apl. Prof. Dr. Frank Werner

Fakultät für Mathematik

Institut für Mathematische Optimierung

https://math.uni-magdeburg.de/∼werner/

Operations Research für MBA

Vorlesungsskript
(auszugsweise)

Wintersemester 2019/20

Bemerkungen:

1. Dieses Skript ersetzt nicht den Besuch der Vorlesung. Zahlenbeispiele werden mehrheitlich nur in der Vorlesung präsentiert.
2. Das Symbol ✏ weist auf zusätzliche, detaillierte Ausführungen in der Vorlesung hin.

LITERATUR

1. Werner, F.; Sotskov, Y.N.: Mathematics of Economics and Business, 1st Edition, Routledge, Abingdon (UK) / New York (USA), 2006, 536 p.

   (**KOSTENLOSER** Download des Ebooks z.B. bei Amazon.de Kindle Store, siehe

   https://www.amazon.de/Mathematics-Economics-Business-English-Werner-ebook/dp/B000Q7ZFKW/)

2. Neumann, K.; Morlock, M.: Operations Research, Carl Hanser Verlag, München, Wien, 1993.

3. Hillier, F; Lieberman, G: Introduction to Operations Research, 9th Edition, MacGraw Hill, New York (USA), 2010.

4. Taha, H.A.: Operations Research - An Introduction, 7th Edition, Prentice Hall, New York (USA), 2003.

5. **zur Auffrischung mathematischer Grundlagen:**
   Werner, F.: A Refresher Course in Mathematics, Bookboon Publishers, 2016, 284 S.

   (**KOSTENLOSER** Download der pdf-Datei unter:

   https://bookboon.com/en/a-refresher-course-in-mathematics-ebook)

Inhaltsverzeichnis

Kapitel 1

Lineare Optimierung

1.1 Einführendes Beispiel

Beispiel 1: Futtermix

Ein Unternehmen produziert einen Futtermix bestehend aus drei Zutaten bezeichnet als R_1, R_2 und R_3. Die Zutaten R_1 und R_2 müssen mit einem Mindestprozentsatz enthalten sein, und Zutat R_3 darf einen Maximalprozentsatz nicht überschreiten. Die Preise sind bekannt, und die Daten sind wie in Tabelle 1:

Tabelle 1: Daten für Beispiel 1

Zutat	erforderlicher Prozentsatz	Preis in EUR per Kilogramm
R_1	mindestens 10 Prozent	25
R_2	mindestens 50 Prozent	17
R_3	höchstens 30 Prozent	12

Gesucht wird eine zulässige Zusammenstellung mit minimalen Kosten. Seien $x_i, i \in \{1, 2, 3\}$, die Prozentsätze von Zutat R_i. Man erhält die folgenden Bedingungen:

$$x_1 + x_2 + x_3 = 100. \tag{1.1}$$

Gleichung (1.1) besagt, dass die Summe aller Prozentsätze gleich 100 ist. Da vom Zutat R_3 nicht mehr als 30 % enthalten sein dürfen, erhält man die Nebenbedingung

$$x_3 \leq 30. \tag{1.2}$$

Der Prozentsatz von R_2 soll mindestens 50 % sein, oder was dasselbe ist, die Summe der Prozentsätze von R_1 und R_3 ist höchstens 50 %:

$$x_1 + x_3 \leq 50. \tag{1.3}$$

Der Prozentsatz von R_1 ist mindestens 10 %, oder gleichbedeutend, die Summe der Prozentsätze von R_2 und R_3 darf 90 Prozernt nicht überschreiten, d.h.

$$x_2 + x_3 \leq 90. \tag{1.4}$$

Alle Variablen müssen nichtnegativ sein:

$$x_1 \geq 0, \quad x_2 \geq 0, \quad x_3 \geq 0. \tag{1.5}$$

Die Kosten für die Herstellung des Produkts sollen minimal sein, d.h. die Zielfunktion ist wie folgt:

$$z = 25x_1 + 17x_2 + 12x_3 \longrightarrow \text{min!} \tag{1.6}$$

$z \to$ min! gibt an, dass der Zielfunktionswert z minimiert werden soll. Wir erhalten ein Problem mit der Zielfunktion (1.6), vier Nebenbedingungen (drei Ungleichungen (1.2),(1.3) und (1.4) und eine Gleichung (1.1)) sowie die Nichtnegativitätsbedingungen (1.5) für alle drei Variablen.

1.2 Grundlagen

Ein lineares Optimierungsproblem (abgekürzt LOP) besteht aus Nebenbedingungen (ein System von linearen Gleichungen oder Ungleichungen), Nichtnegativitätsbedingungen und einer linearen Zielfunktion. Die allgemeine Form eines solchen LOP lautet wie folgt:

Allgemeine Form eines LOP:

$$z = c_1x_1 + c_2x_2 + \cdots + c_nx_n \longrightarrow \begin{cases} \text{max!} \\ \text{min!} \end{cases}$$

unter den Nebenbedingungen (u.d.N.)

$$\begin{array}{c} a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n \ \{\leq, =, \geq\} \ b_1 \\ a_{21}x_1 + a_{22}x_2 + \cdots + a_{2n}x_n \ \{\leq, =, \geq\} \ b_2 \\ \cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots \\ a_{m1}x_1 + a_{m2}x_2 + \cdots + a_{mn}x_n \ \{\leq, =, \geq\} \ b_m \end{array}$$

$$x_j \geq 0, j \in J \subseteq \{1, 2, \ldots, n\}$$

Alternativ kann man die folgende *Matrixdarstellung* eines LOP verwenden:

$$z = \mathbf{c}^T\mathbf{x} \longrightarrow \begin{cases} \text{max!} \\ \text{min!} \end{cases}$$

u.d.N.

$$A\mathbf{x}\{\leq, =, \geq\}\mathbf{b} \tag{1.7}$$

$$\mathbf{x} \geq \mathbf{0}.$$

Die Matrix A ist vom Format $m \times n$. $\mathbf{c} = (c_1, c_2, \ldots, c_n)^T$ ist der Vektor der Koeffizienten der Zielfunktion, und $\mathbf{b} = (b_1, b_2, \ldots, b_m)^T$ bezeichnet den Vektor der rechten Seite.

Geometrische Interpretation eines LOP mit zwei Variablen x_1 und x_2

Angenommen die Nebenbedingungen sind als Ungleichungen gegeben. Die Ungleichungen

$$a_{i1}x_1 + a_{i2}x_2 \, R_i \, b_i, \ R_i \in \{\leq, \geq\}, \ i = 1, 2, \ldots m,$$

sind Halbebenen begrenzt durch die Geraden

$$a_{i1}x_1 + a_{i2}x_2 = b_i.$$

Jede dieser Geraden kann in der Form

$$\frac{x_1}{s_1} + \frac{x_2}{s_2} = 1,$$

geschrieben werden, wobei $s_1 = b_i/a_{i1}$ und $s_2 = b_i/a_{i2}$ die Schnittpunkte der Geraden mit den Koordinatenachsen x_1 und x_2 sind.

Für konstantes z und $c_2 \neq 0$ ist die Zielfunktion

$$z = c_1x_1 + c_2x_2$$

eine Gerade der Form

$$x_2 = -\frac{c_1}{c_2}x_1 + \frac{z}{c_2},$$

d.h., für verschiedene Werte von z erhält man parallele Geraden mit dem Anstieg $-c_1/c_2$. Der Vektor $\mathbf{c} = (c_1, c_2)^T$ gibt die Richtung an, in die der Zielfunktionswert am stärksten steigt. Also: Im Falle eiens Maximierungsproblems mit der Zielfunktion z verschiebt man die Gerade

$$x_2 = -\frac{c_1}{c_2}x_1 + \frac{z}{c_2}$$

in die Richtung des Vektors $\mathbf{c}$, und bei der Minimierung von z wird die Gerade in the entgegengesetzte Richtung beschrieben durch den Vektor $-\mathbf{c}$ geschoben.

Ein LOP der Form (1.7) mit zwei Variablen kann *grafisch* wie folgt gelöst werden:

(1) Ermittle den zulässigen Bereich M (d.h. die Menge der zulässigen Lösungen als Durchschnitt aller zulässigen Halbebenen mit dem ersten Quadranten).

(2) Zeichne die Zielfunktion $z = Z$, wobei Z konstant ist und verschiebe sie parallel entweder in Richtung des Vektors $\mathbf{c}$ (im Fall $z \to$ max!) oder in die Richtung des Vektors $-\mathbf{c}$ (im Fall $z \to$ min!). Wende dieses Vorgehen an solange die Gerade $z = \mathit{const}$ gemeinsame Punkte mit dem zulässigen Bereich hat.

Definition 1: Eine zulässige Lösung $\mathbf{x} = (x_1, x_2, \ldots, x_n)^T$, für die die Zielfunktion den optimalen (d.h. maximalen oder minimalen) Wert annimmt, heißt **optimale Lösung**.

1.3 Eigenschaften eines LOP

Definition 2: Eine Menge M heißt **konvex**, falls für je zwei beliebige Vektoren $\mathbf{x^1}, \mathbf{x^2} \in M$ jede Konvexkombination

$$\lambda\mathbf{x^1} + (1-\lambda)\mathbf{x^2}$$

mit $0 \leq \lambda \leq 1$ auch zur Menge M gehört.

Definition 3: Ein Vektor (Punkt) $\mathbf{x} \in M$ heißt **Eckpunkt** der konvexen Menge M, falls $\mathbf{x}$ nicht als echte Konvexkombination von zwei anderen Vektoren aus M geschrieben werden kann, d.h. $\mathbf{x}$ kann nicht geschrieben werden als

$$\lambda \mathbf{x^1} + (1-\lambda)\mathbf{x^2}$$

mit $\mathbf{x^1}, \mathbf{x^2} \in M$ und $0 < \lambda < 1$.

Theorem 1:
Die Menge M aller zulässigen Lösungen des Systems (1.7) ist entweder leer oder eine konvexe Menge mit einer endlichen Anzahl von Eckpunkten.

Theorem 2:
Falls die Menge M des Systems (1.7) beschränkt ist, kann sie als Menge aller Konvexkombinationen der Eckpunkte $\mathbf{x^1}, \mathbf{x^2}, \ldots, \mathbf{x^s}$ von M geschrieben werden, d.h.:

$$\begin{aligned} M \quad = \quad & \Big\{ \mathbf{x} \in \mathbb{R}^n \mid \mathbf{x} = \lambda_1 \mathbf{x^1} + \lambda_2 \mathbf{x^2} + \cdots + \lambda_s \mathbf{x^s}; \\ & 0 \le \lambda_i \le 1, \quad i = 1, 2, \ldots, s, \quad \sum_{i=1}^{s} \lambda_i = 1 \Big\} \end{aligned}$$

Sei M die Menge der zulässigen Lösungen und es werde die Maximierung der Funktion $z = \mathbf{c}^T \mathbf{x}$ betrachtet.

Dann gibt es die folgenden drei Fälle:

(a) Es gilt $M = \emptyset$. Dann widersprechen sich die Nebenbedingungen, d.h. es existiert keine zulässige Lösung des LOP.

(b) M ist eine nichtleere beschränkte Menge vom $\mathbb{R}^n$.

(c) M ist eine unbeschränkte Menge des $\mathbb{R}^n$, d.h. mindestens eine Variable kann beliebig groß werden, oder, falls die Variablen auch negativ sein können, kann eine von ihnen beliebig klein werden.

Im Fall (c) gibt es zwei Möglichkeiten:

(c1) Die Zielfunktion z ist von oben beschränkt. Dann existiert eine optimale Lösung des Maximierungsproblems.

(c2) Die Zielfunktion z ist nicht beschränkt von oben. Dann existiert keine (endliche) Optimallösung des betrachteten Maximierungsproblems.

Theorem 3:
Falls ein LOP eine (endliche) Optimallösung besitzt, dann existiert mindestens ein optimaler Eckpunkt, in dem die Zielfunktion den optimalen Wert annimmt.

Theorem 4:
Seien $P_1, P_2, \ldots, P_r$ beschrieben durch die Vektoren $\mathbf{x^1}, \mathbf{x^2}, \ldots, \mathbf{x^r}$ optimale Eckpunkte. Dann ist jede Konvexkombination

$$\mathbf{x^0} = \lambda_1\mathbf{x^1} + \lambda_2\mathbf{x^2} + \ldots + \lambda_r\mathbf{x^r}, \quad \lambda_i \geq 0,\ i = 1, 2, \ldots, r, \quad \sum_{i=1}^{r} \lambda_i = 1$$

auch optimal.

1.4 Standardform eines LOP

Sei $r(A)$ der Rang der Matrix A, d.h. die maximale Anzahl linear unabhängiger Spaltenvektoren (oder äquivalent, Zeilenvektoren) der Matrix A.

Definition 4: Ein System $A\mathbf{x} = \mathbf{b}$ von $p = r(A)$ linearen Gleichungen, wobei in jeder Gleichung eine Variable nur in dieser Gleichung auftritt und den Koeffizienten $+1$ hat, heißt lineares Gleichungssystem in **kanonischer Form**. Die eliminierten Variablen heißen **Basisvariable** (BV), während die anderen Variablen **Nichtbasisvariable** (NBV) genannt werden.

Folglich ist die Anzahl der Basisvariablen gleich dem Rang der Matrix A. Falls $r(A) = p < n$, kann das System in der Form

$$I\mathbf{x^B} + A_N\mathbf{x^N} = \mathbf{b},$$

geschrieben werden, wobei $\mathbf{x^B}$ der p-dimensionale Vektor der Basisvariablen ist, $\mathbf{x^N}$ ist der $(n-p)$-dimensionale Vektor der Nichtbasisvariablen, und A_N is eine Teilmatrix von A bestehend aus den Spaltenvektoren, die zu den NBV gehören.

Definition 5: Eine Lösung $\mathbf{x}$ des Gleichungssystems $A\mathbf{x} = \mathbf{b}$ in kanonischer Form, bei der alle Basisvariablen den Wert Null haben, heißt **Basislösung**.

Definition 6: Ein LOP der Form

$$z = \mathbf{c}^T\mathbf{x} \longrightarrow \max!$$

$$\text{u.d.N.} \qquad A\mathbf{x} = \mathbf{b}, \quad \mathbf{x} \geq \mathbf{0},$$

mit $A = (A_N, I)$ und $\mathbf{b} \geq \mathbf{0}$ heißt **Standardform** des LOP.

Die Standardform eines LOP ist durch die folgenden Eigenschaften gekennzeichnet:

- das LOP ist ein Maximierungsproblem;
- die Nebenbedingungen sind als Gleichungssystem in kanonischer Form mit nichtnegativen rechten Seiten gegeben und
- alle Variablen sind nichtnegativ.

Jedes LOP kann in die Standardform transformiert werden mittels der folgenden Regeln. Es werden alle möglichen Verletzungen der Standardform gemäß Definition 6 betrachtet.

(a) Eine Variable x_j ist nicht notwendig nichtnegativ, d.h. x_j kann beliebige Werte annehmen. Dann wird die Variable x_j durch die Differenz zweier nichtnegativer Variablen ersetzt, d.h. wir setzen:

$$x_j = x_j^* - x_j^{**} \qquad \text{with} \quad x_j^* \geq 0 \quad \text{und} \quad x_j^{**} \geq 0.$$

Dann ergibt sich:

$$\begin{aligned} x_j^* > x_j^{**} &\iff x_j > 0 \\ x_j^* = x_j^{**} &\iff x_j = 0 \\ x_j^* < x_j^{**} &\iff x_j < 0. \end{aligned}$$

(b) Die Zielfunktion ist zu minimieren:

$$z = c_1x_1 + c_2x_2 + \cdots + c_nx_n \to \min!$$

Die Minimierung der Funktion z ist äquivalent zur Maximierung der Funktion $\bar{z} = -z$:

$$z = c_1x_1 + c_2x_2 + \cdots + c_nx_n \to \min! \iff \bar{z} = -z = -c_1x_1 - c_2x_2 - \cdots - c_nx_n \to \max!$$

(c) Für eine rechte Seite gilt $b_i < 0$:

$$a_{i1}x_1 + a_{i2}x_2 + \cdots + a_{in}x_n = b_i < 0.$$

Multiplikation dieser Nebenbedingung mit -1 ergibt:

$$-a_{i1}x_1 - a_{i2}x_2 - \cdots - a_{in}x_n = -b_i > 0.$$

(d) Angenommen, einige Nebenbedingungen sind Ungleichungen:

$$a_{i1}x_1 + a_{i2}x_2 + \cdots + a_{in}x_n \leq b_i$$

oder

$$a_{k1}x_1 + a_{k2}x_2 + \cdots + a_{kn}x_n \geq b_k.$$

Dann führt man eine Schlupfvariable u_i bzw. eine Überschussvariable u_k ein, und man erhält eine Gleichung:

$$\begin{aligned} & a_{i1}x_1 + a_{i2}x_2 + \cdots + a_{in}x_n + u_i = b_i \qquad \text{mit} \qquad u_i \geq 0 \\ \text{oder} \quad & \\ & a_{k1}x_1 + a_{k2}x_2 + \cdots + a_{kn}x_n - u_k = b_k \qquad \text{mit} \qquad u_k \geq 0\,. \end{aligned}$$

(e) Angenommen, das gegebene System ist nicht in kanonischer Form, d.h. die Nebenbedingungen sind z.B. wie folgt gegeben:

$$a_{11}x_1 + a_{12}x_2 + \ldots + a_{1n}x_n = b_1$$

$$a_{21}x_1 + a_{22}x_2 + \ldots + a_{2n}x_n = b_2$$

$$\ldots$$
$$a_{m1}x_1 + a_{m2}x_2 + \ldots + a_{mn}x_n = b_m$$

mit $\quad b_i \geq 0,\ i = 1,2,\ldots,m;\ x_j \geq 0,\ j = 1,2,\ldots,n.$

In der obigen Form enthält keine Nebenbedingung eine eliminierte Variable. Man führt in jeder Nebenbedingung eine künstliche Variable x_{Ai} als Basisvariable ein und erhält:

$$\begin{array}{l}
a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n + x_{A1} \qquad\qquad\qquad = b_1 \\
a_{21}x_1 + a_{22}x_2 + \cdots + a_{2n}x_n \qquad + x_{A2} \qquad\quad = b_2 \\
\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \\
a_{m1}x_1 + a_{m2}x_2 + \cdots + a_{mn}x_n \qquad\qquad\quad + x_{Am} = b_m
\end{array}$$

mit $\quad b_i \geq 0,\ i = 1,2,\ldots,m;\ x_j \geq 0,\ j = 1,2,\ldots,n,$ und $x_{Ai} \geq 0, i = 1,2,\ldots,m.$

Beispiel 2:

Gegeben ist das folgende LOP:

$$z = -x_1 + 3x_2 + x_4 \to \min!$$

$$\begin{array}{lrcrcrcrcr}
\text{u.d.N.} & x_1 & - & x_2 & + & 3x_3 & - & x_4 & \geq & 8 \\
& & & x_2 & - & 5x_3 & + & 2x_4 & \leq & -4 \\
& & & & & x_3 & + & x_4 & \leq & 3
\end{array}$$
$$x_2, x_3, x_4 \geq 0.$$

Die Variable x_1 wird substituiert durch die Differenz zweier nichtnegativer Variablen x_1^* und x_1^{**}, d.h. $x_1 = x_1^* - x_1^{**}$ mit $x_1^* \geq 0, x_2^{**} \geq 0$. Des Weiteren multiplizieren wir die Zielfunktion z mit -1 und erhalten:

$$\overline{z} = -z = x_1^* - x_1^{**} - 3x_2 - x_4 \to \max!$$

$$\begin{array}{lrcrcrcrcrcr}
\text{u.d.N.} & x_1^* & - & x_1^{**} & - & x_2 & + & 3x_3 & - & x_4 & \geq & 8 \\
& & & & & x_2 & - & 5x_3 & + & 2x_4 & \leq & -4 \\
& & & & & & & x_3 & + & x_4 & \leq & 3
\end{array}$$
$$x_1^*, x_1^{**}, x_2, x_3, x_4 \geq 0.$$

Durch Multiplikation der zweiten Nebenbedingung mit -1 und Einführung einer Schlupfvariablen x_7 in der dritten Nebenbedingung sowie der Überschussvariablen x_5 und x_6 in der zweiten und dritten Nebenbedingung erhält man alle Nebenbedingungen als Gleichungen mit nichtnegativen rechten Seiten:

$$\overline{z} = -z = x_1^* - x_2^{**} - 3x_2 - x_4 \to \max!$$

$$\begin{array}{lrcrcrcrcrcrcrcrcr}
\text{u.d.N.} & x_1^* & - & x_1^{**} & - & x_2 & + & 3x_3 & - & x_4 & - & x_5 & & & & & = & 8 \\
& & & & - & x_2 & + & 5x_3 & - & 2x_4 & & & - & x_6 & & & = & 4 \\
& & & & & & & x_3 & + & x_4 & & & & & + & x_7 & = & 3
\end{array}$$
$$x_1^*, x_1^{**}, x_2, x_3, x_4, x_5, x_6, x_7 \geq 0.$$

Jetzt wählt man die Variable x_1^* als eliminierte Variable in der ersten Nebenbedingung und die Variable x_7 als eliminierte Variable in der dritten Nebenbedingung, aber es gibt keine eliminierte Variable in der zweiten Nebenbedingung mit dem Koeffizienten $+1$. Daher wird die künstliche

Variable x_{A1} in der zweiten Nebenbedingung eingeführt, und man erhält:

$$\overline{z} = -z = x_1^* - x_2^{**} - 3x_2 - x_4 \to \max!$$

$$\begin{array}{lrrrrrrrrrrl}
\text{u.d.N.} & -x_1^{**} & -\ x_2 & +\ 3x_3 & -\ x_4 & -\ x_5 & & +\ \mathbf{x_1^*} & & & = & 8 \\
& & -\ x_2 & +\ 5x_3 & -\ 2x_4 & & -\ x_6 & & +\ \mathbf{x_{A1}} & & = & 4 \\
& & & x_3 & +\ x_4 & & & & & +\ \mathbf{x_7} & = & 3
\end{array}$$

$$x_1^*, x_1^{**}, x_2, x_3, x_4, x_5, x_6, x_7, x_{A1} \geq 0.$$

Das Problem hat jetzt $n = 9$ Variablen in der Standardform.

1.5 Der Simplex-Algorithmus

Grundidee:

- Von einem Eckpunkt ausgehend (repräsentiert durch eine zulässige Baislösung gemäß einer Standardform des LOP), wird der Zielfunktionswert berechnet und geprüft, ob der Übergang zu einem benachbartem Eckpunkt (mittels eines Pivotschrittes) zu einer Verbesserung führt.

- Falls ja, führe diesen Schritt zum benachbarten Eckpunkt aus und überprüfe, ob weitere Verbesserungen mittels neuem Pivotschritt möglich sind. Sobald ein Eckpunkt erreicht wurde, von dem kein weiterer Pivotschritt zu einer Verbeserung führt, hat man eine Optimallösung gefunden.

Zur Anwendung dieses Vorgehens benötigt man ein Kriterium um zu entscheiden, ob ein Übergang zu einem benachbarten Eckpunkt den Zielfunktionswert verbessert. Angenommen, der Rang der Matrix A ist gleich $m : r(A) = m$, d.h., in der kanonischen Form gibt es m BV unter den n Variablen, und die Anzahl der NBV ist folglich gleich $n' = n - m$. Wir betrachten die *zulässige kanonische Form* mit den BV x_{Bi} und den NBV x_{Nj}:

$$x_{Bi} = \hat{b}_i - \sum_{j=1}^{n'} \hat{a}_{ij} x_{Nj}\ , \quad i = 1, 2, \ldots, m \quad (n' = n - m). \tag{1.8}$$

Dann kann die Zielfunktion z wie folgt geschrieben werden:

$$\begin{aligned}
z &= c_1x_1 + c_2x_2 + \ldots + c_nx_n \\
&= \underbrace{c_{B1}x_{B1} + c_{B2}x_{B2} + \ldots + c_{Bm}x_{Bm}}_{(BV)} + \underbrace{c_{N1}x_{N1} + c_{N2}x_{N2} + \ldots + c_{Nn'}x_{Nn'}}_{(NBV)} \\
&= \textstyle\sum_{i=1}^{m} c_{Bi}x_{Bi} \ + \ \sum_{j=1}^{n'} c_{Nj}x_{Nj}.
\end{aligned}$$

Mit den Gleichungen (1.8) kann man die Basisvariablen ersetzen und die Zielfunktion nur in

Abhängigkeit der Nichtbasisvariablen schreiben. Man erhält

$$\begin{aligned} z &= \sum_{i=1}^{m} c_{Bi} \left(\hat{b}_i - \sum_{j=1}^{n'} \hat{a}_{ij} x_{Nj} \right) + \sum_{j=1}^{n'} c_{Nj} x_{Nj} \\ &= \sum_{i=1}^{m} c_{Bi} \hat{b}_i - \sum_{j=1}^{n'} \left(\sum_{i=1}^{m} c_{Bi} \hat{a}_{ij} - c_{Nj} \right) x_{Nj}. \end{aligned}$$

Die letzte Zeile wird *Zielfunktionszeile* genannt. Außerdem definiert man:

$$z_0 = \sum_{i=1}^{m} c_{Bi} \hat{b}_i \quad \text{(Zielfunktionswert der Basislösung);} \tag{1.9}$$

$$g_j = \sum_{i=1}^{m} c_{Bi} \hat{a}_{ij} - c_{Nj} \quad \text{(Koeffizient der NBV } x_{Nj} \text{ in der Zielfunktionszeile).} \tag{1.10}$$

Für die Berechnung von z_0 gemäß (1.9) sei erinnert, dass in einer Basislösung alle Nichtbasisvariablen gleich Null sind.

Damit erhält man folgende Darstellung der Zielfunktion in Abhängigkeit von den Nichtbasisvariablen x_{Nj}:

$$z = z_0 - g_1 x_{N1} - g_2 x_{N2} - \ldots - g_{n'} x_{Nn'}.$$

Der Koeffizient g_j gibt die Veränderung des Zielfunktionswertes an, wenn die NBV x_{Nj} Basisvariable wird und ihr Wert um eine Einheit steigt. Jetzt ergibt sich das folgende Optimalitätskriterium.

Theorem 5: (Simplex-Kriterium)
Falls $g_j \geq 0$, $j = 1, 2, \ldots, n'$, für alle Koeffizienten der Nichtbasisvariablen in der Zielfunktionszeile gilt, ist die zugehörige Lösung optimal.

Folgerung: Falls eine Spalte l mit $g_l < 0$ in einer zulässigen Basislösung existiert, kann der Zielfunktionswert vergrößert werden durch Aufnahme dieses Spaltenvektors in die Menge der Basisvektoren und x_{Nl} wird Basisvariable im nächsten Schritt.

Ausgehend von dem Starttableau wenden wir die Kurzform des Simplextableaus an. Man verwendet eine zusätzliche Zeile für die Koeffizienten g_j zusammen mit dem Zielfunktionswert z_0.

	NBV	x_{N1}	x_{N2}	$\cdots$	$\mathbf{x_{Nl}}$	$\cdots$	$x_{Nn'}$		
BV	-1	c_{N1}	c_{N2}	$\cdots$	c_{Nl}	$\cdots$	$c_{Nn'}$	0	Q
x_{B1}	c_{B1}	$\hat{a}_{11}$	$\hat{a}_{12}$	$\cdots$	$\hat{a}_{1l}$	$\cdots$	$\hat{a}_{1n'}$	$\hat{b}_1$	
x_{B2}	c_{B2}	$\hat{a}_{21}$	$\hat{a}_{22}$	$\cdots$	$\hat{a}_{2l}$	$\cdots$	$\hat{a}_{2n'}$	$\hat{b}_2$	
$\vdots$	$\vdots$	$\vdots$	$\vdots$		$\vdots$		$\vdots$	$\vdots$	
$\mathbf{x_{Bk}}$	c_{Bk}	$\hat{a}_{k1}$	$\hat{a}_{k2}$	$\cdots$	$\mathbf{\hat{a}_{kl}}$	$\cdots$	$\hat{a}_{kn'}$	$\hat{b}_k$	
$\vdots$	$\vdots$	$\vdots$	$\vdots$		$\vdots$		$\vdots$	$\vdots$	
x_{Bm}	c_{Bm}	$\hat{a}_{m1}$	$\hat{a}_{m2}$	$\cdots$	$\hat{a}_{ml}$	$\cdots$	$\hat{a}_{mn'}$	$\hat{b}_m$	
z		g_1	g_2	$\cdots$	g_l	$\cdots$	$g_{n'}$	z_0	

Bestimmung der Pivotspalte l

Wähle eine Spalte $l, 1 \leq l \leq n'$, mit $g_l < 0$. Oft wird eine Spalte l gewählt mit

$$g_l = \min\{g_j \mid g_j < 0, \; j = 1, 2, \ldots, n'\}.$$

Bestimmung der Pivotzeile k

Nach einem Pivotschritt muss die resultierende Basislösung zulässig sein. Daher wählt man die Zeile k mit $1 \leq k \leq m$ und

$$\frac{\hat{b}_k}{\hat{a}_{kl}} = \min\left\{\frac{\hat{b}_i}{\hat{a}_{il}} \mid \hat{a}_{il} > 0, \; i = 1, 2, \ldots, m\right\}.$$

Zur Berechnung der obigen Quotienten wird eine Spalte Q am Ende des obigen Tableaus hinzugefügt, in die der Quotient eingetragen wird in jeder Zeile mit einem positiven Element in der gewählten Pivotspalte.

Wird Spalte l als Pivotspalte gewählt, wird die entsprechende Variable x_{Nl} zur Basisvariablen im nächsten Schritt. Mit Zeile k als Pivotzeile wird die entsprechende Variable x_{Bk} Nichtbasisvariable im nächsten Schritt. Das Element $\hat{a}_{kl}$ heißt *Pivotelement.*

Die beiden nachfolgenden Theoreme charakterisieren die Fälle wenn eine optimale Lösung nicht existiert bzw. wenn sie nicht eindeutig ist.

Theorem 6:
Falls $g_l < 0$ für einen Koeffizienten einer Nichtbasisvariablen in der Zielfunktionszeile sowie $\hat{a}_{il} \leq 0$ für alle Koeffizienten in der Spalte l gilt, dann hat das LOP keine (endliche) Optimallösung.

Theorem 7:
Falls ein Koeffizient $g_l = 0$ in der Zielfunktionszeile einer optimalen Lösung existiert mit $\hat{a}_{il} > 0$ für mindestens einen Koeffizienten in Spalte l, dann existiert eine andere optimale Basislösung, in der x_{Nl} eine Basisvariable ist.

Simplex-Algorithmus

Schritt 1: Transformiere das LOP in die Standardform mit den Nebenbedingungen in kanonischer Form wie folgt (wir nehmen an, dass **keine** künstlichen Variablen für die Transformation notwendig sind):

$$A_N\mathbf{x_N} + I\mathbf{x_B} = \mathbf{b}, \qquad \mathbf{x_N} \geq \mathbf{0}, \qquad \mathbf{x_B} \geq \mathbf{0}, \qquad \mathbf{b} \geq \mathbf{0}.$$

Die Ausgangsbasislösung ist

$$\mathbf{x} = (\mathbf{x_N}, \mathbf{x_B})^T = (\mathbf{0}, \mathbf{b})^T$$

mit dem Zielfunktionswert

$$z_0 = \mathbf{c}^T \mathbf{x}.$$

Stelle das entsprechende Anfangstableau auf.

Schritt 2: Betrachte die Koeffizienten $g_j,\ j = 1, 2, \ldots, n'$, der Nichtbasisvariablen x_{Nj} in der Zielfunktionszeile.

Falls $g_j \geq 0$ für $j = 1, 2, \ldots, n'$, dann ist die gegenwärtige Basislösung optimal, STOP. Andernfalls gibt es einen Koeffizienten $g_j < 0$ in der Zielfunktionszeile.

Schritt 3: Bestimme eine Spalte l mit

$$g_l = \min\{g_j \mid g_j < 0,\ j = 1, 2, \ldots, n'\}$$

als Pivotspalte.

Schritt 4: If $a_{il} \leq 0$ für $i = 1, 2, \ldots, m$, STOP (in diesem Fall existiert keine Optimallösung für das Problem). Andernfalls gibt es mindestens ein $a_{il} > 0$.

Schritt 5: Bestimme die Pivotzeile k so, dass

$$\frac{b_k}{a_{kl}} = \min\left\{\frac{b_i}{a_{il}} \mid a_{il} > 0,\ i = 1, 2, \ldots, m\right\}.$$

Schritt 6: Tausche die Basisvariable x_{Bk} in Zeile k mit der Nichtbasisvariablen x_{Nl} in Spalte l aus und berechne die folgenden Werte des neuen Tableaus:

$$\begin{aligned}
\hat{a}_{kl} &= \frac{1}{a_{kl}}; \\
\hat{a}_{kj} &= \frac{a_{kj}}{a_{kl}}; \qquad \hat{b}_k = \frac{b_k}{a_{kl}}; \qquad j = 1, 2, \ldots, n',\ j \neq l; \\
\hat{a}_{il} &= -\frac{a_{il}}{a_{kl}}; \qquad i = 1, 2, \ldots, m,\ i \neq k; \\
\hat{a}_{ij} &= a_{ij} - \frac{a_{il}}{a_{kl}} \cdot a_{kj}; \qquad \hat{b}_i = b_i - \frac{a_{il}}{a_{kl}} \cdot b_k; \\
& \qquad\qquad i = 1, 2, \ldots, m,\ i \neq k; \quad j = 1, 2, \ldots, n',\ j \neq l.
\end{aligned}$$

Außerdem erhält man die folgenden Werte in der letzten Zeile des Tableaus:

$$\begin{aligned}
\hat{g}_l &= -\frac{g_l}{a_{kl}}; \\
\hat{g}_j &= g_j - \frac{g_l}{a_{kl}} \cdot a_{kj}; \qquad j = 1, 2, \ldots, n',\ j \neq l; \\
\hat{z}_0 &= z_0 - \frac{g_l}{a_{kl}} \cdot b_k.
\end{aligned}$$

Betrachte das erhaltene Tableau als neue Ausgangslösung und gehe zu Schritt 2.

Beispiel 3:

Ein Unternehmen beabsichtigt drei Typen von Produktion P_1, P_2 und P_3 herzustellen, so dass die gesamten Produktionskosten 32 000 EUR nicht überschreiten. 420 Arbeitsstunden stehen zur Verfügung, und 30 Einheiten des Rohmaterials sind vorhanden. Außerdem sind die Daten gemäß Tabelle 2 gegeben.

Tabelle 2: Daten für das Beispiel 3

Produkt	P_1	P_2	P_3
Verkaufspreis (EUR/Stück)	1600	3000	5200
Produktionskosten (EUR/Stück)	1000	2000	4000
Erforderliches Rohmaterial per Stück	3	2	2
Arbeitszeit in Stunden per Stück	20	10	20

Das Ziel besteht in der Bestimmung der herzustellenden Mengen der Produkte, so dass der Gewinn maximal wird. Sei x_i die Anzahl der produzierten Stücke von $P_i,\ i \in \{1,2,3\}$. Dann ergibt sich folgendes LOP:

$$
\begin{array}{lrcrcrcl}
 & \multicolumn{7}{l}{z = 6x_1 + 10x_2 + 12x_3 \to \max!} \\
\text{u.d.N.} & x_1 & + & 2x_2 & + & 4x_3 & \leq & 32 \\
 & 3x_1 & + & 2x_2 & + & 2x_3 & \leq & 30 \\
 & 2x_1 & + & x_2 & + & 2x_3 & \leq & 42 \\
 & \multicolumn{7}{l}{x_1, x_2, x_3 \geq 0.}
\end{array}
$$

Die Zielfunktion erhält man durch Subtraktion der Produktionskosten von den Erlösen und Division des resultierenden Gewinns durch 100. Außerdem ist die Nebenbedingung für die Produktionskosten durch 1000 dividiert wurden und die Nebenbedingung bzgl. der Arbeitszeit durch 10.

Nach Einführung der Schlupfvariablen $x_{3+i} \geq 0$ in der i-ten Nebenbedingung erhält man das folgende Starttableau:

	NBV	x_1	x_2	$\mathbf{x_3}$		
BV	-1	6	10	12	0	Q
$\mathbf{x_4}$	0	1	2	**4**	32	8
x_5	0	3	2	2	30	15
x_6	0	2	1	2	42	21
		-6	-10	-12	0	

Wählt man x_3 als neue Basisvariable (da sie den kleinsten negativen Koeffizienten in der Zielfunktionszeile besitzt) wird die Variable x_4 neue NBV gemäß Quotientenregel. Man erhält:

	NBV	x_1	$\mathbf{x_2}$	x_4		
BV	-1	6	10	0	0	Q
x_3	12	$\frac{1}{4}$	$\frac{1}{2}$	$\frac{1}{4}$	8	16
$\mathbf{x_5}$	0	$\frac{5}{2}$	**1**	$-\frac{1}{2}$	14	14
x_6	0	$\frac{3}{2}$	0	$-\frac{1}{2}$	26	$-$
		-3	-4	3	96	

Jetzt werden die Variablen x_2 und x_5 ausgetauscht. Es ergibt sich:

	NBV	x_1	x_5	x_4		
BV	-1	6	0	0	0	Q
x_3	12	-1	$-\frac{1}{2}$	$\frac{1}{2}$	1	
x_2	10	$\frac{5}{2}$	1	$-\frac{1}{2}$	14	
x_6	0	$\frac{3}{2}$	0	$-\frac{1}{2}$	26	
		7	4	1	152	

Da alle Koeffizienten g_j positiv sind, ergibt sich die folgende optimale Lösung:

$$x_1 = 0, \quad x_2 = 14, \quad x_3 = 1, \quad x_4 = 0, \quad x_5 = 0, \quad x_6 = 26.$$

Also: Es ist optimal, kein Stück von P_1 herzustellen, 14 Stücke von P_2 sowie ein Stück von P_3. Berücksichtigt man, dass die Koeffizienten der Zielfunktion durch 100 geteilt wurden, erhält man den optimalen Gewinn von 15200 EUR.

Beispiel 4:

Betrachtet wird das folgende LOP:

$$\begin{array}{rrcrcr} & \multicolumn{5}{c}{z = -2x_1 - 2x_2 \to \min!} \\ \text{u.d.N.} & x_1 & - & x_2 & \geq & -1 \\ & -x_1 & + & 2x_2 & \leq & 4 \\ & \multicolumn{5}{l}{x_1, x_2 \geq 0.} \end{array}$$

Man transformiert das gegebene Problem in die Standardform, d.h. man multipliziert die erste Nebenbedingung mit -1 und führt die Schlupfvariablen x_3 und x_4 ein. Man erhält:

$$\begin{array}{rrcrcrcrcr} & \multicolumn{9}{c}{\overline{z} = 2x_1 + 2x_2 \to \max!} \\ \text{u.d.N.} & -x_1 & + & x_2 & + & x_3 & & & = & 1 \\ & -x_1 & + & 2x_2 & & & + & x_4 & = & 4 \\ & \multicolumn{9}{l}{x_1, x_2, x_3, x_4 \geq 0.} \end{array}$$

Jetzt ergibt sich das folgende Starttableau:

	NBV	x_1	$\mathbf{x_2}$		
BV	-1	2	2	0	Q
$\mathbf{x_3}$	0	-1	**1**	1	1
x_4	0	-1	2	4	2
		-2	-2	0	

Im ersten Pivotschritt werden die Variablen x_2 und x_3 ausgetauscht (man kann auch die Spalte von x_1 als Pivotspalte wählen). Es ergibt sich das folgende Tableau:

	NBV	$\mathbf{x_1}$	x_3		
BV	-1	2	0	0	Q
x_2	2	-1	1	1	
$\mathbf{x_4}$	0	**1**	-2	2	2
		-4	2	2	

Im letzten Tableau ist nur ein negativer Koeffizient einer NBV in der Zielfunktionszeile, daher erfolgt ein Austausch der Variablen x_1 und x_4. Man erhält:

	NBV	x_4	x_3		
BV	-1	0	0	0	Q
x_2	2	1	-1	3	
x_1	2	1	-2	2	
		4	-6	10	

Jetzt kann nur die Variable x_3 für einen Austausch ausgewählt werden. Jedoch sind in dieser Spalte alle Einträge negativ, d.h. man kann keinen weiteren Pivotschritt durchführen. Folglich hat das Problem keine endliche Optimallösung für das betrachtete Maximierungsproblem (siehe Theorem 6). (Hinweis: Man erkennt das bereits im Starttableau).

Beispiel 5:

Gegeben ist das folgende LOP:

$$
\begin{array}{lrcrcrcrcrcrcl}
 & \multicolumn{13}{c}{z = x_1 + x_2 + x_3 + x_4 + x_5 + x_6 \to \min!} \\
\text{u.d.N.} & 2x_1 & + & x_2 & + & x_3 & & & & & & & \geq & 4000 \\
 & & & x_2 & & & + & 2x_4 & + & x_5 & & & \geq & 5000 \\
 & & & & & x_3 & & & + & 2x_5 & + & 3x_6 & \geq & 3000 \\
 & \multicolumn{13}{c}{x_1, x_2, x_3, x_4, x_5, x_6 \geq 0.}
\end{array}
$$

Um die Standardform zu erzeugen, ist zu bemerken, dass in jeder Nebenbedingung eine Variable existiert, die nur in einer Bedingung auftritt. Variable x_1 tritt nur in der ersten Nebenbedingung auf, Variable x_4 nur in der zweiten und Variable x_6 nur in der dritten Nebenbedingung. Daher kann man die erste Nebenbedingung durch den Koeffizienten 2 von x_1 dividieren, die zweite ebenfalls durch 2 und die dritte durch 3.

Man führt eine Überschussvariable in jeder Nebenbedingung ein, multipliziert die Zielfunktion mit -1 und erhält die erforderliche Einheitsmatrx (nach Umordnung der Variablen).

$$
\begin{array}{lrcrcrcrcrcrcrcrcl}
 & \multicolumn{17}{c}{\overline{z} = -z = -x_1 - x_2 - x_3 - x_4 - x_5 - x_6 \to \max!} \\
\text{u.d.N.} & \frac{1}{2}x_2 & + & \frac{1}{2}x_3 & & & - & x_7 & & & & & + & x_1 & & & = & 2000 \\
 & \frac{1}{2}x_2 & & & + & \frac{1}{2}x_5 & & & - & x_8 & & & & & + & x_4 & = & 2500 \\
 & & & \frac{1}{3}x_3 & + & \frac{2}{3}x_5 & & & & & - & x_9 & & & + & x_6 & = & 1000 \\
 & \multicolumn{17}{c}{x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8, x_9 \geq 0.}
\end{array}
$$

Man erhält folgendes Starttableau:

	NBV	x_2	x_3	$\mathbf{x_5}$	x_7	x_8	x_9		
BV	-1	-1	-1	-1	0	0	0	0	Q
x_1	-1	$\frac{1}{2}$	$\frac{1}{2}$	0	-1	0	0	2000	$-$
x_4	-1	$\frac{1}{2}$	0	$\frac{1}{2}$	0	-1	0	2500	5000
$\mathbf{x_6}$	-1	0	$\frac{1}{3}$	$\mathbf{\frac{2}{3}}$	0	0	-1	1000	1500
		0	$\frac{1}{6}$	$-\frac{1}{6}$	1	1	1	-5500	

Jetzt tauscht man die Variablen x_5 und x_6 aus. Es ergibt sich das folgende Tableau:

	NBV	x_2	x_3	x_6	x_7	x_8	x_9		
BV	-1	-1	-1	-1	0	0	0	0	Q
x_1	-1	$\frac{1}{2}$	$\frac{1}{2}$	0	-1	0	0	2000	
x_4	-1	$\frac{1}{2}$	$-\frac{1}{4}$	$-\frac{3}{4}$	0	-1	$\frac{3}{4}$	1750	
x_5	-1	0	$\frac{1}{2}$	$\frac{3}{2}$	0	0	$-\frac{3}{2}$	1500	
		0	$\frac{1}{4}$	$\frac{1}{4}$	1	1	$\frac{3}{4}$	-5250	

Aus dem letzten Tableau erhält man die folgende Optimallösung:

$$x_1 = 2000, \quad x_2 = x_3 = 0, \quad x_4 = 1750, \quad x_5 = 1500, \quad x_6 = 0$$

mit dem optimalen Zielfunktionswert $\overline{z}_0^{max} = -5250$, was $z_0^{min} = 5250$ entspricht (für das Ausgangs-Minimierungsproblem). Man bemerke, dass die optimale Lsüng nicht eindeutig bestimmt ist. Im letzten Tableau gibt es in der Zielfunktionszeile einen Koeffizienten mit dem Wert Null, d.h. man könnte noch die Variablen x_2 und x_4 austauschen. Dies ergibt die folgende zulässige Basislösung mit dem gleichen Zielfunktionswert:

$$x_1 = 250, \quad x_2 = 3500, \quad x_3 = x_4 = 0, \quad x_5 = 1500, \quad x_6 = 0.$$

1.6 Der 2-Phasen Simplexalgorithmus

Die Einführung künstlicher Variablen ist erforderlich, falls mindestens eine Nebenbedingung als Gleichung ohne eliminierte Variable vorliegt. Z.B. können die Nebenbedingungen folgende Gestalt haben:

$$\begin{array}{l}
a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n = b_1 \\
a_{21}x_1 + a_{22}x_2 + \cdots + a_{2n}x_n = b_2 \\
\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots \\
a_{m1}x_1 + a_{m2}x_2 + \cdots + a_{mn}x_n = b_m \\
x_j \geq 0,\ i = 1, 2, \ldots, n; \quad b_i \geq 0,\ i = 1, 2, \ldots, m.
\end{array}$$

Wie im Fall (e) in Abschnitt 1.4 diskutiert, führt man eine künstliche Variable x_{Ai} in jeder Gleichung ein. Zusätzlich ersetzt man die Ausgangs-Zielfunktion z durch eine Hilfszielfunktion z_I, die die Summe aller künstlichen Variablen minimiert (bzw. äquivalent die negative Summe aller künstlichen Variablen maximiert), denn das Ziel ist es, dass alle künstlichen Variablen den Wert Null annehmen, um die Zulässigkeit der Lösung zu garantieren. Das ergibt folgendes LOP für Phase I:

$$z_I = -x_{A1} - x_{A2} - \cdots - x_{Am} \longrightarrow \max!$$

$$\begin{array}{lllll}
\text{u.d.N.} & a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n + x_{A1} & & & = b_1 \\
& a_{21}x_1 + a_{22}x_2 + \cdots + a_{2n}x_n & + x_{A2} & & = b_2 \\
\cdots\cdots & \cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots & \cdots\cdots & \cdots\cdots & \cdots \\
& a_{m1}x_1 + a_{m2}x_2 + \cdots + a_{mn}x_n & & + x_{Am} & = b_m
\end{array} \tag{1.11}$$

$$x_j \geq 0,\ j = 1, 2, \ldots, n \quad \text{und} \quad x_{Ai} \geq 0,\ i = 1, 2, \ldots, m.$$

Angenommen man hat eine optimale Lösung des Hilfsproblems (1.11) mit der Simplexmethode gefunden, d.h. das Verfahren bricht ab mit $g_j \geq 0$ für alle Koeffizienten der Nichtbasisvariablen in der Zielfunktionszeile (bezüglich der Hilfsfunktion z_I).

Am Ende von Phase I sind die folgenden zwei (Haupt)fälle möglich:

(1) $z_I^{\max} < 0 \iff$ Das Ausgangsproblem hat keine zulässige Lösung.

(2) Es gilt $z_I^{\max} = 0$ und alle künstlichen Variablen sind Nichtbasisvariable. Dann repräsentiert diese Basislösung eine zulässige kanonische Form für das Ausgangsproblem, und man kann mit Phase II des Simplex-Algorithm gemäß Abschnitt 1.5 beginnen.

Beispiel 6:

Gegeben ist das folgende LOP:

$$\begin{array}{rrcrcl} & z = x_1 - 2x_2 \to \max! & & & & \\ \text{u.d.N.} & x_1 & + & x_2 & \leq & 4 \\ & 2x_1 & - & x_2 & \geq & 1 \\ & x_1, x_2 \geq 0. & & & & \end{array}$$

Wir transformieren das gegebene Problem in die Standardform durch Einführung einer Überschussvariablen (x_3) in der zweiten Nebenbedingung, eienr Schlupfvariablen (x_4) in der ersten Nebenbedingung und einer künstlichen Variablen (x_{A1}) in der zweiten Nebenbedingung. Jetzt ersetzen wir die Zielfunktion z durch die Hilfsfunktion z_I. Daher betrachten wir in Phase I das folgende LOP:

$$\begin{array}{lrcrcrcrcrcl} & & & z_I = -x_{A1} \to \max! & & & & & & & & \\ \text{u.d.N.} & x_1 & + & x_2 & & & + & x_4 & & & = & 4 \\ & 2x_1 & - & x_2 & - & x_3 & & & + & x_{A1} & = & 1 \\ & & & x_1, x_2, x_3, x_4, x_{A1} \geq 0. & & & & & & & & \end{array}$$

Das Starttableau ist wie folgt:

	NBV	$\mathbf{x_1}$	x_2	x_3		
BV	-1	0	0	0	0	Q
x_4	0	1	1	0	4	4
$\mathbf{x_{A1}}$	-1	$\mathbf{2}$	-1	-1	1	$\frac{1}{2}$
		-2	1	1	-1	

Nach Austausch der Variablen x_1 und x_{A1} ergibt sich:

	NBV	x_{A1}	x_2	x_3		
BV	-1	-1	0	0	0	Q
x_4	0	$-\frac{1}{2}$	$\frac{3}{2}$	$\frac{1}{2}$	$\frac{7}{2}$	
x_1	0	$\frac{1}{2}$	$-\frac{1}{2}$	$-\frac{1}{2}$	$\frac{1}{2}$	$-$
		1	0	0	0	

Phase I ist beendet, es wird die Variable x_{A1} mit der entsprechenden Spalte entfernt, und man bestimmt die neuen Koeffizienten g_j der Zielfunktionszeile. Das ergibt:

	NBV	x_2	$\mathbf{x_3}$		
BV	-1	-2	0	0	Q
$\mathbf{x_4}$	0	$\frac{3}{2}$	$\mathbf{\frac{1}{2}}$	$\frac{7}{2}$	7
x_1	1	$-\frac{1}{2}$	$-\frac{1}{2}$	$\frac{1}{2}$	$-$
		$\frac{3}{2}$	$-\frac{1}{2}$	$\frac{1}{2}$	

Jetzt werden die Variablen x_3 und x_4 ausgetauscht. Man erhält das folgende Tableau:

	NBV	x_2	x_4		
BV	-1	-2	0	0	Q
x_3	0	3	2	7	
x_1	1	1	1	4	
		3	1	4	

Da alle Koeffizienten g_j nichtnegativ sind, ist die erhaltene Lösung optimal:

$$x_1 = 4, x_2 = 0.$$

Der optimale Zielfunktionswert ist $z_0^{max} = 4$.

Betrachtet man nun ein anderes LOP mit den gleichen Nebenbedingungen, bei dem die Zielfunktion wie folgt ist:

$$\tilde{z} = -x_1 + 3x_2 \longrightarrow \min!$$

Kann man einfach entscheiden, ob die optimale Lösung für die vorhergehende Zielfunktion auch optimal für die neue Zielfunktion ist? Man braucht nur die Koeffizienten c_1 und c_2 der Zielfunktion im letzten Tableau zu ersetzen (wiederum für die Maximierungsversion), berechnet neu die Koeffizienten g_j der Zielfunktionszeile und erhält das folgende Tableau:

	NBV	x_2	x_4		
BV	-1	-3	0	0	Q
x_3	0	3	2	7	
x_1	1	1	1	4	
		4	1	4	

Da alle Koeffitzienten g_j in der Zielfunktionszeile nichtnegativ sind, ist die Lösung

$$x_1 = 4, x_2 = 0$$

auch optimal für $\tilde{z} = -x_1 + 3x_2 \longrightarrow \min$ mit dem Funktionswert $\tilde{z}_0^{min} = -4$.

Beispiel 7:

Wir wenden den 2-Phasen Simplexalgorithmus auf das Beispiel 1 an. Nach Transformation in die Standardform erhält man:

$$\overline{z} = -25x_1 - 17x_2 - 12x_3 \rightarrow \max!$$

$$\begin{array}{lccccccccccccccc}
\text{u.d.N.} & x_1 & + & x_2 & + & x_3 & + & x_{A1} & & & & & & & = & 100 \\
& & & & & x_3 & & & + & x_4 & & & & & = & 30 \\
& x_1 & & & + & x_3 & & & & & + & x_5 & & & = & 50 \\
& & & x_2 & + & x_3 & & & & & & & + & x_6 & = & 90
\end{array}$$

$$x_1, x_2, x_3, x_4, x_5, x_6, x_{A1} \geq 0.$$

Wir beginnen mit Phase I und ersetzen die Funktion z durch

$$z_I = -x_{A1} \longrightarrow \max!.$$

Man erhält folgendes Ausgangstableau:

	NBV	$\mathbf{x_1}$	x_2	x_3		
BV	-1	0	0	0	0	Q
x_{A1}	-1	1	1	1	100	100
x_4	0	0	0	1	30	–
$\mathbf{x_5}$	0	**1**	0	1	50	50
x_6	0	0	1	1	90	–
		-1	-1	-1	-100	

Mit x_1 als neuer BV erhält man die obigen Quotienten und dann x_5 als neue NBV. Das ergibt:

	NBV	x_5	$\mathbf{x_2}$	x_3		
BV	-1	0	0	0	0	Q
$\mathbf{x_{A1}}$	-1	-1	**1**	0	50	50
x_4	0	0	0	1	30	–
x_1	0	1	0	1	50	–
x_6	0	0	1	1	90	90
		1	-1	0	-50	

Tauscht man jetzt x_2 mit x_{A1} erhält man das folgende Tableau (die jetzt überflüssige Variable x_{A1} ist weggelassen):

	NBV	x_5	x_3		
BV	-1	0	0	0	Q
x_2	-1	-1	0	50	
x_4	0	0	1	30	
x_1	0	1	1	50	
x_6	0	1	1	40	
		0	0	0	

Phase I ist beendet und wir betrachten die Ausgangszielfunktion

$$\overline{z} = -25x_1 - 17x_2 - 12x_3 \longrightarrow \max!$$

Wir berechnen die Koeffizienten der Zielfunktionszeile und erhalten das folgende Tableau:

	NBV	x_5	$\mathbf{x_3}$		
BV	-1	0	-12	0	Q
x_2	-17	-1	0	50	–
$\mathbf{x_4}$	0	0	**1**	30	30
x_1	-25	1	1	50	50
x_6	0	1	1	40	40
		-8	-13	-2100	

Nach Austausch von x_3 und x_4 erhält man:

	NBV	$\mathbf{x_5}$	x_4		
BV	-1	0	0	0	Q
x_2	-17	-1	0	50	$-$
x_3	-12	0	1	30	$-$
x_1	-25	1	-1	20	20
$\mathbf{x_6}$	0	**1**	-1	10	10
		-8	13	-1710	

Jetzt werden x_5 und x_6 ausgetauscht, und es ergibt sich:

	NBV	x_6	x_4		
BV	-1	0	0	0	Q
x_2	-17	1	-1	60	
x_3	-12	0	1	30	
x_1	-25	-1	0	10	
x_5	0	1	-1	10	
		8	5	-1630	

Das letzte Tableau ergibt die folgende Optimallösung:

$$x_1 = 10, \quad x_2 = 60, \quad x_3 = 30, \quad x_4 = 0, \quad x_5 = 10, \quad x_6 = 0$$

mit dem Zielfunktionswert $z_0^{min} = 1630$ für das Minimierungsproblem.

BEISPIEL 8:

Wir betrachten das folgende LOP:

$$\begin{array}{rrcrcl} & \multicolumn{5}{c}{z = x_1 + 2x_2 \to \max!} \\ \text{u.d.N.} & x_1 & - & x_2 & \geq & 1 \\ & 5x_1 & - & 2x_2 & \leq & 3 \\ & \multicolumn{5}{l}{x_1, x_2 \geq 0.} \end{array}$$

Nach Transformation in die Standardform ergibt sich

$$\begin{array}{rcrcrcrcrcl} \multicolumn{11}{c}{z = x_1 + 2x_2 \to \max!} \\ x_1 & - & x_2 & - & x_3 & + & x_{A1} & & & = & 1 \\ 5x_1 & - & 2x_2 & & & & & + & x_4 & = & 3 \\ \multicolumn{11}{c}{x_1, x_2, x_3, x_4, x_5 \geq 0.} \end{array}$$

Man erhält folgendes Starttableau für Phase I mit der Hilfszielfunktion $z_I = -x_{A1} \to \max!$

	NBV	$\mathbf{x_1}$	x_2	x_3		
BV	-1	0	0	0	0	Q
x_{A1}	-1	1	-1	-1	1	1
$\mathbf{x_4}$	0	**5**	-2	0	3	$\frac{3}{5}$
		-1	1	1	-1	

Nach Austausch von den Variablen x_1 und x_4 ergibt sich:

	NBV	x_4	x_2	x_3		
BV	-1	0	0	0	0	Q
x_{A1}	-1	$-\frac{1}{5}$	$-\frac{3}{5}$	-1	$\frac{2}{5}$	
x_1	0	$\frac{1}{5}$	$-\frac{2}{5}$	0	$\frac{3}{5}$	
		$\frac{1}{5}$	$\frac{3}{5}$	1	$-\frac{2}{5}$	

Phase I endet mit $z_I^{max} < 0$. Daher hat dieses LOP keine zulässige Lösung. Im letzten Tableau ist die künstliche Variable x^{A1} noch positiv (d.h. die Nebenbedingung $x_1 - x_2 - x_3 = 1$ ist verletzt).

Übungsaufgaben

1. Lösen Sie die folgenden Probleme grafisch:

(a)
$$
\begin{array}{lrcrcr}
 & & z = x_1 + x_2 \to \min! & & & \\
\text{u.d.N.} & x_1 & + & x_2 & \geq & 2 \\
 & x_1 & - & x_2 & \geq & 3 \\
 & x_1 & + & 2x_2 & \leq & 6 \\
 & -x_1 & + & 4x_2 & \geq & 0 \\
 & & & x_1, x_2 & \geq & 0
\end{array}
$$

(b)
$$
\begin{array}{lrcrcr}
 & & z = x_1 + 4x_2 \to \min! & & & \\
\text{u.d.N.} & x_1 & - & 2x_2 & \leq & 4 \\
 & -x_1 & + & 2x_2 & \leq & 4 \\
 & x_1 & + & 2x_2 & \leq & 8 \\
 & & & x_1, x_2 & \geq & 0
\end{array}
$$

(c)
$$
\begin{array}{lrcrcr}
 & & z = -x_1 - 2x_2 \to \min! & & & \\
\text{u.d.N.} & -x_1 & + & x_2 & \leq & 4 \\
 & x_1 & + & 2x_2 & \leq & 11 \\
 & 2x_1 & + & x_2 & \leq & 10 \\
 & x_1 & & & \leq & 4 \\
 & & & x_1, x_2 & \geq & 0
\end{array}
$$

(d)
$$
\begin{array}{lrcrcr}
 & & z = x_1 + x_2 \to \max! & & & \\
\text{u.d.N.} & x_1 & - & x_2 & \geq & 0 \\
 & -x_1 & - & 2x_2 & \leq & 4 \\
 & & & x_1, x_2 & \geq & 0
\end{array}
$$

2. Ermitteln Sie die Standardform der folgenden LOP:

(a)
$$
\begin{array}{lrcrcrcr}
 & & & z = x_1 - 2x_2 + x_3 \to \min! & & & & \\
\text{u.d.N.} & x_1 & + & x_2 & + & x_3 & \leq & 7 \\
 & 3x_1 & - & x_2 & + & x_3 & \geq & -4 \\
 & & & & & x_1, x_2, x_3 & \geq & 0
\end{array}
$$

(b)
$$
\begin{array}{lrcrcrcrcr}
 & & & z = -x_1 + 2x_2 - 3x_3 + x_4 \to \min! & & & & & & \\
\text{u.d.N.} & 2x_1 & + & 2x_2 & - & x_3 & + & 3x_4 & = & 8 \\
 & x_1 & & & + & 2x_3 & + & x_4 & \leq & 10 \\
 & -2x_1 & & & + & 2x_3 & - & 3x_4 & \geq & 0 \\
 & \multicolumn{9}{l}{x_1 \leq 0, \quad x_2, x_3 \geq 0, \quad x_4 \in \mathbb{R}}
\end{array}
$$

3. (a) Ermitteln Sie eine optimale Lösung für die folgenden Probleme sowohl grafisch als auch mittels Simplexalgorithmus:

$$
\begin{array}{lrcrcr}
 & & z = x_1 + x_2 \to \max! & & & \\
\text{u.d.N.} & 3x_1 & + & 2x_2 & \leq & 6 \\
 & x_1 & + & 4x_2 & \leq & 4 \\
 & & & x_1, x_2 & \geq & 0
\end{array}
$$

4. Lösen Sie die folgenden Probleme mittels Simplexalgorithmus:

(a)
$$
\begin{array}{lrcrcrcrcr}
 & & & z = 7x_1 + 4x_2 + 5x_3 + 6x_4 \to \max! & & & & & & \\
\text{u.d.N.} & 20x_1 & + & 10x_2 & + & 12x_3 & + & 16x_4 & \leq & 400 \\
 & & & & & x_3 & & & \leq & 5 \\
 & x_1 & + & x_2 & + & x_3 & + & x_4 & \leq & 30 \\
 & \multicolumn{9}{l}{x_1, x_2, x_3, x_4 \geq 0}
\end{array}
$$

(b)
$$
\begin{array}{lrcrcrcr}
 & & & z = 2x_1 - 6x_2 \to \min! & & & & \\
\text{u.d.N.} & 2x_1 & - & x_2 & & & \leq & 10 \\
 & x_1 & - & 3x_2 & & & \leq & 15 \\
 & 3x_1 & & & + & x_3 & = & 12 \\
 & & & & & x_1, x_2, x_3 & \geq & 0
\end{array}
$$

5. Lösen Sie die folgenden Probleme mit dem Simplexalgorithmus:

   (a)
$$z = x_1 + x_2 - x_3 \to \text{min!}$$
$$\begin{array}{lrcrcrcr} \text{u.d.N.} & 3x_1 & - & x_2 & - & 4x_3 & \leq & 0 \\ & x_1 & + & 2x_2 & & & \geq & 10 \\ & & & x_2 & + & 3x_3 & \geq & 4 \\ & & & & & x_1, x_2, x_3 & \geq & 0 \end{array}$$

   (b)
$$z = x_1 + 2x_2 + x_3 \to \text{max!}$$
$$\begin{array}{lrcrcrcr} \text{u.d.N.} & x_1 & + & x_2 & & & \leq & 10 \\ & & & x_2 & + & x_3 & \leq & 14 \\ & x_1 & & & + & x_3 & \geq & 15 \\ & & & & & x_1, x_2, x_3 & \geq & 0 \end{array}$$

   (c)
$$z = 2x_1 - x_2 + x_3 \to \text{max!}$$
$$\begin{array}{lrcrcrcrcr} \text{u.d.N.} & x_1 & + & x_2 & - & x_3 & - & x_4 & = & 4 \\ & 2x_1 & + & 3x_2 & - & x_3 & - & 2x_4 & = & 9 \\ & & & x_2 & + & 2x_3 & & & \geq & 3 \end{array}$$
$$x_1, x_2, x_3, x_4 \geq 0$$

Kapitel 2

Diskrete Optimierung

2.1 Grundbegriffe und Beispiele

Diskretes Optimierungsproblem:

$$
\begin{aligned}
f(\mathbf{x}) &\to \min! \quad (\max!) \\
\mathbf{x} &\in S
\end{aligned}
$$

S ist eine diskrete Menge, d.h.

für alle $\mathbf{x} \in S$ existiert eine offene Umgebung, die außer $\mathbf{x}$ kein weiteres Element enthält („isolierte Punkte").

Spezialfall: S ist endlich.

$\rightarrow$ Oft wird S durch (lineare) Ungleichungen/Gleichungen beschrieben.

Ganzzahliges (lineares) Optimierungsproblem:

$$f(\mathbf{x}) = \mathbf{c}^T \cdot \mathbf{x} \to \min! \quad (\max!)$$

u.d.N.

$$
\begin{aligned}
A \cdot \mathbf{x} &\leq \mathbf{b} \\
\mathbf{x} &\in \mathbb{Z}_+^n
\end{aligned}
$$

Parameter A, **b**, **c** ganzzahlig

$\mathbb{Z}_+^n$ - Menge der ganzzahligen, nichtnegativen, n-dimensionalen Vektoren

Gemischt-ganzzahliges (lineares) Optimierungsproblem:

ersetze $\mathbf{x} \in \mathbb{Z}_+^n$ durch

$$
\begin{aligned}
x_1, x_2, \ldots, x_r &\in \mathbb{Z}_+ \\
x_{r+1}, x_{r+2}, \ldots, x_n &\in \mathbb{R}_+
\end{aligned}
$$

Binäres Optimierungsproblem:

ersetze $\mathbf{x} \in \mathbb{Z}_+^n$ durch

$$\begin{aligned} x_1, x_2, \ldots, x_n &\in \{0,1\} \\ \text{d.h.} \quad \mathbf{x} &\in \{0,1\}^n \end{aligned}$$

Gemischt-binäres Optimierungsproblem:

ersetze $\mathbf{x} \in \mathbb{Z}_+^n$ durch

$$\begin{aligned} x_1, x_2, \ldots, x_r &\in \{0,1\} \\ x_{r+1}, x_{r+2}, \ldots, x_n &\in \mathbb{R}_+ \end{aligned}$$

Kombinatorisches Optimierungsproblem (KOP):

Die Menge S ist endlich und nicht leer.

Anwendungen:

- Zuordnungsprobleme, z.B. Stundenplanprobleme, Frequenzzuweisung im Mobilfunk
- Reihenfolgeprobleme, z.B. Scheduling-Probleme, Travelling Salesman Problem (TSP)
- Auswahlprobleme, z.B. Rucksackproblem, Set Covering (Mengenüberdeckung), Set Partitioning (Mengenaufteilung)

Bemerkungen:

1. In der Regel lassen sich die Nebenbedingungen eines kombinatorischen Optimierungsproblems durch lineare Gleichungen/Ungleichungen mit ganzzahligen (oder binären) Entscheidungsvariablen $x_1, x_2, \ldots, x_n$ beschreiben.

   $\Rightarrow$ S ist Menge aller ganzzahligen Gitterpunkte eines konvexen Polyeders in $\mathbb{R}^n$.

2. Angenommen, Variable x kann nur endlich viele Werte $z_1, z_2, \ldots, z_k$ annehmen.

   $\Rightarrow$ Ersetze x durch k binäre Variablen $u_1, u_2, \ldots, u_k$ wie folgt:

$$\begin{aligned} x = z_1 \cdot u_1 + z_2 \cdot u_2 + \ldots + z_k \cdot u_k \\ u_1 + u_2 + \ldots + u_k = 1 \\ u_i \in \{0,1\}, \quad i = 1, 2, \ldots, k \end{aligned}$$

3. Lineare Optimierungsprobleme sind in polynomialer Zeit lösbar. Jedoch sind Verfahren der linearen Optimierung (z.B. Simplexalgorithmus) <u>nicht</u> auf ganzzahlige und kombinatorische Probleme anwendbar.

Beispiele:

Beispiel 1: Zuordnungsproblem ✏

Beispiel 2: Investitionsplanung ✏

Ein Unternehmen zieht 5 Projekte mit den folgenden Aufwendungen (in Mio. €) für die nächsten 3 Jahre in Betracht.

Projekt	Jahr 1	Jahr 2	Jahr 3	Gewinn
1	5	1	8	20
2	4	7	10	40
3	3	9	2	20
4	7	4	1	15
5	8	6	10	30
Verfügbare Mittel	25	25	25	

Welche Projekte sollten mit dem Ziel der Gewinnmaximierung realisiert werden?

Beispiel 3: Zuschnittoptimierung ✏

Drei Leisten sind zur Herstellung eines bestimmten Erzeugnisses notwendig. Zwei Leisten müssen je 1,5 m und eine Leiste muss 2 m lang sein. Es stehen 300 Leisten mit einer Länge von je 6,5 m und 80 Leisten mit einer Länge von je 5,5 m zur Verfügung.
Wie sind die zur Verwendung stehenden Leisten zuzuschneiden, damit eine maximale Stückzahl des Erzeugnisses hergestellt werden kann? Es sollen alle Leisten genutzt werden.

Bestimmung aller Zuschnittvarianten:

(a) Eine 6,5 m Leiste kann nach folgenden Varianten in 2 m bzw. 1,5 m Leisten geteilt werden:

	Anzahl der 2 m Leisten	Anzahl der 1,5 m Leisten
Variante 1:	3	0
Variante 2:	2	1
Variante 3:	1	3
Variante 4:	0	4

(b) Eine 5,5 m Leiste kann nach folgenden Varianten in 2 m bzw. 1,5 m Leisten geteilt werden:

	Anzahl der 2 m Leisten	Anzahl der 1,5 m Leisten
Variante 5:	2	1
Variante 6:	1	2
Variante 7:	0	3

Modellierung verschiedener Sachverhalte:

1. Angenommen, bei der Serienproduktion eines Gutes treten Fixkosten β auf, wenn eine positive Menge produziert wird. Dabei beschreibt x die produzierten Mengeneinheiten.

   $\rightarrow$ Kosten $c(x)$:

$$c(x) := \begin{cases} \alpha x + \beta & \text{falls } x > 0 \\ 0 & \text{falls } x = 0 \end{cases}$$

$\rightarrow$ Führe binäre Variable u ein mit

$$u := \begin{cases} 1 & \text{falls } x > 0 \\ 0 & \text{falls } x = 0 \end{cases}$$

$\Rightarrow$ Kosten haben die Form

$$c(x) = \alpha x + \beta u \quad (x \geq 0)$$

2. Mit Hilfe binärer Variablen lassen sich auch nicht zusammenhängende Bereiche durch Gleichungen bzw. Ungleichungen beschreiben.
   Betrachte:

$$f(x_1, x_2) \rightarrow \min!$$

u.d.N.

$$(x_1, x_2) \in M_1 \cup M_2$$

Illustration der Bereiche ✏

$$\Rightarrow \quad f(x_1, x_2) \rightarrow \min!$$

u.d.N.

$$\begin{aligned} 2x_1 + 3x_2 &\leq 6 \\ x_1 + 2u_1 &\geq 2 \\ x_2 + u_2 &\geq 1 \\ u_1 + u_2 &\leq 1 \\ \\ x_1, x_2 &\in \mathbb{R}_+ \\ u_1, u_2 &\in \{0, 1\} \end{aligned}$$

$\Rightarrow$ für (u_1, u_2) sind 3 Paare zulässig:

$$\begin{aligned} (u_1, u_2) = (0, 0) \quad &\Rightarrow \nexists (x_1, x_2) \in M_1 \cup M_2 \\ (u_1, u_2) = (1, 0) \quad &\Rightarrow (x_1, x_2) \in M_1 \\ (u_1, u_2) = (0, 1) \quad &\Rightarrow (x_1, x_2) \in M_2 \end{aligned}$$

Grafische Lösung: Im Fall $n = 2$ lässt sich ein optimaler Gitterpunkt grafisch ermitteln.

Beispiele (fortgesetzt):

Beispiel 4: Runden oder Abschneiden der optimalen Lösung eines LOP ✏

2.2 Lösungsmethoden

2.2.1 Definitionen und Verfahrensklassen

exakte Verfahren: Bestimmung einer optimalen Lösung in endlich vielen Schritten

vollständige Enumeration: nur bei *sehr kleinen* Problemen möglich

Verfahrensklassen:

- *Branch and Bound Verfahren*
  - Verfahren der impliziten Enumeration: Schließe sukzessiv Teilmengen von S aus, die keine optimale Lösung des Problems enthalten können.
  - Grundidee für Minimierungsprobleme:
    - Verzweigung (Branch): Teile die Lösungsmenge in mindestens zwei (disjunkte) Teilmengen auf.
    - Beschränkung (Bound): Berechne für jede Teilmenge S_i eine untere Schranke LB_i (lower bound).
    - Sei UB eine bekannte obere Schranke (upper bound) und gilt $LB_i \geq UB$ für S_i, so braucht S_i nicht weiter betrachtet werden.
- *Schnittebenenverfahren (für ganzzahlige Optimierung)*
  - Relaxiere die Lösungsmenge durch Streichung der Ganzzahligkeitsbedingung ($\rightarrow$ LOP).
  - Bestimme die (stetige) Optimallösung $\mathbf{x}$ des LOP.
  - Sind einzelne Komponenten von $\mathbf{x}$ nicht ganzzahlig, füge systematisch zusätzliche Nebenbedingungen (*Schnitte*) ein, wobei die ursprüngliche optimale Lösung abgeschnitten wird. Es darf kein ganzzahliger Punkt von S abgeschnitten werden.
- *Dynamische Optimierung* (siehe Kapitel 4)
- *Heuristische Verfahren*
  - bestimmen nur eine Näherungslösung
  - Konstruktive Verfahren: bestimmen eine zulässige Lösung (z.B. Greedy-Verfahren, Prioritätsregelverfahren)
  - Iterative Verfahren: verbessern eine gegebene zulässige Lösung (z.B. Lokale Suche, Metaheuristiken - siehe Kapitel 3)
  - Verkürzte exakte Verfahren: z.B. vorzeitig abgebrochene Branch and Bound Verfahren

2.2.2 Heuristische Verfahren

(a) Konstruktive Verfahren (Eröffnungsverfahren)

wichtige Klasse: Greedy-Algorithmen („gieriger“ Algorithmus)

- treffen in jedem Schritt eine „lokal-optimale“ Entscheidung
- führen in manchen Fällen auch zu einer global-optimalen Lösung

Beispiele und Anwendungen:

- Algorithmus von Kruskal zur Bestimmung eines Minimalgerüstes
- Algorithmus von Dijkstra zur Suche eines kürzesten Weges in einem ungerichteten Graphen
- Bestimmung einer Startlösung für das Transportproblem, z.B. mittels Zeilen- bzw. Spaltenminimumregel
- Einfügeverfahren für das Travelling Salesman Problem (TSP)

Bestimmung eines Minimalgerüsts

Definition 1: Ein zusammenhängender, alle Knoten eines gegebenen (ungerichteten) Graphen G enthaltender Teilgraph von G mit minimaler Kantenanzahl heißt **Gerüst** von G.

$$G = [V, E, c]$$

Bemerkungen: ✏

1. G besitzt ein Gerüst. $\Leftrightarrow$ G ist zusammenhängend.
2. Ein Gerüst ist ein alle n Knoten enthaltender Baum mit $n-1$ Kanten.

Definition 2: Ein Gerüst mit minimaler Summe der Kantenbewertungen heißt **Minimalgerüst**.

Algorithmus von Kruskal

Schritt 1:

Wähle eine Kante $[k, l]$ mit kleinster Bewertung c_{kl} aus. Diese bildet (mit den Knoten k, l) den Teilgraphen G^1.

Schritt i $(i = 2, 3, \dots, n-1)$:

Füge dem bisher erhaltenen Teilgraphen G^{i-1} eine weitere Kante $[u, v]$ mit kleinstmöglicher Bewertung (sowie die Knoten u, v) hinzu, so dass der neue Teilgraph G^i (der nicht zusammenhängend sein muss) keinen Kreis enthält.

$\Rightarrow$ Algorithmus erzeugt eine Folge von Wäldern, wobei der Graph G^{n-1} genau $n-1$ Kanten besitzt und ein Baum (ungerichteter zusammenhängender kreisfreier Graph) ist.

Beispiel 8: Algorithmus von Kruskal ✏

Travelling Salesman Problem:

Ein Handelsvertreter möchte, beginnend in seinem Heimatort (Knoten 1), $n-1$ weitere Kundenorte (Knoten $2, \dots, n$) in einer zu ermittelnden Reihenfolge besuchen und anschließend zum Ausgangspunkt zurückkehren. Dabei soll die insgesamt zurückgelegte Entfernung minimal sein.

Also: Im Graph ist ein Zyklus (Tour; Rundreise) mit minimaler Länge gesucht, der alle Knoten genau einmal enthält.

Konstruktionsverfahren für das TSP

Varianten:

– symmetrischer TSP

– asymmetrisches TSP

- **Heuristik des nächsten Nachbars***:*

  – Beginnend mit einem Startknoten wird als nächstes der verfügbare Knoten mit der kürzesten Entfernung gesucht und die Kante angefügt, bis eine vollständige Tour vorliegt.

- **Greedy-Heuristk***:*

  – Beginnend mit der kürzesten Kante werden schrittweise Kanten hinzugefügt, bis die Tour komplett ist. In jedem Schritt wird dabei die kürzestmögliche Kante gewählt, ohne die Zulässigkeit zu verletzen.

- **Einfügeverfahren***:*

  – Beginnend mit einer Tour bestehend aus einer Stadt werden Städte hinzugefügt, bis alle Städte besucht sind.

  Dabei existieren verschiedene Varianten, z.B.

  – **Nearest Insertion:** Einfügen der Stadt mit geringster Differenz zu einer Stadt aus der aktuellen Tour

  – **Farthest Insertion:** Einfügen der Stadt, bei der die geringste Distanz zu einer Stadt aus der Tour maximal ist

  – **Cheapest Insertion:** Einfügen der Stadt, die die geringste Zunahme der Tourlänge bewirkt

  Einfügeposition: Füge die neue Stadt so ein, dass die Tourlängenzunahme minimial ist

Beispiel 9: Anwendung des Einfügeverfahrens auf das TSP ✏

Verallgemeinerung: **Tourenoptimierung:**

Ausgehend von einem Depot 0 sollen n Kunden mit Fahrzeugen gleicher Ladakapazität Q mit einem Gut beliefert werden, wobei die Fahrzeiten von jedem Kunden zum Depot und vom Depot zu jedem Kunden bekannt sind.

Gesucht ist eine Menge von Fahrten mit minimaler Gesamtdauer, so dass:

- jede Fahrt am Depot beginnt und endet;

- jeder Kunde auf genau einer Fahrt bedient wird;

- die Fahrzeugkapazität nicht überschritten wird sowie

- eine für jede Fahrt gleiche Maximaldauer nicht überschritten wird.

Savingsverfahren*:*

Schritt 1:

Bilde die Pendeltouren $(0, i, 0)$. für $i = 1, \ldots, n$

Berechne für alle $i, j = 1, \ldots, n$ mit $i \neq j$ die **Savings**

$$s_{ij} = t_{i0} + t_{0j} - t_{ij}$$

und speichere die positiven s_{ij} in einer Liste L.

Schritt 2: Falls $L = \emptyset$, STOP!

Andernfalls entferne das erste (d.h. größte) Element aus L, sei dies s_{ij}.

Ist s_{ij} ein zulässiges Saving, so verschmelze die entsprechenden beiden Touren mit der Verbindung (i, j).

Gehe zu Schritt 2.

Beispiel 10: Anwendung des Savingsverfahrens auf die Tourenoptimierung ✏

(b) Iterative Verfahren (Verbesserungsverfahren)

→ siehe Kapitel 3

2.2.3 Software-Pakete zur Lösung ganzzahliger und binärer, linearer Optimierungsprobleme

(a) Excel Solver

(b) LINGO/LINDO

(c) MPL/CPLEX

Einige Bemerkungen zur Nutzung von (b) LINGO:

→ kann als Studentenversion heruntergeladen werden: http://www.lindo.com

(maximal 150 Nebenbedingungen, 300 Variablen, 30 ganzzahlige Variablen)

Modellierung in LINGO

Entscheidungsvariablen

SETS:

WOCHE/1...10/: pmenge,pcost;

ENDSETS

→ definiert Vektoren pmenge und pcost der Dimension 10

Zielfunktion und Nebenbedingungen

MIN=@SUM(WOCHE(i): pcost(i)*pmenge(i));

→ Zielfunktion: $\sum_{i=1}^{10} \text{pcost}(i) \cdot \text{pmenge}(i) \ \rightarrow \min!$

@FOR(WOCHE(i): pmenge(i)<=30);

→ Nebenbedingung: pmenge(i) ≤ 30 für alle $i = 1, \ldots, 10$

DATA:

pcost = 100,50,80,120,30,50,10,60,20,90;

ENDDATA

→ Vektor pcost = $(100, 50, 80, 120, 30, 50, 10, 60, 20, 90)^T$ gesetzt

Generierung des Problems

LINGO → Generate → Display Model

Lösen des Problems

LINGO → Solve

Beispielmodellierung

Betrachtet wird das folgende ganzzahlige Problem der Produktionsplanung:

$$\text{z} = 6x_1 + 12x_2 + 10x_3 + 5x_4 \rightarrow \text{max!}$$

$$\begin{array}{lccccccccc} \text{u.d.N.} & 2x_1 & + & 2x_2 & + & 3x_3 & + & x_4 & \leq & 27 \\ & x_1 & + & 5x_2 & + & x_3 & + & 2x_4 & \leq & 25 \\ & x_1 & & & & & & & \geq & 7 \\ & & & & & x_3 & + & x_4 & = & 5 \end{array}$$

$$x_1, x_2, x_3, x_4 \in \mathbb{Z}_+$$

Es können vier Produkte in ganzzahligen Mengen hergestellt werden, $x_i \in \mathbb{Z}_+$ bezeichne die von Produkt i hergestellte Menge - im nachfolgenden Modell mit prod(i) bezeichnet. Der Gewinn soll maximiert werden und es bestehen die oben angegebenen Restriktionen.

LINGO-Modell:

```
MODEL:
SETS:
PRODUKT/1..4/: prod, gewinn;
ENDSETS
MAX = @SUM(PRODUKT(i): gewinn(i)*prod(i));
2*prod(1) + 2*prod(2) + 3*prod(3) + prod(4) <= 27;
prod(1) + 5*prod(2) + prod(3) + 2*prod(4) <= 25;
prod(1) >= 7;
prod(3) + prod(4) = 5;
@FOR(PRODUKT(i): @GIN(prod(i)));
DATA:
gewinn = 6, 12, 10, 5;
ENDDATA
END
```

Optimallösung:

$x_1 = 7, \quad x_2 = 2; \quad x_3 = 2; \quad x_4 = 3;$

$z = 101$

Bemerkungen

- Zwei Dateitypen in LINGO: .lng (reine Textdateien) und .lg4
- Variablen standardmäßig rational, nichtnegativ

@GIN	ganzzahlig, nichtnegative Variable
@BIN	binäre Variablen
@FREE	frei, d.h. nicht vorzeichenbeschränkt
	z.B. @FOR(WOCHE(i):@GIN(pmenge(i)));
	⇒ pmenge(i) nichtnegativ und ganzzahlig

- logische Operationen: #EQ#, #NE#, #GE#, #GT#, #LE#, #LT#, #AND#, #OR#
- LINGO nicht „case-sensitive": SUM, sum, SuM identisch

2.3 Das Rucksackproblem

Problem: *Ein Bergsteiger hat n Gegenstände $1, 2, \ldots, n$ zur Verfügung, wobei*

c_i - Wert von Gegenstand i

a_i - Volumen von Gegenstand i

V - Volumen des Rucksacks

Ziel: *Bestimme eine Rucksackfüllung mit maximalem Gesamtwert, wobei das Volumen V nicht überschritten wird.*

⇒ Führe eine binäre Variable x_i ein wie folgt:

$$x_i = \begin{cases} 1, \text{ falls Gegenstand } i \text{ in den Rucksack gepackt wird} \\ 0, \text{ sonst} \end{cases}$$

$$\text{für } i = 1, 2, \ldots, n$$

⇒ mathematisches Modell:

$$\boxed{\begin{array}{l} \displaystyle\sum_{i=1}^{n} c_i x_i \to \max! \\ \text{u.d.N.} \\ \left.\begin{array}{l} \displaystyle\sum_{i=1}^{n} a_i x_i \leq V \\ x_1, x_2, \ldots, x_n \in \{0, 1\} \end{array}\right\} =: S \end{array}} \qquad \text{(R)}$$

Das Problem (R) ist ein binäres Optimierungsproblem mit nur einer Nebenbedingung.

Greedy-Algorithmus:

Schritt 1:

Nummeriere die n Gegenstände nach nichtwachsenden Quotienten $\frac{c_i}{a_i}$ und setze $f_G := 0$.

Schritt 2:

Führe für $j = 1, 2, \ldots, n$ aus:

Falls $a_j > V$, setze $k := j$, $x_j^G := 0$ und gehe zu Schritt 3, andernfalls setze $x_j^G := 1$, $f_G := f_G + c_j$ und $V := V - a_j$.

Schritt 3:

Führe für $j = k+1, k+2, \ldots, n$ aus:

Falls $a_j > V$, setze $x_j^G := 0$, andernfalls setze $x_j^G := 1$, $f_G := f_G + c_j$ und $V := V - a_j$.

k - kritischer Index

Beispiel 10: Anwendung des Greedy-Algorithmus beim Rucksackproblem ✏

Der Greedy-Algorithmus für das ganzzahlige Rucksackproblem

Ersetze im binären Problem $\mathbf{x} \in \{0,1\}^n$ durch $\mathbf{x} \in \mathbb{Z}_+^n$.

Schritt 1:

Nummeriere die Gegenstände nach nichtwachsenden Quotienten $\frac{c_i}{a_i}$ und setze $f_G := 0$.

Schritt 2:

Führe für $j = 1, 2, \ldots, n$ aus:

setze $x_j^G := \left[\frac{V}{a_j}\right]$; $\quad f_G := f_G + c_j x_j^G$ und $V := V - a_j x_j^G$.

Bemerkung: Sei f^* der optimale Zielfunktionswert, dann gilt:

$$f^* \leq c_1 \frac{V}{a_1}\ , \quad f_G \geq c_1 \left[\frac{V}{a_1}\right]$$

$$f^* - f_G \leq c_1 \left(\frac{V}{a_1} - \left[\frac{V}{a_1}\right] \right) < c_1 \leq \max_{j=1,\ldots,n} c_j$$

Interpretation: c_j klein $\Rightarrow$ häufig $\mathbf{x}^G$ gut

Übungsaufgaben

1. Aus gegebenem Stangenmaterial der Länge 6500 mm sind mindestens 50 Stangen von 2000 mm Länge, 100 Stangen von 1600 mm Länge und 100 Stangen von 1100 mm Länge anzufertigen.
   (a) Geben Sie alle Zuschnittvarianten an!
   (b) Formulieren Sie das entstehende Optimierungsproblem, wenn die Anzahl zu zerschneidender Stangen möglichst gering sein soll!
2. Aus Blechen der Größe 1000 mm x 1000 mm sollen kleinere rechteckige Bleche T_1, T_2, T_3 mindestens in den Stückzahlen b_1, b_2, b_3 zugeschnitten werden, wobei

$$\begin{array}{ll} T_1 : 400 \text{ mm x } 400 \text{ mm}, & b_1 = 15 \\ T_2 : 300 \text{ mm x } 600 \text{ mm}, & b_2 = 10 \\ T_3 : 700 \text{ mm x } 300 \text{ mm}, & b_3 = \ \ 5 \end{array}$$

   (a) Geben Sie die möglichen Zuschnittvarianten an, wenn alle Bleche des gleichen Typs eine gleiche Lage innerhalb eines Bleches haben müssen (d.h. die Drehung eines Teils des Typs T_i um 90^0 ist nicht zulässig).
   (b) Formulieren Sie das entstehende Optimierungsproblem, wenn der entstehende Abfall minimiert werden soll.
3. Lösen Sie das folgende Problem grafisch:

$$z = 2x_1 + 5x_2 \to \max!$$

$$\begin{array}{lrcrcr} \text{u.d.N.} & -5x_1 & + & 9x_2 & \leq & 36 \\ & x_1 & + & x_2 & \leq & 11 \\ & 3x_1 & - & 2x_2 & \leq & 18 \end{array}$$

$$x_1, x_2 \geq 0, \quad \text{ganzzahlig.}$$

4. Für das folgende Problem zeige man mittels grafischer Darstellung, dass keine ganzzahlige Lösung existiert:

$$z = x_1 + x_2 \to \max!$$

$$\begin{array}{lrcrcr} \text{u.d.N.} & 2x_1 & + & 6x_2 & \geq & 9 \\ & -2x_1 & + & 6x_2 & \geq & 3 \\ & x_1 & & & \leq & 3 \\ & & & 4x_2 & \leq & 7 \end{array}$$

$$x_1, x_2 \geq 0, \quad \text{ganzzahlig}$$

   Geben Sie die optimale nichtganzzahlige Lösung an.
5. Bestimmen Sie für das folgende lineare Optimierungsproblem grafisch eine optimale Lösung:

$$z = x_1 + x_2 \to \min!$$

$$\begin{array}{lrcrcr} \text{u.d.N.} & 5x_1 & + & 2x_2 & \geq & 10 \\ & 2x_1 & - & 3x_2 & \leq & 0 \end{array}$$

$$x_1, x_2 \geq 0.$$

   Führen Sie danach nur für x_1, nur für x_2 sowie für beide Variable Ganzzahligkeitsbedingungen ein und vergleichen Sie die optimalen Lösungen.

6. Ein Unternehmen fertige zwei Produkte, die **alternativ** auf zwei Fertigungslinien hergestellt werden können. Technologie 1 sei durch die Restriktionen

$$\begin{array}{rcrcr} x_1 & + & 5x_2 & \leq & 10 \\ x_1 & + & x_2 & \leq & 6 \end{array}$$

charakterisiert und Technologie 2 durch

$$\begin{array}{rcrcr} 2x_1 & + & 5x_2 & \leq & 20 \\ 2x_1 & + & x_2 & \leq & 6, \end{array}$$

wobei $x_i \geq 0$ die produzierte Menge von Produkt $i, i \in \{1, 2\}$ (in Tonnen) angeben. Der Gewinn pro Tonne von Produkt 1 betrage 1000 EUR und der Gewinn pro Tonne von Produkt 2 betrage 2000 EUR.

(a) Formulieren sie ein mathematisches Modell.
(b) Stellen Sie den zulässigen Bereich grafisch dar.
(c) Bestimmen Sie grafisch eine optimale Lösung.

7. Für die Herstellung von zwei Produkten werden **genau** zwei von drei verfügbaren Anlagen mit unterschiedlicher Produktionscharakteristik benötigt. Alle Anlagen stehen jeweils 36 Stunden zur Verfügung. Damit verbunden sind die folgenden Kapazitätsrestriktionen, die nicht alle gleichzeitig gelten:

$$\begin{array}{lrcrcr} \text{Anlage 1:} & 2x_1 & + & 6x_2 & \leq & 36 \\ \text{Anlage 2:} & 4x_1 & + & 4x_2 & \leq & 36 \\ \text{Anlage 3:} & 3x_1 & + & 2x_2 & \leq & 36 \end{array}$$

Der zu maximierende Gewinn ist $3x_1 + 4x_2$, wobei $x_i \geq 0$ die hergestellte Menge von Produkt $i, i \in \{1, 2\}$ bezeichnet.

(a) Formulieren sie ein mathematisches Modell.
(b) Stellen Sie den zulässigen Bereich grafisch dar.
(c) Bestimmen Sie grafisch eine optimale Lösung.

8. Die Forschungsabteilung eines Unternehmens untersucht die Einführung von drei neuen Produkten, die in zwei möglichen Fabriken hergestellt werden können. Die zugehörigen Daten sind wie folgt gegeben:

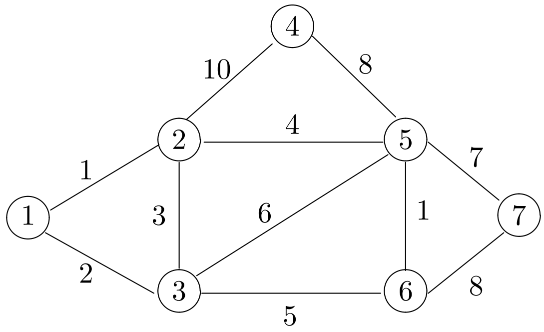
	Produktionszeit (h) per Einheit von			verfügbare
	Produkt 1	Produkt 2	Produkt 3	Kapazität (h)
Fabrik 1	5	4	2	45
Fabrik 2	4	6	3	50
Gewinn pro Einheit	4	5	2	(in 1000 EUR)
maximale Anzahl zu verkaufender Einheiten	8	7	nicht beschränkt	

Das Management fordert:

- Höchstens zwei der drei neuen Produkte sollen ausgewählt werden.
- Genau eine der beiden Fabriken soll alle ausgewählten neuen Produkte herstellen.

Formulieren Sie ein mathematisches Modell für das resultierende Optimierungsproblem.

9. Bestimmen Sie für den folgenden ungerichteten Graphen G mit den angegebenen Kantenbewertungen ein Minimalgerüst mittels Algorithmus von Kruskal!



10. Lösen Sie das folgende Rucksackproblem näherungsweise mittels Greedy-Algorithmus:

$$\begin{array}{cccccccccc} 5x_1 & +8x_2 & +9x_3 & +4x_4 & + & x_5 & +2x_6 & +4x_7 & +6x_8 & \rightarrow & \text{max!} \\ 3x_1 & +8x_2 & +6x_3 & +2x_4 & + & 2x_5 & +3x_6 & +6x_7 & +7x_8 & \leq & 16 \end{array}$$

$$x_1, \cdots, x_8 \in \{0,1\}$$

11. Gegeben sei das lineare Optimierungsproblem

$$\text{z} = 4x_1 + 5x_2 + 3x_3 + x_4 \rightarrow \text{max!}$$

$$\begin{array}{lccccccccc} \text{u.d.N.} & 3x_1 & + & 2x_2 & + & 4x_3 & + & x_4 & \leq & 20 \\ & 2,5x_1 & + & 4x_2 & & & + & 1,5x_4 & \leq & 16,5 \end{array}$$

$$x_1, x_2, x_3, x_4 \in \mathbb{R}_+$$

(a) Bestimmen Sie eine optimale Lösung mit dem Software-Paket LINGO.
(b) Ändern Sie den Vorzeichenbedingungen für die Variablen wie folgt:

$$x_1, x_2 \in \mathbb{Z}_+; \quad x_3 \in \{0,1\}; \quad x_4 \text{ beliebig}$$

und fügen Sie die Restriktion

$$x_3 + x_4 = -1$$

hinzu. Bestimmen Sie eine optimale Lösung mittels LINGO.

12. Lösen Sie das folgende Rucksackproblem näherungsweise mittels Greedy-Algorithmus:

$$\begin{array}{cccccccccc} 5x_1 & +8x_2 & +9x_3 & +4x_4 & + & x_5 & +2x_6 & +4x_7 & +6x_8 & \rightarrow & \text{max!} \\ 3x_1 & +8x_2 & +6x_3 & +2x_4 & + & 2x_5 & +3x_6 & +6x_7 & +7x_8 & \leq & 16 \end{array}$$

$$x_1, x_2, \ldots, x_8 \in \{0,1\}$$

13. Lösen Sie das folgende ganzzahlige Rucksackproblem mittels Greedy-Algorithmus:

$$\begin{array}{cccccccccccc} 7x_1 & +3x_2 & +x_3 & + & 6x_4 & + & 5x_5 & + & 2x_6 & \rightarrow & \text{max!} \\ 7x_1 & +4x_2 & +2x_3 & + & 7x_4 & + & 8x_5 & + & 5x_6 & \leq & 20 \end{array}$$

$$x_i \in \mathbb{Z}_+, \ \ i = 1, 2, \ldots, 6$$

Geben Sie eine Abschätzung für die Differenz zwischen dem Näherungswert des Greedy-Algorithmus und dem optimalen Zielfunktionswert an!

Kapitel 3

Metaheuristiken

3.1 Iterative Verfahren, Grundbegriffe

Lokale Suche (Nachbarschaftssuche)

Führe eine Nachbarschaftsstruktur wie folgt ein:

$$N \; : \; S \to 2^S$$
$$\mathbf{x} \in S \; \Rightarrow \; N(\mathbf{x}) \subseteq 2^S$$

S - Menge der zulässigen Lösungen

$N(\mathbf{x})$ - Menge der Nachbarn der zulässigen Lösung $\mathbf{x} \in S$

Basisalgorithmus: Iterative Verbesserung (Minimierung)

1. Bestimme eine Startlösung $\mathbf{x} \in S$;
   REPEAT
2. Ermittle die beste Lösung $\mathbf{x}' \in N(\mathbf{x})$;
3. IF $f(\mathbf{x}') < f(\mathbf{x})$ THEN $\mathbf{x} := \mathbf{x}'$;
4. UNTIL $f(\mathbf{x}') \geq f(\mathbf{x})$.

$\mathbf{x}'$ - lokaler Minimalpunkt bzgl. Nachbarschaft N

→ Der Algorithmus arbeitet nach dem Prinzip der „größten Verbesserung“ *(best-fit)*.

Modifikation:

Wende das Prinzip der „ersten Verbesserung“ *(first-fit)* an, d.h. durchsuche Nachbarn in systematischer Weise und akzeptiere Nachbarn mit besserem Zielfunktionswert (Zfw.) sofort als Startlösung für die nächste Iteration.

(Abbruch folgt, wenn ein voller Zyklus aller Nachbarn ohne eine Zielfunktionswertverbesserung durchlaufen wurde.)

$|$ $\mathbf{N(x)}$ $|$ **sehr groß** ⇒ Erzeuge Nachbarn zufällig.

⇒ Ersetze Zeile 2 im Algorithmus „Iterative Verbesserung“ durch

$$2^{*}\text{: Ermittle eine Lösung } \mathbf{x}' \in N(\mathbf{x})$$

Abbruch, falls

- vorgegebene maximale Rechenzeit verbraucht ist oder
- vorgegebene Anzahl zulässiger Lösungen erzeugt wurde oder
- vorgegebene Anzahl von Lösungen seit letzter Zielfunktionswertverbesserung erzeugt wurde, ohne den Zielfunktionswert weiter zu verbessern.

Betrachtet wird

$$\boxed{\begin{array}{l} \qquad f(\mathbf{x}) \to \min! \quad (\max!) \\ \text{u.d.N.} \\ \qquad \mathbf{x} \in S \subseteq \{0,1\}^n \end{array}} \tag{2.1}$$

Nachbarschaft $N_k(\mathbf{x})$:

$$N_k(\mathbf{x}) = \{\mathbf{x}' \in S \mid \sum_{i=1}^{n} \mid x_i - x_i' \mid \leq k\}$$

$(\mathbf{x}' \in N_k(\mathbf{x}') \;\Leftrightarrow\; \mathbf{x}'$ zulässig und unterscheidet sich höchstens in k Komponenten von $\mathbf{x})$

$$\Rightarrow \mid N_1(\mathbf{x}) \mid \leq n$$

$$\mid N_2(\mathbf{x}) \mid \leq n + \binom{n}{2} = n + \frac{n(n-1)}{2} = \frac{n(n+1)}{2}$$

Zur systematischen Erzeugung der Nachbarn ändere Komponente 1,2,... usw.

BEISPIEL 1: NACHBARSCHAFTSSUCHE ✏

Minimalgerüst mit Nebenbedingungen

Angenommen, Kante $[u, v]$ kann nur zum Gerüst gehören, falls eine Kante $[k, l]$ zum Gerüst gehört oder eine von mehreren Kanten muss zum Gerüst gehören.

→ Greedy-Algorithmus liefert nicht notwendig eine optimale Lösung.

Sei G eine Gerüst

⇒ $N(G)$ - Menge aller Gerüste G^*, die sich von G durch genau eine Kante unterscheiden.

iterative Verbesserung: Verfahren brechen in einem lokalen Optimum ab.

„Metaheuristiken“:

- Prinzipien zur Steuerung heuristischer Verfahren
- bauen auf lokaler Suche auf
- erlauben Übergänge zu Lösungen mit schlechterem Zfw.

3.2 Simulated Annealing

Randomisiertes Verfahren, da:

- $\mathbf{x}' \in N(\mathbf{x})$ wird zufällig gewählt
- im i-ten Schritt wird $\mathbf{x}'$ mit der Wahrscheinlichkeit

$$\min\left\{1,\ e^{\left(-\frac{f(\mathbf{x}')-f(\mathbf{x})}{t_i}\right)}\right\}$$

als neue Startlösung akzeptiert.

(t_i) - Folge positiver Steuerparameter mit $\lim\limits_{i\to\infty} t_i = 0$ („Temperatur")

Interpretation: (Minimierung!)

$f(\mathbf{x}') < f(\mathbf{x}) \ \Rightarrow$ Verbesserung wird immer akzeptiert

$f(\mathbf{x}') \geq f(\mathbf{x}) \ \Rightarrow$

t_i fixiert: $f(\mathbf{x}') - f(\mathbf{x})$ groß, dann wird $\mathbf{x}'$ mit kleiner Wahrscheinlichkeit akzeptiert

$f(\mathbf{x}') - f(\mathbf{x})$ fixiert: t_i klein, dann wird $\mathbf{x}'$ mit kleiner Wahrscheinlichkeit akzeptiert

<u>Algorithmus Simulated Annealing</u>

1. $i := 0$; wähle t_0;
2. bestimme eine Startlösung $\mathbf{x} \in S$;
3. $best := f(\mathbf{x})$;
4. $\mathbf{x}^* := \mathbf{x}$;

   REPEAT
5. erzeuge zufällig eine Lösung $\mathbf{x}' \in N(\mathbf{x})$;
6. IF $rand[0,1] < \min\left\{1, e^{\left(-\frac{f(\mathbf{x}')-f(\mathbf{x})}{t_i}\right)}\right\}$ THEN $\mathbf{x} := \mathbf{x}'$;
7. IF $f(\mathbf{x}') < best$ THEN

   BEGIN $\mathbf{x}^* := \mathbf{x}'$; $best := f(\mathbf{x}')$ END;
8. $t_{i+1} := g(t_i)$;
9. $i := i + 1$;

   UNTIL Abbruchkriterium ist erfüllt

Fragen

1. Wahl von g: $t_{i+1} := \alpha \cdot t_i$ mit $0 < \alpha < 1$ (geometrisches Abkühlungsschema)
   (oft: Zyklus mit konstanter Temperatur, danach Abkühlung)
2. Abbruchkriterium: analog zu lokaler Suche

Modifikation: Threshold Accepting

„deterministisches" Akzeptanzkriterium: akzeptiere $\mathbf{x}' \in N(\mathbf{x})$ falls $f(\mathbf{x}') - f(\mathbf{x}) \leq t_i$

$t_i \geq 0$ - Threshold im i-ten Schritt

3.3 Tabu-Suche

Grundstrategie:

Speichere bereits überprüfte Lösungen in einer Tabu-Liste TL und akzeptiere nur Nachbarn, die nicht in TL enthalten sind.

$\rightarrow$ aufwändig

- Nutze *Attribute*, um zuletzt besuchte Lösungen zu charakterisieren. ✏
- Länge der Tabu-Liste:

  konstant: Enthält die Liste t Attribute, so lösche bei jedem Übergang das älteste Attribut und nimm ein neues Attribut auf.

  variabel:

  - Lösche die TL bei Verbesserung des besten Zfw.
  - Nutze Schranken: $L_{min} \leq |\ TL\ | \leq L_{max}$.
  - Setze $|\ TL\ | = |\ TL\ | + 1$ falls $f(\mathbf{x}') > f(\mathbf{x})$
    oder $|\ TL\ | = |\ TL\ | - 1$ falls $f(\mathbf{x}') < f(\mathbf{x})$
- Nutzung des *Aspiration-Kriteriums*:

  verbotene Lösungen werden trotzdem akzeptiert (z.B. bei Verbesserung des besten Zfw.)
- $Cand(\mathbf{x}) = \{\mathbf{x}' \in N(\mathbf{x}) \mid$ Übergang von $\mathbf{x}$ zu $\mathbf{x}'$ ist nicht tabu oder $\mathbf{x}'$ erfüllt das Aspiration-Kriterium $\}$ („erlaubte Übergänge“)

  $|\ N(\mathbf{x})\ |$ klein $\Rightarrow$ Wähle besten Nachbarn $\mathbf{x}'$ aus $Cand(\mathbf{x})$.

  $|\ N(\mathbf{x})\ |$ groß $\Rightarrow$ Untersuche nur eine Teilmenge $V \subset Cand(\mathbf{x})$ und wähle besten Nachbarn $\mathbf{x}' \in V$.

Algorithmus Tabu-Suche

1. bestimme eine Startlösung $\mathbf{x} \in S$;
2. $best := f(\mathbf{x})$;
3. $\mathbf{x}^* := \mathbf{x}$;
4. $TL := \emptyset$;

   REPEAT
5. bestimme $Cand(\mathbf{x}) = \{\ \mathbf{x}' \in N(\mathbf{x}) \mid$
   der Übergang von $\mathbf{x}$ zu $\mathbf{x}'$ ist nicht tabu oder $\mathbf{x}'$ erfüllt das Aspiration-Kriterium $\}$;
6. wähle eine Lösung $\overline{\mathbf{x}} \in Cand(\mathbf{x})$;
7. aktualisiere TL;
8. $\mathbf{x} := \overline{\mathbf{x}}$;

9. IF $f(\overline{\mathbf{x}}) < best$ THEN
   BEGIN $\mathbf{x}^* := \overline{\mathbf{x}}$; $best := f(\overline{\mathbf{x}})$ END;
   UNTIL Abbruchkriterium ist erfüllt

Erweiterung: Diversifikation „Gute Lösungen" werden gespeichert. Wird der beste Zfw. während einer vorgegebenen Anzahl von Iterationen nicht verbessert, springe zu einer „guten Lösung" zurück. (LONG-TERM MEMORY)

BEISPIEL 2: TABU-LISTE

3.4 Genetische Algorithmen

- Nutzung von Darwins Evolutionstheorie („survival of the fittest")
- arbeitet mit einer *Population* von Individuen (*Chromosomen*), die durch ihre Fitness charakterisiert werden, z.B.

  fitness(ch)=$f(\mathbf{x})$ bei $f \to$ max!

  fitness(ch)=$\frac{1}{f(\mathbf{x})}$ bei $f \to$ min! und $f(\mathbf{x}) > 0$,

  wobei ch die Codierung der Lösung $\mathbf{x} \in S$ ist

$$\mathbf{x} = (0,1,1,1,0,1,0,1)^T \in \{0,1\}^8$$

$$\text{ch: } \boxed{0\,|\,1\,|\,1\,|\,1\,|\,0\,|\,1\,|\,0\,|\,1}$$

Beispiel: Binäre Codierung

Erzeugung von Nachkommen mittels genetischer Operatoren

Mutation:

„Mutiere" Gene eines Individuums.

Elternchromosom	0 1 1 1 0 1 0 1
(3,5)-Inversion	0 1 ‖ **0 1 1** ‖ 1 0 1
2-Mutation	0 **0** 1 1 0 1 0 1
(1,4,7)-Mutation	**1** 1 1 **0** 0 1 **1** 1

Crossover:

Kombiniere die genetischen Strukturen zweier Individuen und bilde zwei Nachkommen.

1-Punkt-Crossover z.B. (4,8)-Crossover

$$\begin{array}{lll} E_1 \; 1\,0\,1 \,\|\, 0\,0\,1\,0\,1 & & N_1 \; 1\,0\,1 \,\|\, \mathbf{1\,0\,0\,1\,1} \\ & \to & \\ E_2 \; 0\,1\,1 \,\|\, 1\,0\,0\,1\,1 & & N_2 \; 0\,1\,1 \,\|\, \mathbf{0\,0\,1\,0\,1} \end{array}$$

2-Punkt-Crossover z.B. (3,5)-Crossover

E_1	1 0 | 1 0 0 | 1 0 1		N_1	1 0 | **1 1 0** | 1 0 1
		$\rightarrow$		
E_2	0 1 | 1 1 0 | 0 1 1		N_2	0 1 | **1 0 0** | 0 1 1

$\Rightarrow$ Bilde eine neue Generation mit Hilfe der Fitness-Werte entweder nur aus Nachkommen oder aus Eltern und Nachkommen (z.B. gemäß der Fitness der Individuen).

Die Auswahl der Eltern ist häufig proportional zur Fitness („roulette wheel selection").

<u>Algorithmus GEN-ALG</u>

1. Setze Parameter Populationsgröße $POPSIZE$, maximale Generationenanzahl $MAXGEN$, Wkt. P_{CO} für die Anwendung von Crossover und Wkt. P_{MU} für die Anwendung einer Mutation;
2. Erzeuge die Startpopulation POP_0 mit $POPSIZE$ Individuen (Chromosomen);
3. Bestimme die Fitness aller Individuen;
4. $k := 0;$

   WHILE $k < MAXGEN$ DO

   BEGIN
5. $h := 0;$

   WHILE $h < POPSIZE$ DO

   BEGIN
6. Wähle zwei Eltern aus POP_k proportional ihrer Fitness-Werte zufällig aus;
7. Wende mit Wkt. P_{CO} auf die ausgewählten Eltern ein Crossover an;
8. Wende mit Wkt. P_{MU} auf die entstandenen Individuen eine Mutation an;
9. $h := h + 2;$

   END
10. $k := k + 1;$
11. Wähle aus den erzeugten Nachkommen (bzw. auch den Eltern) $POPSIZE$ Individuen der k-ten Generation POP_k aus (z.B. proportional ihrer Fitness-Werte);

   END

BEISPIEL 3: GENETISCHE OPERATOREN ✏

Weitere Metaheuristiken:

- Ameisenalgorithmen („ant colony optimization“): populationsorientiert, modellieren Verhalten von Ameisen auf der Wegsuche
- Partikelschwarmoptimierung: populationsorientiert, imitieren Verhalten von Vogel- oder Fischschwärmen
- Variable Nachbarschaftssuche: nutzt systematisch eine Reihe von Nachbarschaften, oft: $N^1 \subseteq N^2 \subseteq \ldots \subseteq N^k$

Übungsaufgaben

1. Betrachtet wird das folgende binäre Optimierungsproblem

$$4x_1 + x_2 - x_3 + 3x_4 + 6x_5 - x_6 + 7x_7 + 5x_8 \to \max!$$

u.d.N.

$$\begin{array}{rcrcrcrrrrrcl} x_1 & + & 2x_2 & + x_3 & - & x_4 & +8x_5 & +x_6 & +4x_7 & +5x_8 & \le & 12 \\ 3x_1 & + & x_2 & -x_3 & + & 6x_4 & +3x_5 & +x_6 & +3x_7 & +6x_8 & \le & 13 \end{array}$$

$$x_1, \ldots, x_8 \in \{0, 1\}$$

Die Startlösung sei jeweils $x = (0, 0, \ldots, 0)^T$.

a) Bestimmen Sie ein lokales Maximum bzgl. N_1 mittels iterativer Verbesserung nach dem Prinzip der größten Verbesserung.

b) Bestimmen Sie ein lokales Maximum bzgl. N_1 mittels iterativer Verbesserung nach dem Prinzip der ersten Verbesserung.

2. Gegeben sei das folgende ganzzahlige Optimierungsproblem:

$$3x_1 + x_2 + 4x_3 + 2x_4 \to \max!$$

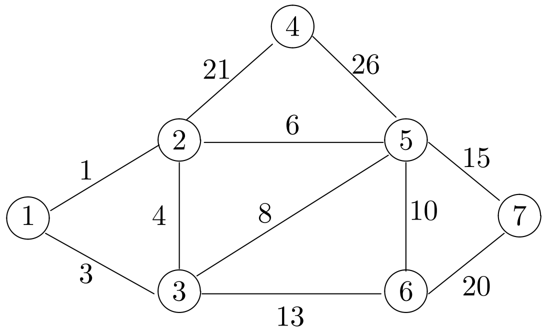
u.d.N.

$$\begin{array}{rcrcrcrcl} 4x_1 & + & x_2 & + & 3x_3 & + & 5x_4 & \le & 26 \\ x_1 & + & 2x_2 & + & 4x_3 & + & 3x_4 & \le & 20 \end{array}$$

$$x_1, \ldots, x_4 \in \mathbb{Z}_+$$

Die Startlösung sei $x = (2, 3, 1, 2)^T$. Bestimmen Sie die Anzahl der zulässigen Lösungen in der Nachbarschaft N_2 und den besten Nachbarn.

3. Gegeben sei der folgende Graph:



(a) Bestimmen Sie ein Minimalgerüst mittels Algorithmus von Kruskal.

(b) Gegeben sind zusätzlich folgende Restriktionen:

1) Kante [1,2] darf nur im Gerüst sein, wenn Kante [2,4] nicht im Gerüst ist.
2) Eine der Kanten [3,2], [3,5] oder [3,6] muss zum Gerüst gehören.
3) Die Kanten [5,7] und [3,6] gehören entweder beide zum Gerüst oder beide nicht zum Gerüst.

Bestätigen Sie, dass das Gerüst G beschreiben durch die Kanten [1,2], [1,3], [3,5], [4,5], [5,6], [6,7] zulässig ist und geben Sie das Gewicht an.

(c) Bestimmen Sie den besten Nachbarn von G in der Nachbarschaft N, wo zur Erzeugung eines Nachbarn genau eine Kante des Gerüsts durch eine andere Kante ersetzt wird.

4. Gegeben sei das folgende binäre Optimierungsproblem:

$$-5x_1 - 2x_2 - 6x_3 - 8x_4 - x_5 - 6x_6 - 4x_7 \to \min!$$

u.d.N.

$$\begin{array}{ccccccccccccccc} 3x_1 & + & 2x_2 & + & 4x_3 & + & 7x_4 & + & 3x_5 & + & 6x_6 & + & 2x_7 & \leq & 17 \\ 7x_1 & + & 3x_2 & + & 5x_3 & + & 2x_4 & + & 6x_5 & + & 2x_6 & + & 5x_7 & \leq & 18 \end{array}$$

$$x_1, \ldots, x_7 \in \{0, 1\}$$

Die Startlösung sei $x = (1, 0, 0, 1, 0, 0, 1)^T$.

(a) Sei $t_0 = 3$. Mit welcher Wahrscheinlichkeit wird der Nachbar aus N_3 bei Simulated Annealing akzeptiert, wenn die 2., 5. und 7. Komponente geändert werden?

(b) Sei $t_0 = 5$. Mit welcher Wahrscheinlichkeit wird der Nachbar aus N_4 bei Simulated Annealing akzeptiert, wenn die 1., 5., 6. und 7. Komponente geändert werden?

(c) Sei $t_0 = 0.1$. Mit welcher Wahrscheinlichkeit wird der Nachbar aus N_1 bei Simulated Annealing akzeptiert, wenn die 2. Komponente geändert wird?

5. Gegeben sei das folgende binäre Optimierungsproblem:

$$4x_1 + 2x_2 + x_3 + 5x_4 + 3x_5 \to \min!$$

u.d.N.

$$\begin{array}{ccccccccccc} 6x_1 & + & 3x_2 & + & 2x_3 & + & 3x_4 & + & 5x_5 & \geq & 12 \\ 3x_1 & + & 5x_2 & + & x_3 & + & 8x_4 & + & 4x_5 & \geq & 14 \end{array}$$

$$x_1, \ldots, x_5 \in \{0, 1\}$$

Die Startlösung sei $x = (1, 0, 1, 1, 1)^T$.

(a) Führen Sie zwei Iterationen von x mittels Tabu Suche (ohne Aspiration Kriterium) aus, wobei als Nachbarschaft N_2 gewählt wird und in jeder Iteration der beste Nachbar aus $Cand(x)$ als Startlösung für die nachfolgende Iteration verwendet wird.

(a) Führen Sie drei Iterationen von x mittels Tabu Suche (ohne Aspiration Kriterium) aus, wobei als Nachbarschaft N_1 gewählt wird und in jeder Iteration der beste Nachbar aus $Cand(x)$ als Startlösung für die nachfolgende Iteration verwendet wird.

6. Gegeben sei das folgende ganzzahlige (nichtlineare) Optimerungsproblem:

$$4x_1^2 + 8x_2 - x_2x_3 - x_3x_4 + 3\sqrt{x_4} + x_5^{3/2} + 2x_6 \to \max!$$

u.d.N.

$$\begin{array}{rcl} x_1 \cdot x_2 & \leq & 25 \\ x_3 + x_4 & \geq & 15 \\ x_5 + 2x_6 & \leq & 15 \end{array}$$

$$x_1, \ldots, x_6 \in \mathbb{Z}_+$$

In einem genetischen Algorithmus bezeichne das i-te Gen den Wert der Variablen x_i. Seien $x^1 = (3, 5, 6, 10, 7, 4)^T$ und $x^2 = (4, 6, 8, 9, 4, 3)^T$ die ausgewählten Eltern. Zur Erzeugung der Nachkommen wird zunächst ein (3,5)-Crossover und danach als Mutationen im ersten Chromosom die sechste Komponente um 1 vergrößert und im zweiten Chromosom die zweite Komponente um 1 verkleinert. Wählen Sie aus den zwei Eltern und den zwei erzeugten Nachkommen die zwei Chromosomen mit der besten Fitness aus.

Kapitel 4

Dynamische Optimierung

→ Es werden Probleme betrachtet, die in einzelne „Stufen“ zerlegt werden können, so dass die Gesamtoptimierung durch eine „stufenweise Optimierung“ ersetzt werden kann.

→ Häufig wird sie angewendet bei der optimalen Steuerung wirtschaftlicher Prozesse, wobei die Stufen einzelnen Zeitperioden entsprechen.

4.1 Einführungsbeispiele

4.1.1 Das Lagerhaltungsproblem

Problemstellung:

- Ein Gut werde während eines endlichen Planungszeitraumes, der aus n Perioden besteht, gelagert.
- In jeder Periode werde das Lager zu Beginn beliefert.
- In jeder Periode trete Nachfrage auf, die unmittelbar nach Belieferung in dieser Periode zu befriedigen ist.

Bezeichnungen:

$u_j \geq 0$ - die zu Beginn der Periode j gelieferte Menge

$r_j \geq 0$ - Nachfrage in Periode j

x_j - Lagerbestand unmittelbar vor Belieferung des Lagers in Periode j $(j = 1, 2, \ldots, n)$

Nebenbedingungen und Kostenbetrachtung ✏

Optimierungsproblem:

$$
\boxed{
\begin{aligned}
&\sum_{j=1}^{n} \left[K\delta(u_j) + hx_{j+1}\right] \to \min! \\
&\text{u.d.N.} \\
&x_{j+1} = x_j + u_j - r_j, \quad j = 1, 2, \dots, n \\
&x_1 = x_{n+1} = 0 \\
&x_j \geq 0, \qquad j = 2, 3, \dots, n \\
&u_j \geq 0, \qquad j = 1, 2, \dots, n
\end{aligned}
}
\tag{4.1}
$$

<u>Bemerkung:</u>

$$x_1 = x_{n+1} = 0 \text{ und } (4.1)$$

$\Rightarrow$ Ersetze in der Zielfunktion hx_{j+1} durch hx_j, damit jeder Summand die Form $g_j(x_j, u_j)$ hat.

$$x_j = x_{j+1} - u_j + r_j \geq 0 \quad \Rightarrow \quad u_j \leq x_{j+1} + r_j$$

Nebenbedingungen lassen sich wie folgt formulieren:

$$
\begin{aligned}
x_1 &= x_{n+1} = 0 \\
x_j &= x_{j+1} - u_j + r_j, \quad j = 1, 2, \dots, n \\
x_j &\geq 0, \qquad j = 1, 2, \dots, n \\
0 \leq u_j &\leq x_{j+1} + r_j, \quad j = 1, 2, \dots, n
\end{aligned}
$$

4.1.2 Das binäre Rucksackproblem

$$u_j := \begin{cases} 1, & \text{falls Gegenstand } j \text{ in den Rucksack gepackt wird} \\ 0, & \text{sonst} \end{cases}$$

<u>Optimierungsproblem:</u>

$$\sum_{j=1}^{n} c_j u_j \to \max!$$

u.d.N.

$$
\begin{aligned}
&\sum_{j=1}^{n} a_j u_j \leq V \\
&u_1, u_2, \dots, u_n \in \{0, 1\}
\end{aligned}
$$

→ Hier sind die Stufen keine Zeitperioden, sondern es erfolgt eine „künstliche Dynamisierung“. Die Entscheidungen, welche der Gegenstände $1, 2, \ldots, n$ in den Rucksack gepackt werden, werden als Entscheidungen in n aufeinanderfolgenden Stufen interpretiert.

x_j - Restvolumen des Rucksacks für die Gegenstände $j,\ j+1, \ldots, n$

$\Rightarrow x_1 = V \quad$ und $\quad x_{j+1} = x_j - a_j u_j \quad$ für alle $j = 1, 2, \ldots, n$

Umformuliertes Optimierungsproblem:

$$\sum_{j=1}^{n} c_j u_j \;\to \max!$$

u.d.N.

$$\begin{array}{lll}
x_{j+1} = x_j - a_j u_j, & & j = 1, 2, \ldots, n \\
x_1 = V & & \\
0 \le x_{j+1} \le V, & & j = 1, 2, \ldots, n \\
u_j \in \{0, 1\}, & \text{falls } x_j \ge a_j, & j = 1, 2, \ldots, n \\
u_j = 0, & \text{falls } x_j < a_j, & j = 1, 2, \ldots, n
\end{array}$$

4.2 Problemstellung

Dynamische Optimierungsprobleme betrachten einen endlichen Planungszeitraum, der in n Perioden oder Stufen eingeteilt ist.

Zustandsvariable x_j:

→ beschreibt Zustand des Systems zu Beginn der Periode j (bzw. am Ende der Periode $j-1$)

→ $x_1 := x_a$ - vorgegebener Anfangszustand des Systems

Entscheidungsvariable u_j:

→ In Periode 1 wird Entscheidung u_1 getroffen, die das System in den Zustand x_2 überführt, d.h.

$$x_2 = f_1(x_1, u_1),$$

wobei mit der Entscheidung u_1 die Kosten $g_1(x_1, u_1)$ verbunden sind.

allgemein:

$x_{j+1} = f_j(x_j, u_j)$	resultierender Zustand
$g_j(x_j, u_j)$	Stufenkosten
$X_{j+1} \neq \emptyset$	*Zustandsbereich*, der mögliche Zustände am Ende von Periode j enthält, wobei $X_1 = \{x_1\}$
$U_j(x_j) \neq \emptyset$	*Steuerbereich*, der mögliche Entscheidungen in Periode j enthält (hängt vom Zustand x_j zu Beginn von Periode j ab)

Illustration ✏

Optimierungsproblem:

$$\sum_{j=1}^{n} g_j(x_j, u_j) \to \min!$$

u.d.N.

$$\begin{aligned} & x_{j+1} = f_j(x_j, u_j), && j = 1, 2, \dots, n \\ & x_1 = x_a, && \\ & x_{j+1} \in X_{j+1}, && j = 1, 2, \dots, n \\ & u_j \in U_j(x_j), && j = 1, 2, \dots, n \end{aligned} \tag{3.2}$$

Bemerkung: Im Allgemeinen wächst der Rechenaufwand bei der Lösung dynamischer Optimierungsprobleme exponentiell mit der Dimension der Steuer- und Zustandsvariablen.

Definition 1:
Eine Folge von Entscheidungen $(u_1, u_2, \dots, u_n)$ wird *Politik* oder *Steuerung* genannt. Die zu einer gegebenen Politik $(u_1, u_2, \dots, u_n)$ gemäß

$$x_1 = x_a \text{ und } x_{j+1} = f_j(x_j, u_j) \qquad \text{für alle } j = 1, 2, \dots, n$$

sukzessiv zu berechnende Folge $(x_1, x_2, \dots, x_n, x_{n+1})$ heißt zugeordnete *Zustandsfolge*. Eine Politik oder Zustandsfolge, die den Nebenbedingungen von (3.2) genügt, heißt *zulässig*.

4.3 Bellmansche Funktionalgleichung und Bellmansches Optimalitätsprinzip

Gegeben sind g_j, f_j, X_{j+1} und U_j für alle $j = 1, 2, \dots, n$.

$\Rightarrow$ Optimierungsproblem hängt von x_1 ab, d.h. $P_1(x_1)$.

analog: $P_j(x_j)$ - Problem für die Perioden $j, j+1, \dots, n$ mit Anfangszustand x_j

Theorem 1: ***(Bellmansches Optimalitätsprinzip)***
Seien $(u_1^*, \dots, u_j^*, \dots, u_n^*)$ eine optimale Politik für das Problem $P_1(x_1)$ und x_j^* der Zustand zu Beginn von Periode j, dann ist $(u_j^*, \dots, u_n^*)$ eine optimale Politik für das Problem $P_j(x_j^*)$, d.h.:
Die Entscheidungen in den Perioden $j, \dots, n$ des n-periodigen Problems $P_1(x_1)$ sind (bei gegebenem Zustand x_j^*) unabhängig von den Entscheidungen in den Perioden $1, \dots, j-1$.

Illustration **Bellmansche Funktionalgleichungen:** ✏

1. Seien $v_j^*(x_j)$ die minimalen Kosten für das Problem $P_j(x_j)$. Die für $j = 1, 2, \ldots, n$ gültige Beziehung

$$\boxed{\begin{aligned} v_j^*(x_j) &= g_j(x_j, u_j^*) + v_{j+1}^*(x_{j+1}^*) \\ &= \min_{u_j \in U_j(x_j)} \Big\{ g_j(x_j, u_j) + v_{j+1}^*[f_j(x_j, u_j)] \Big\} \\ & x_j \in X_j \end{aligned}} \tag{4.3}$$

wird *Bellmansche Funktionalgleichung* (BFGL) genannt, wobei

$$v_{n+1}^*(x_{n+1}) = 0$$

für $x_{n+1} \in X_{n+1}$.

$\Rightarrow$ Funktion v_j^* lässt sich bei bekannten v_{j+1}^* berechnen.

2. BFGL lassen sich auch für die folgenden Fälle angeben:

   (a) $\sum\limits_{j=1}^{n} g_j(x_j, u_j) \ \rightarrow \max!$

   $\Rightarrow$ Ersetze in (3.3) min! durch max!

   (b) $\prod\limits_{j=1}^{n} g_j(x_j, u_j) \ \rightarrow \min!$

   $\Rightarrow$ BFGL:

   $$v_j^*(x_j) = \min_{u_j \in U_j(x_j)} \Big\{ g_j(x_j, u_j) \cdot v_{j+1}^* \big[f_j(x_j, u_j) \big] \Big\}$$

   wobei $v_{n+1}^*(x_{n+1}) := 1$ und

   $g_j(x_j, u_j) > 0$ für alle $x_j \in X_j, u_j \in U_j(x_j), j = 1, 2, \ldots, n$

   (c) $\max\limits_{1 \le j \le n} \big\{ g_j(x_j, u_j) \big\} \rightarrow \min!$

   $\Rightarrow$ BFGL:

   $$v_j^*(x_j) = \min_{u_j \in U_j(x_j)} \Big\{ \max\big\{ g_j(x_j, u_j); v_{j+1}^* \big[f_j(x_j, u_j) \big] \big\} \Big\}$$

   wobei $v_{n+1}^*(x_{n+1}) = 0$

4.4 Bellmansche Funktionalgleichungsmethode

$\Rightarrow$ sukzessive Auswertung von (3.3) für $j = n, n-1, \ldots, 1$ zur Berechnung von $v_j^*(x_j)$

Algorithmus DO

Schritt 1: Rückwärtsrechnung

(a) Setze $v_{n+1}^*(x_{n+1}) := 0$ für alle $x_{n+1} \in X_{n+1}$.

(b) Für $j = n, n-1, \ldots, 1$ führe aus:

für alle $x_j \in X_j$ bestimme $z_j^*(x_j)$ als Minimalstelle der Funktion

$$w_j(x_j, u_j) := g_j(x_j, u_j) + v_{j+1}^*\big[f_j(x_j, u_j)\big]$$

auf $U_j(x_j)$, d.h.

$$w_j(x_j, z_j^*(x_j)) = \min_{u_j \in U_j(x_j)} w_j(x_j, u_j) = v_j^*(x_j) \text{ für } x_j \in X_j$$

Schritt 2: Vorwärtsrechnung

(a) Setze $x_1^* := x_a$.

(b) Für $j = 1, 2, \ldots, n$ führe aus:

$$u_j^* := z_j^*(x_j),\ x_{j+1}^* := f_j(x_j^*, u_j^*)$$

$\Rightarrow$ $(u_1^*, u_2^*, \ldots, u_n^*)$ optimale Politik

$\Rightarrow$ $(x_1^*, x_2^*, \ldots, x_{n+1}^*)$ optimale Zustandsfolge für Problem $P_1(x_1^* = x_a)$

Illustration: Bestimmung von $z_j^*(x_j)$ ✏

Zusammenfassung DO (Dynamische Optimierung)

Phase 1: *Dekomposition*

Phase 2: *Rückwärtsrechnung*

Phase 3: *Vorwärtsrechnung*

Bemerkung: Lassen sich alle Gleichungen

$$x_{j+1} = f_j(x_j, u_j), \quad j = 1, 2, \ldots, n$$

eindeutig nach x_j auflösen, ist der Rechenverlauf umkehrbar, d.h. man kann zunächst eine Vorwärts- und dann eine Rückwärtsrechnung durchführen (z.B. Lagerhaltungsproblem aus 3.1.).

4.5 Beispiele und Anwendungen

4.5.1 Das binäre Rucksackproblem

Annahme: V, a_j, c_j - ganzzahlig

$$\begin{aligned} g_j(x_j,u_j) &= c_j u_j, & j &= 1,2,\dots,n \\ f_j(x_j,u_j) &= x_j - a_j u_j, & j &= 1,2,\dots,n \\ X_{j+1} &= \{0,1,\dots,V\} \end{aligned}$$

$$U_j(x_j) = \begin{cases} \{0,1\} & \text{für } x_j \geq a_j \\ 0 & \text{für } x_j < a_j \end{cases}, j = 1,2,\dots,n$$

BFGL:

$$v_j^*(x_j) = \max_{u_j \in U_j(x_j)} \bigl\{c_j u_j + v_{j+1}^*(x_j - a_j u_j)\bigr\}, \quad 1 \leq j \leq n$$

Rückwärtsrechnung:

$$v_n^*(x_n) = \begin{cases} c_n, & \text{falls } x_n \geq a_n \\ 0, & \text{sonst} \end{cases}$$

$$z_n^*(x_n) = \begin{cases} 1, & \text{falls } x_n \geq a_n \\ 0, & \text{sonst} \end{cases}$$

$j = n-1, n-2, \dots, 1$:

$$v_j^*(x_j) = \begin{cases} \max\bigl\{v_{j+1}^*(x_j);\ c_j + v_{j+1}^*(x_j - a_j)\bigr\}, & \text{falls } x_j \geq a_j \\ v_{j+1}^*(x_j), & \text{sonst} \end{cases}$$

$$z_j^*(x_j) = \begin{cases} 1, & \text{falls } v_j^*(x_j) > v_{j+1}^*(x_j) \\ 0, & \text{sonst} \end{cases}$$

$\Rightarrow v_1^*(V)$ - maximaler Wert der Rucksackfüllung

Vorwärtsrechnung:

$$\begin{aligned} x_1^* &:= V \\ u_j^* &:= z_j^*(x_j^*), & j &= 1,2,\dots,n \\ x_{j+1}^* &:= x_j^* - a_j u_j^*, & j &= 1,2,\dots,n \end{aligned}$$

Beispiel 1: Anwendung der dynamischen Optimierung zur Lösung des binären Rucksackproblems ✏

Illustration: Zustands- und Steuerbereiche ✏

4.5.2 Bestimmung eines kürzesten (längsten) Weges in einem Graphen

Illustration: Kürzeste-Wege-Problem ✏

Ziel: Bestimme einen kürzesten Weg vom Knoten (Ort) x_1 zum Knoten (Ort) x_{n+1}.

Seien:

$X_j = \{x_j^1, x_j^2, \dots, x_j^k\}$ - Menge aller Orte der Stufe j, $\quad 2 \leq j \leq n$

$X_1 = \{x_1\}, \quad X_{n+1} = \{x_{n+1}\}$

$U_j(x_j) = \{x_{j+1} \in X_{j+1} \mid \exists \text{ Bogen von } x_j \text{ nach } x_{j+1}\}, \quad j = 1, 2, \dots, n$

$v_j^*(x_j)$ - Länge eines kürzesten Weges vom Knoten $x_j \in X_j$ zum Knoten x_{n+1}

$g_j(x_j, u_j) = c_{x_j, u_j}$

$f_{j+1}(x_j, u_j) = u_j = x_{j+1}$

$z_j^*(x_j) = u_j = x_{j+1}$ falls x_{j+1} nächster Ort nach x_j auf einem kürzesten Weg von x_j nach x_{n+1} ist

BFGL:

$$v_n^*(x_n) = c_{x_n, x_{n+1}} \quad \text{für} \quad x_n \in X_n$$

$j = n-1, n-2, \dots, 1$:

$$v_j^*(x_j) = \min\Big\{c_{x_j, x_{j+1}} + v_{j+1}^*(x_{j+1}) \mid x_{j+1} \in X_{j+1}, \text{ so dass Bogen } (x_j, x_{j+1}) \text{ existiert}\Big\}$$

Beispiel 2: Kürzeste-Wege-Problem ✏

4.5.3 Personalzuordnung

Beispiel 3: Personalzuordnung

Drei Forschungsteams arbeiten an der Lösung einer Aufgabe. Die Wahrscheinlichkeit eines Misserfolges sei bei den einzelnen Teams mit 0,4, 0,6 und 0,8 gegeben. Zur Erhöhung der Erfolgsaussichten sollen zwei weitere Mitarbeiter eingesetzt werden, wobei die geschätzten Vorteile der erhöhten Mitarbeiterzahl durch die folgende Tabelle der Fehlschlagswahrscheinlichkeiten gegeben sind.

Mitarbeiterzahl	Team 1	Team 2	Team 3
0	0,4	0,6	0,8
1	0,2	0,4	0,5
2	0,15	0,2	0,3

Ziel: Setze die Mitarbeiter so ein, dass die Gesamtwahrscheinlichkeit eines Fehlschlags möglichst klein wird.

Annahme: Die Wahrscheinlichkeiten für einen Misserfolg der einzelnen Teams sind *unabhängig* voneinander.

Seien:

x_j - Anzahl der noch für die Teams $j, \ldots, 3$ zur Verfügung stehenden Mitarbeiter

$x_1 = 2$ und $x_4 = 0$

u_j - Anzahl der dem Team j zugeordneten Mitarbeiter

$W_j(u_j)$ - Fehlschlagswahrscheinlichkeit von Team j bei Einsatz von u_j zusätzlichen Mitarbeitern

Optimierungsproblem:

$$W(u_1) \cdot W(u_2) \cdot W(u_3) \ \to \text{min!}$$

u.d.N.

$$\begin{aligned} & u_1 + u_2 + u_3 = 2 \\ & u_j \in \{0,1,2\}, \quad j = 1,2,3 \\ & x_{j+1} = x_j - u_j, \ j = 1,2,3 \\ & x_1 = 2, \quad x_4 = 0 \end{aligned}$$

Bemerkung:
Die Funktionen $g_j(x_j, u_j) = W_j(u_j)$ hängen nur von den Entscheidungen u_j ab.

$v_j^*(x_j)$ - minimale Fehlschlagswahrscheinlichkeit, falls x_j Mitarbeiter für die Teams $j, \ldots, 3$ zur Verfügung stehen

BFGL:

$$v_j^*(x_j) = \min\left\{W_j(u_j) \cdot v_{j+1}^*(x_j - u_j) \ \mid \ u_j \in \{0, \ldots, x_j\}\right\}$$

wobei $v_{n+1}^*(0) = 1$

Anwendung der dynamischen Optimierung zur Lösung des Problems

4.5.4 Endlagerung eines Schadstoffes

Für die Endlagerung eines Schadstoffes werden 3 mögliche Deponien $D_1,\ D_2,\ D_3$ in Betracht gezogen. Die gesamte Schadenswirkung je Einheit des Schadstoffes wird für die Deponien durch die Kosten $a_1,\ a_2,\ a_3$ beschrieben.

Ziel: Aufteilung von K Mengeneinheiten des Schadstoffes auf die Deponien, so dass der maximale Schaden möglichst klein wird.

Seien:

x_j - die zur Verteilung auf $D_j, \ldots, D_3$ verbleibende Schadstoffmenge

$x_0 = K$ und $x_4 = 0$

u_j - die auf D_j gelagerte Schadstoffmenge

Optimierungsproblem:

$$\max\{a_1u_1, a_2u_2, a_3u_3\} \;\to \min!$$

u.d.N.

$$\begin{aligned}
&u_1 + u_2 + u_3 = K\\
&0 \le u_j \le x_j, \qquad j = 1,2,3\\
&x_{j+1} = x_j - u_j, \quad j = 1,2,3\\
&x_1 = K, \quad x_4 = 0
\end{aligned}$$

$v_j^*(x_j)$ - minimale Kosten bei Verteilung von x_j Einheiten auf die Deponien $D_j, \dots, D_3$

<u>BFGL:</u>

$$v_3^*(x_3) = a_3x_3 \quad (= a_3u_3)$$

$$v_j^*(x_j) = \min\left\{\max(a_ju_j;\ v_{j+1}^*(x_j - u_j)) \;\mid\; 0 \le u_j \le x_j\right\}, \qquad j = 1,2$$

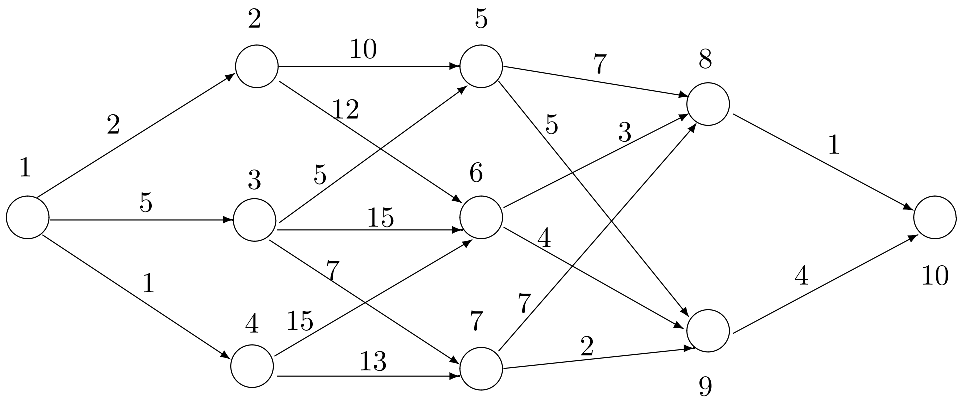
Übungsaufgaben

1. Betrachtet wird das Lagerhaltungsproblem aus Abschnitt 3.1, Teil (a), der Vorlesung. Bestimmen Sie eine optimale Lösung mittels dynamischer Optimierung für den Fall

$$n = 4; \quad K = 2; \quad h = 0,2; \quad r_1 = 3,\ r_2 = 2,\ r_3 = 3,\ r_4 = 2.$$

   Stellen Sie die Zustands- und Steuerbereiche grafisch dar.

2. Um von Kansas City (Knoten 1) nach San Francisco (Knoten 10) zu gelangen, gab es zur Goldgräberzeit die folgenden Verbindungen mit der Post- kutsche:



   Für jeden Fall waren vier Teilstrecken zu überwinden, auf denen man auch Überfällen durch Indianer oder Banditen ausgesetzt war. Ein guter Maßstab für die Gefährlichkeit dieser Teilstrecken waren die Versicherungs- prämien, die für eine Versicherung gegen solche Überfälle erhoben wurden. Für die Strecke von i nach j seien sie c_{ij} (siehe Skizze). Bestimmen Sie den gefahrlosesten Weg von Kansas City nach San Francisco mittels dynamischer Optimierung.

3. Lösen Sie das folgende (binäre) Rucksackproblem mittels dynamischer Optimierung:

$$\begin{array}{ccccccccccccc} 8x_1 & + & 8x_2 & + & 6x_3 & + & 10x_4 & + & 12x_5 & + & 12x_6 & \to & \max! \\ x_1 & + & 2x_2 & + & 2x_3 & + & 4x_4 & + & 6x_5 & + & 10x_6 & \leq & 12 \end{array}$$

$$x_1, x_2, \ldots, x_6 \in \{0, 1\}$$

   Stellen Sie die Zustands- und Steuerbereiche grafisch dar.

4. Auf einer Maschine wird ein bestimmtes Erzeugnis hergestellt. In Abhängig- keit vom Alter a der Maschine (gemessen in Jahren) ergibt sich ein Jahresausstoß von $100 - 25a$. Der Preis je Erzeugniseinheit beträgt 600 EUR, eine neue Maschine kostet 20 000 EUR. Zu Beginn des Planungszeitraumes steht eine Maschine des Alters $a = 1$ zur Verfügung. Für die ersten drei Jahre des Planungszeitraumes soll am Anfang jedes Jahres festgelegt werden, ob die Maschine des Vorjahres weiterhin eingesetzt oder durch eine neue ersetzt wird. Dabei wird ein maximales Ergebnis (Preissumme vermindert um Kosten für neue Maschinen) angestrebt. Bestimmen Sie eine optimale Lösung mittels dynamischer Optimierung.

   Stellen Sie die Zustands- und Steuerbereiche grafisch dar.

Kapitel 5

Warteschlangen

5.1 Charakterisierung von Wartesystemen

prinzipieller Aufbau eines Wartesystems:

Input-Quelle (Kundenquelle)	$\longrightarrow$ *ankommende Elemente (Kunden)*	**Wartesystem**: Warteschlange(n) $\longrightarrow$ Ressourcen (Bedienungsstationen)	$\longrightarrow$ *abgefertigte Elemente (Kunden)*

Einige Beispiele:

ankommende Elemente (*Input*)	Warteschlange	Ressource	abgefertigte Elemente (*Output*)
ankommende Telefongespräche	Gespräche in Leitung	Telefonleitung (Zentrale)	hergestellte Verbindungen
ankommende Autos	Autoschlange (Verkehrsstockung)	Kreuzung (Ampel)	abfahrende Autos
ankommende Autos	Autoschlange	Tankstelle	betankte Autos
Maschinendefekte	zu reparierende Maschinen	Reparatur (Monteur)	intakte Maschinen
Fahrgäste	Fahrgastschlange	Taxis	Fahrgäste auf Fahrt zum Ziel

Die Warteschlangentheorie nutzt stochastische Prozesse:

- Ankunftsprozess
- Bedienungsprozess

abgeleitete stochastische Prozesse:

→ *Warteschlangenprozess:* Zahl der Kunden im System

→ *Wartezeitprozess:* Zeit von Ankunft eines Kunden bis zur Bedienung

→ *Output-Prozess:* Abstand zwischen dem Abschluss zweier aufeinanderfolgender Bedienungen

→ *Betriebsperiode:* Zeit zwischen zwei aufeinanderfolgenden Leerzeiten des Bedienungskanals

Annahmen und Bezeichnungen

Wartesystem zur Zeit 0 leer

$T_1 < T_2 < \ldots$ - Zeitpunkte, zu denen Kunden einzeln an der Bedienungsstation ankommen

$Z_n = T_n - T_{n-1}, \quad n = 1, 2, \ldots, \quad T_0 := 0$ - Zwischenankunftszeit des n-ten Kunden

S_n - Bedienungs- oder Servicezeit des n-ten Kunden

W_n^q - Wartezeit des n-ten Kunden in der Schlange

$W_n := W_n^q + S_n$ - Verweilzeit des n-ten Kunden

$D_n = T_n + W_n^q + S_n$ - n-ter Kunde verlässt Wartesystem

$\mathcal{V}(t)$ - virtuelle Wartezeit: Wartezeit eines Kunden in der Schlange, der zum Zeitpunkt t ankommen würde, d.h. $\mathcal{V}(T_n) = W_n^q$

$\mathcal{L}(t)$ - Anzahl der Kunden im Wartesystem zur Zeit t

$\mathcal{L}^q(t)$ - Anzahl der Kunden in der Schlange zur Zeit t

$\Rightarrow$ $T_n,\ Z_n,\ S_n,\ W_n^q,\ W_n,\ \mathcal{V}(t),\ \mathcal{L}(t),\ \mathcal{L}^q(t)$ sind *Zufallsgrößen*!

Annahmen über Zufallsgrößen:

Z_n unabhängig, identisch verteilt $\Rightarrow$ Zufallsgröße Z (*Zwischenankunftszeit*)

S_n unabhängig, identisch verteilt $\Rightarrow$ Zufallsgröße S (*Bedienungszeit*)

Z und S unabhängig voneinander

Illustration: Realisation von $\mathcal{L}(t)$ ✏

Charakteristika von Wartesystemen:

- Größe des Warteraums:
  - endlich
  - unendlich
- Anzahl der Bedienungsschalter
- Abfertigungsmodus:
  - einzeln

- schubweise
- Warteschlangendisziplin:
  - *first come first served (FCFS)*
  - last come first served
  - zufällige Auswahl

Beschreibung von Wartesystemen

3-Tupel $x \mid y \mid z$

x: Verteilung der Zwischenankunftszeit

y: Verteilung der Bedienungszeit

z: Anzahl der parallelen (identischen) Bedienungsschalter

Ausprägungen von x und y:

M (Markov)	Exponentialverteilung mit Dichtefunktion $f(t) = \alpha e^{-\alpha t}, \quad t \geq 0, \ \alpha > 0$
E_k	Erlangverteilung mit Phasenparameter k und Dichtefunktion
	$f_k(t) = \frac{b^{k+1} t^k e^{-bt}}{k!}, \quad t, b \geq 0, \ k = 0, 1, 2, \ldots$
G (general)	beliebige Verteilung
D	Verteilung deterministischer Größe, z.B. $P(t = a) = 1$

Beispiel: M | M | 1 $\rightarrow Z, S$ exponentialverteilte Zufallsgrößen, 1 Bedienungsschalter

Bei Beschränkungen der Kanäle oder der Zahl der zu berücksichtigenden Input-Elemente nutze:

$$x \mid y \mid z \mid a \mid b,$$

wobei:

a - *begrenzte Kapazität des Systems* (Schlangenkapazität und 1 Element pro Kanal)

b - *Zahl der (endlich vielen) Input-Elemente*

Beispiel: M | M | s | | K

$\rightarrow Z, S$ exponentialverteilte Zufallsgrößen

s Bedienungsschalter

K relevante Input-Elemente

5.2 Das System M|M|1

Annahmen:

1. Z ist exponentialverteilt.

   $P(Z \leq t) = 1 - e^{-\alpha t}$ Verteilungsfunktion

   $f_Z(t) = \alpha e^{-\alpha t}$ Dichtefunktion

   $E(Z) = \frac{1}{\alpha}$ Erwartungswert von Z

   α - Ankunftsrate

   $\Rightarrow$ Ankünfte können durch einen Poisson-Prozess mit dem Parameter α beschrieben werden.

2. S ist exponentialverteilt.

   $P(S \leq t) = 1 - e^{-\beta t}$ Verteilungsfunktion

   $f_S(t) = \beta e^{-\beta t}$ Dichtefunktion

   $E(S) = \frac{1}{\beta}$ Erwartungswert von S

   β - Bedienungsrate

3. unbegrenzter Warteraum, Kunden werden gemäß FCFS bedient

Ereignis: Eintreffen oder Abfertigen eines Kunden

$$\boxed{\varrho := \frac{\alpha}{\beta} \quad \text{Verkehrsintensität}}$$

Zur Anzahl der Kunden im System

Zustandswahrscheinlichkeit:

$$P_n(t) = P(\mathcal{L}(t) = n)$$

Konvergenzbedingung

$\alpha < \beta$ ($\varrho < 1$)

$\Rightarrow$ Zustandwahrscheinlichkeit $P_n(t)$ konvergieren für $t \to \infty$ gegen stationäre Werte P_n

Rekursive Lösung der stationären Punkte liefert:

$$P_n = \left(\frac{\alpha}{\beta}\right)^n \cdot P_0 = \varrho^n \cdot P_0$$

außerdem gilt:

$$\sum_{n=0}^{\infty} P_n = \sum_{n=0}^{\infty} \varrho^n P_0 = \frac{1}{1-\varrho} P_0 = 1$$

$$\Rightarrow P_0 = 1-\varrho, \qquad P_n = \varrho^n (1-\varrho) : \qquad \text{geometrisch verteilt mit Parameter } \varrho = \frac{\alpha}{\beta}$$

Beispiel 1: M|M|1 ✏

An der Kasse eines Supermarktes werden durchschnittlich 25 Kunden pro Stunde bedient, und die durchschnittliche Bedienungsdauer pro Kunde beträgt 2 Minuten.
Annahmen: Z, S exponentialverteilt, System im stationären Zustand (*Gleichgewichtsfall*)

(a) Wie groß ist der Anteil der Leerzeiten der Kassiererin?

(b) Wie groß ist die Wahrscheinlichkeit, dass mehr als 5 Kunden an der Kasse stehen?

Mittlere Anzahl der Kunden im System L/ Mittlere Schlangenlänge L^q

$$L := E(\mathcal{L}) = \sum_{n=0}^{\infty} nP_n = \sum_{n=0}^{\infty} n(1-\varrho)\varrho^n = \frac{\varrho}{1-\varrho}$$

$$L = \frac{\alpha}{\beta-\alpha} \tag{5.1}$$

$$L^q = E(\mathcal{L}^q) = \sum_{n=1}^{\infty} (n-1)P_n = L - (1-P_0) = \frac{\varrho^2}{1-\varrho}$$

$$L^q = \frac{\alpha^2}{\beta(\beta-\alpha)} = L - \frac{\alpha}{\beta}$$

Mittlere Warte- und Verweilzeit der Kunden

Sei $F(\cdot, t)$ die Verteilungsfunktion der virtuellen Wartezeit $\mathcal{V}(t)$

$$\Rightarrow \quad F(v,t) = P(\mathcal{V}(t) \le v) = \sum_{n=0}^{\infty} P(\mathcal{V}(t) \le v \mid \mathcal{L}(t) = n) \cdot P(\mathcal{L}(t) = n)$$

$$\Rightarrow \quad \lim_{t\to\infty} F(v,t) := F_q(v) = \begin{cases} 1 - \left(\frac{\alpha}{\beta}\right) \cdot e^{-(\beta-\alpha)v} & \text{, falls } v \ge 0 \\ 0 \text{ ,} & \text{falls } v < 0 \end{cases}$$

Illustration: F_q ✏

$$W^q := E(\mathcal{W}^q) = \frac{\alpha}{\beta(\beta-\alpha)}$$

$\to W^q$: mittlere Wartezeit eines Kunden in der Schlange im Gleichgewichtsfall analog:

$$\text{Sei } F(w) = P(\mathcal{W} < w) = \begin{cases} 1 - e^{-(\beta-\alpha)w}, & \text{falls } w \geq 0 \\ 0, & \text{falls } w < 0 \end{cases}$$

$$W = E(\mathcal{W}) = W^q + \frac{1}{\beta} = \frac{1}{\beta - \alpha} \tag{5.2}$$

$\rightarrow W$: mittlere Verweilzeit eines Kunden im Gleichgewichtsfall

Aus Beziehung (5.1) und (5.2) folgt:

$$\boxed{\begin{array}{c} L = \alpha \cdot W \\ \text{analog: } L^q = \alpha \cdot W^q \\ \textbf{Little's Formel} \end{array}}$$

Beispiel 2: M|M|1 ✏

Betrachtet wird Beispiel 1.

(a) Wie lange muss ein Kunde durchschnittlich an der Kasse warten?

(b) Wie viele Kunden müssen länger als 10 Minuten verweilen?

5.3 Das System M|M|1|K mit endlichem Warteraum

- maximal K Kunden im System ($K-1$ Kunden in der Warteschlange)
- Gleichgewichtsfall

$$P_n = \begin{cases} \dfrac{1}{1+K} & \text{für } \varrho = 1 \\ \dfrac{(1-\varrho)\varrho^n}{1-\varrho^{K+1}} & \text{für } \varrho \neq 1 \end{cases}$$
$$n = 0, 1, \ldots, K$$

- Sei γ der *Erfassungsgrad der Kunden* (Anteil der Kunden, die sich in die Warteschlange einreihen)

$$\boxed{\Rightarrow \quad \gamma = 1 - P_K}$$

mittlere Anzahl der Kunden im System:

$$L := E(\mathcal{L}) = \frac{\varrho[1 - (K+1)\varrho^K + K\varrho^{K+1}]}{(1-\varrho^{K+1})(1-\varrho)} \quad (\varrho \neq 1)$$

mittlere Anzahl der Kunden in der Warteschlange:

$$L^q = L - \frac{\alpha\gamma}{\beta}$$

mittlere Verweilzeit im System:

$$W = \frac{L}{\alpha\gamma} = \frac{L}{\alpha(1-P_K)} \qquad \text{(Little's Formel)}$$

mittlere Wartezeit in der Schlange:

$$W^q = W - \frac{1}{\beta}$$

Beispiel 3: M|M|1|K ✏

Bei Dr. Dent treffen durchschnittlich 5 Patienten pro Stunde ein, und er kann durchschnittlich 6 Patienten pro Stunde behandeln.

(a) Es liege ein M|M|1 System vor. Wie groß ist die erwartete Anzahl von Patienten in der Praxis? Wie lange muss ein Patient durchschnittlich warten? Welcher Anteil von Patienten muss nicht warten?

(b) Im Wartesystem befinden sich 4 Stühle, wobei ein Stuhl im Behandlungszimmer und 3 Stühle im Wartezimmer stehen. Es wird angenommen, dass Patienten nicht warten, wenn alle Stühle besetzt sind. Wie groß sind die erwartete Anzahl Patienten in der Praxis und die durchschnittliche Wartezeit für dieses System? Wie viele Patienten verliert Dr. Dent pro Stunde durch die Begrenzung des Warteraumes?

5.4 Das System M|M|s

- $s > 1$ parallele, identische Bedienungsschalter
- Gleichgewichtsfall
- Konvergenzbedingung: $\varrho = \frac{\alpha}{\beta} < s$

$$P_n = \begin{cases} \dfrac{\varrho^n}{n!} \cdot P_0 & \text{für } n = 1, 2, \ldots, s-1 \\ \dfrac{\varrho^n}{s! \cdot s^{n-s}} \cdot P_0 & \text{für } n \geq s \end{cases}$$

mit

$$P_0 = \frac{1}{1 + \sum_{j=1}^{s-1} \frac{\varrho^j}{j!} + \frac{\varrho^s}{(s-\varrho)(s-1)!}}$$

mittlere Anzahl der Kunden in der Warteschlange:

$$L^q = \frac{\varrho^{s+1}}{(s-\varrho)^2(s-1)!} \cdot P_0$$

mittlere Anzahl der Kunden im System:

$$L = L^q + \varrho$$

mittlere Wartezeit in der Schlange:

$$W^q = \frac{L^q}{\alpha}$$

mittlere Verweilzeit im System

$$W = \frac{L}{\alpha} = \frac{L^q + \frac{\alpha}{\beta}}{\alpha} = W^q + \frac{1}{\beta}$$

BEISPIEL 4: M|M|S ✏

In einer Telefonzelle werden im Mittel 10 Gespräche pro Stunde geführt, deren erwartete Dauer jeweils 5 Minuten sei. Die zumutbare mittlere Wartezeit vor der Zelle sei 5 Minuten. Wie viele Telefonzellen sind erforderlich? (Annahmen: Z, S exponentialverteilt; Gleichgewichtsfall)

5.5 Das System M|M|s|K mit endlichem Warteraum

- im Wartesystem haben höchstens $K \geq s$ Kunden Platz
- Gleichgewichtsfall

$$P_n = \begin{cases} \frac{\varrho^n}{n!} \cdot P_0 & \text{für } n = 1, 2, \ldots, s-1 \\ \frac{\varrho^n}{s! \cdot s^{n-s}} \cdot P_0 & \text{für } n = s, s+1, \ldots, K \\ 0 & \text{für } n > K \end{cases}$$

mit

$$P_0 = \frac{1}{1 + \sum\limits_{j=1}^{s-1} \frac{\varrho^j}{j!} + \frac{\varrho^s}{s!} \cdot \sum\limits_{j=s}^{K} (\frac{\varrho}{s})^{j-s}}$$

mittlere Anzahl der Kunden in der Warteschlange:

$$L^q = \frac{\varrho^{s+1} P_0}{(s-\varrho)^2 (s-1)!} \cdot \left[1 - (\frac{\varrho}{s})^{K-s} \cdot \big(1 + \frac{(s-\varrho)(K-s)}{s}\big)\right] \qquad (\varrho \neq s)$$

mittlere Wartezeit in der Schlange:

$$W^q = \frac{L^q}{\alpha \cdot \gamma}, \qquad \alpha \cdot \gamma = \alpha_{eff} \qquad \text{mit Erfassungsgrad } \ \gamma = 1 - P_K$$

mittlere Verweilzeit im System:

$$W = W^q + \frac{1}{\beta}$$

mittlere Anzahl der Kunden im System:

$$L = \alpha_{eff} W = L^q + \frac{\alpha_{eff}}{\beta}$$

BEISPIEL 5: M|M|S|K ✏

Ein Reisebüro habe 2 Angestellte, die Anrufe beantworten, und ein Anrufer kann zusätzlich in der „Warteleitung" bleiben. Sind alle 3 Leitungen belegt, hört der Anrufer das Besetztzeichen. Es seien $\alpha = 1$ und $\beta = 2$ (pro Minute) sowie Z, S exponentialverteilt. Gesucht sind die stationären Wahrscheinlichkeiten, dass

(a) ein Anrufer sofort mit einem Angestellten sprechen kann;

(b) ein Anrufer zunächst in der Warteleitung bleibt;

(c) ein Anrufer das Besetztzeichen hört?

Übungsaufgaben

1. Betrachtet wird das Beispiel 3 aus Abschnitt 4.3 der Vorlesung. Angenommen, es sei jetzt $\alpha = 4$ und $\beta = 5$ (pro Stunde). Im System sind 5 Sessel vorhanden (1 Behandlungssessel und 4 Sessel im Wartezimmer).

   (a) Bestimmen Sie für den Fall eines Wartesystems $M|M|1$ die Werte L^q, L, W^q, W, die Wahrscheinlichkeit P_0 sowie die Wahrscheinlichkeit, dass ein Patient stehen muss.

   (b) Bestimmen Sie für den Fall eines Wartesystems $M|M|1|5$ die Werte L^q, L, W^q und W sowie den Erfassungsgrad γ.

2. In einer Werkstatt treffen Aufträge gemäß einem Poissonschen Ankunfts- strom mit einer Rate von 3 pro Tag ein. Die Bearbeitungszeit für einen Auftrag sei exponentialverteilt mit einer Rate von 6 pro Tag. In der Werkstatt können neben dem gerade in Arbeit befindlichen Auftrag zwei wei- tere Werkstücke gelagert werden, der Rest kann außerhalb der Werkstatt gelagert werden.

   (a) Wie groß ist der Anteil der Zeit, während der die Lagerkapazität der Werkstatt ausreicht?

   (b) Angenommen, es stehen keine Lagermöglichkeiten außerhalb der Werkstatt zur Verfügung. Bestimmen Sie für diesen Fall die erwartete Anzahl der Werkstücke in der Werkstatt und den Anteil der angenommenen Aufträge.

3. Betrachtet wird das Beispiel 1 aus diesem Kapitel. Angenommen, es wird eine zweite Kasse eingerichtet, wobei beide Kassen mit der gleichen Geschwindigkeit arbeiten und es gelingt, den Zugang zu den Kassen so zu organisieren, dass sich nur eine gemeinsame Schlange vor den Kassen bildet. Bestimmen Sie für diesen Fall die mittlere Schlangenlänge und die mittlere Wartezeit der Kunden vor der Kasse.

4. Eine Autovermietung hat zu entscheiden, in welcher von zwei möglichen Reparaturwerkstätten ihre Wagen gewartet werden sollen. Sie schätzt, dass alle 40 Minuten ein Wagen zur Wartung eintrifft. In der ersten Werkstatt bestehen zwei parallele Wartungsstationen, von denen jede durchschnittlich 30 Minuten pro Wartung benötigt. In der zweiten Werkstatt besteht eine modernere Wartungsstation mit einer durchschnittlichen Abfertigungszeit von 15 Minuten pro Wagen.

   Angenommen, jede Minute, die ein Wagen in der Wartung verbringt, verringert den Gewinn der Autovermietung um eine Geldeinheit. Seien C_1 und C_2 die Kosten pro Minute Wartungszeit in den Werkstätten 1 bzw. 2. Bestimmen Sie den Kostenunterschied, bei dem die Vermietung in beiden Werkstätten gleich gut bedient wäre.

5. Die Datenverarbeitungsabteilung einer mittleren Firma kann s Standleitungen zu einem kommerziellem Rechenzentrum anmieten, in dem sie die meisten ihrer Aufgaben erledigen lässt. Die Aufgaben fallen nach einer Poisson-Verteilung mit einem Erwartungswert von 10 Minuten an, und die Länge der Übertragungszeiten sei exponentiell mit dem Erwartungswert 15 Minuten verteilt. Bestimmen Sie für die Fälle $s \in \{3, 4\}$ (d.h. für ein $M|M|3$ sowie ein $M|M|4$ Wartesystem) jeweils die erwartete Schlangenlänge sowie die mittlere Wartezeit in der Schlange.

Kapitel 6

Simulation

6.1 Grundbegriffe und Beispiele

Simulation: Nachbilden von Prozessen realer Systeme in einem Modell und anschließendes Durchführen von Experimenten am Modell

Eingangsgrößen	$\longrightarrow$ $\vdots$ $\longrightarrow$	$\boxed{\textbf{Simulationsmodell}}$	$\longrightarrow$ $\vdots$ $\longrightarrow$	*Ausgangsgrößen*

Simulation eines Wartesystems:

variable Eingangsgrößen:	Ankunftszeit der Kunden
Parameter:	mittlere Anzahl ankommender Kunden pro Zeiteinheit
	mittlere Bedienungszeit
Ausgangsgrößen:	mittlere Warteschlangenlänge
	maximale Bedienungszeit eines Kunden

→ oft Nutzung der *Digitalsimulation* (stetige Abläufe oder Größen werden durch diskrete Abläufe/Größen approximiert)

Arten der Simulation

deterministische Simulation	Eingangsgrößen legen Ausgangsgrößen eindeutig fest (deterministisches Modell)
stochastische Simulation	Zufallseinflüsse werden über Zufallsvariable erfasst (stochastisches Modell)

Stufen einer Simulationsstudie

- *Problemformulierung*
- *Entwicklung eines Simulationsmodells*

- *Datenerhebung und Datengenerierung*

  (1) Welche Verteilungsgesetze gelten?

  (2) Wie werden Realisationen von Zufallsgrößen erzeugt?

  zu (1): Nutze Methoden zur Ermittlung von Konfidenzintervallen von Verteilungsparametern und Tests zum Überprüfen von Hypothesen über die Lage von Verteilungsparametern.

  zu (2): siehe Abschnitt 5.2

- *Erstellung eines Computerprogramms*
  (auch Nutzung kommerzieller Simulations-Software, siehe Abschnitt 5.3)

- *Modellvalidierung*
  (Überprüfung der Gültigkeit des Modells mit untersuchtem Realitätsausschnitt)

- *Planung und Durchführung von Simulationsläufen*

  (1) Wie ist der Anfangszustand für die Simulationsläufe zu wählen?

  (2) Wie lässt sich der Simulationsumfang bestimmen?

  zu (1): Verwendung eines 'typischen' Anfangszustandes oder des 'Leerzustandes'

  zu (2): z.B. bezogen auf den Mittelwert einer abhängigen Zufallsgröße:

  - Schätzung des Mittelwertes
  - Konfidenzintervall für den Mittelwert
  - Ermittlung des Stichprobenumfangs bei gegebener Genauigkeit
    (vorgegebene Irrtumswahrscheinlichkeit)

- *Auswertung der Simulationsstudie*

6.2 Einige Bemerkungen zur Nutzung von Simulationssoftware

- **Nutzung von „Spreadsheets“**
- **EXCEL Queueing Simulator** für Warteschlangen

  Eingabe:

  - Anzahl der Bedienungsstationen
  - Zwischenankunftszeit T

    z.B. Exponentialverteilung (Erwartungswert), Erlangverteilung (Erwartungswert, k), $[a, b]$-Gleichverteilung (Parameter a, b)

  - Bedienungszeit S analog zu T
  - Simulationsumfang

  Ausgabe:

  $L,\ L^q,\ W,\ W^q,\ P_0,\ P_1, \ldots,\ P_{10}$

  als Punktschätzung und als 95% Konfidenzintervall

→ *sehr einfache Nutzung*

- **Crystal Ball** 2000.5 Studentenversion (140 Tage)

  4 Schritte:

  – Definition der zufälligen Eingangsgrößen (z.B. 17 verschiedene Verteilungsfunktionen)
  – Definition der Ausgangsgrößen
  – Setzen von Präferenzen (z.B. Simulationsumfang)
  – Ausführen der Simulation

  weitere Erläuterungen: siehe Hillier/Lieberman, Abschnitt 20.6 bzw.

  http://www.oracle.com/appserver/business-intelligence/crystalball/index.html

- **RiskSim**
  – akademische Version: siehe CD-ROM in Hillier/Lieberman (Shareware)
  – nicht so vielseitig wie Crystal Ball, aber einfach zu nutzen

Übungsaufgaben

1. Betrachtet wird die Simulation des Maschinenbelegungsproblems aus Beispiel 1 dieses Kapitels. Ermitteln Sie die Gesamtdurchlaufzeit in tabellarischer Form für den Fall, dass jeweils der Auftrag als nächster eingelastet wird, der die kürzeste Bearbeitungszeit auf M_1 hat.

2. Betrachtet wird die Simulation eines Warteschlangenproblems aus Beispiel 2 dieses Kapitels. In einem Simulationslauf mit $I = 25$ wurden folgende Zwischenankunftszeiten a_i $(i = 1, 2, \ldots, 25)$ ermittelt:

   $$6, 3, 4, 4, 2, 3, 4, 3, 2, 6, 4, 3, 6, 3, 5, 5, 3, 2, 2, 5, 2, 6, 5, 3, 3.$$

   Außerdem wurden folgende Bedienungszeiten b_i $(i = 1, 2, \ldots, 25)$ ermittelt:

   $$1, 5, 3, 5, 6, 4, 4, 2, 1, 1, 4, 4, 4, 1, 3, 1, 5, 5, 6, 5, 2, 6, 6, 1, 5.$$

   Bestimmen Sie die Auslastung ϱ der Bedienungsperson, die mittlere Wartezeit pro Kunde $\overline{w}$ sowie die mittlere Schlangenlänge $\overline{s}$.

3. In einer Arztpraxis in einer Kleinstadt sind die Zwischenankunftszeiten der Patienten und die Behandlungszeiten exponentialverteilt. In einer Stunde treffen durchschnittlich 4 Patienten in der Praxis ein, und die Behandlung eines Patienten dauert durchschnittlich 12 Minuten.

   (a) Bestimmen Sie mittels einer Simulationssoftware (z.B. Excel Queueing Simulator von der CD-ROM aus Hillier/Lieberman) die Werte $L^q, L,$
   W^q, W sowie die Wahrscheinlichkeit, dass kein Patient beim Arzt ist. Verwenden Sie einen Simulationsumfang von $n = 300$.

   (b) Vergleichen Sie die erhaltenen Ergebnisse mit den theoretischen Werten für den Gleichgewichtsfall.

4. In einer Postfiliale sind 2 Schalter geöffnet. Es wird geschätzt, dass durchschnittlich pro Stunde 24 Kunden eintreffen und die Bedienung eines Kunden durchschnittlich 4 Minuten dauert. Es liege ein $M|M|2$ Wartesystem vor.

   (a) Unter Nutzung einer Simulationssoftware berechnen Sie für einen Simulationsumfang von $n_1 = 200$ die Größen L^q, L, W^q, W sowie die Zustandswahrscheinlichkeiten P_0 und P_1.

   (b) Wiederholen Sie Ihre Berechnungen unter den gleichen Voraussetzungen für einen Simulationsumfang von $n_2 = 20000$.

   (c) Berechnen Sie die theoretischen Werte für den Gleichgewichtsfall.

   (d) Seien jetzt die Zwischenankunftszeit Z eine $[2, 6]$-gleichverteilte Zufallsgröße und die Bedienungszeit S eine $[1, 6]$-gleichverteilte Zufallsgröße. Führen Sie erneut eine Simulation mit $n_2 = 20000$ aus und vergleichen Sie die Ergebnisse mit (b).

   (e) Betrachten Sie jetzt ein $M|M|3$ System (d.h. es sind drei Schalter geöffnet) und bestimmen Sie mittels Software die Ergebnisse wiederum für $n_2 = 20000$.