

Einführung in Operations Research für Wi-Ing's

Einführung:

1. Was ist OR?

OR: quantitativ interpretierbare Entscheidungsprobleme (EP)

2. Wo liegt der Ursprung von OR?

Während des 2. Weltkriegs fingen die Engländer an, militärische Verteilungsprobleme mit Hilfe wissenschaftlicher Methoden zu lösen bzw. ihre Effektivität zu erhöhen.

Bsp: Seeminen, Flakstellungen, Nahrungsverteilung

Danach auch Anwendung in der zivilen Nutzung, vor allem durch die Amerikaner vorangetrieben.

3. Entscheidungsprobleme

EP: beste Entscheidungsalternative auswählen unter Beachtung von Ziel und Beschränkungen.

4. Allgemeines mathematisches Entscheidungsproblem

1. Zielfunktion: \min oder $\max z = f(x_1, \dots, x_j, \dots, x_n)$
 2. Restriktionen: $g_j(x_1, \dots, x_j, \dots, x_n) = b_i \quad (i = 1, \dots, m)$
 3. Nichtnegativitätsbedingungen: $x_j \geq 0 \quad (j = 1, \dots, n)$
- $[(x_1, \dots, x_j, \dots, x_n): \text{Entscheidungsalternative}]$
 $[b_i = \text{Beschränkungskonstante}]$

5. Zulässigkeit und Optimalität

$(x_1, \dots, x_j, \dots, x_n)$ ist zulässig, wenn 4.2. und 4.3 erfüllt sind.

$(x_1, \dots, x_j, \dots, x_n)$ ist zulässig und optimal, wenn 4.2. und 4.3. erfüllt sind und die Zielfunktion minimiert bzw. maximiert wird.

- Wir werden zulässige und optimale Lösungen suchen.
- Dazu gibt es verschiedene Algorithmen.

6. Algorithmenklassen

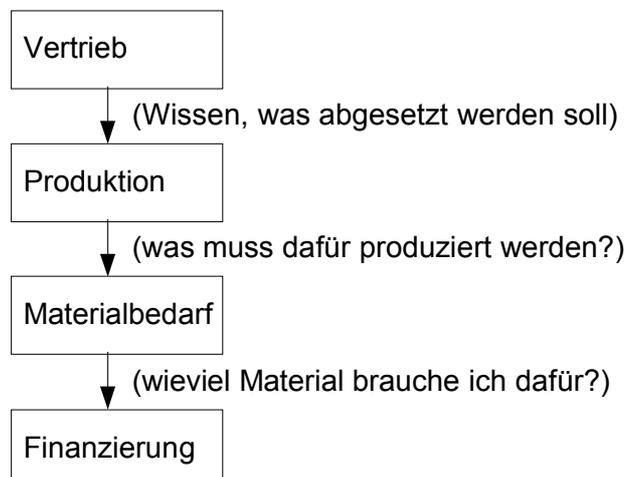
- Dynamische Programmierung
- Lineare Programmierung
- Netzplantechnik

7. Allgemeines Vorgehen

1. EP formulieren
2. EP in mathematisches Entscheidungsproblem transformieren
3. Entscheidungsmodell mit Algorithmus lösen

Dynamische Programmierung

- DP löst EP sequentiell und nicht simultan



sequentiell: In jeder Iteration ist nur eine einzige Variable im Rechengang

simultan: In jeder Iteration sind alle Variablen im Rechengang

Vor- und Nachteile:

1. Vorteile:

- Rechenaufwand gering
- Keine speziellen Anforderungen an die mathematische Struktur des EP

2. Nachteile:

- Bei mehr als einer Beschränkung unverträglich hoher Rechenaufwand
- Es gibt keinen festgefügten Algorithmus

Ein Beispiel zur DP:

Das Ladeproblem

allgemeine Symbole:

- | | |
|-------------------------------|---------------------------------------------------------------------------------------|
| (1) j ($j = 1, \dots, n$) | einzelnes Gut |
| (2) α_j | Gewicht des Gutes j pro Mengeneinheit |
| (3) x_j | unbekannte Lademenge des Gutes j (Entscheidungsvariable) |
| (4) $\Phi_j(x_j)$ | Gewinn aus Beladung des Gutes j auf den LKW in Abhängigkeit von der Lademenge x_j |
| (5) β | Ladekapazität des LKW in Gewichtseinheiten |
| (6) G | Gesamtgewinn aus der Verladung aller n Güter |

Lademodell:

$$(1) \max G = \sum_{j=1}^n \phi_j(x_j) \quad \text{Zielfunktion}$$

$$(2) \sum_{j=1}^n \alpha_j \cdot x_j \leq \beta \quad \text{Ladebeschränkung}$$

$$(3) x_j \geq 0 \quad \text{und ganzzahlig (} j = 1, \dots, n \text{)}$$

Ein Ladeproblem: $n = 3, \beta = 5$

Gut j	Gewinn $\Phi_j(x_j)$	Gewicht α_j
1	$2\sqrt{x_1}$	3
2	$3\sqrt{x_2}$	2
3	$\sqrt{x_3}$	1

→ Lademodell:

(1) Zielfunktion:

$$\max G = 2\sqrt{x_1} \cdot x_1 + 3\sqrt{x_2} \cdot x_2 + \sqrt{x_3} \cdot x_3$$

(2) Restriktion:

$$3 \cdot x_1 + 2 \cdot x_2 + 1 \cdot x_3 \leq 5$$

(3) Nichtnegativität:

$$x_1, x_2, x_3 \geq 0 \quad \text{und ganzzahlig}$$

Exkursion in die lineare Programmierung [kommt später noch]:

(falls die Gewinnfunktionen linear sind)
Speedfaktor:

Gewinn pro Mengeneinheit des Gutes $j = \text{const.} = \frac{\phi_j(x_j)}{x_j} = g_j$

→ Speedfaktor: $\frac{g_j}{\alpha_j} = \text{Gewinn pro Gewichtseinheit}$

$$\frac{g_s}{\alpha_s} = \max_j \left\{ \frac{g_j}{\alpha_j} \right\}$$

Beispiel: $n = 3, \beta = 7$

Gut j	Gewinn $\Phi_j(x_j)$	Gewicht α_j
1	2	$5 x_1 \Rightarrow \underline{2,5}$
2	3	$2 x_2 \Rightarrow 2$
3	5	$1 x_3 \Rightarrow 2,4$

$\overset{opt.}{x} = \frac{7}{2} = 3,5$

optimale Lösung: $\overset{opt.}{x}_1 = 3,5, \overset{opt.}{x}_2, \overset{opt.}{x}_3 = 0$
 $G_{\max} = 5 \cdot 3,5 = 17,5$

Zurück zur DP:

Zerlegung des Lademodells in Stufen:

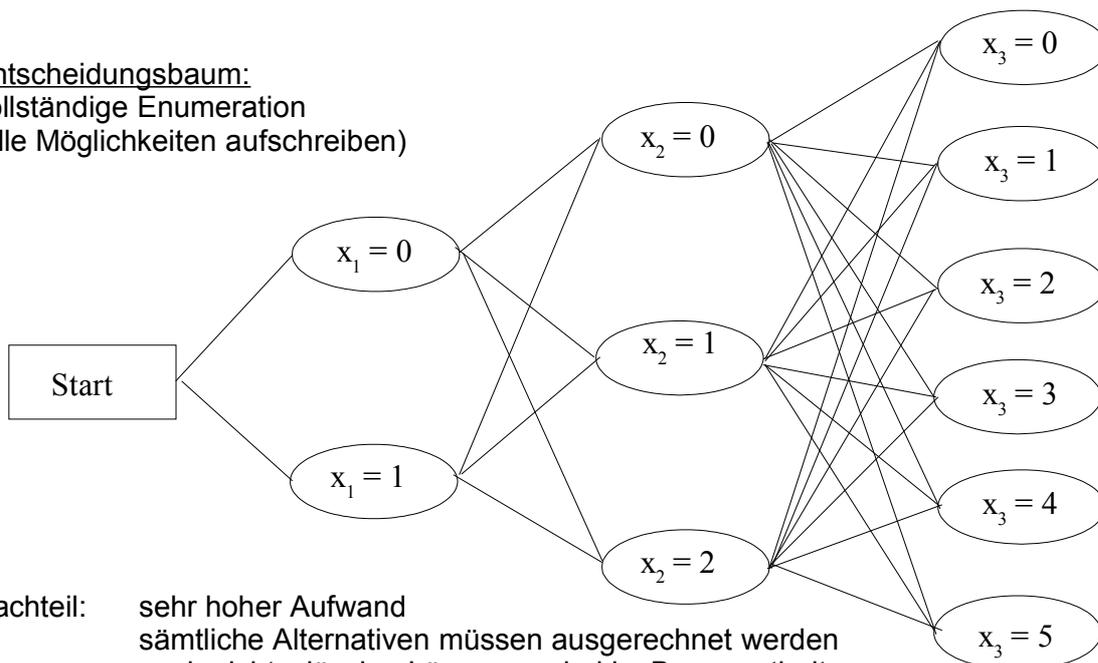
1. Stufe: Gut 1; x_1
2. Stufe: Gut 2; x_2
3. Stufe: Gut 3; x_3

Zulässige Alternativen von x_j auf Stufe j :

1. Stufe: $3 x_1 \leq 5, x_1 \geq 0$ und ganzzahlig \Rightarrow zulässige Beladung: 0 oder 1
2. Stufe: $2 x_2 \leq 5, x_2 \geq 0$ und ganzzahlig \Rightarrow zulässige Beladung: 0, 1 oder 2
3. Stufe: $1 x_3 \leq 5, x_3 \geq 0$ und ganzzahlig \Rightarrow zulässige Beladung: 0, 1, 2, 3, 4 oder 5

Entscheidungsbaum:

vollständige Enumeration
 (alle Möglichkeiten aufschreiben)



Nachteil: sehr hoher Aufwand
 sämtliche Alternativen müssen ausgerechnet werden
 auch nichtzulässige Lösungen sind im Baum enthalten

Dynamische Programmierung

- teilt das Gesamtproblem in kleinere Unterprobleme auf.
- für jede einzelne Stufe wird separat eine zulässige, optimale Entscheidung getroffen, die auch zulässig und optimal für das Entscheidungsproblem ist.

Die wichtigsten theoretischen Grundlagen der Dynamischen Programmierung:

a) Konzept der Zustandsvariable $\beta_j \rightarrow$ Zulässigkeit

β_j : verfügbare Ladekapazität für die Stufen 1 bis j

Annahme: $0 \leq \beta_j \leq \beta$

Warum gewährleistet dies die Zulässigkeit?

Wählt man auf jeder Stufe j den Wert von x_j bezüglich der Zustände β_j so, dass die Beschränkung $\alpha_j \leq \beta_j; 0 \leq \beta_j \leq \beta; x_j \geq 0$ und ganzzahlig ($j = 1, \dots, n$) erfüllt sind, dann ist x_j aufgrund der Definition der Zustandsvariable β_j zulässig für die Stufen 1 bis j ($j = 1, \dots, n$) und daher zulässig für das Gesamtproblem.

b) Zur Optimalität:

Bellmansches Optimalitätsprinzip (BOP)

“Eine optimale Politik (Lösung) hat die Eigenschaft, dass gleichgültig wie der Anfangszustand des Systems und die erste Entscheidung lauten, die restlichen Entscheidungen optimal bzgl. des Zustandes getroffen werden müssen, der aus der ersten Entscheidung resultiert.”

4 Schlüsselbegriffe: Anfangszustand
 erste Entscheidung
 resultierender Zustand
 restliche optimale Entscheidungen

(1) Zielfunktion:

$$\max G = 2\sqrt{x_1} \cdot x_1 + 3\sqrt{x_2} \cdot x_2 + \sqrt{x_3} \cdot x_3$$

(2) Restriktion:

$$3 \cdot x_1 + 2 \cdot x_2 + 1 \cdot x_3 \leq 5$$

(3) Nichtnegativität:

$$x_1, x_2, x_3 \geq 0 \quad \text{und ganzzahlig}$$

(4) $x_1 = 0; 1$; $x_2 = 0; 1; 2$; $x_3 = 0; 1; 2; 3; 4; 5$

1. Stufe: Ermittle zulässige optimale Lademenge x_1 von Gut 1.

$$F_1(\beta_1) = \max \{ \phi_1(x_1) \} \Rightarrow x_1^*(\beta_1)$$

$$\beta_1 = 0, \dots, \beta \quad \alpha_1 x_1 \geq \beta_1$$

$$x_1 \geq 0 \text{ und ganzzahlig}$$

β_1	$2\sqrt{x_1}$		maximaler Gewinn $F_1(\beta_1)$	opt. Lösung $x_1(\beta_1)$
	$2\sqrt{0}$	$2\sqrt{1}$		
0	0	-	0	0
1	0	-	0	0
2	0	-	0	0
3	0	2	2	1
4	0	2	2	1
5	0	2	2	1

Rechenbeispiele:

$$F_1(0) = \max_{\substack{3x_1 \leq 0 \\ x_1 \geq 0 \text{ u.g.}}} \{2\sqrt{0}=0\} = 0 \Rightarrow x_1^o = 0$$

$$F_1(3) = \max_{\substack{3x_1 \leq 3 \\ x_1 \geq 0 \text{ u.g.}}} \{2\sqrt{0}=0; 2\sqrt{1}=2\} = 2 \Rightarrow x_1^o(\beta_1=3) = 1$$

2. Stufe: Ermittle zulässige optimale Lademenge x_2 von Gut 2.

β_2 : verfügbare Ladekapazität für die Güter 1 und (bis) 2

$$F_2(\beta_2) = \max_{\substack{\beta_2=0, \dots, \beta \\ \alpha_2 x_2 \geq \beta_2 \\ x_2 \geq 0 \text{ und ganzzahlig}}} \{ \phi_2(x_2) + F_1(\beta_1 = \beta_2 - \alpha_2 x_2) \} \Rightarrow x_2^o(\beta_2)$$

β_2	$3\sqrt{x_2} + F_1(\beta_2 - \alpha_2 x_2)$			maximaler Gewinn $F_2(\beta_2)$	opt. Lösung $x_2(\beta_2)$
	$X_2=0$	$X_2=1$	$X_2=2$		
0	0	-	-	0	0
1	0	-	-	0	0
2	0	3	-	3	1
3	2	3	-	3	1
4	2	3	4,24	4,24	2
5	2	5	4,24	5	1

Rechenbeispiele:

$$F_2(3) = \max_{\substack{2x_2 \leq 3 \\ x_2 \geq 0 \text{ u.g.}}} \left\{ \begin{array}{l} 3\sqrt{0} + F_1(5) = 0 + 2 \\ 3\sqrt{1} + F_1(2) = 3 + 0 \end{array} \right\} = 3 \Rightarrow x_2^o(\beta_2=3) = 1$$

$$F_2(5) = \max_{\substack{2x_2 \leq 5 \\ x_2 \geq 0 \text{ u.g.}}} \left\{ \begin{array}{l} 3\sqrt{0} + F_1(5) = 0 + 2 \\ 3\sqrt{1} + F_1(5 - 2 \cdot 1) = 3 + 2 \\ 3\sqrt{2} + F_1(5 - 2 \cdot 2) = 4,242 + 0 \end{array} \right\} = 5 \Rightarrow x_2^o(\beta_2=5) = 1$$

weiteres Problem der Dynamischen Programmierung:

Das Rundreiseproblem:

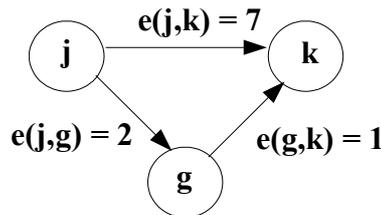
gesucht: entfernungsminimale Rundreise (Start- und Endpunkt sollen identisch sein)

gegeben: $j, k = 0, 1, \dots, n$ (also $n+1$ Orte)

e_{jk} = direkte Entfernung von j nach k

zu beachten:

- a) e_{jk} muss nicht die kürzeste Entfernung sein.
- b) es kann sein, dass $e_{jk} \neq e_{kj}$ (Einbahnstraßen)
- c) evtl. keine direkte Verbindung
- d) möglicherweise kann ein Ort k nur über einen anderen Ort j erreicht und wieder verlassen werden kann (Bergdorf)



Entfernung: $e(j,k) = 7$
 Distanz: $d(j,k) = 3$

Begriffserklärung:

Entfernung: kürzester *direkter* Weg zwischen j und k ohne Zwischenorte

Distanz: kürzester Weg zwischen j und k (also unter Umständen mit Zwischenorten)

Vorgehen beim Lösen des Rundreiseproblems:

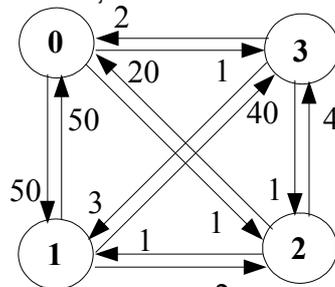
Problemlösung in 2 Schritten:

Ziel: Gesucht ist die kürzeste Rundreise.

1. Schritt: Ermittle mit den e_{jk} die d_{jk} für alle Orte mit Hilfe des Politik-Approximations-Algorithmus (PAA)

2. Schritt: Ermittle mit den d_{jk} die kürzeste Rundreise mit Hilfe des "Branch and Bound" (BaB)

Entfernungsgraph:



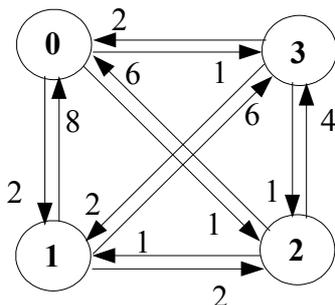
kürzeste Rundreise mit Entfernungen:
 $0 \rightarrow 2 \rightarrow 1 \rightarrow 2 \rightarrow 3 \rightarrow 0$ [Orte]
 $1 + 1 + 2 + 4 + 2 = 10$ [Kilometer]

PAA:

- $d(0,1) = 2$ (über 2 nach 1)
- $d(0,2) = 1$
- $d(0,3) = 1$
- $d(1,0) = 8$ (über 2 nach 3 nach 1)
- $d(1,2) = 2$
- $d(1,3) = 6$ (über 2 nach 3)

- $d(2,0) = 6$ (über 3 nach 0)
- $d(2,1) = 1$
- $d(2,3) = 4$
- $d(3,0) = 2$
- $d(3,1) = 2$ (über 2 nach 1)
- $d(3,2) = 1$

Distanzgraph:



Bei $n+1$ Orten gibt es $n!$ Rundreisen.
 Hier: $3+1 = 4$ Orte $\rightarrow 3! = 6$ Rundreisen.

BAB: (hier nur mit vollständiger Enumeration)

1. 0 → 1 → 2 → 3 → 0	: 2 + 2 + 4 + 2 = <u>10</u>
2. 0 → 1 → 3 → 2 → 0	: 2 + 6 + 1 + 6 = 15
3. 0 → 2 → 1 → 3 → 0	: 1 + 1 + 6 + 2 = <u>10</u>
4. 0 → 2 → 3 → 1 → 0	: 1 + 4 + 2 + 8 = 15
5. 0 → 3 → 1 → 2 → 0	: 1 + 2 + 2 + 6 = 11
6. 0 → 3 → 2 → 1 → 0	: 1 + 1 + 1 + 8 = 11

Kürzeste Rundreise mit Distanzen: 1 und 3.

- 1.) 0 → 1 → 2 → 3 → 0 mit Distanzen entspricht 0 → 2 → 1 → 2 → 3 → 0 mit Entfernungen
- 3.) 0 → 2 → 1 → 3 → 0 mit Distanzen entspricht 0 → 2 → 1 → 2 → 3 → 0 mit Entfernungen

PAA:

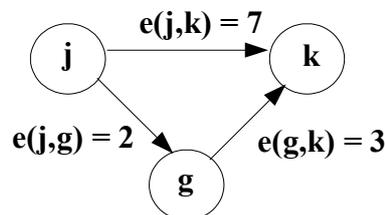
- überflüssig, wenn $e_{jk} = d_{jk}$ für alle Entfernungen
- bei $n+1$ Orten, wird der PAA $n+1$ mal angewendet
- Idee: sukzessive Zwischenorte einführen, bis
 - a) alle möglichen Zwischenorte ($n-1$) eingeführt sind oder
 - b) ein weiterer Zwischenort keine Verkürzung mehr bringt.

PAA ist ein Algorithmus der Dynamischen Programmierung

- **Stufe:** Zahl der Zwischenorte l auf dem Weg von j nach k
- **Zustandskonzept:** Reduktion der Ausgangsfragestellung „kürzeste Entfernung von allen Orten zu allen Orten“ auf „kürzeste Entfernung von allen Orten zu einem Ort“.
- **BOP:** Anfangszustand des Systems: Startorte j ($j = 0, 1, \dots, n$)
 - 1. Entscheidung: Wahl der Teilstrecke (g) von j zum ersten Zwischenort g
 - resultierender Zustand: Zwischenort g
 - restliche optimale Entscheidungen: Wahl der Teilstrecken auf dem kürzesten Weg von g nach k

Symbole für Rekursionsbeziehungen des PAA

- f_{jk}^l : kürzeste Entfernung von j nach k über l Zwischenorte (optimaler Funktionswert)
- p_{jk}^l : optimale Wahl des ersten Zwischenortes g auf dem kürzesten Weg von j nach k über l Zwischenorte (optimale Politik)



Beispiel für $l = 3$

$$f_{jk}^l = \min_{g_1, g_2, g_3} \left\{ e_{jg_1} + f_{g_1k}^2, e_{jg_2} + f_{g_2k}^2, e_{jg_3} + f_{g_3k}^2 \right\} \Rightarrow p_{jk}^3 = g_2$$

$$f_{jk}^{l=0} = e_{jk} \quad \text{für alle } j \neq k \Rightarrow p_{jk}^{l=0} = k$$

$$f_{jk}^{l=0} = 0 \quad \text{für alle } j = k \Rightarrow p_{kk}^{l=0} = k$$

$$f_{jk}^{l>0} = \min_{g \neq j} \left\{ e_{jg} + f_{gk}^{l-1} \right\} \Rightarrow p_{jk}^{l>0}$$

$$f_{kk}^{l=0} = 0 \Rightarrow p_{kk}^{l>0} = k$$

Abbruchkriterium des PAA bei n+1 Orten:

a) $l = n + 1$ Maximum der Zwischenorte ist erreicht.

b) $f_{jk}^l = f_{jk}^{l-1} \quad \forall j$

$j \setminus k$	0	1	2	3	4	5	f_{j0}^0	p_{j0}^0	f_{j1}^1	p_{j1}^1	f_{j2}^2	p_{j2}^2	f_{j3}^3	p_{j3}^3	f_{j4}^4	p_{j4}^4
0	0	4	8	∞	∞	∞	0	0	0	0						
1	4	0	19	∞	5	∞	4	0	4	0						
2	∞	19	0	5	19	∞	∞	-	23	1						
3	∞	∞	5	0	∞	6	∞	-	∞	-						
4	∞	3	2	8	0	23	∞	-	7	1						
5	∞	∞	∞	14	5	0	∞	-	∞	-						

Erstes DP-Problem: von allen Orten j nach Ort k = 0 (Beispielrechnungen)

1. Stufe: l = 0

$j=0: f_{00}^0=0 \Rightarrow p_{00}^0=0$

$j=1: f_{10}^0=4 \Rightarrow p_{10}^0=0$

$j=2: f_{20}^0=\infty \Rightarrow p_{20}^0=-$

2. Stufe: l = 1

$j=0: f_{00}^1=0 \Rightarrow p_{00}^1=0$

$j=1: f_{10}^1 = \min_{g \neq 1} \{e_{10} + f_{00}^0, -, e_{12} + f_{20}^0, e_{13} + f_{30}^0, e_{14} + f_{40}^0, e_{15} + f_{50}^0\} = 4 \Rightarrow p_{10}^1 = 0$

$j=2: f_{20}^1 = \min_{g \neq 2} \{e_{20} + f_{00}^0, e_{21} + f_{10}^0, -, e_{23} + f_{30}^0, e_{24} + f_{40}^0, e_{25} + f_{50}^0\}$
 $= \min_{g \neq 2} \{\infty + 0, 19 + 4, -, 5 + \infty, 19 + \infty, \infty + \infty\} = 23 \Rightarrow p_{20}^1 = 1$

3. Stufe: l = 2

$j=0: f_{00}^2=0 \Rightarrow p_{00}^2=0$

$j=3: f_{30}^2 = \min_{g \neq 3} \{e_{30} + f_{00}^1, e_{31} + f_{10}^1, e_{32} + f_{20}^1, -, e_{34} + f_{40}^1, e_{35} + f_{50}^1\}$
 $= \min_{g \neq 3} \{\infty + 0, \infty + 4, 5 + 23, -, \infty + 7, 6 + \infty\} = 28 \Rightarrow p_{30}^2 = 2$

$j \setminus k$	0	1	2	3	4	5	f_{j0}^0	p_{j0}^0	f_{j1}^1	p_{j1}^1	f_{j2}^2	p_{j2}^2	f_{j3}^3	p_{j3}^3	f_{j4}^4	p_{j4}^4
0	0	4	8	∞	∞	∞	0	0	0	0	0	0	0	0	0	0
1	4	0	19	∞	5	∞	4	0	4	0	4	0	4	0	4	0
2	∞	19	0	5	19	∞	∞	-	23	1	23	1	23	1	23	1
3	∞	∞	5	0	∞	6	∞	-	∞	-	28	2	18	5	18	5
4	∞	3	2	8	0	23	∞	-	7	1	7	1	7	1	7	1
5	∞	∞	∞	14	5	0	∞	-	∞	-	12	4	12	4	12	4

Abbruchkriterien: Entweder n-1 Zwischenorte erreicht oder keine Änderung der f_{jk}^l in der nächsten Stufe.

weitere Beispielrechnungen:

Erstes DP-Problem: von allen Orten j nach Ort k = 0

4. Stufe:

$$j = 3: f_{30}^4 = \min \{ e_{30} + f_{00}^3, e_{31} + f_{10}^3, e_{32} + f_{20}^3, -, e_{34} + f_{40}^3, e_{35} + f_{50}^3 \}$$

$$= \min \{ \infty + 0, \infty + 4, 5 + 23, -, \infty + 7, 6 + 12 \}$$

$$= 18 \Rightarrow p_{30}^4 = 5$$

$$j = 4: f_{40}^4 = \min \{ e_{40} + f_{00}^3, e_{41} + f_{10}^3, e_{42} + f_{20}^3, e_{43} + f_{30}^3, -, e_{45} + f_{50}^3 \}$$

$$= \min \{ \infty + 0, 3 + 4, 2 + 23, 8 + 18, -, 23 + 12 \}$$

$$= 7 \Rightarrow p_{40}^4 = 1$$

→ Beide Abbruchkriterien erfüllt.

Wegen $l = 4$ (1. Abbruchkriterium) und $f_{j0}^4 = f_{j0}^3$ (2. Abbruchkriterium) bricht hier der PAA ab.

Setze $d_{j0} = f_{j0}^4$ für $j = 0, \dots, 5$

f_{j4}^4	p_{j4}^4	d_{j0}	opt. Politik
0	0	0	0 → 0 → 0 → 0 → 0 → 0
4	0	4	1 → 0 → 0 → 0 → 0 → 0
23	1	23	2 → 1 → 0 → 0 → 0 → 0
18	5	18	3 → 5 → 4 → 1 → 0 → 0
7	1	7	4 → 1 → 0 → 0 → 0 → 0
12	4	12	5 → 4 → 1 → 0 → 0 → 0

Die optimale Politik wird nun rekursiv abgeleitet. Man springt beim Zurückgehen in den Spalten von einer Zeile zur passenden vorherigen.

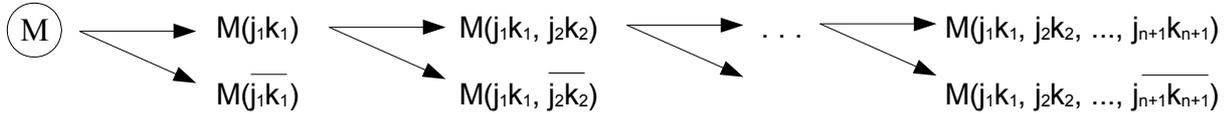
Mit den jeweiligen Distanzen lässt sich nun die Distanzmatrix aufstellen:

$j \setminus k$	0	1	2	3	4	5
0	0	4	8	13	9	19
1	4	0	7	12	5	18
2	23	19	0	5	16	11
3	18	14	5	0	11	6
4	7	3	2	7	0	13
5	12	8	7	12	5	0

Branch and Bound
(Branch: Zweig, Bound: Schranke)

Idee:

- M = Menge aller Rundreisen
- Zerlegung von M in zwei disjunkte Teilmengen.
- $M(j_1k_1)$: Menge aller Rundreisen mit der Teilstrecke j_1k_1 .
- $M(\overline{j_1k_1})$: Menge aller Rundreisen ohne die Teilstrecke j_1k_1 .



Bound einer Menge:

Streckenlänge, die von keiner Rundreise in einer betrachteten Menge unterschritten wird (untere Schranke)

Abbruchkriterien des BaB:

- a) Es wurde eine Untermenge gefunden, die nur eine Rundreise enthält.
- b) Bound einer zulässigen Rundreise ≤ Bounds aller nicht untersuchten Teilmengen.

neue Distanzmatrix:

$j \setminus k$	1	2	3	4	5	6	ZM
1	∞	27	43	16	30	26	16
2	7	∞	16	1	30	25	1
3	20	13	∞	35	5	0	0
4	21	16	25	∞	18	18	16
5	12	46	27	46	∞	5	5
6	23	5	5	9	5	∞	5
							Σ 43

BM:

Bound einer Menge M
Summe aller Distanzen d_{jk} , die man erhält, wenn man jeden Ort $j = 1, \dots, n$ distanzminimal verlässt und jeden Ort $k = 1, \dots, n$ distanzminimal anfährt.

Distanzzuwachsmatrix:

$j \setminus k$	1	2	3	4	5	6
1	∞	11	27	0	14	10
2	6	∞	15	0	29	24
3	20	13	∞	35	5	0
4	5	0	9	∞	2	2
5	7	41	22	43	∞	0
6	18	0	0	4	0	∞
SM	5	0	0	0	0	0

Spaltenminimum: Σ 5

Also:

BM = Summe der ZM + Summe der SM = 43 + 5 = 48
Keine Rundreise in dieser Menge kann kürzer als 48km sein.

Daraus folgt die neue Distanzzuwachsmatrix:

$j \setminus k$	1	2	3	4	5	6
1	∞	11	27	0	14	10
2	1	∞	15	0	29	24
3	15	13	∞	35	5	0
4	0	0	9	∞	2	2
5	2	41	22	43	∞	0
6	13	0	0	4	0	∞

Nun wählt man eine Teilstrecke (j_1, k_1) aus bei der

- a) $BM(j_1, k_1)$ möglichst klein
- b) $BM(\overline{j_1, k_1})$ möglichst groß

Hier: Als (j_1, k_1) Nullelement wählen.

Branch and Bound

- 1) Berechne BM [Bound von M]
 - 2) Teilstrecke (j_1, k_1) auswählen (\rightarrow anhand einer Plausibilitätsüberlegung)
 - a) $BM(j_1, k_1)$ möglichst klein
 \rightarrow als (j_1, k_1) Nullelement auswählen
 - b) $BM(\overline{j_1, k_1})$ möglichst groß
 \rightarrow dasjenige Nullelement auswählen, das bei Nichtbefahren den größten Mindestumwegzuwachs verursachen würde.
- \rightarrow Dadurch steigt die Wahrscheinlichkeit, das die gesuchte kürzeste Rundereise in $M(j_1, k_1)$ und nicht in $M(\overline{j_1, k_1})$ liegt.

$j \setminus k$	1	2	3	4	5	6
1	∞	11	27	0^{10+0}	14	10
2	1	∞	15	0^{1+0}	29	24
3	15	13	∞	35	5	0^{5+2}
4	0^{0+1}	0^{0+0}	9	∞	2	2
5	2	41	22	43	∞	0^{2+2}
6	13	0^{0+0}	0^{0+9}	4	0^{0+2}	∞

Nullstellenbewertung:

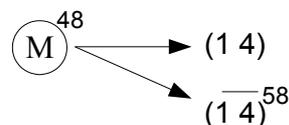
$$\gamma(jk) = \{ \text{Minimum Zeile } j \neq (jk) \} + \{ \text{Minimum Spalte } k \neq (jk) \}$$

$$\gamma(j_1, k_1) = \max_{\text{Nullstellen}} \{ \gamma(jk) \} \Rightarrow (j_1, k_1)$$

Hier:

$$\gamma(j_1, k_1) = \max \{ 10, 1, 5, 1, 0, 2, 0, 9, 2 \} = 10 \Rightarrow (j_1, k_1) = (1, 4)$$

3) $BM(\overline{j_1, k_1}) = BM + \gamma(j_1, k_1) = 48 + 10 = 58$



4) Zulässigkeit:

4a) Kurzzyklen vermeiden

(1 4) → (4 1) wäre möglicher Kurzzyklus
 → (4 1) mit ∞ sperren.

4b) Streiche Zeile j_1 und Spalte k_1

→ hier $j_1 = 1$ und $k_1 = 4$ streichen.

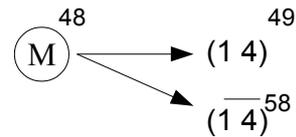
An dieser Stelle beginnt der Algorithmus von vorn.

$j \setminus k$	1	2	3	5	6	ZM
2	1	∞	15	29	24	1
3	15	13	∞	5	0	0
4	∞	0	9	2	2	0
5	2	41	22	∞	0	0
6	13	0	0	0	∞	0
						$\Sigma = 1$

$j \setminus k$	1	2	3	5	6	
2	0	∞	14	28	23	
3	15	13	∞	5	0	
4	∞	0	9	2	2	
5	2	41	22	∞	0	
6	13	0	0	0	∞	
SM	0	0	0	0	0	$\Sigma = 0$

Reduktionskonstante $r = \{\text{Summe der ZM}\} + \{\text{Summe der SM}\} = 1 + 0 = 1$

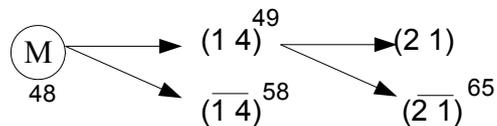
$BM(j_1, k_1) = BM + r = 48 + 1 = 49$



$j \setminus k$	1	2	3	5	6
2	0^{14+2}	∞	14	28	23
3	15	13	∞	5	0^{5+2}
4	∞	0^{2+0}	9	2	2
5	2	41	22	∞	0^{2+0}
6	13	0^{0+0}	0^{0+9}	0^{0+2}	∞

$\gamma(j_2, k_2) = \max \{16, 5, 2, 2, 0, 9, 2\} = 16 \Rightarrow (j_2, k_2) = (2, 1)$

$BM(j_1, k_1, \overline{j_2, k_2}) = BM(j_1, k_1) + \gamma(j_2, k_2)$
 $= 49 + 16 = 65$



Zulässigkeit: 1)

(1 2) braucht nicht mehr gesperrt werden

(4 2) muss gesperrt werden, da sonst ein Kurzzyklus (2 → 1 → 4) möglich wäre

2)

Streichen: Zeile 2, Spalte 1

An dieser Stelle beginnt der Algorithmus wieder von vorn:

$j \setminus k$	2	3	5	6	ZM
3	13	∞	5	0	0
4	∞	9	2	2	2
5	41	22	∞	0	0
6	0	0	0	∞	0
					$\Sigma = 2$

=>

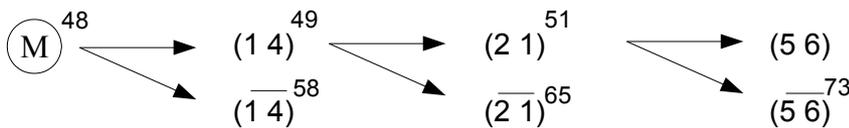
$j \setminus k$	2	3	5	6	
3	13	∞	5	0 ⁵	
4	∞	7	0 ⁰	0 ⁰	
5	41	22	∞	0 ²²	
6	0 ¹³	0 ⁷	0 ⁰	∞	
SM	0	0	0	0	$\Sigma = 0$

$$\Rightarrow r = \sum ZM + \sum SM = 2 + 0 = 2$$

$$\gamma(j_3 k_3) = 22 \Rightarrow (j_3 k_3) = (56)$$

$$BM(j_1 k_1, j_2 k_2) = BM(j_1 k_1) + r = 49 + 2 = 51$$

$$BM(\overline{56}) = BM(21) + \gamma(56) = 51 + 22 = 73$$



- Zulässigkeit: 1.) Rückweg sperren: (6 5) $\rightarrow \infty$
 2.) Zeile 5 und Spalte 6 streichen.

An dieser Stelle beginnt der Algorithmus wieder von vorn:

$j \setminus k$	2	3	5	ZM
3	13	∞	5	5
4	∞	7	0	0
6	0	0	∞	0
				$\Sigma = 5$

=>

$j \setminus k$	2	3	5	
3	8	∞	0 ⁸	
4	∞	7	0 ⁷	
6	0 ⁸	0 ⁷	∞	
SM	0	0	0	$\Sigma = 0$

$$\Rightarrow r = \sum ZM + \sum SM = 5 + 0 = 5$$

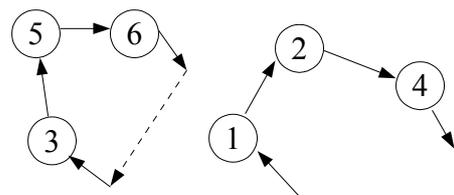
$$BM(56) = BM(21) + r = 51 + 5 = 56$$

Nullstellenbewertung:
 Problem!

gleiche Bewertung von (3 5) und (6 2)
 \rightarrow willkürliche Auswahl von (3 5)

$$\gamma(j_4 k_4) = 8 \Rightarrow (j_4 k_4) = (35)$$

$$BM(\overline{35}) = BM(56) + \gamma(35) = 56 + 8 = 64$$



- Zulässigkeit: 1.) (6 3) sperren, da sonst Kurzyzyklus möglich wäre.
 2.) Zeile 3, Spalte 5 streichen

An dieser Stelle beginnt der Algorithmus wieder von vorn:

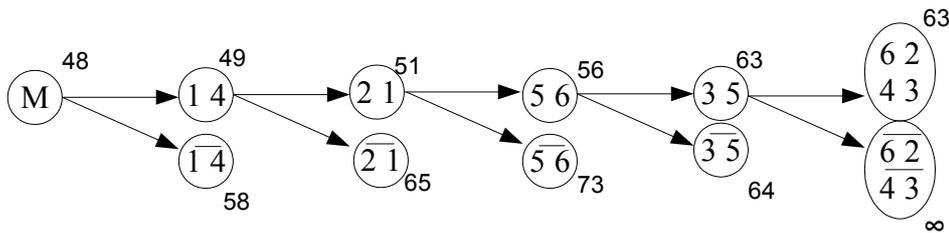
<i>j \ k</i>	2	3	ZM
4	∞	7	7
6	0	∞	0
			∑ = 7

=>

<i>j \ k</i>	2	3	
4	∞	0	
6	0	∞	
SM	0	0	∑ = 0

$\Rightarrow r = \sum ZM + \sum SM = 7 + 0 = 0$

$BM(35) = BM(56) + r = 56 + 7 = 63$



Wurde die ursprüngliche 6x6-Matrix durch Streichung von Zeilen und Spalten auf eine 2x2-Matrix reduziert, dann stehen mit den Nullelementen in dieser 2x2-Matrix die restlichen beiden Teilstrecken fest.

Denn nur diese beiden schließen die Rundreise so, dass jeder Ort nur einmal verlassen und einmal angefahren wird und das Zyklenverbot eingehalten wird.

zulässige Rundreise:

$(1\ 4) - (4\ 3) - (3\ 5) - (5\ 6) - (6\ 2) - (2\ 1)$ mit der Länge 63 [BM(62) = BM(43)]

Wegen $BM(\overline{14}) = 58 < 63 = BM(62)$ ist hier das Abbruchkriterium noch nicht erfüllt.

→ Das heißt: Prüfen, ob die kürzeste Rundreise evtl. in $M(\overline{14})$ enthalten ist.

Wiederholung: Abbruchkriterien des BaB:

- a) Es wurde eine Untermenge gefunden, die nur eine Rundreise enthält.
- b) Bound einer zulässigen Rundreise ≤ Bounds aller nicht untersuchten Teilmengen.

Suchen in $M(\overline{14})$: Dazu wird die Ausgangsmatrix zu M genommen und die Strecke (1 4) gesperrt.

<i>j \ k</i>	1	2	3	4	5	6	ZM
1	∞	11	27	∞	14	10	10
2	6	∞	15	0	29	24	0
3	20	13	∞	35	5	0	0
4	5	0	9	∞	2	2	0
5	7	41	22	43	∞	0	0
6	18	0	0	4	0	∞	0
							∑ = 10

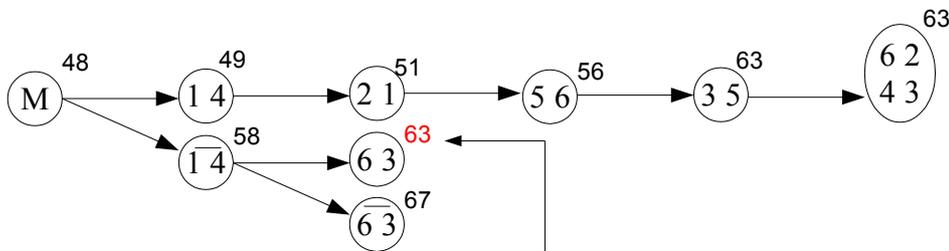
=>

<i>j \ k</i>	1	2	3	4	5	6	
1	∞	1	17	∞	4	0	
2	6	∞	15	0	29	24	
3	20	13	∞	35	5	0	
4	5	0	9	∞	2	2	
5	7	41	22	43	∞	0	
6	18	0	0	4	0	∞	
SM	5	0	0	0	0	0	∑ = 5

Suche nach der optimalen Lösung:

$j \setminus k$	1	2	3	4	5	6	ZM
1	∞	11	27	∞	14	10	10
2	1	∞	15	0	29	24	0
3	15	13	∞	35	5	0	0
4	0	0	9	∞	2	2	0
5	2	41	22	43	∞	0	0
6	13	0	0	4	0	∞	0
							$\Sigma=10$

$j \setminus k$	1	2	3	4	5	6	
1	∞	1	17	∞	4	0 ¹	
2	1	∞	15	0 ⁵	29	24	
3	15	13	∞	35	5	0 ⁵	
4	0 ¹	0 ¹	9	∞	2	2	
5	2	41	22	43	∞	0 ²	
6	13	0 ⁰	0 ⁹	4	0 ²	∞	
SM	0	0	0	0	0	0	$\Sigma=0$



$j \setminus k$	1	2	4	5	6	ZM
1	∞	1	∞	4	0	0
2	1	∞	0	29	24	0
3	15	13	35	5	∞	5
4	0	0	∞	2	2	0
5	2	41	43	∞	0	0
						$\Sigma=5$

$j \setminus k$	1	2	4	5	6	
1	∞	1	∞	4	0	
2	1	∞	0	29	24	
3	10	8	30	0	∞	
4	0	0	∞	2	2	
5	2	41	43	∞	0	
SM	0	0	0	0	0	$\Sigma=0$

Wir haben mit der angegebenen Rundreise bereits eine optimale Lösung des Problems gefunden! Jede weitere Rundreise ist mindestens genauso lang.

Lineare Programmierung:

	DP	LP
Vorgehen	sequentiell	simultan
Zielfunktion	nicht linear	linear
Restriktionen	eine Beschränkung (kann linear sein)	lineare Beschränkungen

Besonderheiten:

- komplette Theorie:
- grundlegend für andere Algorithmen

Produktionsproblem:

Maschine	Gut 1	Gut 2	Kapazität im Planungsabschnitt
A	1	2	10
B	1	1	7
C	3	0	18
Gewinn / ME:	3€	5€	

Produktionsmodell:

- x_1 : herzustellende Menge von Gut 1
 x_2 : herzustellende Menge von Gut 2
 G: Gesamtgewinn aus dem Verkauf beider Güter
 ZF: $\max G = 3x_1 + 5x_2$
 NB: $1x_1 + 2x_2 \leq 10$
 $1x_1 + 1x_2 \leq 7$
 $3x_1 + 0x_2 \leq 18$
 NNB: $x_1, x_2 \geq 0$

Lösen von linearen Gleichungssystemen mit m Gleichungen und m Variablen:

$$\begin{aligned}
 2x_1 + 1x_2 + 3x_2 &= 9 \\
 1x_1 + 2x_2 + 4x_2 &= 6 \\
 4x_1 + 1x_2 + 2x_2 &= 10
 \end{aligned}$$

Ziel: äquivalentes Gleichungssystem in kanonischer Form erzeugen.

$$\begin{aligned}
 1x_1 + 0x_2 + 0x_2 &= z_1 \\
 0x_1 + 1x_2 + 0x_2 &= z_2 \\
 0x_1 + 0x_2 + 1x_2 &= z_3
 \end{aligned}$$

Vorgehen:

- 1) elementare Zeilenoperationen: additive und/oder multiplikative Verknüpfungen der Gleichungen; besonders: Wert der Variablen bleibt unverändert
- 2) äquivalentes Gleichungssystem in kanonischer Form:
besonders: die gesuchte Lösung kann direkt abgelesen werden.

	x_1	x_2	x_3		elem. Zeilenoperation
	2	1	3	9	I_0
	1	2	4	6	II_0
	4	1	2	10	III_0
%	1	$\frac{1}{2}$	$\frac{1}{3}$	$\frac{9}{2}$	$I_1 = I_0 / 2$
	0	$\frac{3}{2}$	$\frac{5}{2}$	$\frac{3}{2}$	$II_1 = II_0 - 1I_1$
	0	-1	-4	-8	$III_1 = III_0 - 4I_1$
%	1	0	$\frac{2}{3}$	4	$I_2 = I_1 - \frac{1}{2}II_2$
	0	1	$\frac{5}{3}$	1	$II_2 = II_1 \cdot \frac{2}{3}$
	0	0	$-\frac{7}{3}$	-7	$III_2 = III_1 + II_2$
	1	0	0	2	$I_3 = I_2 - \frac{2}{3}III_3$
	0	1	0	-4	$II_3 = II_2 - \frac{5}{9}III_3$
%	0	0	1	3	$III_3 = (-\frac{3}{7})III_2$

$x_1 = 2; x_2 = 4; x_3 = 3$

Die Standardform der Linearen Programmierung:

Begründung für Standardform: einheitliche Darstellung von Algorithmen der LP.
 Aussehen:

- (1) Zielfunktion ist zu minimieren $\min z = \sum_{j=1}^n \delta_j x_j$
- (2) nur Gleichungsbeschränkung
- (3) nichtnegative Beschränkungskonstanten $\sum_{j=1}^n \alpha_{ij} x_j = \beta_i \quad (i = 1, \dots, m) \beta_i \geq 0$
- (4) nichtnegative Entscheidungsvariablen $x_j \geq 0 \quad (j = 1, \dots, n)$

Überführung in die Standardform:

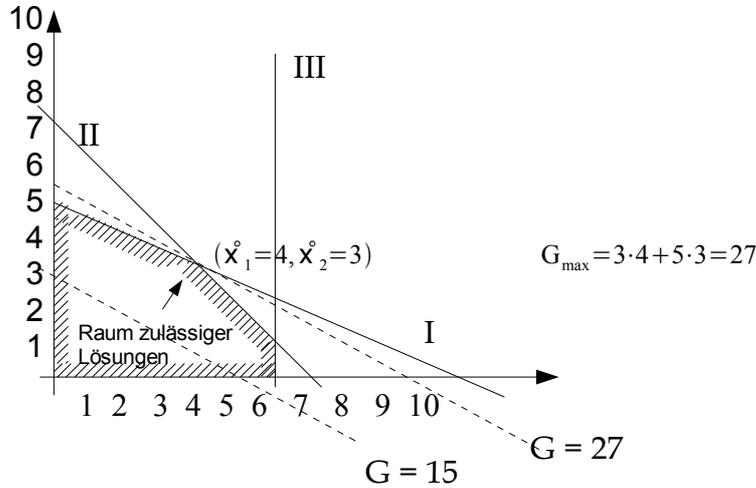
- (1) ZF * (-1) und minimieren
- (2) bei \leq : nichtnegative Schlupfvariable auf der linken Seite addieren.
 bei \geq : nichtnegative Überschussvariable auf der linken Seite subtrahieren.
- (3) *(-1) und Schlupfvariable einführen
- (4) -> Skript [nicht klausurrelevant]

Hier:	$\max G = 3x_1 + 5x_2$	→	Standardform: $\min (-G) = -3x_1 - 5x_2$
	$1x_1 + 2x_2 \leq 10$	→	$1x_1 + 2x_2 + x_{2+1} = 10$
	$1x_1 + 1x_2 \leq 7$	→	$1x_1 + 1x_2 + x_{2+2} = 7$
	$3x_1 + 0x_2 \leq 18$	→	$3x_1 + 0x_2 + x_{2+3} = 18$
	$x_1, x_2 \geq 0$	→	$x_1, x_2, x_{2+1}, x_{2+2}, x_{2+3} \geq 0$

Bei übersichtlichen Problemen der linearen Programmierung bietet sich eine graphische Lösung an. Hierfür ist die Standardform auch nicht erforderlich.

graphische Lösung des Problems:

1. Einzeichnen der drei Beschränkungen.
2. Kennzeichnen des zulässigen Lösungsraums
3. Eintragen einer Zielfunktion beliebigen Wertes.
4. Verschieben der Zielfunktion zum Optimum.



rechnerische Lösung:

- m: Zahl der Beschränkungen in der Standardform = 3
- n: Zahl der Variablen in der Standardform = 5

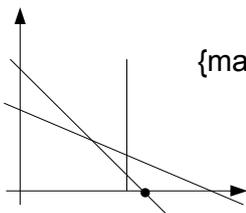
$m < n \rightarrow$ unterbestimmtes Gleichungssystem

Simplex-Ecken-Theorem:

Gibt es eine optimale Lösung für das gegebene Problem, dann liegt sie in mindestens einem Eckpunkt des Raums der zulässigen Lösungen.

Konsequenz:

- \rightarrow unterbestimmte lineare Gleichungssysteme sind eindeutig lösbar.
- im Eckpunkt sind $n-m$ Variablen = 0.
- \rightarrow Schritte zur Lösung sind endlich.



{maximale Anzahl der Eckpunkte} = {Anzahl der Möglichkeiten $n-m$ von n Variablen Null zu setzen.}

Hier: $\binom{n}{n-m} = \binom{5}{2} = 10$ Hierbei werden allerdings

sämtliche Eckpunkte gezählt.
(sowohl zulässige als auch unzulässige)

Hier sind es auch nicht die maximal 10, sondern nur 9 Punkte, da sich Gerade III und die x_2 -Achse nicht schneiden.

1.KF	x_1	x_2	x_{2+1}	x_{2+2}	x_{2+3}	G	(β_i)
x_{2+1}	1	2	1	0	0	0	10
x_{2+2}	1	1	0	1	0	0	7
x_{2+3}	3	0	0	0	1	0	18
G	-3	-5	0	0	0	1	0

nicht-Basisvariable: Die n-m im Eckpunkt mit Null vorbestimmten Variablen
 Basisvariable: Die m im Eckpunkt nicht mit Null vorbestimmten nichtnegativen Variablen. Sie bilden die kanonische Form.

Optimalitätskriterien der Simplex-Methode:

Die Lösung ist optimal, falls alle $\delta_j(l) \geq 0$
 Begründung: Der Wert der Zielfunktion kann nur noch vergrößert, aber nicht mehr verkleinert werden.

→ 1. zulässige Lösung:

$$\begin{aligned} x_1 = x_2 &= 0 \text{ (NBV)} \\ x_{2+1} &= 10 \\ x_{2+2} &= 7 \\ x_{2+3} &= 18 \end{aligned}$$

Eckenwechsel mit Variablentausch

- benachbarter Eckpunkt
- bisherige NBV wird BV
- bisherige BV wird NBV

Auswahlregel:

1. Regel: Wähle diejenige NBV x_s als Eintrittsvariable, für die gilt:
 $\delta_s(l) = \min_j \{ \delta_j(l) \}$ für $\delta_j(l) < 0$
2. Regel: Wähle diejenige BV x_r als Austrittsvariable, für die gilt:
 $\frac{\beta_r(l)}{\alpha_{rs}(l)} = \min_j \left\{ \frac{\beta_i(l)}{\alpha_{is}(l)} \right\}$ für $\alpha_{is}(l) > 0$

Hier: $\min \left\{ \frac{10}{2}, \frac{7}{1} \right\} = 5$

Rechnerisch:

$$\begin{aligned} 1x_1 + 2x_2 + 1x_{2+1} &= 10 && \rightarrow x_1 \text{ ist } 0 \rightarrow x_{2+1} = 10 - 2x_2 \geq 0 && 0 \leq x_2 \leq 5 \\ 1x_1 + 1x_2 + 1x_{2+2} &= 7 && \rightarrow x_1 \text{ ist } 0 \rightarrow x_{2+2} = 7 - 1x_2 \geq 0 && 0 \leq x_2 \leq 7 \\ 3x_1 &+ 1x_{2+3} &= 18 && && \end{aligned}$$

Die stärkere Restriktion greift.
 Also ist x_{2+1} die Austrittsvariable.

1.KF	x_1	x_2	x_{2+1}	x_{2+2}	x_{2+3}	G	(β_j)	
x_{2+1}	1	2	1	0	0	0	10	I ₁
x_{2+2}	1	1	0	1	0	0	7	II ₁
x_{2+3}	3	0	0	0	1	0	18	III ₁
G	-3	-5	0	0	0	1	0	IV ₁
	↓							
2.KF	x_1	x_2	x_{2+1}	x_{2+2}	x_{2+3}	G	(β_j)	
x_2	1/2	1	1/2	0	0	0	5	I ₂ = I ₁ / 2
x_{2+2}	1/2	0	-1/2	1	0	0	2	II ₂ = II ₁ - 1I ₂
x_{2+3}	3	0	0	0	1	0	18	III ₂ = III ₁
G	-1/2	0	5/2	0	0	1	25	IV ₂ = IV ₁ + 5I ₂

→ 2. zulässige Lösung:

$$\begin{aligned} x_1 = x_{2+1} &= 0 \text{ (NBV)} \\ x_2 &= 5 \\ x_{2+2} &= 2 \\ x_{2+3} &= 18 \end{aligned}$$

2. Regel: $\frac{\beta_r(2)}{\alpha_{rs}(2)} = \min_j \left\{ \frac{5}{1/2}, \frac{1}{1/2}, \frac{18}{3} \right\} = 4 \Rightarrow x_r = x_{2+2}$

3.KF	x₁	x₂	x₂₊₁	x₂₊₂	x₂₊₃	G	(β_j)	
x ₂	0	1	1	-1	0	0	3	I ₃ = I ₂ - 1/2 II ₃
o/o x ₂₊₂	1	0	-1	2	0	0	4	II ₃ = II ₂ • 2
x ₂₊₃	0	0	3	-6	1	0	6	III ₃ = III ₂ - 3II ₃
G	0	0	2	1	0	1	27	IV ₃ = IV ₂ + 1/2 III ₃

→ 3. zulässige Lösung:

$x_1 = 4, x_2 = 3, x_{2+3} = 6$ (BV)
 $x_{2+1} = x_{2+2} = 0$ (NBV)

→ Optimalitätsbedingung erfüllt!

Zulässige optimale Lösung: $(x_1^o = 4, x_2^o = 3), G_{max} = 27$

$x_{2+1} = 0$
 $x_{2+2} = 0$
 $x_{2+3} = 6$

Maschine A und B sind voll ausgelastet.

Zwei-Phasen-Simplex-Methode:

Ein Verschnittproblem:

Breite der Papierrollen: 110 cm
 Bestellung: 6x 45 cm
 1x 65 cm

Zuschnittskombination:

Bestellte Breite	ZK1	ZK2	ZK3	ZK4
45 cm	1	2	1	0
65 cm	1	0	0	1
Verschnitt	0	20	65	45

Relevante Zuschnittskombinationen:
 der Verschnitt in der Breite ist kleiner als die kleinste bestellte Breite.

Das Verschnittmodell:

x₁: Anzahl der in der ZK1 zu zerschneidenden Standardrollen
 x₂: Anzahl der in der ZK2 zu zerschneidenden Standardrollen
 K: Summe der in der ZK1 und ZK2 zu zerschneidenden Standardrollen

Zielfunktion: $\min K = x_1 + x_2$
 Nebenbedingungen: $1x_1 + 2x_2 \geq 6$
 $1x_1 + 0x_2 \geq 1$
 $x_1, x_2 \geq 0$

2-Phasen-Simplex-Methode:

Anwendungsfall: Sie wird angewendet, wenn es nicht nur \leq -, sondern auch \geq - oder $=$ -Beschränkungen gibt.

1.Phase: Ausgehend von einer unzulässigen Lösung wird eine zulässige Lösung ermittelt. (falls diese existiert)

1) Problem in Standardform überführen

$$\begin{aligned} \min K = x_1 + x_2 &\rightarrow \min K = x_1 + x_2 \Rightarrow x_1 + x_2 + 1(-K) = 0 \\ 1x_1 + 2x_2 \geq 6 &\rightarrow 1x_1 + 2x_2 - 1x_{2+1} = 6 \\ 1x_1 + 0x_2 \geq 1 &\rightarrow 1x_1 + 0x_2 - 1x_{2+2} = 1 \\ x_1, x_2, x_{2+1}, x_{2+2} &\geq 0 \end{aligned}$$

	x_1	x_2	x_{2+1}	x_{2+2}	$-K$	
	1	2	-1	0	0	6
	1	0	0	-1	0	1
$-K$	1	1	0	0	1	0

keine kanonische Form

2) nicht negative künstliche Schlupfvariable x_{2+1}^* und x_{2+2}^* einführen.

Künstliche Schlupfvariablen: nichtnegative Variablen, die in der Standardform in einer \geq - oder $=$ -Beschränkung auf der linken Seite addiert werden.

	x_1	x_2	x_{2+1}	x_{2+2}	x_{2+1}^*	x_{2+2}^*	$-K$	
	1	2	-1	0	1	0	0	6
	1	0	0	-1	0	1	0	1
$-K$	1	1	0	0	0	0	1	0

Kanonische Form mit unzulässiger Lösung, weil alle künstlichen Schlupfvariablen positiv sind.

→ Zulässigkeit ist dann gegeben, wenn alle künstlichen Schlupfvariablen 0 sind.

3) Summe W der künstlichen Schlupfvariablen einführen.

Zielfunktion der 1. Phase: $\min W = 1x_{2+1}^* + 1x_{2+2}^*$

	x_1	x_2	x_{2+1}	x_{2+2}	x_{2+1}^*	x_{2+2}^*	$-K$	$-W$	
x_{2+1}^*	1	2	-1	0	1	0	0	0	6
x_{2+2}^*	1	0	0	-1	0	1	0	0	1
$-K$	1	1	0	0	0	0	1	0	0
$-W$	0	0	0	0	1	1	0	1	0

Beschränkungszeilen, die künstliche Schlupfvariablen enthalten, addieren und von der $-W$ -Zeile abziehen.

1. Regel in der 1. Phase auf $-W$ -Zeile anwenden.

1. Regel: $\delta_s(1) = \min_j \{-2, -2\} = -2 \Rightarrow x_s = x_1$

2. Regel: $\frac{\beta_r(1)}{\alpha_{r_1}(1)} = \min_j \left\{ \frac{6}{1}, \frac{1}{1} \right\} = 1 \Rightarrow x_r = x^*_{2+2}$

Anfangstabelle der 2-Phasen-Simplex-Methode:

1.KF	x_1	x_2	x_{2+1}	x_{2+2}	x^*_{2+1}	x^*_{2+2}	-K	-W		
x^*_{2+1}	1	2	-1	0	1	0	0	0	6	I ₁
x^*_{2+1}	1	0	0	-1	0	1	0	0	1	II ₂
-K	1	1	0	0	0	0	1	0	0	III ₃
-W	-2	-2	1	1	0	0	0	1	-7	IV ₄

2.KF	x_1	x_2	x_{2+1}	x_{2+2}	x^*_{2+1}	x^*_{2+2}	-K	-W		
x^*_{2+1}	0	2	-1	1	1	-1	0	0	5	I ₂ = I ₁ - II ₂
x_1	1	0	0	-1	0	1	0	0	1	II ₂ = II ₁
-K	0	1	0	1	0	-1	1	0	-1	III ₂ = III ₁ - II ₂
-W	0	-2	1	-1	0	2	0	1	-5	IV ₂ = IV ₁ + 2II ₂
3.KF	x_1	x_2	x_{2+1}	x_{2+2}	x^*_{2+1}	x^*_{2+2}	-K	-W		
x^*_2	0	1	-1/2	1/2	1/2	-1/2	0	0	5/2	I ₃ = I ₂ /2
x^*_1	1	0	0	-1	0	1	0	0	1	II ₃ = II ₂
-K	0	0	1/2	1/2	-1/2	-1/2	1	0	-7/2	III ₃ = III ₂ - I ₃
-W	0	0	0	0	1	1	0	1	0	IV ₄ = IV ₂ - 2I ₃

3. Eckpunkt: (zulässige Lösung, da $W = 0$)

$x_1 = 1, x_2 = 5/2$

$x_{2+1} = x_{2+2} = x^*_{2+1} = x^*_{2+2} = 0$ (Nichtbasisvariablen)

Mögliche Fälle:

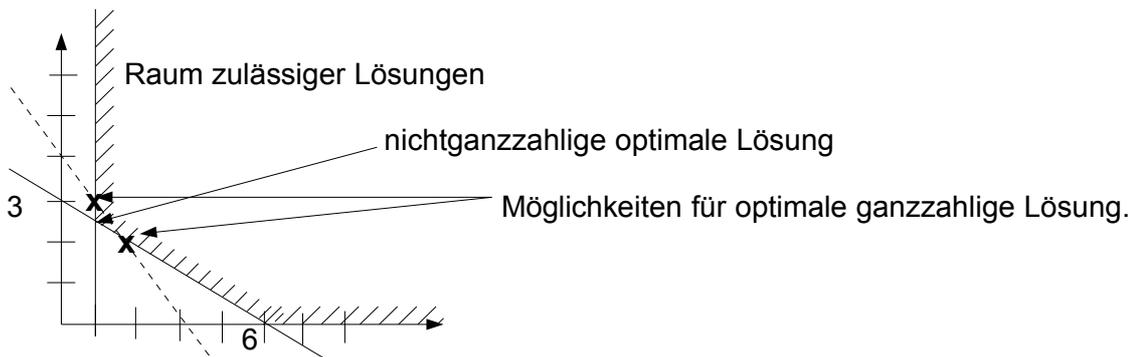
- a) $\min W > 0$ → mindestens eine KSV ist positiv → keine zulässige Lösung
- b) $\min W = 0$ → alle KSV sind 0 → zulässige Lösung → schon optimal?

2. Phase: Ausgehend von einer zulässigen Lösung der 1. Phase wird eine optimale Lösung ermittelt.

→ Vorgehen wie bei der einfachen Simplex-Methode

→ Lösung ist wegen $\delta(3) \geq 0$ bereits optimal

Allerdings haben wir die Beschränkung der Ganzzahligkeit. x_2 darf nicht $5/2$ betragen.
graphische Lösung:



Überprüfung mit Hilfe des Gomory-Algorithmus:

Der Gomory-Algorithmus zur Erzeugung einer ganzzahligen optimalen Lösung:

- Idee:**
- man hat eine nichtganzzahlige Lösung
 - in jeder Iteration wird eine zusätzliche Beschränkung konstruiert.
 - sie verkleinert den Raum der zulässigen Lösungen
 - sie schließt keine ganzzahlige Lösung aus.
 - es entsteht mindestens ein neuer optimaler Eckpunkt
 - so lange bis ein ganzzahliger optimaler Eckpunkt gefunden ist, oder sich herausstellt, dass es keine ganzzahlige optimale Lösung gibt.

Konstruieren der 1. zusätzlichen Beschränkung:

Regel: Wähle unter den nichtganzzahligen Basisvariablen die maximale und verwende die zugehörige Zeile zur Konstruktion.

(1) Zeile aussuchen und rausschreiben

$$0x_1 + 1x_2 - \frac{1}{2}x_{2+1} + \frac{1}{2}x_{2+2} + 0(-K) = \frac{5}{2}$$

(2) $\alpha_{ij}(l)$ und $\beta_i(l)$ in ganzzahlige und nichtganzzahlige Bestandteile zerlegen.

$$(0+0)x_1 + (1+0)x_2 + (-1 + \frac{1}{2})x_{2+1} + (0 + \frac{1}{2})x_{2+2} = 2 + \frac{1}{2}$$

(3) Umsortieren:

$$1x_2 - 1x_{2+2} - 2 = \frac{1}{2} - \left(\frac{1}{2}x_{2+1} + \frac{1}{2}x_{2+2} \right)$$

$\underbrace{\hspace{10em}}_{\geq 0}$
 $\underbrace{\hspace{10em}}_{\text{ganzzahlig}}$

(4) rechte Seite betrachten

$$\frac{1}{2} - \left(\frac{1}{2}x_{2+1} + \frac{1}{2}x_{2+2} \right) < 1 \quad \text{und ganzzahlig}$$

$$\frac{1}{2} - \left(\frac{1}{2}x_{2+1} + \frac{1}{2}x_{2+2} \right) \leq 0$$

$$-\frac{1}{2}x_{2+1} - \frac{1}{2}x_{2+2} \leq -\frac{1}{2}$$

(5) $-\frac{1}{2}x_{2+1} - \frac{1}{2}x_{2+2} + x_{2+3} = -\frac{1}{2}$ Gomoryvariable (Schlupfvariable) einführen

(zusätzliche Beschränkung)

Da x_{2+1} und x_{2+2} NBV sind, ist $x_{2+3} = -\frac{1}{2}$, das heißt, die nicht ganzzahlige Lösung ist unzulässig bzgl. der zusätzlichen Beschränkung.

Unzulässigkeit der negativen Gomoryvariable mit Hilfe der dualen Simplexmethode beheben:

Idee: Mit der Beseitigung der Negativität wird gleichzeitig versucht, die Ganzzahligkeit zu erreichen.

↓

4.KF	x₁	x₂	x₂₊₁	x₂₊₂	x₂₊₃	-K	
x₂	0	1	-1/2	1/2	0	0	5/2
x₁	1	0	0	-1	0	0	1
x₂₊₃	0	0	-1/2	-1/2	1	0	-1/2
-K	0	0	1/2	1/2	0	1	-7/2

←

Eckenwechsel:

Austrittsvariable x_r : steht mit der Gomoryvariable fest.

Eintrittsvariable x_s so wählen, dass die Optimalität erhalten bleibt.

$$\delta_j(l+1) = \delta_j(l) - \delta_s(l) \frac{\alpha_{rj}(l)}{\alpha_{rs}(l)}$$

$$\delta_j(l) - \delta_s(l) \cdot \frac{\alpha_{rj}(l)}{\alpha_{rs}(l)} \stackrel{!}{\geq} 0$$

$$\delta_j(l) \stackrel{!}{\geq} \delta_s(l) \cdot \frac{\alpha_{rs}(l)}{\alpha_{rj}(l)} \quad | : \alpha_{rj}(l) < 0$$

$$\frac{\delta_j(l)}{\alpha_{rj}(l)} \stackrel{!}{\leq} \frac{\delta_s(l)}{\alpha_{rs}(l)}$$

Wichtigste Formel der dualen Simplex-Methode:

$$\frac{\delta_s(l)}{\alpha_{rs}(l)} = \max_j \frac{\delta_j(l)}{\alpha_{rj}(l)} \quad \text{für } \alpha_{rj} < 0$$

Hier:
$$\frac{\delta_s(4)}{\alpha_{rs}(4)} = \max_j \left\{ \frac{1/2}{-1/2}, \frac{1/2}{-1/2} \right\} = -1 \Rightarrow x_2 = 2_{2+1}$$

	4.KF	x_1	x_2	x_{2+1}	x_{2+2}	x_{2+3}	-K		
	x_2	0	1	0	1	-1	0	3	$I_5 = I_4 + 1/2 III_5$
	x_1	1	0	0	-1	0	0	1	$II_5 = II_4$
%	x_{2+1}	0	0	0	1	-2	0	1	$III_5 = III_4 * (-2)$
	-K	0	0	0	0	1	1	-4	$IV_5 = IV_4 - 1/2 III_5$

5. Eckpunkt:

$$\begin{aligned} x_1 &= 1 \\ x_2 &= 3 \\ x_{2+1} &= 1 \\ x_{2+2} &= x_{2+3} = 0 \end{aligned}$$

optimale zulässige Lösung: $(\hat{x}_1=1, \hat{x}_2=3), \underline{K_{min}}=4$

Vergleich Simplex-Methode und duale Simplex-Methode:

SM: geht von einer **zulässigen**, nicht **optimalen** Lösung aus und erzeugt unter Beibehaltung der **Zulässigkeit** eine **optimale** Lösung.
 Erst **Eintrittsvariable** bestimmen, dann **Austrittsvariable**.
 hat beim Pivotelement ein **positives** Vorzeichen.

DSM: geht von einer **optimalen**, nicht **zulässigen** Lösung aus und erzeugt unter Beibehaltung der **Optimalität** eine **zulässige** Lösung.
 Erst **Austrittsvariable** bestimmen, dann **Eintrittsvariable**.
 hat beim Pivotelement ein **negatives** Vorzeichen.