

Operations Research

Wolfram Hirsching

21. Juli 2022

Inhaltsverzeichnis

Algorithmenverzeichnis

1	Primaler Simplexalgorithmus	7
2	Dualer Simplexalgorithmus	12
3	Allgemeiner Simplexalgorithmus	16
4	Verkürzter Simplexalgorithmus	18
5	Stepping-Stone	37
6	MODI	40
7	Branch & Bound	48
8	Branch & Bound, allgemeiner	51
9	Rucksack-Problem	53
10	Dijkstra	60
11	FIFO	64
12	Tripel	68
13	Kruskal	73
14	Prim	75

Was ist Operations Research?

Unter Operations Research wird eine Sammlung von mathematischen Optimierungsmethoden verstanden, die bei quantifizierbaren Problemen eine Planungs- und Entscheidungsfindung erleichtern. Ein umfassenderes Verständnis des Begriffs bezieht auch die vorhergehende Analyse des Problems und die Datenbeschaffung mit ein. Im Rahmen dieser Mathematikvorlesung werden die verschiedenen mathematischen Methoden behandelt.

Vorbemerkung

Das Manuskript umreißt die Inhalte der Vorlesung, ersetzt aber nicht ein weit ausführlicheres Lehrbuch.

Speziell empfohlen wird hier das Buch von Prof. Gert Heinrich[?]. Es deckt den Inhalt dieser Vorlesung nahezu vollständig ab und zeichnet sich durch anschauliche und vollständig durchgerechnete Beispiele aus. Große Teile dieses Manuskripts sind an [?] angelehnt, die Beispiele unterscheiden sich.

Lineare Optimierung

Eine Aufgabenstellung der linearen Optimierung könnte sein: In einer Firma werden zwei Arten von Farbe hergestellt. Die *Herstellung* kostet 3 bzw. 2 GE (Geldeinheiten) pro Liter. Pro Tag stehen 28 GE zur Verfügung. Die *Arbeitszeit* beträgt 1 bzw. 2 Stunden pro Liter. Pro Tag arbeiten 2 Arbeitskräfte zu je 8 Stunden. Aufgrund von Kapazitätsbeschränkungen können von der *zweiten Farbe* nur 7 Liter pro Tag hergestellt werden. Der Erlös beträgt 3 bzw. 4 GE. Gesucht ist die Menge, die von jeder Farbe hergestellt werden muss, dass der Erlös maximiert wird.

Allgemein heißt dies: bei der linearen Optimierung wird eine Kombination von Variablen x_i gesucht, so dass eine lineare Funktion, die *Zielfunktion*, möglichst groß wird, z.B. $F = 5x_1 + 3x_2$. Für die x_i gelten Einschränkungen der Art $x_1 + 3x_2 \leq 5$ sowie die *Nicht-negativitätsbedingung* $x_i \geq 0$. Die Einschränkungen („Nebenbedingungen“) können sich z.B. aus Beschränkungen von Ressourcen, Geldmitteln oder Arbeitszeit ergeben. Situationen dieser Art treten häufig auf, z.B. durch Lieferbedingungen, Maschinenauslastungen

oder Arbeitszeitbeschränkungen. Typischerweise kommt keine negative Anzahl von erzeugten Produkten oder geleisteten Arbeitsstunden vor.

Allgemein ist ein lineares Optimierungsproblem (oder „lineares Programmierungsproblem“) wie folgt definiert:

$$F = c_1 \cdot x_1 + c_2 \cdot x_2 + c_3 \cdot x_3 + c_4 \cdot x_4 + \dots + c_p \cdot x_p \rightarrow \max$$

$$a_{11}x_1 + a_{12}x_2 + \dots + a_{1p}x_p \leq b_1$$

$$a_{21}x_1 + a_{22}x_2 + \dots + a_{2p}x_p \leq b_2$$

$$a_{31}x_1 + a_{32}x_2 + \dots + a_{3p}x_p \leq b_3$$

$$a_{41}x_1 + a_{42}x_2 + \dots + a_{4p}x_p \leq b_4$$

$$\vdots$$

$$x_i \geq 0 \quad \forall i \quad [„\forall“ \text{ bedeutet „für alle“}]$$

Formulierungen, die sich nicht an dieses Schema halten, werden in den Kapiteln ??, ?? und Ergänzungen Kap.?? behandelt. Wenn zusätzlich Ganzzahligkeit gefordert wird, ist das Problem komplizierter. Dies ist das Thema von Kapitel ??.

1 Grafische Lösung von linearen Optimierungsproblemen

Im Fall von nur zwei Variablen x_1 und x_2 lässt sich ein lineares Optimierungsproblem grafisch lösen. Obwohl dieser einfache Fall in der Praxis kaum eine Rolle spielt, ist er sehr hilfreich für das grundsätzliche Verständnis der Problematik.

Beispiel: Das oben beschriebene Problem lässt sich mathematisch durch folgende Bedingungen formulieren:

$$x_1 + 2x_2 \leq 16$$

$$3x_1 + 2x_2 \leq 28$$

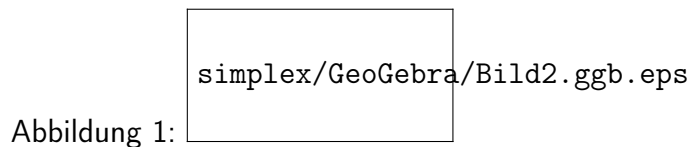
$$x_2 \leq 7$$

$$x_1, x_2 \geq 0$$

Die Zielfunktion $G(x_1, x_2) = 3x_1 + 4x_2$ soll maximiert werden.

x_1 und x_2 ist die Menge (Anzahl Liter) die von Farbe 1 und 2 pro Tag hergestellt wird.

Wenn statt der Ungleichheitszeichen zunächst Gleichheitszeichen verwendet werden, ergeben sich 5 Geradengleichungen. Auch die Zielfunktion lässt sich als Gerade formulieren: $3x_1 + 4x_2 = G$ mit der noch unbekannten (und zu maximierenden) Größe G .



Die 5 Bedingungen ergeben die drei Geraden a, b und c sowie die beiden Koordinatenachsen. Die Lösungsmenge liegt innerhalb des Fünfecks. Man spricht hier von "zulässigen Lösungen". Die Zielfunktion führt auf die Gerade d, die hier mit beliebigem G gezeichnet ist. Eine Änderung von G bedeutet eine Parallelverschiebung von d. Um eine Funktion einer zulässigen Lösung zu bekommen, muss d das Fünfeck schneiden oder zumindest berühren. Für ein möglichst großes G muss die Gerade d so lange parallel verschoben werden, bis sie das Fünfeck berührt. Hier ist das in Punkt A (6|5) der Fall.

Normalerweise berührt die optimale Zielfunktionsgerade die Fläche der zulässigen Lösungen in einem Punkt. Im Spezialfall einer Zielfunktionsgeraden, die parallel zu einer Begrenzung liegt, wird die Fläche auf einem endlichen Geradenstück berührt. Damit gibt es unendlich viele optimale Lösungen.

Bei Problemen dieser Art ist die Fläche der Lösungsmenge (das „Lösungspolyeder“) immer „konvex“, d.h. jede Verbindungsstrecke zweier Eckpunkte ist entweder eine Begrenzung der Fläche oder sie liegt innerhalb der Fläche.

2 Die Simplex Methode

Um eine rechnerische Methode zur Lösung von linearen Optimierungsproblemen zu bekommen, müssen zulässige Lösungen in die Zielfunktion eingesetzt werden, bis diese ein Maximum ergibt. Tatsächlich besteht die Menge der zulässigen Lösungen aus unendlich

vielen Punkten. Allerdings können wir den Überlegungen von Kapitel ?? entnehmen, dass es ausreicht, die Eckpunkte der Fläche der zulässigen Lösungen zu untersuchen. Im allgemeinen Fall von n Variablen handelt es sich jedoch nicht um Schnittpunkte von zwei Geraden in einem zweidimensionalen Variablenraum, sondern um Schnittpunkte von m Hyperebenen in einem n -dimensionalen Raum.

2.1 Der primale Simplexalgorithmus

Hier besteht das System aus Bedingungen der Form \leq sowie den Nichtnegativitätsbedingungen für die Variablen. Die rechten Seiten sind positive Zahlen. Dies entspricht dem Beispiel aus Kapitel ??.

Um ein Gleichungssystem zu erstellen, das den Ungleichungen entspricht, werden "Schlupfvariablen" eingeführt. Für das Beispiel aus Kap. ?? ergibt sich:

$$\begin{aligned}x_1 + 2x_2 + y_1 &= 16 \\3x_1 + 2x_2 + y_2 &= 28 \\x_2 + y_3 &= 7 \\x_1, x_2, y_1, y_2, y_3 &\geq 0\end{aligned}$$

Damit liegt ein unterbestimmtes Gleichungssystem vor mit 3 Gleichungen und den 5 Unbekannten $x_1; x_2; y_1; y_2; y_3$. Ein solches Gleichungssystem hat im allgemeinen unendlich viele Lösungen.

Die gegebene Form wird auch als "reduziertes lineares Gleichungssystem" bezeichnet. Sie zeichnet sich dadurch aus, dass jede Gleichung eine Variable enthält, die in den anderen Gleichungen nicht vorkommt und den Vorfaktor 1 hat. Dies sind hier die Variablen y_1 , y_2 und y_3 . Diese Variablen sind zu Beginn die "Basisvariablen", im Folgenden BV. Die anderen heißen "Nichtbasisvariablen", im Folgenden NBV. Allgemein gibt es immer so viele BV, wie unabhängige Bedingungen befriedigt werden müssen - außer den Nichtnegativitätsbedingungen. Eine Lösung des Gleichungssystems erhält man sehr einfach durch Nullsetzen der NBV. Die BV müssen dann den Wert der rechten Seite der zugehörigen Gleichung annehmen. Damit ist bereits ein Eckpunkt des Lösungspolyeders gefunden: der Ursprung. Das Simplexverfahren stellt eine Methode zur Verfügung, mit der ausgehend von einer solchen zulässigen Lösung ein weiterer Eckpunkt bestimmt wird, für den der Wert der Zielfunktion größer wird. In einer endlichen Anzahl von Schritten kommt man so zu einem optimalen Eckpunkt.

Allgemein gilt, dass so viele Variablen bestimmt werden müssen, wie Gleichungen vorliegen. Alle anderen Variablen sind in den Eckpunkten des Polyeders immer Null.

Die Variablen die in der Zielfunktion vorkommen, heißen "Strukturvariablen". In Kapitel ?? war die Zielfunktion $G(x_1, x_2) = 3x_1 + 4x_2$. Anfangs sind die Strukturvariablen also die NBV. Letztere werden im Lauf der Rechnungen (teilweise) gegen BV getauscht.

Beispiel: Aus 3 Rohstoffen R_1 , R_2 und R_3 sollen drei Endprodukte E_1 , E_2 und E_3 hergestellt werden. Die folgende Tabelle gibt an, wie viele Rohstoffe (in ME) jeweils für ein Endprodukt gebraucht werden und wie viele Rohstoffe vorhanden sind, sowie den Erlös für jedes einzelne Endprodukt (in GE):

Rohstoff	gebrauchte Menge			vorhandene Menge
	E_1	E_2	E_3	
R_1	2	1	6	300
R_2	6	5	2	540
R_3	4	2	4	320
Erlös	10	6	4	

Der Erlös soll maximiert werden, wobei die einzelnen Rohstoffe nicht vollständig aufgebraucht werden müssen. Damit ergibt sich zunächst ein Ungleichungssystem:

$$2x_1 + x_2 + 6x_3 \leq 300$$

$$6x_1 + 5x_2 + 2x_3 \leq 540$$

$$4x_1 + 2x_2 + 4x_3 \leq 320$$

sowie die zu maximierende Zielfunktion $G = 10x_1 + 6x_2 + 4x_3$.

x_i sind die Mengen der hergestellten Endprodukte E_i . Für eine grafische Darstellung bräuchte man ein dreidimensionales Koordinatensystem. Die Bedingungen kann man sich hier noch als Ebenen vorstellen. Je drei Ebenen schneiden sich in einem (Eck-) Punkt. Bei mehr als 3 Dimensionen versagt die Vorstellung.

Das Ungleichungssystem wird mit Hilfe von Schlupfvariablen in ein Gleichungssystem umgewandelt:

$$2x_1 + x_2 + 6x_3 + y_1 = 300$$

$$6x_1 + 5x_2 + 2x_3 + y_2 = 540$$

$$4x_1 + 2x_2 + 4x_3 + y_3 = 320$$

x_1, x_2, x_3 sind die Strukturvariablen. y_1, y_2, y_3 sind die Schlupfvariablen. Jede Gleichung bekommt also ihre eigene Schlupfvariable. Sie repräsentieren in unserem Beispiel ungenutzte Ressourcen. Zu Beginn sind sie immer die Basisvariablen. Damit sind zunächst alle Ressourcen ungenutzt. Dies ändert sich bei jeder Iteration. Gesamtsituation in Tabellenform (Simplextableau):

	x_1	x_2	x_3	y_1	y_2	y_3	rechte Seite
y_1	2	1	6	1	0	0	300
y_2	6	5	2	0	1	0	540
y_3	4	2	4	0	0	1	320
G	-10	-6	-4	0	0	0	0

Die letzte Zeile entspricht der Zielfunktion, wobei die Vorzeichen der Koeffizienten umgedreht sind. Dies ist mathematisch nicht zwingend erforderlich, entspricht aber einem üblichen Vorgehen beim Simplexalgorithmus. Die erste Spalte enthält den Namen der Basisvariable, deren Wert in der letzten Spalte steht. Die Nichtbasisvariablen sind immer Null. Die erste Lösung ist also $\mathbb{L} = \{0, 0, 0, 300, 540, 320\}$; Reihenfolge: $x_1; x_2; x_3; y_1; y_2; y_3$. Damit befinden wir uns im Ursprung des Koordinatensystems. Die Zielfunktion hat den Wert 0. Diese Lösung ist zulässig (erfüllt alle Bedingungen), ist aber sicher nicht optimal. Tatsächlich ist sie die schlechteste aller zulässigen Lösungen. Ziel ist es nun, eine Ecke zu bestimmen, bei der die Zielfunktion größer ist. Eine solche Ecke bekommt man durch Austausch einer BV mit einer NBV. Dieser Schritt entspricht einem Austauschschritt beim Lösen eines linearen Gleichungssystems. Um die optimale Lösung zu erhalten, müssen üblicherweise mehrere BV ausgetauscht werden. Dies erfolgt mit dem Simplexalgorithmus.

Algorithmus 1 Primaler Simplexalgorithmus

1. In der letzten Zeile müssen negative Elemente vorkommen, sonst ist keine Verbesserung der Lösung möglich. Man wählt das kleinste Element, d.h. das Element, das am weitesten im negativen Bereich liegt. Die zugehörige Spalte heißt "Pivotspalte". Hier: erste Spalte mit -10 in der Zielfunktionszeile
2. Die Werte der rechten Seite müssen *positiv* sein. Man bildet alle Quotienten aus dem Wert der rechten Seite und dem Wert auf der Kreuzung dieser Zeile und der Pivotspalte. Negative Quotienten werden ignoriert.
Hier: $300/2=150$; $540/6=90$ und $320/4=80$

3. Der kleinste dieser (positiven) Quotienten bestimmt die "Pivotzeile". Negative Quotienten werden ignoriert.

Hier: kleinster Wert ist 80, die dritte Zeile ist die Pivotzeile.

4. Die Zahl, auf der Kreuzung von Pivotzeile und Pivotspalte, heißt "Pivotelement". Nun wird die BV der Pivotzeile durch die NBV der Pivotspalte ersetzt. Am Ende der Iteration¹ (nach Schritt 6) wird diese dann neue BV sein.

Hier: 4 ist Pivotelement; in der Variablenspalte (Spalte 1) wird " y_3 " durch " x_1 " ersetzt.

5. Die Elemente der Pivotzeile (einschließlich der rechten Seite und des Pivotelements selbst) werden durch das Pivotelement geteilt.

Hier: $2 \rightarrow \frac{2}{4} = \frac{1}{2}$; $4 \rightarrow \frac{4}{4} = 1$; $1 \rightarrow \frac{1}{4}$ und $320 \rightarrow \frac{320}{4} = 80$

6. Alle anderen Elemente werden nach folgendem Schema (Rechteckregel) berechnet: die Elemente der Matrix seien a_{ij} ; die Pivotzeile sei Zeile k ; die Pivotspalte sei Spalte l ;

das Pivotelement ist also a_{kl} . Die Elemente der rechten Seite seien b_i . Dann gilt:

$$a_{ij} \rightarrow a_{ij} - \frac{a_{il} \cdot a_{kj}}{a_{kl}} \text{ und } b_i \rightarrow b_i - \frac{a_{il} \cdot b_k}{a_{kl}}$$

Damit wird das Pivotelement 1 und die anderen Werte der Pivotspalte werden 0.

hier exemplarisch: Zeile 1 ; Spalte 2: $1 \rightarrow 1 - \frac{2 \cdot 2}{4} = 0$; Zeile 2, Spalte 3: $2 \rightarrow 2 - \frac{6 \cdot 4}{4} = -4$;

Zeile 2, Spalte 5: $1 \rightarrow 1 - \frac{6 \cdot 0}{4} = 1$ d.h. die BV bleiben erhalten, außer der einen in der Pivotzeile.

Das neue Simplextableau lautet nach der vollständigen Umrechnung

	x_1	x_2	x_3	y_1	y_2	y_3	rechte Seite
y_1	0	0	4	1	0	-0.5	140
y_2	0	2	-4	0	1	-1.5	60
x_1	1	0.5	1	0	0	0.25	80
G	0	-1	6	0	0	2.5	800

Die Zielfunktion hat nun den Wert 800. D.h. der Gewinn ist nun 800 anstatt 0. Es gibt aber noch einen negativen Wert in der Zielfunktionszeile. In einer neuen Iteration müssen sämtliche Schritte noch einmal durchlaufen werden mit dem Pivotelement 2 in Zeile 2

¹von lat. iterare ‚wiederholen‘; hier bezieht sich „Iteration“ auf jeden Durchgang

und Spalte 2. Damit ergibt sich das neue Simplextableau:

	x_1	x_2	x_3	y_1	y_2	y_3	rechte Seite
y_1	0	0	4	1	0	-0.5	140
x_2	0	1	-2	0	0.5	-0.75	30
x_1	1	0	2	0	-0.25	0.625	65
G	0	0	4	0	0.5	1.75	830

Nun sind alle Elemente der Zielfunktionszeile nicht negativ. Es handelt sich um die optimale Lösung. x_3 wurde nicht ausgetauscht. Der Gewinn ist maximiert und beträgt 830GE.

Bemerkung: Die Schritte 1-?? dienen der Vorbereitung des eigentlichen Austauschschrittes, der in den zwei weiteren Schritten ?? und ?? durchgeführt wird. Schritt 5 entspricht der Multiplikation einer Gleichung mit einer Zahl. In unserem Fall ist das Ziel, in einer Spalte eine 1 stehen zu haben.

In Schritt 6 wird von jeder Zeile (außer der Pivotzeile) ein geeignetes Vielfaches der Pivotzeile abgezogen, so dass in der Pivotspalte Nullen stehen.

Übung: Berechnen Sie das Beispiel von Kap.?? mit der Simplexmethode.

2.1.1 Interpretation des Endtableaus

1. Ergebnisse (Ergebnisspalte; rechte Seite):

- (a) 830 in der Zielfunktionszeile ist der maximal erzielbare Erlös.
- (b) 30 und 65 sind die Werte für x_2 und x_1 , also die Mengeneinheiten der hergestellten Endprodukte E_2 und E_1 .
- (c) $y_1 = 140$ besagt, dass von Rohstoff R_1 noch 140 unverbrauchte Einheiten übrig sind.

2. Zielfunktionszeile (G-Zeile):

- (a) Die Nullen in den Spalten für x_1 , x_2 und y_1 deuten an, dass diese Größen bereits durch die Ergebnisspalte festgelegt sind.
- (b) Die Strukturvariable x_3 ist im Endtableau immer noch NBV, also Null. D.h. Endprodukt E_3 wird nicht hergestellt, wenn der Erlös maximiert werden soll.

Die 4 in der Zielfunktionszeile besagt, dass der Erlös um 4GE sinkt, wenn wider besseres Wissen doch eine Einheit von E_3 hergestellt wird.

In anderen Worten: E_3 würde nur hergestellt werden, wenn die Produktionskosten um mindestens diesen Betrag sinken würden oder der Erlös durch dieses Produkt um mindestens diesen Betrag erhöht würde. Man spricht auch von *reduzierten Kosten*.

- (c) 0.5 in der y_2 -Spalte und 1.75 in der y_3 -Spalte sind *Schattenvariablen*. Sie geben an, um wieviel der Erlös steigen würde, wenn doch eine Einheit mehr von R_2 bzw. R_3 zur Verfügung stehen würde. D.h. wenn eine Einheit von R_2 für weniger als 0.5GE erhältlich ist, lohnt sich der Zukauf. Entsprechendes gilt für R_3 zum Preis von weniger als 1.75GE.

Man spricht auch vom *Schattenpreis*, *Opportunitätskosten*, *Effizienzpreis* oder vom *Knappheitsgrad*.

3. Auch die Zahlen im Inneren des Tableaus lassen sich verstehen:

- (a) Die Spalten für x_1 , x_2 und y_1 enthalten eine 1 und sonst Nullen. Diese Größen sind bereits durch die Ergebnisspalte festgelegt (s. 1b und 1c).
- (b) Die Existenz einer x_3 -Spalte (bzw. Nichtexistenz einer x_3 -Zeile) bedeutet zunächst, dass x_3 NBV geblieben ist, also $x_3 = 0$ (s. 2b). Würde man doch eine Einheit von E_3 herstellen, bedeutet dies, dass

- i. der Erlös um 4GE sinkt, s. 2b.
- ii. x_1 um 2 *kleiner* wird, von Endprodukt E_1 also nur noch 2 Einheiten weniger hergestellt werden können.
- iii. x_2 um 2 *größer* wird, von Endprodukt E_2 also 2 Einheiten mehr hergestellt werden können. Da E_1 den höheren Stückpreis hat, führt dies im Endeffekt zum erwähnten Verlust von 4GE: $-2 \cdot 10 + 2 \cdot 6 + 1 \cdot 4 = -4$

- (c) Die Werte in der y_2 -Spalte und der y_3 -Spalte ergänzen die Aussage 2c.

Wären von Rohstoff 2 541ME statt 540ME vorhanden, so könnten 0.5ME von Endprodukt 2 mehr produziert werden. Allerdings müsste man Endprodukt 1 um 0.25ME verringern. Für den Erlös bedeutet dies: $\frac{1}{2} \cdot 6 - \frac{1}{4} \cdot 10 = +0.5$, was 2c entspricht.

Analog gilt: Wären von Rohstoff 3 321ME statt 320ME vorhanden, so könnten 0.625ME von Endprodukt 1 mehr produziert werden. Allerdings müsste man Endprodukt 2 um 0.75ME verringern. Für den Erlös bedeutet dies:

$0.625 \cdot 10 - 0.75 \cdot 6 = +1.75$, was $2c$ entspricht. Außerdem würde von Rohstoff 1 zusätzlich 0.5ME verbraucht werden, so dass nur noch 139.5ME übrig bleiben .

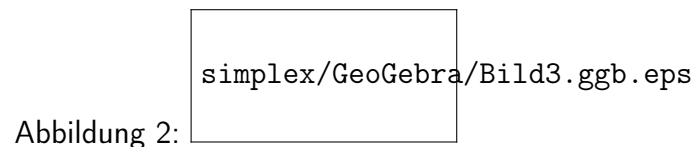
2.2 Der duale Simplexalgorithmus

Auch hier liegen alle Bedingungen in Form von Ungleichungen vor. Es dürfen jedoch auch \geq -Bedingungen vorkommen. Diese werden mit -1 durchmultipliziert und werden somit zu \leq -Bedingungen. Nachteil: die rechte Seite ist jetzt negativ, was beim primalen Simplex-Algorithmus verboten ist (vergleiche Schritt ??).

Beispiel: Gegeben ist folgendes Ungleichungssystem:

$$\begin{aligned}x_1 + 2x_2 &\leq 16 \\3x_1 + 2x_2 &\leq 28 \\x_2 &\leq 7 \\x_1 + x_2 &\geq 3 \\x_1, x_2 &\geq 0\end{aligned}$$

Die Zielfunktion $G(x_1, x_2) = 3x_1 + 4x_2$ soll maximiert werden. Es handelt sich um eine Erweiterung des Beispiels aus Kap. ??. Die entsprechende Forderung könnte lauten, dass mindestens 3 Liter Farbe pro Tag hergestellt werden müssen. Die grafische Lösung ginge von folgender Abbildung aus:



Die Gerade e stellt eine untere Grenze dar. Der Ursprung ist also nicht mehr in der Menge der zulässigen Lösungen enthalten, so dass eine andere Anfangslösung ermittelt werden muss.

Das Gleichungssystem mit den Schlupfvariablen lautet:

$$\begin{aligned}x_1 + 2x_2 + y_1 &= 16 \\3x_1 + 2x_2 + y_2 &= 28 \\x_2 + y_3 &= 7 \\-x_1 - x_2 + y_4 &= -3 \\x_1, x_2, y_1, y_2, y_3, y_4 &\geq 0\end{aligned}$$

Simplextableau:

	x_1	x_2	y_1	y_2	y_3	y_4	rechte Seite
y_1	1	2	1	0	0	0	16
y_2	3	2	0	1	0	0	28
y_3	0	1	0	0	1	0	7
y_4	-1	-1	0	0	0	1	-3
G	-3	-4	0	0	0	0	0

Ziel ist es, durch geeignete Vorarbeit zu einem Simplextableau zu kommen, auf das der primale Simplexalgorithmus angewendet werden kann, d.h. eine zulässige Anfangslösung zu bestimmen. Dann sind alle rechten Seiten wieder positiv. Dies geht in mehreren Schritten:

Algorithmus 2 Dualer Simplexalgorithmus

1. Von allen negativen Werten der rechten Seite wird der kleinste Wert bestimmt. Er legt die Pivotzeile fest.
Hier: 4. Zeile mit -3 auf der rechten Seite
2. Es muss mindestens ein negativer Wert in der Pivotzeile vorkommen, sonst gibt es keine Lösung. Ein beliebiger dieser negativen Werte kann als Pivotelement verwendet werden.
Hier sind dies die Zahlen : -1 in der 1. Spalte und -1 in der 2. Spalte
3. In manchen Fällen ist es vorteilhaft, aus dieser Vorauswahl den kleinsten Zielfunktionswert zur Bestimmung der Pivotspalte zu verwenden.
Hier wird entgegen dieser Regel die erste Spalte gewählt.
4. Das Pivotelement ist der Wert, der sowohl in der Pivotzeile als auch der Pivotspalte steht. Nun wird die BV der Pivotzeile gegen die NBV der Pivotspalte eingetauscht.
Hier: -1 ist das Pivotelement; in der Variablenspalte (vorderste Spalte) wird " y_4 " durch " x_1 " ersetzt.

5. Nun folgen die Schritte ?? und ?? des primalen Simplex-Algorithmus.
hier ergibt sich folgendes Simplextableau:

	x_1	x_2	y_1	y_2	y_3	y_4	rechte Seite
y_1	0	1	1	0	0	1	13
y_2	0	-1	0	1	0	3	19
y_3	0	1	0	0	1	0	7
x_1	1	1	0	0	0	-1	3
G	0	-1	0	0	0	-3	9

Sind noch weitere negative Werte auf der rechten Seite vorhanden, muss das Verfahren erneut durchlaufen werden. Sind alle Werte der rechten Seite positiv, haben wir eine zulässige Anfangslösung gefunden. Trotzdem können noch negative Werte in der Zielfunktionszeile stehen. Ab hier ist der primale Simplex-Algorithmus anzuwenden.

Bemerkung: Es gibt verschiedene Möglichkeiten, nach der Auswahl der Pivotzeile die Pivotspalte zu wählen (vergl. Schritt 3 oben). Jede führt zum gleichen Ziel. Bei ungeschickter Wahl sind mehr Optimierungsschritte, d.h. mehr Schritte in Phase 2 nötig. Die beste Möglichkeit hängt von der konkreten Aufgabe ab und kann daher nicht allgemein festgelegt werden.

Im Beispiel gibt es keine weiteren negativen Werte auf der rechten Seite mehr, wohl aber in der Zielfunktionszeile. Es geht mit dem primalen Simplex-Algorithmus weiter. Pivotspalte ist Spalte 6 (-3 in der Zielfunktionszeile). Kleinster Quotient ist $19/3$, also ist Zeile 2 die Pivotzeile und 3 das Pivotelement.

Neues Simplextableau nach vollständigem Austausch:

	x_1	x_2	y_1	y_2	y_3	y_4	rechte Seite
y_1	0	$4/3$	1	$-1/3$	0	0	$20/3$
y_4	0	$-1/3$	0	$1/3$	0	1	$19/3$
y_3	0	1	0	0	1	0	7
x_1	1	$2/3$	0	$1/3$	0	0	$28/3$
G	0	-2	0	1	0	0	28

Mit -2 in der Zielfunktionszeile ist noch nicht die optimale Lösung erreicht. Ein weiterer Austauschschritt mit dem Pivotelement $4/3$ in Zeile 1 und Spalte 2 führt auf folgendes

Simplextableau:

	x_1	x_2	y_1	y_2	y_3	y_4	rechte Seite
x_2	0	1	$3/4$	$-1/4$	0	0	5
y_4	0	0	$1/4$	$1/4$	0	1	8
y_3	0	0	$-3/4$	$1/4$	1	0	2
x_1	1	0	$-1/2$	$1/2$	0	0	6
G	0	0	$3/2$	$1/2$	0	0	38

Nun sind alle Werte in der Zielfunktionszeile nicht negativ. Es ist die optimale Lösung erreicht mit $x_1 = 6$ und $x_2 = 5$. Für die Zielfunktion gilt $G(6; 5) = 38$. Alle NBV, also Variablen, die nicht in der ganz links stehenden Variablenliste vorkommen, sind Null. Die Schlupfvariablen y_3 und y_4 haben im Fall der optimalen Lösung die Werte $y_3 = 2$ und $y_4 = 8$.

$y_3 = 2$ gibt an, wie viel von Farbe 2 mehr hergestellt werden dürfte, bis die Bedingung ($x_2 \leq 7$) verletzt wird.

$y_4 = 8$ sagt, dass 8 Einheiten Farbe weniger hergestellt werden dürften, bevor Bedingung $x_1 + x_2 \geq 3$ verletzt wird. Beachten Sie, dass es sich hier um eine \geq -Bedingung handelt und die Interpretation dementsprechend angepasst werden muss.

Zusammenfassung: Liegt ein Maximierungsproblem mit gemischten (\leq und \geq) Bedingungen vor, wird die Lösung in zwei Phasen ermittelt. In Phase 1 wird eine zulässige Ausgangslösung ermittelt, indem zuerst die negativen BV (negative rechte Seite) mit Hilfe des dualen Simplexalgorithmus ausgetauscht werden. Meist hat dann die Zielfunktionszeile noch negative Werte, so dass in Phase 2 der primale Simplexalgorithmus zur Optimierung der nun zulässigen Ausgangslösung eingesetzt wird.

2.3 Der allgemeine Fall

Eine noch allgemeinere Situation ergibt sich, wenn in den Bedingungen (zusätzlich) Gleichungen auftreten. Auch in diesem Fall wird ein Simplextableau erstellt, das für jede Gleichung eine Schlupfvariable enthält. Allerdings muss sicher gestellt werden, dass jede Schlupfvariable, die einer Gleichung hinzugefügt wurde, am Ende Null wird, um die Gleichung zu erfüllen. Da sämtliche NBV immer Null sind, wird dafür gesorgt, dass diese Schlupfvariablen zu NBV werden. Sie müssen ausgetauscht werden.

Beispiel: Aus 2 Futtermitteln soll eine Mischung hergestellt werden. Der Nutzen von Futtermittel A ist doppelt so groß, wie der von B. Aus ernährungsphysiologischen Gründen muss die Summe aus der 3-fachen Menge von Futtermittel A und der einfachen Menge von Futtermittel B mindestens 30ME ergeben. Die einzelnen Komponenten kosten 4GE/ME (Geldeinheiten pro Mengeneinheit) und 3GE/ME. Die Gesamtkosten dürfen 60GE nicht übersteigen. Den Futtermitteln ist ein Medikament zugesetzt, das in genauer Dosierung verabreicht werden muss. 20 ME von A oder 10 ME von B oder eine entsprechende Mischung aus beiden ergibt die richtige Menge.

Damit ergibt sich folgende mathematische Formulierung: Die Zielfunktion $G(x_1; x_2) = 2x_1 + x_2$ soll maximiert werden unter den Nebenbedingungen

$$3x_1 + x_2 \geq 30$$

$$4x_1 + 3x_2 \leq 60$$

$$x_1 + 2x_2 = 20$$

$$x_1, x_2 \geq 0$$

Das zugehörige Simplextableau ist:

	x_1	x_2	y_1	y_2	y_3	rechte Seite	
y_1	-3	-1	1	0	0	-30	
y_2	4	3	0	1	0	60	
y_3	1	2	0	0	1	20	*
G	-2	-1	0	0	0	0	

Algorithmus 3 Allgemeiner Simplexalgorithmus

1. Alle Schlupfvariablen (zu Beginn die BV) die zu einer *Gleichung* gehören, werden markiert. Dies markiert zugleich Zeilen des Ausgangstableaus.
Hier: Zeile 3 wird markiert, s. zusätzliche Spalte im Tableau.
2. Eine beliebige dieser markierten Zeilen wird Pivotzeile.
Hier: Zeile 3 wird Pivotzeile
3. Als Pivotspalte kann eine beliebige Spalte gewählt werden, für die gilt:
 - (a) Das damit festgelegte Pivotelement darf nicht Null sein.
 - (b) Die Spalte darf nicht markiert sein. Dies ist beim ersten Austauschschritt natürlich nie der Fall. In einem späteren Schritt könnte es jedoch passieren, dass eine bereits ausgetauschte Variable wieder zurückgetauscht wird, was durch diese Bedingung verhindert wird. Die Markierung wird in einer Markierungszeile für Spalten vorgenommen (s. unten).
Hier: z.B. für Spalte 1 ist die Bedingung (nicht markiert) erfüllt.
4. Das Pivotelement ist wie immer im Kreuzungspunkt von Pivotzeile und Pivotspalte.
Hier: 1 in Zeile 3 und Spalte 1 wird Pivotelement.
5. Nun folgen die Schritte ?? und ?? des primalen Simplex-Algorithmus, also der eigentliche Austausch.
Es ergibt sich folgendes Simplextableau:

	x_1	x_2	y_1	y_2	y_3	rechte Seite
y_1	0	5	1	0	3	30
y_2	0	-5	0	1	-4	-20
x_1	1	2	0	0	1	20
G	0	3	0	0	2	40
					*	

Falls mehrere Variablen markiert wurden (mehrere Gleichungen im Ausgangsproblem) werden die Schritte 2 - 5 wiederholt, bis alle markierten Variablen NBV sind.
Hier sind keine weiteren Variablen markiert.

Falls negative Werte auf der rechten Seite existieren, folgt nun der duale Simplex-Algorithmus.

Hier ist noch ein negativer Wert auf der rechten Seite. Die zweite Zeile ist die nächste Pivotzeile. Als Pivotspalte wird Spalte 2 gewählt, da sie auf ein negatives Pivotelement führt. Spalte 5 scheidet aus, da sie zur Variablen y_3 gehört und daher markiert ist. Das neue Simplextableau lautet:

	x_1	x_2	y_1	y_2	y_3	rechte Seite
y_1	0	0	1	1	-1	10
x_2	0	1	0	-0.2	0.8	4
x_1	1	0	0	0.4	-0.6	12
G	0	0	0	0.6	-0.4	28
					*	

Falls erforderlich erfolgt die weitere Optimierung über den primalen Simplex-Algorithmus.

Im allgemeinen Fall wird die Lösung also in drei Phasen ermittelt.

Hier ist eine weitere Optimierung nicht mehr möglich, da die einzige Spalte mit negativem Wert in der Zielfunktionszeile zu einer markierten Variablen gehört und damit nicht mehr ausgetauscht werden darf. Die Zielfunktion hat nun den Wert 28. Dies scheint eine Verschlechterung gegenüber dem vorherigen Schritt zu sein (dort war $G=40$, s.o.). Allerdings war die Lösung nicht zulässig (negativer Wert auf der rechten Seite) und damit der Zielfunktionswert sinnlos.

simplex/GeoGebra/Bild4.ggb.eps

Abbildung 3:

Abbildung 3 zeigt die Gerade a als untere Grenze, die Gerade b als obere Grenze und die Gerade c auf der die Lösung liegen muss. Sie repräsentiert die Gleichung in den Bedingungen. Die Gerade d entspricht der Zielfunktion mit einem beliebigen (hier zu großen) Zielfunktionswert. Sie muss parallel nach links verschoben werden, bis sie den gültigen Bereich berührt. Dies ist in Punkt A der Fall.

Interpretation: Von den Futtermitteln werden 12 bzw. 4 ME gebraucht.

$y_1 = 10$ heißt, dass 10 „Ernährungseinheiten“ zu viel verbraucht wurden.

y_2 – Spalte : Mit einer Einheit mehr an erlaubten Gesamtkosten (also 61), würde

der Nutzen um 0.6 Einheiten steigen.

y_3 – Spalte : Müsste eine Einheit an Medikamenten mehr verabreicht werden, würde der Nutzen um 0.4 Einheiten sinken (s. neg. Vorzeichen).

Zusammenfassung: Liegt ein gemischtes Maximierungsproblem mit \leq , $=$ und \geq Bedingungen vor, werden auch die Gleichungen mit einer Schlupfvariablen versehen und markiert. Die Lösung erfolgt nun in 3 Phasen. Phase 0 eliminiert die markierten Variablen aus der Basis. Dann folgen Phase 1 und Phase 2 wie in Kap.?? beschrieben.

2.4 Der verkürzte Simplexalgorithmus

Ist m die Anzahl der Bedingungen, so gibt es immer m Spalten, in denen eine 1 und sonst nur Nullen stehen. Jede dieser Spalten gehört zu einer Basisvariablen. Die 1 steht in der Zeile, die zur gleichen Basisvariablen gehört. Das Tableau enthält also redundante Informationen. In einer verkürzten Version werden diese Spalten weggelassen. Damit ergeben sich Änderungen. Die Schritte 1-3 sind identisch mit denen zum primalen Simplexalgorithmus und werden hier nur der Übersichtlichkeit halber wiederholt. Die Schritte 4 und 5 werden ergänzt. In Schritt 6 bezieht sich "alle anderen Elemente" nur noch auf alle Elemente, die weder in der Pivotzeile noch in der Pivotspalte stehen.

Als Beispiel dient das Beispiel aus Kap.???. Die einzelnen Schritte sind im Anschluss an den Algorithmus dargestellt.

Algorithmus 4 Verkürzter Simplexalgorithmus

1. In der letzten Zeile müssen negative Elemente vorkommen, sonst ist keine Verbesserung der Lösung mehr möglich. Man wählt das kleinste Element. Die zugehörige Spalte heißt "Pivotspalte".
Hier: erste Spalte mit -10 in der Zielfunktionszeile
2. Die Werte der rechten Seite müssen positiv sein. Man bildet alle Quotienten aus dem Wert der rechten Seite und dem Wert auf der Kreuzung dieser Zeile und der Pivotspalte. Negative Quotienten werden ignoriert.
Hier: $300/2=150$; $540/6=90$ und $320/4=80$
3. Der kleinste dieser (positiven) Quotienten bestimmt die "Pivotzeile".
Hier: kleinster Wert ist 80, die dritte Zeile ist die Pivotzeile.

4. Die Zahl, auf der Kreuzung von Pivotzeile und Pivotspalte, heißt "Pivotelement". Nun wird die BV der Pivotzeile durch die NBV der Pivotspalte ersetzt. Am Ende der Iteration (nach Schritt 6) wird diese dann neue BV sein.

Zusätzlich wird in der Variablenzeile ganz oben die NBV durch die BV ersetzt.

Hier: 4 ist Pivotelement; in der Variablenspalte wird " y_3 " durch " x_1 " ersetzt und in der Variablenzeile wird " x_1 " durch " y_3 " ersetzt.

5. Die Elemente der Pivotzeile, einschließlich der rechten Seite jedoch außer dem Pivotelement selbst, werden durch das Pivotelement geteilt.

Hier: $2 \rightarrow \frac{2}{4} = \frac{1}{2}$; $4 \rightarrow \frac{4}{4} = 1$ und $320 \rightarrow \frac{320}{4} = 80$

Die Elemente der Pivotspalte (außer dem Pivotelement selbst) werden durch das Pivotelement geteilt und mit -1 multipliziert.

Hier: $2 \rightarrow -\frac{2}{4} = -\frac{1}{2}$; $6 \rightarrow -\frac{6}{4} = -\frac{3}{2}$; $-10 \rightarrow \frac{10}{4}$

Das Pivotelement geht in seinen Kehrwert über.

Hier: $4 \rightarrow \frac{1}{4} = 0.25$

6. Alle anderen Elemente werden nach folgendem Schema berechnet:

die Elemente der Matrix seien a_{ij} ; die Pivotzeile sei Zeile k ; die Pivotspalte sei Spalte l

das Pivotelement ist also a_{kl} . Die Elemente der rechten Seite seien b_i . Dann gilt:

$$a_{ij} \rightarrow a_{ij} - \frac{a_{il} \cdot a_{kj}}{a_{kl}} \text{ und } b_i \rightarrow b_i - \frac{a_{il} \cdot b_k}{a_{kl}}$$

hier exemplarisch: Zeile 1 ; Spalte 3: $6 \rightarrow 6 - \frac{2 \cdot 4}{4} = 4$; Zeile 2, Spalte 2: $5 \rightarrow 5 - \frac{6 \cdot 2}{4} = 2$

Das Beispiel aus Kap. ?? wird dann zu (Pivotelemente gekennzeichnet):

	x_1	x_2	x_3	r. S.		y_3	x_2	x_3	r. S.		y_3	y_2	x_3	r. S.
y_1	2	1	6	300	y_1	-0.5	0	4	140	y_1	-0.5	0	4	140
y_2	6	5	2	540	y_2	-1.5	2	-4	60	x_2	-0.75	0.5	-2	30
y_3	4	2	4	320	x_1	0.25	0.5	1	80	x_1	0.625	-0.25	2	65
G	-10	-6	-4	0	G	2.5	-1	6	800	G	1.75	0.5	4	830

2.5 Sensitivitätsanalyse

Bei *parametrischer Optimierung* wird geprüft, wie stark sich die optimale Lösung ändert, wenn Veränderungen an den Ausgangsdaten vorgenommen werden. Wird immer nur eine Größe geändert, spricht man von *Sensitivitätsanalyse*. Die Veränderungen können sich auf die Koeffizienten der Zielfunktion, die rechten Seiten der Ungleichungen und die Koeffizienten der Ungleichungen beziehen.

Auf die Herleitung wird hier verzichtet. Die einzelnen Schritte sind zwar leicht nachvollziehbar. Die Fülle eines mehrseitigen Rechengangs wird aber schnell unübersichtlich. Eine gute Darstellung findet sich in [?].

Allgemein wird ein Koeffizient mit dem Wert c_k durch die Variable c ersetzt und geprüft, in welchem Intervall $[c_k - c_k^-; c_k + c_k^+]$ sich c befinden muss, so dass die Optimallösung für c_k zumindest qualitativ erhalten bleibt. Nicht nur die Herleitung, sondern auch die rechnerische Untersuchung ist langwierig und wird in dieser Vorlesung nicht explizit behandelt. Das Vorgehen ist in den Ergänzungen Kap.2 dargestellt.

Statt dessen wird hier der einfache Fall mit nur zwei Variablen betrachtet. Hier ist eine graphische Veranschaulichung möglich und die Überlegungen sind dementsprechend einfacher und übersichtlicher. Wir werden das Intervall $[c^-; c^+]$ bestimmen, in dem sich c bewegen darf.

Es wird das Beispiel aus Kap.?? verwendet:

$$\begin{aligned} x_1 + 2x_2 &\leq 16 \\ 3x_1 + 2x_2 &\leq 28 \\ x_2 &\leq 7 \\ x_1, x_2 &\geq 0 \end{aligned}$$

mit der geänderten Zielfunktion $G(x_1, x_2) = 3x_1 + x_2$. D.h. der Erlös der Farben ist anders als in Kapitel ??.

OR-Skript-Abb/Bild6.eps

Abbildung 4:

Der Optimalpunkt ist also D statt A wie in Kap.???. D.h. es wird nur Farbe 1 hergestellt.

2.5.1 Änderung der Zielfunktionskoeffizienten

Die Koeffizienten der Zielfunktion legen die Steigung der zugehörigen Geraden fest. Gesucht ist nun der Bereich $[c^-; c^+]$, in dem sich ein Zielfunktionskoeffizient c_k ändern darf, ohne dass sich die Optimallösung ändert. Es ist leicht einzusehen, dass der Punkt $D \left(\frac{28}{3} | 0 \right)$ so lange die optimale Lösung repräsentiert, als die Zielfunktion nicht flacher wird als die Gerade b mit Steigung $-\frac{3}{2}$. Erst wenn die Zielfunktion flacher wird, lohnt es sich, Farbe 2 herzustellen. Da D auf einer Koordinatenachse liegt, gibt es keine obere Grenze.

Hier wird der Fall untersucht, dass c als Vorfaktor von x_2 in der Zielfunktion vorkommt:

$$G = 3x_1 + cx_2$$

Für $c = c_k = 1$ ergibt sich also das obige Problem. Gesucht ist nun, in welchem Intervall c variieren darf, ohne den optimalen Punkt $D \left(\frac{28}{3} | 0 \right)$ zu ändern.

Die relevanten Geraden werden in Normalform umgeformt ($x_1 \rightarrow x$; $x_2 \rightarrow y$) :

$$\begin{aligned} \text{Zielfunktion : } y &= -\frac{3}{c}x + \frac{G}{c} \\ \text{2.Bed. (Gerade b) : } y &= -\frac{3}{2}x + 14 \end{aligned}$$

Nun muss gelten:

$$\begin{aligned} -\frac{3}{2} &\geq -\frac{3}{c} \\ \frac{2}{3} &\geq \frac{c}{3} \\ c &\leq 2 \end{aligned}$$

Die untere Grenze wird so bestimmt, dass die Steigung (pos.) Null ist. Also $c \geq -\infty$.
D.h.

$$-\infty \leq c \leq 2$$

2.5.2 Änderung eines Koeffizienten einer Bedingung

Hierdurch ändert sich Steigung einer Randbedingung. Der Fixpunkt um den sich die entsprechende Gerade dreht, liegt immer auf einer Koordinatenachse. Die Grenzen ergeben sich aus folgenden Möglichkeiten:

1. Der Optimalpunkt ist der Schnittpunkt der beweglichen Geraden mit einer anderen (festen) Geraden einer Randbedingung. Der Optimalpunkt wird mit der beweglichen Geraden entlang der festen Geraden verschoben bis einer der folgenden Fälle eintritt:
 - (a) Der verschobene Optimalpunkt erreicht einen anderen Schnittpunkt von Randbedingungen.
 - (b) Die Steigung der beweglichen Geraden erreicht den Wert der Steigung der Zielfunktion.
2. Der Optimalpunkt ist der Fixpunkt. Dann ergibt sich die Grenze durch Vergleich mit der Steigung der Zielfunktion.

1. **Beispiel:** Hier soll exemplarisch die zweite Bedingung in

$$cx_1 + 2x_2 \leq 28$$

verallgemeinert werden. Fixpunkt (0|14). Für $c=3$ ergibt sich obiges Problem:

OR-Skript-Abb/Bild252_1a.eps

Der Punkt D darf auf der x -Achse *verschoben* werden.

Die Gerade b wird zu:

$$y = -\frac{c}{2}x + 14$$

Wenn die Steigung (neg.) Null wird, rutscht der Punkt D auf der x -Achse ins Unendliche. Wenn er den Punkt F (Schnitt zwischen a und der x -Achse) überschreitet, liegt er außerhalb des zulässigen Bereichs und F ist der neue Optimalpunkt. Dies geschieht bei $c = \frac{7}{4}$, also

$$c \geq \frac{7}{4}$$

OR-Skript-Abb/Bild252_1b.eps

Wird die Gerade b aber steiler als die Zielfunktionsgerade, so wird der Punkt A zum Optimalpunkt. Es muss also gelten:

$$\begin{aligned} -\frac{c}{2} &\geq -3 \\ c &\leq 6 \end{aligned}$$

OR-Skript-Abb/Bild252_1c.eps

D.h.

$$\frac{7}{4} \leq c \leq 6$$

2. Beispiel: Das Optimierungsproblem

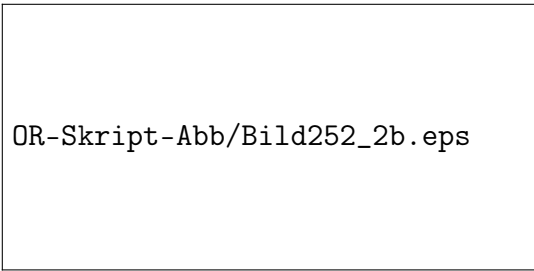
$$\begin{aligned} (I) \quad & x_1 + 3x_2 \leq 9 \\ (II) \quad & 2x_1 + c \cdot x_2 \leq 8 \\ (III) \quad & x_1; x_2 \geq 0 \\ & G = x_1 + x_2 \rightarrow \text{MAX} \end{aligned}$$

mit $c = 1$ hat die Lösung $x_1 = 3$ und $x_2 = 2$. Führen Sie folgende Sensitivitätsanalyse durch: bestimmen Sie, in welchem Bereich der Vorfaktor c in der Bedingung (II) variieren darf, so dass die gegebene Lösung die Optimallösung bleibt. Beachten Sie: die Lösung wird sich mit c *verschieben*. Gefragt ist, ab welchen Werten von c die Lösung nicht mehr durch den Schnitt der beiden Bedingungen (I) und (II) entsteht.

Fixpunkt (4|0)

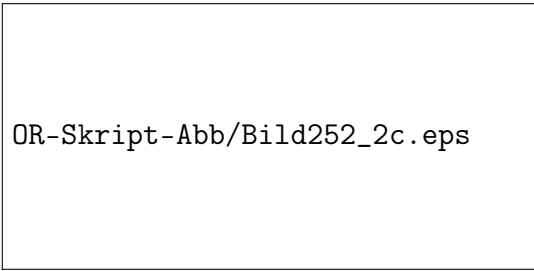
OR-Skript-Abb/Bild252_2a.eps

Grenzpunkt: $P(9|0) : 0 = -\frac{2}{c} \cdot 9 + \frac{8}{c} = -\frac{10}{c} \implies c \rightarrow -\infty$. Die Zeichnung zeigt den Fall für $c = -20$



OR-Skript-Abb/Bild252_2b.eps

Für $c=2$ erreicht die Grenzgerade die Steigung der Zielfunktion. Also ist $c=2$ die andere Grenze, da ab hier der Punkt $(4|0)$ der Optimalpunkt ist. Die Zeichnung zeigt den Fall für $c=2,67$. Hier ist die Begrenzungsgerade also schon zu flach und die Zielfunktionsgerade geht durch $(4|0)$.



OR-Skript-Abb/Bild252_2c.eps

Lösung: $-\infty \leq c \leq 2$.

2.5.3 Änderung der rechten Seiten

Die rechten Seiten geben typischerweise Maximalmengen oder -kosten an. Eine Änderung entspricht einer Parallelverschiebung der zugehörigen Begrenzung, in unserem Fall also von Geraden.

Wichtig: Eine solche Änderung *verschiebt* den optimalen Berührungspunkt möglicherweise. Gesucht ist der Bereich der Variation eines Wertes der rechten Seite, so dass immer noch der gleiche (evtl. jedoch verschobene) Punkt zu einem Maximum der Zielfunktion führt.

Erste alternative Formulierung: die Basisvariablen sollen die gleichen bleiben, ihr Wert darf sich aber ändern. Der Optimalwert der Zielfunktion ändert sich damit auch.

Zweite alternative Formulierung: der Optimalpunkt muss der Schnittpunkt der gleichen Geraden bleiben, selbst wenn sich dieser Schnittpunkt verschiebt. Von einer anderen Lösung sprechen wir erst, wenn der Optimalpunkt durch einen Schnitt zwischen anderen Geraden gegeben ist.

simplex/GeoGebra/Bild7a.ggb.eps

Abbildung 5:

Hier wird die Gerade b mit ursprünglich $c=28$ verschoben:

$$3x_1 + 2x_2 \leq c$$

Sie lautet in Normalform:

$$y = -\frac{3}{2}x + \frac{c}{2}$$

Mit c verschiebt sich die Gerade b parallel. b kann nach links bis zum Ursprung verschoben werden. D.h. $c \geq 0$.

Nach rechts kann die Gerade b verschoben werden, bis sie die x -Achse im gleichen Punkt schneidet, wie die Gerade a . Ab da ist der Schnittpunkt zwischen a und der x -Achse der Optimalpunkt. Dies ist bei $x=16$ der Fall. Für die Obergrenze von c muss also gelten:

$$\begin{aligned} 0 &= -\frac{3}{2} \cdot 16 + \frac{c}{2} \\ 24 &= \frac{c}{2} \end{aligned}$$

D.h. $c \leq 48$ oder $0 \leq c \leq 48$

2.6 Dualität

Ein Abschnitt zur Dualität ist in den Ergänzungen Kap.1. Hier soll nur kurz der Nutzen des dualen Problems umrissen werden.

Die Formulierung des dualen Problems folgt aus der Formulierung des ursprünglichen (primalen) Problems. Dies gilt auch für das Ergebnis, also die optimale Lösung.

Die optimalen Variablen des dualen Problems korrespondieren mit dem Knappheitsgrad des primalen Problems.

Die optimalen dualen Strukturvariablen geben den Schattenpreis der Bedingungen des primalen Problems an.

Die dualen Schlupfvariablen geben im (optimalen) Ergebnis die Opportunitätskosten an, d.h. den zusätzlichen Gewinn bei Aufnahme einer zusätzlichen Einheit. Sie sind den primalen Strukturvariablen zugeordnet.

Diese Informationen entnehmen wir jedoch schon dem Optimaltableau des primalen Problems (s. ??), so dass das Thema Dualität hier nicht weiter vertieft wird. Andererseits folgen diese Interpretationen des Optimaltableaus erst aus der Theorie der Dualität. Vergl. z.B. [?].

2.7 Zusammenfassung Simplex

Maximierungsproblem: Zielfunktion \Rightarrow MAX

1. \leq Bedingungen: primaler Simplexalgorithmus = Phase 2
 - (a) PS (Pivotspalte) aus dem kleinsten negativen ZFK (Koeffizienten der Zielfunktionszeile)
 - (b) PZ (Pivotzeile) aus dem kleinsten Quotienten > 0
 - (c) Austausch
 - (d) falls mindestens ein ZFK < 0 nochmal ab (a)
sonst fertig
2. (zusätzlich) \geq Bedingungen: dualer Simplexalgorithmus = Phase 1
 - (a) PZ aus dem kleinsten negativen Wert der RS (rechten Seite)
 - (b) PS beliebig jedoch mit Pivotelement < 0
 - (c) Austausch
 - (d) falls mindestens ein Wert der RS < 0 nochmal ab (a)
sonst: falls mindestens ein ZFK < 0 weiter ab 1.
sonst fertig
3. (zusätzlich) Gleichungen: entsprechende Zeilen markieren; Phase 0
 - (a) eine markierte Zeile ist PZ
 - (b) PS aus den nicht markierten Spalten wählen, so dass Pivotelement $\neq 0$
 - (c) Austausch und Spalte markieren
 - (d) falls weitere Zeilenmarkierungen existieren, weiter ab (a)
sonst: falls negative RS weiter ab 2.
sonst: falls mindestens ein ZFK < 0 weiter ab 1.
sonst fertig
4. Interpretation des Optimaltableaus
 - (a) BV = RS
 - (b) NBV = 0
 - (c) Optimalwert der Zielfunktion = RS
 - (d) Schattenvariablen als ZFK

Weitere Stichworte:

- verkürzter Simplexalgorithmus: ist effektiver, also gut zu wissen
- Dualität: ökonomische Interpretation; nützlich z.B. für Minimierungsprobleme; bei uns nur am Rand
- Sensitivitätsanalyse: in der Praxis sehr wichtig, von Hand eine langwierige Rechnung; bei uns: prinzipielles Verständnis auf grafischer Ebene wichtig
- Interpretation der Zielfunktionszeile und der rechten Seite im Optimaltableau

3 Transportprobleme

Zunächst soll die Problemstellung an Hand eines Beispiels dargestellt werden:

Eine Kaufhauskette versorgt vier Filialen F_j mit einem Gut aus drei Lagerstellen L_i . Die *gesamte* Lagermenge soll verteilt werden. Lagermengen und Bedarf entsprechen Tabelle 1, die Entfernungen (oder Kosten) Tabelle 2:

Lager	Lagermenge a_i		Filiale	Bedarfsmenge b_j
L_1	20		F_1	17
L_2	15		F_2	18
L_3	20		F_3	8
			F_4	12
Summe	55		Summe	55

Tabelle 1

	F_1	F_2	F_3	F_4
L_1	11	3	8	15
L_2	6	2	5	1
L_3	1	6	7	4

Tabelle 2

Der Bedarf soll bei minimalem Transportaufwand (gefahrte Kilometer oder Fahrtkosten) befriedigt werden.

Es handelt sich um ein Minimierungsproblem mit Nebenbedingungen und nicht negativen Variablen.

x_{ij} gibt an, wie viele ME von Lager L_i in die Filiale F_j geliefert werden. Damit gilt für die Zielfunktion:

$$G = 11x_{11} + 3x_{12} + 8x_{13} + 15x_{14} + 6x_{21} + 2x_{22} + 5x_{23} + x_{24} + x_{31} + 6x_{32} + 7x_{33} + 4x_{34} \rightarrow \min$$

Wir betrachten hier nur lineare Zielfunktionen. D.h. die Kosten wachsen linear mit der Menge an transportierten Gütern sowie dem Transportweg. Die Güter werden also einzeln geliefert bzw. berechnet.

Die Nebenbedingungen lauten:

$$x_{11} + x_{12} + x_{13} + x_{14} = 20 \quad (1)$$

$$x_{21} + x_{22} + x_{23} + x_{24} = 15 \quad (2)$$

$$x_{31} + x_{32} + x_{33} + x_{34} = 20 \quad (3)$$

$$x_{11} + x_{21} + x_{31} = 17 \quad (4)$$

$$x_{12} + x_{22} + x_{32} = 18 \quad (5)$$

$$x_{13} + x_{23} + x_{33} = 8 \quad (6)$$

$$x_{14} + x_{24} + x_{34} = 12 \quad (7)$$

$$x_{ij} \geq 0 \quad \forall i, j \quad (8)$$

(1)-(3) sind die Bedingungen für die Lagermengen. (4)-(7) sind die Bedingungen für die Bestellmengen. (8) gilt generell, da alle Mengen positiv sind.

Dieses lineare Optimierungsproblem könnte mit Hilfe des Simplexalgorithmus gelöst werden. Allerdings sind die Bedingungen (außer den Nichtnegativitätsbedingungen) immer in Form von Gleichungen gegeben. Dieser Umstand ermöglicht speziell angepasste Lösungsmethoden, die effektiver sind.

Die Ausgangsgrößen sind die gegebene Entfernungs- oder Kostenmatrix (c_{ij}) und die gesuchte Mengenmatrix (x_{ij}) :

c_{ij}	F_1	F_2	F_3	F_4
L_1	11	3	8	15
L_2	6	2	5	1
L_3	1	6	7	4

x_{ij}	F_1	F_2	F_3	F_4	
L_1	x_{11}	x_{12}	x_{13}	x_{14}	20
L_2	x_{21}	x_{22}	x_{23}	x_{24}	15
L_3	x_{31}	x_{32}	x_{33}	x_{34}	20
	17	18	8	12	55

Die allgemeine Formulierung eines Transportproblems lautet also:

$$G = \sum_{i=1}^m \sum_{j=1}^n c_{ij} x_{ij} \rightarrow \min \quad (9)$$

$$\sum_{j=1}^n x_{ij} = a_i; i \in [1; m] \quad (10)$$

$$\sum_{i=1}^m x_{ij} = b_j; j \in [1; n] \quad (11)$$

$$\sum_{i=1}^m a_i = \sum_{j=1}^n b_j \quad (12)$$

$$x_{ij} \geq 0 \quad (13)$$

Wie beim Simplexalgorithmus gibt es so viele Basisvariablen (BV), also x_{ij} , wie unabhängige Gleichungen. Dies sind die Gleichungen (10) und (11), also $m+n$ Gleichungen. Gleichung (12) ist keine unabhängige Gleichung sondern eine zusätzliche Information und reduziert die Zahl der BV auf $m+n-1$.

Gleichung (10) bedeutet, dass die aus einem Lager ausgelieferten Einzelmengen in der Summe die dort vorhandene Gesamtmenge ergeben. Wie erwähnt, soll die gesamte Lagermenge ausgeliefert werden. Diese Menge steht in der Tabelle für x_{ij} rechts.

Gleichung (11) bedeutet, dass die an eine Filiale gelieferten Mengen in der Summe die Bedarfsmenge ergeben. Diese Menge steht in der Tabelle für x_{ij} unten.

Da hier nicht von vornherein eine zulässige Lösung existiert, muss in einer ersten Stufe eine solche gefunden werden. In Stufe 2 wird diese in Bezug auf die Zielfunktion optimiert.

Bemerkung: Die Gleichung (12) - Forderung nach Gleichheit der Summe von Angebot und Nachfrage - stellt meist keine wirkliche Einschränkung dar. Ggf. muss ein Dummy-Anfrager ein Überangebot rechnerisch aufnehmen. Die Transportkosten werden Null gesetzt, so dass die Gesamttransportkosten nicht beeinflusst werden. Rein rechnerisch kann ein Angebotsmangel entsprechend behandelt werden. Dies bedeutet, dass Filialen mit den höchsten Transportkosten nachrangig behandelt und zunächst gar nicht beliefert werden. Im konkreten Fall werden evtl. weitere Überlegungen berücksichtigt werden müssen, wie z.B. vorrangige Behandlung von Groß- oder Stammkunden.

3.1 Ermittlung einer Ausgangslösung

Es muss eine Lösung für $m+n-1$ Variablen x_{ij} gefunden werden, wobei alle anderen Null sind. Diese Lösung muss die Gleichungen (10) und (11) erfüllen. Auf das Beispiel bezogen muss die Mengenmatrix (s.o.) so ausgefüllt werden, dass alle Zeilensummen und alle Spaltensummen erfüllt sind.

Hier werden zwei Methoden vorgestellt: die Northwest-Ecken-Regel und das Rangfolgeverfahren oder Matrixminimumverfahren.

3.1.1 Northwest-Ecken-Regel

Folgende Tabelle muss gefüllt werden

	F_1	F_2	F_3	F_4	
L_1					20
L_2					15
L_3					20
	17	18	8	12	55

Bei der Northwest-Ecken-Regel wird links oben (Nordwest-Ecke) begonnen und die größte Zahl eingetragen, so dass weder die Zeilen- noch die Spaltensumme überschritten wird, hier also $x_{11} = 17$. x_{11} ist damit Basisvariable. Die erste Spalte ist erledigt. Es wird in der ersten Zeile weiter gemacht. Mit $x_{12} = 3$ ist auch diese Zeile erfüllt:

	F_1	F_2	F_3	F_4	
L_1	17	3			20
L_2					15
L_3					20
	17	18	8	12	55

Die nächste Tabelle enthält eine 15 in der zweiten Spalte, so dass auch diese Summe erfüllt wird, ohne dass die Zeilensumme überschritten wurde. Tatsächlich ist diese (zufällig) genau erfüllt. Eine Basisvariable wird also Null. Z.B. $x_{23} = 0$. Dies darf trotz des Wertes Null nicht übergangen werden, da wir sonst eine BV zu wenig haben. Es geht weiter mit x_{33} . Mit einer 8 ist die Spaltensumme erfüllt. Dann fehlt nur noch ein letzter Wert (12) in der vierten Spalte, der auch diese Spaltensumme erfüllt. Aufgrund der Gleichung (4) geht diese Rechnung mit dem letzten Wert immer auf. Mit (gedachten) Nullen in den unausgefüllten Plätzen ist eine zulässige Anfangslösung gefunden:

	F_1	F_2	F_3	F_4	
L_1	17	3			20
L_2		15	0		15
L_3			8	12	20
	17	18	8	12	55

Entfernungsmatrix:

c_{ij}	F_1	F_2	F_3	F_4
L_1	11	3	8	15
L_2	6	2	5	1
L_3	1	6	7	4

Diese Lösung ist sicher nicht optimal, da die Entfernungsmatrix noch gar nicht berücksichtigt wurde. Für die Zielfunktion gilt: $G = 17 \cdot 11 + 3 \cdot 3 + 15 \cdot 2 + 8 \cdot 7 + 12 \cdot 4 = 330$. Die Summe der Transportwege beträgt 330 Längeneinheiten.

Bemerkung: Wird im ersten Schritt nicht, wie hier, die Spaltensumme erfüllt, sondern die Zeilensumme, muss im nächsten Schritt in der ersten Spalte und der zweiten Zeile weiter gemacht werden.

Zwei Verfahren zur Optimierung werden in Kap. ?? behandelt.

3.1.2 Rangfolgeverfahren

Eine bessere Anfangslösung bekommt man mit dem Rangfolgeverfahren, da es die Entfernungsmatrix (oder Kostenmatrix) berücksichtigt. Anstatt links oben zu beginnen und zeilen- bzw. spaltenweise fortzuschreiten, wird der Platz besetzt, auf dem der kleinste Wert der Entfernungsmatrix steht. Im Beispiel Zeile 3 Spalte 1 (mit Wert 1 in der Entfernungsmatrix):

	F_1	F_2	F_3	F_4	
L_1					20
L_2					15
L_3	17				20
	17	18	8	12	55

Entfernungsmatrix:

c_{ij}	F_1	F_2	F_3	F_4
L_1	11	3	8	15
L_2	6	2	5	1
L_3	1	6	7	4

Spalte 1 ist damit erfüllt und wird gestrichen. D.h. im weiteren Verlauf der Rechnung werden die Werte aus Spalte 1 nicht mehr berücksichtigt.

Weiter geht es mit Zeile 2 und Spalte 4 (Wert 1):

	F_1	F_2	F_3	F_4	
L_1					20
L_2				12	15
L_3	17				20
	17	18	8	12	55

Spalte 4 ist erfüllt und wird gestrichen.

Beachten Sie, dass die Reihenfolge in der die beiden bisherigen BV gefunden wurden, willkürlich war. Die entsprechenden Werte in der Entfernungsmatrix sind gleich. In unserem Beispiel ergibt sich dadurch kein Unterschied in der zu bestimmenden Ausgangslösung. In einem anderen Fall, wenn zwei gleiche Werte in der Entfernungsmatrix in der gleichen Zeile oder der gleichen Spalte stehen, könnte sich jedoch ein Unterschied ergeben. Das Rangfolgeverfahren gibt keine Regel an, wie in einem solchen Fall zu verfahren ist, d.h. die Wahl erfolgt zufällig.

Der nächstkleinere Wert (2) steht in Zeile 2 Spalte 2. Hier muss aufgrund der Zeilensumme eine 3 eingetragen werden.

	F_1	F_2	F_3	F_4	
L_1					20
L_2		3		12	15
L_3	17				20
	17	18	8	12	55

Damit wird Zeile 2 gestrichen.

Es folgen diese Schritte:

	F_1	F_2	F_3	F_4			F_1	F_2	F_3	F_4			F_1	F_2	F_3	F_4	
L_1		15			20	L_1		15			20	L_1		15	5		20
L_2		3		12	15	L_2		3		12	15	L_2		3		12	15
L_3	17				20	L_3	17		3		20	L_3	17		3		20
	17	18	8	12	55		17	18	8	12	55		17	18	8	12	55

Entfernungsmatrix (zur Berechnung der Zielfunktion):

c_{ij}	F_1	F_2	F_3	F_4
L_1	11	3	8	15
L_2	6	2	5	1
L_3	1	6	7	4

Nun sind $m+n-1=6$ (Basis-)Variablen besetzt und die Bedingungen sind alle erfüllt.

Für die Zielfunktion gilt: $G = 15 \cdot 3 + 5 \cdot 8 + 3 \cdot 2 + 12 \cdot 1 + 17 \cdot 1 + 3 \cdot 7 = 141$. Die Summe der Transportwege beträgt 141 Längeneinheiten, ist also viel besser als beim Nordwest-Ecken-Verfahren.

Bemerkung: Ein noch besseres Verfahren ist das Vogelsche Approximationsverfahren.

Es liefert meist schon sehr gute Anfangslösungen, ist aber sehr aufwändig und soll hier nicht besprochen werden. Eine gute Beschreibung mit durchgerechnetem Beispiel ist in [?].

3.2 Ermittlung der optimalen Lösung

Zur Ermittlung der optimalen Lösung werden zwei Methoden behandelt: die Stepping-Stone-Methode und die MODI-Methode.

3.2.1 Stepping-Stone-Methode

Die Optimierung geschieht durch eine Umverteilung der Transportmengen, was einem Austausch der BV entspricht. Dies erfolgt immer in einem rechtwinkligen Polygonzug. Das Vorgehen wird am obigen Beispiel gezeigt, wobei von der Anfangslösung nach Abschnitt ?? (Rangfolgeverfahren) ausgegangen wird.

Nun wird systematisch probiert, welche Änderungen vorteilhaft sind. Hierfür werden nacheinander alle NBV vorübergehend auf 1 gesetzt (die anderen NBV sind Null), was Änderungen an den BV nötig macht, so dass die Zeilen- und Spaltensummen erhalten bleiben.

Zunächst wird $x_{11} = 1$ gesetzt:

Tab.1	F_1	F_2	F_3	F_4	
L_1	0 → 1	15	5 → 4		20
L_2		3		12	15
L_3	17 → 16		3 → 4		20
	17	18	8	12	55

Entf.matrix:

c_{ij}	F_1	F_2	F_3	F_4
L_1	11	3	8	15
L_2	6	2	5	1
L_3	1	6	7	4

Im geschlossenen rechtwinkligen Polygonzug wird also abwechselnd der Wert 1 addiert und subtrahiert. Dadurch bleiben die Zeilen- und Spaltensummen erhalten. Beachten Sie, dass nur die NBV x_{11} geändert wurde, während alle anderen Änderungen an BV vorgenommen wurden. Dies ist ein Grundprinzip der Methode und muss generell eingehalten

werden. Nur dann ergibt sich immer ein eindeutiger rechtwinkliger Polygonzug in der Tabelle.

Die Zielfunktion ändert sich durch diese spezielle Änderung um $11-8+7-1=9$. D.h. der Weg wird um 9 Einheiten länger. Also muss diese Änderung verworfen werden.

Ist die Untersuchung für eine NBV abgeschlossen, werden alle Änderungen wieder zurück gesetzt. Erst dann kann die nächste NBV untersucht werden.

Für jeden freien Platz, der neu besetzt werden kann (d.h. jede NBV), existiert ein geeigneter Polygonzug, der nur im einfachsten Fall ein Rechteck ist. Die notwendigen Änderungen bei x_{34} sind z.B.:

Tab.2	F_1	F_2	F_3	F_4	
L_1		15 → 14	5 → 6		20
L_2		3 → 4		12 → 11	15
L_3	17		3 → 2	0 → 1	20
	17	18	8	12	55

Änderung der Zielfunktion: $4-7+8-3+2-1=3$, also nicht günstig.

Beachten Sie: Ein geschlossener Polygonzug kann sowohl leere als auch besetzte Felder geradlinig *überspringen*. Diese Felder werden nicht verändert. Für die Berechnung sind ausschließlich die Eckpunkte zu betrachten, also die Punkte in denen der Polygonzug rechtwinklig abknickt,

Die restlichen Möglichkeiten sind in folgenden Tabellen aufgeführt:

Tab.3	F_1	F_2	F_3	F_4	
L_1		15 → 16	5 → 4		20
L_2	0 → 1	3 → 2		12	15
L_3	17 → 16		3 → 4		20
	17	18	8	12	55

Tab.4	F_1	F_2	F_3	F_4	
L_1		15 → 14	5 → 6		20
L_2		3		12	15
L_3	17	0 → 1	3 → 2		20
	17	18	8	12	55

Tab.5	F_1	F_2	F_3	F_4	
L_1		15 → 16	5 → 4		20
L_2		3 → 2	0 → 1	12	15
L_3	17		3		20
	17	18	8	12	55

Tab.6	F_1	F_2	F_3	F_4	
L_1		15 → 14	5	0 → 1	20
L_2		3 → 4		12 → 11	15
L_3	17		3		20
	17	18	8	12	55

Nun müssen noch die Änderungen der Zielfunktionswerte verglichen werden.

1. $\Delta_{11} = 11 - 8 + 7 - 1 = 9$; ungünstig
2. $\Delta_{34} = 4 - 1 + 2 - 3 + 8 - 7 = 3$; ungünstig
3. $\Delta_{21} = 6 - 2 + 3 - 8 + 7 - 1 = 5$; ungünstig
4. $\Delta_{32} = 6 - 7 + 8 - 3 = 4$; ungünstig
5. $\Delta_{23} = 5 - 2 + 3 - 8 = -2$; günstig
6. $\Delta_{14} = 15 - 3 + 2 - 1 = 13$; ungünstig

In diesem Beispiel gibt es nur *einen* Basistausch, der zu einer Verbesserung, d.h. Verringerung des Gesamttransportwegs, führt. Falls mehrere günstige Ergebnisse vorliegen, wird die stärkste Veränderung (betragsgrößter negativer Wert) vorgenommen.

Da nun bekannt ist, dass x_{23} vergrößert werden muss, wird untersucht, welcher Wert maximal erlaubt ist. Die Grenze ergibt sich durch die Überlegung, dass keine Variable negativ werden darf. Im gegebenen Fall werden x_{22} und x_{13} um einen Betrag d verringert, während x_{23} und x_{12} jeweils um d vergrößert werden. Die maximal erlaubte Verringerung ist $d=3$, da x_{22} dann Null wird. Das Verfahren muss wie oben durchgeführt werden aber mit 3 statt 1. Es ergibt sich

	F_1	F_2	F_3	F_4				F_1	F_2	F_3	F_4	
L_1		15 → 18	5 → 2		20		L_1		18	2		20
L_2		3 → 0	0 → 3	12	15	=	L_2			3	12	15
L_3	17		3		20		L_3	17		3		20
	17	18	8	12	55			17	18	8	12	55

Mit $x_{22} = 0$ und $x_{23} = 3$ ist x_{23} neue BV und x_{22} wird zur NBV.

Entfernungsmatrix:	c_{ij}	F_1	F_2	F_3	F_4
	L_1	11	3	8	15
	L_2	6	2	5	1
	L_3	1	6	7	4

Jetzt muss die gesamte Berechnung für die neuen NBV (die nun leeren Stellen) mit den neuen Werten erneut durchgeführt werden. Tatsächlich ergeben sich nur positive Werte, so dass keine weitere Verbesserung mehr möglich ist. Das gegebene Tableau ist optimal. Die zu fahrende Strecke beträgt $G = 18 \cdot 3 + 2 \cdot 8 + 3 \cdot 5 + 12 \cdot 1 + 17 \cdot 1 + 3 \cdot 7 = 135$ Längeneinheiten, ist also besser als die Ausgangslösung.

Beachten Sie: In der neuen Iteration müssen *alle* NBV neu berechnet werden, also auch die, die zuvor schon berechnet worden waren.

Algorithmus 5 Stepping-Stone

1. Zulässige Ausgangslösung (Basislösung) bestimmen, s.Kap. ??
2. Ein freies Feld wird mit 1 besetzt
3. Korrekturen entlang einem rechtwinkligen Entfernungs- oder Kostenmatrixpolygonzug, so dass alle Bedingungen erfüllt bleiben
4. Berechnung der Entfernungs- oder Kostenänderung Δ_{ij}
5. Schritte 2-4 für alle freien Felder
6. Sind alle $\Delta_{ij} \geq 0$, liegt eine optimale Basislösung vor, sonst weiter mit Schritt 7
7. Es liegt mindestens ein negatives Δ_{ij} vor. Gibt es mehrere, so wird das betragsgrößte davon gewählt
8. Die Transportmatrix wird entlang dem in Schritt 3 verwendeten Polygonzug so korrigiert, dass eine der Basisvariablen Null wird, aber keine kleiner als Null. Dazu muss untersucht werden, welche Werte entlang des Polygonzuges verkleinert werden. Der kleinste dieser Werte sei d. Die Berechnung entspricht nun der von Schritt 3 mit d anstatt mit 1.
Die Null gewordene ehemalige BV ist nun NBV. Die von Null auf d angewachsene ehemalige NBV ist neue BV.
Im Fall von Entartung können mehrere BV zugleich Null werden. Dann verlässt nur eine davon die Basis. Die anderen bleiben in der Basis, jedoch mit Wert 0.

Die Schritte 2-8 entsprechen einer Iteration. Sie müssen so lange wiederholt werden, bis in Schritt 6 die Abbruchbedingung erfüllt ist.

Übung: 1) Berechnen Sie die Δ_{ij} und weisen Sie nach, dass das Tableau die optimale Transportverteilung darstellt.
2) Optimieren Sie die Anfangslösung von Kap. ?? (Nordwest-Ecken-Regel).

3.2.2 MODI-Methode

Die MODI-Methode (modifizierte Distributions Methode) unterscheidet sich vom Stepping-Stone Verfahren hauptsächlich in der Bestimmung der neuen Basisvariablen. Die Umverteilung entlang eines rechtwinkligen Polygonzuges bleibt erhalten.

Das Verfahren erfordert einen geringeren Rechenaufwand, ist aber mathematisch schwerer zu verstehen. Ohne Beweis wird von der Tatsache ausgegangen, dass die Werte der Entfernungsmatrix oder Kostenmatrix c_{ij} an den Stellen der Basisvariablen folgender Gleichung genügen: $c_{ij} = u_i + v_j$. u_i und v_j sind *Dualvariablen*. Es wird wieder das Beispiel von oben verwendet mit der Anfangslösung nach Abschnitt ?? (NWE-Regel).

Die folgenden Tabellen zeigen die Basislösung aus Abschnitt ??, die gesamte Entfernungsmatrix c_{ij} und die Matrix z_{ij} , die an den Stellen der Basisvariablen die Werte der *Entfernungsmatrix* c_{ij} enthält:

Basisv.	F_1	F_2	F_3	F_4	c_{ij}	F_1	F_2	F_3	F_4	z_{ij}	v_1	v_2	v_3	v_4
L_1	17	3			L_1	11	3	8	15	u_1	11	3		
L_2		15	0		L_2	6	2	5	1	u_2		2	5	
L_3			8	12	L_3	1	6	7	4	u_3			7	4

Nun müssen die Unbekannten u_i und v_j nach der Bedingung $c_{ij} = u_i + v_j$ bestimmt werden. Es gibt immer $m+n$ solcher Variablen, jedoch nur $m+n-1$ Gleichungen zu ihrer Bestimmung (entsprechend der $m+n-1$ Basisvariablen). Ein Wert kann also beliebig gewählt werden, z.B. $v_1 = 0$. Damit lassen sich die weiteren Größen sukzessiv bestimmen:

z_{ij}	0	-8	-5	-8
11	11	3		
10		2	5	
12			7	4

Die Bedingungen zur Bestimmung dieser Werte können auch als ein lineares Gleichungssystem geschrieben werden, das mit den bekannten Methoden gelöst wird.

Nun werden die freien Stellen berechnet.

Es muss gelten: $z_{ij} = u_i + v_j$:

z_{ij}	0	-8	-5	-8
11	11	3	6	3
10	10	2	5	2
12	12	4	7	4

Nun wird die Differenzmatrix $d_{ij} = c_{ij} - z_{ij}$ gebildet:

d_{ij}				
	0	0	2	12
	-4	0	0	-1
	-11	2	0	0

d_{ij} enthält an den Stellen der BV also immer Nullen.

Gäbe es in der Differenzmatrix keine negativen Werte, wäre die Anfangslösung optimal gewesen. Von den negativen Werten wird der betragsgrößte gewählt: $d_{31} = -11$. Diese Stelle legt die neue Basisvariable fest. D.h. x_{31} wird neue Basisvariable. Die dadurch notwendigen Korrekturen werden wieder entlang einem rechtwinkligen Polygonzug über die BV gebildet. Dieser Schritt entspricht Schritt 8 der Stepping-Stone-Methode.

Transportm.	F_1	F_2	F_3	F_4
L_1	17 → 9	3 → 11		
L_2		15 → 7	0 → 8	
L_3	0 → 8		8 → 0	12

=

Transportm.	F_1	F_2	F_3	F_4
L_1	9	11		
L_2		7	8	
L_3	8			12

x_{31} ist neue BV statt x_{33} .

Das beschriebene Verfahren wird wiederholt, bis in der Differenzmatrix nur noch positive Werte stehen: (T.m.=Transportmatrix)

z_{ij}	0	-8	-5	3
11	11	3	6	14
10	10	2	5	13
1	1	-7	-4	4

d_{ij}				
	0	0	2	1
	-4	0	0	-12
	0	13	11	0

T.m.	F_1	F_2	F_3	F_4
L_1	2	18		
L_2			8	7
L_3	15			5

z_{ij}	0	-8	7	3
11	11	3	18	14
-2	-2	-10	5	1
1	1	-7	8	4

d_{ij}				
	0	0	-10	1
	8	12	0	0
	0	13	-1	0

T.m.	F_1	F_2	F_3	F_4
L_1		18	2	
L_2			6	9
L_3	17			3

z_{ij}	0	2	7	3
1	1	3	8	4
-2	-2	0	5	1
1	1	3	8	4

d_{ij}				
	10	0	0	11
	8	2	0	0
	0	3	-1	0

T.m.	F_1	F_2	F_3	F_4
L_1		18	2	
L_2			3	12
L_3	17		3	

z_{ij}	0	1	6	2
2	2	3	8	4
-1	-1	0	5	1
1	1	2	7	3

d_{ij}				
	9	0	0	11
	7	2	0	0
	0	4	0	1

Nun sind alle Werte in der Differenzmatrix d_{ij} nicht negativ. Die letzte Transportmatrix war also optimal. Ein Vergleich mit dem Ergebnis aus Abschnitt ?? zeigt, dass sich erwartungsgemäß die gleiche Matrix ergeben hat.

Die BV sind x_{12} , x_{13} , x_{23} , x_{24} , x_{31} und x_{33} .

Algorithmus 6 MODI

1. Zulässige Ausgangslösung (Basislösung) bestimmen, s.Kap. ??
2. Es wird eine neue Matrix z_{ij} erstellt, die an den Stellen der Basisvariablen die Werte der gegebenen Entfernungs- oder Kostenmatrix c_{ij} enthalten. Die anderen Stellen bleiben leer.

Achtung: Nicht die Werte der Transportmatrix verwenden!
3. Bestimmung der Dualvariablen u_i und v_j mit $c_{ij} = u_i + v_j$. Eine Variable ist frei wählbar, bei uns immer $v_1 = 0$.
4. Berechnung der noch unbestimmten Größen z_{ij} , so dass $z_{ij} = u_i + v_j$.
5. Berechnung der Differenzmatrix $d_{ij} = c_{ij} - z_{ij}$.
6. Wenn alle $d_{ij} \geq 0$ sind, ist die letzte Transportmatrix x_{ij} optimal. Sonst weiter mit Schritt 7.
7. Von allen $d_{ij} < 0$ wird der betragsgrößte Wert gewählt. Diese Stelle bestimmt die neue Basisvariable x_{ij} .
8. Dieser Schritt entspricht Schritt 8 der Stepping-Stone-Methode, wodurch die neue, verbesserte Transportmatrix bestimmt wird.

Die Schritte 2-8 entsprechen einer Iteration. Sie müssen so lange wiederholt werden, bis in Schritt 6 die Abbruchbedingung erfüllt ist.

Übung: Optimieren Sie die Anfangslösung von Kap. ?? (Rangfolgeverfahren).

3.3 Das lineare Zuordnungsproblem

Das lineare Zuordnungsproblem ist ein Spezialfall der Transportprobleme. Anstatt z.B. 20 Güter von 3 Anbietern auf 5 Nachfrager zu verteilen, werden die 20 Güter von je einem (also insgesamt 20) Anbietern auf 20 Nachfrager verteilt. Die Frage ist jetzt nur noch, wer wen beliefert. Die Aufgabe ist in dieser Form meist unrealistisch. Mathematisch identisch ist aber das Problem, von n Arbeitern n verschiedene Aufgaben zu unterschiedlichen Kosten ausführen zu lassen. Die Kosten hängen von der Kombination Arbeiter/Tätigkeit ab. Wird eine bestimmte Tätigkeit X von Arbeiter A ausgeführt, können die Kosten höher sein, als wenn die gleiche Tätigkeit von Arbeiter B erledigt wird. Andererseits könnte die Situation auf Tätigkeit Y bezogen genau umgekehrt sein, was offensichtlich auf ein Optimierungsproblem führt.

Die allgemeine Formulierung des linearen Zuordnungsproblems lautet:

$$G = \sum_{i=1}^m \sum_{j=1}^n c_{ij} x_{ij} \rightarrow \min$$

$$\sum_{j=1}^n x_{ij} = 1; \forall i$$

$$\sum_{i=1}^m x_{ij} = 1; \forall j$$

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

Dies lässt sich mit den oben behandelten Methoden lösen.

Speziell angepasst ist die *Ungarische Methode* (s.[?] und [?]), die hier nicht besprochen werden soll.

Beispiel: Das folgende Zuordnungsproblem soll gelöst werden. Die Ausgangslösung wird mit dem Rangfolgeverfahren bestimmt und mit der MODI-Methode optimiert. Drei Baustellenkräne wechseln ihren Einsatzort. Die Entfernungen zwischen den momentanen Standorten S_i und den Zielorten Z_j entsprechen folgender Tabelle:

Entfernungsmatrix	Zielort: Z_1	Z_2	Z_3
Standort: S_1	10	5	3
S_2	5	6	3
S_3	1	7	3

Welche Verteilung minimiert den gesamten Transportweg?

Rangfolgeverfahren und MODI-Methode:

Die Einsen (der tatsächlich stattfindende Transport entsprechend der Ausgangslösung) ergeben sich nach dem Rangfolgeverfahren.

Allerdings haben wir dann noch nicht genug Basisvariablen. Es müssen noch zwei weitere BV gewählt werden. Diese werden durch Nullen in der Tabelle gekennzeichnet. Sie tragen also nicht zum Transport bei. Trotzdem kann sich eine ungeschickte Wahl auf die weiteren Berechnungen entscheidend auswirken. Es können sich insbesondere zwei Probleme ergeben. Dies wird an diesem Beispiel gezeigt. Eine Systematik einer besseren Wahl wird im folgenden Abschnitt (??) betrachtet.

Erstes Problem:

T.m.	Z_1	Z_2	Z_3
S_1		0	1
S_2		1	0
S_3	1		

z	0		
		5	3
		6	3
1	1		

Hier enden die Möglichkeiten, da die Basisvariablen so ungeschickt gewählt wurden, dass eine BV sowohl in ihrer Zeile als auch in ihrer Spalte alleine steht. Die Bestimmung der Dualvariablen bricht in einem solchen Fall vorzeitig ab. Dies muss vermieden werden.

Zweites Problem:

Die folgende Wahl vermeidet das erste Problem, ist jedoch immer noch nicht optimal, da hier bei ungeschickter Fortführung der Iteration eine Endlosschleife entstehen könnte:

T.m.	Z_1	Z_2	Z_3
S_1	0	*	1
S_2		1	
S_3	1	0	

z	0	6	-7
10	10	16	3
0	0	6	-7
1	1	7	-6

d			
	0	-11	0
	5	0	10
	0	0	9

Mit * ist die Stelle in der Transportmatrix gekennzeichnet, die zur neuen Basisvariablen gehört. Diese Stelle konnte erst nach der Berechnung der Differenzmatrix d bestimmt werden.

T.m.	Z_1	Z_2	Z_3
S_1		0	1
S_2		1	
S_3	1	0	*

z	0	6	4
-1	-1	5	3
0	0	6	4
1	1	7	5

d			
	11	0	0
	5	0	-1
	0	0	-2

In diesem Schritt gab es keine Veränderung der Zuordnung. Nur die Basisvariablen wurden getauscht.

T.m.	Z_1	Z_2	Z_3
S_1		*	1
S_2		1	
S_3	1	0	0

z	0	6	2
1	-1	6	3
0	0	6	2
1	1	7	3

d			
	9	-1	0
	5	0	1
	0	0	0

Auch hier wurden nur Basisvariablen getauscht. Eine mögliche neue Transportmatrix ist:

T.m.	Z_1	Z_2	Z_3
S_1		0	1
S_2		1	*
S_3	1	<u>0</u>	-

(wie oben; Endlosschleife möglich)

Damit liegt wieder die Situation von zwei Zeilen weiter oben vor und es könnte eine *Endlosschleife* entstehen. Die Wahl der Basisvariablen ist jedoch teilweise willkürlich. Anstatt der unterstrichenen Null könnte die mit „-“ gekennzeichnete Stelle zur neuen BV werden:

T.m.	Z_1	Z_2	Z_3
S_1		0	1
S_2		1	*
S_3	1		0

z	0	4	2
1	1	5	3
2	2	6	4
1	1	5	3

d			
	9	0	0
	3	0	-1
	0	2	0

T.m.	Z_1	Z_2	Z_3
S_1		1	
S_2		0	1
S_3	1		0

z	0	5	2
0	0	5	2
1	1	6	3
1	1	6	3

d			
	10	0	1
	4	0	0
	0	1	0

Diese Wahl hat zu einer Veränderung der Zuordnung geführt. Alle Werte in der Differenzmatrix sind nicht negativ. Die letzte Transportmatrix stellt die optimale Zuordnung dar.

Mehrdeutigkeiten ergeben sich durch die nicht eindeutige Wahl der Basisvariablen. Bei jedem linearen Zuordnungsproblem ergibt sich diese Mehrdeutigkeit schon bei der Anfangslösung. Hier müssen BV gewählt werden, die nicht zum Transport beitragen, also mit Null belegt werden. Man spricht von *Entartung*.

3.4 Entartung

Entartung (oder *Degeneration*) tritt auf, wenn bei der Bestimmung der Anfangslösung oder bei einem Optimierungsschritt die Wahl der BV nicht eindeutig ist.

Bei einem Optimierungsschritt ist dies der Fall, wenn zwei (oder mehr) Basisvariablen zugleich Null werden. Damit ist nicht klar, welche zur NBV wird. In einem solchen nicht-eindeutigen Fall kann bei ungünstiger Wahl sogar eine Endlosschleife auftreten (s. o.).

Zunächst ist offensichtlich, dass in einem solchen Fall (mindestens) eine Variable, die Null wurde, in der Basis verbleibt. Die Wahl muss immer so erfolgen, dass folgende Bedingungen erfüllt sind:

1. keine der BV darf sowohl in der Zeile als auch in der Spalte alleine stehen (siehe erstes Problem im Beispiel von Abschnitt ??)
2. die BV müssen so verteilt sein, dass kein geschlossener rechtwinkliger Polygonzug über sie gelegt werden kann.

Damit ist die Situation oft noch nicht eindeutig. Meist hilft jedoch folgende

Regel: Stehen zwei Variablen zur Wahl, von denen eine in der Basis verbleiben muss, so wird die Variable verwendet, an deren Stelle in der Kosten- oder Entfernungsmatrix der kleinere Wert steht. Dieses Verfahren erinnert an die Belegung der Ausgangsmatrix nach dem Rangfolge-Verfahren. In der neuen Transportmatrix wird diese Stelle mit Null belegt. Es findet also auf der zugehörigen Strecke kein Transport statt. Die Variable wird jedoch als BV mitgeführt, so dass im nächsten Schritt die Zwischenmatrix eindeutig bestimmt werden kann.

Besonders bei linearen Zuordnungsproblemen tritt der Fall der Entartung generell auf, da die Transportmatrix hier nur die Werte Null und Eins annehmen kann.

Beispiel: Diese Regeln werden nun auf das Problem von ?? angewendet.

Entfernungsmatrix:

	Z_1	Z_2	Z_3
S_1	10	5	3
S_2	5	6	3
S_3	1	7	3

Zunächst werde drei Einsen entsprechend ihrer Rangfolge eingetragen. Anschließend werden die zwei noch nötigen Nullen auch nach Rangfolge eingetragen, allerdings unter Berücksichtigung der oben genannten Bedingungen. Diese ändern in unserem Beispiel hier allerdings nichts.

T.m.	Z_1	Z_2	Z_3	z	0	5	2	d			
S_1		*	1	1	1	6	3		9	-1	0
S_2		1	0	1	1	6	3		4	0	0
S_3	1		0	1	1	6	3		0	1	0

Mit * ist die Stelle in der Transportmatrix gekennzeichnet, die zur neuen BV gehört. Diese Stelle konnte erst nach der Berechnung der Differenzmatrix d bestimmt werden.

T.m.	Z_1	Z_2	Z_3	z	0	4	2	d			
S_1		1	<u>0</u>	1	1	5	3		9	0	0
S_2		-	1	1	1	5	3		4	1	0
S_3	1		0	1	1	5	3		0	2	0

Hier wurde die mit - gekennzeichnete Stelle aus der Basis entfernt und die unterstrichene 0 als BV beibehalten. Dies ergibt sich aus einem Vergleich mit den entsprechenden Werten der Entfernungsmatrix (3 und 6). Alle Werte in der Differenzmatrix sind nun nicht negativ. Die letzte Transportmatrix stellt die optimale Zuordnung dar.

Dieses Ergebnis entspricht erwartungsgemäß dem Ergebnis aus ??, wurde aber in viel weniger Schritten erhalten. Im Fall dieses Beispiels waren unter Berücksichtigung der Regeln dieses Kapitels alle Schritte eindeutig.

3.5 Transport Zusammenfassung

Minimierungsproblem mit ausschließlich Gleichungen als Bedingungen.
Das Gleichungssystem ist unterbestimmt.

1. Ausgangslösung bestimmen
 - (a) Northwest-Ecken-Regel
 - (b) Rangfolgeverfahren
 - (c) (Vogelsches Approximationsverfahren)

Es müssen für die Transportmatrix Werte bestimmt werden, die sowohl die Zeilen- als auch die Spaltensumme erfüllen. Damit sind die vorläufigen BV festgelegt.

2. Optimierung der Lösung
 - (a) Stepping-Stone-Methode

Umverteilung entlang eines rechtwinkligen Polygonzuges, so dass *eine* Nicht-basisvariable zunächst den Wert 1 erhält und die entsprechenden BV entlang des Polygonzuges abwechselnd um 1 verringert und erhöht werden. Dies für alle NBV. Wenn sich Verbesserungen ergeben, d.h. der Zielfunktionswert verringert wird, wird die maximale Verbesserung angewendet, was einen Basiswechsel bedeutet.

Mit den neuen NBV von vorne, bis sich keine Verbesserung mehr ergibt.

(b) MODI-Methode

Die NBV, die die maximale Verbesserung bringt, wird anders bestimmt als bei (a):

i. Zwischenmatrix z_{ij} bestimmen

A. An den Stellen der BV stehen die gegebenen Werte der Entfernungsmatrix c_{ij} .

B. Es werden die Dualvariablen u_i und v_j so bestimmt, dass an diesen Stellen immer $u_i + v_j = z_{ij}$ gilt. Dabei gibt es immer eine Bedingung zu wenig, so dass eine der Dualvariablen frei gewählt werden kann, bei uns immer $v_1 = 0$.

C. Die nicht besetzten Stellen der Zwischenmatrix z_{ij} (also die Stellen der NBV) werden gefüllt, so dass auch dort gilt: $u_i + v_j = z_{ij}$.

ii. Differenzmatrix d_{ij} bilden: $d_{ij} = c_{ij} - z_{ij}$

An den Stellen der BV müssen also immer Nullen stehen.

Gibt es keine negativen Werte mehr, ist die letzte Transportmatrix optimal, sonst weiter.

iii. Die Stelle mit dem kleinsten Wert (also dem betragsgrößten negativen Wert) bestimmt die NBV, die in die Basis kommt. Diese wird entlang eines rechtwinkligen Polygonzuges maximiert, so dass keine BV kleiner als Null wird.

Mit den neuen BV von vorne, bis sich keine Verbesserung mehr ergibt.

Ein Spezialfall davon ist das *lineare Zuordnungsproblem*. Es entspricht formal dem Transportproblem, wobei jede Zeilen- und jede Spaltensumme immer genau 1 ergeben muss. Besonders einfach ist eine Ausgangslösung, die in der Hauptdiagonalen Einsen enthält. Um die Anzahl der BV zu erreichen, kann rechts neben jedes Diagonalelement eine Null geschrieben werden (bis auf das letzte Element).

Vorteilhaft ist jedoch eine Belegung, die die Transportmatrix berücksichtigt. Bei der Bestimmung der Ausgangslösung entspricht dies dem Rangfolgeverfahren.

Die Optimierung kann mit einem der oben beschriebenen Verfahren erfolgen.

4 Ganzzahlige und kombinatorische lineare Optimierung

Ganzzahlige lineare Probleme unterscheiden sich nicht grundsätzlich von allgemeinen linearen Optimierungsproblemen, die mit dem Simplexalgorithmus gelöst werden können. Da die Lösung jedoch ganzzahlig sein muss, liefert Simplex meist eine unzulässige Lösung. Man könnte meinen, dass es ausreicht, „benachbarte“ ganzzahlige Lösungen auf Zulässigkeit und Optimalität zu überprüfen. Dies gilt jedoch nur in einem sehr weiten und insbesondere nicht eindeutig festgelegten Sinn von „benachbart“. Sinnvollerweise wird statt dessen ein systematisches Verfahren angewendet, das zunächst auch nicht-ganzzahlige Lösungen findet, diese aber ganzzahlig optimiert, das „Branch-and-Bound-Verfahren“.

Kombinatorische Optimierung ist auch eine Form von ganzzahliger Optimierung, wobei hier typischerweise nicht auf den Simplexalgorithmus zurück gegriffen wird. Es handelt sich z.B. um Zuordnungsprobleme (s.??), Reihenfolgeprobleme (z.B. Traveling-Salesman) oder Auswahlprobleme (z.B. Rucksack-Problem=Knapsack-Problem). Auch sie werden oft mit Hilfe des Branch-and-Bound-Verfahrens gelöst. In speziellen Fällen liegt die Anpassung der Lösungsmethode an die Aufgabenstellung evtl. nicht auf der Hand.

4.1 Branch-and-Bound-Verfahren

Die Problemstellung ist zunächst die eines allgemeinen linearen Optimierungsproblems (s. ??) und der zusätzlichen Bedingung

$$x_i \in \mathbb{N}_0$$

Algorithmus 7 Branch & Bound

1. Berechnen der optimalen nicht-ganzzahligen Lösungen x_i mit Simplex.
 2. Falls sich mehrere nicht-ganzzahlige Lösungsvariablen ergeben, entscheidet man sich für eine von ihnen mit Index i und setzt $d = x_i$
Verzweigen (branching) in zwei Teilprobleme A und B.
Teilproblem A hat die zusätzliche Bedingung $x_i \leq n_i$, wobei n_i die größte ganze Zahl mit $n_i < d$ ist.
Teilproblem B hat die zusätzliche Bedingung $x_i \geq n_i + 1$; $n_i + 1$ ist die kleinste ganze Zahl $> d$.
Die Teilprobleme A und B haben also für x_i eine obere bzw. untere Schranke (bound).
 3. Lösen der Teilprobleme. Daraus erhält man jeweils eine obere Schranke \overline{G}_A bzw. \overline{G}_B für den Wert der Zielfunktion.
 4. Endgültige Bewertung (ausloten) des Zweiges. D.h. das gesamte Verfahren wird auf jeden Teilzweig angewendet.
-

Beispiel: Das Beispiel aus Kapitel ?? wird in einer Bedingung verändert, so dass der (optimale) Punkt A nicht mehr ganzzahlige Koordinaten hat.

$$\begin{aligned}
 x_1 + 2x_2 &\leq 16 \\
 3x_1 + 2x_2 &\leq 27 \\
 x_2 &\leq 7 \\
 x_1, x_2 &\geq 0 \\
 x_i &\in \mathbb{N}
 \end{aligned}$$

Die Zielfunktion bleibt gleich $G(x_1, x_2) = 3x_1 + 4x_2$. Die Ganzzahligkeit ergibt sich oft, wenn nicht in beliebigen Mengen, sondern in Stückzahlen oder Verpackungsmengen gerechnet wird.

simplex/GeoGebra/Bild9a1.ggb.eps

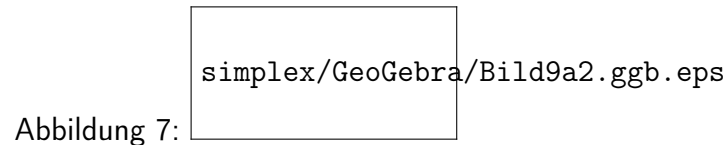
Abbildung 6:

A ist nach wie vor der optimale Punkt im Fall nicht-ganzzahliger Lösungen:

$A(5.5|5.25)$ und $G = 37.5$. Zulässig sind jedoch nur ganzzahlige Lösungen, hier also die Kreuzungspunkte des Koordinatengitters.

Es wird verzweigt in $x_1 \leq 5$ und $x_1 \geq 6$. Die Verzweigung könnte auch mit x_2 vorgenommen werden.

Es ergibt sich eine neue Abbildung mit zwei getrennten zulässigen Bereichen, die zwei voneinander unabhängigen Simplexproblemen entsprechen:

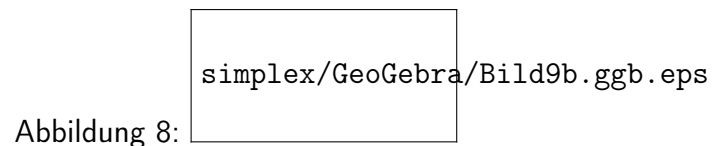


Beides wird mit Simplex gelöst und liefert:

1. $x_1 \leq 5$: $x_1 = 5$ und $x_2 = 5.5$; $G = 37$; nicht zulässig, also Kandidat für weitere Verzweigung
2. $x_1 \geq 6$: $x_1 = 6$ und $x_2 = 4.5$; $G = 36$; nicht zulässig, also Kandidat für weitere Verzweigung

Zunächst scheint es gleichgültig zu sein, ob 1. oder 2. zuerst verzweigt wird.

1. hat jedoch das größere G und ist damit der aussichtsreichere Kandidat für eine optimale Lösung. Wenn eine zulässige Lösung mit $G \geq 36$ gefunden wird, muss 2. nicht mehr verzweigt werden. 1. wird verzweigt in $x_2 \leq 5$ und $x_2 \geq 6$. Die Teilung ergibt:



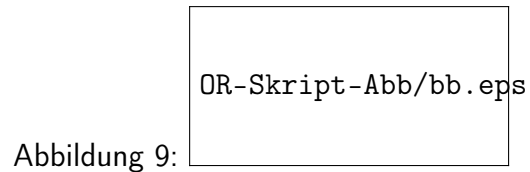
Beides wird mit Simplex gelöst und liefert:

3. $x_2 \leq 5$ (und $x_1 \leq 5$ von oben): $x_1 = 5$ und $x_2 = 5$; $G = 35$; zulässig

4. $x_2 \geq 6$ (und $x_1 \leq 5$ von oben): $x_1 = 4$ und $x_2 = 6$; $G = 36$; zulässig

Beide Lösungen sind zulässig. 4. hat mit $G = 36$ den höheren Zielfunktionswert und ist die optimale Lösung. 2. wird nicht mehr untersucht, da der Zielfunktionswert $G = 36$ von 2. durch eine weitere Einschränkung nicht vergrößert werden kann und 4. bereits eine zulässige Lösung mit $G = 36$ ist.

Häufig wird die Verzweigungshierarchie in einem Baum dargestellt:



Jeder Knoten repräsentiert eine Lösung. Die Beschriftung bedeutet:

<Name>: <Zielfunktionswert>; zulässige/unzulässige Lösung.

Die Verzweigungsbedingung steht an den Pfeilen.

Probleme in mehr Schritten sind in [?] und [?] dargestellt.

Zusammenfassung: Mit dem Branch-and-Bound-Verfahren können „herkömmliche“ Probleme auf ganzzahlige Lösungen beschränkt werden. Das Branch-and-Bound-Verfahren ist jedoch nicht auf solche Probleme beschränkt. Daher ist es angemessen, die einzelnen Schritte allgemeiner zu betrachten.

Algorithmus 8 Branch & Bound, allgemeiner

1. Falls eine zulässige Lösung vorliegt, ist deren Zielfunktionswert eine (vorläufige) untere Schranke. Zweige, die nicht besser sein können, müssen nicht weiter betrachtet werden.
2. Zunächst wird ein „relaxiertes“ Problem gelöst, für das ein bekanntes Lösungsverfahren existiert. D.h. es werden noch nicht alle Forderungen gestellt. Üblich ist, die Forderung nach Ganzzahligkeit zurückzustellen.
Beim herkömmlichen Problem ist das Simplexverfahren eine geeignete Methode, das relaxierte Problem zu lösen.
3. Es wird eine Regel zum Verzweigen gebraucht.
Beim herkömmlichen Problem wird der Lösungsraum bei einer nicht-ganzzahligen Variablen geeignet unterteilt.
4. Jedes Teilproblem wird entsprechend 2. gelöst. Es können sich folgende Fälle ergeben:
 - (a) Das *relaxierte* Problem hat eine (evtl. unzulässige) Lösung, die schlechter ist, als die bisher beste zulässige Lösung (bester Wert der Zielfunktion). Der Zweig muss nicht weiter behandelt werden.
 - (b) Das relaxierte Problem hat eine zulässige Lösung, die besser ist als die bisher beste zulässige Lösung. Damit ergibt sich ein neuer Referenzwert für die beste zulässige Lösung.
 - (c) Das relaxierte Problem hat keine Lösung. Eine weitere Betrachtung des Zweiges ist sinnlos.
 - (d) Das relaxierte Problem hat eine unzulässige Lösung, die aber besser ist, als die bisher beste zulässige Lösung. Das Problem muss weiter verzweigt werden.
5. Sind alle Teilprobleme so lange verzweigt worden, bis am Ende entweder 4a, 4b oder 4c steht, so ist die optimale Lösung (die mit der besten zulässigen Lösung) gefunden. Ist ein Zweig bis zu Ende betrachtet, so gilt er als „ausgelotet“.

Optimierungsprobleme, die keinen Simplexschritt erfordern, sind z.B. das Traveling-Salesman-Problem und das Rucksackproblem. Letzteres soll hier genauer untersucht werden.

4.2 Rucksack-Problem

Beim Rucksackproblem (Knapsackproblem) sollen Gegenstände in einen Rucksack gepackt werden. Jeder Gegenstand hat ein bestimmtes Gewicht (oder Volumen) und einen bestimmten Nutzen. Es darf ein vorgegebenes Maximalgewicht (oder maximales Gesamtvolumen) nicht überschritten werden. Der Nutzen soll maximiert werden. Jeder Gegenstand wird einzeln aufgeführt und kann entweder eingepackt werden oder eben nicht. Deshalb kann die Anzahl x_i eines Gegenstands i nur entweder $x_i = 1$ oder $x_i = 0$ sein. Man spricht auch von *binärer linearer Optimierung*.

Typische Anwendungen sind Investitionsentscheidungen.

Das Vorgehen wird direkt am Beispiel erläutert.

Beispiel: Im Fluggepäck dürfen maximal 20kg mitgenommen werden. Der erste Koffer ist gepackt und wiegt 12 kg. Für den zweiten stehen 7 Gegenstände mit unterschiedlichem Gewicht und unterschiedlichem Nutzen (in willkürlichen Einheiten) zur Verfügung:

Gewicht	5	4	3	2	1	1	1
Nutzen	5	6	1	6	8	6	4

Aus diesen Angaben wird zunächst Nutzen/Gewicht berechnet und die Nummerierung der Gegenstände in der Reihenfolge abnehmenden Nutzen/Gewicht-Verhältnisses zugeordnet.

Nutzen/Gewicht	1	1.5	$\frac{1}{3}$	3	8	6	4
Gegenstand Nr.	6	5	7	4	1	2	3

Gesucht ist nun, welche Gegenstände unter Berücksichtigung des Maximalgewichts den Nutzen maximieren.

Hier lässt sich das Problem durch eine geeignete Anpassung der Regeln effektiver lösen als durch die (prinzipiell mögliche) Rechnung über einen Simplex-Ansatz.

Neue Regeln:

Algorithmus 9 Rucksack-Problem

1. Sortierung nach Nutzen/Gewicht-Verhältnis
2. Gegenstände werden in dieser Reihenfolge eingepackt, bis die Bedingung genau erfüllt ist. Meist kann somit vom letzten Gegenstand (mit der Nummer k) nur ein Teil eingepackt werden. Die Lösung ist also unzulässig. Dies entspricht der Lösung eines *relaxierten* Problems.
3. Nun wird das unzulässige Problem verzweigt. Es sind verschiedene Verzweigungsregeln möglich. Hier wird folgende Regel verwendet: Verzweigt wird ein unzulässiges Problem, indem in einem Zweig $x_k = 0$ und im anderen Zweig $x_k = 1$ gesetzt wird. Eine andere Möglichkeit wird in [?] beschrieben.
4. In beiden Zweigen wird unter der gegebenen Verzweigungsbedingung wieder von vorne wie unter 2. eingepackt. Unzulässige Lösungen werden weiter verzweigt, wobei sich die Bedingungen weiter vererben.
5. Ein Zweig ist *ausgelotet*, wenn
 - (a) eine zulässige Lösung entsteht. Ist der Zielfunktionswert besser (hier also höher) als der bisher beste Zielfunktionswert einer zulässigen Lösung, entsteht eine neue Grenze.
 - (b) eine zulässige oder eine unzulässige Lösung entsteht, deren Zielfunktionswert unter der Grenze liegt, also schlechter ist, als die bisher beste zulässige Lösung.
 - (c) keine Lösung existiert, die die Bedingungen erfüllt.
6. Das Verfahren endet, wenn alle Zweige ausgelotet sind.

Bemerkung: Das Verhältnis und die Sortierung müssen je nach Aufgabenstellung evtl. angepasst werden.

Auf das Beispiel angewendet ergibt sich die Ausgangslösung Z0:

Z0	1	2	3	4	5	6	7	Summe
Gewicht	1	1	1	2	4	5	3	8
Nutzen	8	6	4	6	6	5	1	28.5
Anzahl x_i	1	1	1	1	$\frac{3}{4}$	0	0	

→ Z00; Z01

Die Lösung ist unzulässig, da $x_5 = \frac{3}{4}$ weder 0 noch 1 ist. Der (theoretische) Nutzen beträgt 28.5. Es wird verzweigt in Z00 mit $x_5 = 0$ und Z01 mit $x_5 = 1$. Die Darstellung ist hier zunächst noch sehr ausführlich. Die in Klausuren verwendete Darstellung ist kompakter und wird anschließend gezeigt. Beide Verfahren sind völlig äquivalent.

Z00 (1|1|1|1|0*| $\frac{3}{5}$ |0) mit N=27 → Z000; Z001

Z01 (1|1|1| $\frac{1}{2}$ |1*|0|0) mit N=27 → Z010; Z011

Der vorgegebene Wert ist mit * gekennzeichnet. Die weiteren sind von vorne aufgefüllt, bis sich wieder ein Gewicht von 8 ergibt. Die Lösungen sind beide unzulässig und müssen weiter verzweigt werden in Z000 und Z001 sowie in Z010 und Z011.

Z000 (1|1|1|1|0*|0*|1) mit N=25, dies ist die bisher einzige zulässige Lösung.

Z000	1	2	3	4	5	6	7	Summe
Gewicht	1	1	1	2	4	5	3	8
Nutzen	8	6	4	6	6	5	1	25
Anzahl x_i	1	1	1	1	0	0	1	

Z001 (1|1|1|0|0*|1*|0) mit N=23. Diese Lösung ist auch zulässig aber schlechter als Z000.

Da der Nutzen der unzulässigen Lösung Z01 mit N=27 höher liegt als der Nutzen der zulässigen Lösung Z000 mit N=25, wird Z01 weiter verzweigt.

Z010 (1|1|1|0*|1*| $\frac{1}{5}$ |0) mit N=25. Diese Lösung ist unzulässig und könnte damit weiter verzweigt werden. Allerdings ist der Nutzen mit N=25 nicht höher als der von Z000. Eine Verbesserung ist also nicht mehr möglich. Wenn nicht alle optimalen Lösungen gesucht sind, sondern nur eine, muss hier nicht weiter verzweigt werden.

Z011 (1|1|0|1*|1*|0|0) mit N=26. Diese Lösung ist zulässig und besser als Z000. Außerdem ist sie besser als Z010, sodass letztere also ab hier nicht mehr verzweigt werden muss, selbst wenn alle optimale Lösungen gesucht sind.

Z011	1	2	3	4	5	6	7	Summe
Gewicht	1	1	1	2	4	5	3	8
Nutzen	8	6	4	6	6	5	1	26
Anzahl x_i	1	1	0	1	1	0	0	

Z011 ist die optimale Lösung.

Andere Darstellung:

Genau die gleichen Schritte können auch in folgender Tabelle (ohne Erklärung der einzelnen Schritte) zusammengefasst werden. Ausgangspunkt ist die Tabelle der Aufgabe,

sowie

Nutzen/Gewicht	1	1.5	$\frac{1}{3}$	3	8	6	4
Gegenstand Nr.	6	5	7	4	1	2	3

(s.o.)

sortiert:

	Gewicht	1	1	1	2	4	5	3
	Nutzen	8	6	4	6	6	5	1

Nr.		eingepackte Menge							Nutzen	z/uz
1	Z0	1	1	1	1	$\frac{3}{4}$	0	0	28.5	uz
2	Z00	1	1	1	1	0*	$\frac{3}{5}$	0	27	uz
3	Z01	1	1	1	$\frac{1}{2}$	1*	0	0	27	uz
4	Z000	1	1	1	1	0*	0*	1	25	z
5	Z001	1	1	1	0	0*	1*	0	23	z
6	Z010	1	1	1	0*	1*	$\frac{1}{5}$	0	25	uz
7	Z011	1	1	0	1*	1*	0	0	26	z

Dies entspricht logisch genau der vorhergehenden Lösung, allerdings ohne Erklärung, dafür kompakt dargestellt.

Es ergibt sich der Verzweigungsbaum in Abbildung 10. Die Beschriftung folgt dem Schema von Abb. 9.

OR-Skript-Abb/Rucksack_B.eps

Abbildung 10:

Formulierungsvariante: Das Beispiel oben wurde im Rahmen der Namensgebung „Rucksackproblem“ formuliert. Ein Beispiel zu einer Investitionsentscheidung könnte lauten: In einer Siebdruckerei stehen maximal 8000€ zur Investition in neue Druckmaschinen zur Verfügung. Im Angebot sind drei Maschinen zu je 1000€ sowie je eine Maschine zu 2000€, 4000€, 5000€ und 3000€. Pro Stunde können bei

gleicher Qualität 8000, 6000, 4000, 6000, 6000, 5000 und 1000 Drucke hergestellt werden. Welche Maschinen sollten angeschafft werden?

Wenn die GE in 1000€ und die Anzahl der Drucke in 1000 Stück pro Stunde gemessen werden, ergibt sich rechnerisch genau die Aufgabe des oben durchgerechneten Beispiels.

4.3 Branch&Bound Zusammenfassung

“Herkömmliches” Optimierungsproblem mit der zusätzlichen Bedingung für Ganzzahligkeit

1. Mathematisches Modell aufstellen
2. Lösen des relaxierten Problems, bei dem die Ganzzahligkeit noch nicht berücksichtigt wird (Simplex). Damit ergibt sich eine Lösung, bei der $x_i = a$ nicht ganzzahlig ist.
3. Aufspalten des Problems in zwei Teilprobleme mit jeweils einer zusätzlichen Bedingung:
 - (a) $x_i \leq n$, wobei n die größte ganze Zahl kleiner a ist.
 - (b) $x_i \geq n + 1$
4. Lösen der relaxierten Teilprobleme und weiteres Aufteilen, bis die beste rein ganzzahlige Lösung übrig bleibt.

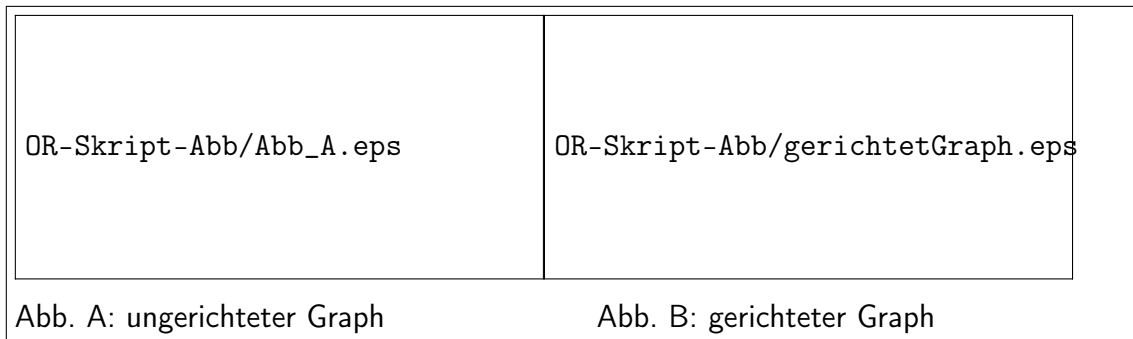
Rucksackproblem

1. Sortierung nach Nutzen/Gewicht-Verhältnis
2. Gegenstände einpacken, bis die Bedingung genau erfüllt ist (relaxiertes Problem).
3. Verzweigen; mit je einer geeigneten weiteren Bedingung für jeden Zweig
4. In beiden Zweigen wird unter der gegebenen Verzweigungsbedingung wieder wie unter 2. eingepackt. Unzulässige Lösungen werden weiter verzweigt, wobei sich die Bedingungen weiter vererben.
5. ausloten der Zweige

Das Verfahren endet, wenn alle Zweige ausgelotet sind.

5 Graphentheorie

Ein Graph besteht aus Knoten V_i (Vertex) und sie verbindenden Kanten E_j (Edge). Sind die Kanten gerichtet, spricht man von Pfeilen. Die Kanten sind meist gewichtet (oder bewertet).



Ein Knoten kann z.B. eine Stadt darstellen und die gewichteten Kanten sind die Verbindungswege oder -kosten z.B. eines Versorgungsnetzes (Strom, Wasser, Gas,...). Eine typische Fragestellung ist die nach dem kürzesten oder dem billigsten Weg von einer Stadt zu einer anderen.

Einige Grundbegriffe

- Bei einem *gerichteten* Graphen bestehen die Kanten aus Pfeilen (Abb. B). Sind die Kanten nicht mit einer Richtung versehen, spricht man von *ungerichteten* Graphen (Abb. A).
- *Parallele* Pfeile oder Kanten sind solche mit gleichen Anfangs- und Endknoten. Die Beispielgraphen haben keine parallelen Kanten oder Pfeile.
- Eine Kante oder ein Pfeil, die/der einen Knoten mit diesem direkt verbindet, heißt *Schlinge*. Die Beispielgraphen haben keine Schlingen.
- Ein Graph ohne parallele Pfeile oder Kanten und ohne Schlingen heißt *schlichter* Graph. Beide Graphen der Abbildungen sind schlichte Graphen.
- Ein schlichter, gerichteter Graph heißt *Digraph*, s. Abb. B.
- Ein *bewerteter* Graph hat Gewichtungen an Pfeilen oder Kanten. Die obigen Abbildungen stellen beide bewertete Graphen dar.

- Besteht zwischen zwei Knoten eines Graphen eine Verbindung über andere Knoten, so heißt diese eine *Kette*. Z.B. gibt es eine Kette von Knoten 1 zu Knoten 4 in beiden Abbildungen A und B.
- Der Weg von einem Ausgangsknoten über weitere Knoten zurück zum Ausgangsknoten heißt *Kreis* oder *Zyklus*.
- In einem *zusammenhängenden* Graphen sind alle Knoten direkt oder über eine Kette verbunden. Beide Graphen der Abbildungen sind zusammenhängende Graphen. Der gerichtete Graph (Abb. B) ist jedoch nur *schwach zusammenhängend*, da erst ein zusammenhängender Graph entsteht, wenn die gerichteten Kanten durch ungerichtete ersetzt werden.
Bei einem *stark zusammenhängenden* gerichteten Graphen müsste dagegen ein gerichteter Weg von jedem Knoten zu jedem anderen bestehen.
- Ein zusammenhängender Graph ohne Kreis heißt *Baum*.
- Ein zusammenhängender und kreisloser Teil eines Graphen, der alle Knoten dieses Graphen enthält, heißt *spannender Baum*.
- Ein *minimal spannender Baum* ist ein spannender Baum mit minimaler Summe der Kantengewichtungen.

5.1 Kürzeste Wege in Graphen

Hier gibt es zwei Fragestellungen:

1. Welche kürzesten Ketten verbinden einen bestimmten Ausgangsknoten mit allen anderen Knoten des Graphen?
2. Welche kürzesten Ketten verbinden jeden Knoten mit allen anderen Knoten des Graphen?

5.1.1 Dijkstra Algorithmus

Mit Hilfe des Dijkstra Algorithmus wird die erste der Ausgangsfragen beantwortet: Welches sind die kürzesten Wege von einem gegebenen Ausgangsknoten zu allen anderen Knoten des Graphen? Hier folgt zunächst ganz formal der Algorithmus. Es ist aber sinnvoll, diesen gleich anhand des Beispiels zu betrachten.

Algorithmus 10 Dijkstra

Voraussetzungen

- gegebener Startknoten $start$
- Vektor $D(i)$, in dem zu jedem Zeitpunkt der kürzeste Abstand von $start$ zum Knoten i gespeichert ist.
- Vektor $V(i)$, der zu jedem Zeitpunkt den Vorgänger des Knotens i enthält auf dem bis dahin bekannten kürzesten Weg von $start$ zu i .
- M ist die Menge von Knoten aus der der nächste zu behandelnde Knoten gewählt wird.
- $N(a)$ ist die Menge aller direkt nachfolgenden Knoten von Knoten a .
- k_{ij} ist die Gewichtung des Pfeiles von Knoten i zu Knoten j . Wenn keine Verbindung besteht, ist der entsprechende Wert ∞ .

Initialisierungen

- $M = \{start\}$, zu Beginn besteht die Menge der ausgewählten Knoten nur aus dem Startknoten
 - $D(start) = 0$ und $D(i) = \infty$ für alle anderen Knoten
 - $V(i)$ ist leer für alle Knoten
-

Iteration

1. Aus der Menge M wird ein Knoten a ausgewählt. Wenn (wie meist) mehrere Knoten in der Menge stehen, wird der Knoten mit minimalem Abstand zu $start$ gewählt. Für a gilt also, dass $D(a) = \min [D(i) \text{ mit } i \in M]$.
2. Die Nachbarknoten von a (in Vorwärtsrichtung) stehen in $N(a)$. Die folgende Untersuchung wird auf alle diese Knoten angewendet.
Falls $D(a) + k_{aj} < D(j)$ gibt es einen kürzeren Weg zu Knoten j . Dann wird gesetzt:
 - $D(j) = D(a) + k_{aj}$
 - $V(j) = a$
 - j wird in die Menge M aufgenommen, falls j nicht schon enthalten ist
3. a wird aus der Menge M der markierten Knoten entfernt.

Falls die Menge M leer ist, also $M = \{\}$, ist das Verfahren zu Ende. Sonst geht es weiter ab Schritt 1.

In $V(i)$ steht nun der Vorgängerknoten von Knoten i auf dem kürzesten Weg vom Anfangsknoten. Um den kürzesten Weg zum Zielknoten z zu bestimmen, wird der Weg ausgehend von z rückwärts verfolgt.

Eine ausführliche Schritt für Schritt Berechnung zum Graphen B ist in den Ergänzungen Kap.???. Entsprechend dieser Berechnung kann gegebenenfalls ein Programm geschrieben werden.

Im Folgenden wird eine mehr intuitive Vorgehensweise besprochen, wie sie auch in der Klausur vorkommen kann.

Beispiel: Es sollen die kürzesten Wege des gerichteten Graphen B' von Knoten 1 zu allen anderen Knoten bestimmt werden.

OR-Skript-Abb/GraphB2.eps

Abb. gerichteter Graph B'

Es handelt sich um den Graphen B von oben mit vertauschter Bezeichnung der Knoten 2 und 3. Dies ermöglicht eine deutlichere Abgrenzung der beiden hier betrachteten Algorithmen Dijkstra und FIFO (nächster Abschnitt). Letzterer behandelt die gleiche Fragestellung, unterscheidet sich im Lösungsweg aber geringfügig.

Die Abfolge der einzelnen Lösungsschritte ist in folgender Tabelle zusammengefasst und wird anschließend erklärt:

a	i=	1	2	3	4	5	Menge M
	D(i)=	0	∞	∞	∞	∞	
-	V(i)=	-	-	-	-	-	1
	D(i)=	0	2	1	∞	∞	
1	V(i)=	-	1	1	-	-	2, 3
	D(i)=	0	2	1	7	10	
3	V(i)=	-	1	1	3	3	2, 4, 5
	D(i)=	0	2	1	5	7	
2	V(i)=	-	1	1	2	2	4, 5
	D(i)=	0	2	1	5	6	
4	V(i)=	-	1	1	2	4	5
	D(i)=	0	2	1	5	6	
5	V(i)=	-	1	1	2	4	

Zunächst ist der Anfangsknoten $a=1$ in der Menge M vorgegeben. Dementsprechend ist die Entfernung $D=0$ und es gibt keinen Vorgänger V . Im ersten Schritt wird untersucht, ob es irgendwelche Verbindungen gibt. Dies ist der Fall: nach Knoten 2 und 3. Die Entfernungen werden eingetragen. Vorgänger ist jeweils Knoten 1. Die Menge M wird um die Knoten 2 und 3 erweitert und der betrachtete Knoten 1 wird aus M gelöscht.

Im zweiten Schritt wird der Knoten 3 als Anfangsknoten gewählt, da er den geringeren Abstand D zu Knoten 1 hat. Über Knoten 3 kommt man sowohl nach 4 als auch

nach 5. Nun wird die **Gesamtentfernung** von Knoten 1 eingetragen sowie Knoten 3 als Vorgänger. Die Menge M erhält zusätzlich 4 und 5. 3 wird entfernt.

Im dritten Schritt wird eine entsprechende Untersuchung über Knoten 2 durchgeführt, da Knoten 2 den geringsten Abstand D zu Knoten 1 hat.

Im vierten Schritt wird eine entsprechende Untersuchung über Knoten 4 durchgeführt, da Knoten 4 den geringsten Abstand D zu Knoten 1 hat.

Im fünften Schritt wird eine entsprechende Untersuchung über Knoten 5 durchgeführt. Hier ergibt sich keine Änderung in Abstand und Vorgänger mehr. Die Menge M wird geleert und das Verfahren ist beendet.

Übung 1: Führen Sie die Rechnung mit Graph B durch.

Ergebnis:

(Graph B) a	i=	1	2	3	4	5	Menge M
	D(i)=	0	∞	∞	∞	∞	
-	V(i)=	-	-	-	-	-	1
	D(i)=	0	1	2	∞	∞	
1	V(i)=	-	1	1	-	-	2, 3
	D(i)=	0	1	2	7	10	
2	V(i)=	-	1	1	2	2	3, 4, 5
	D(i)=	0	1	2	5	7	
3	V(i)=	-	1	1	3	3	4, 5
	D(i)=	0	1	2	5	6	
4	V(i)=	-	1	1	2	4	5
	D(i)=	0	1	2	5	6	
5	V(i)=	-	1	1	2	4	

Übung 2: Führen Sie das gleiche Verfahren mit dem ungerichteten Graphen A durch. Beachten Sie hierbei, dass die Kanten in beide Richtungen durchlaufen werden können. Die ausführliche Rechnung mit Ergebnis ist in den Ergänzungen Kap.??.

5.1.2 FIFO-Algorithmus

Der FIFO-Algorithmus löst das gleiche Problem, wie der Dijkstra Algorithmus und unterscheidet sich von diesem nur in der Auswahl des Knotens a . Er wird einer Warteschlange entnommen, die nach dem FIFO-(First In First Out) Prinzip organisiert ist. Die anderen Schritte sind wie bei Dijkstra.

Algorithmus 11 FIFO

Voraussetzungen

- gegebener Startknoten $start$
 - Vektor $D(i)$, in dem zu jedem Zeitpunkt der kürzeste Abstand von $start$ zum Knoten i gespeichert ist.
 - Vektor $V(i)$, der zu jedem Zeitpunkt den Vorgänger des Knotens i enthält auf dem bis dahin bekannten kürzesten Weg von $start$ zu i .
 - S ist die Schlange von Knoten aus der der nächste Knoten gewählt wird.
 - KS ist der Kopf der Schlange. Er ersetzt den Knoten a bei Dijkstra.
 - ES ist das Ende der Schlange.
 - $N(KS)$ ist die Menge aller nachfolgenden Knoten von KS .
 - k_{ij} ist die Gewichtung des Pfeiles von Knoten i zu Knoten j . Wenn keine Verbindung besteht, ist der entsprechende Wert ∞ .
-

Initialisierungen

- $KS = ES = start$, zu Beginn besteht die Schlange nur aus dem Startknoten
 - $D(start) = 0$ und $D(i) = \infty$ für alle anderen Knoten
 - $V(i)$ ist leer für alle Knoten
-

Iteration

1. KS ist der Kopf der Schlange
2. Alle Nachbarknoten von KS (in Vorwärtsrichtung) stehen in $N(KS)$. Der Index j läuft über alle $N(KS)$.
Falls $D(KS) + k_{KS,j} < D(j)$ gibt es einen kürzeren Weg zu Knoten j . Dann wird gesetzt:
 - $D(j) = D(KS) + k_{KS,j}$
 - $V(j) = KS$
 - j wird an die Warteschlange hinten angestellt, sofern es noch nicht in der Schlange enthalten ist.
3. Falls $KS = ES$, ist das Verfahren zu Ende.
4. Der alte Schlangenkopf KS wird entfernt. Damit rückt das nächste Element nach. Weiter mit Schritt 1.

Wie oben, wird auch hier eine kompakte Vorgehensweise betrachtet. Die ausführliche Rechnung zum Graphen B ist in den Ergänzungen Kap.??.

Beispiel: Es sollen die kürzesten Wege des gerichteten Graphen Abb. B' von Knoten 1 zu allen anderen Knoten bestimmt werden.

OR-Skript-Abb/GraphB2.eps

Die Knoten werden wieder entsprechend ihrer Nummerierung abgearbeitet:

KS	i=	1	2	3	4	5	Schlange
	D(i)=	0	∞	∞	∞	∞	
-	V(i)=	-	-	-	-	-	<1
	D(i)=	0	2	1	∞	∞	
1	V(i)=	-	1	1	-	-	<2, 3
	D(i)=	0	2	1	5	7	
2	V(i)=	-	1	1	2	2	<3, 4, 5
***	D(i)=	0	2	1	5	7	
3	V(i)=	-	1	1	2	2	<4, 5
	D(i)=	0	2	1	5	6	
4	V(i)=	-	1	1	2	4	<5
	D(i)=	0	2	1	5	6	
5	V(i)=	-	1	1	2	4	

Beachten Sie den Unterschied: hier wird im markierten Schritt (mit Anfangsknoten 3) keine Verbesserung mehr vorgenommen. Bei Dijkstra war dies anders: dort war Knoten 3 bereits im vorhergehenden Schritt Anfangsknoten, statt Knoten 2, da dort nicht die Schlange abgearbeitet wird, sondern immer der Knoten mit kleinstem Abstand gewählt wird.

Die Markierung *** hat nichts mit dem Lösungsgang zu tun. Sie wurde hier nur eingefügt, um den Verweis eindeutig zu machen.

Übung: Es sollen wie oben die kürzesten Wege des gerichteten Graphen Abb. B von Knoten 1 zu allen anderen Knoten bestimmt werden.

Ergebnis:

KS	i=	1	2	3	4	5	Schlange
	D(i)=	0	∞	∞	∞	∞	
-	V(i)=	-	-	-	-	-	<1
	D(i)=	0	1	2	∞	∞	
1	V(i)=	-	1	1	-	-	<2, 3
	D(i)=	0	1	2	7	10	
2	V(i)=	-	1	1	2	2	<3, 4, 5
	D(i)=	0	1	2	5	7	
3	V(i)=	-	1	1	3	3	<4, 5
	D(i)=	0	1	2	5	6	
4	V(i)=	-	1	1	3	4	<5
	D(i)=	0	1	2	5	6	
5	V(i)=	-	1	1	3	4	

Bemerkung: Wir können leicht sehen, dass die einzelnen Schritte denen von Dijkstra entsprechen. Lediglich die Darstellung der letzten Spalte ist anders. Das Vorgehen unterscheidet sich zwar in folgenden zwei Punkten, die Abfolge ändert sich gegenüber Dijkstra in diesem Beispiel jedoch nicht.

1. Jeder verbesserte Knoten wird in der Schlange *hinten* angestellt.
2. Der nächste Anfangsknoten ist immer der Schlangenkopf.

Tatsächlich müssen nur die Endergebnisse der beiden Verfahren übereinstimmen. Die Zwischenschritte können sich unterscheiden.

5.1.3 Tripel Algorithmus

Auch beim Tripel Algorithmus (auch Floyd-Algorithmus) werden kürzeste Wege bestimmt, jedoch von jedem Knoten zu jedem anderen.

Algorithmus 12 Tripel

Voraussetzungen

- Graph mit n Knoten
 - Matrix $D(i, j)$, in der zu jedem Zeitpunkt der kürzeste Abstand von Knoten i zum Knoten j gespeichert ist. Zu Beginn wird damit der Graph definiert.
 - Matrix $V(i, j)$, die zu jedem Zeitpunkt den Vorgänger des Knotens j enthält auf dem bis dahin bekannten kürzesten Weg von i zu j .
-

Initialisierungen

- $D(i, j) = \begin{cases} 0 & \text{für } i = j \\ k_{ij} & \text{für eine direkte Verbindung von } i \text{ zu } j \\ \infty & \text{sonst} \end{cases}$
 - $V(i, j) = \begin{cases} i & \text{für } i = j \\ i & \text{falls eine direkte Verbindung von } i \text{ zu } j \text{ existiert} \\ 0 & \text{sonst} \end{cases}$
-

Iteration

- Schleife $j = 1$ bis n # mittlerer Knoten bzw. Verbindungsknoten
 - Schleife $i = 1$ bis n # Anfang

- * Schleife $k = 1$ bis $n \#$ Ende
 - Falls $D(i, j) + D(j, k) < D(i, k)$ gibt es einen kürzeren Weg von Knoten i zu Knoten k . Dann wird gesetzt:

$$D(i, k) = D(i, j) + D(j, k)$$

$$V(i, k) = V(j, k)$$
- * Ende Schleife über k
- Ende Schleife über i
- Ende Schleife über j

Am Ende enthält die Matrix D die kürzesten Wege (Ketten) von jedem Knoten zu jedem anderen. Der Weg über die einzelnen Zwischenknoten kann ausgehend vom Endknoten über die jeweiligen Vorgänger aus Matrix V ermittelt werden.

Beispiel: Es sollen die kürzesten Wege des gerichteten Graphen Abb. B von allen Knoten zu allen anderen Knoten bestimmt werden.

OR-Skript-Abb/gerichtetGraph.eps

Initialisierungen: $D_{ij} = \begin{bmatrix} 0 & 1 & 2 & \infty & \infty \\ \infty & 0 & \infty & 6 & 9 \\ \infty & \infty & 0 & 3 & 5 \\ \infty & \infty & \infty & 0 & 1 \\ \infty & \infty & \infty & 2 & 0 \end{bmatrix}$ und $V_{ij} = \begin{bmatrix} 1 & 1 & 1 & 0 & 0 \\ 0 & 2 & 0 & 2 & 2 \\ 0 & 0 & 3 & 3 & 3 \\ 0 & 0 & 0 & 4 & 4 \\ 0 & 0 & 0 & 5 & 5 \end{bmatrix}$

Bemerkung: Zu Beginn enthält die Vorgängermatrix in jeder Zeile also ausschließlich Nullen oder die Zahl, die ihrer Zeilennummer entspricht. D.h. ein Knoten der Spaltennummer hat den Knoten der Zeilennummer entweder als (direkten) Vorgänger oder nicht. Jeder Knoten gilt als sein eigener Vorgänger.

Achtung: Im Lauf der Iterationen ändert sich die Vorgängermatrix und als Vorgänger gelten nicht nur direkte Vorgänger, sondern auch solche, die über eine Kette zum Zielknoten (entsprechend der Spaltennummer) führen.

Iteration

Falls keine Wege von einem Knoten i zu Knoten j oder von j zu einem anderen Knoten k existieren, muss die Schleife nicht betrachtet werden. In anderen Worten: der Verbindungsknoten j muss andere Knoten miteinander verbinden. Sonst ist eine weitere Betrachtung sinnlos.

Der Weg von j zu j ist uninteressant, da $D_{jj} = 0$ und $V_{jj} = j$ sich nicht ändern.

OR-Skript-Abb/gerichtetGraph.eps

Schleife $j=1$

da kein Weg zu j führt, ist die Schleife zu Ende und die Matrizen bleiben gleich.

Schleife $j=2$

- zu $j=2$ $D_{12} = 1$
- von $j=2$ $D_{24} = 6$; $D_{25} = 9$

Es müssen also zwei Kombinationen überprüft werden:

1. $D_{12} + D_{24} = 7$ im Vergleich zu $D_{14} = \infty$

Es gibt eine Verbesserung. Es wird gesetzt $D_{14} = 7$ und $V_{14} = 2$

2. $D_{12} + D_{25} = 10$ im Vergleich zu $D_{15} = \infty$

Es gibt eine Verbesserung. Es wird gesetzt $D_{15} = 10$ und $V_{15} = 2$

$$\text{Matrizen: } D_{ij} = \begin{bmatrix} 0 & 1 & 2 & 7 & 10 \\ \infty & 0 & \infty & 6 & 9 \\ \infty & \infty & 0 & 3 & 5 \\ \infty & \infty & \infty & 0 & 1 \\ \infty & \infty & \infty & 2 & 0 \end{bmatrix} \quad \text{und } V_{ij} = \begin{bmatrix} 1 & 1 & 1 & 2 & 2 \\ 0 & 2 & 0 & 2 & 2 \\ 0 & 0 & 3 & 3 & 3 \\ 0 & 0 & 0 & 4 & 4 \\ 0 & 0 & 0 & 5 & 5 \end{bmatrix}$$

Schleife $j=3$

- zu $j=3$ $D_{13} = 2$

- von $j=3$ $D_{34} = 3$; $D_{35} = 5$

Es müssen also zwei Kombinationen überprüft werden:

1. $D_{13} + D_{34} = 5$ im Vergleich zu $D_{14} = 7$

Es gibt eine Verbesserung. Es wird gesetzt $D_{14} = 5$ und $V_{14} = 3$

2. $D_{13} + D_{35} = 7$ im Vergleich zu $D_{15} = 10$

Es gibt eine Verbesserung. Es wird gesetzt $D_{15} = 7$ und $V_{15} = 3$

$$\text{Matrizen: } D_{ij} = \begin{bmatrix} 0 & 1 & 2 & 5 & 7 \\ \infty & 0 & \infty & 6 & 9 \\ \infty & \infty & 0 & 3 & 5 \\ \infty & \infty & \infty & 0 & 1 \\ \infty & \infty & \infty & 2 & 0 \end{bmatrix} \quad \text{und } V_{ij} = \begin{bmatrix} 1 & 1 & 1 & 3 & 3 \\ 0 & 2 & 0 & 2 & 2 \\ 0 & 0 & 3 & 3 & 3 \\ 0 & 0 & 0 & 4 & 4 \\ 0 & 0 & 0 & 5 & 5 \end{bmatrix}$$

Schleife $j=4$

- zu $j=4$ $D_{14} = 5$; $D_{24} = 6$; $D_{34} = 3$; $D_{54} = 2$

- von $j=4$ $D_{45} = 1$

Es müssen also vier Kombinationen überprüft werden:

1. $D_{14} + D_{45} = 6$ im Vergleich zu $D_{15} = 7$

Es gibt eine Verbesserung. Es wird gesetzt $D_{15} = 6$ und $V_{15} = 4$

2. $D_{24} + D_{45} = 7$ im Vergleich zu $D_{25} = 9$

Es gibt eine Verbesserung. Es wird gesetzt $D_{25} = 7$ und $V_{25} = 4$

3. $D_{34} + D_{45} = 4$ im Vergleich zu $D_{35} = 5$

Es gibt eine Verbesserung. Es wird gesetzt $D_{35} = 4$ und $V_{35} = 4$

4. $D_{54} + D_{45}$ führt von 5 zu 5, muss also nicht betrachtet werden.

$$\text{Matrizen: } D_{ij} = \begin{bmatrix} 0 & 1 & 2 & 5 & 6 \\ \infty & 0 & \infty & 6 & 7 \\ \infty & \infty & 0 & 3 & 4 \\ \infty & \infty & \infty & 0 & 1 \\ \infty & \infty & \infty & 2 & 0 \end{bmatrix} \quad \text{und } V_{ij} = \begin{bmatrix} 1 & 1 & 1 & 3 & 4 \\ 0 & 2 & 0 & 2 & 4 \\ 0 & 0 & 3 & 3 & 4 \\ 0 & 0 & 0 & 4 & 4 \\ 0 & 0 & 0 & 5 & 5 \end{bmatrix}$$

Schleife $j=5$

Von jedem Knoten aus führen alle optimalen Wege über 4 nach 5. Da Knoten 5 nur zu Knoten 4 zurückführt, ergibt sich keine Verbesserung.

Die Matrizen nach Schleife 4 stellen also die optimalen Wege dar. Der kürzeste Weg wird rekursiv ermittelt. Z.B. von 1 nach 5 (vergl. Dijkstra oder FIFO): Die Gesamtlänge

beträgt $D_{15} = 6$ und der Vorgänger von 5 auf diesem Weg ist $V_{15} = 4$. Der Vorgänger von 4 auf diesem Weg ist $V_{14} = 3$. Der Vorgänger von 3 auf diesem Weg ist $V_{13} = 1$. Der Weg ist also 1-3-4-5.

Übungen: 1. Führen Sie die Schleife $j=5$ (Bsp. oben) aus.

2. Wenden Sie den Tripelalgorithmus auf den ungerichteten Graphen Abb. A an.

$$\text{Ergebnis: } D_{ij} = \begin{bmatrix} 0 & 1 & 3 & 4 & 5 \\ 1 & 0 & 2 & 3 & 4 \\ 3 & 2 & 0 & 1 & 2 \\ 4 & 3 & 1 & 0 & 1 \\ 5 & 4 & 2 & 1 & 0 \end{bmatrix} \text{ und } V_{ij} = \begin{bmatrix} 1 & 1 & 2 & 3 & 4 \\ 2 & 2 & 2 & 3 & 4 \\ 2 & 3 & 3 & 3 & 4 \\ 2 & 3 & 4 & 4 & 4 \\ 2 & 3 & 4 & 5 & 5 \end{bmatrix}$$

5.2 Minimal spannende Bäume

Minimal spannende Bäume sind Teilgraphen von gewichteten Graphen. Sie enthalten und verbinden alle Knoten. Die Summe der Gewichtungen soll so klein wie möglich sein.

Ein leicht vorstellbares Beispiel: Einige Ortschaften sollen an ein Gasnetz angeschlossen werden. Der Verteiler liegt in einem der Orte. Die verschiedenen Verbindungsmöglichkeiten zwischen den Orten mit Gasleitungen sind einschließlich ihrer Kosten bekannt. Ziel ist es, jeden Ort, evtl. über andere Orte, mit dem zentralen Verteiler zu verbinden. Die Baukosten sollen minimal sein.

Die Orte können als Knoten eines Graphen dargestellt werden. Die möglichen Gasleitungen sind die Kanten und die Gewichtung der Kanten entspricht den Baukosten.

5.2.1 Kruskal Algorithmus

Es wird ein Graph K erstellt, der alle Knoten des gegebenen Graphen G enthält, aber keine Kanten. Die Kanten des Graphen G werden nach aufsteigender Reihenfolge ihrer Bewertungen k_{ij} sortiert. Daraus wird der minimal spannende Baum konstruiert: die Kanten werden nacheinander in den Graphen K in der Reihenfolge aufsteigender Bewertung eingefügt. Entsteht durch Hinzunahme einer Kante ein Kreis, wird diese Kante weggelassen. Als Ergebnis entsteht der minimal spannende Baum.

Algorithmus 13 Kruskal

Voraussetzung

- ein bewerteter, ungerichteter, zusammenhängender, schlingenfreier Graph G mit n Knoten und m Kanten
 - ein Graph K mit den n Knoten von G aber ohne Kanten
-

Lösungsschritte

1. Die Kanten werden nach aufsteigender Reihenfolge ihrer Bewertungen k_{ij} sortiert.
2. Schleife $i = 1$ bis m (Schleife über alle Kanten)
 - Kante i wird dem Graphen K hinzugefügt, wenn dadurch kein Kreis entsteht. Sonst wird die Kante übergangen.

Ende der Schleife über i

Wenn $(n - 1)$ Kanten in den Graphen K eingefügt wurden, ist das Verfahren zu Ende. K ist der minimal spannende Baum.

Beispiel: Es soll der minimal spannende Baum eines ungerichteten Graphen bestimmt werden. Es wird eine modifizierte Version des Graphen der Abbildung A verwendet:

OR-Skript-Abb/Abb_A_mod.eps

1. $k_{12} = 1$ (Kante 1); $k_{45} = 1$ (Kante 2); $k_{23} = 2$ (Kante 3); $k_{13} = 3$ (Kante 4);
 $k_{24} = 4$ (Kante 5); $k_{34} = 5$ (Kante 6); $k_{35} = 5$ (Kante 7)
2. Schleifen (Die folgenden Abbildungen zeigen nur die bereits behandelten Knoten.)
 - (a) Schleife $i=1$; Kante (1, 2) wird hinzugefügt

OR-Skript-Abb/kruskalBaum0.eps

- (b) Schleife $i=2$; Kante $(4, 5)$ wird hinzugefügt

OR-Skript-Abb/kruskalBaum1.eps

Der Graph K ist zu diesem Zeitpunkt nicht zusammenhängend.

- (c) Schleife $i=3$; Kante $(2, 3)$ wird hinzugefügt

OR-Skript-Abb/kruskalBaum2.eps

Auch hier sind noch zwei nicht zusammenhängende Teilgraphen.

- (d) Schleife $i=4$; Kante $(1, 3)$ wird nicht hinzugefügt, da sonst ein Kreis entsteht.

- (e) Schleife $i=5$; Kante $(2, 4)$ wird hinzugefügt

OR-Skript-Abb/kruskalBaum4.eps

- (f) Schleife $i=6$ muss nicht mehr untersucht werden, da bereits $n - 1 = 4$ Kanten eingefügt wurden. Der minimal spannende Baum ist vollständig.

Wir werden intern für eine kompaktere Darstellung des Ergebnisses in der Klausur folgende Darstellung verwenden:

OR-Skript-Abb/kruskalBaum4Reihenfolge.eps

Hier stehen an den Kanten nicht deren Gewichtungen, sondern die Reihenfolge, in der die Kanten eingefügt wurden. Dies ist keine offizielle Darstellung, spart aber in der Klausur Zeit und enthält alle für eine Klausur wichtigen Informationen.

5.2.2 Prim Algorithmus

Der Prim Algorithmus ist dem von Kruskal sehr ähnlich. Auch hier werden die Kanten nach aufsteigender Reihenfolge ihrer Bewertungen k_{ij} sortiert. Der neue Graph K besteht am Anfang aus einem Knoten. Von diesem Knoten ausgehend werden in der vorgegebenen Reihenfolge nur Kanten und Knoten hinzugefügt, die einen zusammenhängenden Teilbaum ergeben. Auch hier werden Kanten übergangen, die zu einem Kreis führen. Der Graph K wächst, bis alle Knoten von G und damit $n - 1$ Kanten enthalten sind.

Algorithmus 14 Prim

Voraussetzung

- ein bewerteter, ungerichteter, zusammenhängender, schlingenfreier Graph G mit n Knoten und m Kanten
 - ein Anfangsknoten
-

Lösungsschritte

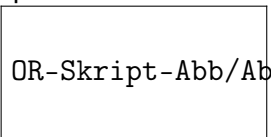
1. Die Kanten werden nach aufsteigender Reihenfolge ihrer Bewertungen k_{ij} sortiert.
2. Schleife $i = 1$ bis $n-1$ $\#$ Anzahl der benötigten Kanten
 - Schleife $j = 1$ bis m_i $\#$ Schleife über alle sortierten Nachbarknoten des Teilgraphen
 - Kante j (einschließlich des Endknoten) wird dem Graphen K hinzugefügt, wenn dieser dadurch zusammenhängend bleibt und kein Kreis entsteht. Sonst wird die Kante übergangen.

Ende der Schleife über j , nachdem eine Kante hinzugefügt wurde

Ende der Schleife über i

Der Graph K enthält nun $(n - 1)$ Kanten. Das Verfahren endet. K ist der minimal spannende Baum.

- 1. Beispiel:** Es soll der minimal spannende Baum des ungerichteten Graphen des Beispiels in Abschnitt ?? bestimmt werden. Der Anfangsknoten sei 1.



OR-Skript-Abb/Abb_A_mod.eps

1. $k_{12} = 1$ (Kante 1); $k_{45} = 1$ (Kante 2); $k_{23} = 2$ (Kante 3); $k_{13} = 3$ (Kante 4);
 $k_{24} = 4$ (Kante 5); $k_{34} = 5$ (Kante 6); $k_{35} = 5$ (Kante 7)
2. Schleifen:

- (a) Schleife $i=1$; Kante $(1, 2)$ (=Kante 1) wird hinzugefügt

OR-Skript-Abb/kruskalBaum0.eps

- (b) Schleife $i=2$; Kante $(4, 5)$ (=Kante 2) wird nicht hinzugefügt, da weder Knoten 4 noch Knoten 5 im bisherigen Teilbaum enthalten sind. Die Kante mit der kleinsten Bewertung, die diese Bedingung erfüllt, ist Kante $(2, 3)$ (=Kante 3). Sie wird hinzugefügt

OR-Skript-Abb/primBaum12.eps

- (c) Schleife $i=3$; Kante $(2, 4)$ (=Kante 5) wird hinzugefügt

OR-Skript-Abb/primBaum24.eps

- (d) Schleife $i=4$; Kante $(4, 5)$ (=Kante 2) wird hinzugefügt

OR-Skript-Abb/primBaum31.eps

- (e) Nun sind $n - 1 = 4$ Kanten eingefügt. Der minimal spannende Baum ist vollständig.

Unsere interne, kompaktere Darstellung ist hier:

OR-Skript-Abb/primBaum31Reihenfolge.eps

- 2. Beispiel:** Hier soll gezeigt werden, dass das Ergebnis unabhängig vom Startknoten ist. Es soll wieder der minimal spannende Baum des ungerichteten Graphen von Beispiel 1 in Abschnitt ?? bestimmt werden. Der Anfangsknoten sei 5.

OR-Skript-Abb/Abb_A_mod.eps

1. $k_{12} = 1$ (Kante 1); $k_{45} = 1$ (Kante 2); $k_{23} = 2$ (Kante 3); $k_{13} = 3$ (Kante 4);
 $k_{24} = 4$ (Kante 5); $k_{34} = 5$ (Kante 6); $k_{35} = 5$ (Kante 7)

2. Schleifen:

- (a) Schleife $i=1$; Kante (1, 2) (=Kante 1) wird nicht hinzugefügt, da weder Knoten 1 noch Knoten 2 im bisherigen Teilbaum enthalten sind.

Kante (4, 5) (=Kante 2) wird hinzugefügt

OR-Skript-Abb/primBaum54.eps

- (b) Schleife $i=2$; Kanten 1, 2 und 3 werden nicht hinzugefügt, da sich kein zusammenhängender Teilgraph ergeben würde. Die Kante mit der kleinsten Bewertung, die diese Bedingung erfüllt, ist Kante (2, 4) (=Kante 5). Sie wird hinzugefügt

OR-Skript-Abb/primBaum245.eps

- (c) Schleife $i=3$; Kante (1, 2) (=Kante 1) wird hinzugefügt

OR-Skript-Abb/primBaum1245.eps

- (d) Schleife $i=4$; Kante (2, 3) (=Kante 3) wird hinzugefügt

OR-Skript-Abb/primBaum31.eps

- (e) Nun sind $n - 1 = 4$ Kanten eingefügt. Der minimal spannende Baum ist vollständig. Das Ergebnis ist wie oben.

Unsere interne, kompaktere Darstellung unterscheidet sich allerdings da die Reihenfolge anders ist:

OR-Skript-Abb/primBaum1245Reihenfolge.eps

5.2.3 Vergleich der Algorithmen

Die beiden Algorithmen zur Bestimmung eines minimal spannenden Baumes sind hier auf eine leicht nachvollziehbare Art angegeben. Implementierungen laufzeitoptimierter Versionen sprengen den Rahmen dieses Manuskripts und der Vorlesung.

Für die Laufzeit gilt: Wenn $|E|$ und $|V|$ die Anzahl von Knoten und Kanten eines Graphen ist, ist für Kruskal die Laufzeit $\mathcal{O}(|E| \log |V|)$ und für Prim $\mathcal{O}(|E| + |V| \log |V|)$. Prim ist also schneller, wenn $|E| \gg |V|$.

5.3 Graphentheorie Zusammenfassung

A. Grundbegriffe

B. kürzeste Wege

1. Dijkstra

- (a) Anfangsknoten $a \in M$ mit kürzestem Abstand zum Startknoten wählen
- (b) alle Nachfolger auf kürzeren Abstand über a untersuchen
- (c) falls ja
 - i. Menge M erweitern
 - ii. Entfernungsvektor (-tabelle) anpassen
 - iii. Vorgängervektor (-tabelle) anpassen

2. FIFO

- (a) Anfangsknoten ist KS (Kopf der Schlange)
- (b) alle Nachfolger auf kürzeren Abstand über KS untersuchen
- (c) falls ja
 - i. Schlange erweitern
 - ii. Entfernungsvektor (-tabelle) anpassen
 - iii. Vorgängervektor (-tabelle) anpassen

3. Tripel

- (a) mittleren Knoten j wählen (Schleife über alle Knoten des Graphen)
- (b) Verbindung von Vorgänger zu Nachfolger auf kürzeren Abstand über j untersuchen
- (c) falls ja
 - i. Entfernungsmatrix (-tabelle) anpassen
 - ii. Vorgängermatrix (-tabelle) anpassen

C. minimal spannende Bäume

1. Kruskal

- (a) Graph K wie Ausgangsgraph G aber ohne Kanten
- (b) Kanten von G nach aufsteigender Reihenfolge der Bewertung sortieren
- (c) Kanten in dieser Reihenfolge in K einfügen, sofern sich kein Kreis ergibt

2. Prim

- (a) Graph K mit nur einem Knoten aus G
- (b) Kanten von G nach aufsteigender Reihenfolge der Bewertung sortieren
- (c) Kanten in dieser Reihenfolge untersuchen: die erste Kante, die von K ausgeht und keinen Kreis ergibt, wird eingefügt, einschließlich Endknoten bis alle Knoten aus G auch in K sind

6 Netzplantechnik

Die Netzplantechnik ist eine Planungsmethode, bei der Aktivitäten nach ihrem Zeitbedarf und ihren gegenseitigen Abhängigkeiten in ihrer Reihenfolge geplant werden. Die Dauer der einzelnen Tätigkeiten ist bekannt. Außerdem ist bekannt, welche Tätigkeiten abgeschlossen sein müssen, bevor eine bestimmte weitere begonnen werden kann. Dies wird in einer *Vorgangsliste* festgehalten.

Gesucht sind ein Strukturplan, der die notwendigen Reihenfolgen graphisch darstellt, sowie ein Zeitplan, dem entnommen werden kann, wann welche Tätigkeit frühestens begonnen werden kann und wann sie spätestens abgeschlossen sein muss, wenn die mindestens notwendige Gesamtdauer des Projekts nicht verlängert werden soll. Dies wird dann im Netzplan zusammen gefasst.

Beispiele sind: Bauprojekte aller Art, Betriebsabläufe, Veranstaltungen, Lehrpläne, Softwareprojekte u.s.w.

Folgende Abkürzungen sind üblich:

- FAZ(i) - frühest möglicher Anfangszeitpunkt der Tätigkeit i
- FEZ(i) - frühest möglicher Endzeitpunkt der Tätigkeit i
- SAZ(i) - spätest möglicher Anfangszeitpunkt der Tätigkeit i, ohne das Projekt zu verzögern
- SEZ(i) - spätest möglicher Endzeitpunkt der Tätigkeit i, ohne das Projekt zu verzögern
- V(i) - Menge der Vorgänger von i, d.h. Menge der Tätigkeiten, die abgeschlossen sein müssen, bevor Tätigkeit i anfangen kann
- N(i) - Menge der Nachfolger von i, d.h. Menge der Tätigkeiten, die erst begonnen werden können, wenn i abgeschlossen ist

Beispiel: Ein Gebäude soll erweitert werden. Dabei stehen (u.a.) folgende Tätigkeiten an (Dauer in Tagen):

Tätigkeit	Beschreibung	Dauer	Vorgänger
A	Bauaushub	10	-
B	Wände und Decken	20	A
C	Dach	5	B
D	Türen und Fenster	3	B
E	Elektroinstallation	4	B
F	Boden	5	D, E
G	Streichen	2	C, E, F

Die Spalte der Vorgänger enthält die unabhängigen Vorgänger. Z.B. muss Tätigkeit A vor allen anderen Tätigkeiten ausgeführt werden. Dies wird nur bei B erwähnt, da die anderen Tätigkeiten nach B erfolgen müssen. Manchmal sind auch einige abhängige Vorgänger mit aufgeführt. Dies ändert am Gesamtablauf nichts, da die Information redundant ist.

Aus dieser Tabelle ergibt sich der *Strukturplan*:

OR-Skript-Abb/StrukturUmbau2.eps

Die Zahlen entsprechen der Dauer des Vorgangs. Die Pfeile geben die jeweiligen Nachfolger an. Zeitliche Mindestabstände könnten hier notiert werden. Im gegebenen Beispiel treten diese nicht auf.

Die Zeitplanung ist aus der Strukturplanung noch nicht ersichtlich. Hierfür müssen die Anfangs- und Endzeiten berechnet werden.

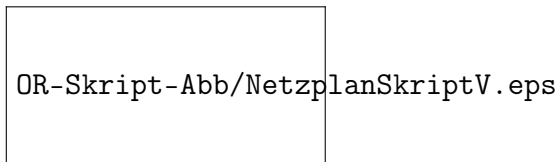
Die ausführliche Rechnung, wie sie gegebenenfalls programmiert wird, ist in den Ergänzungen Kap.?? beschrieben. Hier soll die mehr intuitive Erstellung des Netzplans betrachtet werden. Auch sie wird in die Teile „Vorwärtsrechnung“ und „Rückwärtsrechnung“ geteilt. Gestartet wird mit einem leeren Netzplan. Dieser enthält lediglich die Informationen des Strukturplans. Die einzelnen Knoten des Netzplans, die den Vorgängen entsprechen, enthalten die Informationen in folgender Form:

Vorgang i	Zeitbedarf $t(i)$
FAZ(i)	SAZ(i)
FEZ(i)	SEZ(i)
Pufferzeit $P(i)$	

Die frühesten Anfangs- und Endzeiten, $FAZ(i)$ und $FEZ(i)$, werden über die *Vorwärtsrechnung* bestimmt, die spätesten Anfangs- und Endzeiten $SAZ(i)$ und $SEZ(i)$ über die Rückwärtsrechnung.

6.1 Vorwärtsrechnung

Hier wird zunächst das Ergebnis der Vorwärtsrechnung dargestellt und anschließend besprochen:



Vor der Rechnung ist ein teilweise gefüllter Netzplan gegeben. Er enthält

1. die Informationen des Strukturplans, also Namen des Knoten, d.h. des Vorgangs sowie
2. Dauer des Vorgangs (rechts daneben) und
3. Pfeile, also die Information über Vorgänger
4. Anfangszeit 0 im ersten Knoten (mit * gekennzeichnet)

Der FEZ ergibt sich aus $FEZ = FAZ + \text{Dauer}$.

Die FAZ jedes Knotens kann aus der FEZ des Vorgängers direkt übernommen werden. Hat ein Knoten mehrere Vorgänger, muss der späteste FEZ genommen werden.

So wird Schritt für Schritt durch den gesamten Plan FAZ und FEZ bestimmt.

Damit sind die frühesten Anfangs- und Endzeiten bekannt und können für die Rückwärtsrechnung verwendet werden.

6.2 Rückwärtsrechnung

Die Voraussetzungen sind wie oben. Allerdings muss die Vorwärtsrechnung für die Initialisierung abgeschlossen sein, da $FEZ(n)$ bekannt sein muss. Über die Rückwärtsrechnung werden $SAZ(i)$ und $SEZ(i)$ bestimmt.

Das Ergebnis sieht folgendermaßen aus

OR-Skript-Abb/NetzplanSkriptR.eps

Ausgangspunkt war das Ergebnis der Vorwärtsrechnung. Und zusätzlich der SEZ(n) des letzten Knotens, hier mit * gekennzeichnet. Dieser ist gleich dem FEZ des gleichen Knotens, da sich sonst die Gesamtdauer vergrößert.

Der SAZ ergibt sich aus $SAZ = SEZ - \text{Dauer}$.

Der SEZ jedes Knotens kann aus dem SAZ des Nachfolgers direkt übernommen werden. Hat ein Knoten mehrere Nachfolger, muss der früheste SAZ genommen werden.

So wird Schritt für Schritt durch den gesamten Plan SEZ und SAZ bestimmt.

Nun können noch die Pufferzeiten eingetragen werden. Sie ergeben sich für jeden Vorgang aus $\text{Pufferzeit} = SAZ - FAZ = SEZ - FEZ$. Die Pufferzeit beschreibt die Zeit, um die ein Vorgang maximal verzögert werden darf, ohne dass das Ende des Gesamtprojekts dadurch verzögert wird. Man beachte jedoch: wird diese Pufferzeit bei einem Vorgang ausgeschöpft, dann haben die Nachfolger keine Pufferzeit mehr und müssen zu ihrem SAZ beginnen.

Das Ergebnis ist der gesuchte Netzplan.

Es gibt mindestens einen Weg vom ersten zum letzten Knoten, bei dem keine Pufferzeiten vorkommen. Kein Vorgang, der auf diesem Weg liegt, kann verzögert werden, ohne das gesamte Projekt zu verzögern. Dieser Weg heißt *kritischer Pfad*. Im oben stehenden Beispiel ist der kritische Pfad A-B-E-F-G.

7 Simulation

Dieses Kapitel kann keine Vorlesungsreihe über Simulation ersetzen. Es soll hier lediglich ein elementares Verständnis vermittelt werden.

Der bisher unwichtige Begriff des Zufalls bzw. der Zufallszahlen ist bei vielen Simulationen mathematisch zentral und soll daher als Einziges auch rechnerisch zur Sprache kommen.

Bisher wurden Situationen betrachtet, in denen klar definierte Anfangszustände zu klar berechenbaren Endsituationen führen. Simulationen spielen immer auch dort eine Rolle, wo dies nicht der Fall ist. Das kann an einem zu komplexen mathematischen Modell liegen

oder an der Tatsache, dass bei der Ausgangssituation einige Parameter zufällige Werte haben, z.B. Schwankungen in Liefermengen oder Verkaufszahlen. Um diese Schwankungen berücksichtigen zu können, braucht man geeignete mathematische Methoden zur Darstellung zufälliger Ereignisse. Z.B. kann das wiederholte Werfen eines Würfels nur simuliert werden, wenn ganze Zahlen von 1 bis 6 in zufälliger Reihenfolge ausgegeben werden. Hinzu kommt, dass hier im langfristigen Mittel jede Zahl gleich oft vorkommen muss. In anderen Situationen gibt es einen zentralen Wert, um den sich die Zahlen häufen, z.B. bei Fertigungsungenauigkeiten, die um den Sollwert streuen. In den Kapiteln ?? und ?? wird die Berechnung zweier Fälle vorgestellt. Das wichtigste Simulationsverfahren im Rahmen der Stochastik ist die Monte-Carlo-Simulation.

7.1 Monte-Carlo-Simulation

Die Monte-Carlo-Simulation wird zur Darstellung zufälliger Prozesse verwendet. Ein anderer Anwendungsbereich ergibt sich, wenn von einem Prozess nur der Endzustand bekannt ist, nicht aber die Anfangsparameter. Diese werden dann zufallsverteilt in eine große Zahl gleichartiger Rechnungen eingesetzt und überprüft, welche Kombinationen von Anfangsparametern den bekannten Endzustand ergeben.

Ähnlich wie beim Roulette werden nach einer vorgewählten Wahrscheinlichkeitsverteilung Zufallszahlen gewählt, die als Inputgrößen für weitere Berechnungen dienen. Beim Roulette ist die Situation besonders einfach: alle Zahlen haben die gleiche Wahrscheinlichkeit (gleichverteilte Zufallszahlen). Außerdem werden keine weiteren Berechnungen mehr angestellt, außer der Gewinnausschüttung.

Andere Situationen können mit der Monte-Carlo-Simulation durch geeignete Anpassung der Wahrscheinlichkeitsverteilung simuliert werden. Typischerweise dienen die Inputgrößen als statistische Anfangswerte für weitere Berechnungen.

Ein wichtiger Bereich des Finanzwesens, in dem die Monte-Carlo-Simulation eine große Rolle spielt, ist die Berechnung komplexer (nicht-europäischer) Optionen.

Eine wesentliche Eigenschaft von Zufallszahlen ergibt sich aus der Forderung, dass jede Zahl unabhängig von den bisher gezogenen Zahlen sein muss. In diesem Sinn kann ein bestimmter Algorithmus immer nur Pseudo-Zufallszahlen erzeugen. Die Verteilung einer Serie entspricht zwar echten Zufallszahlen. Ein erneutes Aufrufen des Algorithmus bei gleichen Anfangsbedingungen erzeugt aber wieder die gleiche Serie.

7.2 Gleichverteilte Zufallszahlen

Das Beispiel des Würfels erfordert eine Gleichverteilung der (ganzen) Zufallszahlen innerhalb des Intervalls $[1; 6]$. Beim Roulette werden die 37 Zahlen von 0 bis 36 gezogen. Ohne Beschränkung der Allgemeinheit werden intern reelle Zufallszahlen oft im Intervall $[0; 1]$ berechnet. Dies lässt sich leicht auf jedes beliebige Intervall umrechnen. Man spricht von *Standard-Gleichverteilung* bzw. von Standardzufallszahlen.

Midsquare-Methode

Voraussetzung: eine beliebige 4-stellige Zahl x

Berechnung: $y = x^2$; von y werden abwechselnd vorne und hinten Stellen abgestrichen, bis wieder eine 4-stellige Zahl x übrig ist. Diese ist Ausgangspunkt für die weitere Berechnung.

Am Ende hat man eine Folge von zufällig verteilten vierstelligen Zahlen, die nach einer Division durch 10000 im gewünschten Bereich liegen.

Die nächste Methode nutzt Modulo, also die Division mit Rest. Z.B.: $13 \bmod 3 = 1$, denn 3 passt 4 Mal in 13 und 1 bleibt übrig.

Kongruenzmethode von Lehner

Voraussetzung: Vier natürliche Zahlen a, b, Z_0, m

Berechnung: $Z_{k+1} = (a \cdot Z_k + b) \bmod m$ ergibt weitere Werte von Z

$S_k = \frac{Z_k}{m}$ sind Standardzufallszahlen

Bei der Auswahl der Zahlen a, b, Z_0, m müssen einige Restriktionen beachtet werden, dass keine periodischen Zahlen entstehen, z.B. $m = 2^l$ mit $30 \leq l \leq 40$ und $Z_0 \ll m$.

Ein Beispiel ist in den Ergänzungen enthalten.

7.3 Normalverteilte Zufallszahlen

Wenn eine Größe um einen Mittelwert streut, z.B. bei Fertigungstoleranzen, sind diese Werte typischerweise normalverteilt. Die Dichtefunktion der Normalverteilung lautet:

$$f(x) = \frac{1}{\sqrt{2\pi}\sigma} \cdot e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

mit dem Erwartungswert μ und der Varianz σ^2 . σ heißt *Standardabweichung*. Der Erwartungswert entspricht der Sollgröße bei Fertigungen und die Varianz ist ein Maß für die Streuung. Normalverteilte Zufallszahlen lassen sich über verschiedene Methoden erzeugen, s.[?].

Zwölferregel

Voraussetzung: k Serien zu je 12 Standardzufallszahlen $x_i(k)$

Berechnung: $N_k = \sum_{i=1}^{12} x_i(k) - 6$

N_i ist eine Folge von näherungsweise normalverteilten Zufallszahlen.

Box-Muller-Methode

Voraussetzung: k Serien zu je 2 Standardzufallszahlen $x_1(k)$ und $x_2(k)$

Berechnung: $N_k = \sqrt{-2 \cdot \ln x_1(k)} \cdot \cos(2\pi \cdot x_2(k))$

N_i ist eine Folge von normalverteilten Zufallszahlen.

Abbildung 11:

OR-Skript-Abb/Dichtefunktion.png

Abb. 11 (Aus Wikipedia, „Normalverteilung“, September 2010) zeigt die Dichtefunktion der Normalverteilung. Der Erwartungswert ist hier $\mu = 0$ und die Varianz ist $\sigma^2 = 1$.

Anhang

A Übungen

Die folgenden Übungen werden im Rahmen der Vorlesungszeit bearbeitet. In diesem Rahmen werden auch die Lösungen zur Verfügung gestellt.

Übungsblatt 1

1. Maximieren Sie die Zielfunktion $G = 5x_1 + 3x_2$ unter den Bedingungen

$$\begin{aligned}x_1 + x_2 &\leq 14 \\4x_1 + 9x_2 &\leq 81 \\x_1 &\leq 11 \\x_1, x_2 &\geq 0\end{aligned}$$

Lösen Sie die Aufgabe sowohl zeichnerisch als auch rechnerisch.

2. Die Zielfunktion G soll unter folgenden Nebenbedingungen maximiert werden:

$$\begin{aligned}2x_1 + 2x_2 &\leq 24 \\-4x_1 + 3x_2 &\leq 12 \\x_1 &\leq 10 \\x_2 &\leq 7 \\x_1, x_2 &\geq 0\end{aligned}$$

(a) $G = 2x_1 + 3x_2$

(b) $G = 2x_1 + x_2$

3. Wie 2. mit $x_2 = 7$ statt $x_2 \leq 7$. Welche der Lösungen bleibt erhalten - warum?
4. Wie 2. mit $x_2 \geq 7$ statt $x_2 \leq 7$.
5. Ein Rohstoff wird zu drei Gütern G_1 , G_2 und G_3 verarbeitet. Man benötigt für G_1 5kg/Stück, für G_2 9kg/Stück und für G_3 7kg/Stück. Benötigte Arbeitszeit in Stunden pro Stück: 5 für G_1 , 8 für G_2 und 4 für G_3 . Es stehen maximal 140

Arbeitsstunden und 190kg Rohstoff zur Verfügung. G_1 und G_2 werden an der gleichen Maschine vorbereitet. Diese steht insgesamt nur 1 Stunde zur Verfügung. Von G_1 können pro Stunde an dieser Maschine 10 Stück vorbereitet werden, von G_2 5 Stück. Gewinn pro hergestelltem Stück: 4€ für G_1 , 4€ für G_2 und 5€ für G_3 . Wie soll produziert werden, dass der Gewinn maximal wird?

Interpretieren Sie das Ergebnis.

6. Wie 5., wobei sicher gestellt werden soll, dass die genannte Maschine genau eine Stunde genutzt wird.
7. Wie 5., wobei die Maschine *mindestens* eine Stunde genutzt werden soll.
Interpretieren Sie das Ergebnis.

Übungsblatt 2

1. (vergl. Beispiel 2.1 aus [?]) Ein Betrieb fertigt zwei Artikel A_1 und A_2 mit zwei Maschinen M_1 und M_2 und einer Montagegruppe. Der jeweilige Gewinn beträgt $g_1 = 5GE$ und $g_2 = 8GE$. Während beide Maschinen jeweils 24 Stunden pro Tag zur Verfügung stehen, stellt die Montagegruppe insgesamt 36 Arbeitsstunden zur Verfügung. Für die Herstellung von A_1 werden 5 Stunden auf M_1 und eine Stunde auf M_2 sowie 6 Montagestunden gebraucht. Für die Herstellung von A_2 werden 2 Stunden auf M_1 und 5 Stunden auf M_2 sowie 6 Montagestunden gebraucht.
 - (a) Stellen Sie ein mathematisches Modell auf.
 - (b) Erstellen Sie das entsprechende Simplextableau (Ausgangstableau).
 - (c) Lösen Sie das Problem mit Hilfe des Simplexalgorithmus.
 - (d) Gibt es noch ungenutzte (Maschinen- oder Arbeitszeit-) Kapazitäten?
2. (vergl. Beispiel 2.4 aus [?]) Der in Aufg. 1 genannte Betrieb erweitert sein Sortiment um einen Artikel A_3 , dessen Fertigung 3 Stunden auf Maschine M_1 benötigt. Montagezeit fällt keine an. Ein Artikel A_3 kann nur zusammen mit je einem Artikel A_1 verkauft werden. Genau ein Artikel A_1 wird ohne A_3 verkauft. Ein Artikel A_3 trägt mit einer GE zum Gewinn bei.
 - (a) Erweitern Sie das mathematische Modell von Aufg. 1 geeignet.
 - (b) Lösen Sie das Problem mit Hilfe des Simplexalgorithmus.
 - (c) Ist die Auslastung der Kapazitäten nun besser?

- (d) Wie wäre die Auslastung ohne die Bindung von A_3 an A_1 ?
3. Bestimmen Sie das zu Aufg. 1 duale Problem und lösen Sie es mit dem Simplexalgorithmus.
4. Machen Sie eine Sensitivitätsanalyse von Aufg. 1. Verwenden Sie hierfür die zeichnerische Lösung. Es reicht, die Zielfunktionskoeffizienten sowie die rechten Seiten der Bedingungen zu untersuchen.
Untersuchen Sie auch den Koeffizienten c in der zweiten Gleichung.

$$M_2 - c * x_1 + 5x_2$$

ausgehend von $c=1$. D.h. in welchem Bereich darf sich c ändern, so dass die Lösung (Schnitt der gleichen Geraden) erhalten bleibt?

Übungsblatt 3

1. Folgende Angebots- und Nachfragemengen sollen nach einem kostenminimalen Transportplan ausgeliefert werden:

Lager	Liefermenge a_i	Filiale	Bedarfsmenge b_j					
L_1	5	F_1	6	Kosten	F_1	F_2	F_3	F_4
L_2	8	F_2	5	L_1	2	9	3	5
L_3	7	F_3	4	L_2	8	7	8	7
		F_4	5	L_3	4	10	6	11
Summe	20	Summe	20					

- (a) Bestimmen Sie eine Anfangslösung mit der Northwest-Ecken-Regel.
- (b) Bestimmen Sie eine Anfangslösung mit dem Rangfolgeverfahren.
- (c) Ermitteln Sie eine optimale Lösung mit der Stepping-Stone-Methode. Gehen Sie von der Lösung (a) aus.
- (d) Ermitteln Sie eine optimale Lösung mit der MODI-Methode. Gehen Sie von der Lösung (b) aus.
2. Sechs Fahrzeugparks P_i stellen ihre Fahrzeuge vier Baustellen B_j zur Verfügung. Die Entfernungstabelle gibt an, wie groß jeweils die Entfernung von Park P_i zu Baustelle B_j ist.

Park	Anzahl der Fahrzeuge	Baustelle	Bedarf
1	13	1	23
2	7	2	14
3	9	3	17
4	12	4	11
5	6		
6	18		
Summe	65		65

Entf.	1	2	3	4
1	4	1	6	13
2	7	3	4	12
3	3	4	6	8
4	9	10	7	9
5	8	13	5	8
6	12	18	4	7

Verteilen Sie Fahrzeuge so, dass insgesamt eine möglichst geringe Gesamtweglänge gefahren werden muss. Verwenden Sie für die Ausgangslösung das Rangfolgeverfahren. Optimieren Sie die Lösung mit dem MODI-Verfahren.

3. Lösen Sie folgendes Zuordnungsproblem. Für drei gleich ausgebildete Personen stehen drei Stellen zur Verfügung. Der Weg vom Wohnort zur möglichen Arbeitsstelle ist durch folgende Entfernungsmatrix gegeben:

Entfernungsmatrix	A_1	A_2	A_3
P_1	5	4	7
P_2	5	2	3
P_3	4	6	5

Welche Person müsste welche Arbeitsstelle bekommen, so dass der insgesamt gefahrene Weg zwischen Wohnung und Arbeitsstelle minimal ist?

Übungsblatt 4

1. (vergl. Beispiel 4.1 aus [?]) Auf zwei Maschinengattungen M_1 und M_2 werden zwei Artikel A_1 und A_2 gefertigt. Es stehen mehrere Montagekräfte zur Verfügung. Beschränkungen (Maschinenzeit in Stunden/Stück) und Gewinn bei jedem Artikel entsprechen folgender Tabelle:

	A_1	A_2	Kapazität
M_1	5	2	24h/Tag
M_2	1	5	24h/Tag
Montagekräfte	6	6	36h/Tag
Gewinn	5GE	8GE	

- (a) Erstellen Sie ein mathematisches Modell zur Bestimmung des Gewinnmaximums unter Berücksichtigung der Kapazitätsgrenzen.
- (b) Stellen Sie das Problem grafisch dar.
- (c) Pro Tag soll immer eine ganzzahlige Anzahl von Artikeln hergestellt werden. Lösen Sie dieses Problem grafisch und rechnerisch mit dem Branch-and-Bound-Verfahren.
2. Lösen Sie Aufgabe 7 von Blatt 1 für ganzzahlige Werte.
3. Lösen Sie das Rucksackproblem von Kap. 4.2 mit folgender Verzweigungsregel: Verzweigt wird ein unzulässiges Problem, indem in einem Zweig $x_i = 0$ und im anderen Zweig $x_i = 1$ gesetzt wird. i soll dabei möglichst klein sein, zu Beginn also 1. Da die Verzweigungsentscheidung weiter vererbt wird, wird i immer größer. Diese Verzweigungsregel wird bei Gert Heinrich verwendet.
4. Das bei Gert Heinrich verwendete Beispiel führt auf folgende Tabelle:

Kosten	5000	3000	6000	4500	6000	5000
Nutzen	40	10	30	30	15	20
Kosten/Nutzen						
Gegenstand Nr.						

Lösen Sie dieses Problem mit dem im Skript besprochenen Verfahren. Der Nutzen muss mindestens 90 betragen. Die Kosten sollen minimiert werden. Beachten Sie, dass es sich hier um ein Minimierungsproblem handelt. Die Schritte müssen also sinngemäß angepasst werden.

Übungsblatt 5

1. Über folgendes Verkehrsnetz werden die fünf Firmen 2-6 von Lager 1 beliefert. Bestimmen Sie mit dem Dijkstra-Algorithmus und mit dem FIFO-Algorithmus die jeweils kürzesten Wege.

Übungen_skript/Abb/V_1.eps

Was ändert sich, wenn das Lager statt in Ort 1 im Ort 2 steht und entsprechend die Firma in 1?

2. In obigem Verkehrsnetz sollen die kürzesten Wege von jedem Ort zu jedem anderen Ort bestimmt werden. Verwenden Sie den Tripel-Algorithmus.
3. In obigem Verkehrsnetz sollen die Richtungen ignoriert werden. Bestimmen Sie die kürzesten Wege von jedem Ort zu jedem anderen Ort. Verwenden Sie den Tripel-Algorithmus.
4. In obigem Graphen sollen die Richtungen ignoriert werden. Bestimmen Sie den minimal spannenden Baum mit dem Kruskal-Algorithmus und mit dem Prim-Algorithmus und beginnen Sie bei diesem mit Knoten 1.

Übungsblatt 6

1. (aus [?]) Eine (etwas altertümliche) Liste für die Erstellung einer EDV-Anlage könnte so aussehen:

Vorgang	Beschreibung	Dauer	Vorgänger
A	Entwurf	10	-
B	Fertigung der Zentraleinheit	5	A
C	Bereitstellung der Ein- und Ausgabegeräte	2	A
D	Erstellung der Grundsatzprogramme	4	A
E	Erstellung des Prüfprogramms	4	D
F	Erstellung der Kundenprogramme	3	D
G	Funktionsprüfung	2	B, C, E
H	Bereitstellung der Anschlussgeräte	5	C
I	Auslieferung, Installation	1	F, G, H

Erstellen Sie einen Netzplan und bestimmen Sie den kritischen Pfad.

2. Ergänzen Sie den folgenden unvollständig ausgefüllten Netzplan und erstellen Sie eine Tabelle mit den Spalten „Vorgang“, „Dauer“ und „Vorgänger“. Kennzeichnen Sie den kritischen Pfad.

OR-Skript-Abb/NetzplanAufg.eps

Übungsblatt 7

1. (aus [?], andere Zahlen) Ein verschuldeter Fußballverein muss Spieler im Wert von 10 Mio. € verkaufen. Die Spieler haben diese Marktwerte im Mio.€: 4, 3, 1, 2, 2, 5. Es muss mit folgenden Zuschauerverlusten bei jedem Verkauf gerechnet werden: 500, 300, 200, 300, 280, 600. Minimieren Sie den Schaden unter Einhaltung des Mindesterloßes.

2. Aus drei verschiedenen Produkten A, B und C soll eine Futtermittelmischung hergestellt werden. Die Kosten pro ME betragen 10€, 20€ und 15€. In den Produkten sind die Nährstoffe N_1 und N_2 enthalten. A enthält 2 bzw. 3 ME. B enthält 4 bzw. 2 ME. C enthält 1 bzw. 5 ME. Die Mindestmenge von N_1 beträgt 30 ME und von N_2 50 ME. Welche Mischung aus A, B und C ist herzustellen, um die Kosten zu minimieren?
Stellen Sie ein mathematisches Modell auf und bestimmen Sie das Ausgangstableau.

3. (aus [?]) Ein Landwirt will auf maximal 40 ha Boden Zuckerrüben und Weizen anbauen. Er kann 2400€ und 312 Arbeitstage einsetzen. Die Anbaukosten für Rüben betragen 40€/ha und für Weizen 120€/ha. Für Rüben werden 7 und für Weizen 12 Arbeitstage/ha benötigt. Der Gewinn beträgt bei Rüben 100€/ha und bei Weizen 250€/ha. Maximieren Sie den Gewinn.
Stellen Sie das mathematische Modell auf, bestimmen Sie das Ausgangstableau und berechnen Sie die Zielfunktionszeile sowie die rechte Seite der ersten verbesserten Lösung von Hand. Bestimmen Sie die Optimallösung mit Hilfe eines Simplexrechners und interpretieren Sie die Werte der rechten Seite und der Zielfunktionszeile.
Bestimmen Sie die Lösung zeichnerisch.

4. Ein Optimierungsproblem mit der Zielfunktion $G = 5x_1 + 3x_2 \rightarrow \text{MAX}$ mit den Bedingungen $x_1 \leq 11$ und $4x_1 + 9x_2 \leq 71$ hat sein Optimum bei $x_1 = 11$ und $x_2 = 3$ (ähnlich Aufg. 1 von Blatt 1). In welchem Bereich darf der Zielfunktionskoeffizient $c_2 = 3$ variieren, so dass die optimale Lösung erhalten bleibt?

5. Vier Fabriken erzeugen ein Produkt, das von 6 Abnehmern gebraucht wird. Die Transportkosten pro Einheit sowie Kapazität und Bedarf betragen:

	A_1	A_2	A_3	A_4	A_5	A_6
F_1	3	8	3	9	8	12
F_2	1	3	4	10	13	8
F_3	6	4	6	5	6	4
F_4	19	12	8	9	7	7

Kapazität	F_1	F_2	F_3	F_4		
	23	14	17	11		
Bedarf	A_1	A_2	A_3	A_4	A_5	A_6
	13	7	9	12	6	18

Bestimmen Sie den optimalen Transportplan mit einer geeigneten Methode Ihrer Wahl.

6. Diese und die darauf aufbauende nächste Aufgabe wurden in dieser Form neu aufgenommen. Die Lösungen können noch fehlerhaft sein.

Eine Versicherung bietet eine flexible Kombinationsversicherung an, d.h. aus den Leistungen kann eine beliebige Kombination (jeweils ein Mal) gewählt werden. Einzige Voraussetzung ist, dass mindestens zwei der angebotenen Leistungen in Anspruch genommen werden. Monatliche Beiträge sowie Maximalwert der Versicherung entsprechen folgender Tabelle (in €):

	Beitrag	Deckungssumme	Wahrsch.	erwarteter Schaden
priv. Haftpflicht	10	10 Mio.	10%	500€
Unfall	8	300000	1%	80000€
Rechtsschutz	10	unbegrenzt	10%	8000€
Gebäude	40	400000	0,25%	500000€
Hausrat	3	45000	0,3%	20000€

Zusätzlich sind die persönliche Einschätzung einer Eintrittswahrscheinlichkeit und der zu erwartende Schaden angegeben.

Der Versicherungsnehmer plant, einen monatlichen Beitrag von (maximal) 40€ einzusetzen. Er möchte seinen Nutzen maximieren und berechnet diesen zu

$$\text{Nutzen} = \text{Eintrittswahrscheinlichkeit} * \text{Schaden}$$

- An welcher Stelle muss die Nutzenberechnung modifiziert werden, um sinnvoll zu sein?
- Mit welcher Optimierungsmethode würden Sie die für den Versicherungsnehmer optimale Zusammensetzung bestimmen?
- Stellen Sie geeignete Anfangstabellen auf. Berücksichtigen Sie das Ergebnis von Teilaufgabe (a).
- Optimieren Sie die Ausgangssituation ein Mal. Erklären Sie kurz die verwendete Methode.

7. Lösen Sie die vorhergehende Aufgabe mit folgenden Änderungen:

- Die einzelnen Teilversicherungen können in beliebigem Umfang abgeschlossen werden; d.h. es ist z.B. möglich, 1,5 Haftpflichtversicherungen abzuschließen. Damit wäre der monatliche Einsatz 15€ und die Deckungssumme würde sich entsprechend auf 15Mio.€ erhöhen.
- Die Versicherungsgesellschaft hat jedoch die maximalen Deckungssummen begrenzt (außer Rechtsschutz):

	Beitrag	Deckungss./Max.	Wahrsch.	erwarteter Schaden
priv. Haftpflicht	10	10 Mio. / 20Mio.	10%	500€
Unfall	8	300000 / 1Mio.	1%	80000€
Rechtsschutz	10	unbegrenzt	10%	8000€
Gebäude	40	400000 / 1Mio.	0,25%	500000€
Hausrat	3	45000 / 100000	0,3%	20000€

- außerdem gilt noch folgende Begrenzungen: Gebäude+Hausrat maximal 1Mio.; d.h., wenn z.B. der Gebäudeschutz vollständig ausgenutzt wird, kann keine Hausratsversicherung mehr abgeschlossen werden.

8. Diese Aufgabe ist gegenüber früher verändert. Die Lösung kann noch fehlerhaft sein.

Eine Transportfirma hat in fünf Städten je einen leeren LKW übrig und benötigt diese in fünf anderen Städten. Wie muss man die LKWs dirigieren, dass die gefahrene Gesamtstrecke möglichst minimal wird? Die Entfernung der Orte voneinander beträgt:

Ausgangsorte (unten) Bestimmungsorte (rechts)	1	2	3	4	5
1	6	3	11	13	16
2	2	4	17	9	7
3	12	9	4	8	6
4	9	11	9	7	14
5	6	8	9	3	3

26	Anfangslösung				
	0	1			
	1				
	0		1		
	0				1
	0			1	

Beginnen Sie mit der gegebenen (ungeschickten) Anfangslösung.

Beachten Sie: Ausgangs- und Bestimmungsorte sind jeweils von 1 bis 5 durchnummeriert. Die Orte unterscheiden sich.

9. Bestimmen Sie die kürzesten Wege von Knoten 1 zu allen anderen Knoten in folgendem Graph:

Uebungen_skript/Abb/Baum_Blatt7_aufg7.eps

10. Vernachlässigen Sie die Richtungen im obigen Graphen und bestimmen Sie den minimal spannenden Baum. Hinweis: es gibt mehrere Lösungen mit der gleichen minimalen Gesamtgewichtung.
11. Sie haben eine Idee für ein neues Produkt. Bis zur Markteinführung müssen folgende Schritte durchlaufen werden. Stellen Sie einen Netzplan auf und gehen Sie von einer möglichst schnellen Markteinführung aus. Bestimmen Sie den kritischen Pfad.

Vorgang	Tätigkeit	Zeit	Vorgänger
A	Machbarkeitsstudie	5	-
B	Produktentwicklung	5	A
C	Fertigung	10	B
D	Markteinführung	20	C, G
E	Planung der Kommunikationsstrategie	3	A
F	Erstellen der Pressemappen	5	E
G	Informieren der Presse	5	F

Literatur

- [1] Domschke, Drexl
Einführung in Operations Research; Springer
Grundlegendes Lehrbuch, das die meisten Inhalte der Vorlesung abdeckt.
- [2] Domschke, Drexl, Klein, Scholl, Voss
Übungen und Fallbeispiele zum Operations Research; Springer
- [3] Ellinger, Beuermann, Leisten
Operations Research; Eine Einführung; Springer
Dieses Buch deckt nicht den gesamten Umfang der Vorlesung ab, die Erklärungen sind aber besonders anschaulich.
- [4] Heinrich, Gert
Operations Research; Verlag Oldenburg
Die meisten Vorlesungsinhalte sind enthalten. Das Lehrbuch zeichnet sich durch vollständig durchgerechnete Beispiele aus, an denen die einzelnen Schritte sehr gut nachvollzogen werden können.
- [5] Neumann, Klaus; Morlock, Martin
Operations Research; Verlag Hauser
Umfangreiches Lehrbuch mit vielen guten Erklärungen. Mathematisch exakt und damit in diesem Bereich teilweise etwas abstrakt. Enthält auch Beweise.
- [6] Zimmermann
Operations Research - Quantitative Methoden zur Entscheidungsvorbereitung, Verlag Oldenbourg
In einigen Kapiteln eine empfehlenswerte Ergänzung zu [?]
- [7] Gohout
Operations Research; Verlag Oldenburg
Kurzes Lehrbuch, das besonders für die Kapitel ?? bis ?? sehr zu empfehlen ist.
- [8] Werners
Grundlagen des Operations Research; Springer