Bei der *impliziten Enumeration* versucht man, sukzessive Teilmenge des Lösungsbereiches zu finden, in denen keine optimale Lösung von  $P$  liegen kann und die folglich ausgesondert werden können. Dies kann mit Hilfe eines Suchbaums geschehen, d.h. eines Wurzelbaumes, dessen Knoten gewissen Teilmenge des Lösungsbereiches entsprechen. Diese Vorgehensweise ist unter dem Namen *Branch-and-Bound-Methode* bekannt. Branch-and-Bound-Verfahren benötigen im Allgemeinen exponentiellen Rechenaufwand, liefern aber stets eine optimale Lösung (falls überhaupt eine existiert).

Anstelle ein Problem  $P$  zu lösen, konstruiert man eine Folge von Problemen  $P_k$  (mit Lösungsbereich  $X_k$ ) mit der Zielstellung:

- a) entweder optimale Lösung von  $P_k$  bestimmen oder
- b) zeigen, dass Optimalwert von  $P_k$  nicht besser als bisher bekannter ZF-Wert ist oder
- c) zeigen, dass  $X_k$  leer ist.

Lösungsaufwand verringern durch:

- geschicktes Verzweigen in Teilproblem  $P_k$  (Branching)
- Terminieren von Teilbäumen vor deren vollständiger Generierung nach b) oder c) (Bounding)

Grundelemente des Branch and Bound Verfahrens

### 1. Initialisierung

$P_1$ -Definition des Ausgangsmodells (Wurzelknoten)

$P_1$   $\nearrow$  ursprüngliches Problem  $P$   
 $P_1$   $\searrow$  Relaxation von  $P$  (Überdeckung  $T$  von  $S$ ,  $S \subseteq T$ , z.B.  $\mathbb{N} \subseteq \mathbb{R}$ )

Maximierungsproblem: Aktuelle untere Schranke  $\underline{z}$  angeben (für zulässige Lösung).

Minimierungsproblem: Aktuelle obere Schranke  $\bar{z}$  angeben (für zulässige Lösung).

(Dazu bekannte zulässige Lösungen benutzen oder  $\underline{z} = -\infty$  bzw.  $\bar{z} = +\infty$  setzen.)

### 2. Verzweigung (Branching)

Bedeutet:

- Auswahl eines noch nicht untersuchten Teilmodells (Verzweigungsknoten) oder
- Erzeugen von Teilmodellen (Verzweigungsknoten) aus einem untersuchten aber nicht terminierten (Teil-) Modell

Die Verzweigung wird durch Regeln gesteuert; diese definieren

- wie aus einem Modell die Teilmodelle (Folgeknoten) generiert werden und
- welcher Knoten als nächster betrachtet (entwickelt) wird (Knotenauswahlregeln, Kontrollstrategien).

Beispiel für Knotenauswahlregeln:

1. Orientiert an den Bounds der aktiven (noch nicht überprüften) Knoten, z.B. best-first-search.

2. Extern festgelegte Regeln.
3. Schrankenunabhängige Regeln, z.B. LIFO (last in first out) soweit möglich vom zuletzt erzeugten Knoten aus verzweigen, bei Backtracking zum zuletzt generierten noch aktiven Knoten gehen (depth-first)

Knotenauswahlregeln können dynamisch verändert werden: z.B. zuerst LIFO (um schnell  $\underline{z}$  bzw.  $\bar{z}$  zu bekommen), danach schrankenabhängige Regeln benutzen.

### 3. Terminierung (Bounding)

eines Knotens (Teilbaums) erfolgt, wenn entweder

- nachgewiesen ist, dass der Lösungsbereich  $X_k$  von Teilmodell  $P_k$  leer ist oder
- eine obere Schranke  $\bar{z}_k$  für den ZF-Wert von  $P_k$  angegeben wird, die nicht größer als die aktuelle untere Schranke  $\underline{z}$  des Ausgangsproblems (bei Maximierung) ist.  
 $\Rightarrow$  Eine weitere Verzweigung des Knotens ist nicht mehr sinnvoll.

## 14 Dynamische Programmierung

Bislang haben wir uns mit „statischen“ Problemen beschäftigt, d.h. der zugrundeliegende Sachverhalt wird nur zu einem Zeitpunkt oder während eines kurzen Zeitintervalls ohne jede zeitliche Änderung betrachtet. In diesem Abschnitt werden „dynamische“ Modelle und Methoden behandelt, denen ein in der Zeit ablaufender Prozess zugrunde liegt. Die Vorhersage zukünftiger Entwicklungen ist in der Praxis häufig mit großen Unsicherheiten verbunden, daher werden zur Beschreibung solcher Phänomene oft Zufallsgrößen verwendet. Hier werden allerdings ausschließlich deterministische Größen betrachtet.

In der dynamischen Programmierung werden Probleme betrachtet, die in einzelne Stufen zerlegt werden können, sodass die Gesamtoptimierung auf eine stufenweise Optimierung reduziert werden kann. Bietet sich an, wenn die Stufen einzelnen Zeitperioden entsprechen. Als typisches Beispiel lässt sich das Lagerhaltungsproblem anführen, auf das in Abschnitt 14.2 eingegangen wird, oder kürzeste Routen in Netzwerken.

Die einzelnen Stufen brauchen nicht notwendig Zeitperioden zu sein, wesentlich ist nur, dass das Problem eine Interpretation als ein aus einzelnen Stufen bestehender Prozess erlaubt (künstliche Dynamisierung des Problems). Als Beispiel sei das in Abschnitt 11 angesprochene Rucksackproblem

$$\begin{aligned} \max \quad & \sum_{j=1}^n c_j u_j \\ \text{u.d.N.} \quad & \sum_{j=1}^n a_j u_j \leq b \\ & u_j \in \{0, 1\}, \text{ für } j = 1, \dots, n \\ & u_j = \begin{cases} 1, & \text{falls Gegenstand } j \text{ in den Rucksack gepackt wird} \\ 0, & \text{sonst} \end{cases} \end{aligned}$$

(mit  $x$  durch  $u$  ersetzt) genannt. Die künstliche Dynamisierung besteht nun darin, dass die Entscheidungen, welche der Gegenstände  $1, \dots, n$  in den Rucksack gepackt werden, als Entscheidungen in  $n$  aufeinanderfolgenden Stufen oder Perioden zu interpretieren. Dazu werden sukzessive nichtnegative Größen  $x_j$  gemäß

$$\begin{aligned} x_1 &= b \\ x_{j+1} &= x_j - a_j u_j, \text{ für } j = 1, \dots, n \end{aligned}$$



eingeführt. Hierbei stellt  $x_j$  das für die Gegenstände  $j, \dots, n$  noch verfügbare Restgewicht des Rucksacks dar. Damit hat das Rucksackproblem die folgende Gestalt:

$$\begin{array}{l} \max \quad \sum_{j=1}^n c_j u_j \\ \text{u.d.N.} \quad x_{j+1} = x_j - a_j u_j, \text{ für } j = 1, \dots, n \\ \quad \quad \quad x_1 = b \\ \quad \quad \quad \left. \begin{array}{l} 0 \leq x_{j+1} \leq b \\ u_j \in \{0, 1\}, \text{ falls } x_j \geq a_j \\ u_j = 0, \text{ falls } x_j < a_j \end{array} \right\} \text{ für } j = 1, \dots, n \end{array}$$

## 14.1 Optimalitätsprinzip und Bellmansche Gleichungen

- Es wird ein System während eines endlichen Planungszeitraums betrachtet, der in  $n$  Stufen eingeteilt sei ( $n$ -stufiger Entscheidungsprozess).
- Zustandsvariablen  $x_{i-1}$  beschreibt den Zustand des Systems zu Beginn der Stufe  $i$  (bzw. am Ende der Stufe  $i-1$ ).
- Zu Beginn des Planungszeitraums befindet sich das System im vorgegebenen Anfangszustand  $x_0 = x_a$ .
- In Stufe  $i$  wird eine Entscheidung  $y_i$  getroffen, durch die das System in der Zustand  $x_i$  übergeht. Dieser Zustand  $x_i$  hängt nur vom vorherigen Zustand  $x_{i-1}$  und der getroffenen Entscheidung  $y_i$  ab, d.h.  $x_i = T_i(x_{i-1}, y_i)$  (Zustandstransformation).
- Mit einer Entscheidung  $y_i$  sind Kosten  $c_i(x_{i-1}, y_i)$  (Bewertung) verbunden.
- $y = (y_1, \dots, y_n)$  Entscheidungs-, Steuervariablen, Strategie, Entscheidungspolitik  
 $x = (x_0, x_1, \dots, x_n)$  Zustandstrajektorie, -folge
- Die möglichen Zustände am Ende von Stufe  $i$  (d.h. nach Entscheidung  $y_i$ ) sind aus einem nichtleeren Zustandsbereich  $X_i$ .  
Die möglichen Entscheidungen in Stufe  $i$  sind aus einem nichtleeren Steuerbereich  $Y_i(x_{i-1})$ , der noch vom Zustand  $x_{i-1}$  zu Beginn der Stufen  $i$  abhängen kann.  
Der einzig mögliche Zustand am Anfang von Stufe 1 ist  $x_0$ .

Zielkriterium:

$$z(x, y) = F(c_1(x_0, y_1), \dots, c_n(x_{n-1}, y_n))$$

Zerlegung des Problems mit  $n$  Entscheidungsvariablen in  $n$  Probleme mit je einer EV-Variable.

Restprozess ab Stufe  $i$ : beginnt mit  $x_{i-1}$ , Strategie  $(y_i, \dots, y_n)$

### Bellmansches Optimalitätsprinzip

Seien  $(y_1^*, \dots, y_i^*, \dots, y_n^*)$  eine optimale Politik für das Problem  $P_1(x_0)$  und  $x_{i-1}^*$  der Zustand zu Beginn von Stufe  $i$ . Dann ist  $(y_i^*, \dots, y_n^*)$  eine optimale Politik für das Problem  $P_i(x_{i-1}^*)$ . In anderen Worten: Die Entscheidungen auf den Stufen  $i, \dots, n$  des  $n$ -stufigen Problems  $P_1(x_0)$  sind (bei gegebenem Zustand  $x_{i-1}$ ) unabhängig von den getroffenen Entscheidungen der Stufen  $1, \dots, i-1$ . Die Optimalität des Restprozesses beginnend ab Stufe  $i$  und der

Strategie  $(y_1^*, \dots, y_n^*)$  hängt nur vom Anfangszustand  $x_{i-1}$  dieses Restprozesses und nicht davon ab, durch welche Entscheidungen  $(y_1, \dots, y_{i-1})$  der Zustand  $x_{i-1}$  entstanden ist.

Wie der Name verrät ist dies zunächst nur ein Prinzip. Für welche Problemklassen kann man zeigen, dass das Prinzip erfüllt (also eigentlich ein Satz) ist?

Ist eine additive Zielfunktion ohne zusätzliche Nebenbedingungen gegeben

$$z(x, y) = \sum_{i=1}^n c_i(x_{i-1}, y_i),$$

und sei  $f_i^*$  ein auf dem Zustandsbereich  $X_i$  definierte Wertfunktion ( $1 \leq i \leq n$ ), dann lautet die Bellmansche Funktionalgleichung

$$\begin{aligned} f_{n+1}^*(x_n) &:= 0 \quad (\text{wird explizit gesetzt}) \\ f_i(x_{i-1}, y_i) &= c_i(x_{i-1}, y_i) + \overbrace{f_{i+1}^*(T_i(x_{i-1}, y_i))}^{x_i} \\ f_i^*(x_{i-1}) &= \min_{y_i \in Y_i(x_{i-1})} f_i(x_{i-1}, y_i), \quad \text{für } i = n, \dots, 1 \end{aligned}$$

d.h. es handelt sich um  $n$  Teilprobleme (parametrische Optimierungsprobleme). Die Bellmansche Funktionalgleichung stellt eine Beziehung zwischen zwei aufeinander folgenden Wertfunktionen  $f_i^*$  und  $f_{i+1}^*$  dar und erlaubt es, bei bekanntem  $f_{i+1}^*$  die Funktion  $f_i^*$  und damit jeweils für eine weitere Stufe die Wertfunktion zu bestimmen.

Dann gilt ( $\Rightarrow$ ):

Ist  $y_i \in Y_i$  (wobei  $Y_i$  von  $x_{i-1}$  abhängen darf, d.h.  $Y_i = Y_i(x_{i-1})$ ), dann liefert jede Strategie, die sich als Lösung der oben definierten Folge von Optimierungsproblemen ergibt, mit Sicherheit nur optimale Prozesse (abhängig von  $x_0$ ) und  $f_1^*(x_0)$  ist das Minimum der Zielfunktion, d.h.  $f_1^*(x_0) = \min_y z(x_0, x_1, \dots, x_n, y_1, \dots, y_n)$ .

Wie liefert die Folge der Optimierungsprobleme eine Strategie?

Sei  $i = n$

$$\begin{aligned} f_n(x_{n-1}, y_n) &= c_n(x_{n-1}, y_n) + \overbrace{f_{n+1}^*(x_n)}^0 \\ f_n^*(x_{n-1}) &= \min_{y_n \in Y_n(x_{n-1})} f_n(x_{n-1}, y_n) \end{aligned}$$

$Y_i(x_{i-1})$  entstehen z.B., wenn Zustandsnebenbedingungen  $x_i \in X_i$  gefordert werden.

Falls für jedes feste  $x_{n-1}$  dies eindeutig lösbar ist, existiert eine Funktion  $h_n^*$  mit

$$y_n^* = h_n^*(x_{n-1}) \quad \text{optimale Entscheidungsfunktion für Stufe } n \text{ (bei bekanntem } x_{n-1}\text{)}$$

$$\begin{aligned} f_{n-1}(x_{n-2}, y_{n-1}) &= c_{n-1}(x_{n-2}, y_{n-1}) + f_n^*(x_{n-1}) \\ f_{n-1}^*(x_{n-2}) &= \min_{y_{n-1} \in Y_{n-1}(x_{n-2})} f_{n-1}(x_{n-2}, y_{n-1}) \end{aligned}$$

$$y_{n-1}^* = h_{n-1}^*(x_{n-2})$$

usw.

$$\begin{array}{ccccccc} & & T_1(x_0, y_1^*) & & T_2(x_1^*, y_2^*) & & T_3(x_2^*, y_3^*) & & \dots \\ & & \rightarrow & & \rightarrow & & \rightarrow & & \\ x_0 & & & x_1^* & & x_2^* & & & \\ y_1^* = h_1^*(x_0) & & & y_2^* = h_2^*(x_1^*) & & y_3^* = h_3^*(x_2^*) & & & \end{array}$$

**Allgemeinere Probleme bzw. Modifikationen**

## 1. Terminales Zielkriterium

Jedes Problem mit einer ZF

$$z(x, y) = \sum_{i=1}^n c_i(x_{i-1}, y_i) + c_{n+1}(x_n) \quad (17)$$

(die Funktion enthält einen zusätzlichen Summanden, der vom Endzustand abhängt,  $f_{n+1}^*(x_n) := c_{n+1}(x_n)$ ) kann man auf ein Problem mit  $z(x, y) = \varphi_{n+1}(x_n, z_n)$  zurückführen. Der Preis dafür ist eine zusätzliche Zustandsgröße und Zustandsgleichung pro Stufe.

Modell:  $x_i = T_i(x_{i-1}, y_i)$  und (17)

$$z_1 = c_1(x_0, y_1) \quad (= \varphi_1(x_0, y_1))$$

$$z_2 = z_1 + c_2(x_1, y_2) \quad (= \varphi_2(x_1, z_1, y_2))$$

...

$$z_i = z_{i-1} + c_i(x_{i-1}, y_i) \quad (= \varphi_i(x_{i-1}, z_{i-1}, y_i)) \text{ für } i = 1, \dots, n \text{ mit } z_0 = 0$$

$$z_{n+1} = z_n + c_{n+1}(x_n)$$

Es gilt:  $z_{n+1} = z(x, y) = \varphi_{n+1}(x_n, z_n) = z_n + c_{n+1}(x_n)$

## 2. Vektoren statt Skalare im Modell und in den Funktionalgleichungen

$$x_i \in \mathbb{R}^n, \quad i = 0, \dots, n$$

$$y_j \in \mathbb{R}^m, \quad j = 1, \dots, n$$

Vektortransformation

$$x_i = T_i(x_{i-1}, y_j), \quad x_i = Ax_i + By_j$$

$$y_j \in Y_j(x_{j-1}) \subseteq \mathbb{R}^m \quad Cy_j \leq b \quad \text{oder} \quad Cy_j + Dx_{j-1} \leq b_{j-1}$$

wobei  $A, B, C, D$  Matrizen sind.

$$z = z(x, y) = \varphi_{n+1}(x_n) \rightarrow \min$$

Die Funktionalgleichungen sind formal identisch.

Sei  $x_i = (x_i^1, \dots, x_i^n)^T$  und gibt es für ein  $x_i^k$  genau  $\alpha^{(k)}$  viele Werte, so gibt es  $\prod_{k=1}^n \alpha^{(k)}$  viele Optimierungsprobleme, d.h. der Rechenaufwand bei der Lösung wächst exponentiell mit der Dimension der Zustands- und Steuervariablen.

## 3. Nicht-additive Zielfunktion — monotone Separabilität

**Definition 22**

$z = (x, y)$  heißt *separabel*, falls gilt:

$\exists$  Verknüpfungsooperatoren  $\circ$  und Zerlegungsfunktionen  $g_i$ , sodass  $z$  darstellbar ist als

$$z(x, u) = g_1(c_1(x_0, y_1), \dots, c_n(x_{n-1}, y_n)) \text{ mit}$$

$$g_n(c_n(x_{n-1}, y_n)) := c_n(x_{n-1}, y_n)$$

$$g_i(c_i(x_{i-1}, y_i), \dots, c_n(x_{n-1}, y_n)) := c_i(x_{i-1}, y_i) \circ g_{i+1}(c_{i+1}(x_i, y_{i+1}), \dots, c_n(x_{n-1}, y_n))$$

wobei der Verknüpfungsooperator  $\circ$  von  $i$  abhängt.

Hilfssatz:

Sei  $z(x, y)$  separabel und gelte für alle  $i$ :

Die Funktionen  $g_i = c_i \circ g_{i+1}$  sind für jeden zulässigen Wert von  $c_i$  monoton wachsend in  $g_{i+1}$

$$\begin{aligned} \Rightarrow \min_{y_1, \dots, y_n} g_i(c_i, g_{i+1}) &= \min_{y_i} g_i(c_i, \min_{y_{i+1}, \dots, y_n} g_{i+1}) \\ &= \min_{y_i} (c_i \circ \min_{y_{i+1}, \dots, y_n} g_{i+1}) \end{aligned}$$

$\Rightarrow$  Die Rekursionsformeln sind wegen des Hilfsatzes analog zum oben behandelten additiven Fall, die monotone Separabilität vorausgesetzt.

Rekursionsgleichungen

Dynamische Optimierung bei monoton separablen Problemen

$$\begin{aligned} f_{n+1}^*(x_n) &:= 0 \text{ bzw. } 1 \quad (\text{oder } \varphi_{n+1}(x_n, z_n)) \\ f_i(x_{i-1}, y_i) &= c_i(x_{i-1}, y_i) \circ f_{i+1}^*(T_i(x_{i-1}, y_i)) \\ f_i^*(x_{i-1}) &= \min_{y_i \in Y_i(x_{i-1})} f_i(x_{i-1}, y_i), \quad \text{bzw. } \max \\ &= \min_{y_i, \dots, y_n} g_i(c_i, \dots, c_n), \quad \text{bzw. } \max \\ i &= n, \dots, 1 \\ f_1^*(x_0) &= \min_{y_1, \dots, y_n} z, \quad \text{bzw. } \max \\ f_i^*(x_{i-1}) &= \min_{y_i} (c_i(x_{i-1}, y_i) \circ f_{i+1}^*(x_i)) \end{aligned}$$

## 14.2 Anwendung: Lagerhaltungsmodelle

Ein Lager hat die Funktion eines Puffers innerhalb des Gütestromes, der bei einem Güterprozess vom Einkauf ggf. über verschiedene Produktionsstufen zum Verbraucher fließt. Ein Lager dient der zeitlichen Entkopplung von Tätigkeiten, etwa zwischen Produktion und Auslieferung, somit erlaubt es den Ausgleich zwischen Produktions- und Nachfrageschwankungen. Lager können an verschiedenen Stellen im Produktionsprozess auftreten, z.B. Einkaufslager, Zwischenlager, Fertigungslager, Handelslager und Ersatzteillager.

Zentrale Frage: Wann soll wie viel bestellt werden?

D.h. es ist eine Bestellpolitik gesucht, die Bestellzeitpunkte und Bestellmenge optimal festlegt, sodass die gesamten Lagerhaltungskosten minimal werden.

Grundsätzlicher Aufbau der Lagerhaltung:



- Ist Lagerhalter = Produzent, dann ist Bestellung = Auflegen einer Produktionsserie (Fertigungslos)
- Nachfragebefriedigung = Deckung des Bedarfs

Es gelten im weiteren Verlauf die folgenden Beschränkungen:

- nur ein Lager

- nur ein zu lagerndes Gut (also keine Wechselwirkung mit anderen gelagerten Gütern)
- Lagergut ist diskret (Stückgut) oder kontinuierlich (unbeschränkt teilbar)
- zwischen Bestellung und Lieferung bzw. Verfügbarkeit liegt in der Praxis meist eine gewisse Lieferzeit (Wiederbeschaffungszeit), die gelegentlich vernachlässigt wird

### 14.2.1 Klassische Losgrößenmodelle

#### Klassisches Modell von Harris-Wilson

Voraussetzungen/Annahmen:

- Lagerung von einem Gut während unbegrenztem Planungszeitraum (statisches Lagerhaltungsmodell)
- Nachfrage pro Zeiteinheit (ZE) (Abgangsrate)  $\mu$  ist konstant.
- Jeder Bedarf wird vom Lager sofort befriedigt (keine Fehlmengen).
- Lieferzeit  $\lambda$  ZE ist konstant
- Beschaffungskosten  $B = K \cdot \delta(Q) + c \cdot Q$ , mit  $\delta(Q) = \begin{cases} 1, & \text{falls } Q > 0 \\ 0, & \text{falls } Q = 0 \end{cases}$  und
 

$Q \geq 0$	Bestellmenge
$K \geq 0$	fixe Bestellkosten (Ausfertigung einer Bestellung, unabhängig von deren Quantität)
$c$	Preis (Bestellkostensatz) pro ME
$c \cdot Q$	variable Bestellkosten
- Lagerungskosten pro ME und ZE sind gleich  $h > 0$
- Gesamte Lagerungskosten sind am kleinsten, falls die Lieferung genau dann eintrifft, wenn Bestand = 0 (Fehlmenge sind nicht zugelassen).

Da Abgangsrate  $\mu$  und Lieferzeit  $\lambda$  konstant sind  $\Rightarrow$  optimal, wenn jeweils die gleiche Menge  $Q$  bestellt wird ( $Q$  ist also auch konstant).

Der Bestellzyklus ist die Zeit  $\tau$  zwischen zwei aufeinanderfolgenden Bestellungen:

$$\tau = \frac{Q}{\mu} \approx \text{Länge einer Bestellperiode (BP)}$$

Auch  $\tau$  ist konstant.

Aus der Abbildung ist unmittelbar der mittlere Lagerbestand  $\frac{Q}{2}$  während der Periode ablesbar.

Zielfunktion:

$$\begin{aligned} \min C &= \text{minimale durchschnittliche Gesamtkosten pro ZE} \\ C &= \text{Kosten pro BP/Länge BP} \end{aligned}$$

$$\begin{array}{rcc} \text{Kosten} & = & \text{Beschaffungskosten} + \text{Lagerungskosten} \\ \text{pro BP} & & \text{pro BP} \quad \quad \quad \text{pro BP} \\ & & \downarrow \quad \quad \quad \downarrow \\ & & K + c \cdot Q \quad \quad \quad h \cdot \tau \cdot \frac{Q}{2} \end{array}$$

Für die mittleren Gesamtkosten pro ZE gilt

$$\begin{aligned} C &= \frac{1}{\tau} \cdot \left( K + c \cdot Q + h \cdot \tau \cdot \frac{Q}{2} \right) \text{ mit } \tau = \frac{Q}{\mu} \\ C &= f(Q) = \frac{h}{2} \cdot Q + \frac{K \cdot \mu}{Q} + c \cdot \mu \text{ mit } Q > 0 \end{aligned}$$

Da  $f(Q)$  konvex und für  $Q > 0$  differenzierbar ist, kann die optimale Losgröße  $Q^*$  durch Nullsetzen der ersten Ableitung von  $f(Q)$  gefunden werden.

Klassische Losgrößenformel

$$\begin{aligned} f'(Q) &= \frac{h}{2} - \frac{K\mu}{Q^2} \stackrel{!}{=} 0 \\ Q^* &= \sqrt{\frac{2K\mu}{h}} \end{aligned}$$

Als optimale Periodenlänge ergibt sich  $\tau^* = \frac{Q^*}{\mu} = \sqrt{\frac{2K}{h\mu}}$ .

Der optimale Bestellpunkt  $r^*$  (Lagerbestand bei dessen Unterschreitung bestellt wird) sei

$$r^* = \begin{cases} \lambda\mu, & \text{falls } \lambda < \tau^* \\ \left(\lambda - \frac{\lambda}{\tau^*}\right) \cdot \mu, & \text{sonst} \end{cases}$$

Die minimalen Gesamtkosten pro ZE betragen

$$C^* = \frac{h}{2} \cdot Q^* + \frac{K \cdot \mu}{Q^*} + c \cdot \mu = \sqrt{2Kh\mu} + c\mu$$

Wie man sieht sind  $Q^*$  und  $\tau^*$  unabhängig vom Bestellkostensatz  $c$ , da dieser über die Zeit hinweg konstant ist. Man spricht auch von einer  $(r, Q)$ -Bestellpolitik, da eine Bestellung immer dann aufgegeben wird, wenn der Lagerbestand höchstens gleich dem Bestellpunkt  $r$  ist, und zwar wird dann das Lager bis zur Bestellgrenze  $Q$  aufgefüllt.

### Modifikation des Harris-Wilson-Modells

Im klassischen Losgrößenmodell sind Fehlmengen nicht erlaubt. Eine Fehlmenge tritt auf, wenn das Lager nicht lieferbar ist, also mehr Menge nachgefragt wird als vorhanden ist. Es kann aber kostengünstiger sein, Fehlmengen zuzulassen, da hierdurch die optimale Periodenlänge vergrößert werden kann, was eine Reduzierung der auflagefixen Kosten pro ZE bewirkt.

- Fehlmengen (unbefriedigter Bedarf) werden vorgemerkt, d.h. sie werden von der nächsten verfügbaren Lieferung unmittelbar nach deren Eintreffen gedeckt.
- Fehlmengenkosten  $p$  pro ME und ZE: „Goodwill-Verluste“ und Verluste durch „Abspringen“ der Kunden (schwer messbar)
- $\mu$  und  $\lambda$  sind wie zuvor konstant

Konsequenzen:

- Fehlmengen als negativen Bestand interpretieren
- Bei Eintreffen einer Lieferung ist Bestand =  $-v < 0$  (also nicht = 0)
- Der Lagerbestand nach Wiederauffüllung des Lagers sei  $y = Q - v$

Im Verlauf einer Periode ist der Bestand

- nach  $\tau' = \frac{y}{\mu}$  ZE gleich 0 und
- nach  $\tau'' = \frac{v}{\mu} = \frac{Q-y}{\mu}$  ZE gleich  $v$

Bestellperiode:  $\tau = \tau' + \tau'' = \frac{Q}{\mu}$

Trifft zur Zeit  $t = 0$  eine Lieferung  $Q$  ein, wird sofort die Fehlmenge  $v$  realisiert und mit der Restmenge  $y = Q - v$  wird das Lager aufgefüllt.

Nun sollen die durchschnittlichen Gesamtkosten pro ZE

$$C = \frac{\text{Kosten pro BP}}{\text{Länge der BP}}$$

optimiert werden.

1. Während Intervall (Länge  $\tau'$ ) mit Bestand  $\geq 0$  ist der mittlere Bestand  $\frac{y}{2}$ .  
 $\Rightarrow$  Lagerkosten pro Bestellmenge  $h \cdot \tau' \cdot \frac{y}{2}$

2. Während Intervall (Länge  $\tau''$ ) mit Fehlbestand (also Bestand  $< 0$ ) ist der mittlere Fehlbestand  $\frac{y}{2} = \frac{Q-y}{2}$ .

$\Rightarrow$  Fehlmengenkosten pro Bestellperiode:

$$p \cdot \tau'' \cdot \frac{Q-y}{2}$$

$$C = \frac{1}{\tau} \cdot \left( K + c \cdot Q + h \cdot \tau' \frac{y}{2} + p \cdot \tau'' \cdot \frac{Q-y}{2} \right)$$

$$\text{mit } \tau = \frac{Q}{\mu}, \tau' = \frac{y}{Q}, \tau'' = \frac{Q-y}{Q} \text{ (Strahlensatz)}$$

$$\Rightarrow C = f(Q, y) = \frac{h}{2} \cdot \frac{y^2}{Q} + \frac{p \cdot (Q-y)^2}{2 \cdot Q} + \frac{K \cdot \mu}{Q} + c \cdot \mu$$

Durch Nullsetzen der partiellen Ableitungen  $\frac{\partial C}{\partial y}$  und  $\frac{\partial C}{\partial Q}$  folgen die optimalen Werte:

$$y^* = \sqrt{\frac{2K\mu}{h}} \cdot \sqrt{\frac{p}{h+p}}, \quad Q^* = \sqrt{\frac{2K\mu}{h}} \cdot \sqrt{\frac{h+p}{p}}$$

Desweiteren folgt

$$\tau^* = \frac{Q^*}{\mu} = \sqrt{\frac{2K}{h\mu}} \cdot \sqrt{\frac{h+p}{p}}, \quad \frac{\tau'^*}{\tau^*} = \frac{y^*}{Q^*} = \frac{p}{h+p}$$

und die minimalen Lagerhaltungskosten pro ZE betragen

$$C^* = \sqrt{2Kh\mu} \cdot \sqrt{\frac{p}{h+p}} + c\mu$$

Das klassische Losgrößenmodell, in dem Fehlmengen nicht erlaubt sind, kann als Modell mit unendlich großen Fehlmengenkosten interpretiert werden. Führt man in den Formeln für  $y^*$ ,  $Q^*$  und  $\tau^*$  den Grenzübergang  $p \rightarrow \infty$  durch, so erhält man die entsprechenden Formeln des klassischen Modells, da

$$\lim_{p \rightarrow \infty} \frac{h+p}{p} = \lim_{p \rightarrow \infty} \frac{p \left( \frac{h}{p} + 1 \right)}{p} = 1$$

Ist der Quotient  $\frac{h+p}{p} > 1 \Rightarrow Q^*(p) > Q^*(\infty)$  und  $C^*(p) < C^*(\infty)$

### 14.2.2 Dynamisches Deterministisches Modell

Besteht der Planungszeitraum aus (endlich vielen) einzelnen Planungsperioden, wobei pro Periode höchstens eine Bestellung aufgegeben wird (i.A. zu Beginn der Periode) und wird die Nachfrage als deterministische Größe betrachtet, so spricht man von einem dynamischen deterministischen Lagerhaltungsmodell.

- es liege ein endlicher Planungszeitraum  $[t_a, t_e]$  vor
- die Nachfragerate über  $[t_a, t_e]$  sei nicht konstant
- der Planungszeitraum  $[t_a, t_e]$  sei in  $n$  gleich lange Perioden eingeteilt  
 $t_0 := t_a, t_1, \dots, t_{n-1}, t_n := t_e$  sind die Endpunkte der Perioden  
 $\tau := t_j - t_{j-1}$  für  $j = 1, \dots, n$  (konstant)



Voraussetzungen/Annahmen:

- Lieferzeit  $\lambda$   
Bestellzeitpunkte  $t'_j := t_j - \lambda$ ,  $j = 0, \dots, n-1$  (Bestellung ist zu den Zeitpunkten  $t_j$  verfügbar bzw. trifft dann ein)
- $z_j \geq 0$  Nachfrage in der  $j$ -ten Periode  $[t_{j-1}, t_j]$ ,  $j = 1, \dots, n$   
 $u_j \geq 0$  zu Beginn von Periode  $j$  gelieferte Bestellmenge  
 $x_j$  Lagerbestand am Ende der Periode  $j$  ( $t_j - \varepsilon$ )  
(unmittelbar vor Auffüllen zur Zeit  $t_j$ ),  $j = 1, \dots, n$

$\Rightarrow$  Es gilt dann die Lagerbilanzgleichung

$$x_j = x_{j-1} + u_j - z_j, \quad j = 1, \dots, n$$

- Fehlmengen sind nicht zugelassen, d.h.  $x_j \geq 0$ ,  $j = 0, \dots, n$   
 $\Rightarrow x_{j-1} + u_j \geq z_j$  und nur  $u_j$  ist beeinflussbar.  
Anfangs- und Endbestand sind vorgegeben und o.B.d.A. gilt  $x_0 = x_n = 0$
- Nachfrageentwicklung in  $j$ -ter Periode:  
 $\beta_j(t)$  ist der bis zur Zeit  $t$  in der Periode  $j$  (gerechnet vom Beginn der Periode) angefallene Bedarf,  $0 \leq t \leq \tau$  mit  $\beta_j(0) = 0$ ,  $\beta_j(\tau) = z_j$
- Kosten:  
K fixe Bestellkosten (für alle Perioden gleich)  
c Preis pro ME  
h Lagerungskosten pro ZE und ME

$\rightarrow$  Lagerungskosten in der  $j$ -ten Periode ergeben sich dann zu

$$\begin{aligned} & K \cdot \delta(u_j) + c \cdot u_j + h \cdot \int_0^\tau \overbrace{(x_{j-1} + u_j - \beta_j(t))}^{\text{mittlerer Bestand}} dt \\ & \quad \underbrace{x_{j-1} + u_j}_{\text{Bestand Zeitpunkt } t} \\ = & K \cdot \delta(u_j) + c \cdot u_j + h \cdot \tau \cdot x_j + h \cdot \int_0^\tau (z_j - \beta_j(t)) dt \end{aligned}$$

Diskussion von  $h \cdot \int_0^\tau (z_j - \beta_j(t)) dt$  „Kosten für die Mindestreserve“

Lagerungskosten für die zu Beginn der  $j$ -ten Periode zur Verfügung gestellten  $z_j$  ME, um die Nachfrage in dieser Periode zu befriedigen.

Die mittlere Mindestreserve in Periode  $j$  ist

$$\xi_j = \frac{1}{\tau} \cdot \int_0^\tau (z_j - \beta_j(t)) dt$$

Also sind die Kosten für diese Mindestreserve:  $h \cdot \tau \cdot \xi_j$  (unabhängig von der Bestellmenge)

Ist die Nachfrage  $\beta_j(t)$  konstant, also  $\beta_j(t) = \frac{z_j}{\tau} \cdot t$ ,  $\Rightarrow \xi_j = \frac{z_j}{2}$  das bisherige Modell.

Zur Optimierung der Bestellpolitik genügt es, die Minimierung von

$$\sum_{j=1}^n (K \cdot \delta(u_j) + \underbrace{c \cdot u_j}_{=c \sum z_j} + h \cdot \tau \cdot x_j)$$

zu betrachten, da die variablen Bestellkosten über alle Perioden eine Konstante ist.

$$\begin{aligned} \text{Es gilt: } \sum_{j=1}^n u_j &= \sum_{j=1}^n (x_j - x_{j-1} + z_j) \\ &= \sum_{j=1}^n z_j + x_n - x_0 \\ &= \sum_{j=1}^n z_j \quad (\text{also konstant, d.h. } c \cdot u_j \text{ lässt sich rausziehen}) \end{aligned}$$

Damit sieht das dynamische Optimierungsmodell wie folgt aus:

$$\begin{aligned} \min \quad & \sum_{j=1}^n \underbrace{(K \cdot \delta(u_j) + h \cdot \tau \cdot x_j)}_{f(u_j, x_j)} \\ \text{s.d.} \quad & x_j = x_{j-1} + u_j - z_j, \quad j = 1, \dots, n \\ & x_0 = x_n = 0 \\ & x_j \geq 0, \quad j = 1, \dots, n-1 \\ & u_j \geq 0, \quad j = 1, \dots, n \end{aligned}$$

### 14.2.3 Beispiel eines 3-stufigen Lagerhaltungsmodells mit Dynamischer Programmierung

Lagerhaltungsmodell

Stufe 1: Zeitintervall  $[t_0, t_1)$

Stufe 2: Zeitintervall  $[t_1, t_2)$

Stufe 3: Zeitintervall  $[t_2, t_3)$

$x_i$ : Lagerbestand am Ende der Stufe  $i$  (Beginn  $i+1$ )

$y_i$ : Bestellmengeentscheidung auf Stufe  $i$

$d_i$ : Nachfrage auf der Stufe  $i$  (bekannt, wird realisiert, d.h. keine Fehlmengen)

Zustandsgleichungen

$$x_1 = x_0 + y_1 - d_1$$

$$x_2 = x_1 + y_2 - d_2$$

$$x_3 = x_2 + y_3 - d_3$$

$$x_{i+1} = T(x_i, y_{i+1}), \quad i = 0, 1, 2$$

Stufenkosten:  $c_i(x_{i-1}, y_i)$

abhängig von Anfangsbestand und der Bestellentscheidung der Stufe

Gesamtkosten

$$= \sum_i (\text{Stufenkosten Stufe } i)$$

$$= c_1(x_0, y_1) + c_2(x_1, y_2) + c_3(x_2, y_3) (+c_4(x_3))$$

Die Optimierung beginnt mit der letzten Stufe und bezieht schrittweise die vorgehenden Stufen ein.

Zusätzliche (bisher nicht formulierte) Bedingung:

Der Endzustand sei vorgegeben:  $x_3 = x_3^* = 0$

$$x_3 = x_2 + y_3 - d_2 = 0 = x_3^* \Rightarrow y_3 = d_2 - x_2$$

liegt in Abhängigkeit von  $x_2$  fest (d.h.  $x_2$  ist noch unbekannt)

$$y_3^* = y_3^*(x_2) = h_3^*(x_2) = d_2 - x_2 (\geq 0)$$

Danach für jeden Zustand  $x_1$  eine optimale Entscheidung  $y_2^*(x_1) = h_2^*(x_1)$  für Stufe 2 bestimmen.

$$f_2^*(x_1) = \min_{y_2} \left( \underbrace{c_2(x_1, y_2)}_{\text{Stufenkosten}} + \underbrace{f_3^*(x_1 + y_2 - d_1)}_{\substack{\text{min. Kosten} \\ \text{für Restprozess}}} \right)$$

$$\text{mit } f_3^*(x_2) = \min_{y_3: x_2 + y_3 - d_2 = 0} (c_3(x_2, y_3)) = c_3(x_2, d_2 - x_2)$$

#### 14.2.4 Das Verfahren von Wagner und Whitin

Beim Verfahren von Wagner und Whitin wird das dynamische Optimierungsmodell aus Abschnitt 14.2.2 benutzt, mit  $x_j$  anstelle von  $x_{j-1}$  in der Zielfunktion und der Annahme, dass  $x_j = x_{j-1} + u_j - z_j$  eindeutig nach  $x_{j-1}$  auflösbar ist.

In diesem Fall ist die Optimierungsrichtung der Dynamischen Optimierung (Rückwärtsrechnung) umkehrbar. Wie zuvor (Abschnitt 14.2.2) gezeigt, ist aufgrund der Bedingung  $x_0 = x_{n+1} = 0$  die Summe der variablen Beschaffungskosten über den gesamten Planungszeitraum hinweg eine Konstante; daher müssen die variablen Beschaffungskosten bei der Minimierung der Lagerhaltungskosten nicht berücksichtigt werden.

Wird also von den variablen Beschaffungskosten abgesehen, so genügen die minimalen Kosten  $C_j^*(x_j)$  bei gegebenem Lagerbestand  $x_j$  am Ende von Periode  $j$  der Bellmanschen Funktionalgleichung (den Bellmanschen Rekursionsformeln):

$$\begin{aligned} C_0^*(x_0) &:= 0 && \text{Umkehrung des} \\ C_j^*(x_j) &= \min_{0 \leq u_j \leq x_j + z_j} (K\delta(u_j) + h\tau x_j + C_{j-1}^*(x_{j-1})) && \text{Rechenverlaufs} \\ &\text{mit } x_{j-1} = x_j - u_j + z_j, \text{ für } j = 1, \dots, n \end{aligned}$$

und es gilt:  $C_n^*(x_n) = C_n^*(0)$  ist das Minimum der relevanten Kosten.

Die Funktionalgleichungen lassen sich nun direkt numerisch lösen. Wager und Whitin haben, die spezielle Struktur des vorliegenden dynamischen Lagerhaltungsproblems ausnutzend, einen Algorithmus zur Bestimmung einer optimalen Bestellpolitik entwickelt, der wesentlich weniger Rechenaufwand erfordert.

Eigenschaften der Bellmanschen-FG:

$(u_1^*, \dots, u_n^*)$  optimale Bestellpolitik

$(x_0^*, x_1^*, \dots, x_n^*)$  zugehörige Folge  $x_j = x_{j-1} + u_j - z_j$   
von Lagerbeständen mit  $x_0^* = x_n^* = 0$ .

Dann heißt  $(x_{j-1}^*, u_j^*)$  ein „optimales Lösungspaar“ (aus dem aktuellen Bestand  $\Rightarrow$  die nachfolgende Bestellung).

(E1) Für ein optimales Lösungspaar  $(x_{j-1}^*, u_j^*)$  gilt für alle  $j = 1, \dots, n$   
 entweder  $x_{j-1}^* = 0 \wedge u_j^* > 0$   
 oder  $x_{j-1}^* > 0 \wedge u_j^* = 0$   
 Aus dieser Eigenschaft folgt

(E2) Für eine optimale Bestellung  $u_j^*$  in der  $j$ -ten Periode gilt  
 $u_j^* = \{0, z_j, z_j + z_{j+1}, \dots, z_j + z_{j+1} + \dots + z_n\}$ ,  $j = 1, \dots, n$  ( $n - j + 2$  mögliche Werte)  
 Dies ergibt sich daraus, dass Lieferungen nur bei Periodenbeginn und geräumtem Lager erfolgen und so dimensioniert sein müssen, dass der Bedarf bis zur nächsten Lieferung gedeckt ist.

Damit lässt sich die Bellmansche-FG vereinfachen:

$$C_0^* = 0, C_1^* = K, C_j^* = \min_{k=1, \dots, j} \{p_j(k)\},$$

$$\text{mit } p_j(k) := K + h\tau \cdot \underbrace{\sum_{i=k}^j \sum_{\gamma=i+1}^j z_\gamma}_{z_{k+1} + 2z_{k+2} + \dots + (j-k)z_j} + C_{k-1}^*$$

Es gilt also:

$$C_1^* = p_1(1) = K + C_0^* = K$$

$$p_j(\kappa_j^*) = \min_{k=1, \dots, j} p_j(k) = C_j^*$$

Es muss die Minimalstelle berechnet werden, wird diese für mehr als einen Index  $k$  angenommen, so ist es günstig, den größten Index von diesen mit  $\kappa_j^*$  zu bezeichnen.

### Algorithmus

- Für  $j = 1$  ist  $\kappa_j^* = \kappa_1^* = 1$
- Obige FG in Vorwärtsrechnung für  $j = 2, \dots, n$  lösen, d.h.  $C_j^*$  und  $\kappa_j^*$  ermitteln
- Optimale Bestellmengen durch Rückwärtsrechnung bestimmen:  
 sei  $s := \kappa_n^*$   
 $\Rightarrow u_s^* = z_s + z_{s+1} + \dots + z_n \quad \wedge \quad u_{s+1}^* = \dots = u_n^* = 0$   
 sei weiter  $m := \kappa_{s+1}^*$   
 $\Rightarrow u_m^* = z_m + z_{m+1} + \dots + z_{s-1} \quad \wedge \quad u_{m+1}^* = \dots = u_{s-1}^* = 0$

Die Rechnung kann durch Ausnutzung der folgenden Eigenschaft stark vereinfacht werden:

- (E3) Die Folge der  $\kappa^*$  ist monoton wachsend, d.h.  $\kappa_1^* \leq \dots \leq \kappa_n^*$   
 $\Rightarrow$  Bei Lösung der FG nur Minimum der Werte  $p_j(\kappa_{j-1}^*), \dots, p_j(j)$  (auf die Betrachtung von  $p_j(1), \dots, p_j(\kappa_{j-1}^* - 1)$  kann verzichtet werden).
- (E4) Die  $p_j(k)$  genügen für  $j = 2, \dots, n$  der Rekursion  
 $p_j(k) = p_{j-1}(k) + (j - k)h\tau z_j$ ,  $k = 1, \dots, j - 1$   
 $p_j(j) = K + C_{j-1}^*$   
 $\Rightarrow$  Die explizite Berechnung der Doppelsumme kann entfallen.

Die minimale gesamten Lagerungskosten  $C^*$  ergeben sich zu

$$C^* = C_n^* + c \cdot \sum_{j=1}^n z_j + h\tau \cdot \sum_{j=1}^n \xi_j$$

### 14.3 Anwendung: Marketing-Beispiel

Die Unternehmung H. hat für die Einführung eines neuen Waschmittels einen Etat von 5 Mio €. Die Produkteinführung soll in drei Phasen abgewickelt werden:

1. Herabsetzen der Einführungspreise, um Erstkäufer zu gewinnen.
2. Werbekampagne: Erstkäufer überzeugen, dass der regulären Preis akzeptabel ist.
3. Sonderangebote und Werbemaßnahmen, um die regulären Käufer zu halten.

Man verteile den Etat optimal auf die drei Phasen.

→ Man muss die Ausgaben in den Phasen mit resultierendem Marktanteil in Verbindung bringen.

Auf der Basis von Markkanalysen gibt es die folgenden Schätzungen

Ausgaben in Phase 1 [Mio €], $y_1$	0	1	2	3	4	5
Resultierender Marktanteil [%], $M_1(y_1)$	0	10	15	22	27	30
Ausgaben in Phase 2, $y_2$	0	1	2	3	4	
Anteil gehaltener Kunden, $M_2(y_2)$	30	55	70	80	85	
Ausgaben in Phase 3, $y_3$	0	1	2	3	4	
Anteil gehaltener Kunden, $M_3(y_3)$	50	70	85	90	93	

$M_1$  in Phase 1 erzielter Marktanteil  $M_1(y_1)$

$M_2, M_3$  davon in Phase 2 (3) gehaltener Marktanteil,  $M_2(y_2), M_3(y_3)$

$y_i$  in Phase  $i$  eingesetzter Betrag (in Mio €),  $i = 1, 2, 3$  (Entscheidung)

Offenbar muss man  $y_1 \geq 1$  fordern.

Modell:

max  $M_1 \circ M_2 \circ M_3 \cdot 1$  (künstlich, um Iteration zu starten,  $M_4(x_3) = 1$ )

mit  $y_1 + y_2 + y_3 \leq 5$

$y_1 \geq 1$   $y_2, y_3 \geq 0$  und ganzzahlig

Normierung der  $M_i$  auf  $0 \leq M_i \leq 1$  durch Division durch 100%.

Dynamische Optimierung:

Stufen  $i = 1, 2, 3$  (entsprechen den Phasen)

EV:  $y_i$

Zustände: Rest-Etats nach Stufe  $i$

$x_0 = 5$  (5 Mio € sollen eingesetzt werden)

$x_i = x_{i-1} - y_i$  und

$x_3 = 0$

Rekursion:

$f_4^*(x_3) = f_4^*(0) = M_4(0) = 1$

$f_i^*(x_{i-1}) = \max(M_i(y_i) \circ f_{i+1}^*(\overbrace{x_{i-1} - y_i}^{x_i}))$ ,  $i = 3, 2, 1$

Da alle  $M_i > 0$  sind, ist die Zielfunktion monoton separabel.

$i = 3$

$x_3 = 0 \Rightarrow x_3 = x_2 - y_3 = 0 \Rightarrow y_3 = x_2$ , d.h die Entscheidung  $y_3$  ist festgelegt.

$$f_3^*(x_2) = \max(M_3(y_3) \cdot 1) = M_3(x_2) \text{ und } y_3^* = h_3^*(x_2) = x_2$$

(Die Endbedingung erzwingt die Entscheidung auf der letzten Stufe.)

$x_2$	0	1	2	3	4
$y_3^* = h_3^*(x_2) = x_2$	0	1	2	3	4
$f_3^*(x_2) = M_3(x_2)$	0,50	0,70	0,85	0,90	0,93

$i = 2$

$x_2 = x_1 - y_2 \geq 0 \Leftrightarrow y_2 \leq x_1$  (in Phase 2 wird höchstens soviel investiert, wie nach Phase 1 noch übrig ist)

$$f_2^*(x_1) = \max_{0 \leq y_2 \leq x_1} \underbrace{(M_2(y_2) \circ f_3^*(x_1 - y_2))}_{= f_2(x_1, y_2)}$$

		$f_2(x_1, y_2)$						
$x_1$		$y_2 = 0$	$y_2 = 1$	$y_2 = 2$	$y_2 = 3$	$y_2 = 4$	$y_2^*$	$f_2^*(x_1)$
0		0,150	-	-	-	-	0	0,150
1		0,210	0,275	-	-	-	1	0,275
2		0,255	0,385	0,350	-	-	1	0,385
3		0,270	0,468	0,490	0,400	-	2	0,490
4		0,279	0,495	0,595	0,560	0,425	2	0,595

$$f_2(3, 0) = M_2(0) \circ f_3^*(3 - 0) = 0,30 \cdot 0,90 = 0,270$$

$$f_2(4, 3) = M_2(3) \circ f_3^*(4 - 3) = 0,80 \cdot 0,70 = 0,560$$

$i = 1$

$$f_1^*(x_0) = f_1^*(5) = \max_{1 \leq y_1 \leq x_0=5} (M_1(y_1) \circ f_2^*(5 - y_1))$$

$x_0 = 5$	$y_1 = 1$	$y_1 = 2$	$y_1 = 3$	$y_1 = 4$	$y_1 = 5$	$y_1^*$	$f_1^*(5)$
	0,0595	0,0735	0,0847	0,0743	0,450	3	0,0847

$\Rightarrow$  Optimale Politik: In Phase 1 3 Mio €, danach je 1 Mio € einsetzen (Vorwärtsrechnung).

Optimale Marktanteil: 8,47%

## 15 Nichtlineare Optimierung

Ein nichtlineares Optimierungsproblem liegt dann vor, wenn die Zielfunktion und die in den Nebenbedingungen vorkommenden Funktionen der Entscheidungsvariablen nicht mehr sämtlich lineare Funktionen darstellen. Beispiel: In der Produktionsplanung sind die Verkaufspreise oder Produktionskosten pro ME nicht mehr konstant. Dies führt auf Optimierungsprobleme mit nichtlinearer Zielfunktion. Eine besondere Schwierigkeit besteht darin, dass man (bei Minimierungsaufgaben) zwischen lokalen und globalen Minima unterscheiden muss.

Für die nichtlineare Optimierung (NLP) gibt es kein „Standardverfahren“ (wie den Sim-

plexalgorithmus für lineare Optimierung, Abschnitt 2), stattdessen sind für verschiedene Klassen nichtlinearer Optimierungsprobleme geeignete Lösungsverfahren entwickelt worden, die spezielle Eigenschaften des jeweiligen Problemtyps ausnutzen.

Man unterscheidet zwischen unstringierten und restringierten Optimierungsaufgaben. Bei letzterem ist der zulässige Bereich eine echte Teilmenge des  $\mathbb{R}^n$ . Eine spezielle Klassen restringierter Optimierungsprobleme sind die konvexen Optimierungsprobleme, bei denen die Zielfunktion konvex und der zulässige Bereich eine konvexe Menge ist. Bei quadratischen konvexen Optimierungsproblemen ist die Zielfunktion quadratisch und der zulässige Bereich der gesamte  $\mathbb{R}^n$  oder ein konvexes Polyeder.

## 15.1 Grundlagen

Sei das folgenden Optimierungsproblem mit Zielfunktion  $f$  und zulässigem Bereich  $M$  zugrundegelegt:

$$\begin{array}{ll} \min & f(x) \\ \text{u.d.N.} & x \in M \end{array} \quad (18)$$

Dabei ist  $f$  eine auf dem  $\mathbb{R}^n$  definierte Funktion und  $\emptyset \neq M \subseteq \mathbb{R}^n$ . Ein Punkt  $x \in M$  heißt *zulässige Lösung* von (18). Bei der NLP müssen zwischen globalen und lokalen Minima unterschieden werden, im Gegensatz zu LP, wo lokale und globale Minima zusammenfallen. Ein Punkt  $x^* \in M$  heißt *globaler Minimalpunkt* oder optimale Lösung von (18), wenn für alle  $x \in M$ :  $f(x^*) \leq f(x)$  ist.

Sei  $U_\epsilon(\bar{x}) := \{x \in \mathbb{R}^n \mid |x - \bar{x}| < \epsilon\}$  die  $\epsilon$ -Umgebung mit  $\epsilon > 0$  eines Punktes  $\bar{x} \in \mathbb{R}^n$ .  $U_\epsilon(\bar{x})$  ist die Menge aller Punkte im Inneren einer Kugel des  $\mathbb{R}^n$  mit dem Mittelpunkt  $\bar{x}$  und dem Radius  $\epsilon$ .  $\bar{x} \in M$  heißt *lokaler Minimalpunkt* von  $f$  auf  $M$ , wenn es eine Zahl  $\epsilon > 0$  mit  $f(\bar{x}) \leq f(x)$  für alle  $x \in M \cap U_\epsilon(\bar{x})$  gibt.

## 15.2 Konvexe Kuhn-Tucker-Theorie

Da eine optimale Lösung von (18) ein globaler Minimalpunkt von  $f$  auf  $M$  ist, ist man an Zielfunktionen interessiert, bei denen lokale und globale Minimalpunkte zusammenfallen. Eine Menge  $M$  heißt *konvex*, wenn für je zwei Punkte  $x', x'' \in M$  die Verbindungsstrecke von  $x'$  und  $x''$ , d.h. die Menge aller Punkte  $x = \lambda x' + (1 - \lambda)x''$  mit  $0 \leq \lambda \leq 1$  zu  $M$  gehören.

Sei nun  $M \subseteq \mathbb{R}^n$  eine konvexe Menge. Dann heißt eine Funktion  $f : M \rightarrow \mathbb{R}$  *konvex*, wenn für je zwei Punkte  $x', x'' \in M$  und alle  $\lambda \in (0, 1)$ :  $f(\lambda x' + (1 - \lambda)x'') \leq \lambda f(x') + (1 - \lambda)f(x'')$  gilt. Für eine auf  $\mathbb{R}$  definierte Funktion  $f$  bedeutet dies, dass die Verbindungsstrecke der Punkte  $(x', f(x'))$  und  $(x'', f(x''))$  nirgendes unterhalb der Kurve liegt.

### Satz 14

Sei  $M \subseteq \mathbb{R}^n$  eine konvexe Menge,  $f : M \rightarrow \mathbb{R}$  konvex und  $a \in \mathbb{R}$ . Dann gilt:

- a) Jeder lokale Minimalpunkt von  $f$  auf  $M$  ist auch ein globaler Minimalpunkt.
- b) Die abgeschlossene Menge  $M_a = \{x \in M \mid f(x) \leq a\}$  ist konvex.

Das *konvexe Optimierungsproblem* hat die Form

$$\begin{array}{ll} \min & f(x), x \in \mathbb{R}^n \\ \text{s.d.} & g_i(x) \leq 0, \text{ für } i = 1, \dots, m \text{ (konvex} \Rightarrow \text{Linearkombination auch konvex)} \end{array}$$

(Insbesondere sind alle LOPs konvex, da sich jedes Gleichungssystem in Form zweier Ungleichungssysteme schreiben lässt, die beide konvexen Mengen entsprechen.)

Plausibilitätsbetrachtung:

Seien  $f$  und  $g_i$  differenzierbar

$\Rightarrow \exists$  Gradient  $\nabla f(x^*)$  (Vektor der partiellen Ableitungen) von  $f$  im Punkt  $x^*$  mit

1. Der Gradient steht senkrecht auf der Tangentialhyperebene in  $x^*$  an  $\{x | f(x) = f(x^*)\}$  und
2. zeigt in die Richtung des stärksten Anstieges der Funktionswerte  $f(x)$ , d.h.  $f(x)$  wächst in Richtung des Gradienten.

$\Rightarrow$  Für  $\leq$ -Restriktionen zeigt  $\nabla g_i(x^*)$  in einem Punkt  $x^*$  mit  $g_i(x^*) = 0$  aus  $Z$  hinaus, wobei  $Z = \{x \in \mathbb{R}^n | g_i(x) \leq 0, \text{ für alle } i\}$ . Man nennt eine Restriktion  $g_i$  in Punkt  $x$  *aktiv*, falls  $g_i(x) = 0$ .

Lokale Optimalität eines Punktes  $x^*$  bedeutet:

- $x^*$  ist zulässig,
- $x^*$  muss in einer gewissen Umgebung der beste Punkt sein  $\Rightarrow x^* \in \text{Rand}(Z)$   
 $\Rightarrow$  Jeder verbessernde Richtung muss aus  $Z$  hinauszeigen (d.h. die Randpunkte sind von Interesse)

„Verbessernde Richtungen“

1. Richtung des steilsten Abstiegs in  $x^*$  ist  $-\nabla f(x^*)$ . Jede verbessernde Richtung bildet mit  $-\nabla f(x^*)$  einen spitzen Winkel ( $< 90^\circ$ ).
2. Liegt  $-\nabla f(x^*)$  in dem Kegel  $K(x^*)$ , der durch die Gradienten der aktiven (in  $x^*$ ) Restriktion aufgespannt wird, dann zeigt jede verbessernde Richtung aus  $Z$  hinaus.
3. Sei  $I = I(x^*)$  die Menge der Indizes der in  $x^*$  aktiven Restriktionen  
 $\Rightarrow K(x^*) = \{d \in \mathbb{R}^n | d = \sum_{i \in I} u_i \cdot \nabla g_i(x^*), u_i \geq 0\}$  (Menge aller Linearkombinationen der aktiven Restriktionen (ist auch konvex))



4.  $-\nabla f(x^*)$  liegt genau dann in  $K(x^*)$ , wenn gilt:  
 $\forall i \in I, \exists u_i \geq 0$  mit  $-\nabla f(x^*) = \sum_{i \in I} u_i \cdot \nabla g_i(x^*)$  (negativer Gradient fällt in den Kegel)

Daraus leiten sich die sogenannten *Kuhn-Tucker-Bedingungen* ab

$$(KTB) \left\{ \begin{array}{l} \nabla f(x^*) + \sum_{i \in I} u_i \cdot \nabla g_i(x^*) = 0 \\ u_i \geq 0 \quad \forall i \in I \\ g_i(x^*) \leq 0 \quad i = 1, \dots, m \end{array} \right.$$

**Satz 15 (Hinreichende KTB)**

Seien  $f : \mathbb{R}^n \rightarrow \mathbb{R}$  und  $g_i : \mathbb{R}^n \rightarrow \mathbb{R}$ ,  $i = 1, \dots, m$  stetig differenzierbar und konvexe Funktionen und sei das NLP:

$$\begin{array}{ll} \min & f(x) \\ \text{s.d.} & g_i(x) \leq 0, \quad i = 1, \dots, m \end{array}$$

Dann gilt:

Gibt es in einem Punkt  $x^* \in \mathbb{R}^n$  für alle aktiven Restriktionen  $i \in I$  nichtnegative Skalare  $u_i \geq 0$ , sodass die KTB erfüllt sind, dann ist  $x^*$  globales Minimum des NLP.

Notwendige Kuhn-Tucker-Bedingung:

Konvexität wird nicht verlangt, aber Regularitätsannahme (lineare Unabhängigkeit der Gradienten der in  $x^*$  aktiven Restriktionen)

$\Rightarrow$  Zu jedem lokalen Minimum kann man eine Lösung der KTB finden.

**Satz 16 (Notwendige KTB)**

Seien  $f : \mathbb{R}^n \rightarrow \mathbb{R}$  und  $g_i : \mathbb{R}^n \rightarrow \mathbb{R}$ ,  $i = 1, \dots, m$  stetig differenzierbare Funktionen,  $x^* \in \mathbb{R}^n$ ,  $I = \{i \in \mathbb{N} | g_i(x^*) = 0\}$  und  $\nabla g_i(x^*)$  linear unabhängig  $\forall i \in I$ , dann gilt:

Ist  $x^*$  ein lokales Minimum des NLP, dann gibt es  $u_i \geq 0 \forall i \in I$ , sodass die KTB erfüllt sind.

Die KTB bilden ein Analogon zu dem aus der Analysis bekannten Resultat, dass man die Minimalstellen einer konvexen Funktion durch Nullsetzen der partiellen Ableitungen erhält.

Kritik: Sucht man (konstruktiv) nach Punkten, die die KTB erfüllen sollen, weiß man nicht, ob bzw. welche Restriktionen in derartigen Punkten aktiv sind.  $\Rightarrow$  modifizierte Form der KTB:

$$(KTB') \left\{ \begin{array}{l} \nabla f(x^*) + \sum_{i=1}^m u_i \cdot \nabla g_i(x^*) = 0 \\ u_i \cdot g_i(x^*) = 0 \\ u_i \geq 0 \\ g_i(x^*) \leq 0 \end{array} \right\} \quad i = 1, \dots, m$$

Zusätzliche Nebenbedingungen, damit diejenigen Restriktionen, die nicht aktiv sind wieder rausfliegen ( $\Rightarrow u_i = 0$  für alle nicht aktiven Restriktionen).

### 15.3 Kuhn-Tucker-Theorie Zugang über Lagrangefunktion

Es sei das Optimierungsproblem

$$\begin{aligned} \min \quad & f(x) \\ \text{s.d.} \quad & g_i(x) \leq 0 \quad i = 1, \dots, m \\ & x \geq 0 \end{aligned} \quad (19)$$

zugrunde gelegt. Für jede Nebenbedingung  $g_i(x) \leq 0$  wird eine *Lagrange-Multiplikator*  $u_i$  eingeführt.

#### Definition 23

Die Funktion  $\mathcal{L} : \mathbb{R}^{n+m} \rightarrow \mathbb{R}$

mit  $\mathcal{L}(x, u) = f(x) + \sum_{i=1}^m u_i g_i(x) = f(x) + u^T g(x)$  heißt *Lagrange-Funktion* zu Modell (19), wobei  $u = (u_1, \dots, u_m)^T$  und  $g = (g_1, \dots, g_m)^T$ .

#### Definition 24

Ein Vektor  $(x_0, u_0) \in \mathbb{R}^{n+m}$  mit  $x_0, u_0 \geq 0$  heißt *Sattelpunkt* von  $\mathcal{L}(x, u)$ , wenn  $\forall x \in \mathbb{R}^n, x \geq 0$  und  $\forall u \in \mathbb{R}^m, u \geq 0$  gilt:  $\mathcal{L}(x_0, u) \leq \mathcal{L}(x_0, u_0) \leq \mathcal{L}(x, u_0)$  (Minimum bzgl.  $x$ , Maximum bzgl.  $u$ )

#### Satz 17 (Notwendige KTB)

Seien  $f(x)$  und  $g_i(x)$  für  $i = 1, \dots, m$  partiell stetig differenzierbar.  $L_x, L_u$  bezeichnen die Gradienten von  $\mathcal{L}$  bzgl.  $x$  bzw.  $u$ , d.h.

$$L_x = \nabla f(x) + \sum_{i=1}^m u_i \cdot \nabla g_i(x) \quad \text{und} \quad L_u = g(x) = (g_1(x), \dots, g_m(x))^T.$$

Notwendige Bedingung für einen Sattelpunkt von  $\mathcal{L}$  in  $(x_0, u_0)^T$ :

$$\left. \begin{array}{l} L_x(x_0, u_0) \geq 0 \quad x_0^T \cdot L_x(x_0, u_0) = 0 \\ L_u(x_0, u_0) \leq 0 \quad u_0^T \cdot L_u(x_0, u_0) = 0 \\ u_0 \geq 0, \quad x_0 \geq 0 \end{array} \right\} \text{Orthogonalität: Gradient steht} \\ \text{senkrecht auf dem Sattelpunkt}$$

Dadurch, dass die Restriktionen mit in die Zielfunktion hineingezogen werden, was zur Lagrange-Funktion führt, liefert dann das Nullsetzen der partiellen Ableitung der Lagrange-Funktion die notwendigen Bedingungen für die Extremwerte.

#### Satz 18

Ist  $(x_0, u_0)^T, x_0, u_0 \geq 0$  ein Sattelpunkt von  $\mathcal{L}$ , so ist  $x_0$  eine optimale Lösung von Modell (19).

Bemerkung: Die Konvexität von  $\mathcal{L}(x, u)$  in  $x$  und die Konkavität in  $u$  wurden bisher nicht vorausgesetzt. Setzt man dies zusätzlich voraus, so sind die KTB notwendig und hinreichend für einen Sattelpunkt von  $\mathcal{L}$ .

#### Definition 25

In Modell (19) sei  $I = \{1, \dots, m\}$  derart disjunkt in  $I_1 \cup I_2$  aufgespalten, dass  $I_2$  die Indexmenge der linearen Restriktionen bezeichnet. Falls ein  $x \in \mathbb{R}^n$  derart existiert, dass  $\forall t \in I_1$  gilt  $g_t(x) < 0$ , so erfüllt  $x$  die sogenannte *Slater-Bedingung*. ( $x$  ist innerer Punkt bzgl. der nichtlinearen Restriktionen.)

**Satz 19**

Erfüllt das Modell (19) (einschließlich Konvexitätsvoraussetzungen) die Slater-Bedingung, so sind die KTB notwendig und hinreichend für eine optimale Lösung von Modell (19).

Nützlichkeit der KTB

- Überprüfung einer Kandidatenlösung auf Optimalität
- Grundlage für Theorie (z.B. Dualität der NLP)
- In Spezialfällen zur konstruktiven Bestimmung optimaler Lösungen verwendbar (z.B. Quadratisches Programmieren)

**15.4 Quadratisches Programmieren**

Bei quadratischen konvexen Optimierungsproblemen ist die Zielfunktion quadratisch und der zulässige Bereich ein konvexes Polyeder des  $\mathbb{R}^n$ . Das Optimierungsproblem lässt sich folgendermaßen formulieren:

$$\begin{aligned} \min \quad & f(x) = c^T x + \frac{1}{2} x^T Q x \\ \text{s.d.} \quad & Ax \leq b \\ & x \geq 0 \\ & c, x \in \mathbb{R}^n, A = A_{m,n} \end{aligned} \quad (20)$$

Wie im Fall der LOP ist das kovexe Polyeder der  $\mathbb{R}^n$  in der Form  $Ax \leq b$ ,  $x \geq 0$  gegeben.  $Q$  ist symmetrisch und positiv semidefinit ( $\Rightarrow f(x)$  ist konvex).

Damit hat die Lagrange-Funktion die folgende Gestalt:

$$\mathcal{L}(x, u) = c^T x + \frac{1}{2} x^T Q x + u^T (Ax - b)$$

und  $\mathcal{L}$  ist konvex in  $x$  und konkav in  $u$ .

$\Rightarrow$  KTB sind notwendig und hinreichend für die optimale Lösung, die Slater-Bedingung ist nicht relevant (da nur lineare Restriktionen), sie lauten:

$$\left. \begin{aligned} L_x(x, u) = c + Qx + A^T u &\geq 0 \\ L_u(x, u) = Ax - b &\leq 0 \end{aligned} \right\} (\alpha)$$

$$L_x \cdot x_0 = 0, \quad L_u \cdot u_0 = 0 \quad (\beta)$$

Hinzufügen von Schlupfvariablen zu  $(\alpha) \Rightarrow$

$$\left. \begin{aligned} c + Qx + A^T u - s_1 &= 0 \\ -b + Ax + s_2 &= 0 \end{aligned} \right\} (\gamma)$$

Mit dem Algorithmus von Wolfe lässt sich eine zulässige Lösung von  $(\gamma)$  mit Hilfe der Simplexmethode (Abschnitt 2) unter Beachtung von  $(\beta)$  finden.

**Algorithmus von Wolfe**

1. Formuliere die KTB für Modell (20).
2. Füge linearen Bedingungen Schlupfvariablen hinzu.  
Falls eine zulässige Basislösung unter Beachtung von  $(\beta)$  vorliegt, ist die Lösung optimal, sonst

3. Füge, wo nötig, Hilfsvariablen zur Erlangung einer Ausgangsbasislösung ( $\gamma$ ) und ( $\beta$ ) hinzu. Iteriere mit  $M$ -Methode bis zulässige Lösung erreicht ist oder feststeht, dass der Lösungsraum leer ist. Dabei ist ( $\beta$ ) durch eine beschränkten Basiseintritt zu beachten: Ist  $x_j$  in Lösung, darf  $s_{1j}$  nicht aufgenommen werden und umgekehrt, damit die Orthogonalität erhalten bleibt.

## 15.5 Verfahren der NLP

### 15.5.1 Minimierung ohne Restriktionen

Prinzipielles Vorgehen

Problem:  $\min f(x)$

gegeben:  $x^k \in \mathbb{R}^n$ , d.h. ein zulässiger Iterationspunkt

gesucht:  $x^{k+1} \in \mathbb{R}^n$  mit:  $f(x^{k+1}) < f(x^k)$

Die Iteration besteht meist aus zwei Schritten:

1. Bestimme eine Richtung  $d^k \in \mathbb{R}^n$  (Richtungsvektor der  $k$ -ten Iteration), in der  $f$  fällt (ZF-Wert verbessert).
2. Löse  $\min\{f(x^k + \mu d^k) | \mu > 0\}$   
Dabei handelt es sich um eine eindimensionale Optimierung (Strahloptimierung). Die Funktion wird nur noch auf diesem verbessernden Strahl betrachtet.  
Setze:  $x^{k+1} = x^k + \mu^k d^k$ ;  $\mu^k$  ist hier die optimale Schrittweite.

### 15.5.2 Mehrdimensionales Suche (unrestringiert) — Differenzierbarkeit vorausgesetzt

#### a) Steilster Abstieg: Gradientenverfahren

Wie bereits in Abschnitt 15.2 erwähnt, zeigt  $\nabla f(x)$  in Richtung des stärksten Wachstums von  $f(x)$ .

$\Rightarrow$  Richtung des steilsten Abstieges im Punkt  $x^k$  ist  $-\nabla f(x^k)$

*Gradientenverfahren*

- gegeben:  
 $x^k$   $k$ -ter Iterationspunkt  
 $d^k = -\nabla f(x^k)$  immer in Richtung des steilsten Abstiegs
- bestimme:  
 $\mu^k = \arg \min\{f(x^k + \mu d^k) | \mu \geq 0\}$   
 $x^{k+1} = x^k + \mu^k d^k$
- Abbruchbedingung:  $|\nabla f(x^k)| < \varepsilon$

Für die erzeugte Punktfolge gilt:  $f(x^{k+1}) < f(x^k)$

(Häufig schlechte Konvergenz, in der Nähe des Optimums Zick-Zack-Kurs.)

Wesentlich schneller als das klassische Gradientenverfahren konvergiert das Newton-Verfahren.

#### b) Mehrdimensionales Newton-Verfahren

Sei jetzt  $f : \mathbb{R}^n \rightarrow \mathbb{R}$  zweimal stetig differenzierbar.

$\Rightarrow$  Quadratische Näherung von  $f(x)$  in  $x^k$  (mehrdimensionale Taylorentwicklung)

$$q(x) = f(x^k) + \nabla f(x^k)^T(x - x^k) + \frac{1}{2}(x - x^k)^T H(x^k)(x - x^k)$$

mit der Hesseschen Matrix  $H$  ( $n \times n$ -Matrix der zweiten partiellen Ableitungen).

Es wird nun wie im eindimensionalen Fall vorgegangen:

$$\nabla q(x) = 0 \Leftrightarrow \nabla f(x^k) + H(x^k)(x - x^k) = 0$$

Ist  $H(x)$  invertierbar, so gilt:  $x^{k+1} = x^k - H^{-1}(x^k)\nabla f(x^k)$

Idee des Newton-Verfahrens:

Das quadratische Problem  $f(x) = \frac{1}{2}x^T Ax - b^T x + c$  ist in einem Iterationsschritt lösbar, wenn man  $x^* = x^0 - A^{-1}(Ax^0 - b) = A^{-1}b$  für ein beliebiges  $x^0 \in \mathbb{R}^n$  setzt ( $A$  ist dabei positiv definit). Man muss dann lediglich  $Ax = b$  lösen.

Analogieüberlegung zum quadratischen Problem:

$$\left. \begin{array}{l} \nabla f(x^0) = Ax^0 - b \\ H(x^0) = \nabla^2 f(x^0) = A^{-1} \\ \nabla f_k = \nabla f(x_k) \end{array} \right\} \text{allgemein } x^{k+1} = x^k - H^{-1}(x^k)\nabla f(x^k)$$

Der Vergleich mit der allgemeinen Abstiegsformel  $x^{k+1} = x^k + \mu^k d^k$  ergibt:

$d^k = -H^{-1}(x^0)\nabla f(x^0)$  und  $\mu^k = 1$  (keine explizite Schrittweitenbestimmung in Optimumsnähe).

### Satz 20

Sei  $x^*$  optimal für das unrestringierte Optimierungsproblem:

$$\nabla f(x^*) = 0$$

$$\nabla^2 f(x^*) = H(x^*)$$

Falls  $x^0$  hinreichende nahe bei  $x^*$  liegt, konvergiert das Newton-Verfahren

$$x^{k-1} = x^k - H^{-1}(x^k)\nabla f(x^k)$$

quadratisch gegen  $x^*$ , d.h. es gilt:  $\|x^{k+1} - x^*\| < \gamma \|x^k - x^*\|^2$ ,  $0 < \gamma < 1$

Vorteil: Quadratische Konvergenz

Nachteil:

- In jedem Iterationsschritt muss die Hessesche Matrix neu berechnet und
- das lineare Gleichungssystem  $H(x^k)d^k = -\nabla f(x^k)$  gelöst werden, um die Richtung zu bestimmen.
- $H(x^k)$  kann singulär oder schlecht konditioniert sein.
- Das Verfahren ist nur lokal konvergent ( $x^0$  muss hinreichend nahe bei  $x^*$  liegen)

→ Besser: Quasi-Newton-Verfahren

### 15.5.3 Verfahren zulässiger Richtungen (restringiert)

- Explizite Berücksichtigung der Nebenbedingungen im Verlaufe des Optimierungsverfahrens.

- Prinzipielles Vorgehen:

Problem:  $\min_{x \in Z} f(x)$ ,  $x^k \in Z \subset \mathbb{R}^n$  sei gegeben.

$Z$  wie zuvor, also  $Z = \{x \in \mathbb{R}^n | g_i(x) \leq 0, i = 1, \dots, m\}$

Ziel: Verbesserung der zulässigen Lösung.

Vorgehen: Bestimme Vektor  $d^k \in \mathbb{R}^n$  mit

1.  $x^k + \mu d^k \in Z \forall \mu \in \mathbb{R}$  mit  $0 \leq \mu \leq \varepsilon$
2.  $f(x^k + \mu d^k) < f(x^k) \forall \mu \in \mathbb{R}$  mit  $0 \leq \mu \leq \varepsilon$   
 $d^k$  heißt *verbessernde zulässige Richtung*, denn
  - bewegt man sich von  $x^k$  in Richtung  $d^k$  verbessert sich der ZF-Wert (wird kleiner) und
  - geht man nicht zu weit, bleibt man zulässig.
3. Bestimmung einer optimalen Schrittweite  $\mu \in [0, \varepsilon]$
4. Festlegung des nächsten Iterationspunktes  $x^{k+1} = x^k + \mu d^k$

Zur Bestimmung von  $d^k$  existieren unterschiedliche Methoden, hier soll das Verfahren von Zoutendijk benutzt werden.

### Satz 21

Sei  $x^k \in Z, I = \{i \in \{1, \dots, m\} | g_i(x^k) = 0\}$  (aktive Restriktionen).

Wenn  $d \in \mathbb{R}^n$  die Ungleichungen  $\nabla f(x^k)^T d < 0$  und  $\nabla g_i(x^k)^T d < 0$  für alle  $i \in I$  erfüllt, dann ist  $d$  eine verbessernde zulässige Richtung. Das Skalarprodukt der Vektoren  $d$  und  $\nabla f(x^k)$  bzw.  $d$  und  $\nabla g_i(x^k)$  bilden einen stumpfen Winkel. Schreitet man von  $x^k$  in Richtung  $d$  nimmt der ZF-Wert ab.

Für gegebenes  $x^k$  lässt sich ein derartige Richtung  $d$  durch Lösung eines LPs mit den Variablen  $d_j$  und  $z$  bestimmen.

$$\begin{array}{ll} \text{LP} & \min \quad z \\ & \text{s.d.} \quad \nabla f(x^k)^T d - z \leq 0 \\ & \quad \quad \nabla g_i(x^k)^T d - z \leq 0 \quad \text{für } i \in I \\ & \quad \quad -1 \leq d_j \leq 1 \quad \quad j = 1, \dots, n \end{array}$$

Dabei schränken die Bedingungen  $-1 \leq d_j \leq 1$  die Länge des Lösungsvektors  $d$  ein und verhindern somit unbeschränkte Lösungen.

Es gilt:

Sei  $(\bar{z}, \bar{d})$  eine Lösung von  $LP(\bar{x}^k)$ .

Falls  $\bar{z} < 0 \Rightarrow \bar{d}$  ist eine verbessernde zulässige Richtung.

Falls  $\bar{z} = 0 \Rightarrow \bar{x}^k$  ist ein Kuhn-Tucker-Punkt.

### Algorithmus von Zoutendijk

1. Wähle eine Startpunkt  $x^0 \in Z$  und  $0 < \varepsilon < 1$ .
2. Setze  $x^{-1} := x^0 + (1, \dots, 1)^T$ ,  $k := 0$
3. Bestimme eine Lösung von  $LP(x^k)$ , d.h.  $(d^k, z^k)$ .

### 15.5.4 Minimierung mit Nebenbedingungen

(siehe auch Verfahren von Wolfe)

Strafkosten und Barriere-Verfahren

Das zu lösende restringierte Optimierungsproblem wird durch unrestringierte Probleme approximiert, indem man entweder zur Zielfunktion einen „Strafterm“ addiert, der das Verletzen der Restriktion mit hohen Kosten bestraft oder auf dem Rand des zulässigen Bereiches eine „Barriere“ errichtet, die das Verlassen verhindert.

- Penalty (Strafkosten)-Verfahren:  
 $\{x^k\}_{k=0,1,\dots}$  Folge unzulässiger Punkte (Näherungen an den Lösungsraum „von außen“)
- Barriere-Verfahren:  
 $\{x^k\}_{k=0,1,\dots}$  Folge zulässiger Punkte ( $\Rightarrow x^k \in Z$  muss bekannt sein).

#### Penalty-Verfahren

$$\begin{array}{ll} \text{NLP} & \min_{x \in Z} f(x) \\ & \text{mit } Z = \{x \in \mathbb{R}^n \mid g_i(x) \leq 0, i = 1, \dots, m\} \end{array}$$

Strafkostenfunktion:  $S(x) = \sum_{i=1}^m (\max\{0, g_i(x)\})^p$  mit  $1 \leq p \leq N$  (meist  $p = 2$ )

Es gilt:  $S(x) \geq 0$  und  $x \in Z$  (d.h. zulässig)  $\Leftrightarrow S(x) = 0$ .

Je stärker ein Punkt Restriktionen verletzt, desto größer ist der Wert der Strafkostenfunktion.

Axiomatik:

Sei  $Z$  wie oben definiert,  $S_r(x) : \mathbb{R}^n \rightarrow \mathbb{R}$  mit  $S_r(x) \begin{cases} = 0, & \text{für } x \in Z, \forall r \geq 0 \\ > 0, & \text{für } x \notin Z, \forall r \geq 0 \end{cases}$  und  $\mu_r > 0$

für alle  $r \geq 0$ . Man definiert:

Eine Folge von Funktionen  $\{\mu_r, S_r(x)\}_{r=0,1,\dots}$  heißt eine Folge von Strafkostenfunktionen für den Bereich  $Z$ , falls für aller  $r = 0, 1, \dots$  gilt:

1.  $\mu_r S_r(x)$  ist stetig in  $\mathbb{R}^n$
2.  $\mu_r S_r(x) = 0 \forall x \in Z$  (zulässige Werte nicht bestrafen)
3.  $\mu_{r+1} S_{r+1}(x) > \mu_r S_r(x) > 0 \forall r$  und  $\forall x \notin Z$  (monoton wachsend)
4.  $\mu_r S_r(x) \rightarrow \infty$  für  $r \rightarrow \infty$  und  $z \notin Z$  (Strafe wird beliebig groß)

Verwendung in einem Ansatz: Die Minimierung von  $f$  wird nun ersetzt durch die Minimierung von  $f(x) + \mu S(x)$  (freies, unrestringiertes Problem)

Lösung mit entsprechender Methode der vorangegangenen Abschnitte 15.5.1 und 15.5.2. Parameter  $\mu > 0$

- $\mu$  klein: geringes Gewicht der Restriktionen, Problem ist  $\min_{x \in \mathbb{R}^n} f(x)$  ähnlich.
- $\mu$  groß: Restriktionen dominieren, eine Verletzung wird stark bestraft.

Vorgehen: Mit kleinem  $\mu$  beginnen, dann  $\mu$  vergrößern, um sich  $Z$  von außen zu nähern.  $\mu$  also schrittweise erhöhen, was zum Lösen der Problemfolge  $\{\min_{x \in \mathbb{R}^n} (f(x) + \mu_r S_r(x))\}_{r=0,1,\dots}$  führt.

Algorithmus

1. Wähle  $\varepsilon > 0$  (Toleranz),  $\mu_0 > 0$  (etwa  $\mu_0 = 1$ ),  $\beta > 1$  und  $x^0 \in \mathbb{R}^n$ .  
Setze  $r = 0$

2. WHILE  $\mu_r S(x^r) > \varepsilon$  DO

$$x^{r+1} := \arg \min (f(x) + \mu_r S(x))$$

$$\mu_r := \beta \mu_r$$

$$r := r + 1$$

## Literatur

- [1] H.-J. Sebastian. Vorlesung OR I, Produktionsplanung und Projektmanagement. RWTH Aachen, 2002.
- [2] H.-J. Sebastian. Vorlesung OR II, Transportlogistik und Scheduling. RWTH Aachen, 2003.
- [3] Klaus Neumann and Martin Morlock. *Operations Research*. Hanser, 2 edition, Oktober 2002.