

Lösungen zum Buch  
Einführung in die Optimierung  
Konzepte, Methoden und Anwendungen

Christian Grimme      Jakob Bossek

Version vom 3. Mai 2018

**Hinweis:** Kein Buch ist frei von Fehlern. Daher finden sich mit hoher Wahrscheinlichkeit auch in den folgenden Musterlösungen einige Fehler. Fehlerberichte bitte via E-Mail an [info@optimierung-grundlagen.de](mailto:info@optimierung-grundlagen.de) schicken.

## Lösungen zu Kapitel 2

**2.1.** Ordnen Sie die folgenden Funktion aufsteigend bzgl. ihrer Größenordnung:

$$f_1 = \log(3n!), f_2 = n^4 - 3n, f_3 = n!, f_4 = n \log(n)^2 + 72n, f_5 = 14n, f_6 = 4n^{n-1}.$$

$$\text{Es gilt: } f_5 \leq f_1 \leq f_4 \leq f_2 \leq f_3 \leq f_6.$$

**2.2.** Zeigen Sie:  $4^n = O(n!)$

Wir müssen zeigen, dass eine Konstante  $c > 0$  existiert und ein  $n_0 \in \mathbb{N}$  derart, dass  $4^n \leq c \cdot n!$  gilt  $\forall n \geq n_0$ . Hierzu schätzen wir geschickt ab:

$$\begin{aligned} 4^n &= 4 \cdot 4 \cdot 4 \cdot \underbrace{4 \cdot \dots \cdot 4}_{(n-3) \text{ mal}} \\ &\leq 64 \cdot 1 \cdot 2 \cdot 3 \cdot \underbrace{4 \cdot \dots \cdot 4}_{(n-3) \text{ mal}} \\ &\leq 64 \cdot 1 \cdot 2 \cdot 3 \cdot 4 \cdot 5 \cdot \dots \cdot n \\ &= 64 \cdot n! \end{aligned}$$

Wähle also z. B.  $c = 64$  und  $n = 1$ .

**2.3.** Zeigen Sie, dass für Funktionen  $f, g, h : \mathbb{N} \rightarrow \mathbb{R}^+$  die folgenden Eigenschaften gelten:

- a)  $f \in \Theta(f)$
- b)  $f \in \Theta(g) \Rightarrow g \in \Theta(f)$
- c)  $f \in \Theta(g) \vee g \in \Theta(h) \Rightarrow f \in \Theta(h)$

Seien  $f, g, h : \mathbb{N} \rightarrow \mathbb{R}^+$  Funktionen.

- ad a) Wir wählen  $c_1 = c_2 = 1$  sowie  $n_0 = 1$ . Dann gilt  $c_1 f \leq f \leq c_2 f$ , also  $f \in \Theta(f)$ .
- ad b) Sei  $f \in \Theta(g)$ . D. h.  $\exists c_1, c_2 \in \mathbb{R}^+, n_0 \in \mathbb{N}$  derart, dass  $c_1 g \leq f \leq c_2 g \forall n \geq n_0$ . Aus  $c_1 g \leq f$  folgt nun durch einfache Umformung ( $c_1 > 0$ )  $g \leq \frac{1}{c_1} f$ . Analog erhalten wir  $g \geq \frac{1}{c_2} f$ . Insgesamt also

$$\underbrace{\frac{1}{c_2}}_{=: \tilde{c}_1} f \leq g \leq \underbrace{\frac{1}{c_1}}_{=: \tilde{c}_2} f \Leftrightarrow \tilde{c}_1 f \leq g \leq \tilde{c}_2 f,$$

woraus  $g \in \Theta(f)$  folgt.

- ad c) Aus  $f \in \Theta(g)$  und  $g \in \Theta(h)$  folgt:  $\exists c_1, c_2 \in \mathbb{R}^+, n_1 \in \mathbb{N}$  mit  $c_1 g \leq f \leq c_2 g$  für alle  $n \geq n_1$  sowie  $\exists d_1, d_2 \in \mathbb{R}^+, n_2 \in \mathbb{N}$  mit  $d_1 h \leq g \leq d_2 h$  für alle  $n \geq n_2$ . Wir setzen  $e_1 = c_1 d_1$  und  $e_2 = c_2 d_2$ . Damit gilt nämlich

$$e_1 h = c_1 d_1 h \leq c_1 g \leq f \leq c_2 g \leq c_2 d_2 h = e_2 h,$$

womit die Behauptung gezeigt ist.

2.4. Zeigen Sie:  $\log(n!) = O(n \log(n))$ .

Hier nutzen wir einfach die Rechenregeln für Logarithmen:  $\log(a \cdot b) = \log(a) + \log(b)$ . Damit gilt die folgende Ungleichungskette:

$$\log(n!) = \log\left(\prod_{k=1}^n k\right) = \sum_{k=1}^n \log(k) \leq \sum_{k=1}^n \log(n) = n \log(n).$$

Diese Ungleichung gilt für alle  $n \in \mathbb{N}$ . Wir können daher  $c = 1$  und  $n_0 = 1$  wählen.

2.5. Konstruieren Sie eine Turingmaschine, die eine binäre Zeichenkette, d. h. eine Zeichenkette über dem Alphabet  $\Sigma = \{0, 1\}$  invertiert.

Beispiel: aus 1 0 0 1 0 0 1 wird 0 1 1 0 1 1 0.

Der Lese-/Schreibkopf stehe zu Beginn auf dem ersten Element der Eingabe und am Ende der Operation ebenfalls auf dem ersten Element der Ausgabe.

Wenden Sie nach Konstruktion der Turingmaschine diese auf die Zeichenkette 1 0 1 0 an und zeigen Sie die einzelnen Schritte und Übergänge.

**Lösung:** Die Turingmaschine lässt sich wie folgt konstruieren:

- $Z = \{z_0, z_1, e\}$
- $\Pi = \{0, 1, \square\}$
- $\Sigma = \{0, 1\}$
- $E = \{e\}$ .

Die Zustandübergangsfunktion  $\delta$  lautet:

	0	1	$\square$
$z_0$	$(z_0, 1, \rightarrow)$	$(z_0, 0, \rightarrow)$	$(z_1, \square, \leftarrow)$
$z_1$	$(z_1, 0, \leftarrow)$	$(z_1, 1, \leftarrow)$	$(e, \square, \rightarrow)$
$e$	—	—	—

Anwendung auf Beispiel: 1 0 1 0

$\square$	<u>1</u>	0	1	0	$\square$	$(z_0, 1) \rightarrow (z_0, 0, \rightarrow)$
$\square$	0	<u>0</u>	1	0	$\square$	$(z_0, 0) \rightarrow (z_0, 1, \rightarrow)$
$\square$	0	1	<u>1</u>	0	$\square$	$(z_0, 1) \rightarrow (z_0, 0, \rightarrow)$
$\square$	0	1	0	<u>0</u>	$\square$	$(z_0, 0) \rightarrow (z_0, 1, \rightarrow)$
$\square$	0	1	0	1	<u><math>\square</math></u>	$(z_0, \square) \rightarrow (z_1, \square, \leftarrow)$
$\square$	0	1	0	<u>1</u>	$\square$	$(z_1, 1) \rightarrow (z_1, 1, \leftarrow)$
$\square$	0	1	<u>0</u>	1	$\square$	$(z_1, 0) \rightarrow (z_1, 0, \leftarrow)$
$\square$	0	<u>1</u>	0	1	$\square$	$(z_1, 1) \rightarrow (z_1, 1, \leftarrow)$
$\square$	<u>0</u>	1	0	1	$\square$	$(z_1, 0) \rightarrow (z_1, 0, \leftarrow)$
$\square$	<u>0</u>	1	0	1	$\square$	$(z_1, \square) \rightarrow (e, \square, \rightarrow)$
$\square$	<u>0</u>	1	0	1	$\square$	$e$

2.6. In dieser Aufgabe soll es darum gehen, ein Programm zur Berechnung der Summe der ersten  $n$  natürlichen Zahlen in Brainfuck zu schreiben. Lesen Sie hierzu als ersten Schritt mittels ‚,‘ eine natürliche Zahl in die erste Byte-Zelle des Datenbandes ein.

**Lösungsvorschlag:** Zu berechnen ist die Summe der ersten  $n$  natürlichen Zahlen:  $\sum_{k=1}^n k$ . Wir müssen also wiederholt addieren. Günstig ist es die Summe absteigend zu berechnen, also  $n + (n - 1) + \dots + 1$ . Befindet sich  $n$  zunächst in der ersten Byte-Zelle, so genügt es nicht eine einfache Addition durchzuführen. Wir müssen ja auch noch  $n$  iterativ dekrementieren. Wir benötigen zur Lösung demnach mindestens drei Byte-Zellen: eine enthält das aktuelle  $k$ , eine temporäre Zelle zur Zwischenspeicherung und eine Ergebniszelle. Das Programm ist damit recht kurz darstellbar:

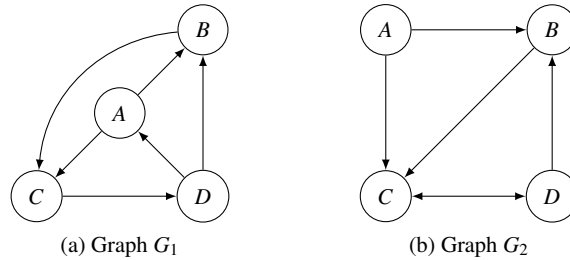
Das Programm gliedert sich in drei Teile. Zunächst lesen wir  $n$  in die erste Datenzelle ein. Anschließend erfolgt die Summation, gefolgt von der Ausgabe. Die Summation läuft wie folgt ab: In der ersten Datenzelle liegt das  $n$ , welches wir in jedem Schritt der äußeren Schleife zunächst in die zweite Datenzelle verschieben (erste innere Schleife). Anschließend addieren wir in der zweiten inneren Schleife die zweite Zelle sowohl auf die dritte (Ergebnis)Zelle sowie auch zurück auf die erste Zelle bevor wir abschließend den Wert der ersten Zelle um 1 verkleinern. Nachfolgend ist nochmals der Quellcode mit Kommentaren dargestellt.

Angenommen wir lesen  $n = 3$  ein. Dann sieht das ganze wie folgt aus (dargestellt sind jeweils nur die ersten drei Zellen).

DZ1	DZ2	DZ3	Kommentar
3	0	0	Nach dem Einlesen
0	3	0	Nach der 1. inneren Schleife
3	0	3	Nach der 2. inneren Schleife
2	0	3	Nach dem 1. Durchlauf der äußeren Schleife
0	2	3	Nach der 1. inneren Schleife
2	0	5	Nach der 2. inneren Schleife
1	0	5	Nach dem 2. Durchlauf der äußeren Schleife
0	1	5	Nach der 1. inneren Schleife
1	0	6	Nach der 2. inneren Schleife
0	0	6	Nach dem 3. Durchlauf der äußeren Schleife

## Lösungen zu Kapitel 3

**3.1.** Betrachten Sie die beiden Graphen in Abbildung 3.24. Entscheiden Sie jeweils, ob der Graph (schwach/stark) zusammenhängend ist.



**Abb. 3.24** Graphen zu Aufgabe 3.

**Lösungsvorschlag:**  $G_1$  ist stark zusammenhängend ( $\exists$  gerichteter Weg von jedem Knoten zu jedem anderen Knoten) und damit auch schwach zusammenhängend. Der Graph  $G_2$  ist schwach zusammenhängend. Er ist jedoch nicht stark zusammenhängend, da Knoten  $A$  über keinen gerichteten Pfad erreichbar ist.

**3.2.** Gegeben sei der ungerichtete Graph mit folgender Inzidenzmatrix  $B$ , wobei  $b_{ij} = 1$  bedeutet, dass  $v_i \in e_j$  ist:

$$B = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

- Bestimmen Sie aus der Matrix den maximalen Knotengrad und geben Sie für jeden Knoten den Grad an.
- Zeichnen Sie den Graphen und geben Sie die Adjazenzmatrix des Graphen an.
- Aus wie vielen "Zusammenhangskomponenten" besteht der Graph?

**Lösungsvorschlag:**

- Jeder Knoten hat einen Grad. Dieser kann direkt als Zeilensumme der Inzidenzmatrix abgelesen werden:  $A = 2, B = 2, C = 3, D = 4, E = 3, F = 5, G = 1, H = 1, I = 1$ . Damit ist der maximale Knotengrad 5.

b) Der Graph ist exemplarisch in Abbildung 3.25 dargestellt. Die Adjazenzmatrix ("Knoten zu Knoten") ist gegeben durch:

$$A = \begin{pmatrix} 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{pmatrix}.$$

c) Der Graph besteht folglich aus zwei Zusammenhangskomponenten (siehe Abbildung 3.25)

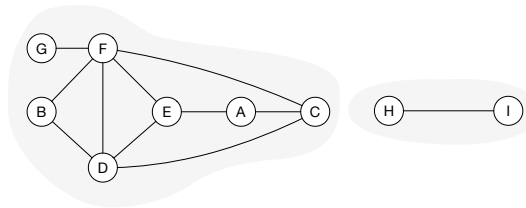


Abb. 3.25 Graph zur Inzidenzmatrix  $B$ .

3.3. Bestimmen Sie mit dem *Algorithmus von Dijkstra* den kürzesten Weg vom Startknoten 1 zum Zielknoten 4 für den Graphen aus Abbildung 3.26. Gehen Sie dabei wie in der Vorlesung vor und geben Sie alle Zwischenschritte an.

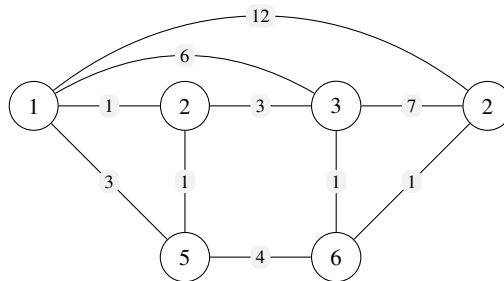


Abb. 3.26 Graph zu Aufgabe 2.

**Beispiellösung:** Die Anwendung des Dijkstra-Algorithmus ausgehend von Knoten 1 resultiert in folgendem tabellarischem Optimierungspfad.

Iteration	1		2		3		4		5	
Knoten	dist	vor	dist	vor	dist	vor	dist	vor	dist	vor
1	0	1	0	1	0	1	0	1	0	1
2	1	1	1	1	1	1	1	1	1	1
3	6	1	4	2	4	2	4	2	4	2
4	12	1	12	1	12	1	11	3	8	6
5	3	1	2	2	2	2	2	2	2	2
6					6	5	5	3	5	3

Damit erhalten wir als kürzesten Pfad von Knoten 1 zu Knoten 4:  $1 \rightarrow 2 \rightarrow 3 \rightarrow 6 \rightarrow 4$ .

**3.4.** Zeigen Sie: das Kürzeste-Wege-Problem erfüllt das *Bellman'sche Optimalitätsprinzip*, d. h. führt der kürzeste Weg von  $A$  nach  $B$  über  $C$ , so müssen auch die Teilwege von  $A$  nach  $C$  und von  $C$  nach  $B$  die kürzesten sein.

**Beispiellösung:** Wir führen einen einfachen Beweis durch Widerspruch. Sei der kürzeste Weg  $p$  von  $A$  nach  $B$  in einem Graphen mit Zwischenstop  $C$  gegeben. Wir nehmen o.B.d.A. an, dass der Teilweg  $p_1$  von  $A$  nach  $C$  nicht minimal sei. Dann existiert aber ein kürzerer Weg  $p'_1$  von  $A$  nach  $C$  mit  $c(p'_1) < c(p_1)$ . Ersetzen wir nun  $p_1$  in  $p$  durch  $p'_1$ , so ist der resultierende Pfad  $p'$  kürzer als  $p$ . Dies ist ein Widerspruch zur Annahme.

**3.5.** Student Christian soll das Kürzeste-Wege-Problem auf einem Graphen  $G = (V, E, c)$  mit negativen Kantengewichten (aber keinen negativen Kreisen) lösen. Christian erinnert sich an die Vorlesung. Dort wurde erklärt, dass der Dijkstra-Algorithmus nur auf Graphen mit ausschließlich positiven Kantengewichten  $c(e) \geq 0, \forall e \in E$  eine optimale Lösung garantieren kann. Christian hat jedoch folgende Idee: Er modifiziert den Graphen durch Addition des Absolutbetrags des kleinsten Kantengewichtes  $c_{\min}$  zu  $G'$ . Damit sind alle Kanten positiv. Nun wendet Christian den Dijkstra-Algorithmus auf den modifizierten Graphen an, und subtrahiert anschließend  $c_{\min}$  wieder. Führt dieses Vorgehen in jedem Graphen mit negativen Kantengewichten zu einer optimalen Lösung?

**Beispiellösung:** Nein, das Verfahren findet im allgemeinen nicht die kürzesten Wege. Betrachte das Gegenbeispiel in Abbildung 3.27. Offensichtlich ist im Originalgraphen der obere  $u - v$  Pfad minimal mit Gesamtkosten  $-2$ . Nach Modifikation durch Addition von  $|-1.5|$  ist jedoch der untere direkte Pfad minimal.

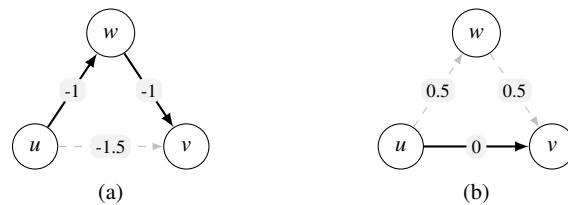


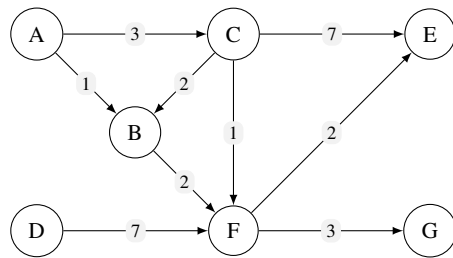
Abb. 3.27 Gegenbeispiel für Christians Idee.

**3.6.** Zeigen oder widerlegen Sie: Der Dijkstra-Algorithmus bestimmt für jedes Paar von Startknoten  $s$  und Knoten  $v \in V \setminus \{s\}$  einen kürzesten Pfad mit minimaler Anzahl von Kanten.

**Beispiellösung:** Diese Aussage ist falsch. Betrachte etwa das folgende minimale Gegenbeispiel. Der kürzeste Weg  $A - C$ -Weg ist hier  $A \rightarrow B \rightarrow C$  mit Gesamtkosten 3. Der hinsichtlich der Anzahl Kanten kürzeste Weg führt über die direkte Kante von  $A$  nach  $C$ .

**3.7.** Gegeben sei ein gerichteter Graph  $G = (V, E, c)$  mit nicht-negativer Gewichtsfunktion  $c : E \rightarrow \mathbb{R}^+$ . Seien  $S, D \subseteq V$  Mengen von Knoten mit  $S \cap D = \emptyset$ . Wir interessieren uns für den kürzesten Weg zwischen Knoten aus  $S$  und Knoten aus  $D$ .

- Wie können wir dieses Problem mit Hilfe von Dijkstra naiv lösen?
- Wie kann man dieses Problem effizient unter Zuhilfenahme von Dijkstra lösen? Hinweis: Eine Modifikation des Graphen ist notwendig.
- Nutze den Algorithmus aus b) zur Lösung des beschriebenen Problems auf dem folgenden kantengewichteten Graphen  $G = (V, E, c)$  mit  $S = \{A, D\}$  und  $D = \{E, G\}$ .

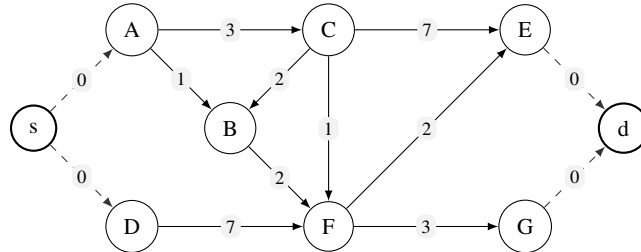


**Beispiellösung:**

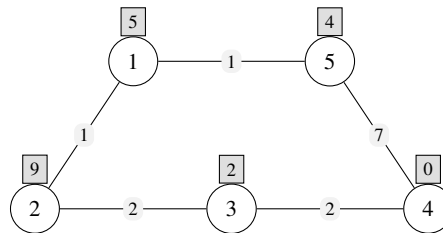
- Der naive Ansatz startet den Dijkstra-Algorithmus sequenziell  $|S|$ -mal, also einmal für jeden Startknoten  $v \in S$  und wählt unter allen bestimmten kürzesten Wegen mit Endknoten in  $D$  den kürzesten. Da  $|S| = O(|V|)$  ist die Laufzeit asymptotisch in  $O(|V|) \cdot O(|V|^2) = O(|V|^3)$ .
- Die Idee besteht darin zwei zusätzliche Knoten  $s, d$  hinzuzufügen. Der Knoten  $s$  ist dabei über eine gerichtete Kante mit jedem Knoten  $v \in S$  verbunden. Analog erhält jeder Knoten  $v \in D$  eine gerichtete Kante mit Zielknoten  $d$ . Sämtliche neue Kanten erhalten Kosten 0. Der modifizierte Graph ist folglich  $G' = (V', E', c)$  mit  $V' = V \cup \{s, d\}$ ,  $E' = E \cup \{(s, v) \mid v \in S\} \cup \{(v, d) \mid v \in D\}$  sowie  $c(v, w) = 0$  für  $v = s$  bzw.  $w = d$ . In  $G'$  suchen wir nun einen kürzesten  $s - d$ -Weg  $P' = ss'_1 \dots d'_j d$ . Der kürzeste gesuchte Weg ist dann  $P = s'_1 \dots d'_j$  mit  $s_i \in S$  und  $d_j \in D$ .
- Der modifizierte Graph ist  $G' = (\{A, \dots, G, s, d\}, E \cup \{(s, A), (s, D), (E, d), (G, d)\})$  (siehe Abbildung). Die Anwendung von Dijkstra liefert den kürzesten  $s - d$ -



Weg  $(s, A, B, F, E, d)$  mit Kosten 5. Der gesuchte Weg ist damit  $(A, B, F, E)$  mit  $A \in S$  und  $E \in D$ . Die Kosten liegen natürlich auch bei 5.



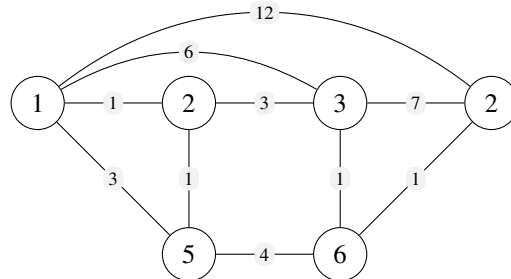
**3.8.** Bestimmen Sie mit dem  $A^*$ -Algorithmus den kürzesten Weg von Knoten 1 zu Knoten 4 für den gegebenen Graphen der folgenden Abbildung (Die Werte der zu verwendenden Heuristik  $h$  stehen jeweils in den eckigen grauen Kästen. Zum Beispiel ist  $h(1) = 5$ .) Was fällt Ihnen auf? Erklären Sie den Grund für Ihre Beobachtung.



**Beispiellösung:** Die Anwendung des  $A^*$ -Algorithmus ergibt die folgende Tabelle. Offensichtlich stoppt die Abarbeitung der Knoten in der zweiten Iteration, da der Zielknoten bereits gefunden wurde. Ein Blick auf den Graphen offenbart jedoch, dass der gefundene Pfad  $1 \rightarrow 5 \rightarrow 4$  mit Kosten  $1 + 7 = 8$  nicht optimal ist. Grund hierfür ist die Unzulässigkeit der Schätzfunktion/Heuristik. Diese darf die wahren Kosten nicht überschätzen. Für Knoten 2 ist diese Forderung nicht erfüllt, da  $h(2) = 9 > 4 = h^*(v)$ , wobei  $h^*(v)$  die wahren optimalen Kosten von Knoten 2 zum Zielknoten beschreibt.

Iteration	1			2		
	dist	f	vor	dist	f	vor
1	1	6	1	1	6	1
2	1	10	1	1	10	1
3						
4				8	8	5
5	1	5	1	1	5	1

3.9. Lösen Sie das *All-Pairs-Shortest-Path Problem (APSP)* für den folgenden Graphen mit dem *Algorithmus von Floyd*.

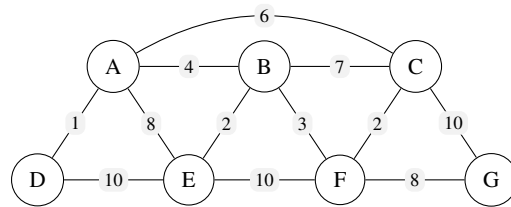


**Beispiellösung:** Im Gegensatz zum Dijkstra-Algorithmus berechnet der Algorithmus von Floyd oder auch Algorithmus von Floyd-Warshall direkt alle paarweisen kürzesten Wege. Hierzu pflegt der Algorithmus dem Prinzip der *dynamischen Programmierung* folgend eine Tabelle bzw. (Bi-)Matrix  $A$  (auch *Wegematrix* genannt). Die Komponente  $A_{ij}$  enthält ein 2-Tupel mit den Kosten von  $i$  nach  $j$  und dem Vorgängerknoten von  $j$ . Zu Beginn sind nur jene Kosten ungleich unendlich, für die eine direkte Verbindung (also eine Kanten im Graphen zwischen den entsprechenden Knoten) existiert. Anschließend wird iterativ geprüft, ob sich Wege von  $i$  nach  $j$  über einen Knoten  $k$  kürzen lassen.

	1	2	3	4	5	6
1	(0, 1)	(1, 1)	(6, 1)	(12, 1)	(3, 1)	( $\infty$ , -)
2	(1, 2)	(0, 2)	(3, 2)	( $\infty$ , -)	(1, 2)	( $\infty$ , -)
Schritt 0: 3	(6, 3)	(3, 3)	(0, 3)	(7, 3)	( $\infty$ , -)	(1, 3)
4	(12, 4)	( $\infty$ , -)	(7, 4)	(0, 4)	( $\infty$ , -)	(3, 4)
5	(3, 5)	(1, 5)	( $\infty$ , -)	( $\infty$ , -)	(0, 5)	(4, 5)
6	( $\infty$ , -)	( $\infty$ , -)	(1, 6)	(3, 6)	(4, 6)	(0, 6)
	1	2	3	4	5	6
1	(0, 1)	(1, 1)	(6, 1)	(12, 1)	(3, 1)	( $\infty$ , -)
2	(1, 2)	(0, 2)	(3, 2)	(13, 1)	(1, 2)	( $\infty$ , -)
Schritt 1: 3	(6, 3)	(3, 3)	(0, 3)	(7, 3)	(9, 1)	(1, 3)
4	(12, 4)	(13, 1)	(7, 4)	(0, 4)	(15, 1)	(3, 4)
5	(3, 5)	(1, 5)	(9, 1)	(15, 1)	(0, 5)	(4, 5)
6	( $\infty$ , -)	( $\infty$ , -)	(1, 6)	(3, 6)	(4, 6)	(0, 6)
	1	2	3	4	5	6
1	(0, 1)	(1, 1)	(4, 2)	(12, 1)	(2, 2)	( $\infty$ , -)
2	(1, 2)	(0, 2)	(3, 2)	(13, 1)	(1, 2)	( $\infty$ , -)
Schritt 2: 3	(4, 2)	(3, 3)	(0, 3)	(7, 3)	(4, 2)	(1, 3)
4	(12, 4)	(13, 1)	(7, 4)	(0, 4)	(14, 2)	(3, 4)
5	(2, 2)	(1, 5)	(4, 2)	(14, 2)	(0, 5)	(4, 5)
6	( $\infty$ , -)	( $\infty$ , -)	(1, 6)	(3, 6)	(4, 6)	(0, 6)
	1	2	3	4	5	6
1	(0, 1)	(1, 1)	(4, 2)	(11, 3)	(2, 2)	(5, 3)
2	(1, 2)	(0, 2)	(3, 2)	(10, 3)	(1, 2)	(4, 3)
Schritt 3: 3	(4, 2)	(3, 3)	(0, 3)	(7, 3)	(4, 2)	(1, 3)
4	(11, 3)	(10, 3)	(7, 4)	(0, 4)	(11, 3)	(3, 4)
5	(2, 2)	(1, 5)	(4, 2)	(11, 3)	(0, 5)	(4, 5)
6	(5, 3)	(4, 3)	(1, 6)	(3, 6)	(4, 6)	(0, 6)

	1	2	3	4	5	6	
Schritt 4:	1	(0, 1)	(1, 1)	(4, 2)	(11, 3)	(2, 2)	(5, 3)
	2	(1, 2)	(0, 2)	(3, 2)	(10, 3)	(1, 2)	(4, 3)
	3	(4, 2)	(3, 3)	(0, 3)	(7, 3)	(4, 2)	(1, 3)
	4	(11, 3)	(10, 3)	(7, 4)	(0, 4)	(11, 3)	(3, 4)
	5	(2, 2)	(1, 5)	(4, 2)	(11, 3)	(0, 5)	(4, 5)
	6	(5, 3)	(4, 3)	(1, 6)	(3, 6)	(4, 6)	(0, 6)
Schritt 5:	1	(0, 1)	(1, 1)	(4, 2)	(11, 3)	(2, 2)	(5, 3)
	2	(1, 2)	(0, 2)	(3, 2)	(10, 3)	(1, 2)	(4, 3)
	3	(4, 2)	(3, 3)	(0, 3)	(7, 3)	(4, 2)	(1, 3)
	4	(11, 3)	(10, 3)	(7, 4)	(0, 4)	(11, 3)	(3, 4)
	5	(2, 2)	(1, 5)	(4, 2)	(11, 3)	(0, 5)	(4, 5)
	6	(5, 3)	(4, 3)	(1, 6)	(3, 6)	(4, 6)	(0, 6)
Schritt 6:	1	(0, 1)	(1, 1)	(4, 2)	(8, 6)	(2, 2)	(5, 3)
	2	(1, 2)	(0, 2)	(3, 2)	(7, 6)	(1, 2)	(4, 3)
	3	(4, 2)	(3, 3)	(0, 3)	(4, 6)	(4, 2)	(1, 3)
	4	(8, 6)	(7, 6)	(4, 6)	(0, 4)	(7, 6)	(3, 4)
	5	(2, 2)	(1, 5)	(4, 2)	(7, 6)	(0, 5)	(4, 5)
	6	(5, 3)	(4, 3)	(1, 6)	(3, 6)	(4, 6)	(0, 6)

**3.10.** Gegeben sei der kantengewichtete, ungerichtete Graph  $G = (V, E, c)$  aus Abbildung 3.28. Bestimmen Sie einen minimalen Spannbaum  $T$  zu  $G$  mit a) dem Algorithmus von Kruskal sowie b) Prim's Algorithmus.



**Abb. 3.28** Ungerichteter Graph  $G = (V, E, c)$ .

**Beispiellösung:** Der resultierende (eindeutige) minimale Spannbaum ist in Abbildung 3.29 dargestellt.

- Kruskal:** es werden nacheinander die Kanten  $\{A, D\}$ ,  $\{B, E\}$ ,  $\{C, F\}$ ,  $\{B, F\}$ ,  $\{A, B\}$  sowie  $\{F, G\}$  selektiert<sup>6</sup>.
- Prim:** Wir bilden den Spannbaum ausgehend von Knoten A und erweitern den zunächst trivialen Spannbaum durch die Kanten  $\{A, D\}$ ,  $\{A, B\}$ ,  $\{B, E\}$ ,  $\{B, F\}$ ,  $\{C, F\}$  und schließlich  $\{F, G\}$ .

<sup>6</sup> In welcher konkreten Reihenfolge die Kanten  $\{B, E\}$  und  $\{C, F\}$  hinzugefügt werden hängt vom verwendeten Sortieralgorithmus ab.

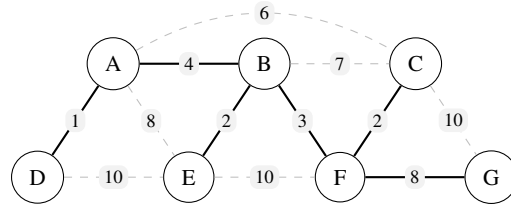


Abb. 3.29 Spannbaum nach Prim.

3.11. Sind minimale Spannäume eindeutig, wenn wir keinerlei Einschränkungen für die Kostenfunktion fordern? Wenn ja, dann führen Sie einen Beweis. Andernfalls konstruieren Sie ein Gegenbeispiel.

**Beispiellösung:** Minimale Spannäume sind in der Regel nicht eindeutig. Die Eindeutigkeit ist nur bei Injektivität der Kostenfunktion gegeben, d.h. in anderen Worten wenn die Kantengewichte/-kosten paarweise verschieden sind. Ein Gegenbeispiel ist in Abb. 3.30 gegeben.

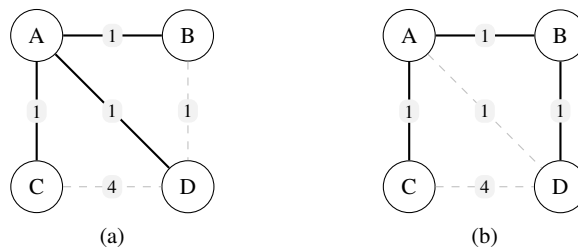


Abb. 3.30 Gegenbeispiel für Eindeutigkeit des minimalen Spannbaums.

3.12. Sei  $G = (V, E)$  ein ungerichteter Graph. Zeigen Sie: Ist  $e \in E$  eine Kante von  $G$ , die in keinem Kreis liegt, so ist  $e$  in jedem Spannbaum von  $G$  enthalten.

**Beispiellösung:** Wir führen einen Beweis durch Widerspruch.

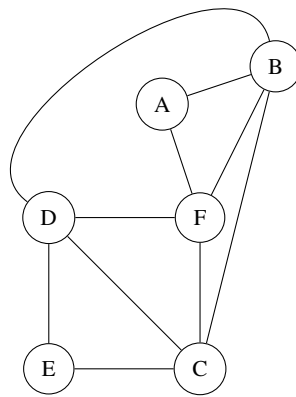
*Beweis.* Sei  $G = (V, E)$  ein ungerichteter Graph und  $e = \{u, v\} \in E$  in keinem Kreis enthalten. Angenommen es gibt einen Spannbaum  $T$  von  $G$ , der  $e$  nicht enthält. Da  $T$  ein Spannbaum ist, ist  $T$  insbesondere zusammenhängend. Damit gibt es einen Weg von jedem Knoten zu jedem anderen Knoten, also insbesondere auch einen Weg von  $u$  nach  $v$ . Damit gibt es aber in  $G$  zwei unterschiedliche Wege von  $u$  nach  $v$  und damit einen Kreis, was einen Widerspruch zu Annahme ergibt.

Formaler: Sei  $T = (V, E')$  ein Spannbaum, der  $e$  nicht enthält. Definiere  $G' = (V, E'')$  mit  $E'' = E' \cup \{e\}$ . Damit ist  $G'$  nicht kreisfrei; es gibt in  $G'$  einen Kreis, der  $e$  enthält. Sei  $C$  dieser Kreis. Nun gilt jedoch

$$C' \subseteq E' \cup \{e\} \subseteq E$$

und damit ist der Kreis  $C$  auch in  $G$  enthalten. Das ist ein Widerspruch zur Annahme, dass  $e$  in keinem Kreis von  $G$  liegt.

**3.13.** Ein Postbote ist für die Zustellung der Post in dem in Abbildung 3.31 gegebenen Straßennetz zuständig. Er startet im Postdepot (Knoten  $D$ ) und muss in jeder Straße Briefe zustellen. Er möchte die zurückgelegte Strecke optimieren, um keine unnötigen Wege zu gehen und am Ende wieder im Depot ankommen. Welches Problem liegt hier vor? Ist es lösbar? Wenn ja, bestimmen Sie eine optimale Lösung für den Postboten.



**Abb. 3.31** Arbeitsgebiet des Postboten.

**Lösungsvorschlag:** Der Postbote ist offensichtlich an einer Eulertour im Straßennetz  $G = (V, E)$  interessiert. Glücklicherweise hat - wie man sich leicht überzeugt - jeder Knoten des Graphen geraden Grad. Somit ist der Graph *eulersch* und wir können uns daran machen den *Algorithmus von Hierholzer* anzuwenden um eine optimale Route für den Postboten zu bestimmen. Der Algorithmus traversiert den Graphen von einem beliebigen Startknoten aus bis ein Kreis/Zyklus  $Z \subseteq E$  gefunden ist. Sämtliche Kanten des Zyklus werden anschließend entfernt und es wird auf dem Graphen  $G' = (V, E \setminus Z)$  ein weiterer Kreis gesucht. Der nächste Startknoten wird aus der Liste aller Knoten gewählt, welche noch inzident zu mindestens einer Kante sind. Der neue Kreis wird an der Stelle des Startknotens in den neuen bestehenden Kreis eingepflegt. Dieses Prozedere wird fortgeführt bis alle Kanten abgearbeitet sind.

Wir beginnen im Depot (Knoten  $D$ ). Allerdings ist der Startpunkt für die Optimierung irrelevant. Nun suchen wir (in einer Implementierung würde man hier vermutlich eine modifizierte Tiefensuche benutzen) ausgehend von  $D$  einen Kreis, der keine Kante zweifach enthält. Hier gibt es nun mehrere Möglichkeiten. Der tatsächliche Kreis hängt von der Repräsentation des Graphen im Programm und

der Implementierung der Tiefensuche ab. Hier nehmen wir an der Algorithmus findet zunächst den Kreis  $Z_1 = (D, C, F, D)$ . Im nächsten Schritt wählen wir den ersten Knoten in  $Z_1$ , welcher noch unbesuchte, inzidente Kanten hat (das ist hier  $D$ ) und suchen davon ausgehend erneut einen Kreis. Wir finden etwa den Kreis  $Z_2 = (D, E, C, B, D)$ . Schließlich wählen wir den Knoten  $B$  und finden den Kreis  $Z_3 = B, A, F, D$ . Die Kreise sind exemplarisch in Abb. 3.32 dargestellt. Damit sind alle Kanten abgearbeitet und es bleibt nur noch das zusammensetzen der Kreise:

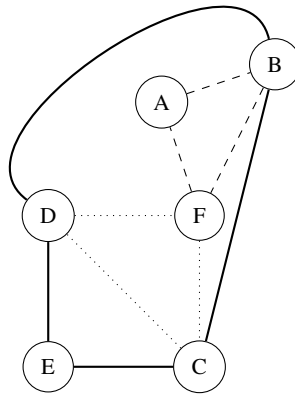
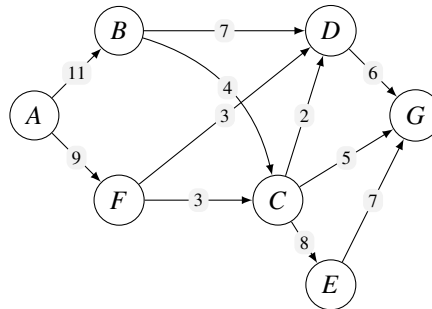


Abb. 3.32 Graphische Darstellung der vom Hierholzer-Algorithmus gefundenen Kreise.

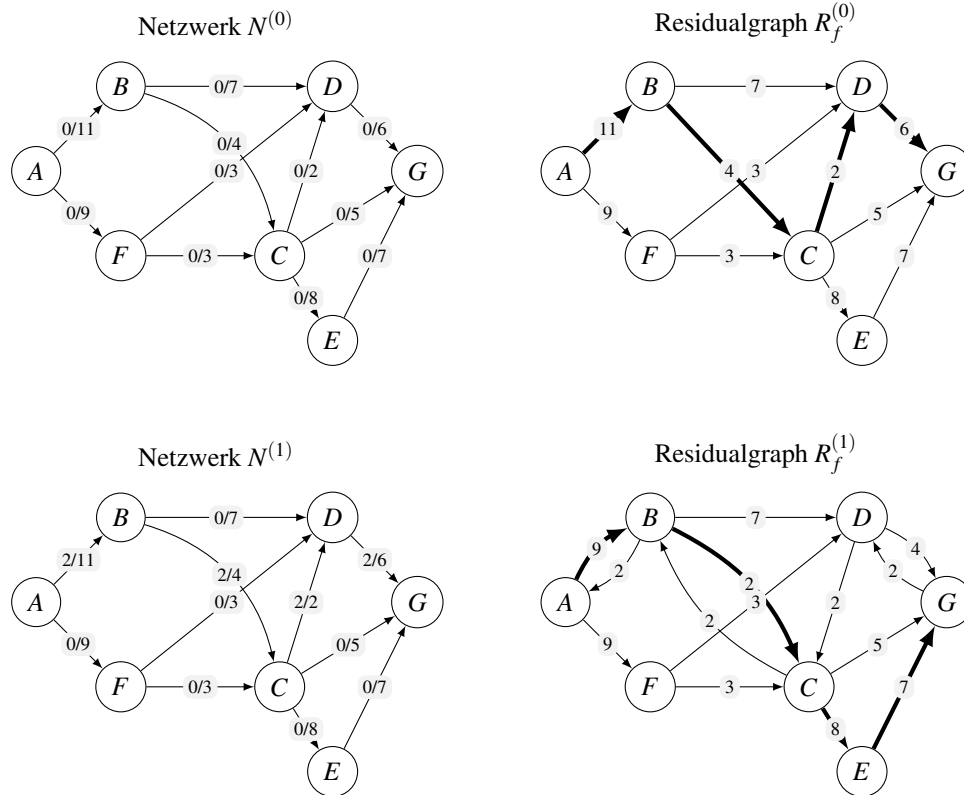
Wir fügen zunächst  $Z_2$  in  $Z_1$  ein und erhalten  $Z_4 = (D, E, C, B, D, C, F, D)$  und schließlich  $Z_3$  in  $Z_4$  was den Eulerkreis  $Z_5 = (D, E, C, B, A, F, B, D, C, F, D)$ .

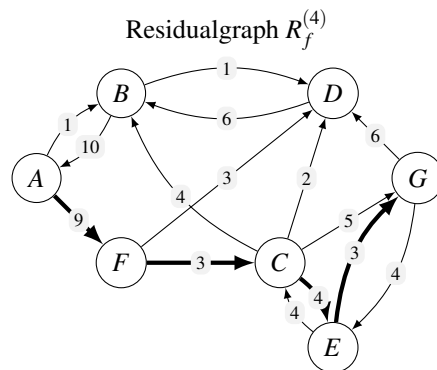
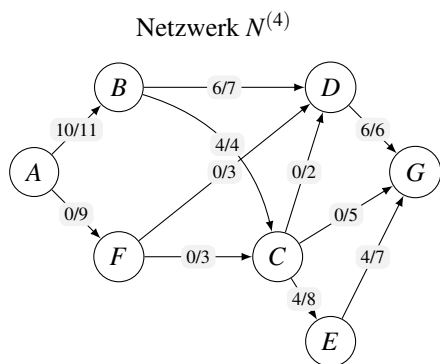
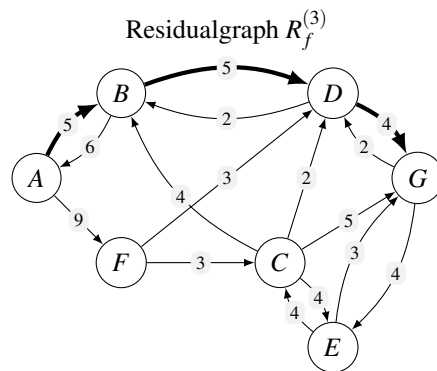
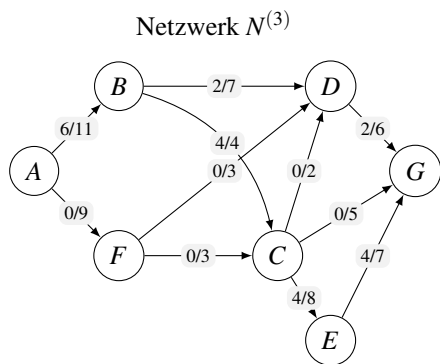
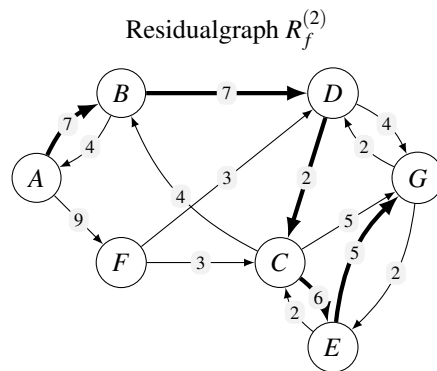
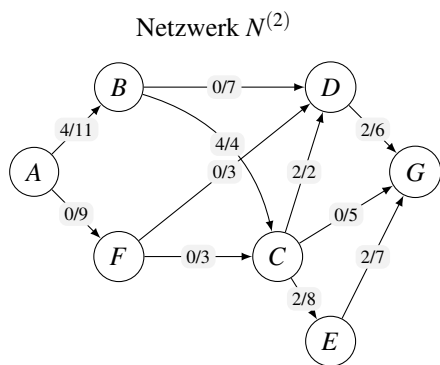
**3.14.** Gegeben sei das folgende Netzwerk  $N = (V, E, A, G, c)$  mit Kapazitätsfunktion  $c : E \rightarrow \mathbb{N}$ . Berechnen Sie mit dem Algorithmus von Ford & Fulkerson einen maximalen Fluss in  $N$ . Bei der Suche nach  $f$ -augmentierenden Pfaden arbeiten Sie dabei stets die Nachbarknoten in aufsteigender lexikographischer Reihenfolge ab (besuche etwa B vor C). Geben Sie in jeder Iteration das Netzwerk  $N$  sowie den Residualgraphen  $R_f$  an. Wie groß ist der maximale Fluss  $f^*$ ?



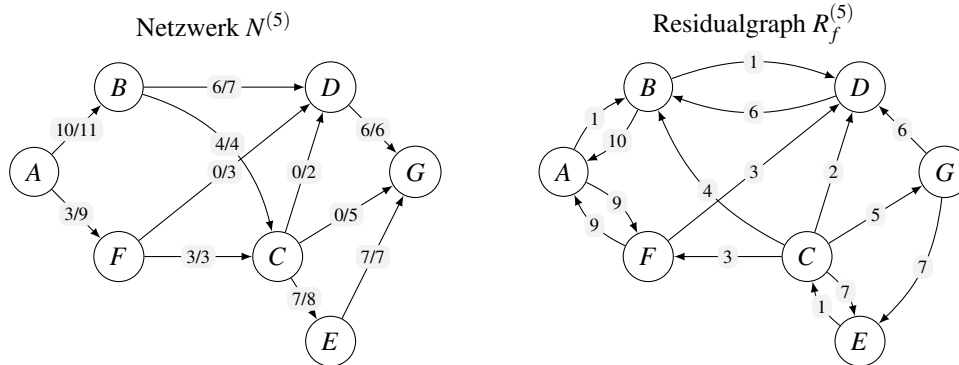
Wir bezeichnen im folgenden mit  $N^{(i)}$  das Netzwerk und mit  $R_f^{(i)}$  den Residualgraphen in der  $i$ -ten Iteration. Der Residualgraph zu Beginn ist gleich dem Netzwerk. Wir finden durch Folgen der Knoten von der Quelle  $A$  den  $f$ -augmentierenden Pfad  $A - B - C - D - G$ . Über diesen Pfad kann der Fluss (mit 0 initialisiert) um den Bottleneck  $(C, D)$  mit  $c(C, D) = 2$  erhöht werden. Die Kante  $(C, D)$  ist dadurch  $f$ -saturiert und taucht in  $R_f^{(1)}$  daher nun noch als Rückwärtskante auf. Die nachfolgenden Iterationen verlaufen nach dem gleichen Prinzip. Zu beachten ist hier die Reihenfolge der Abarbeitung der Knoten bei der Suche nach  $f$ -augmentierenden Wegen. So würde man etwa in  $R_f^{(4)}$  zunächst den Weg  $A - F - C - B - A$  finden, welcher allerdings einen Kreis schließt. Daher wird an dieser Stelle mittels Backtracking von  $B$  aus weiter gesucht. Auch die Alternative über den Knoten  $D$ , also  $A - F - C - B - D - B$  schließt einen Kreis. Auf diese Weise kommt schließlich der  $f$ -augmentierende Pfad  $A - F - C - E - G$  zustande.

Nach 5 Iterationen terminiert der Algorithmus, da die kein  $A - G$ -Weg mehr in  $R_f^{(5)}$  existiert. Der maximale Fluss  $f^*$  kann aus  $N^{(5)}$  abgelesen werden. Der Wert ist  $|f^*| = \sum_{e \in \delta_-(A)} f(e) = 13$ .







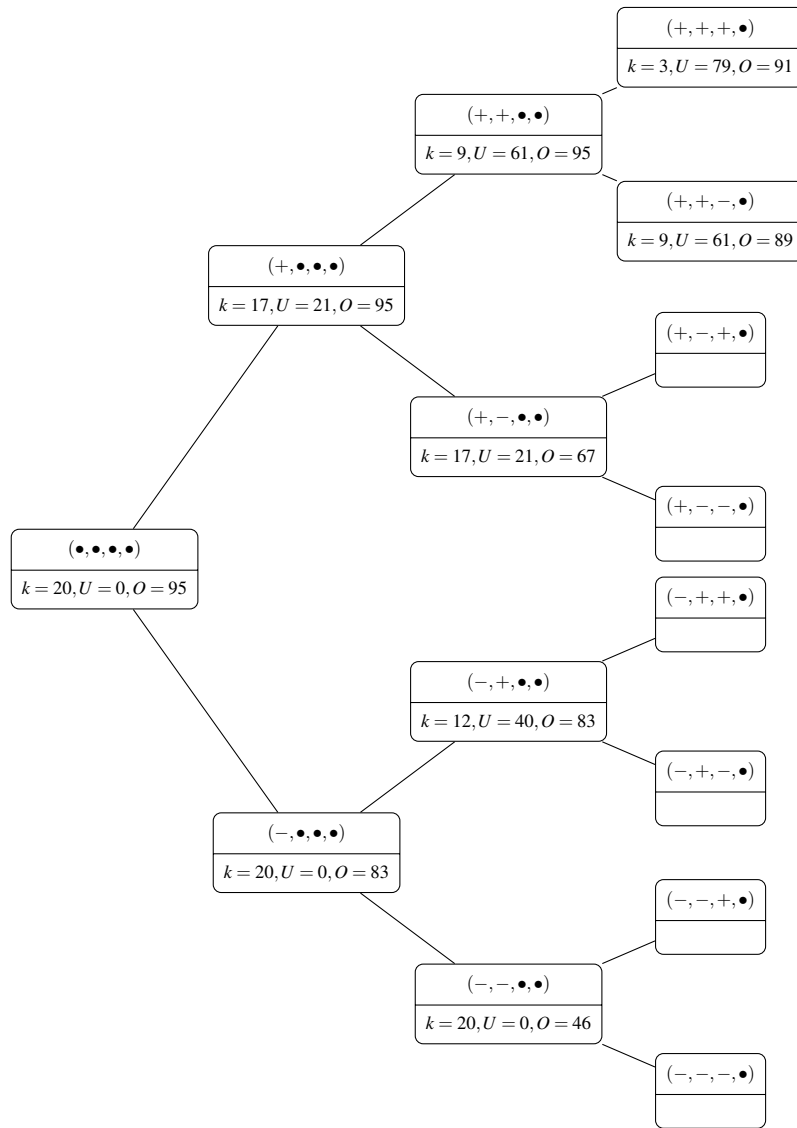


**3.15.** Eine Logistikfirma möchte ihren LKW möglichst gut beladen. Sie hat Güter auf Lager, die verschiedenes Gewicht und zugleich unterschiedlichen Nutzen aufweisen. Die folgende Tabelle gibt Ihnen einen Überblick über die Güter und deren Nutzen.

Gegenstand	Gewicht (t)	Nutzen (T€)
G1	3	21
G2	8	40
G3	6	18
G4	7	28

Leider kann der LKW nur ein Gewicht von 20 t transportieren. Die Herausforderung ist nun, die optimale Beladung für den LKW zu finden, um den transportierten Nutzen zu maximieren. Der Spediteur hat bereits angefangen, ein Branch & Bound-Verfahren durchzuführen (siehe Gafik), ist aber nicht fertig geworden ( $k$  = Restkapazität,  $U$  = untere Schranke,  $O$  = obere Schranke).

- Bestimmen Sie im gegebenen Baum, welche Zweige Sie nicht weiterverfolgen müssen. Streichen Sie die entsprechenden Einträge und begründen Sie kurz.
- Berechnen Sie für die übrigen Teillösungen die Restkapazität ( $k$ ), die untere Schranke ( $U$ ) und die obere Schranke ( $O$ ). Entscheiden Sie erneut, welche Teillösungen gestrichen werden können.
- Berechnen Sie die optimale Lösung, indem Sie den Branch & Bound-Ansatz bis zum Ende führen. Erklären Sie kurz (!) Ihr Vorgehen und geben Sie die optimale Lösung konkret an.



- Gestrichen werden können die Elemente  $(+, -, \bullet, \bullet)$  und  $(-, -, \bullet, \bullet)$  sowie alle darunter hängenden Lösungen, da die obere Schranke der Teillösungen durch die untere Schranke der Teillösung  $(+, +, +, \bullet)$  dominiert werden (siehe ausgegraute Knoten in Abbildung 3.33).
- Die berechneten Werte stehen in der Lösungsgrafik 3.33. Es kann nun noch die Teillösung  $(-, +, -, \bullet)$  entfernt werden.
- Es müssen nur noch die Lösungen  $(+, +, +, +)$  (ungültig, da Kapazität überschritten), die Lösung  $(+, +, +, -)$ , die Lösung  $(+, +, -, +)$  (Optimum) und die Lösung  $(+, +, -, -)$  untersucht werden.

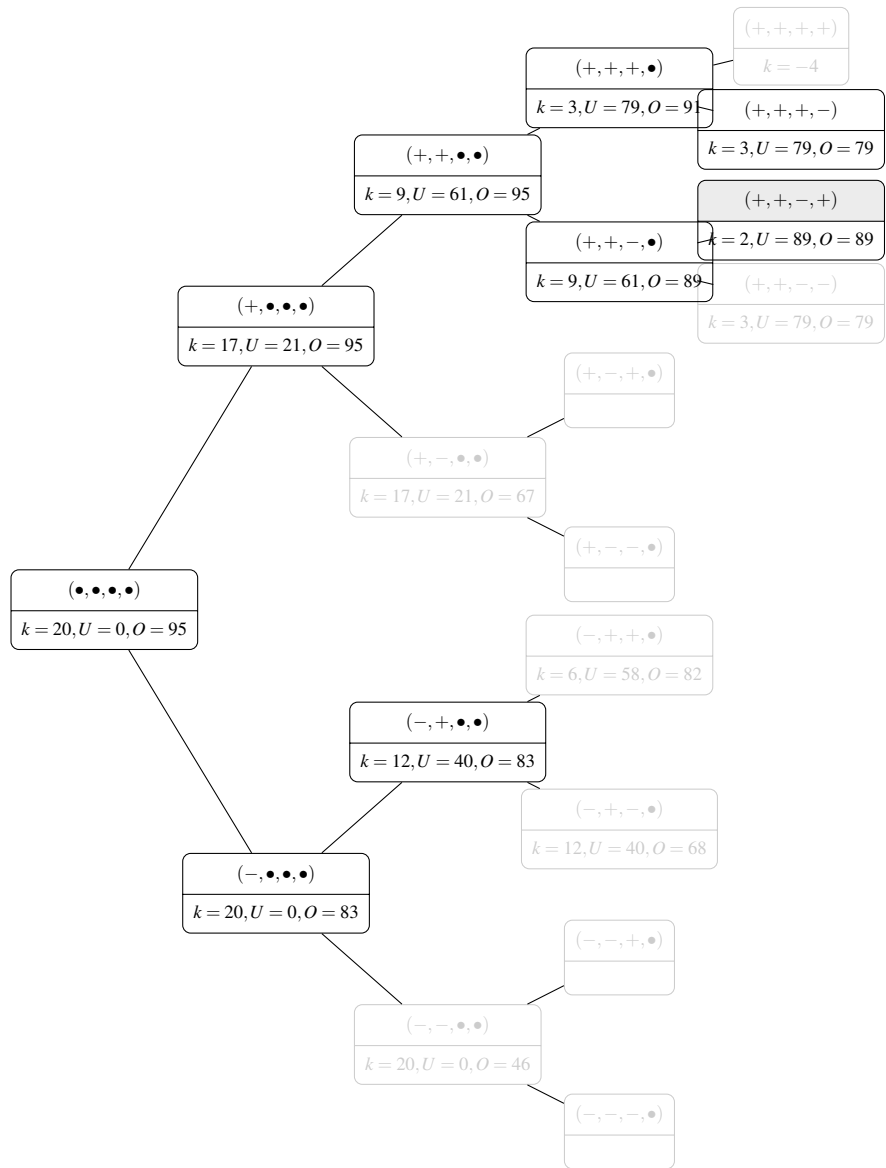


Abb. 3.33 Vervollständigter Branch & Bound Baum.

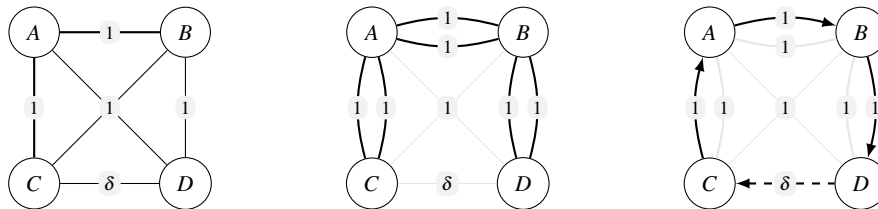
Für den optimalen Knoten  $(+, +, -, +)$  ergibt sich:

$$k = 2, \quad U = 89, \quad O = 89.$$

Damit dominiert diese Lösung alle übrigen noch offenen Lösungen. Sie ist damit auch die einzige optimale Lösung. Die in Teilaufgabe (b) berechnete und übriggebliebene Lösung kann damit ebenfalls gestrichen werden.

**3.16.** Zeigen Sie durch Angabe eines Beispielgraphen, dass die MST-Heuristik auf nicht-metrischen Graphen beliebig schlechte Lösungen produzieren kann.

**Beispiellösung:** Wir konstruieren ein Minimalbeispiel eines vollständigen ungerichteten Graphen  $G = (V, E)$  mit 4 Knoten  $V = \{A, B, C, D\}$  an, welcher die Dreiecksungleichung nicht erfüllt und damit die dritte Metrik-Eigenschaft verletzt. Wir setzen  $c(C, D) = \delta > 2$  beliebig und  $c(e) = 1$  für alle übrigen Kanten (siehe Abbildung 3.34 links). Die Dreiecksungleichung ist verletzt, da etwa  $\delta = c(C, D) > c(C, B) + c(B, D) = 2$ . Ein möglicher minimaler Spannbaum enthält nun die Kanten  $\{A, B\}$ ,  $\{A, C\}$  und  $\{B, D\}$ . Der eulersche Graph mit verdoppelten Kanten ist in Abbildung 3.34 (mitte) dargestellt. Eine Möglichkeit eine Tour mittels der MST-Heuristik ausgehend vom Startknoten  $A$  zu konstruieren besteht darin den Kanten der Eulertour zunächst nach  $B$  und anschließend nach  $D$  zu folgen. Von  $D$  aus können wir allerdings nur zurück nach  $B$ . Gemäß der MST-Heuristik wählen wir als nächsten Knoten  $C$  und schließen den Kreis durch die Kante  $\{C, A\}$ . Die produzierte Tour ist folglich  $T_{\text{MST}} = (A, B, D, C, A)$  (siehe Abbildung 3.34 rechts) mit Kosten  $c(T_{\text{MST}}) = 1 + 1 + \delta + 1$ . Die optimale Tour  $T_{\text{OPT}} = (A, D, B, C, A)$  hat jedoch nur die Gesamtkosten  $c(T_{\text{OPT}}) = 1 + 1 + 1 + 1 = 4$ . Wir können  $\delta$  beliebig groß wählen. Die konstruierte Tour wird damit beliebig schlecht.

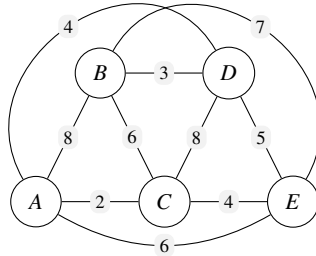


**Abb. 3.34** Beispielgraph (links), Beispielgraph mit minimalem Spannbaum und verdoppelten Kanten (mitte), konstruierte Tour  $T_{\text{MST}}$  (rechts).

**3.17.** In der Vorlesung wurde die MST-Heuristik für das (metrische) TSP vorgestellt. Diese erzeugt zunächst einen minimalen Spannbaum und im zweiten Schritt durch Verdopplung der Kanten einen Eulergraphen. Schließlich wird der Eulergraph traversiert und es werden Abkürzungen (nicht im Eulergraphen enthaltene Kanten) eingefügt, falls ein bereits besuchter Knoten erneut besucht würde.

- Zeigen Sie formal, dass der durch Verdopplung der Kanten entstehende Graph tatsächlich einen Eulerkreis enthält.
- Bestimmen Sie eine Lösung für den obigen Graphen.

**Beispiellösung:**



- a) Sei  $T$  der im ersten Schritt erzeugte minimale Spannbaum. Sei ferner  $G$  der durch Verdopplung der Baumkanten entstehende Graph. Dann gilt für alle Knoten  $v \in V$

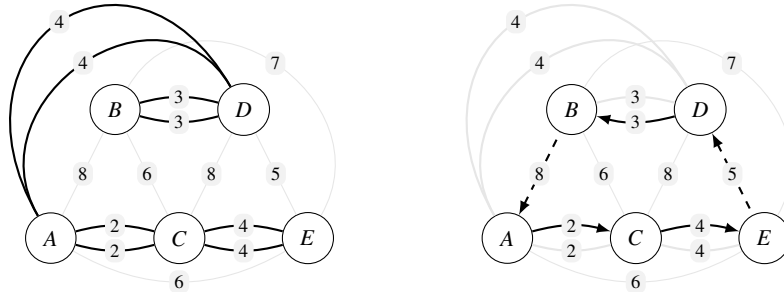
$$\delta_G(v) = \underbrace{2\delta_T(v)}_{\text{gerade}}.$$

Damit hat in  $G$  jeder Knoten geraden Grad und somit ist  $G$  eulersch.

- b) Wir konstruieren zunächst (z. B. mit dem Algorithmus von Kruskal) einen minimalen Spannbaum<sup>7</sup> und duplizieren die Kanten. Dadurch erhalten wir einen eulerschen Graphen (siehe Abbildung). Wir durchlaufen nun den Eulergraphen und wählen zunächst die Kanten  $\{A, C\}$  und  $\{C, E\}$ . Folgen wir weiterhin den Kanten des Eulergraphen, so führt nur ein Weg zurück zu bereits besuchten Knoten. Die MST-Heuristik wählt in diesem Fall die kürzeste Kante zu einem noch nicht besuchten Knoten, nämlich  $\{E, D\}$  und läuft weiterhin anhand der Kanten des Eulergraphen über  $\{D, B\}$ . Schließlich muss die Kante  $\{B, A\}$  gewählt werden um die Rundreise abzuschließen. Die resultierende Tour  $[A, C, E, D, B, A]$  hat Kosten 22 und ist optimal<sup>8</sup>.

<sup>7</sup> Der Spannbaum ist hier eindeutig.

<sup>8</sup> In Abhängigkeit von der Reihenfolge in der man die Kanten des Eulergraphen abläuft kann auch eine andere, suboptimale Tour entstehen; etwa  $[A, D, B, C, E, A]$  mit Kosten 23



**Abb. 3.35** Verdoppelte Spannbaum-Kanten (links) und mittels MST-Heuristik konstruierte TSP-Tour (rechts).

## Literaturverzeichnis

1. Dorigo, M., Maniezzo, V., Coloni, A.: Ant System: Optimization by a Colony of Cooperating Ants. *IEEE Transactions on Systems Man and Cybernetics - Part B: Cybernetics* **26**(1), 29–41 (1996)
2. Euler, L.: *Solutio problematis ad geometriam situs pertinentis*. *Commentarii academiae scientiarum Petropolitanae* **8**, 128–140 (1741)
3. Laux, H., Gillenkirch, R.M., Schenk-Mathes, H.Y.: *Entscheidungstheorie*, 9 edn. Springer, Heidelberg (2014)
4. Nachtigall, W.: *Bionik, Grundlagen und Beispiele für Ingenieure und Naturwissenschaftler*, 2 edn. Springer (2002)
5. Schwefel, H.P.: *Evolution and Optimum Seeking*. Wiley, New York (1995)
6. Taha, H.A.: *Operations Research: An Introduction*, 9 edn. Pearson, New Jersey (2011)
7. Winter, S.: *Grundzüge der Spieltheorie - Ein Lehr- und Arbeitsbuch für das (Selbst-)Studium*. Springer Gabler, Heidelberg (2015)

## Lösungen zu Kapitel 4

**4.1.** Bestimmen Sie für die folgenden Optimierungsprobleme (OP) mit Nebenbedingungen, ob es sich um ein lineares Optimierungsproblem (LOP) handelt oder nicht. Begründen Sie Ihre Antwort. Sollte es sich um ein LOP handeln, so stellen Sie das Problem grafisch dar und überführen Sie es in *Standardform*. Beantworten Sie zudem folgenden Frage: Ist das Problem lösbar und wenn ja, ist es eindeutig lösbar?

<p><b>OP1</b> <math>\frac{1}{3}x_1 + x_2 \rightarrow \max</math>  u. d. N. <math>x_1, x_2 \geq 0</math>  <math>x_2 \leq -\frac{2}{3}x_1 + 3</math></p>	<p><b>OP2</b> <math>2x_1x_2 + 3x_2 \rightarrow \min</math>  u. d. N. <math>3 \geq x_2 \geq 0</math>  <math>3 \geq x_1 \geq 0</math></p>	<p><b>OP3</b> <math>-2^3x_1 + 2x_2 \rightarrow \max</math>  u. d. N. <math>x_1, x_2 \geq 0</math>  <math>x_2 \geq -x_1 + 2</math>  <math>x_2 \leq x_1 + 2</math>  <math>x_2 \geq x_1 - 2</math>  <math>-4x_1 - x_2 \geq -12</math></p>
--	---	--

**Beispiellösung:** Die Probleme **OP1** sowie **OP3** sind lineare Optimierungsprobleme, da die Zielfunktion von der Form  $f(x) = c^T x$  und alle Restriktionen/Nebenbedingungen durch lineare Ungleichungen gegeben sind. **OP2** hingegen ist nicht linear, da der Faktor  $x_1x_2$  darin vorkommt.

Die Überführung in Standardform erfordert das “flippen” der Zielfunktion (Maximierung von  $f$  entspricht der Minimierung von  $-f$ ), dem Einführen von Schlupfvariablen sowie Sicherstellung, dass die Komponenten des inhomogenen Anteils (d.h. der rechten Seite der Gleichungen) des entstehenden linearen Gleichungssystems alle positiv sind. Nachfolgend sind die Standardformen von **OP1** und **OP3** dargestellt. Die grafische Darstellung findet sich in Abb. 4.6.

<p><b>OP1</b> <math>-\frac{1}{3}x_1 - x_2 \rightarrow \min</math>  u. d. N. <math>\frac{2}{3}x_1 + x_2 + x_3 = 3</math>  <math>x_1, x_2, x_3 \geq 0</math></p>	<p><b>OP3</b> <math>8x_1 - 2x_2 \rightarrow \min</math>  u. d. N. <math>x_1 + x_2 - x_3 = 2</math>  <math>-x_1 + x_2 + x_4 = 2</math>  <math>x_1 - x_2 + x_5 = 22</math>  <math>4x_1 + x_2 + x_6 = 12</math>  <math>x_1, x_2, \dots, x_6 \geq 0</math></p>
--	--

Wir erkennen, dass beide Probleme lösbar sind, da der zulässige Bereich nicht leer ist und das Simplex beschränkt ist. **OP1** hat eine eindeutige optimale Lösung im Punkt  $(0, 3)$  mit Zielfunktionswert  $\frac{1}{3}x_1 + x_2 = 0 + 3 = 3$ . Bei **OP3** hingegen können wir die Iso-Gewinngerade so weit nach rechts verschieben, bis sie die Nebenbedingungen  $-4x_1 - x_2 \geq -12$  (NB) überdeckt; die Iso-Gewinngerade ist dem-

nach parallel zu NB. In diesem Fall sind alle Lösungen auf der Strecke, zwischen den beiden Eckpunkten, durch die die Gerade zu NB verläuft, optimal.

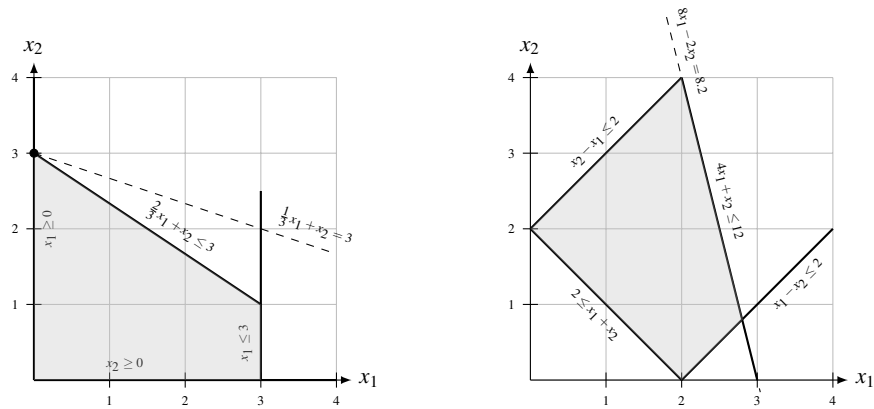


Abb. 4.6 OP1 (links) und OP3 (rechts).

**4.2.** Christian ist im Fitnessstudio und kann an zwei Geräten A und B trainieren. Die Übungen an Gerät A verbrennen 10 kCal pro Minute, jene an Gerät B ganze 12 kCal pro Minute. Da Christian als Student der Wirtschaftsinformatik sehr beschäftigt ist, möchte er die Zeit im Fitnessstudio effizient hinsichtlich Kalorienverbrauch nutzen und in seiner höchstens 30-minütigen Trainingszeit besonders viele Kalorien verbrennen. Allerdings kann er Gerät B höchstens 10 Minuten in Anspruch nehmen (das Gerät ist sehr begehrt).

- Formulieren Sie Christians Problem als *lineares Programm* und lösen Sie es graphisch. Skizzieren Sie hierzu den zulässigen Bereich. Gibt es eine eindeutige Lösung?
- Bringen Sie das lineare Programm in *Standardform*.
- Wir nehmen nun an, dass zusätzlich die Bedingungen  $x_1, x_2 \in \mathbb{Z}$  gelten. Ist die ermittelte Lösung weiterhin optimal für das nun resultierende Problem?

**Beispiellösung:** Bezeichne  $x_1$  die Anzahl Minuten an Gerät A und  $x_2$  entsprechend die Trainingszeit an Gerät B.

- Dann gilt es die lineare Zielfunktion  $z = 10x_1 + 12x_2$  zu maximieren unter den Nebenbedingungen  $x_1, x_2 \geq 0$  (die Trainingszeit an den Geräten ist stets positiv),  $x_2 \leq 10$  (maximal 10 Minuten an Gerät B),  $x_1 + x_2 \leq 30$  (maximal 30 Minuten Training). Der zulässige Bereich ist grün in Abbildung 4.7 dargestellt. Die blaue Linie entspricht der maximalen Isogewinn-Geraden  $10x_1 + 12x_2 = 320$  für  $x_1 = 20$  Minuten Training an Gerät A und  $x_2 = 10$  Minuten Training an Gerät B.
- Um die Standardform herzustellen, müssen 4 Schlupfvariablen eingefügt werden:



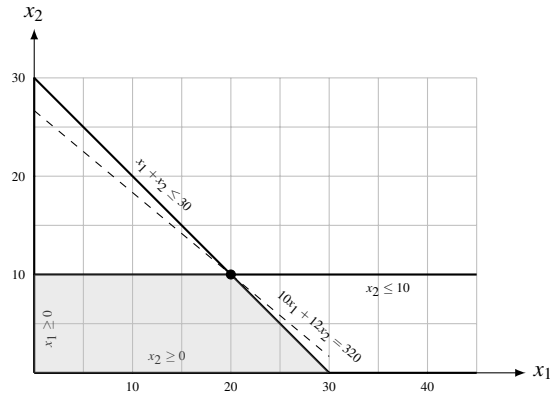


Abb. 4.7 Grafische Lösung von Christians Fitness Optimierungsproblem.

$$\begin{array}{rclcl}
 x_1 & - & s_1 & & = & 0 \\
 x_2 & & - & s_2 & & = & 0 \\
 x_1 + x_2 & & & + & s_3 & = & 30 \\
 x_2 & & & & + & s_4 & = & 10 \\
 x_1, x_2, s_1, s_2, s_3, s_4 & \geq & & & & & & 0
 \end{array}$$

- c) Die zusätzlichen Bedingungen  $x_1, x_2 \in \mathbb{Z}$  machen aus dem LP ein ILP, also ein *ganzzahliges lineares Programm*. In diesem speziellen Fall sind alle Ecken des Simplex, insbesondere also die optimale Lösung ganzzahlig. Damit ist die grafisch ermittelte Lösung also auch weiterhin optimal. Im Allgemeinen sind die Ecken des Simplex jedoch nicht ganzzahlig, womit auch die durch die Simplex-Methode von Dantzig ermittelten Lösungen keine Lösungen des ILP sind.

**Für mathematisch Interessierte:** in diesem Fall ist die Koeffizientenmatrix  $A$  des linearen Programms *total unimodular*, d.h., es gilt  $\det(A) \in \{-1, +1\}$ . Ferner ist  $b \in \mathbb{Z}^n$ . Man kann zeigen, dass daraus die Ganzzahligkeit aller Ecken des Simplex folgt.

#### 4.3. Wir betrachten das folgende lineare Optimierungsproblem **LOP**

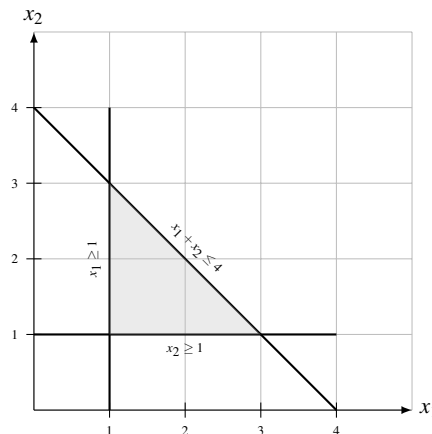
$$\begin{array}{l}
 \mathbf{LOP} \quad \frac{1}{2}x_1 + x_2 \rightarrow \max \\
 \text{u. d. N.} \quad x_1 + x_2 \leq 4 \\
 \quad \quad \quad x_1, x_2 \geq 1
 \end{array}$$

- Stellen Sie das LOP grafisch dar. Was können Sie über den Lösungsraum aussagen?
- Überführen Sie das Problem in *Standardform*.
- Erstellen Sie die erweiterte Koeffizientenmatrix des aus der Standardform entstehenden linearen Gleichungssystems und bringen Sie es in Basisform (die ersten drei Spalten sollen in der Basis liegen). Lesen Sie die entstehende *spezielle*

- Lösung* ab und berechnen Sie den Zielfunktionswert. Betrachten Sie Ihre Grafik aus Aufgabenteil a). Wo befindet sich die gefundene Lösung? Ist sie optimal?
- d) Stellen Sie die Basisvariablen in Abhängigkeit der Nicht-Basisvariablen dar und setzen Sie Ihr Ergebnis in die Zielfunktion ein, um diese in Abhängigkeit der Nicht-Basisvariablen darzustellen. Welche Schlüsse können Sie aus der allgemeinen Darstellung ziehen?
- e) Bestimmen Sie nun, welche Variable die Basis verlässt und welche hinzugenommen wird. Bestimmen Sie hierzu die Engpässe.
- f) Führen Sie den Basiswechsel durch und lesen Sie die neue Lösung ab. Ist diese optimal?

**Beispiellösung:** Ziel dieser Aufgabe ist es sich nochmal mit der “Herleitung” des Simplex Algorithmus von Danzig zu beschäftigen um ein besseres Verständnis der Arbeitsweise zu erlangen und die Schritte, die man im Rahmen des Simplex Algorithmus durchführt, besser nachvollziehen zu können.

- a) Das LOP ist in Abbildung 4.8 dargestellt. Wir erkennen, dass der zulässige Bereich nicht leer ist, da sich keine Nebenbedingungen widersprechen. Weiterhin ist das LOP beschränkt, so dass die Existenz einer eindeutigen Lösung sichergestellt ist.



**Abb. 4.8** Grafische Darstellung des LOP.

- b) Die Umwandlung in Standardform erfordert die Umwandlung des Maximierungsproblems in ein Minimierungsproblem und das Einfügen von Schlupfvariablen um das lineare Ungleichungssystem in ein äquivalentes lineares Gleichungssystem (LGS) umzuwandeln.

$$\begin{aligned}
\mathbf{LOP} \quad & -\frac{1}{2}x_1 - x_2 \rightarrow \min \\
\text{u. d. N.} \quad & x_1 - x_3 = 1 \\
& x_2 - x_4 = 1 \\
& x_1 + x_2 + x_5 = 4 \\
& x_1, x_2, x_3, x_4, x_5 \geq 0
\end{aligned}$$

- c) Wir können jedes LGS in Matrixschreibweise  $Ax = b$  darstellen. Für die Variablenelimination nach Gauss ist es zweckmäßig die erweiterte Koeffizientenmatrix  $(A|b)$  aufzustellen. Nachfolgend tun wir genau das und formen mit elementaren Zeilenumformungen das LGS derart um, dass die drei Spalten in Zeilenstufenform sind (also Einsen nur auf der Diagonale und sonst ausschließlich Nullen).

$$\begin{array}{ccc}
\begin{pmatrix} 1 & 0 & -1 & 0 & 0 & 1 \\ 0 & 1 & 0 & -1 & 0 & 1 \\ 1 & 1 & 0 & 0 & 1 & 4 \end{pmatrix} & \begin{array}{l} \left[ \begin{array}{l} - \\ - \\ - \end{array} \right]^{-1} \\ \leftarrow + \end{array} & \rightsquigarrow \begin{pmatrix} 1 & 0 & -1 & 0 & 0 & 1 \\ 0 & 1 & 0 & -1 & 0 & 1 \\ 0 & 1 & 1 & 0 & 1 & 3 \end{pmatrix} \begin{array}{l} \left[ \begin{array}{l} - \\ - \\ - \end{array} \right]^{-1} \\ \leftarrow + \end{array} \\
\rightsquigarrow \begin{pmatrix} 1 & 0 & -1 & 0 & 0 & 1 \\ 0 & 1 & 0 & -1 & 0 & 1 \\ 0 & 0 & 1 & 1 & 1 & 2 \end{pmatrix} \begin{array}{l} \leftarrow + \\ \left[ \begin{array}{l} - \\ - \\ - \end{array} \right] \end{array} & & \rightsquigarrow \begin{pmatrix} 1 & 0 & 0 & 1 & 1 & 3 \\ 0 & 1 & 0 & -1 & 0 & 1 \\ 0 & 0 & 1 & 1 & 1 & 2 \end{pmatrix}
\end{array}$$

Daraus können wir die *spezielle Lösung*  $x_1 = 3, x_2 = 1, x_3 = 1, x_4 = x_5 = 0$  direkt ablesen. Als Funktionswert erhalten wir hier  $-\frac{1}{2}x_1 - x_2 = -\frac{1}{2} \cdot 3 - 1 = -\frac{5}{2}$ . Da die Schlupfvariablen – also insbesondere  $x_3$  – in der Zielfunktion nicht vorkommen, entspricht die Lösung der Ecke  $(x_1 = 3, x_2 = 1)$  des Simplex (siehe Abb. 4.8). Die Lösung ist nicht optimal!

- d) Nun stellen wir die Basisvariablen als Funktion der Nicht-Basisvariablen dar um einen Überblick über das Verbesserungspotential zu bekommen.

$$\begin{aligned}
x_1 &= 4 - x_2 - x_5 = 3 - x_4 - x_5 \\
x_2 &= 1 + x_4 \\
x_3 &= x_1 - 1 = 2 - x_4 - x_5
\end{aligned}$$

Einsetzen in die Zielfunktion ergibt

$$-\frac{1}{2}x_1 - x_2 = -\frac{1}{2}(3 - x_4 - x_5) - (1 + x_4) = -\frac{5}{2} - \frac{1}{2}x_4 + \frac{1}{2}x_5.$$

Da es die Zielfunktion zu minimieren gilt erreichen wir also eine Verbesserung im Sinne der Minimierung indem wir  $x_4$  verkleinern (eine Vergrößerung von  $x_5$  hingegen würde den Zielfunktionswert vergrößern). Formaler haben wir nun die  $\delta$ -Werte  $\delta_4 = \frac{1}{2}$  und  $\delta_5 = -\frac{1}{2}$  (bzw.  $-\delta_4 = -\frac{1}{2}$  und  $-\delta_5 = \frac{1}{2}$ ). An dieser Stelle wählt der Simplex-Algorithmus die Spalte des Simplex-Tableau mit maximalem nicht-negativem  $\delta$ -Wert.

e) Wir können  $x_4$  natürlich nicht beliebig kleiner machen, denn irgendwann wird einer der übrigen Nebenbedingungen des LOP verletzt und die ermittelte Lösung wäre unzulässig. Aus diesem Grund bestimmt man die sogenannten *Engpässe*. Das sind Werte, die in gewissem Sinne aussagen, welche Nebenbedingungen oder Nichtnegativitätsbedingungen zuerst verletzt würden. Da in der Standardform alle Variablen  $\geq 0$  sein müssen, nutzen wir nun die Darstellung aus c) und prüfen, unter welchen Bedingungen die Variablen  $\geq 0$  bleiben.

$$\begin{aligned}x_1 = 3 - x_4 \geq 0 &\Leftrightarrow x_4 \leq 3 \\x_2 = 1 + x_4 \geq 0 &\Leftrightarrow x_4 \geq -1 \text{ (kein Engpass)} \\x_3 = 2 - x_4 \geq 0 &\Leftrightarrow x_4 \leq 2\end{aligned}$$

D. h. sobald  $x_4 > 2$  ist, ist die Nicht-Negativitätsbedingung an  $x_3$  verletzt.  
f) Nun führen wir einen Basiswechsel durch:  $x_3$  verlässt die Basis,  $x_4$  wird in die Basis aufgenommen.

$$\left( \begin{array}{cccc} 1 & 0 & 0 & 1 & 1 & 3 \\ 0 & 1 & 0 & -1 & 0 & 1 \\ 0 & 0 & 1 & 1 & 1 & 2 \end{array} \right) \begin{array}{l} \leftarrow + \\ \leftarrow + \\ \leftarrow - \end{array} \rightsquigarrow \left( \begin{array}{cccc} 1 & 0 & -1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 & 1 & 3 \\ 0 & 0 & 1 & 1 & 1 & 2 \end{array} \right)$$

Wir können nun die Lösung  $x_1 = 1, x_2 = 3, x_3 = x_5 = 0, x_4 = 2$  ablesen. Der Zielfunktionswert entspricht  $-\frac{7}{2}$  und entspricht der optimalen Lösung.

4.4. Gegeben ist das folgende lineare Optimierungsproblem

$$4x_1 - 3x_2 - 2x_3 \stackrel{!}{=} \min$$

mit einigen Nebenbedingungen

$$\begin{aligned}3x_1 &+ 3x_3 \leq 10 \\x_2 - x_3 &\leq 15 \\3x_1 + 2x_2 &= 21 \\x_1, x_2, x_3 &\geq 0\end{aligned}$$

Lösen Sie das Problem mit der **Zweiphasenmethode** und beschreiben Sie knapp die von Ihnen durchgeführten Schritte und Umformungen.

**Lösungsvorschlag:** Zu lösen ist das folgende lineare Optimierungsproblem:

Zielfunktion:

$$4x_1 - 3x_2 - 2x_3 \stackrel{!}{=} \min$$

Nebenbedingungen:

$$\begin{aligned}3x_1 &+ 3x_3 \leq 10 \\1x_2 - 1x_3 &\leq 15 \\3x_1 + 2x_2 &= 21 \\x_1, x_2, x_3 &\geq 0\end{aligned}$$

Um die Standardform herzustellen, müssen 2 Schlupfvariablen eingefügt werden:

$$\begin{aligned}
3x_1 &+ 3x_3 + s_1 &= 10 \\
x_2 - x_3 &+ s_2 &= 15 \\
3x_1 + 2x_2 &&= 21 \\
x_1, x_2, x_3, s_1, s_2 &\geq 0
\end{aligned}$$

Beginn der ersten Phase:

Es sind noch nicht genügend Pivotspalten vorhanden, sodass für die erste Phase 1 künstliche Variable benötigt wird. Zudem ist die Zielfunktion folgendermaßen anzupassen:

$$\begin{array}{l}
1k_1 \stackrel{!}{=} \min \\
\begin{array}{c|cccccc|c|c}
& 0 & 0 & 0 & 0 & 0 & 1 & & \\
\hline
& x_1 & x_2 & x_3 & s_1 & s_2 & k_1 & & \\
\hline
0 \ s_1 & \boxed{3} & 0 & 3 & 1 & 0 & 0 & 10 & \frac{10}{3} \\
0 \ s_2 & 0 & 1 & -1 & 0 & 1 & 0 & 15 & - \\
1 \ k_1 & 3 & 2 & 0 & 0 & 0 & 1 & 21 & 7 \\
\hline
& 3 & 2 & 0 & 0 & 0 & 0 & 21 & 
\end{array}
\end{array}$$

Spalte  $s_1$  verlässt die Basis, Spalte  $x_1$  geht in die Basis.

Umformungen:

$$I \rightarrow \frac{1}{3}I$$

$$II \rightarrow II - 0I$$

$$III \rightarrow III - 1I$$

$$\begin{array}{c|cccccc|c|c}
& 0 & 0 & 0 & 0 & 0 & 1 & & \\
\hline
& x_1 & x_2 & x_3 & s_1 & s_2 & k_1 & & \\
\hline
0 \ x_1 & 1 & 0 & 1 & \frac{1}{3} & 0 & 0 & \frac{10}{3} & - \\
0 \ s_2 & 0 & 1 & -1 & 0 & 1 & 0 & 15 & 15 \\
1 \ k_1 & 0 & \boxed{2} & -3 & -1 & 0 & 1 & 11 & \frac{11}{2} \\
\hline
& 0 & 2 & -3 & -1 & 0 & 0 & 11 & 
\end{array}$$

Spalte  $k_1$  verlässt die Basis, Spalte  $x_2$  geht in die Basis.

Umformungen:

$$I \rightarrow I - 0III$$

$$II \rightarrow II - \frac{1}{2}III$$

$$III \rightarrow \frac{1}{2}III$$

$$\begin{array}{c|cccccc|c|c}
& 0 & 0 & 0 & 0 & 0 & 1 & & \\
\hline
& x_1 & x_2 & x_3 & s_1 & s_2 & k_1 & & \\
\hline
0 \ x_1 & 1 & 0 & 1 & \frac{1}{3} & 0 & 0 & \frac{10}{3} & - \\
0 \ s_2 & 0 & 0 & \frac{1}{2} & \frac{1}{2} & 1 & -\frac{1}{2} & \frac{19}{2} & - \\
0 \ x_2 & 0 & 1 & -\frac{3}{2} & -\frac{1}{2} & 0 & \frac{1}{2} & \frac{11}{2} & - \\
\hline
& 0 & 0 & 0 & 0 & 0 & -1 & 0 & 
\end{array}$$

Die erste Phase ist nun beendet. Alle künstlichen Variablen haben die Basis verlassen und einen Deltawert von -1. Die restlichen Deltawerte, sowie der Zielwert sind 0. Im nächsten Schritt wird das Problem in der zweiten Phase optimiert, damit man eine zulässige Lösung für das Ausgangsproblem erhält.

Beginn der zweiten Phase:

Die künstlichen Variablen werden eliminiert und die Deltawerte werden neu berechnet. Zusätzlich wird die ursprüngliche Zielfunktion ( $4x_1 - 3x_2 - 2x_3 \stackrel{!}{=} \min$ ) gesetzt.

		4	-3	-2	0	0		
		$x_1$	$x_2$	$x_3$	$s_1$	$s_2$		
4	$x_1$	1	0	1	$\frac{1}{3}$	0	$\frac{10}{3}$	$\frac{10}{3}$
0	$s_2$	0	0	$\frac{1}{2}$	$\frac{1}{2}$	1	$\frac{19}{2}$	19
-3	$x_2$	0	1	$-\frac{3}{2}$	$-\frac{1}{2}$	0	$\frac{11}{2}$	-
		0	0	$\frac{21}{2}$	$\frac{17}{6}$	0	$-\frac{19}{6}$	

Spalte  $x_1$  verlässt die Basis, Spalte  $x_3$  geht in die Basis.

Umformungen:

$$I \rightarrow 1I$$

$$II \rightarrow II - \frac{1}{2}I$$

$$III \rightarrow III + \frac{3}{2}I$$

		4	-3	-2	0	0		
		$x_1$	$x_2$	$x_3$	$s_1$	$s_2$		
-2	$x_3$	1	0	1	$\frac{1}{3}$	0	$\frac{10}{3}$	-
0	$s_2$	$-\frac{1}{2}$	0	0	$\frac{1}{3}$	1	$\frac{47}{6}$	-
-3	$x_2$	$\frac{3}{2}$	1	0	0	0	$\frac{21}{2}$	-
		$-\frac{21}{2}$	0	0	$-\frac{2}{3}$	0	$-\frac{229}{6}$	

Die zweite Phase ist nun beendet, da alle Deltawerte negativ sind. Der Zielwert des Optimierungsproblems lautet:  $-\frac{229}{6}$ .

Im Optimum haben die Variablen die folgenden Werte:

$$x_1 = 0$$

$$x_2 = \frac{21}{2}$$

$$x_3 = \frac{10}{3}$$

4.5. Gegeben ist das folgende lineare Optimierungsproblem:

$$4x_1 + 2x_2 + 5x_3 + x_4 + 3x_5 + 5x_6 \stackrel{!}{=} \min$$

$$x_1 + 4x_2 + 4x_3 + x_4 + 2x_6 \geq 600$$

$$x_1 + x_3 + 4x_4 + 2x_5 + x_6 \geq 400$$

$$x_1, x_2, x_3, x_4, x_5, x_6 \geq 0$$

Lösen Sie das Problem mit dem *dualen* Simplex-Algorithmus und geben Sie die Lösung vollständig an.

**Beispiellösung:**

Um die Standardform herzustellen, müssen 2 Schlupfvariablen eingefügt werden:

$$-x_1 - 4x_2 - 4x_3 - x_4 - 2x_6 + s_1 = -600$$

$$-x_1 - x_3 - 4x_4 - 2x_5 - x_6 + s_2 = -400$$

$$x_1, x_2, x_3, x_4, x_5, x_6, s_1, s_2 \geq 0$$

	4	2	5	1	3	5	0	0	
	$x_1$	$x_2$	$x_3$	$x_4$	$x_5$	$x_6$	$s_1$	$s_2$	
0 $s_1$	-1	-4	-4	-1	0	-2	1	0	-600
0 $s_2$	-1	0	-1	-4	-2	-1	0	1	-400
	-4	-2	-5	-1	-3	-5	0	0	0
	4	$\frac{1}{2}$	$\frac{5}{4}$	1	-	$\frac{5}{2}$	-	-	-

Spalte  $s_1$  verlässt die Basis, Spalte  $x_2$  geht in die Basis.

Umformungen:

$$I \rightarrow -\frac{1}{4}I$$

$$II \rightarrow II - 0I$$

	4	2	5	1	3	5	0	0	
	$x_1$	$x_2$	$x_3$	$x_4$	$x_5$	$x_6$	$s_1$	$s_2$	
2 $x_2$	$\frac{1}{4}$	1	1	$\frac{1}{4}$	0	$\frac{1}{2}$	$-\frac{1}{4}$	0	150
0 $s_2$	-1	0	-1	-4	-2	-1	0	1	-400
	$-\frac{7}{2}$	0	-3	$-\frac{1}{2}$	-3	-4	$-\frac{1}{2}$	0	300
	$\frac{7}{2}$	-	3	$\frac{1}{8}$	$\frac{3}{2}$	4	-	-	-

Spalte  $s_2$  verlässt die Basis, Spalte  $x_4$  geht in die Basis.

Umformungen:

$$I \rightarrow I + \frac{1}{16}II$$

$$II \rightarrow -\frac{1}{4}II$$

	4	2	5	1	3	5	0	0	
	$x_1$	$x_2$	$x_3$	$x_4$	$x_5$	$x_6$	$s_1$	$s_2$	
2 $x_2$	$\frac{3}{16}$	1	$\frac{15}{16}$	0	$-\frac{1}{8}$	$\frac{7}{16}$	$-\frac{1}{4}$	$\frac{1}{16}$	125
1 $x_4$	$\frac{1}{4}$	0	$\frac{1}{4}$	1	$\frac{1}{2}$	$\frac{1}{4}$	0	$-\frac{1}{4}$	100
	$-\frac{27}{8}$	0	$-\frac{23}{8}$	0	$-\frac{11}{4}$	$-\frac{31}{8}$	$-\frac{1}{2}$	$-\frac{1}{8}$	350
	-	-	-	-	-	-	-	-	-

Das Tableau ist nun primal zulässig und somit konnte das Problem mit der dualen Simplex-Methode erfolgreich gelöst werden. Der optimale Zielwert lautet: 350.

Im Optimum haben die Variablen die folgenden Werte:

$$x_1 = 0$$

$$x_2 = 125$$

$$x_3 = 0$$

$$x_4 = 100$$

$$x_5 = 0$$

$$x_6 = 0$$

**4.6.** Zeigen Sie: Für alle primal zulässigen Punkte  $x$  mit  $Ax = b, x \geq 0$  und alle zulässigen Punkte  $y$  mit  $A^T y \geq c$  gilt

$$b^T y \geq c^T x.$$

Welche Bedeutung hat diese Aussage?

**Beispiellösung:** Der Beweis ist ziemlich einfach. Wir nutzen einfach alles aus, was in der Aufgabenstellung steht.

$$b^T y = (Ax)^T y = (x^T A^T) y = x^T \underbrace{(A^T y)}_{\geq c} \geq x^T c = c^T x.$$

Bezeichne  $g(y) = b^T y$  die Lösung des dualen Problems und  $f(x) = c^T x$  die Lösung des primalen Problems, so besagt die Aussage, dass stets  $g(y) \geq f(x)$  gilt. D.h. eine zulässige Lösung des dualen Problems liefert eine obere Schranke für den Optimalwert des primalen Problems.

**4.7.** Sie möchten für den Bau einer Turingmaschine bunte Legosteine bestellen. Dazu haben Sie im Internet folgende Lieferanten, deren Lieferkapazitäten und die zugehörigen Lieferkosten ermittelt und (wie für OR-Studenten üblich) fein säuberlich in eine Tabelle eingetragen.

Lieferant	Kosten für Bausteine				Gesamtmenge
	Rot (r)	Grün (g)	Blau (b)	Gelb (y)	
Toys'R U (A)	5	4	3	7	50
Spielmarkt.de (B)	3	2	6	4	75
Bloxx.com (C)	2	4	4	3	45

Sie möchten für den geplanten Bau 80 rote, 60 grüne, 25 blaue und 35 gelbe Steine bestellen.

- Was fällt Ihnen auf, wenn Sie Liefer- und Lagerkapazität aus der gegebenen Auflistung vergleichen und was müssen Sie beim Aufstellen des Transporttableaus deshalb durchführen?
- Lösen Sie das Transportproblem. Starten Sie dabei mit der Nord-West-Eckenregel (NWR) und verfolgen Sie bei der Stepping Stone-Methode die Strategie, den größten  $\delta$ -Wert für den Basiswechsel zu verwenden. Geben Sie schließlich das Ergebnis sinnvoll an.

**Beispiellösung:**

- ad a) Die Summen von Liefermenge (170) und Lagerkapazität (200) sind nicht gleich. Daher muss ein virtueller Lieferant (X) eingeführt werden, der weitere 30 Einheiten zu Kosten 0 liefern kann.

Das entstehende Tableau sieht dann so aus:



	r	g	b	y	
A	5	4	3	7	50
B	3	2	6	4	75
C	2	4	4	3	45
X	0	0	0	0	30
	80	60	25	35	

ad b) Starttableau nach Anwendung der NWR:

	r	g	b	y	
A	5	4	3	7	50
	(50)				
B	3	2	6	4	75
	(30)	(45)			
C	2	4	4	3	45
		(15)	(25)	(5)	
X	0	0	0	0	30
				(30)	
	80	60	25	35	605

Schritt 1:  
 $\delta$ -Werte berechnen.

	r	g	b	y	
A	5 (50)	4 0	3 1	7 -4	50
B	3 (30)	2 (45)	6 -4	4 -3	75
C	2 3	4 (15)	4 (25)	3 (5)	45
X	0 2	0 1	0 1	0 (30)	30
	80	60	25	35	605

$x_{3,1}$  geht in die Basis.

Schritt 2:

Neues Tableau mit verschobenen Transportmengen und  $\delta$ -Werte berechnen.

	r	g	b	y	
A	5 (50)	4 0	3 4	7 -1	50
B	3 (15)	2 (60)	6 -1	4 0	75
C	2 (15)	4 -3	4 (25)	3 (5)	45
X	0 -1	0 -2	0 1	0 (30)	30
	80	60	25	35	560

$x_{1,3}$  geht in die Basis.

Schritt 3:

Neues Tableau mit verschobenen Transportmengen und  $\delta$ -Werte berechnen.

	r	g	b	y	
A	5 (25)	4 0	3 (25)	7 -1	50
B	3 (15)	2 (60)	6 -5	4 0	75
C	2 (40)	4 -3	4 -4	3 (5)	45
X	0 -1	0 -2	0 -3	0 (30)	30
	80	60	25	35	460

Alle  $\delta \leq 0 \Rightarrow$  optimale Lösung erreicht.

**Antwort:** Sie sollten z.B. folgendermaßen einkaufen:

- bei Lieferant A kaufen Sie 25 rote und 25 blaue Steine
- bei Lieferant B kaufen Sie 15 rote und 60 grüne Steine
- bei Lieferant C kaufen Sie 40 rote und 5 gelbe Steine.

Es fehlen noch 30 gelbe Steine. Dafür müssen Sie sich noch einen Lieferanten suchen. Unter den gegebenen Bedingungen ist die gefundene Transportplanung kostenoptimal (460 Kosteneinheiten).

**4.8.** Die Firma LieferMax hat vier Materialstandorte  $S_0$  bis  $S_3$  für Kleinteile zur Herstellung von Automobilen. Es sind 60 Teile bei  $S_0$ , 70 Teile bei  $S_1$ , 35 Teile bei  $S_2$  und 55 Teile bei  $S_3$  verfügbar.

Ein Automobilhersteller möchte seine vier Produktionsstätten mit entsprechenden Kleinteilen von LieferMax beliefern lassen. Bei Werk 1 werden 80 Teile, bei Werk 2 werden 60 Teile, bei Werk 3 werden 15 Teile und bei Werk 4 werden 65 Teile benötigt.

LieferMax hat festgestellt, dass Transportkosten von den Lagern zu dem Werken des Automobilherstellers entsprechend der folgenden Kostentabelle entstehen:

	$W_1$	$W_2$	$W_3$	$W_4$
$S_0$	6	3	2	1
$S_1$	5	4	3	7
$S_2$	3	4	2	4
$S_3$	5	2	3	6

Natürlich möchte LieferMax die Transportkosten zwischen den Standorten und Kundenlagern minimieren. Versuchen Sie mit Transporttableaus, eine optimale Lösung zu ermitteln. Bitte beachten Sie beim Lösungsansatz: Verwenden Sie die

*Nordwestecken-Regel* zum Ermitteln einer Startlösung. Wenn zwei gleiche  $\delta$ -Werte auftauchen, wählen Sie als nächstes jede Zelle mit größeren Kosten.

**Beispiellösung:** Das Transport-Tableau nach Anwendung der Nordwestecken-Regel sieht folgendermaßen aus (erster Satz  $\delta$ -Werte ist bereits eingetragen, der fett gedruckte  $\delta$ -Wert wird für den nächsten Basiswechsel genutzt).

	$W_1$	$W_2$	$W_3$	$W_4$	
$S_0$	6	3	2	1	
	(60)	2	1	4	60
$S_1$	5	4	3	7	
	(20)	(50)	-1	-3	70
$S_2$	3	4	2	4	
	2	(10)	(15)	(10)	35
$S_3$	5	2	3	6	
	2	<b>4</b>	1	(55)	55
	80	60	15	65	1100

Für den Basiswechsel wird wie vorgeschrieben der  $\delta$ -Wert 4 mit dem größeren Kostenwert für genutzt.

	$W_1$	$W_2$	$W_3$	$W_4$	
$S_0$	6	3	2	1	
	(60)	2	5	<b>8</b>	60
$S_1$	5	4	3	7	
	(20)	(50)	3	1	70
$S_2$	3	4	2	4	
	-2	-4	(15)	(20)	35
$S_3$	5	2	3	6	
	-2	(10)	1	(45)	55
	80	60	15	65	1060

	$W_1$	$W_2$	$W_3$	$W_4$	
$S_0$	6	3	2	1	
	(15)	2	-3	(45)	60
$S_1$	5	4	3	7	
	(65)	(5)	-5	-7	70
$S_2$	3	4	2	4	
	<b>6</b>	4	(15)	(20)	35
$S_3$	5	2	3	6	
	-2	(55)	-7	-8	55
	80	60	15	65	700

	$W_1$	$W_2$	$W_3$	$W_4$	
$S_0$	6	3	2	1	
	-6	-4	-3	60	60
$S_1$	5	4	3	7	
	65	5	1	-1	70
$S_2$	3	4	2	4	
	15	-2	15	5	35
$S_3$	5	2	3	6	
	-2	55	-1	-2	55
	80	60	15	65	610

	$W_1$	$W_2$	$W_3$	$W_4$	
$S_0$	6	3	2	1	
	-6	-4	-4	60	60
$S_1$	5	4	3	7	
	50	5	15	-1	70
$S_2$	3	4	2	4	
	30	-2	-1	5	35
$S_3$	5	2	3	6	
	-2	55	-2	-2	55
	80	60	15	65	595

Das Tableau ist optimal, da alle  $\delta$ -Werte negativ sind. Die Kosten für den Transport betragen 595 Kosteneinheiten. Es werden

- von  $S_0$  60 Teilen zu  $W_4$  gebracht
- von  $S_1$  50 Teile nach  $W_1$ , 5 Teile nach  $W_2$ , 15 Teile nach  $W_3$  gebracht
- von  $S_2$  30 Teile nach  $W_1$ , 5 Teile nach  $W_4$  gebracht
- von  $S_3$  55 Teile nach  $W_2$  gebracht.

## Lösungen zu Kapitel 5

5.1. Es sei die folgende Funktion gegeben:

$$f(x_1, x_2, x_3) = x_1 + 2x_3 + x_2x_3 - x_1^2 - x_2^2 - x_3^2.$$

Ermitteln Sie, ob diese Funktion eine kritische Stelle besitzt und überprüfen Sie ggf., ob es sich dabei um ein Maximum oder Minimum handelt.

**Lösungsvorschlag:** Hier führen wir eine Kurvendiskussion mit mehreren Variablen durch (Partielle Ableitung, Hessematrix): In Kurzform sind hier die Schritte angegeben.

Gradient:

$$\nabla f(x) = \begin{pmatrix} 1 - 2x_1 \\ x_3 - 2x_2 \\ 2 + x_2 - 2x_3 \end{pmatrix}$$

Gradienten = 0 setzen führt zu dem kritischen Punkt  $(\frac{1}{2}, \frac{2}{3}, \frac{4}{3})$ .

Die Hesse-Matrix ergibt sich als

$$H(x) = \begin{pmatrix} -2 & 0 & 0 \\ 0 & -2 & 1 \\ 0 & 1 & -2 \end{pmatrix}$$

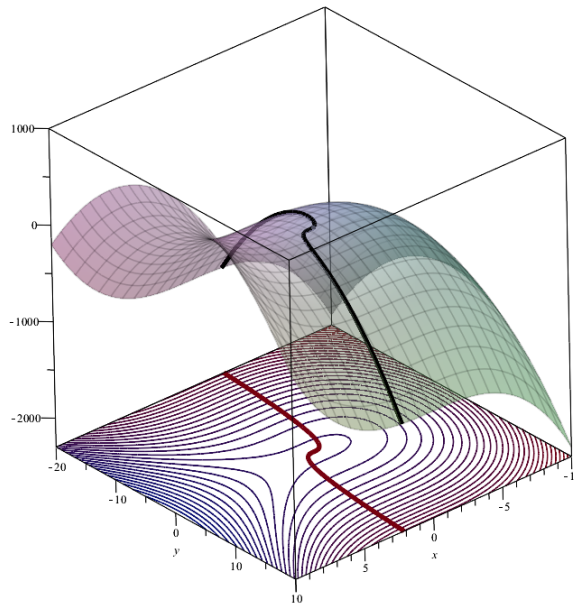
Mit dem Hauptunterdeterminantenkriterium ergibt sich  $\det(H_1) = -2$ ,  $\det(H_2) = 4$ ,  $\det(H_3) = -6$  und damit, dass die Hesse-Matrix negativ definit ( $H_j$  ist neg. definit  $\Leftrightarrow (-1)^j \det(H_j) > 0$ ) ist. Damit liegt ein Maximum im kritischen Punkt vor.

5.2. Gegeben ist das folgende nicht-lineare Optimierungsproblem mit Nebenbedingung:

$$\begin{aligned} f(x_1, x_2) &= x_1^3 - 3x_2^2 \\ h(x_1, x_2) &= 2x_1^3 - x_2 = 0 \end{aligned}$$

- Bestimmen Sie zuerst, ob das Problem ohne Berücksichtigung der Nebenbedingung  $h$  eine kritische Stelle hat, und ob dort ein Optimum vorliegt.
- Bestimmen Sie weiterhin, ob die Funktion unter Berücksichtigung der Randbedingung ein Optimum aufweist.

**Lösungsvorschlag:**



a) Betrachtung des Problems in unbeschränkter Variante liefert:

$$\nabla f(x) = \begin{pmatrix} 3x_1^2 \\ 6x_2 \end{pmatrix}$$

Nach Null-Setzen ergibt sich als einziger Kandidat für einen kritischen Punkt:  $(0, 0)$ .

Die Hesse-Matrix dafür lautet

$$H(x) = \begin{pmatrix} 0 & 0 \\ 0 & 6 \end{pmatrix}$$

und ist indefinit. Es liegt kein Maximum/Minimum vor.

b) Hier wird die Lagrange-Methode angewendet. Wir formulieren die Lagrange-Funktion:

$$L(x_1, x_2, \lambda) = x_1^3 - 3x_2^2 + \lambda(2x_1^3 - x_2)$$

Mit  $\nabla L = 0$  ergibt sich:

$$3x_1^2 + 6x_1^2\lambda = 0$$

$$-6x_2 - \lambda = 0$$

$$2x_1^3 - x_2 = 0$$

Wir ermitteln  $x_1 = 0 \Rightarrow x_2 = 0 \Rightarrow \lambda = 0$ , oder

$$\lambda = -\frac{1}{3} \Rightarrow x_2 = -\frac{1}{18} \Rightarrow x_1 = \sqrt[3]{-\frac{1}{36}}$$

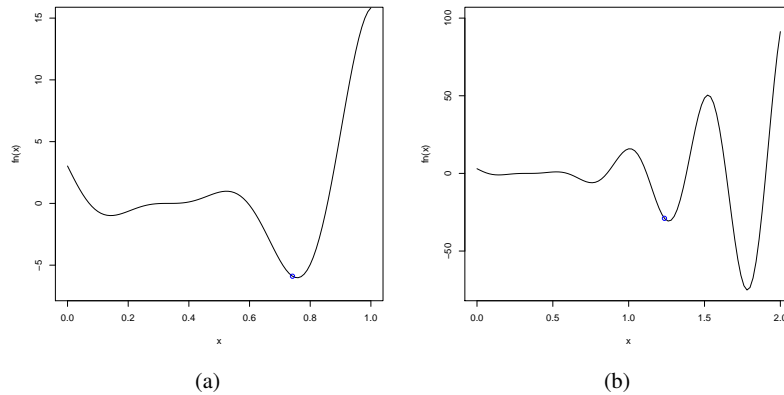
Weiter berechnen wir  $f(0,0) = 0$ , glob. Maximum und  $f(\sqrt[3]{-\frac{1}{36}}, -\frac{1}{18}) = -\frac{1}{27}$  lokales Minimum. Aussage über Grafik validiert.

**5.3.** In dieser Aufgabe soll es um die Optimierung einer eindimensionalen Zielfunktion gehen. Diese sei durch die Funktionsvorschrift

$$f(x) = (6x - 2)^2 \sin(12x - 4)$$

gegeben. Führen Sie jeweils  $N = 10$  Iterationen der Fibonacci-Suche a) mit dem Startintervall  $[0, 1]$  und b)  $[0, 2]$  durch.

**Lösungsvorschlag:** Für das Startintervall  $[0, 1]$  erhalten ergibt sich der folgende Optimierungsverlauf:



**Abb. 5.24** Ergebnisse der Fibonacci-Suche auf dem Startintervall  $[0, 1]$  (links) sowie  $[0, 2]$  rechts.

$k$	$a$	$d$	$c$	$b$	$f(a)$	$f(d)$	$f(c)$	$f(b)$	$\frac{F_{N-(k+1)}}{F_{N-k}}$	
1	1.00	0.00	0.38	0.62	1.00	3.03	0.05	-0.79	15.83	0.62
2	2.00	0.38	0.62	0.76	1.00	0.05	-0.79	-6.00	15.83	0.62
3	3.00	0.62	0.76	0.85	1.00	-0.79	-6.00	-0.35	15.83	0.62
4	4.00	0.62	0.71	0.76	0.85	-0.79	-4.93	-6.00	-0.35	0.62
5	5.00	0.71	0.76	0.80	0.85	-4.93	-6.00	-5.06	-0.35	0.62
6	6.00	0.71	0.74	0.76	0.80	-4.93	-5.90	-6.00	-5.06	0.62
7	7.00	0.74	0.76	0.78	0.80	-5.90	-6.00	-5.84	-5.06	0.60
8	8.00	0.74	0.75	0.76	0.78	-5.90	-6.01	-6.00	-5.84	0.67
9	9.00	0.74	0.75	0.75	0.76	-5.90	-6.01	-6.01	-6.00	0.50
10	10.00	0.74	0.74	0.75	0.75	-5.90	-5.90	-6.01	-6.01	1.00

Suchen wir hingegen im Intervall  $[0, 2]$ , so läuft die Fibonacci-Suche in ein lokales Minimum. Die Fibonacci-Suche kann i.A. nur dann das globale Optimum finden,



wenn  $f$  unimodal ist, d.h. nur ein Optimum/Minimum besitzt. Das gilt insbesondere für konvexe Funktionen, bei denen jedes lokale Optimum zugleich ein globales Optimum darstellt.

$k$	$a$	$d$	$c$	$b$	$f(a)$	$f(d)$	$f(c)$	$f(b)$	$\frac{f_{N-(k+1)}}{f_{N-k}}$	
1	1.00	0.00	0.76	1.24	2.00	3.03	-6.00	-28.94	91.29	0.62
2	2.00	0.76	1.24	1.53	2.00	-6.00	-28.94	50.36	91.29	0.62
3	3.00	0.76	1.06	1.24	1.53	-6.00	12.83	-28.94	50.36	0.62
4	4.00	1.06	1.24	1.35	1.53	12.83	-28.94	-13.98	50.36	0.62
5	5.00	1.06	1.17	1.24	1.35	12.83	-14.13	-28.94	-13.98	0.62
6	6.00	1.17	1.24	1.28	1.35	-14.13	-28.94	-30.07	-13.98	0.62
7	7.00	1.24	1.28	1.30	1.35	-28.94	-30.07	-27.07	-13.98	0.60
8	8.00	1.24	1.26	1.28	1.30	-28.94	-30.64	-30.07	-27.07	0.67
9	9.00	1.24	1.26	1.26	1.28	-28.94	-30.64	-30.64	-30.07	0.50
10	10.00	1.24	1.24	1.26	1.26	-28.94	-28.94	-30.64	-30.64	1.00

#### 5.4. Leiten Sie die Iterationsformel des Newton-Verfahrens

$$x_{k+1} = x_k - \frac{f(x_k)}{f'(x_k)}$$

her. **Hinweis:** Eine Approximation der Funktion  $f$  mittels Taylorpolynom könnte hilfreich sein.

**Beispiellösung:** Sei  $f: \mathbb{R} \rightarrow \mathbb{R}, x \mapsto f(x)$  differenzierbar mit Ableitung  $f'(x)$ . Das Newton Verfahren approximiert in jeder Iteration  $f$  durch die Tangente im Punkt  $x_k$ . Die Formel der Tangente entspricht dem Taylorpolynom ersten Grades mit Entwicklungspunkt  $x_k$ :

$$T_1(x; x_k) = \sum_{l=0}^1 \frac{f^{(l)}(x_k)}{l!} (x - x_k)^l = f(x_k) + f'(x_k)(x - x_k).$$

Die nächste Punkt wird als Nullstelle der Tangente gewählt. Durch gleichsetzen der Tangentengleichung mit Null und Umformung nach  $x$  erhalten wir den rekursiven Newton-Operator:

$$\begin{aligned} T_1(x; x_k) &\stackrel{!}{=} 0 &\Leftrightarrow & f(x_k) + f'(x_k)(x - x_k) = 0 \\ &&\Leftrightarrow & f(x_k) + f'(x_k)x - f'(x_k)x_k = 0 \\ &&\Leftrightarrow & f'(x_k)x = f'(x_k)x_k - f(x_k) \\ &&\Leftrightarrow & x = x_k - \frac{f(x_k)}{f'(x_k)} \end{aligned}$$

#### 5.5. Newton-Verfahren

- a) Implementieren Sie das Newton-Verfahren in einer Programmiersprache Ihrer Wahl.

- b) Betrachten Sie die Funktion  $f : \mathbb{R} \rightarrow \mathbb{R}, x \mapsto 3x^3 - 4x^2 + 2$  über dem Intervall  $I = [-1, 2]$ . Wenden Sie Ihre Implementierung aus a) zur Bestimmung einer Nullstelle von  $f$  an. Starten Sie das Verfahren einmal mit dem Startpunkt  $x_0 = 0.5$ , bis die Nullstelle auf  $\varepsilon = 0.01$  genau bestimmt ist.
- c) Starten Sie das Verfahren nun mit dem Startwert  $x_0 = 1$ . Visualisieren Sie die ersten beiden Iterationen grafisch, d. h., stellen Sie die Funktion sowie die ersten beiden linearen Approximationen dar. Was beobachten Sie? Wie kann das Newton-Verfahren geeignet angepasst werden?
- d) Welche weiteren Punkte führen zu derselben Problematik wie in c)?

**Beispiellösung:**

- ad a) Das folgende Listing zeigt eine beispielhafte Implementierung des Newton-Verfahrens in Python 3. Die Problematik aus Aufgabenteil d) wird dabei nicht berücksichtigt.
- ad b) Nachfolgend die Ausgabe der Python Implementierung für den entsprechenden Aufruf der Methode.

Iter	x	f(x)	f'(x)
1	+0.50000000	+1.37500000	-1.75000000
2	+1.28571429	+1.76384840	+4.59183673
3	+0.90158730	+0.94715320	+0.10303855
4	-8.29063455	-1982.49936056	+684.93666726
5	-5.39620729	-585.87355223	+305.24113625
6	-3.47682788	-172.44047844	+136.60961187
7	-2.21454131	-50.19838811	+61.85406929
8	-1.40297970	-14.15808193	+28.93900585
9	-0.91374099	-3.62839932	+14.82423119
10	-0.66897960	-0.68830761	+9.37964021
11	-0.59559645	-0.05277745	+7.95738771
12	-0.58896394	-0.00041089	+7.83361816

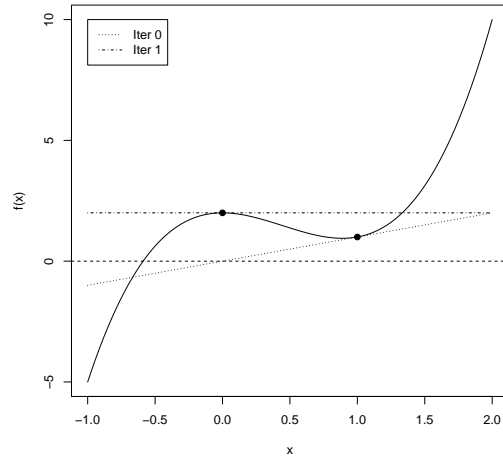
- ad c) Wir erhalten  $x_1 = 1$ . Allerdings ist nun  $f'(x_1) = 0$  und damit ist der Ausdruck  $f(x_1)/f'(x_1)$  im Newton-Operator nicht definiert. Die Ausgabe der obigen Python-Implementierung ist hier:

Iter	x	f(x)	f'(x)
1	+1.00000000	+1.00000000	+1.00000000
2	+0.00000000	+2.00000000	+0.00000000

f1(x) is zero!

In der grafischen Darstellung erkennen wir, dass der Graph der Tangente im Punkt  $x_1 = 1$  eine konstante Funktion ist, die parallel zur  $x$ -Achse verläuft. Entsprechend hat die Tangente auch keine Nullstelle, die das Newton-Verfahren als nächste Approximation der Nullstelle von  $f$  verwenden könnte.

Die obige Python Implementierung wirft in diesem Fall einen *ZeroDivisionError: float division by zero* Fehler. Beheben lässt sich das Problem indem explizit darauf getestet wird. Verschwindet die Ableitung an einem Punkt  $x_k$ , so wählt man z.B. zufällig ein neues  $\tilde{x}_k$  in der Umgebung von  $x_k$ .



**Abb. 5.25** Grafische Darstellung der ersten beiden Iterationen des Newton-Verfahrens zur Bestimmung einer Nullstelle von  $f(x) = 3x^3 - 4x^2 + 2$  mit Initialwert  $x_0 = 1$ .

- ad d) Problematisch sind offensichtlich Optima von  $f$  (bzw. bei der Minimierung von  $f$  die Optima von  $f'$ ), denn eine notwendige Bedingung für ein Optimum ist ja gerade, dass die Ableitung verschwindet. Es ist  $f(x) = 3x^3 - 4x^2 + 2$  und  $f'(x) = 9x^2 - 8x$ . Setzen wir  $f'$  gleich Null so erhalten wir mit ein wenig Rechnung:

$$9x^2 - 8x = 0 \Leftrightarrow x(9x - 8) = 0 \Leftrightarrow x = 0 \vee x = \frac{8}{9}.$$

D. h. wenn das Newton-Verfahren ein  $x_k$  mit  $x_k = \frac{8}{9}$  erzeugt, so kann das Verfahren ohne Anpassung nicht weiterarbeiten.

## Lösungen zu Kapitel 6

**6.1.** Wir wollen die binäre Funktion  $f(x) = \sum_{i=1}^n x_i$  mit  $x \in \{0, 1\}^n$  mit einem evolutionären Algorithmus minimieren. Wir nutzen eine (1 + 1)-Strategie und repräsentieren Lösungskandidaten als Binärstrings. In jeder Iteration wird aus dem Elter  $x_k$  ein neues Individuum  $x_{k+1}$  durch Bitflip-Mutation mit Mutationswahrscheinlichkeit  $p \in (0, 1)$  durchgeführt. Dabei wird jedes Bit unabhängig von den anderen geflippt.  $x_{k+1}$  ersetzt  $x_k$ , falls  $f(x_{k+1}) \leq f(x_k)$  gilt.

- Bestimmen Sie die Wahrscheinlichkeit, dass eine Mutation das Individuum nicht verändert, d.h.  $x_k = x_{k+1}$ .
- Sei  $n \in \mathbb{N}$  gegeben. Zeigen Sie, dass die Wahrscheinlichkeit, die optimale Lösung in einem Schritt zu erreichen, echt größer 0 ist.

### Beispiellösung:

- ad a) Die einzelnen Bits werden voneinander unabhängig mutiert; jedes Bit mit Wahrscheinlichkeit  $p$ . Keines der Bits darf geflippt werden, damit das Individuum gleich bleibt. Dies geschieht für jedes Bit mit Wahrscheinlichkeit  $(1 - p)$ . Aufgrund der Unabhängigkeit folgt daraus  $P(x_{k+1} = x_k) = (1 - p)^n$ .
- ad b) Wir generieren die Initiallösung  $x_0$  zufällig. Sei  $H(x, y) = \#\{i : x_i \neq y_i\}$  der Hamming-Abstand von  $x_0$  zur Optimallösung, d.h., die Anzahl der Stellen an denen sich  $x$  von  $y$ . Dann ist  $H(x, 0)$  der Abstand einer Lösung zum Nullvektor, d.h. der Optimallösung. Es müssten nun alle 1en gekippt werden und alle 0en beibehalten werden. Die Wahrscheinlichkeit dafür ist

$$P(x_0 \rightarrow 0) = p^{H(x,0)} \cdot (1 - p)^{n-H(x,0)} > 0.$$

Die Ungleichung gilt, da  $0 < p < 1$ .

**6.2.** Gegeben seien die folgenden zwei Elternindividuen in Permutationskodierung:

$$p_1 : 1 \ 3 \ 6 \ 9 \ 4 \ 5 \ 2 \ 7 \ 8, \quad p_2 : 9 \ 1 \ 3 \ 4 \ 8 \ 7 \ 6 \ 2 \ 5$$

- Wenden Sie den *Partially-Mapped-Crossover* Operator (PMX) an. Als "Cut-Points" wählen Sie die Positionen hinter Element 3 sowie hinter Element 7. Geben Sie dabei alle Zwischenschritte an.
- Wenden Sie nun den *Ordered-Crossover* Operator (OX) an. Die Cut-Points wählen Sie dabei wie in (a).

**Lösungsvorschlag:** Nachfolgend ist der Ablauf der Rekombinationsoperatoren detailliert dargestellt.

- ad a) Wir vertauschen zunächst die Segmente zwischen den Cutpoints (Schritt 1), kopieren anschließend alle Elter-Allele, die nicht in Konflikt mit dem eingefügten mittleren Segment stehen (Schritt 2), und folgen schließlich der Abbildung  $4 \leftrightarrow 9, 8 \leftrightarrow 4, 7 \leftrightarrow 5, 6 \leftrightarrow 2$ .

<b>Schritt 1</b>	$c_1$ : ? ? ?	4 8 7 6	? ?
	$c_2$ : ? ? ?	9 4 5 2	? ?
<b>Schritt 2</b>	$c_1$ : 1 3 ?	4 8 7 6	? ?
	$c_2$ : ? 1 3	9 4 5 2	? ?
<b>Schritt 3</b>	$c_1$ : 1 3 2	4 8 7 6	5 9
	$c_2$ : 8 1 3	9 4 5 2	6 7

ad b) Zunächst übernehmen wir - analog zum Vorgehen beim PMX-Operator - die Teilsequenzen zwischen den Cut-Points (Schritt 1). Nun bilden wir die Gesamtsequenzen der Elter vom zweiten Cut-Point aus gesehen und entfernen sämtliche Einträge, die vom anderen Elter bereits in den entsprechenden Nachkommen eingefügt wurden. Dies ist 7 8 1 3 6 9 4 5 2 (ungekürzt) und 1 3 9 5 2 (gekürzt) für  $p_1$  sowie 2 5 9 1 3 4 8 7 6 (ungekürzt) und 1 3 8 7 6 (gekürzt) für  $p_2$  (Schritt 2). Diese reduzierten Sequenzen werden nun ab dem zweiten Cut-Point in die Nachkommen eingesetzt (Schritt 3).

<b>Schritt 1</b>	$c_1$ : ? ? ?	9 4 5 2	? ?
	$c_2$ : ? ? ?	4 8 7 6	? ?
<b>Schritt 1</b>	$c_1$ : 8 7 6	9 4 5 2	1 3
	$c_2$ : 9 5 2	4 8 7 6	1 3

**6.3.** Es gilt, eine kontinuierliche Zielfunktion  $f : [L, R] \rightarrow \mathbb{R}$  zu minimieren. Sie haben keinerlei problemspezifisches Wissen und analytische/direkte Verfahren scheiden aus. Ihre Arbeitsgruppe entscheidet sich für den Einsatz eines EA, der ausschließlich reellwertige Mutation  $X_{k+1} = X_k + Z_k$  nutzt (also keine Rekombination). Ein Brainstorming ergab folgende Vorschläge für die Verteilung der Mutation  $Z_k$ :

- Diskrete Gleichverteilung.
- Exponentialverteilung, d.h.  $Z_k \sim \text{Exp}(\lambda)$ ,  $\lambda \in (0, \infty)$ .
- Normalverteilung, d.h.  $Z_k \sim \mathcal{N}(\mu, \sigma^2)$ .

Diskutieren Sie die Vorschläge. Welche Verteilungen sind geeignet und welche nicht? Begründen Sie Ihre Antwort.

**Beispiellösung:** Es handelt sich um eine kontinuierliche Zielfunktion. Alle Werte  $x \in [L, R]$  kommen für ein Optimum in Frage. Allein aus diesem Grund eignet sich die diskrete Gleichverteilung überhaupt nicht, da so nicht alle  $x$  erreichbar wären. Bei der Exponentialverteilung sind kleine Werte wahrscheinlicher als Große. Das ist gut, denn Mutation soll ja i. d. R. gerade kleine Änderungen vornehmen. Jedoch nehmen Exponential-verteilte Zufallszahlen nur positive Werte an, wodurch eine Verzerrung entsteht. Die Normalverteilung ist im Gegensatz zur Exponentialverteilung symmetrisch um  $\mu$  mit steuerbarer Varianz  $\sigma^2$ . Da wird keinerlei spezifisches Wissen über  $f$  haben ist die Normalverteilung hier der beste Vorschlag, allerdings nur mit  $\mu = 0$ .

**6.4.** Sie haben u. a. die *fitnessproportionale Selektion* kennengelernt. Gehen Sie nun davon aus, dass ein Maximierungsproblem vorliegt.

- Wie wirken sich negative Fitnesswerte aus? Finden Sie eine Möglichkeit, dieses Problem zu umgehen.

b) Nennen Sie einen weiteren großen Nachteil der Fitnessproportionalen Selektion.

**Lösungsvorschlag:** Die Fitnessproportionale Selektion ist zwar intuitiv und sehr einfach zu implementieren, hat jedoch einige klare Nachteile.

a Negative Fitnesswerte sind problematisch, da in diesem Fall die Summe der Fitnesswerte gleich 0 werden kann. Angenommen wir haben eine Population mit drei Individuen und Fitnesswerten  $f_1 = -3, f_2 = 1, f_3 = 2$ , so gilt  $\sum f_i = 0$  und damit ist  $p_i$  gar nicht definiert. Die Bildung des Absolutbetrags der Fitnesswerte wäre eine Lösung, allerdings keine besonders gute. In diesem Fall wäre die Selektionswahrscheinlichkeit für Individuum 1 - das schlechteste - mit  $1/2$  deutlich größer als für die besseren Individuen. Eine bessere Lösung besteht darin die Werte um den minimalen Absolutbetrag des kleinsten negativen Wertes plus eine kleine Konstante zu transformieren:

$$\tilde{f}_i = \begin{cases} f_i + |\min(f_i)| + 1, & \text{falls } \exists i : f_i < 0 \\ f_i, & \text{falls } \forall i : f_i \geq 0. \end{cases}$$

In unserem Beispiel wären damit  $\tilde{f}_1 = 1, \tilde{f}_2 = 5, \tilde{f}_3 = 6$ .

b Das weitaus größere Problem der Fitnessproportionalen Selektion besteht in der raschen Fokussierung des Verfahrens auf wenige Individuen mit sehr großer Fitness. Für  $f_1 = 2, f_2 = 5, f_3 = 1000$  besteht nur noch eine sehr geringe Möglichkeit für schlechtere Individuen selektiert zu werden, nämlich  $p_1 = 0.001986, p_2 = 0.004965$ .

**6.5.** Sei eine Population von  $N$  Individuen gegeben mit Fitness-Werten  $f_1, \dots, f_N$ . Wir nehmen an, dass fitnessproportionale bzw. Turnirselektion mit Turniergröße  $k = 2$  (mit/ohne Zurücklegen) als Survival-Selektion zur Selektion von  $m < n$  Individuen benutzt wird. Wie groß ist jeweils die Wahrscheinlichkeit, dass das beste Individuum überlebt? Nehmen Sie der Einfachheit halber an, dass keine Bindungen zwischen den Fitness-Werten existieren, d. h.  $f_i \neq f_j$  für  $1 \leq i < j \leq n$ .

**Beispiellösung:** Zunächst zur Roulette-Rad Selektion. Die Fitness-Werte sind  $f_1, \dots, f_n$ . Daher ist die Wahrscheinlichkeit zur Auswahl von Individuum  $i$  gleich

$$p_i = \frac{f_i}{\sum_{i=1}^n f_i}.$$

Dann ist  $j = \operatorname{argmax}_i f_i, \dots, f_n$  das fitteste Individuum und  $p_j$  die Wahrscheinlichkeit dieses zu wählen. Die Wahrscheinlichkeit dieses Individuum nicht auszuwählen ist  $(1 - p_j)$ . Entsprechend ist die Wahrscheinlichkeit in keinem der  $m$  Schritte das fitteste Individuum zu wählen  $(1 - p_j)^m$ . Die Gegenwahrscheinlichkeit  $1 - (1 - p_j)^m$  entspricht der gesuchten Wahrscheinlichkeit.

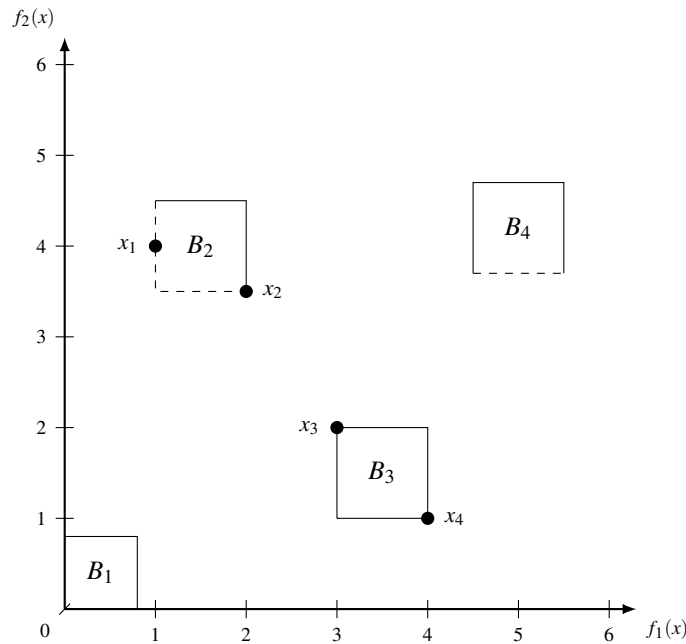
Nur zur Turnirselektion mit Zurücklegen. Gehen wir zunächst von einem einzigen Turnier aus. Die Wahrscheinlichkeit, dass  $j$  nicht unter den  $k = 2$  gleichverteilt ins Turnier gewählten Individuen ist, beträgt  $(1 - \frac{1}{n})^2$ , denn es sind nur die Konstellationen günstig in denen sowohl die erste als auch die zweite Wahl nicht gleich

$j$  ist. Andernfalls wäre mind. eines der gewählten Individuen gleich  $j$  und da das Element mit Rang 1 das Turnier gewinnt würde  $j$  überleben. Die Wahrscheinlichkeit, dass in  $M$  Turnieren  $j$  nicht vorkommt ist damit aufgrund der Unabhängigkeit gleich  $(1 - \frac{1}{n})^{2m}$ . Die Wahrscheinlichkeit, dass das beste Individuum in mindestens einem Turnier gewählt wird ist damit  $1 - (1 - \frac{1}{n})^{2m}$ .

Ohne Zurücklegen verringert sich die Anzahl der Möglichkeiten jeweils um 1. Damit beträgt die Wahrscheinlichkeit, dass  $j$  nicht überlebt  $\prod_{l=1}^m (1 - \frac{1}{n-l+1})$ . Wieder führt der Übergang zur Gegenwahrscheinlichkeit zum Ziel: das beste Individuum wird mit Wahrscheinlichkeit  $1 - \prod_{l=1}^m (1 - \frac{1}{n-l+1})$  selektiert.

## Lösungen zu Kapitel 7

7.1. Betrachten Sie die folgende Abbildung mit der Punktmenge  $X = \{x_1, x_2, x_3, x_4\}$ . Ziel ist die Minimierung beider Ziele.



- a) Welche Aussagen gelten für Punkte in den Bereichen  $B_1$  bis  $B_4$ ? Beachten Sie: Gestrichelte Ränder bedeuten, dass die Punkte auf dem Rand ausgeschlossen werden. Durchgezogene Linien hingegen bedeuten, dass Punkte auf dem Rand zum Bereich gehören.

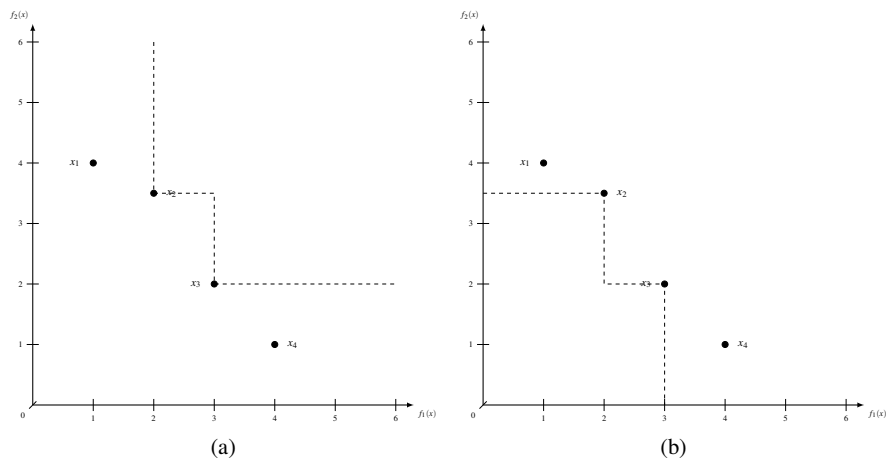
	$B_1$	$B_2$	$B_3$	$B_4$
Enthält Punkte, die mind. ein $x \in X$ dominieren				
Enthält Punkte, die alle $x \in X$ dominieren				
Enthält Punkte, die von mind. einem Punkt $x \in X$ dominiert werden				
Enthält Punkte, die von allen $x \in X$ dominiert werden				

- b) Zeichnen Sie den Bereich der Punkte ein, der (i) von  $x_2$  oder  $x_3$  dominiert wird bzw. (ii) zu  $x_2$  und  $x_3$  unvergleichbar ist bzw. (iii) von mindestens einem der beiden dominiert wird.

**Beispiellösung:** Die nachfolgende Tabelle enthält die korrekten Antworten zu Aufgabenteil a).



	$B_1$	$B_2$	$B_3$	$B_4$
Enthält Punkte, die mind. ein $x \in X$ dominieren	✓	✗	✓	✗
Enthält Punkte, die alle $x \in X$ dominieren	✓	✗	✗	✗
Enthält Punkte, die von mind. einem Punkt $x \in X$ dominiert werden	✗	✓	✓	✓
enthält Punkte, die von allen $x \in X$ dominiert werden	✗	✗	✗	✓



**Abb. 7.4** Bereich der von  $x_2$  oder  $x_3$  dominiert wird (links) bzw. der keinen der Punkte  $x_2, x_3$  dominiert.

**7.2.** Gegeben sei die folgende *Entscheidungsmatrix*. Wir betrachten ein Entscheidungsszenario unter Unsicherheit.

	$z_1$	$z_2$	$z_3$	$z_4$
$a_1$	8	6	4	7
$a_2$	9	5	3	2
$a_3$	4	8	5	3
$a_4$	10	8	5	8
$a_5$	10	9	7	9

- Sei der *Gewichtsvektor*  $w = (4, 5, 2, 1)$  gegeben. Quantifizieren Sie die Ziele mit  $w$ . Für welche Alternative entscheiden Sie sich?
- Sei nun der *Zielvektor*  $\hat{u} = (8, 3, 5, 5)$  gegeben. Nutzen Sie das Prinzip des *Goal-Programmings* und bestimmen Sie die entsprechende Alternative.

**Beispiellösung:** Zu a) Wir bestimmen die skalaren Werte der Nutzenfunktion.

$$\begin{aligned}
u(a_1) &= 4 \cdot 8 + 5 \cdot 6 + 2 \cdot 4 + 1 \cdot 7 = 77 \\
u(a_2) &= 4 \cdot 9 + 5 \cdot 5 + 2 \cdot 3 + 1 \cdot 2 = 69 \\
u(a_3) &= 4 \cdot 4 + 5 \cdot 8 + 2 \cdot 5 + 1 \cdot 3 = 69 \\
u(a_4) &= 4 \cdot 10 + 5 \cdot 8 + 2 \cdot 5 + 1 \cdot 8 = 98 \\
u(a_5) &= 4 \cdot 10 + 5 \cdot 9 + 2 \cdot 7 + 1 \cdot 9 = \mathbf{108}
\end{aligned}$$

Hier ist nun  $107 = u(a_5) = \max_j u(a_j)$ . Demnach ist nach diesem Kriterium die Alternative  $a_5$  zu wählen.

Für b) ist es notwendig die Summe der absoluten Abstände  $u(a_i) = \sum_{j=1}^m |e_{i,j} - u_j|$  zum Referenzvektor zu berechnen. Wir erhalten hier:

$$\begin{aligned}
u(a_1) &= |8 - 8| + |6 - 3| + |5 - 5| + |7 - 5| = \mathbf{6} \\
u(a_2) &= |9 - 8| + |3 - 5| + |3 - 5| + |2 - 5| = 8 \\
u(a_3) &= |4 - 8| + |8 - 3| + |5 - 5| + |3 - 5| = 11 \\
u(a_4) &= |10 - 8| + |8 - 3| + |5 - 5| + |8 - 5| = 10 \\
u(a_5) &= |10 - 8| + |9 - 3| + |7 - 5| + |9 - 5| = 14
\end{aligned}$$

Der minimale Abstand ist hier  $u(a_1) = 6$ . Folglich entscheiden wir uns hier für Alternative 1.

**7.3.** Sie bereiten sich auf die OR-Klausur vor. Leider haben Sie sich einen Infekt eingefangen und beschließen, zum Arzt zu gehen. Da Sie keinen Termin vereinbart haben, wissen Sie nicht, wie lange die Wartezeit (WZ) betragen wird. Um eine eventuelle Wartezeit sinnvoll zu nutzen, überlegen Sie, ihr schweres OR-Buch mitzunehmen. Dieses wiegt recht viel und sie werden es nur dann brauchen, wenn tatsächlich gewartet werden muss. Sie stellen folgende Entscheidungsmatrix auf und quantifizieren ihre Zufriedenheit mit Ihrer Entscheidung und dem Umweltzustand numerisch, wobei höhere Zahlen Zufriedenheit, niedrigere oder gar negative Zahlen Unzufriedenheit ausdrücken.

	$z_1 = \text{keine WZ}$	$z_2 = \text{mittlere WZ}$	$z_3 = \text{lange WZ}$
$a_1 = \text{Buch mitnehmen}$	3	5	8
$a_2 = \text{Buch nicht mitnehmen}$	10	-5	-10

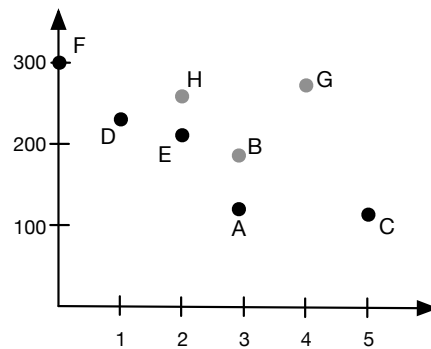
- Wählen Sie mit der MaxiMin-Regel die Alternative mit dem besten unter den schlechten Ergebnissen.
- Wenden Sie nun die Laplace-Regel zur Entscheidungsfindung an.
- Aus Erfahrung wissen Sie, dass das Glück meistens auf Ihrer Seite ist und Sie nur kurz warten müssen. Seien  $p_1 = 0.73$ ,  $p_2 = 0.1$  und  $p_3 = 0.17$  die Eintrittswahrscheinlichkeiten. Welche Alternative wird nach der Bayes-Regel ausgewählt?

**Beispiellösung:**

- ad a) Die Maximin-Regel lautet formal  $\max_{a \in A} (\min_{z \in Z} (e_{a,z}))$ . Das Minimum der ersten Alternative ist **3**, jenes der zweiten Alternative  $-10$ . Wir würden uns demnach entscheiden das Buch mitzunehmen.

- ad b) Die Nutzenfunktion nach der Laplace-Regel entspricht dem arithmetischen Mittel der Konsequenzen, d.h.  $u(a_i) = \frac{1}{m} \sum_{j=1}^m e_{a_i, z_j}$ . Es ist damit  $u(a_1) = \mathbf{16/3}$  und  $u(a_2) = -5/3$ . Der größte Nutzen-Mittelwert ergibt sich also für Alternative 1.
- ad c) Wir berechnen den Erwartungswert als Nutzen  $u(a_i) = \sum_{j=1}^m p_j \cdot e_{a_i, z_j}$ , also den mit den Eintretenswahrscheinlichkeiten gewichteten Mittelwert. Wir erhalten  $u(a_1) = 0.73 \cdot 3 + 0.1 \cdot 5 + 0.17 \cdot 8 = 4.05$  und  $u(a_2) = 0.73 \cdot 10 - 0.1 \cdot 5 - 0.17 \cdot 10 = \mathbf{5.1}$ . Hier würden wir uns für die zweite Alternative entscheiden.

**7.4.** Sie planen eine Reise von  $A$  nach  $B$  per Zug. Die Reise sollte zum einen nicht zu viel kosten. Weiterhin möchten Sie ungern oft umsteigen müssen. Im Zeitraum Ihrer Abreise stehen die folgenden Zugverbindungen zur Verfügung, wobei die erste Komponente stets die Reisekosten in Euro und die zweite Komponente die Anzahl der nötigen Umstiege beschreibt:  $A = (120, 3)$ ,  $B = (190, 3)$ ,  $C = (110, 5)$ ,  $D = (230, 1)$ ,  $E = (210, 2)$ ,  $F = (300, 0)$ ,  $G = (280, 4)$ ,  $H = (260, 2)$ . Selbstverständlich gilt es, sowohl Fahrtkosten als auch die Anzahl der Zugwechsel zu minimieren. Welche Alternativen werden (strikt) dominiert? Stellen Sie die Pareto-optimalen Alternativen grafisch dar.



**Abb. 7.5** Darstellung der Alternativen zur Reiseplanung in 2D. Schwarze Punkte sind nicht-dominiert und damit Pareto-optimal. Graue Punkte werden von mindestens einem anderen Punkt dominiert.

**Beispiellösung:** Die Punkte  $A, C, D, E$  sind Pareto-optimal, da sie jeweils nicht dominiert werden.  $H$  wird von  $D$  strikt dominiert und von  $E$  dominiert. Letztere Dominanzrelation ist jedoch nicht strikt, da die Anzahl der notwendigen Umstiege gleich ist.  $A$  dominiert (nicht strikt)  $B$  und  $G$  strikt.  $G$  wird weiterhin von  $B, E, H$  und  $D$  strikt dominiert. Die grafische Darstellung findet sich in Abbildung 7.5.