Operations Research

Jochen Werner

Wintersemester 2000/2001

Inhaltsverzeichnis

Kapitel 1

Einführung

Stellt man der Suchmaschine Yahoo[1] die Frage "Was ist Operations Research" so erhält man 12 Web-Seiten, auf denen diese Frage ebenfalls gestellt wurde und auf denen man vielleicht auch eine Antwort findet. Als Antwort erhält man z. B. (Universität Ulm, Fakultät für Mathematik und Wirtschaftswissenschaften)

- Gegenstand des modernen Operations Research sind Verfahren, die der Strukturierung und quantitativen Beschreibung komplexer Entscheidungssituationen dienen, sowie Algorithmen zur Lösung der so gewonnenen Entscheidungsmodelle auf dem Computer. Letzten Endes erlaubt die Aufstellung geeigneter, in der Sprache der Mathematik formulierter Modelle erst die Berechnung optimaler Entscheidungen. Dabei haben sich die folgenden Klassen von Entscheidungsmodellen bzw. von Algorithmen als so leistungsfähig erwiesen, dass sie heute zu jeder Ausbildung in Operations Research zählen:

  - Die heute als klassisch anzusehende lineare Optimierung, deren Grundlage unabhängig voneinander in den USA durch den Kreis um Dantzig und Koopmans sowie in Russland durch Kantorovich entwickelt wurden; ferner als Verallgemeinerung davon nichtlineare Optimierungsmodelle.
  - Aufgaben der Verkehrs- oder Terminplanung, die mit Hilfe graphentheoretischer Methoden und insbesondere mit kombinatorischen Optimierungsalgorithmen gelöst werden können.
  - Modelle, bei denen die Dynamik (Ablauf in der Zeit) und die Stochastik wesentlich zum Erklärungsgehalt geh^ren und die Konsequenzen von Entscheidungen bestimmen: Stochastische dynamische Optimierungsmodelle, Warteschlangensysteme, Lagerhaltungs- und Investitionsmodelle. Diese Modelle hängen oft eng mit Problemen der Zuverlässigkeitstheorie, der Erneuerungstheorie und der Qualitätskontrolle zusammen.

Das klingt ganz vernünftig, wobei wir auf den letzten Punkt mit seinen stochastischen Elementen nicht eingehen werden. Als Antwort erhält man auch (Universität Zürich):

[1] Ich habe diese Vorlesung im WS 2000/2001 gehalten und im März 2013 sehr geringfügig überarbeitet. An einigen Stellen erkennt man, dass einige Zeit vergangen ist.

- Operations Research ist ein Teilgebiet der Wirtschaftswissenschaften mit dem Hauptziel, Methoden aus der Mathematik auf wirtschaftliche Probleme anzuwenden. Operations Research ist eine theoretische Disziplin. Sie beschäftigt sich vor allem mit Anwendungen, die aufgrund der ungeheuren Komplexität einer Problemsituation eine abstrakte mathematische Betrachtung erfordern.

Auf die Frage "What is Operarations Research" an Google erhält man (99 700 Antworten) z. B. (Stanford, Einleitung eines Buches über Operations Research):

- Operations Research is concerned with optimal decision making in, and modeling of, deterministic and probabilistic systems that originate from real life. These applications, which occur in government, business, engineering, economics, and the natural and social sciences, are characterized largely by the need to allocate limited resources. In these situations, considerable insight can be obtained from scientific analysis such as that provided by operations research. The contribution from the operations research approach stems primarily from:

  Structuring the real-life situation into a mathematical model, abstracting the essential elements so that a solution relevant to the decision maker's objectives can be sought. This involves looking at the problem in the context of the entire system. Exploring the structure of such solutions and developing systematic procedures for obtaining them. Developing a solution, including the mathematical theory, if necessary, that yields an optimal value of the system measure of desirability (or possibly comparing alternative courses of action by evaluating their measure of desirability).

Oder auch:

- As the name implies, operations research is concerned with "research on the operations of organizational structures". Classical examples of such operations are the assignment of flight crews for an airlines daily operations, or the planning, coordination, and scheduling of an integrated production activity such as the production of the Boeing 777. The goal in operations research is to first model and then formulate optimal decision policies for complex organizational structures arising in real life.

Man findet im Netz Meinungen wie:

- For me, a big appeal of OR is that it can be very theoretical and mathematical, but it can also be very practical. This allows some very abstract thinking, but it is always firmly tied to reality.

oder auch: Antwort:

- Oh, that is a good question. Let me explain it in simple terms: Operations Research is a scientific approach to solve problems - mainly of economic nature - in order to prepare decisions.

Über die erforderlichen Qualifikationen liest man weiter:

- As the examples and applications of the theoretical concepts are the major part of the studies and the proofs are less important, the mathematics of the preliminary diploma should be sufficient. You should also be able to think in logic abstract terms. Extensive examples are naturally solved by computer, so it is quite important that you know the basics of a computer. Experience with programming is not essential, but it is useful to know how to switch the computer on and off.

Mit dem Hinweis auf `http://mat.gsia.cmu.edu/index.html` (Michael Trick's Operations Research Page) wollen wir die Suche im Internet zunächst beenden.

Wir werden (natürlich) Operations Research als eine durch die Wirtschaftswissenschaften motivierte mathematische Disziplin ansehen, und nicht etwa als ein Teilgebiet der Wirtschaftswissenschaften. Interessant ist es, in das Inhaltsverzeichnis eines Buches über Operations Research für Studierende der Wirtschaftswissenschaften zu schauen. Wir geben das Inhaltsverzeichnis (genauer die Kapitel) des offenbar weitverbreiteten Buches von W. Domschke, A. Drexl (1998)[2] an.

1. Einführung.
2. Lineare Optimierung.
3. Graphentheorie.
4. Lineare Optimierungsprobleme mit spezieller Struktur.
5. Netzplantechnik.
6. Ganzzahlige und kombinatorische Optimierung.
7. Dynamische Optimierung.
8. Nichtlineare Optimierung.
9. Warteschlangentheorie.
10. Simulation.

Dagegen wendet sich das (mathematisch orientiertere) Buch von K. Neumann, M. Morlock (1992)[3] an Wirtschaftsingenieure, Wirtschaftsmathematiker und Informatiker. Hier sind die Kapitelüberschriften (nach einer Einführung in Kapitel 0):

1. Lineare Optimierung.
2. Graphen und Netzwerke.
3. Ganzzahlige und kombinatorische Optimierung.
4. Nichtlineare Optimierung.

[2] W. Domschke, A. Drexl (1998) *Einführung in Operations Research. Vierte, verbesserte Auflage*. Springer, Berlin-Heidelberg-New York.

[3] K. Neumann, M. Morlock (1992) *Operations Research*. Carl Hanser Verlag, München-Wien.

5. Dynamische und stochastische Modelle und Methoden.

In den *Handbooks in Operations Research and Management Science* (Editors: G. L. Nemhause, A. H. G. Rinnoy Kan) (1989) ist Volume 1 der Optimierung gewidmet. Wir zitieren aus dem Beginn des Vorworts: "Not surprisingly, the first volume in this series is devoted to the topic of *Optimization*. No other class of techniques is so central to operations research and management science, both as a solution tool and as a modeling device. To translate a problem into a mathematical optimization model is charachteristic of the discipline, located as it is at the crossroads of economocs, mathematics and engineering".

Nun haben wir genug Material darüber gesammelt, was Operations Research ist. Wir werden eine *mathematische* Vorlesung halten, welche im Aufbau einer wirtschaftswissenschaftlichen Vorlesung über Operations Research folgen wird. Der Unterschied zu einer (mathematischen) Vorlesung über Optimierung besteht vor allem darin, dass wir uns auf lineare und kombinatorische Optimierungsaufgaben und vor allem Optimierungsprobleme auf Graphen oder Netzwerken konzentrieren werden. Eine eher untergeordnete Rolle werden daher unrestringierte und nichtlineare Optimierungsaufgaben (mit Ausnahme von quadratischen Optimierungsproblemen, die z. B. bei Portfolio Selection Problemen auftreten) spielen. Bei der "normalen" linearen Optimierung werden wir uns relativ kurz fassen und die wesentlichen Ergebnisse der Dualitätstheorie fast ohne Beweise zitieren. Auch auf das (revidierte) Simplexverfahren soll nur relativ kurz eingegangen werden, weil davon ausgegangen wird, dass entsprechende Vorkenntnisse aus anderen Veranstaltungen her bekannt sind. Auch auf die vielen Beispiele der linearen Optimierung (Produkrionsplanungsproblem, Diätproblem, Mischungsprobleme) wollen wir praktisch nicht eingehen (was natürlich in einer wirtschaftswissenschaftlichen Vorlesung kaum denkbar wäre).

Wenigstens ein Beispiel wollen wir in dieser Einführung bringen. Da es sich hierbei um die Maximierung eines Ertrages handelt, kann es im weiteren Sinne dem Operations Research zugeordnet werden. Es stammt aus der Landwirtschaft, die entsprechende Problematik ist an mich von Herrn Prof. Dr. R. Rauber (Institut für Pflanzenbau und Pflanzenzüchtung der Universität Göttingen) herangetragen worden. Bei der Problembeschreibung folge ich seinen Aufzeichnungen.

Beispiel: Es werden Gemenge (Mischbestände) aus Erbsen und Hafer angebaut. Die Gemenge sind deshalb interessant, weil sie oft höhere Erträge hervorbringen als das Mittel der Reinbestände. "Gedreht" werden kann bei Feldversuchen an der Anzahl d_e bzw. d_h der Erbsen- bzw. Haferpflanzen pro m^2 (bzw. genauer der Saatkörner), gemessen werden die Erträge (in Gramm) w_e bzw. w_h der Erbsen bzw. des Hafers, genauer das Gewicht einer Erbsen- bzw. Haferpflanze. Zu maximieren ist der Gesamtertrag $Y_{eh} := w_e d_e + w_h d_h$. Im einfachsten Modell (sogenanntes hyperbolisches Modell) geht man davon aus, dass der reziproke Ertrag linear von d_e und d_h abhängt, dass also ohne Beobachtungsfehler

$$w_e^{-1} = b_0 + b_1 d_e + b_2 d_h, \qquad w_h^{-1} = c_0 + c_1 d_h + c_2 d_e.$$

Es wird davon ausgegangen, dass die entsprechenden Konstanten b_0, b_1, b_2 und c_0, c_1, c_2 bekannt sind, z. B. durch eine lineare Ausgleichsrechnung nach einer hinreichend großen

Zahl von Feldversuchen. Den Gesamtertrag zu maximieren führt auf die unrestringierte Optimierungsaufgabe

$$\text{(P)} \qquad \text{Maximiere} \quad Y_{eh}(d_e, d_h) := \frac{d_e}{b_0 + b_1 d_e + b_2 d_h} + \frac{d_h}{c_0 + c_1 d_h + c_2 d_e},$$

wobei allerdings nur positive (d_e, d_h) sinnvoll sind. In einer Lösung verschwindet der Gradient (notwendige Optimalitätsbedingung erster Ordnung), wir versuchen das entsprechende nichtlineare Gleichungssystem von zwei Gleichungen in zwei Unbekannten zu lösen. Ausführlich geschrieben lautet es:

$$(*) \qquad \begin{cases} \dfrac{\partial Y_{eh}}{\partial d_e}(d_e, d_h) &= \dfrac{b_0 + b_2 d_h}{(b_0 + b_1 d_e + b_2 d_h)^2} - \dfrac{c_2 d_h}{(c_0 + c_1 d_h + c_2 d_e)^2} &= 0, \\ \dfrac{\partial Y_{eh}}{\partial d_h}(d_e, d_h) &= \dfrac{c_0 + c_2 d_e}{(c_0 + c_1 d_h + c_2 d_e)^2} - \dfrac{b_2 d_e}{(b_0 + b_1 d_e + b_2 d_h)^2} &= 0. \end{cases}$$

Zur Lösung benutzen wir das mathematische Anwendersystem *Mathematica*. Wir geben ein:

```
yeh[de_,dh_]:=de/(b0+b1*de+b2*dh)+dh/(c0+c1*dh+c2*de);
f1[de_,dh_]:=D[yeh[de,dh],de];
f2[de_,dh_]:=D[yeh[de,dh],dh];
loesung=Solve[{f1[de,dh]==0,f2[de,dh]==0},{de,dh}];
Simplify[%]
```

und erhalten: Ist

$$d := -b_1^2 b_2 c_0^2 + 2 b_0 b_1 b_2 c_0 c_2 + c_2 [b_2^2 c_0^2 + b_0^2 c_1^2 - b_0 b_2 (2 c_0 c_1 + b_0 c_2)] \neq 0,$$

so ist

$$d_e^* := -\frac{c_0 (b_2 c_0 - b_0 c_1)^2}{d}, \qquad d_h^* := \frac{b_0 (b_1 c_0 - b_0 c_2)^2}{d}$$

eine Lösung des nichtlinearen Gleichungssystems $(*)$. Nun ist (d_e^*, d_h^*) zunächst einmal als Lösung von $\nabla Y_{eh}(d_e, d_h) = 0$ nur eine kritische Lösung von (P). Um nachzuprüfen, ob es sich hier wenigstens um eine lokale Lösung von (P) handelt, muss noch die Hessesche $\nabla^2 Y_{eh}(d_e^*, d_h^*)$ untersucht werden. Ist diese negativ definit, so liegt in (d_e^*, d_h^*) ein lokales Maximum (hinreichende Optimalitätbedingung zweiter Ordnung). Ist sie dagegen nicht negativ semidefinit, so liegt in (d_e^*, d_h^*) kein lokales Maximum (notwendige Optimalitätbedingung zweiter Ordnung). Mit Hilfe der weiteren Befehle

```
h11[de_,dh_]:=D[yeh[de,dh],{de,2}];
h12[de_,dh_]:=D[yeh[de,dh],de,dh];
h22[de_,dh_]:=D[yeh[de,dh],{dh,2}];
g11=h11[de,dh]/.loesung;
g22=h22[de,dh]/.loesung;
g12=h12[de,dh]/.loesung;
Simplify[g11[[1]]*g22[[1]]-g12[[1]]^2]
```

berechnen wir

$$\det(\nabla^2 Y_{eh}(d_e^*, d_h^*)) = -\frac{d^6}{(b_2c_0 - b_0c_1)^8(b_1c_0 - b_0c_2)^8}.$$

Hieraus schließen wir, dass die Hessesche $\nabla^2 Y_{eh}(d_e^*, d_h^*)$ (bis auf Ausnahmefälle) einen positiven und einen negativen Eigenwert besitzt und daher die Funktion Y_{eh} auf $\mathbb{R}^2$ oder auch auf dem positiven Orthanten $\mathbb{R}^2_+$ weder ein lokales Maximum noch ein lokales Minimum besitzen kann. Sinnvoller ist es daher, positive $a_e < b_e$ und $a_h < b_h$ vorzugeben und statt (P) die Aufgabe zu betrachten, $Y_{eh}(d_e, d_h)$ unter der Nebenbedingung $(d_e, d_h) \in [a_e, b_e] \times [a_h, b_h]$ zu maximieren. Eine solche Aufgabe besitzt natürlich eine Lösung. Klar ist ferner, dass wenigstens eine Restriktion aktiv sein muss (denn sonst hätte man wieder eine im wesentlichen unrestringierte Optimierungsaufgabe).

Ein Zahlenbeispiel wollen wir betrachten. Es sei nämlich

$$b_0 = -0.001, \qquad b_1 = 0.002, \qquad b_2 = 0.0003$$

und

$$c_0 = 0.024, \qquad c_1 = 0.0024, \qquad c_2 = 0.0038.$$

Als Lösung von $\nabla Y_{eh}(d_e, d_h) = 0$ berechnen wir

$$d_e^* = 4.86371, \qquad d_h^* = 5.90029,$$

ferner ist $Y_{eh}(d_e^*, d_h^*) = 567.487$. Als Hessesche von Y_{eh} in (d_e^*, d_h^*) berechnen wir

$$\nabla^2 Y_{eh}(d_e^*, d_h^*) = \begin{pmatrix} -1.72518 & 1.73076 \\ 1.73076 & -0.365251 \end{pmatrix}.$$

Diese Matrix besitzt die Eigenwerte

$$\lambda_1 = -2.90476, \qquad \lambda_2 = 0.814327,$$

wie oben schon allgemein hergeleitet wurde ist sie also nicht positv semidefinit oder sogar positiv definit, in (d_e^*, d_h^*) liegt also kein unrestringiertes Extremum von Y_{eh}. In Abbildung 1.1 geben wir einen Plot von Y_{eh} auf $[4, 6] \times [5, 7]$ und $[1, 7] \times [1, 8]$ an. Da die Diagonalelemente von $\nabla^2 Y_{eh}(d_e^*, d_h^*)$ aber negativ sind, besitzt $Y_{eh}(\cdot, d_h^*)$ bzw. $Y_{eh}(d_e^*, \cdot)$ ein lokales Maximum in d_e^* bzw. d_h^*. In Abbildung 1.2 wird dies verdeutlicht. Wenn die Funktionen $Y_{eh}(\cdot, d_h^*)$ bzw. $Y_{eh}(d_e^*, \cdot)$ in d_e^* bzw. d_h^* ein Maximum besitzt, so bedeutet das natürlich nicht, dass auch Y_{eh} als Funktion in zwei Variablen in (d_e^*, d_h^*) ein Maximum besitzt. In Abbildung 1.3 tragen wir die Funktion $f(d) := Y_{eh}(d_e^* + d, d_h^* + d)$ auf dem Intervall $[-2, 2]$ auf. Wir erkennen, dass f in $d^* = 0$ sogar ein zumindestens lokales Minimum besitzt, da $f'(0) = 0$ und $f''(0) > 0$.

Beim sogenannten *parabolischen Modell* wird davon ausgegangen, dass die reziproken Erträge quadratisch in (d_e, d_h) sind, also

$$\begin{aligned} w_e^{-1} &= b_0 + b_1 d_e + b_2 d_h + b_3 d_e^2 + b_4 d_h^2 + b_5 d_e d_h, \\ w_h^{-1} &= c_0 + c_1 d_h + c_2 d_e + c_3 d_h^2 + c_4 d_e^2 + c_5 d_h d_e \end{aligned}$$

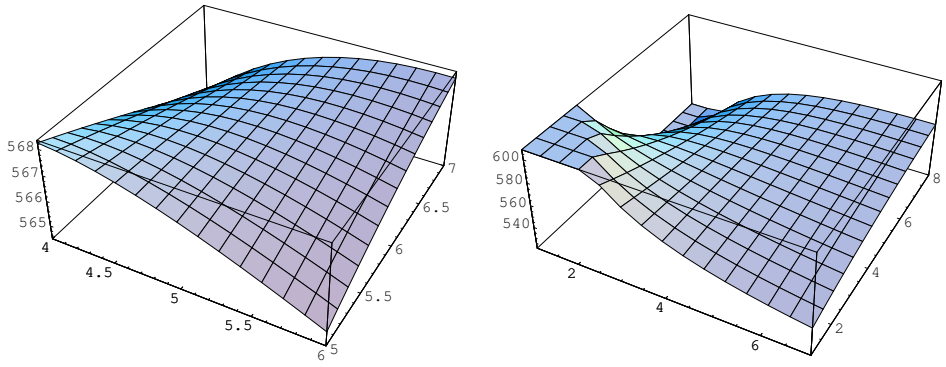


Abbildung 1.1: Ein Plot von Y_{eh} auf $[4,6] \times [5,7]$ und $[1,7] \times [1,8]$

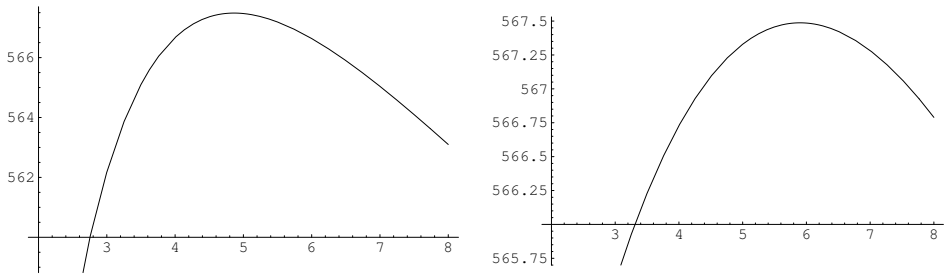


Abbildung 1.2: $Y_{eh}(\cdot, d_h^*)$ und $Y_{eh}(d_e^*, \cdot)$ auf $[2,8]$

gilt. Durch Feldversuche bzw. Beobachtungen und anschließendes Lösen von zwei linearen Ausgleichsproblemen können die Prameter b_i, c_i, $i = 0, \ldots, 5$, bestimmt werden. Danach betrachtet man unter geeigneten Nebenbedingungen die Aufgabe,

$$\begin{aligned} Y_{eh}(d_e, d_h) \quad := \quad & \frac{d_e}{b_0 + b_1 d_e + b_2 d_h + b_3 d_e^2 + b_4 d_h^2 + b_5 d_e d_h} \\ & + \frac{d_h}{c_0 + c_1 d_h + c_2 d_e + c_3 d_h^2 + c_4 d_e^2 + c_5 d_h d_e} \end{aligned}$$

zu maximieren. Diese Aufgabe dürfte (bei bekannten Koeffizienten b_i und c_i) nur numerisch zu lösen sein. □

Zum Schluss einer Einführung wird eine Übersicht erwartet. Der folgende Plan ist unverbindlich, es wird sich sicher einiges ändern. Geplant sind Kapitel über:

- Lineare Optimierungsaufgaben.

  In aller Kürze soll eine Darstellung der Dualitätstheorie, des revidierten Simplexverfahren und eines primal-dualen Innere-Punkt-Verfahrens erfolgen.

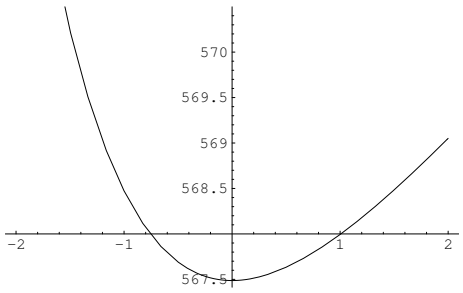


Abbildung 1.3: Die Funktion $f(d) := Y_{eh}(d_e^* + d, d_h^* + d)$ auf $[-2, 2]$

- Graphentheorie.

  Geplant ist ein Steilkurs über Graphentheorie, wobei nicht nur für den Operations Research interessante Probleme angesprochen werden sollen.

- Netzwerkflussprobleme.

- Kombinatorische Optimierungsprobleme.

Literatur werden wir wenn nötig angeben.

Kapitel 2

Lineare Optimierung

Unter einer linearen Optimierungsaufgabe versteht man bekanntlich die Aufgabe, eine lineare reellwertige Funktion unter linearen Nebenbedingungen zu minimieren. Eine lineare Optimierungsaufgabe ist in *Standardform*, wenn sie durch

$$\text{(P)} \qquad \text{Minimiere} \quad c^T x \quad \text{auf} \quad M := \{x \in \mathbb{R}^n : x \geq 0,\ Ax = b\}$$

gegeben ist. Hierbei seien $A \in \mathbb{R}^{m \times n}$, $b \in \mathbb{R}^m$ und $c \in \mathbb{R}^n$. Sind nur gewisse Variable vorzeichenbeschränkt und ein Teil der impliziten Restriktionen (gleichgerichtete) Ungleichungen, die anderen Gleichungen, so hat die entsprechende Optimierungsaufgabe die Form

$$\text{Minimiere} \quad \sum_{j=1}^{n} c_j x_j \quad \text{auf}$$

$$M := \left\{ x \in \mathbb{R}^n : x_j \geq 0 \quad (j = 1, \ldots, n_0), \quad \begin{array}{l} \displaystyle\sum_{j=1}^{n} a_{ij} x_j \geq b_i \quad (i = 1, \ldots, m_0), \\ \displaystyle\sum_{j=1}^{n} a_{ij} x_j = b_i \quad (i = m_0 + 1, \ldots, m) \end{array} \right\}$$

mit $A = (a_{ij}) \in \mathbb{R}^{m \times n}$, $b = (b_i) \in \mathbb{R}^m$, $c = (c_j) \in \mathbb{R}^n$ und $m_0 \in \{0, \ldots, m\}$ sowie $n_0 \in \{0, \ldots, n\}$. Durch Einführung von *Schlupfvariablen* und Darstellung nicht vorzeichenbeschränkter Variablen als Differenz von nichtnegativen Variablen kann man diese Aufgabe auf äquivalente Standardform bringen. Es sind aber noch andere Formen denkbar, wie man im Netz (Linear Programming: Frequently Asked Questions) nachlesen kann.

- Although all linear programs can be put into the Standard Form, in practice it may not be necessary to do so. For example, although the Standard Form requires all variables to be non-negative, most good LP software allows general bounds $l \leq x \leq u$, where l and u are vectors of known lower and upper bounds. Individual elements of these bounds vectors can even be infinity and/or minus-infinity. This allows a variable to be without an explicit upper or lower bound, although of course the constraints in the A-matrix will need to put implied limits

on the variable or else the problem may have no finite solution. Similarly, good software allows $b1 \leq Ax \leq b2$ for arbitrary $b1, b2$; the user need not hide inequality constraints by the inclusion of explicit "slack" variables, nor write $Ax \geq b1$ and $Ax \leq b2$ as two separate constraints. Also, LP software can handle maximization problems just as easily as minimization (in effect, the vector c is just multiplied by -1)[1].

Wir wollen in diesem Kapitel die klassische Theorie linearer Optimierungsaufgaben darstellen, dabei in der Darstellung ein wenig allgemeiner sein als dies im allgemeinen üblich ist. Das bezieht sich vor allem auf den ersten Abschnitt über die Dualitätstheorie linearer Optimierungsaufgaben. Im zweiten Abschnitt wird im wesentlichen das (revidierte) Simplexverfahren dargestellt. Im dritten Abschnitt wollen wir auf Innere-Punkt-Verfahren bei linearen Optimierungsaufgaben eingehen. In dem gesamten Kapitel werden keine speziell strukturierten linearen Optimierungsaufgaben betrachtet.

2.1 Die Dualitätstheorie linearer Optimierungsaufgaben

2.1.1 Das duale Programm, der schwache Dualitätssatz

Wir gehen aus von einem linearen Programm (synonym für Optimierungsaufgabe) der Form

$$\text{(P)} \qquad \text{Minimiere} \quad c^T x \quad \text{auf} \quad M := \{x \in \mathbb{R}^n : x \in C,\ Ax - b \in K\}.$$

Hierbei seien grundsätzlich $A \in \mathbb{R}^{m \times n}$, $b \in \mathbb{R}^m$ und $c \in \mathbb{R}^n$, derner $C \subset \mathbb{R}^n$ und $K \in \mathbb{R}^m$ nichtleere Mengen mit gewissen Eigenschaften, die später spezifiziert werden. Ist (P) in Standardform, so ist z. B. $C = \{x \in \mathbb{R}^n : x \geq 0\}$ der sogenannte nichtnegative Orthant und $K = \{0\}$, während die zu Beginn dieses Kapitels angegebene "allgemeine lineare Optimierungsaufgabe" sich mittels

$$\begin{aligned} C &= \{x \in \mathbb{R}^n : x_j \geq 0\ (j = 1, \ldots, n_0)\}, \\ K &= \{y \in \mathbb{R}^m : y_i \geq 0\ (i = 1, \ldots, m_0),\ y_i = 0\ (i = m_0 + 1, \ldots, m)\} \end{aligned}$$

unterordnet.

Als *duales Programm* zu dem sogenannten *primalen Programm* (P) formulieren wir die Aufgabe

$$\text{(D)} \qquad \text{Maximiere} \quad b^T y \quad \text{auf} \quad N := \{y \in \mathbb{R}^m : y \in K^+,\ c - A^T y \in C^+\}.$$

[1] Durch die Funktion `linprog` in der Optimization Toolbox von MATLAB können z. B. lineare Optimierungsaufgaben der Form "Minimiere $c^T x$ unter den Nebenbedingungen $Ax \leq b$, $A_{eq} x = b_{eq}$ und $l \leq x \leq u$" gelöst werden. Dagegen sind die Möglichkeiten von `LinearProgramming` in *Mathematica* bescheidener. Durch `LinearProgramming[c,A,b]` wird das lineare Programm, $c^T x$ unter den Nebenbedingungen $Ax \geq b$ und $x \geq 0$ zu minimieren, gelöst.

Hierbei sind C^+ und K^+ die sogenannten *dualen Kegel* zu den Mengen C bzw. K. Z. B. ist

$$C^+ := \{x \in \mathbb{R}^n : x^T z \geq 0 \text{ für alle } z \in C\},$$

die Menge K^+ ist natürlich entsprechend definiert.

Bemerkung: Unter einem *Kegel* versteht man eine Menge mit der Eigenschaft, dass mit jedem Punkt der gesamte Strahl vom Nullpunkt durch diesen Punkt zu der Menge gehört. Formal bedeutet dies also, dass eine Menge $C \in \mathbb{R}^n$ ein Kegel ist, wenn $\lambda x \in C$ für alle $x \in C$ und alle nichtnegativen Skalare λ. Offenbar ist der duale Kegel C^+ einer beliebigen nichtleeren Menge $C \subset \mathbb{R}^n$ wirklich ein Kegel (es wäre auch traurig, wenn das nicht der Fall wäre). Der duale Kegel C^+ der nichtleeren Menge $C \subset \mathbb{R}^n$ ist ferner offenbar konvex und abgeschlossen. □

Beispiele: Ist

$$C := \{x \in \mathbb{R}^n : x \geq 0\}, \qquad K := \{0\},$$

handelt es sich bei (P) also um ein lineares Programm in Standardform, so ist

$$C^+ = \{x \in \mathbb{R}^n : x \geq 0\}, \qquad K^+ = \mathbb{R}^m.$$

Daher ist das duale Programm zu

$$\text{(P)} \qquad \text{Minimiere} \quad c^T x \quad \text{auf} \quad M := \{x \in \mathbb{R}^n : x \geq 0,\ Ax = b\}$$

gegeben durch

$$\text{(D)} \qquad \text{Minimiere} \quad b^T y \quad \text{auf} \quad N := \{y \in \mathbb{R}^m : A^T y \leq c\}.$$

Sind C bzw. K die nichtnegativen Orthanten im $\mathbb{R}^n$ bzw. $\mathbb{R}^m$, so ehält man die besonders hübsche symmetrische Form der primalen und daualen linearen Programme, nämlich

$$\text{(P)} \qquad \text{Minimiere} \quad c^T x \quad \text{auf} \quad M := \{x \in \mathbb{R}^n : x \geq 0,\ Ax \geq b\}$$

und

$$\text{(D)} \qquad \text{Maximiere} \quad b^T y \quad \text{auf} \quad N := \{y \in \mathbb{R}^m : y \geq 0,\ A^T y \leq c\}.$$

Ist dagegen

$$\begin{aligned} C &= \{x \in \mathbb{R}^n : x_j \geq 0\ (j = 1, \ldots, n_0)\}, \\ K &= \{y \in \mathbb{R}^m : y_i \geq 0\ (i = 1, \ldots, m_0),\ y_i = 0\ (i = m_0 + 1, \ldots, m)\}, \end{aligned}$$

so ist

$$\begin{aligned} C^+ &= \{x \in \mathbb{R}^n : x_j \geq 0\ (j = 1, \ldots, n_0),\ x_j = 0\ (j = n_0 + 1, \ldots, n)\}, \\ K^+ &= \{y \in \mathbb{R}^m : y_i \geq 0\ (i = 1, \ldots, m_0)\}. \end{aligned}$$

Das duale Programm zu

$$\text{Minimiere} \quad \sum_{j=1}^{n} c_j x_j \quad \text{auf}$$

$$M := \left\{ x \in \mathbb{R}^n : x_j \geq 0 \quad (j = 1, \ldots, n_0), \quad \begin{array}{ll} \sum_{j=1}^{n} a_{ij} x_j \geq b_i & (i = 1, \ldots, m_0), \\ \sum_{j=1}^{n} a_{ij} x_j = b_i & (i = m_0 + 1, \ldots, m) \end{array} \right\}$$

ist also durch

$$\text{Maximiere} \quad \sum_{i=1}^{m} b_i y_i \quad \text{auf}$$

$$N := \left\{ y \in \mathbb{R}^m : y_i \geq 0 \quad (i = 1, \ldots, m_0), \quad \begin{array}{ll} \sum_{i=1}^{m} a_{ij} y_i \leq c_j & (j = 1, \ldots, n_0), \\ \sum_{i=1}^{m} a_{ij} y_i = c_j & (j = n_0 + 1, \ldots, n) \end{array} \right\}$$

gegeben. □

Beispiel: Das klassische Transportproblem ist (nach eventuellem Einführen eines den Überschuss aufnehmenden fiktiven Kunden) gegeben durch

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

unter den Nebenbedingungen

$$\sum_{j=1}^{n} x_{ij} = l_i \qquad (i = 1, \ldots, m), \qquad \sum_{i=1}^{m} x_{ij} = k_j \qquad (j = 1, \ldots, n)$$

sowie

$$x_{ij} \geq 0 \qquad (i = 1, \ldots, m,\ j = 1, \ldots, n)$$

bzw.

$$\text{(P)} \qquad \text{Minimiere} \quad c^T x \quad \text{auf} \quad M := \{x = (x_{ij}) \in \mathbb{R}^{mn} : x \geq 0,\ Ax = b\},$$

wobei

$$A := \begin{pmatrix} e^T & 0^T & \cdots & 0^T \\ 0^T & e^T & & 0^T \\ \vdots & \vdots & \ddots & \vdots \\ 0^T & 0^T & \cdots & e^T \\ I & I & \cdots & I \end{pmatrix} \in \mathbb{R}^{(m+n) \times mn}, \quad b = \begin{pmatrix} l \\ k \end{pmatrix} \in \mathbb{R}^{m+n}, \quad c = (c_{ij}) \in \mathbb{R}^{mn}.$$

Hierbei besitzt $e \in \mathbb{R}^n$ nur Einsen als Komponenten und I ist die $n \times n$-Einheitsmatrix. Die duale Variable y besitzt $m+n$ Komponenten. Es liegt nahe, sie durch

$$y = \begin{pmatrix} u \\ v \end{pmatrix}$$

zu partitionieren. Wegen

$$A^T y = \begin{pmatrix} e & 0 & \cdots & 0 & I \\ 0 & e & \cdots & 0 & I \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & e & I \end{pmatrix} \begin{pmatrix} u \\ v \end{pmatrix} = \begin{pmatrix} u_1 e + v \\ u_2 e + v \\ \vdots \\ u_m e + v \end{pmatrix}, \qquad b^T y = l^T u + k^T v$$

erhält man als duales Problem die Aufgabe

$$\text{(D)} \quad \begin{cases} \text{Maximiere} \quad l^T u + k^T v \quad \text{auf} \\ N := \{(u,v) \in \mathbb{R}^m \times \mathbb{R}^n : u_i + v_j \le c_{ij},\ (i,j) \in \{1,\dots,m\} \times \{1,\dots,n\}\}. \end{cases}$$

□

Beispiel: Die sogenannte lineare diskrete Tschebyscheffsche Approximationsaufgabe

$$\text{(T)} \qquad \text{Minimiere} \quad \|Ax - b\|_\infty, \quad x \in \mathbb{R}^n,$$

wobei $A \in \mathbb{R}^{m \times n}$ und $b \in \mathbb{R}^m$ gegeben sind, kann als eine lineare Optimierungsaufgabe geschrieben werden. Man ordne ihr nämlich das lineare Programm

$$\text{(P)} \qquad \begin{cases} \text{Minimiere} \quad \delta \quad \text{unter den Nebenbedingungen} \\ -\delta e \le Ax - b \le \delta e \end{cases}$$

zu. Hierbei sei $e \in \mathbb{R}^m$ der Vektor, dessen Komponenten sämtlich gleich 1 sind. Es ist leicht nachzuweisen, dass die beiden Probleme (T) und (P) in dem folgenden Sinne äquivalent sind:

- Ist x^* eine Lösung von (T), so ist $(x^*, \|Ax^* - b\|_\infty)$ eine Lösung von (P).
- Ist (x^*, δ^*) eine Lösung von (P), so ist x^* eine Lösung von (T) und $\|Ax^* - b\|_\infty = \delta^*$.

Um das duale lineare Programm zu (P) aufzustellen, schreiben wir (P) in der Form

$$\text{(P)} \qquad \begin{cases} \text{Minimiere} \quad \begin{pmatrix} 0 \\ 1 \end{pmatrix}^T \begin{pmatrix} x \\ \delta \end{pmatrix} \quad \text{unter den Nebenbedingungen} \\ (x, \delta) \in \mathbb{R}^n \times \mathbb{R}, \quad \begin{pmatrix} -A & e \\ A & e \end{pmatrix} \begin{pmatrix} x \\ \delta \end{pmatrix} - \begin{pmatrix} -b \\ b \end{pmatrix} \in \mathbb{R}^m_{\ge 0} \times \mathbb{R}^m_{\ge 0}. \end{cases}$$

Wegen $(\mathbb{R}^n \times \mathbb{R})^+ = \{0\} \times \{0\}$ und $(\mathbb{R}^m_{\ge 0} \times \mathbb{R}^m_{\ge 0})^+ = \mathbb{R}^m_{\ge 0} \times \mathbb{R}^m_{\ge 0}$ ist das zu (P) duale Programm

$$\text{(D)} \qquad \begin{cases} \text{Maximiere} \quad \begin{pmatrix} -b \\ b \end{pmatrix}^T \begin{pmatrix} u \\ v \end{pmatrix} \quad \text{unter den Nebenbedingungen} \\ (u, v) \in \mathbb{R}^m_{\ge 0} \times \mathbb{R}^m_{\ge 0}, \quad \begin{pmatrix} -A^T & A^T \\ e^T & e^T \end{pmatrix} \begin{pmatrix} u \\ v \end{pmatrix} - \begin{pmatrix} 0 \\ 1 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}. \end{cases}$$

Dieses duale Programm hat also (was nicht verwunderlich ist) Standardform. □

Beispiel: Sei $A \in \mathbb{R}^{m\times n}$ (Auszahlungsmatrix) gegeben. Sei

$$X := \{x \in \mathbb{R}^n : x \geq 0,\ e^T x = 1\}, \qquad Y := \{y \in \mathbb{R}^m : y \geq 0,\ e^T y = 1\},$$

wobei e der Vektor des $\mathbb{R}^n$ bzw. $\mathbb{R}^m$ sei, dessen Komponenten alle gleich 1 sind. Man betrachte die beiden Aufgaben

$$(\mathrm{P}_0) \qquad \text{Minimiere} \quad \phi(x) := \max_{y\in Y} y^T A x, \quad x \in X,$$

und

$$(\mathrm{D}_0) \qquad \text{Maximiere} \quad \psi(y) := \min_{x\in X} y^T A x, \quad y \in Y.$$

Wie man leicht nachweist, ist

$$\phi(x) = \max_{i=1,\dots,m} (Ax)_i, \qquad \psi(y) = \min_{j=1,\dots,n} (A^T y)_j.$$

Mit dem selben Trick wie im vorigen Beispiel können die beiden Aufgaben (P_0) und (D_0) als äquivalent zu linearen Programmen entlarvt werden, nämlich zu

$$(\mathrm{P}) \qquad \begin{cases} \text{Minimiere} \quad \alpha \quad \text{unter den Nebenbedingungen} \\ x \geq 0, \quad Ax \leq \alpha e, \quad e^T x = 1 \end{cases}$$

bzw.

$$(\mathrm{D}) \qquad \begin{cases} \text{Maximiere} \quad \beta \quad \text{unter den Nebenbedingungen} \\ y \geq 0, \quad A^T y \geq \beta e, \quad e^T y = 1. \end{cases}$$

Die Äquivalenz ist wieder so wie im letzten Beispiel zu verstehen. Es ist sehr einfach zu sehen, dass (D) gerade das zu (P) duale lineare Programm ist. □

Beispiel: Die[2] Spieler P und D haben je 3 Karten auf der Hand, und zwar P die Karten Pik As, Karo As und Pik Zwei, D die Karten Pik As, Karo As und Karo Zwei. Beide Spieler legen jeweils zugleich eine ihrer Karten auf den Tisch. D gewinnt, wenn die hingelegten Karten die gleiche Farbe haben, andernfalls P. Ein As hat den Wert 1, eine Zwei den Wert 2. Die Höhe des Gewinnes ist gleich dem Wert derjenigen Karte, die der Gewinner hingelegt hat. Das Spiel hat also die Auszahlungsmatrix

D \ P	♢	♠	♠♠
♢	1	−1	−2
♠	−1	1	1
♢♢	2	−1	−2

[2]Siehe

L. Collatz, W. Wetterling (1971) *Optimierungsaufgaben*. Springer-Verlag, Berlin-Heidelberg-New York.

Man hat den Eindruck, das Spiel sei unfair, weil die Auszahlungsmatrix 5 negative Elemente gegenüber 4 positiven enthält. Das gibt Anlass zur Formulierung der

Zusatzregel: Wenn beide Spieler ihre Zweierkarte hinlegen, so soll keiner an den anderen etwas zahlen, d. h. das Element -2 in der rechten unteren Ecke der Auszahlungsmatrix wird durch 0 ersetzt.

Wir wollen für das Spiel ohne und mit Zusatzregel mit Hilfe des mathematischen Anwendersystems jeweils optimale gemischte Strategien für P und D berechnen und damit entscheiden, welches der beiden Spiele fair ist.

Mit

$$A := \begin{pmatrix} 1 & -1 & -2 \\ -1 & 1 & 1 \\ 2 & -1 & -2 \end{pmatrix}$$

kann das von Spieler P zu lösende Problem geschrieben werden als

$$\text{(P)} \qquad \begin{cases} \text{Minimiere} \quad \begin{pmatrix} 0 \\ 1 \\ -1 \end{pmatrix}^T \begin{pmatrix} x \\ \alpha_+ \\ \alpha_- \end{pmatrix} \quad \text{unter den Nebenbedingungen} \\ \begin{pmatrix} x \\ \alpha_+ \\ \alpha_- \end{pmatrix} \geq 0, \quad \begin{pmatrix} -A & e & -e \\ e^T & 0 & 0 \\ -e^T & 0 & 0 \end{pmatrix} \begin{pmatrix} x \\ \alpha_+ \\ \alpha_- \end{pmatrix} \geq \begin{pmatrix} 0 \\ 1 \\ -1 \end{pmatrix}. \end{cases}$$

Entsprechend kann das von dem Spieler D zu lösende Problem als

$$\text{(D)} \qquad \begin{cases} \text{Minimiere} \quad \begin{pmatrix} 0 \\ -1 \\ 1 \end{pmatrix}^T \begin{pmatrix} y \\ \beta_+ \\ \beta_- \end{pmatrix} \quad \text{unter den Nebenbedingungen} \\ \begin{pmatrix} y \\ \beta_+ \\ \beta_- \end{pmatrix} \geq 0, \quad \begin{pmatrix} A^T & -e & e \\ e^T & 0 & 0 \\ -e^T & 0 & 0 \end{pmatrix} \begin{pmatrix} y \\ \beta_+ \\ \beta_- \end{pmatrix} \geq \begin{pmatrix} 0 \\ 1 \\ -1 \end{pmatrix} \end{cases}$$

geschrieben werden. Nach

```
<<LinearAlgebra‘MatrixManipulation‘
m=2;n=3;A={{1,-1,-2},{-1,1,1},{2,-1,-2}};
em=Table[1,{i,m},{j,1}];
en=Table[1,{i,n},{j,1}];
zerom=Table[0,{i,m}];
zeron=Table[0,{j,n}];
ap=BlockMatrix[{{-a,em,-em},{
        Transpose[en],{{0}},{{0}}},{-Transpose[en],{{0}},{{0}}}}];
ad=BlockMatrix[{{Transpose[a],-en,en},{
        Transpose[em],{{0}},{{0}}},{-Transpose[em],{{0}},{{0}}}}];
bp=Flatten[{zerom,1,-1}];
bd=Flatten[{zeron,1,-1}];
cp=Flatten[{zeron,1,-1}];
cd=Flatten[{zerom,-1,1}];
```

```
x=LinearProgramming[cp,ap,bp]
wertp=cp.x
y=LinearProgramming[cd,ad,bd]
wertd=cd.y
```

erhalten wir die Lösungen

$$x^* = (\tfrac{1}{2}, 0, \tfrac{1}{2})^T, \qquad y^* = (0, \tfrac{2}{3}, \tfrac{1}{3})^T,$$

der Wert ist jeweils 0, das Spiel also fair. Mit der Zusatzregel sind die Lösungen

$$x^* = (\tfrac{2}{5}, \tfrac{3}{5}, 0)^T, \qquad y^* = (0, \tfrac{3}{5}, \tfrac{2}{5})^T,$$

der Wert ist $\frac{1}{5}$. In diesem Falle ist das Spiel also nicht fair, der Spieler P verliert. □

Einige Vokabeln sollten wir jetzt noch einführen. Eine (nicht notwendig lineare) Optimierungsaufgabe (oder auch *Programm*) besteht darin, eine *Zielfunktion* (*Kostenfunktion* bei einer Minimierungsaufgabe, *Gewinnfunktion* bei einer Maximierungsaufgabe) auf einer Menge, der Menge der *zulässigen Lösungen*, zu minimieren oder zu maximieren. Das Programm selber heißt *zulässig*, wenn die Menge der zulässigen Lösungen nichtleer ist.

Jetzt geben wir den fast trivialen schwachen Dualitätssatz an:

Satz 1.1 *Gegeben seien die Programme*

(P) $$\textit{Minimiere} \quad c^T x \quad \textit{auf} \quad M := \{x \in \mathbb{R}^n : x \in C,\ Ax - b \in K\}$$

und

(D) $$\textit{Maximiere} \quad b^T y \quad \textit{auf} \quad N := \{y \in \mathbb{R}^m : y \in K^+,\ c - A^T y \in C^+\},$$

wobei $C \subset \mathbb{R}^n$ und $K \subset \mathbb{R}^m$ (noch) beliebige nichtleere Mengen sind. Dann gilt:

1. *Ist $x \in M$ und $y \in N$, so ist $b^T y \le c^T x$. Daher ist*[3]

$$\sup(\mathrm{D}) := \sup_{y \in N} b^T y \le \inf_{x \in M} c^T x =: \inf(\mathrm{P}).$$

2. *Ist $x^* \in M$, $y^* \in N$ und $b^T y^* = c^T x^*$, so ist*

$$c^T x^* = \inf(\mathrm{P}) = \sup(\mathrm{D}) = b^T y^*,$$

insbesondere also x^* eine Lösung von (P) und y^* eine Lösung von (D).

[3]Das Supremum über die leere Menge ist als $-\infty$, das Infimum über die leere Menge als $+\infty$ definiert.

Beweis: Ist $x \in M$ und $y \in N$, so ist

$$\begin{aligned} c^T x &= (c - A^T y + A^T y)^T x \\ &= \underbrace{\underbrace{(c - A^T y)}_{\in C^+}{}^T \underbrace{x}_{\in C} + y^T A x}_{\geq 0} \\ &\geq y^T (Ax - b + b) \\ &= \underbrace{\underbrace{y}_{\in K^+}{}^T (\underbrace{Ax - b}_{\in K}) + b^T y}_{\geq 0} \\ &\geq b^T y, \end{aligned}$$

womit der erste Teil bewiesen ist. Der zweite Teil ist genau so einfach einzusehen. Denn ist $x^* \in M$, $y^* \in N$, so ist

$$b^T y^* \leq \sup(\mathrm{D}) \leq \inf(\mathrm{D}) \leq c^T x^*.$$

Hieraus folgt die Behauptung. □

Beispiel: Beim Produktionsplanungsproblem will ein Betrieb unter Kapazitätsbeschränkungen an benötigte Hilfsmittel seinen Gesamtgewinn maximieren. Als lineares Programm lautet es:

$$\text{Maximiere} \quad p^T x \quad \text{unter den Nebenbedingungen} \quad x \geq 0, \quad Ax \leq b.$$

Dieses Programm kann auch in der Form

$$\text{Minimiere} \quad (-p)^T x \quad \text{unter den Nebenbedingungen} \quad x \geq 0, \quad (-A)x - (-b) \geq 0$$

geschrieben werden. Das hierzu duale Programm ist

$$\text{Maximiere} \quad (-b)^T y \quad \text{unter den Nebenbedingungen} \quad y \geq 0, \quad -p - (-A)^T y \geq 0$$

bzw.

$$\text{Minimiere} \quad b^T y \quad \text{unter den Nebenbedingungen} \quad y \geq 0, \quad A^T y \geq p.$$

Dieses duale Programm kann man folgendermaßen interpretieren: Ein Konkurrent macht dem Betrieb das Angebot, alle m Hilfsmittel zu mieten bzw. zu kaufen und bietet für das i-te Hilfsmittel $y_i \geq 0$ Geldeinheiten pro Einheit. Insgesamt sind seine Kosten $b^T y = \sum_{i=1}^m b_i y_i$ und diese wird er versuchen zu minimieren. Der Betrieb geht auf diesen Vorschlag aber nur dann ein, wenn $\sum_{i=1}^m a_{ij} y_i \geq p_j$, $j = 1, \ldots, n$, bzw. $A^T y \geq p$, d. h. wenn der gezahlte Preis für sämtliche Hilfsmittel zur Produktion einer Einheit des j-ten Produktes nicht kleiner ist als der Reingewinn p_j, den der Betrieb erhalten hätte, wenn er die Produktion selbst durchführen würde. Der Konkurrent hat also genau das duale Programm zu lösen. □

2.1.2 Die Sätze von Weyl und Minkowski, das Farkas-Lemma

Eine *Hyperebene* im $\mathbb{R}^n$ ist mit $(y, \gamma) \in (\mathbb{R}^n \setminus \{0\}) \times \mathbb{R}$ durch

$$H := \{x \in \mathbb{R}^n : y^T x = \gamma\}$$

gegeben. In Abbildung 2.1 wird dies veranschaulicht. Die Hyperebene H erzeugt die

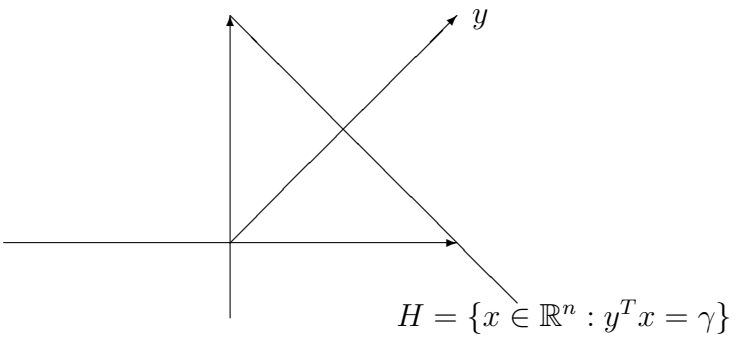


Abbildung 2.1: Hyperebene

beiden abgeschlossenen *Halbräume*

$$H_+ := \{x \in \mathbb{R}^n : y^T x \geq \gamma\}, \qquad H_- := \{x \in \mathbb{R}^n : y^T x \leq \gamma\}.$$

Ein *Polyeder* ist der endliche Durchschnitt von abgeschlossenen Halbräumen. Sind die Halbräume speziell durch Hyperebenen, die den Nullpunkt enthalten, erzeugt, so spricht man speziell von einem *polyedrischen Kegel*. Ein solcher lässt sich also in der Form $\{x \in \mathbb{R}^n : U^T x \geq 0\}$ mit einer Matrix $U \in \mathbb{R}^{n \times m}$ darstellen. Ist $C \subset \mathbb{R}^n$ nichtleer, so heißt der kleinste konvexe Kegel im $\mathbb{R}^n$, der die Menge C enthält, also der Durchschnitt aller die Menge C enthaltenden konvexen Kegel, der von A erzeugte konvexe Kegel und wird mit $\operatorname{cone}(C)$ bezeichnet. Es ist leicht nachzuweisen, dass $\operatorname{cone}(C)$ die Menge aller endlichen, nichtnegativen Linearkombinationen von Elementen aus C ist. Ist C eine endliche Menge, so heißt $\operatorname{cone}(C)$ ein *endlich erzeugter Kegel*. Jeder endlich erzeugte Kegel im $\mathbb{R}^n$ lässt dich daher in der Form $\{Uy : y \in \mathbb{R}^m,\ y \geq 0\}$ darstellen, wobei $U \in \mathbb{R}^{n \times m}$. Nun geben wir ohne Beweis[4] die Sätze von Weyl und Minkowski an.

Satz 1.2 (Weyl) *Ein endlich erzeugter Kegel ist ein polyedrischer Kegel.*

Satz 1.3 (Minkowski) *Ein polyedrischer Kegel ist ein endlich erzeugter Kegel.*

[4]Einen Beweis findet man auf S. 55 ff. bei

J. Stoer, C. Witzgall (1970) *Convexity and Optimization in Finite Dimensions I*. Springer-Verlag, Berlin-Heidelberg-New York

oder auch in §2 bei

J. Werner (1989) *Optimierung*. Kurs der Fernuniversität Hagen.

Bemerkung: Die beiden obigen Sätze sind keineswegs trivial. Sie sind aus verschiedenen Gründen nützlich. Z. B. sind polyedrische Kegel trivialerweise abgeschlossen. Wegen des Satzes von Weyl sind daher auch endlich erzeugte Kegel abgeschlossen. Weiter ist das lineare Bild eines eines endlich erzeugten Kegels offensichtlich ein endlich erzeugter Kegel. Denn ist etwa $C = \{Uy : y \in \mathbb{R}^m,\ y \geq 0\}$ ein endlich erzeugter Kegel im $\mathbb{R}^n$, also $U \in \mathbb{R}^{n\times m}$, und $A \in \mathbb{R}^{k\times n}$, so ist

$$A(C) = \{AU : y \in \mathbb{R}^m,\ y \geq 0\}$$

ein endlich erzeugter Kegel im $\mathbb{R}^k$. Wegen des Satzes von Minkowski ist daher auch das lineare Bild eines polyedrischen Kegels ein polyedrischer Kegel. Leicht kann nan sich weitere Beispiele überlegen. Z. B. ist der Durchschnitt endlich erzeugter Kegel ebenfalls ein endlich erzeugter Kegel (denn die entsprechende Aussage ist für polyedrische Kegel trivial). Auch die Summe endlich erzeugter Kegel ist fast trivialerweise ein endlich erzeugter Kegel. Denn ist etwa

$$C_1 := \{U_1y_1 : y_1 \in \mathbb{R}^{m_1},\ y_1 \geq 0\}, \qquad C_2 := \{U_2y_2 : y_2 \in \mathbb{R}^{m_2},\ y_2 \geq 0\}$$

mit $U_1 \in \mathbb{R}^{n\times m_1}$, $U_2 \in \mathbb{R}^{n\times m_2}$, so ist

$$\begin{aligned} C_1 + C_2 &= \{U_1y_1 + U_2y_2 : (y_1, y_2) \in \mathbb{R}^{m_1} \times \mathbb{R}^{m_2},\ y_1 \geq 0,\ y_2 \geq 0\} \\ &= \{Uy : y \in \mathbb{R}^{m_1+m_2},\ y \geq 0\} \end{aligned}$$

mit $U := \begin{pmatrix} U_1 & U_2 \end{pmatrix} \in \mathbb{R}^{n\times(m_1+m_2)}$. Daher ist auch die Summe polyedrischer Kegel wieder ein polyedrischer Kegel. Ist $C := \{Uy : y \geq 0\}$ ein endlich erzeugter Kegel im $\mathbb{R}^n$, so ist offensichtlich $C^+ = \{x \in \mathbb{R}^n : U^Tx \geq 0\}$ ein polyedrischer Kegel. Insbesondere ist der zu einem polyedrischen Kegel duale Kegel wieder ein polyedrischer Kegel. □

Es folgt der bekannte *Projektionssatz für konvexe Mengen*. Hier und im folgenden sei $\|\cdot\|$ die euklidische Norm.

Satz 1.4 (Projektionssatz) *Sei $M \subset \mathbb{R}^n$ nichtleer, abgeschlossen und konvex, $z \in \mathbb{R}^n$. Dann besitzt die Aufgabe*

$$\text{(P)} \qquad \textit{Minimiere} \quad \|x - z\|, \quad x \in M,$$

genau eine Lösung x^*, die sogenannte Projektion von z auf M. Ferner ist ein $x^* \in M$ genau dann eine Lösung von (P), wenn $(x^* - z)^T(x - x^*) \geq 0$ für alle $x \in M$.

Beweis: Man wähle sich $x_0 \in M$ beliebig und bilde die Niveaumenge $L_0 := \{x \in M : \|x - z\|_2 \leq \|x_0 - z\|_2\}$. Diese ist der Durchschnitt der abgeschlossenen Menge M und einer abgeschlossenen Kugel, also kompakt, woraus die Existenz einer Lösung folgt. Für $x, y \in M$ und $\lambda \in (0, 1)$ gilt mit $f(x) := \frac{1}{2}\|x - z\|_2^2$ nach leichter Rechnung

$$(1 - \lambda)f(x) + \lambda f(y) - f((1 - \lambda)x + \lambda y) = \frac{\lambda(1 - \lambda)}{2}\|x - y\|_2^2.$$

woraus unmittelbar Eindeutigkeit einer Lösung von (P) folgt. Wegen $\nabla f(x^*)^T(x - x^*) = (x^* - z)^T(x - x^*)$ folgt leicht der letzte Teil des Projektionssatzes. □

Als Folgerung aus dem Projektionssatz erhalten wir einen *starken Trennungssatz*.

Satz 1.5 *Sei $K \subset \mathbb{R}^n$ nichtleer, konvex und abgeschlossen. Dann existiert zu jedem $z \notin K$ ein $y \in \mathbb{R}^n \setminus \{0\}$ mit $y^T z < \inf_{x \in K} y^T x$.*

Beweis: Sei x^* die Projektion von z auf K. Man definiere $y := (x^* - z)/\|x^* - z\|$. Für beliebiges $x \in K$ ist wegen des Projektionssatzes $y^T(x - x^*) \geq 0$ und folglich

$$\begin{aligned} y^T x &= \frac{1}{2} y^T (x^* + z) + \underbrace{y^T (x - x^*)}_{\geq 0} + \frac{1}{2} y^T (x^* - z) \\ &\geq \frac{1}{2} y^T (x^* + z) + \frac{1}{2} y^T (x^* - z) \\ &= \frac{1}{2} y^T (x^* + z) + \frac{1}{2} \|x^* - z\| \\ &= y^T z + \|x^* - z\|. \end{aligned}$$

Folglich ist

$$\inf_{x \in K} y^T x \geq y^T z + \underbrace{\|x^* - z\|}_{>0} > y^T z,$$

was zu zeigen war. □

Als Korollar zum starken Trennungssatz erhalten wir:

Korollar 1.6 *Sei $C \subset \mathbb{R}^n$ ein abgeschlossener, konvexer Kegel, also z. B. ein polyedrischer Kegel. Dann ist $C^{++} := (C^+)^+ = C$.*

Beweis: Zunächst überlegen wir uns, dass $C \subset C^{++}$. Denn ist $x \in C$ und $y \in C^+$, so ist $x^T y \geq 0$ und daher $x \in (C^+)^+ = C^{++}$. Angenommen, es gibt ein $x \in C^{++} \setminus C$. Wegen des starken Trennungssatzes gibt es ein $y \in \mathbb{R}^n$ mit $y^T x < \inf_{z \in C} y^T z$. Hieraus folgt $y \in C^+$, woraus sich der Widerspruch

$$0 \leq y^T x < \inf_{z \in C} y^T z = 0$$

ergibt. □

Bemerkung: Sind $C \subset \mathbb{R}^n$ und $K \subset \mathbb{R}^m$ polyedrische Kegel, so sind auch C^+ und K^+ polyedrische Kegel, so dass das zum primalen Programm

(P) Minimiere $c^T x$ auf $M := \{x \in \mathbb{R}^n : x \in C,\ Ax - b \in K\}$

duale Programm

(D) Maximiere $b^T y$ auf $N := \{y \in \mathbb{R}^m : y \in K^+,\ c - A^T y \in C^+\}$,

jedenfalls wenn man dieses in der äquivalenten Form

Minimiere $(-b)^T y$ auf $N := \{y \in \mathbb{R}^m : y \in K^+,\ (-A^T)y - (-c) \in C^+\}$

schreibt, dieselbe Struktur hat. Dualisieren dieser Aufgabe (bzw. von (D)) ergibt wegen $C^{++} = C$ und $K^{++} = K$ genau wieder das Ausgangsproblem (P). □

Schließlich geben wir das Farkas-Lemma (bzw. eine Verallgemeinerung) an.

Satz 1.7 (Farkas) *Seien $A \in \mathbb{R}^{m\times n}$ und $b \in \mathbb{R}^m$ gegeben. Ferner seien $C \subset \mathbb{R}^n$ und $K \subset \mathbb{R}^m$ polyedrische Kegel. Dann gilt genau eine der beiden folgenden Aussagen.*

1. *Es existiert ein $x \in \mathbb{R}^n$ mit $b - Ax \in K$, $x \in C$.*

2. *Es existiert ein $y \in \mathbb{R}^m$ mit $A^T y \in C^+$, $y \in K^+$ und $b^T y < 0$.*

Hierbei bedeuten C^+ und K^+ die zu C bzw. K dualen Kegel.

Beweis: Angenommen, beide Aussagen würden gelten. Dann existieren also $x \in C$ mit $b - Ax \in K$ und $y \in K^+$ mit $A^T y \in C^+$ und $b^T y < 0$. Dann wäre

$$0 > b^T y = \underbrace{(b - Ax)^T y}_{\geq 0} + \underbrace{x^T A^T y}_{\geq 0} \geq 0,$$

ein Widerspruch. Nun nehmen wir an, dass die erste Aussage nicht gilt. Dann ist $b \notin A(C) + K$. Nach den den Sätzen von Weyl und Minkowski folgenden Bemerkungen ist $A(C) + K$ ein polyedrischer Kegel, insbesondere also nichtleer, konvex und abgeschlossen. Wegen Satz 1.5 existiert ein $y \in \mathbb{R}^m$ mit

$$y^T b < \inf_{x\in C,\, z\in K} y^T(Ax + z).$$

Hieraus schließt man sehr leicht, dass $A^T y \in C^+$, $y \in K^+$ und $b^T y < 0$, also die zweite Aussage richtig ist. □

Bemerkung: Das Farkas-Lemma in der obigen Form enthält einige bekannte Spezialfälle. Ist z. B. $C := \mathbb{R}^n$ (und daher $C^+ = \{0\}$) und $K := \{0\}$ (und daher $K^+ = \mathbb{R}^m$), so erhält man die wichtige Aussage der linearen Algebra: Das lineare Gleichungssystem $Ax = b$ ist genau dann lösbar, wenn $b^T y = 0$ für alle $y \in \text{Kern}\,(A^T)$. Diese Aussage kann man auch als $\text{Bild}\,(A) = \text{Kern}\,(A^T)^\perp$ formulieren. Das *klassische Farkas-Lemma* erhält man, wenn man $C := \{x \in \mathbb{R}^n : x \geq 0\}$ und $K := \{0\}$ setzt. Die entsprechende Aussage lautet dann:

- Genau eine der beiden folgenden Aussagen gilt.

  1. Es existiert ein $x \in \mathbb{R}^n$ mit $Ax = b$, $x \geq 0$.
  2. Es existiert ein $y \in \mathbb{R}^m$ mit $A^T y \geq 0$ und $b^T y < 0$.

Weitere Spezialfälle sind leicht herzuleiten. Interessante historische Bemerkungen zu Polyedern, linearen Ungleichungen (insbesondere den Sätzen von Weyl, Minkowski und Farkas) und linearen Optimierungsaufgaben findet man bei A. Schrijver (1986, S. 209 ff.)[5]. □

[5] A. Schrijver (1986) *Theory of Linear and Integer Programming.* J. Wiley, Chichester.

2.1.3 Existenzsatz, starker Dualitätssatz

Ist eine reellwertige Funktion auf einer gewissen Menge nach unten beschränkt, so nimmt sie nicht notwendig auf dieser Menge ihr Minimum an (man denke etwa an $f(x) := e^{-x}$ auf $\mathbb{R}$). Bei linearen Optimierungsaufgaben ist das anders, wie der Existenzsatz der linearen Optimierung aussagt.

Satz 1.8 *Gegeben sei das lineare Programm*

$$\text{(P)} \qquad \textit{Minimiere} \quad c^T x \quad \textit{auf} \quad M := \{x \in \mathbb{R}^n : x \in C,\ Ax - b \in K\},$$

wobei $C \subset \mathbb{R}^n$ und $K \subset \mathbb{R}^m$ polyedrische Kegel sind. Ist dann (P) zulässig (bzw. $M \neq \emptyset$) und $\inf(\text{P}) := \inf_{x \in M} c^T x > -\infty$, *so besitzt (P) eine Lösung.*

Beweis: Sei $\{x_k\} \subset M$ eine Folge mit $c^T x_k \to \inf(\text{P})$. Dann ist

$$\begin{aligned} (c^T x_k, 0) &= (c^T x_k, Ax_k - b - (Ax_k - b)) \\ &\in \{(c^T x, Ax - b - z) \in \mathbb{R} \times \mathbb{R}^m : x \in C,\ z \in K\} \\ &=: \Lambda. \end{aligned}$$

Die Menge Λ ist abgeschlossen (als verschobener polyedrischer Kegel in $\mathbb{R} \times \mathbb{R}^m$), daher ist auch $(\inf(\text{P}), 0) \in \Lambda$, was die Existenz eines $x \in M$ mit $c^T x = \inf(\text{P})$ bzw. die Lösbarkeit von (P) impliziert. □

Beispiel: Mit $e \in \mathbb{R}^m$ sei der Vektor bezeichnet, dessen Komponenten sämtlich gleich 1 ist. Sei $A \in \mathbb{R}^{m \times n}$ und $b \in \mathbb{R}^m$, $b \geq 0$. Wegen des Existenzsatzes besitzt die Aufgabe

$$\begin{cases} \text{Minimiere} \quad e^T z = \begin{pmatrix} 0 \\ e \end{pmatrix}^T \begin{pmatrix} x \\ z \end{pmatrix} \quad \text{auf} \\ \left\{(x,z) \in \mathbb{R}^n \times \mathbb{R}^m : \begin{pmatrix} x \\ z \end{pmatrix} \geq 0,\ \begin{pmatrix} A & I \end{pmatrix} \begin{pmatrix} x \\ z \end{pmatrix} = b\right\} \end{cases}$$

eine Lösung. □

Es folgt der starke Dualitätssatz der linearen Optimierung. Hierbei schreiben wir $\min(\text{P})$ statt $\inf(\text{P})$, wenn das Programm (P) lösbar ist. Ensprechendes gilt für $\max(\text{D})$ und $\sup(\text{D})$.

Satz 1.9 *Seien $C \subset \mathbb{R}^n$ und $K \subset \mathbb{R}^m$ polyedrische Kegel. Hiermit betrachte man die zueinander dualen linearen Programme*

$$\text{(P)} \qquad \textit{Minimiere} \quad c^T x \quad \textit{auf} \quad M := \{x \in \mathbb{R}^n : x \in C,\ Ax - b \in K\}$$

und

$$\text{(D)} \qquad \textit{Maximiere} \quad b^T y \quad \textit{auf} \quad N := \{y \in \mathbb{R}^m : y \in K^+,\ c - A^T y \in C^+\}.$$

Dann gilt:

1. *Sind (P) und (D) zulässig, so besitzen (P) und (D) jeweils eine Lösung und es ist* $\max(\mathrm{D}) = \min(\mathrm{P})$.
2. *Ist (D) zulässig, aber (P) nicht zulässig, so ist* $\sup(\mathrm{D}) = +\infty$.
3. *Ist (P) zulässig, aber (D) nicht zulässig, so ist* $\inf(\mathrm{P}) = -\infty$.

Beweis: Sind (P) und (D) zulässig, so folgt aus dem schwachen Dualitätssatz, dass

$$-\infty < \sup(\mathrm{D}) \le \inf(\mathrm{P}) < +\infty.$$

Aus dem Existenzsatz der linearen Optimierung folgt, dass (P) und (D) jeweils eine Lösung besitzen. Wir zeigen die Existenz eines $x \in M$ mit $c^T x = \max(\mathrm{D})$. Ist dies nicht der Fall, so hat

$$Ax - b \in K, \quad x \in C, \quad c^T x = \max(\mathrm{D})$$

bzw.

$$\text{(I)} \qquad \begin{pmatrix} b \\ \max(\mathrm{D}) \end{pmatrix} - \begin{pmatrix} A \\ c^T \end{pmatrix} x \in -K \times \{0\}, \qquad x \in C$$

keine Lösung. Das Farkas-Lemma liefert, dass das System

$$\text{(II)} \quad \begin{pmatrix} A^T & c \end{pmatrix} \begin{pmatrix} u \\ \delta \end{pmatrix} \in C^+, \qquad (u, \delta) \in -K^+ \times \mathbb{R}, \qquad \begin{pmatrix} b \\ \max(\mathrm{D}) \end{pmatrix}^T \begin{pmatrix} u \\ \delta \end{pmatrix} < 0$$

bzw.

$$A^T u + \delta c \in C^+, \qquad b^T u + \delta \max(\mathrm{D}) < 0$$

eine Lösung $(u, \delta) \in -K^+ \times \mathbb{R}$ besitzt. Ist $\delta > 0$, so ist $y := -u/\delta$ dual zulässig und $\max(\mathrm{D}) < b^T y$, ein Widerspruch zur Definition des Optimalwertes $\max(\mathrm{D})$. Daher nehmen wir nun an, es sei $\delta \le 0$. Mit einem primal zulässigen x ist dann (unter Benutzung des schwachen Dualitätssatzes)

$$b^T u < -\delta \max(\mathrm{D}) \le -\delta c^T x \le b^T u,$$

ein Widerspruch. Folglich ist (I) lösbar, es existiert also ein primal zulässiges x mit $c^T x = \max(\mathrm{D})$. Der schwache Dualitässatz liefert, dass x eine Lösung von (P) ist und $\max(\mathrm{D}) = \min(\mathrm{P})$ gilt.

Sei (D), aber nicht (P) zulässig. Da $b - Ax \in -K$, $x \in C$ keine Lösung besitzt, liefert das Farkas-Lemma ein $v \in -K^+$ mit $A^T v \in C^+$, $b^T v < 0$. Mit einem beliebigen dual zulässigen y ist dann $y - tv$ dual zulässig für alle $t \ge 0$ und $b^T(y - tv) \to +\infty$ für $t \to +\infty$, also $\sup(\mathrm{D}) = +\infty$.

Ist schließlich (P), nicht aber (D) zulässig, so erhält man $\inf(\mathrm{P}) = -\infty$, indem man von (D) (als Minimierungsaufgabe geschrieben) als Ausgangsproblem und (P) als hierzu duales Problem betrachtet und den gerade bewiesenen Teil anwendet[6]. Damit ist der starke Dualitätssatz bewiesen. □

[6]Natürlich kann man auch folgendermaßen argumentieren: Da (D) nicht zulässig ist, besitzt $c - A^T y \in C^+$, $y \in K^+$ keine Lösung. Wegen des Farkas-Lemmas existiert ein $u \in (C^+)^+ = C$ mit $Au \in (K^+)^+ = K$ und $c^T u < 0$. Mit einem beliebigen primal zulässigen x ist $x + tu$ primal zulässig für alle $t \ge 0$ und $c^T(x + tu) \to -\infty$ fütr $t \to +\infty$, also $\inf(\mathrm{P}) = -\infty$.

Während der schwache Dualitätssatz hinreichende Optimalitätsbedingungen bei linearen Programmen angibt, sagt der starke Dualitätssatz u. a. aus, dass diese Bedingungen sogar notwendig sind.

Beispiel: Als Folgerung aus dem starken Dualitätssatz erhält man sofort den Beweis des Hauptsatzes der Theorie der Matrixspiele:

- Ist $A \in \mathbb{R}^{m\times n}$ und

$$X := \{x \in \mathbb{R}^n : x \geq 0,\ e^T x = 1\}, \qquad Y := \{y \in \mathbb{R}^m : y \geq 0,\ e^T y = 1\},$$

so ist

$$\min_{x\in X}\max_{y\in Y} y^T A x = \max_{y\in Y}\min_{x\in X} y^T A x.$$

Denn in einem früheren Beispiel haben wir die Aufgaben

$$\text{Minimiere} \quad \phi(x) := \max_{y\in Y} y^T A x, \quad x \in X,$$

und

$$\text{Maximiere} \quad \psi(y) := \min_{x\in X} y^T A x, \quad y \in Y,$$

als offensichtlich zulässige und zueinander duale lineare Programme entlarvt, deren Optimalwert also übereinstimmt. □

Die folgende Aussage wird (bei nicht notwendig linearen Programmen) *Satz von Kuhn-Tucker* genannt.

Korollar 1.10 *Seien $C \subset \mathbb{R}^n$ und $K \subset \mathbb{R}^m$ polyedrische Kegel. Hiermit betrachte man das lineare Programm*

(P) $\qquad$ *Minimiere* $\quad c^T x \quad$ *auf* $\quad M := \{x \in \mathbb{R}^n : x \in C,\ Ax - b \in K\}.$

Ein $x^* \in M$ ist genau dann eine Lösung von (P), wenn ein $y^* \in \mathbb{R}^m$ mit

$$(*) \quad y^* \in K^+, \qquad c - A^T y^* \in C^+, \qquad (y^*)^T(Ax^* - b) = 0, \qquad (x^*)^T(c - A^T y^*) = 0$$

existiert.

Beweis: Ist $x^* \in M$ eine Lösung von (P), so existiert wegen des starken Dualitätssatzes eine Lösung y^* des zu (P) dualen linearen Programms

(D) $\qquad$ Maximiere $\quad b^T y \quad$ auf $\quad N := \{y \in \mathbb{R}^m : y \in K^+,\ c - A^T y \in C^+\}$

und es ist $c^T x^* = b^T y^*$. Wegen

$$0 \leq (y^*)^T(Ax^* - b) = (x^*)^T A^T y^* - c^T x^* = -(x^*)^T(c - A^T y^*) \leq 0$$

genügt y^* den Bedingungen $(*)$. Genügt $y^* \in \mathbb{R}^m$ umgekehrt den Bedingungen $(*)$ und ist x^* primal zulässig, so folgt $b^T y^* = c^T x^*$, der schwache Dualitätssatz liefert, dass x^*

eine Lösung von (P) (und y^* eine Lösung des hierzu dualen linearen Programms (D)) ist. □

Beispiel: Im Anschluss an den schwachen Dualitätssatz hatten wir das zum Produktionsplanungsproblem

$$\text{Maximiere} \quad p^T x \quad \text{unter den Nebenbedingungen} \quad x \geq 0, \quad Ax \leq b$$

duale Programm

$$\text{Minimiere} \quad b^T y \quad \text{unter den Nebenbedingungen} \quad y \geq 0, \quad A^T y \geq p$$

als Aufgabe für einen Konkurrenten interpretiert. Der schwache Dualitätssatz sagt aus: Ist $x \in \mathbb{R}^n$ ein zulässiger Produktionsplan für den Betrieb und $y \in \mathbb{R}^m$ ein akzeptables Angebot des Konkurrenten, so ist $p^T x \leq b^T y$. D. h. der Betrieb kann keinen größeren Reingewinn machen als den Betrag, den er bei einem akzeptablen Angebot vom Konkurrenten erhalten würde (er erspart sich sogar die Suche nach einem optimalen Produktionsplan). Dagegen sagt der starke Dualitätssatz aus, dass der maximale Reingewinn des Betriebes gleich den minimalen Kosten des Konkurrenten sind (wenn zulässige Produktionspläne und akzeptable Angebote existieren). Ferner gilt: Ist x^* ein optimaler, zulässiger Produktionsplan mit einem Reingewinn $p^T x^*$, so gibt es ein für den Konkurrenten optimales, zulässiges Angebot mit den Kosten $b^T y^* = p^T x^*$. Notwendigerweise gelten die sogenannten *Gleichgewichtsbedingungen*

$$(A^T y^* - p)^T x^* = 0, \qquad (b - Ax^*)^T y^* = 0.$$

Wird in einem optimalen Produktionsplan x^* das j-te Produkt hergestellt, ist also $x_j^* > 0$, so ist notwendig $(A^T y^*)_j = p_j$. Wird die Kapazitätsbeschränkung für das i-te Hilfsmittel in einem optimalen Produktionsplan nicht voll ausgeschöpft, ist also $(Ax^*)_i < b_i$, so ist notwendig $y_i^* = 0$, der Konkurrent wird daher in einem für ihn optimalen Angebot für das i-te Hilfsmittel nichts bezahlen. □

Lineare Programme zeichnen sich gegenüber wesentlich allgemeineren Aufgaben dadurch aus, dass bei ihnen ein *strikt komplementäres, optimales Paar* existiert. Dies wird im folgenden Satz, den wir nur für ein lineares Programm in Standardform angeben, präzisiert. Er geht auf Goldman-Tucker (1956) zurück.

Satz 1.11 *Die zueinander dualen linearen Programme*

(P) $$\textit{Minimiere} \quad c^T x \quad \textit{auf} \quad M := \{x \in \mathbb{R}^n : x \geq 0,\ Ax = b\}$$

und

(D) $$\textit{Maximiere} \quad b^T y \quad \textit{auf} \quad N := \{y \in \mathbb{R}^m : A^T y \leq c\}$$

seien zulässig. Mit M_{opt} bzw. N_{opt} seien die (nichtleeren) Lösungsmengen von (P) bzw. (D) bezeichnet. Dann existiert ein Paar $(x^*, y^*) \in M_{\text{opt}} \times N_{\text{opt}}$ mit $x^* + c - A^T y^* > 0$.

Beweis: Zunächst wollen wir uns überlegen, dass es genügt, die folgende Hilfsbehauptung zu beweisen:

- Sei $k \in \{1, \ldots, n\}$. Existiert kein $y \in N_{\text{opt}}$ mit $(c - A^T y)_k > 0$, so existiert ein $x \in M_{\text{opt}}$ mit $x_k > 0$.

Ist diese Aussage bewiesen, so existieren für $k = 1, \ldots, n$ Paare $(x^{(k)}, y^{(k)}) \in M_{\text{opt}} \times N_{\text{opt}}$ mit $(x^{(k)} + c - A^T y^{(k)})_k > 0$. Durch

$$x^* := \frac{1}{n} \sum_{k=1}^{n} x^{(k)}, \qquad y^* := \frac{1}{n} \sum_{k=1}^{n} y^{(k)}$$

ist das gesuchte Paar (x^*, y^*) gefunden.

Es genügt also, die obige Hilfsbehauptung zu beweisen. Gibt es bei gegegebenem $k \in \{1, \ldots, n\}$ kein $y \in N_{\text{opt}}$ mit $(c - A^T y)_k > 0$, so gilt die Implikation

$$y \in N, \quad b^T y \geq \max(\mathrm{D}) \Longrightarrow (-Ae_k)^T y \leq -c_k,$$

wobei e_k den k-ten Einheitsvektor im $\mathbb{R}^n$ bedeutet. Dann hat das lineare Programm

$$\text{Maximiere} \quad (-Ae_k)^T y \quad \text{auf} \quad N_{\text{opt}} = \left\{ y \in \mathbb{R}^m : \begin{pmatrix} A^T \\ -b^T \end{pmatrix} y \leq \begin{pmatrix} c \\ -\max(\mathrm{D}) \end{pmatrix} \right\}$$

eine Lösung mit einem Wert, der nicht größer als $-c_k$ ist. Der starke Dualitätssatz liefert, dass das dazu duale Programm

$$\begin{cases} \text{Minimiere} \quad c^T z - \lambda \max(\mathrm{D}) \quad \text{unter den Nebenbedingungen} \\ \qquad\qquad z \geq 0, \quad \lambda \geq 0, \quad Az - \lambda b = -Ae_k \end{cases}$$

eine Lösung (z^*, λ^*) mit dem gleichen Wert besitzt, so dass also

$$c^T z^* - \lambda^* \max(\mathrm{D}) \leq -c_k.$$

Ist $\lambda^* = 0$, so hat man ein z^* mit $z^* \geq 0$, $Az^* = -Ae_k$ und $c^T z^* \leq -c_k$ gefunden. Definiert man daher $x^{(k)} := x + z^* + e_k$ mit einem beliebigen $x \in M_{\text{opt}}$, so ist $x^{(k)} \in M$, $c^T x^{(k)} \leq c^T x$, folglich $x^{(k)} \in M_{\text{opt}}$, und $x_k^{(k)} \geq 1$. Ist dagegen $\lambda^* > 0$, so definiere man $x^{(k)} := (z^* + e_k)/\lambda^*$. Wieder ist $x^{(k)} \in M$, ferner $c^T x^{(k)} \leq \max(\mathrm{D}) = \min(\mathrm{P})$, also $x^{(k)} \in M_{\text{opt}}$. Wegen $x_k^{(k)} \geq 1/\lambda^*$ hat man auch in diesem Falle eine Lösung von (P) gefunden, deren k-te Komponente positiv ist. Insgesamt ist der Satz bewiesen. □

2.1.4 Sensitivitätsanalyse

In der Sensitivitätsanalyse untersucht man, wie sich das "Wackeln" in den Daten eines Problems auf die Lösung auswirkt. Bei K. Neumann, M. Morlock (1993, S. 119) wird formuliert:

- Die Aufgabe der Sensitivitätsanalyse lässt sich allgemein folgendermaßen beschreiben. Wir legen wieder das Standardproblem der linearen Optimierung

$$\text{(L)} \qquad \text{Minimiere} \quad c^T x \quad \text{u. d. N.} \quad Ax = b,\ x \geq 0$$

zugrunde, wobei A eine $m \times n$-Matrix mit $\text{Rang}\,(A) = m < n$, $b \in \mathbb{R}^m$ und $c \in \mathbb{R}^n$. Kennzeichnen wir die (additiven) Änderungen der Eingabedaten durch Δ, so geht (L) in das Problem

$$(\mathrm{L}_\Delta) \qquad \text{Minimiere} \quad (c+\Delta c)^T x \quad \text{u. d. N.} \quad (A+\Delta A)x = b + \Delta b,\ x \geq 0$$

über. Im Rahmen der Sensitivitätsanalyse ist nun zu untersuchen, wie groß die Änderungen ΔA, Δb und Δc sein können, ohne eine optimale Lösung x^* von (L) qualitativ zu ändern.

Dies ist natürlich nur ziemlich unscharf formuliert. Denn was soll "ohne eine optimale Lösung x^* von (L) qualitativ zu ändern" eigentlich heißen? Genauer will man zumindest für kleine Störungen in den Daten die Auswirkungen auf den Optimalwert abschätzen. Man denke z. B. an die entsprechende Problemstellung bei linearen Gleichungssystemen oder linearen Ausgleichsproblemen.

Beispiel: Wir betrachten die lineare Optimierungsaufgabe[7]

$$(\mathrm{P}_0) \qquad \text{Minimiere} \quad c^T x \quad \text{auf} \quad M := \{x \in \mathbb{R}^5 : x \geq 0,\ Ax = b\}$$

in Standardform, wobei

$$A := \begin{pmatrix} -1 & 1 & 1 & 0 & 0 \\ 1 & 2 & 0 & 1 & 0 \\ 2 & 1 & 0 & 0 & 1 \end{pmatrix}, \qquad b := \begin{pmatrix} 2 \\ 6 \\ 6 \end{pmatrix}, \qquad c := \begin{pmatrix} -1 \\ -1 \\ 0 \\ 0 \\ 0 \end{pmatrix}.$$

Die Lösung dieses (ungestörten) Problems ist $x_0 = (2,2,2,0,0)^T$, der Optimalwert ist $\min(\mathrm{P}_0) = -4$. Wir wollen die rechte Seite b in der letzten Komponente stören und die Auswirkungen auf die Lösung bzw. den Optimalwert untersuchen. Hierzu wenden wir *Mathematica* an. Nach

```
tildea={{-1,1,1,0,0},{1,2,0,1,0},{2,1,0,0,1},{1,-1,-1,0,0},
{-1,-2,0,-1,0},{-2,-1,0,0,-1}};
tildeb={2,6,6,-2,-6,-6};
c={-1,-1,0,0,0};d={0,0,-1,0,0,1};
w[t_]:=c.LinearProgramming[c,tildea,tildeb+t*d];
Plot[w[t],{t,-2,6}]
```

erhalten wir den in Abbildung 2.2 angegebenen Plot für den Optimalwert $\min(\mathrm{P}_t)$ des linearen Programms

$$(\mathrm{P}_t) \qquad \text{Minimiere} \quad c^T x \quad \text{auf} \quad M_t := \{x \in \mathbb{R}^5 : x \geq 0,\ Ax = b + td\}$$

mit $d := (0,0,-1)^T$ auf dem Intervall $[-2,6]$. Offensichtlich ist $\min(\mathrm{P}_t)$ stückweise linear und es stellt sich natürlich die Frage, ob das ein Zufall ist. Durch ein ähnliches

[7] Dieses Beispiel haben wir K. Neumann, M. Morlock (1993, S. 129 ff.) entnommen.

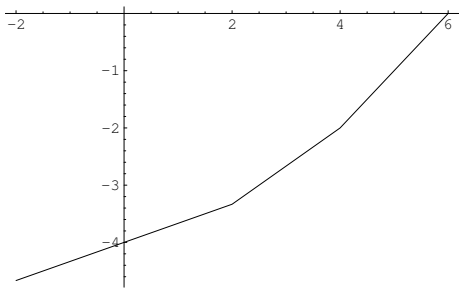


Abbildung 2.2: $w(t) := \min(\mathrm{P}_t)$ auf $[-2, 6]$

Experiment stellt man fest, dass auch die Komponenten der Lösung $x(t)$ von (P_t) stückweise linear in t sind. Das zu (P_t) duale lineare Programm ist

$$(\mathrm{D}_t) \qquad \text{Maximiere} \quad (b+td)^T y \quad \text{auf} \quad N := \{y \in \mathbb{R}^3 : A^T y \le c\}.$$

Für $t > 6$ ist die letzte Komponente von $b + td$ negativ, woraus man sehr leicht (der Vektor $(0, 0, y_3)^T$ ist für alle $y_3 \le -1$ dual zulässig) auf $\sup(\mathrm{D}_t) = +\infty$ bzw. die Unzulässigkeit von (P_t) schließt. Entsprechendes kann für $t < -2$ ausgesagt werden. □

Gewöhnlich werden Erkenntnisse über das (revidierte) Simplexverfahren eingesetzt, um eine Sensitivitätsanalyse durchzuführen, siehe z. B. V. Chvátal (1983)[8]. Unabhängig hiervon zeigen wir:

Satz 1.12 *Bei gegebenen $A \in \mathbb{R}^{m\times n}$, $b \in \mathbb{R}^m$, $c \in \mathbb{R}^n$ und $d \in \mathbb{R}^m$ betrachte man für $t \in [0, 1]$ das lineare Programm in Standardform*

$$(\mathrm{P}_t) \qquad \textit{Minimiere} \quad c^T x \quad \textit{auf} \quad M_t := \{x \in \mathbb{R}^n : x \ge 0,\ Ax = b + td\},$$

dessen duales Programm durch

$$(\mathrm{D}_t) \qquad \textit{Maximiere} \quad (b+td)^T y \quad \textit{auf} \quad N := \{y \in \mathbb{R}^m : A^T y \le c\}$$

gegeben ist. Es wird vorausgesetzt, dass M_0, M_1 und N nichtleer sind. Dann gilt:

1. *Für alle $t \in [0, 1]$ sind (P_t) und (D_t) lösbar.*
2. *Die Aufgabe*

   $(\mathrm{D}_{\mathrm{opt}})$ *Maximiere* $d^T y_0$ *auf* $N_{\mathrm{opt}} := \{y_0 \in \mathbb{R}^m : y_0$ *ist Lösung von* $(D_0)\}$

   besitzt eine Lösung y_0^*.

[8] V. Chvátal (1983) *Linear Programming*. W. H. Freeman and Company, New York.

3. *Ist y_0^* eine Lösung von (D_{opt}), so ist y_0^* für alle hinreichend kleinen $t > 0$ auch eine Lösung von (D_t).*

4. *Für alle hinreichend kleinen $t > 0$ ist*

$$\min(P_t) = \min(P_0) + t \max(D_{\text{opt}}) = \min(P_0) + t\, d^T y_0^*,$$

wobei y_0^* eine Lösung von (D_{opt}) ist.

Beweis: Da $M_0, M_1 \neq \emptyset$ ist $M_t \neq \emptyset$ für alle $t \in [0,1]$. Folglich sind (P_t) und (D_t) für alle $t \in [0,1]$ zulässig, wegen des starken Dualitätssatzes also auch lösbar.

Die Aufgabe (D_{opt}) ist selbst ein lineares Programm, da man sie z. B. in der Form

$$\text{Maximiere} \quad d^T y_0 \quad \text{auf} \quad N_{\text{opt}} = \{y_0 \in \mathbb{R}^m : b^T y_0 \geq \max(D_0),\ A^T y_0 \leq c\}$$

schreiben kann. Für $y_0 \in N_{\text{opt}}$ ist

$$\max(D_1) \geq (b+d)^T y_0 = \max(D_0) + d^T y_0.$$

Also ist (D_{opt}) zulässig und $\sup(D_{\text{opt}}) < +\infty$, aus dem starken Dualitätssatz folgt die Existenz eine Lösung y_0^* von (D_{opt}).

Sei y_0^* eine Lösung von (D_{opt}). Wegen (des notwendigen Teiles) des Satzes von Kuhn-Tucker existiert ein Paar $(\lambda, p) \in \mathbb{R} \times \mathbb{R}^n$ mit

$$\begin{pmatrix} \lambda \\ p \end{pmatrix} \geq 0, \quad -d + Ap - \lambda b = 0, \qquad p^T(c - A^T y_0^*) = 0.$$

Um nachzuweisen, dass y_0^* für alle hinreichend kleinen $t > 0$ eine Lösung von (D_t) ist, ist (wegen des hinreichenden Teiles des Satzes von Kuhn-Tucker) für alle hinreichend kleinen $t > 0$ die Existenz eines Vektors $p_t \in \mathbb{R}^n$ mit

$$(*) \qquad p_t \geq 0, \qquad -(b + td) + Ap_t = 0, \qquad p_t^T(c - A^T y_0^*) = 0$$

zu zeigen. Sei x_0 eine Lösung von (P_0). Man setzt $p_t := (1 - t\lambda)x_0 + tp$ und zeigt, dass p_t für alle $t \in (0, 1/\lambda)$ (also für $\lambda = 0$ für alle $t > 0$) der Bedingung $(*)$ genügt. Damit ist auch dieser Teil des Satzes bewiesen.

Wegen des gerade eben bewiesenen Teiles des Satzes ist y_0^* für alle hinreichend kleinen $t > 0$ eine Lösung von (D_t). Für diese t ist

$$\min(P_t) = \max(D_t) = (b+td)^T y_0^* = \max(D_0) + t \max(D_{\text{opt}}) = \min(P_0) + td^T y_0^*.$$

Damit ist auch der letzte Teil des Satzes bewiesen. □

Bemerkung: Ist unter den Voraussetzungen des letzten Satzes das duale Problem (D_0) *eindeutig* lösbar bzw. ist $N_{\text{opt}} = \{y_0^*\}$ einpunktig, so bestimmt $d^T y_0^*$ die Sensitivität des Optimalwertes von (P_t), da $\min(P_t) = \min(P_0) + td^T y_0^*$ für alle hinreichend kleinen $t > 0$. Zu dem im vorigen Beispiel angegebenen Problem (P_0) erhält man $y_0^* = (0, -\frac{1}{3}, -\frac{1}{3})^T$ als Lösung des zu (P_0) dualen linearen Programms (D_0) (ein Beweis kann leicht mit

Hilfe des schwachen Dualitätssatzes erfolgen). Mit $d = (0, 0, -1)$ ist daher $\min(\mathrm{P}_t) = -4 + \frac{1}{3}t$ für alle hinreichend kleinen $t > 0$. □

Beispiel: Wir betrachten die im Kostenvektor gestörte lineare Optimierungsaufgabe in Standardform

$$(\mathrm{P}_t) \qquad \text{Minimiere} \quad (c + td)^T x \quad \text{auf} \quad M := \{x \in \mathbb{R}^5 : x \geq 0,\ Ax = b\},$$

wobei wie im letzten Beispiel

$$A := \begin{pmatrix} -1 & 1 & 1 & 0 & 0 \\ 1 & 2 & 0 & 1 & 0 \\ 2 & 1 & 0 & 0 & 1 \end{pmatrix}, \qquad b := \begin{pmatrix} 2 \\ 6 \\ 6 \end{pmatrix}, \qquad c := \begin{pmatrix} -1 \\ -1 \\ 0 \\ 0 \\ 0 \end{pmatrix}$$

und diesmal $d := (2, -1, 0, 0, 0)^T$ (auch dieses Beispiel ist K. Neumann, M. Morlock (1993, S. 129) entnommen. Wir wollen wieder *Mathematica* benutzen, um $\min(\mathrm{P}_t)$ zu plotten. Nach

```
tildea={{-1,1,1,0,0},{1,2,0,1,0},{2,1,0,0,1},{1,-1,-1,0,0},
{-1,-2,0,-1,0},{-2,-1,0,0,-1}};
tildeb={2,6,6,-2,-6,-6};
c={-1,-1,0,0,0};d={2,-1,0,0,0};
w[t_]:=(c+t*d).LinearProgramming[c+t*d,tildea,tildeb];
Plot[w[t],{t,-2,6}]
```

erhalten wir den in Abbildung 2.3 links angegebenen Plot von $\min(\mathrm{P}_t)$ auf $[-2, 6]$

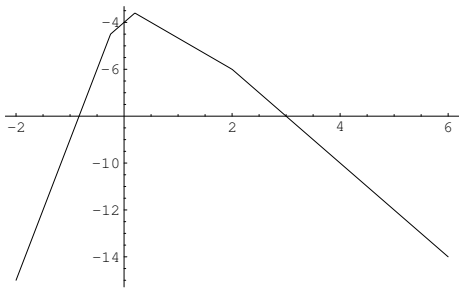


Abbildung 2.3: $\min(\mathrm{P}_t)$ auf $[-2, 6]$

(wobei diesmal das Intervall $[-2, 6]$ durch nichts ausgezeichnet ist). In Abbildung 2.4 geben wir die ersten beiden Komponenten der Lösung von (P_t) an. Wir erkennen, dass diese stückweise konstant sind. Wer kann dies erklären? Die einfachste Erklärung erfolgt sicherlich mit Hilfe des Simplexverfahrens. □

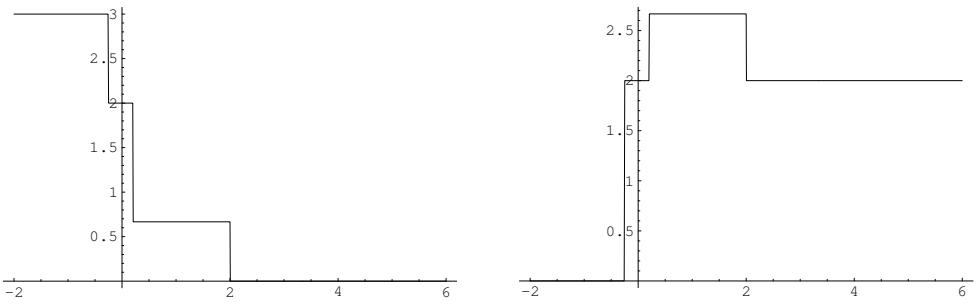


Abbildung 2.4: Die Komponenten $x_1(t)$ und $x_2(t)$ der Lösung von (P_t)

2.1.5 Aufgaben

1. Eine nichtleere, abgeschlossene, konvexe Menge $C \subset \mathbb{R}^n$ ist der Durchschnitt aller abgeschlossenen Halbräume, die C enthalten.

   Hinweis: Man wende den starken Trennungssatz an.

2. In[9] zwei Rangierbahnhöfen A und B stehen 18 bzw. 12 leere Güterwagen. In den drei Bahnhöfen R, S und T werden 11, 10 bzw. 9 Güterwagen zum Verladen von Waren benötigt. Die Distanzen in km von den Rangierbahnhöfen zu den Bahnhöfen sind durch

	R	S	T
A	5	4	9
B	7	8	10

   gegeben. Die Güterwagen sind so zu leiten, dass die totale Anzahl der durchfahrenen Leerkilometer minimal ist. Man formuliere diese Aufgabe als lineare Optimierungsaufgabe und löse sie mit einem mathematischen Anwendersystem.

3. Seien α, β und γ Winkel in einem Dreieck mit $90^\circ \geq \alpha \geq \beta \geq \gamma \geq 0$ und natürlich $\alpha + \beta + \gamma = 180^\circ$. Unter allen solchen Winkeln bestimme man diejenigen, für die

$$g(\alpha, \beta, \gamma) := \min\{\gamma, \beta - \gamma, \alpha - \beta, 90^\circ - \alpha\}$$

   maximal ist. Hierzu formuliere man diese Aufgabe als ein lineares Programm.

   Hinweis: Die obige Aufgabenstellung steht im engen Zusammenhang mit einem (sehr witzigen) Aufsatz von B. Tergan (1980)[10], in dem gezeigt wird, dass es (bis auf Ähnlichkeit) genau ein allgemeines, spitzwinkliges Dreieck gibt, dessen Winkel durch $\alpha^* := 75^\circ$, $\beta^* := 60^\circ$ und $\gamma^* := 45^\circ$ gegeben sind.

[9] Diese Aufgabe ist ein Beispiel in dem Lehrbuch über Numerische Mathematik von H. R. Schwarz.

[10] Siehe den Anhang 2 bei

F. Wille (1982) *Humor in der Mathematik*. Vandenhoek & Ruprecht, Göttingen.

4. Die[11] Funktion $f(t) := \cos((\pi/2)t)$ soll im Intervall $[0,1]$ durch ein Polynom vierten Grades im Tschebyscheffschen Sinne approximiert werden bezüglich der elf äquidistanten Abszissen $t_i := (i-1)/10$, $i = 1, \ldots, 11$. Gesucht ist also eine Lösung der Aufgabe

$$\text{Minimiere} \quad f(a) := \max_{i=1,\ldots,11} \left| \sum_{j=1}^{5} a_j t_i^{j-1} - \cos((\pi/2)t_i) \right|, \quad a \in \mathbb{R}^5.$$

Man führe diese Aufgabe in eine lineare Optimierungsaufgabe über und löse sie mit einem mathematischen Anwendersystem. Anschließend plotte man den Fehler

$$d(t) := \sum_{j=1}^{5} a_j t^{j-1} - \cos((\pi/2)t)$$

über dem Intervall $[0,1]$.

5. Sei $A \in \mathbb{R}^{m\times n}$. Man beweise den Alternativsatz von Gordan: Genau eine der beiden Aussagen

(I) $\qquad Ax = 0, \quad x \geq 0, \quad x \neq 0 \qquad$ hat eine Lösung $x \in \mathbb{R}^n$

bzw.

(II) $\qquad A^T y > 0 \qquad$ hat eine Lösung $y \in \mathbb{R}^m$

ist richtig.

6. Sei $A \in \mathbb{R}^{m\times n}$, $b \in \mathbb{R}^m$. Man beweise den Alternativsatz von Gale: Genau eine der beiden Aussagen

(I) $\qquad Ax \leq b \qquad$ hat eine Lösung $x \in \mathbb{R}^n$

bzw.

(II) $\qquad A^T y = 0, \quad y \geq 0, \quad b^T y < 0 \qquad$ hat eine Lösung $y \in \mathbb{R}^m$

ist richtig.

7. Gegeben sei das sogenannte *Quotientenprogramm*

(P) $\qquad \text{Minimiere} \quad \dfrac{c^T x + c_0}{d^T x + d_0} \quad \text{auf} \quad M := \{x \in \mathbb{R}^n : x \geq 0,\ Ax = b\}.$

Es wird vorausgesetzt, dass $d^T x + d_0 > 0$ für alle $x \in M$. Der Optimierungsaufgabe (P) ordne man das lineare Programm

$$(\hat{\mathrm{P}}) \quad \begin{cases} \text{Minimiere} \quad \begin{pmatrix} c \\ c_0 \end{pmatrix}^T \begin{pmatrix} z \\ z_0 \end{pmatrix} \quad \text{auf} \\ \hat{M} := \left\{ (z, z_0) \in \mathbb{R}^n \times \mathbb{R} : \begin{pmatrix} z \\ z_0 \end{pmatrix} \geq 0,\ \begin{pmatrix} A & -b \\ d^T & d_0 \end{pmatrix} \begin{pmatrix} z \\ z_0 \end{pmatrix} = \begin{pmatrix} 0 \\ 1 \end{pmatrix} \right\}. \end{cases}$$

Man zeige: Ist (z^*, z_0^*) eine Lösung von $(\hat{\mathrm{P}})$ mit $z_0^* > 0$, so ist $x^* := (1/z_0^*)z^*$ eine Lösung von (P).

8. Sei $A \in \mathbb{R}^{m\times n}$, $b \in \mathbb{R}^m$. Die Menge $M := \{x \in \mathbb{R}^n : x \geq 0,\ Ax \geq b\}$ sei nichtleer. Man zeige, dass M genau dann beschränkt ist, wenn es ein $u \in \mathbb{R}^m$ mit $u \geq 0$ und $A^T u < 0$ gibt. Wie lautet die entsprechende Aussage, wenn $M := \{x \in \mathbb{R}^n : x \geq 0,\ Ax = b\}$?

[11] Diese Aufgabe ist als Beispiel im Lehrbuch von H. R. Schwarz über Numerische Mathematik enthalten. Hierdurch ist es möglich, die erhaltenen Ergebnisse zu vergleichen.

2.2 Das Simplexverfahren

Wir wollen uns möglichst kurz fassen, da bei den Hörern sicherlich Vorkenntnisse zum Simplexverfahren vorhanden sind. Zum Teil werden wir daher auf Beweise relativ bekannter Aussagen verzichten. Wie allgemein üblich werden wir uns auf das lineare Programm in Standardform

$$\text{(P)} \qquad \text{Minimiere} \quad c^T x \quad \text{auf} \quad M := \{x \in \mathbb{R}^n : x \geq 0,\ Ax = b\},$$

wobei $A = (a_{ij}) \in \mathbb{R}^{m\times n}$, $b = (b_i) \in \mathbb{R}^m$ und $c = (c_j) \in \mathbb{R}^n$, konzentrieren[12] und erst später einige Hinweise auf die Behandlung allgemeinerer Fälle geben.

2.2.1 Ecken und Basislösungen

Ist $C \subset \mathbb{R}^n$ konvex, so heißt ein Punkt $x \in C$ bekanntlich eine *Ecke* von C, wenn sich x nicht als Konvexkombination von zwei anderen (d. h. von x verschiedenen) Punkten aus C darstellen lässt, also die Implikation

$$x_1, x_2 \in C \setminus \{x\},\ \lambda \in (0,1) \Longrightarrow x \neq (1-\lambda)x_1 + \lambda x_2$$

gilt.

Für einen Beweis des folgenden Lemmas verweisen wir z. B. auf J. Werner (1992, S. 89)[13].

Lemma 2.1 *Sei* $M := \{x \in \mathbb{R}^n : x \geq 0,\ Ax = b\}$ *mit* $A = (\ a_1\ \cdots\ a_n\ ) \in \mathbb{R}^{m\times n}$ *und* $b \in \mathbb{R}^m$. *Dann ist* $x \in M$ *genau dann eine Ecke von* M, *wenn die zu positiven Komponenten von* x *gehörenden Spalten von* A *linear unabhängig sind, wenn also die Spalten* $\{a_j\}_{j\in B(x)}$ *mit* $B(x) := \{j \in \{1,\ldots,n\} : x_j > 0\}$ *linear unabhängig sind. Ist ferner* $M \neq \emptyset$, *so besitzt* M *mindestens eine, aber höchstens endlich viele Ecken.*

Beispiel: Sei $m := 4$, $n := 7$ und

$$A := \begin{pmatrix} 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 3 & 1 & 0 & 0 & 0 & 1 \end{pmatrix}, \qquad b := \begin{pmatrix} 4 \\ 2 \\ 3 \\ 6 \end{pmatrix}.$$

Eine spezielle Ecke von $M := \{x \in \mathbb{R}^n : x \geq 0,\ Ax = b\}$ erkennt man mit Hilfe des vorigen Lemmas sofort, nämlich $x = (0,0,0,4,2,3,6)^T$. Die zugehörigen linear

[12] Bei

J. L. Nazareth (1987) *Computer Solution of LinearPrograms*. Oxford University Press, New York-London

wird allerdings von Anfang an ein lineares Programm der Form

$$\text{Minimiere} \quad c^T x \quad \text{auf} \quad M := \{x \in \mathbb{R}^n : l \leq x \leq u,\ Ax = b\}$$

betrachtet.

[13] J. Werner (1992) *Numerische Mathematik 2*. Vieweg-Verlag, Braunschweig-Wiesbaden.

unabhängigen Spalten sind die Einheitsvektoren in den letzten vier Spalten von A. Eine weitere Ecke von A ist $x = (2, 2, 0, 0, 0, 3, 0)^T$. Man spricht von einer *entarteten Ecke*, da sie weniger als $m = 4$ positive Komponenten enthält. □

Als Folgerung aus Lemma 2.1 erhält man einen berühmten Satz, der auf Birkhoff und von Neumann zurückgeht. Hierbei erinnern wir daran, dass eine *Permutationsmatrix* eine (quadratische) Matrix ist, die in jeder Zeile und jeder Spalte genau eine Eins und sonst nur Nullen besitzt. Ferner heißt eine (quadratische) Matrix *doppeltstochastisch*, wenn ihre Einträge nichtnegativ sind und alle Zeilen- und Spaltensummen gleich Eins sind. Offenbar bilden die doppeltstochastischen $n \times n$-Matrizen eine konvexe Menge in $\mathbb{R}^{n\times n}$.

Satz 2.2 *Eine Matrix ist genau dann eine Ecke der Menge der doppeltstochastischen $n \times n$-Martrizen, wenn sie eine Permutationsmatrix ist.*

Beweis: Sei P eine Permutationsmatrix, X_1, X_2 zwei doppeltstochastische Matrizen, $\lambda \in (0, 1)$ und $P = (1 - \lambda)X_1 + \lambda X_2$. Sei $(i, j) \in \{1, \ldots, n\} \times \{1, \ldots, n\}$ eine Position mit $p_{ij} = 0$. Wegen $P = (1-\lambda)X_1 + \lambda X_2$ müssen auch X_1 und X_2 in der Position (i, j) einen Null-Eintrag haben. Hat P in der Position (i, j) dagegen einen Eins-Eintrag, so ist

$$1 = \underbrace{(1-\lambda)}_{>0}\underbrace{(X_1)_{ij}}_{\leq 1} + \underbrace{\lambda}_{>0}\underbrace{(X_2)_{ij}}_{\leq 1} \leq 1,$$

so dass insgesamt $X_1 = X_2 = P$, also P eine Ecke ist.

Umgekehrt sei nun $P \in \mathbb{R}^{n\times n}$ eine Ecke der Menge der doppeltstochastischen $n \times n$-Matrizen. Wie beim Transportproblem schreibe man eine $n \times n$-Matrix $X = (x_{ij})$ als einen Vektor

$$x = (x_{11}, \ldots, x_{1n}, x_{21}, \ldots, x_{2n}, \ldots, x_{n1}, \ldots, x_{nn})^T.$$

Dann ist die Menge C_n der doppeltstochastischen Matrizen gegeben durch

$$C_n = \{x \in \mathbb{R}^{n^2} : x \geq 0,\ Ax = b\},$$

wobei

$$A := \begin{pmatrix} e^T & 0^T & \cdots & 0^T \\ 0^T & e^T & & 0^T \\ \vdots & \vdots & \ddots & \vdots \\ 0^T & 0^T & \cdots & e^T \\ I & I & \cdots & I \end{pmatrix}, \qquad b := \begin{pmatrix} e \\ e \end{pmatrix}.$$

Dann ist A eine $2n \times n^2$-Matrix, folglich (wir nehmen $n \geq 2$ an) Rang $(A) \leq 2n$. Nun ist aber die Summe der ersten n Zeilen von A gleich der Summe der letzten n Zeilen, also ist Rang $(A) \leq 2n-1$. Angenommen, die Matrix P hätte in jeder Zeile mindestens zwei positive Einträge. Die entsprechenden Spalten in A, also mindestens $2n$, wären linear unabhängig, was einen Widerspruch zu Rang $(A) \leq 2n-1$ bedeutet. Also gibt es mindestens eine Zeile in P, in der genau eine 1 und sonst nur Nullen stehen. O. B. d. A. sei $p_{11} = 1$, also alle weiteren Einträge in der ersten Zeile und ersten Spalte Nullen. Der untere $(n-1) \times (n-1)$-Block von P ist eine doppeltstochastische Matrix, die Ecke der

Menge der $(n-1) \times (n-1)$-doppeltstochastischen Matrizen ist. Nach endlich vielen Schritten hat man die Aussage auf den Fall $n = 2$ (oder $n = 1$) zurückgeführt, wofür sie richtig ist. □

Der folgende Satz ist wohlbekannt (siehe z. B. J. WERNER (1992, S. 89)).

Satz 2.3 *Sei $M := \{x \in \mathbb{R}^n : x \geq 0,\ Ax = b\} \neq \emptyset$, wobei $A \in \mathbb{R}^{m \times n}$ und $b \in \mathbb{R}^m$. Mit $\{v_i : i \in I\}$ werde die nichtleere, endliche Menge der Ecken von M bezeichnet. Dann lässt sich jeder Punkt $x \in M$ in der Form*

$$x = \sum_{i \in I} \lambda_i v_i + d$$

darstellen, wobei

$$\lambda_i \geq 0 \quad (i \in I), \qquad \sum_{i \in I} \lambda_i = 1$$

und $d \geq 0$, $Ad = 0$. Insbesondere gilt: Ist M beschränkt, so lässt sich jeder Punkt aus M als Konvexkombination der endlich vielen Ecken von M darstellen.

Der folgende Satz ist grundlegend für lineare Optimierungsaufgaben (siehe z. B. J. WERNER (1992, S. 91)). Für lineare Programme in Standardform (und damit für "allgemeine" lineare Programme) hat man einen neuen Beweis des Existenzsatzes.

Satz 2.4 *Gegeben sei das lineare Programm in Standardform*

$$\text{(P)} \qquad \text{Minimiere} \quad c^T x \quad \text{auf} \quad M := \{x \in \mathbb{R}^n : x \geq 0,\ Ax = b\},$$

wobei $A \in \mathbb{R}^{m \times n}$, $b \in \mathbb{R}^m$, $c \in \mathbb{R}^n$ und $M \neq \emptyset$. Dann gilt: Entweder besitzt (P) eine der endlich vielen Ecken von M als Lösung, oder es ist $\inf(\text{P}) = -\infty$, die Zielfunktion von (P) also auf der Menge der zulässigen Lösungen nicht nach unten beschränkt.

Weiter sei die lineare Optimierungsaufgabe Normalform

$$\text{(P)} \qquad \text{Minimiere} \quad c^T x \quad \text{auf} \quad M := \{x \in \mathbb{R}^n : x \geq 0,\ Ax = b\}$$

gegeben. Zusätzlich setzen wir voraus, dass Rang $(A) = m$. Zumindestens theoretisch ist dies keine Einschränkung, da ja notfalls redundante Gleichungen gestrichen werden können. Im folgenden bezeichnen wir mit $a_j \in \mathbb{R}^m$ die j-te Spalte von A, $j = 1, \ldots, n$, es sei also $A = (\ a_1 \ \cdots \ a_n\ )$.

Um eine Vektor-Matrix-Schreibweise benutzen zu können, wird es zweckmäßig sein, die folgenden Bezeichnungen zu benutzen.

Ist

$$B := \{j(1), \ldots, j(m)\} \subset \{1, \ldots, n\}$$

eine m-punktige Teilmenge von $\{1, \ldots, n\}$, so sei $A_B \in \mathbb{R}^{m \times m}$ definiert durch

$$A_B := (\ a_{j(1)} \ \cdots \ a_{j(m)}\ ),$$

d. h. die Spalten von A_B werden aus den zur Indexmenge B gehörenden Spalten von A (in der durch B "festgelegten Reihenfolge") gebildet. Entsprechend ist für einen Vektor $z = (z_j) \in \mathbb{R}^n$ der Vektor $z_B \in \mathbb{R}^m$ durch

$$z_B := \begin{pmatrix} z_{j(1)} \\ \vdots \\ z_{j(m)} \end{pmatrix}$$

definiert. Entsprechende Bezeichnungen benutzen wir für nicht notwendig m-elementige Teilmengen von $\{1, \ldots, n\}$. Ist also z. B. $N := \{1, \ldots, n\} \setminus B$, so ist

$$Az = \sum_{j=1}^{n} z_j a_j = \sum_{j \in B} z_j a_j + \sum_{j \in N} z_j a_j = A_B z_B + A_N z_N.$$

Nun kommen wir zu einer für das Simplexverfahren entscheidenden Definition.

Definition 2.5 Seien $A = (\ a_1 \ \cdots \ a_n\ ) \in \mathbb{R}^{m \times n}$ mit Rang $(A) = m$ und $b \in \mathbb{R}^m$ gegeben. Ist $B \subset \{1, \ldots, n\}$ eine m-punktige Indexmenge mit der Eigenschaft, dass die zu Indizes aus B gehörenden Spalten von A, also $\{a_j\}_{j \in B}$, linear unabhängig sind bzw. $A_B \in \mathbb{R}^{m \times m}$ nichtsingulär ist, so heißt $x \in \mathbb{R}^n$ mit $x_B := A_B^{-1} b$ und $x_N := 0$ eine *Basislösung* von $Ax = b$ mit den *Basisindizes* B (oder *zur Basis* B). Hierbei ist $N := \{1, \ldots, n\} \setminus B$ die Menge der *Nichtbasisindizes*. Die Basislösung x zur Basis B heißt eine für das lineare Programm in Normalform (P) *zulässige Basislösung*, wenn $x_B \geq 0$ bzw. $x_j \geq 0$ für alle $j \in B$. Eine zulässige Basislösung x zur Basis B heißt *nichtentartet*, wenn $x_B > 0$ bzw. $x_j > 0$ für alle $j \in B$.

Offensichtlich gilt:

Lemma 2.6 *Sei $M := \{x \in \mathbb{R}^n : x \geq 0,\ Ax = b\}$ mit $A \in \mathbb{R}^{m \times n}$, Rang $(A) = m$ und $b \in \mathbb{R}^n$. Dann ist $x \in M$ genau dann eine Ecke von M, wenn x eine zulässige Basislösung von $Ax = b$ zu einer geeigneten Basis $B \subset \{1, \ldots, n\}$ ist.*

Beispiel: Von V. Klee, G. Minty (1972)[14] stammt ein berühmtes Beispiel eines linearen Programms im $\mathbb{R}^d$, bei dem das Simplexverfahren 2^d Ecken durchläuft ehe es in der Lösung endet. Das Simplexverfahren ist also kein *polynomiales* Verfahren und im "worst case" ein schlechtes Verfahren. Das Beispiel von Klee-Minty lautet folgendermaßen: Sei $\epsilon \in (0, \frac{1}{2})$ gegeben, betrachte die Aufgabe

$$\text{Minimiere} \quad -x_d \quad \text{unter den Nebenbedingungen}$$
$$1 \geq x_1 \geq \epsilon, \qquad 1 - \epsilon x_{j-1} \geq x_j \geq \epsilon x_{j-1} \quad (j = 2, \ldots, d).$$

Anschaulich bedeutet diese Aufgabe, dass in einem "gestörten" Einheitswürfel der Punkt mit der größten d-ten Komponente zu bestimmen ist. Für $d = 2$ und $\epsilon = \frac{1}{4}$ erhält man z. B. die in Abbildung 2.5 angegebene Menge zulässiger Lösungen. Nach

[14] V. Klee, G. Minty (1972) "How good is the simplex algorithm?", 159–175, in: *Inequalities III*, ed. O. Shisha, Academic Press, 1972.

Ausführlich (und verständlicher als im Original) ist das Beispiel von Klee-Minty bei C. H. Papadimitriou, K. Steiglitz (1982, S. 166 ff.) dargestellt.

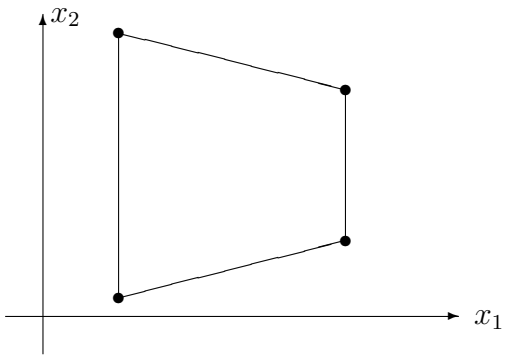


Abbildung 2.5: Das Beispiel von Klee-Minty für $d = 2$

einer Überführung in Normalform durch Einführung von $2d$ Schlupfvariablen $r_1, \ldots, r_d$ und $s_1, \ldots, s_d$ haben die Nebenbedingungen die Form

$$
(*) \qquad \begin{array}{rcll} x_1 - r_1 & = & \epsilon, & \\ x_1 + s_1 & = & 1, & \\ -\epsilon x_{j-1} + x_j - r_j & = & 0, & j = 2, \ldots, d, \\ \epsilon x_{j-1} + x_j + s_j & = & 1, & j = 2, \ldots, d, \\ x_j,\, r_j,\, s_j & \geq & 0, & j = 1, \ldots, d. \end{array}
$$

Der zulässige Bereich kann in der Form $M = \{z \in \mathbb{R}^n : z \geq 0,\ Az = b\}$ dargestellt werden, wobei $n := 3d$ die Anzahl der Variablen und $m := 2d$ die Anzahl der Gleichungen ist. Durch die $2d$ Schlupfvariablen enthält die Matrix A die $2d$ Einheitsvektoren des $\mathbb{R}^{2d}$ (bzw. deren Negatives) in den Spalten, so dass die Rangvoraussetzung erfüllt ist. Welche Indexmengen $B \subset \{1, \ldots, 3d\}$ kommen als Indexmengen für eine zulässige Basislösung (x, r, s) in Frage? Zunächst stellt man sofort fest, dass notwendigerweise $\{1, \ldots, d\} \subset B$ ist. Denn ist (x, r, s) eine Lösung von $(*)$, so ist $x_1 \geq \epsilon$ und $x_{j+1} \geq \epsilon x_j$, $j = 1, \ldots, d-1$, und folglich $x_j \geq \epsilon^j > 0$, $j = 1, \ldots, d$. Nun wollen wir uns überlegen, dass entweder $d + j \in B$ oder $2d + j \in B$, $j = 1, \ldots, d$. Hierzu nehmen wir an, (x, r, s) genüge $(*)$ und es sei $r_j = s_j = 0$ für ein $j \in \{1, \ldots, n\}$. Ist $j = 1$, so folgt $\epsilon = x_1 = 1$, was ein Widerspruch zu $\epsilon \in (0, \frac{1}{2})$ ist. Ist $j > 1$, so erhält man $2\epsilon x_{j-1} = 1$. Da weiter $x_{j-1} \leq 1$ gilt, hat man einen Widerspruch zu $\epsilon < \frac{1}{2}$. Daher muss $d + j$ oder $2d + j$ in B enthalten sein. Zusammen mit $\{1, \ldots, d\}$ hat man die möglichen Basisindizes gefunden. Die zugehörigen zulässigen Basislösungen sind sämtlich nichtentartet, da die zu Basisindizes gehörenden Komponenten von (x, r, s), wie wir gesehen haben, notwendigerweise positiv sind. □

Bemerkung: Ist $A \in \mathbb{R}^{m \times n}$ und Rang $(A) = m$, so kann es soviele Basislösungen geben wie man m-elementige Teilmengen aus $\{1, \ldots, n\}$ auswählen kann, und das sind bekanntlich $\binom{n}{m}$ Möglichkeiten. Natürlich brauchen nicht alle zulässig zu sein. Es ist wichtig, sich den Unterschied zwischen einer nichtentarteten und einer entarteten zulässigen Basislösung von $Ax = b$ bzw. Ecke von $M := \{x \in \mathbb{R}^n : x \geq 0,\ Ax = b\}$ klar zu machen. Ist x eine nichtentartete zulässige Basislösung, so ist die zugehörige Basis hierdurch eindeutig bestimmt, nämlich als Menge derjenigen Indizes aus $\{1, \ldots, n\}$,

für die die entsprechenden Komponenten von x positiv sind. Dagegen kann es zu einer entarteten Ecke mit $p < m$ positiven Komponenten bis zu $\binom{n-p}{n-m}$ verschiedene Basisdarstellungen ein und derselben entarteten Ecke geben. □

Beispiel: Die gegebene Aufgabe habe die Form

$$\text{(P)} \qquad \text{Minimiere} \quad c^T x \quad \text{auf} \quad M := \{x \in \mathbb{R}^n : x \geq 0,\ Ax \leq b\},$$

wobei $A \in \mathbb{R}^{m \times n}$, $b \in \mathbb{R}^m$ mit $b \geq 0$ und $c \in \mathbb{R}^n$. Nach Einführung von Schlupfvariablen $y \in \mathbb{R}^m$ hat (P) die äquivalente Form

$$(\tilde{\text{P}}) \qquad \text{Minimiere} \quad \tilde{c}^T z \quad \text{auf} \quad \tilde{M} := \{z \in \mathbb{R}^{n+m} : z \geq 0,\ \tilde{A}z = b\}$$

mit

$$\tilde{A} := \begin{pmatrix} A & I \end{pmatrix} \in \mathbb{R}^{m \times (n+m)}, \qquad \tilde{c} := \begin{pmatrix} c \\ 0 \end{pmatrix}.$$

Offensichtlich ist Rang $(\tilde{A}) = m$, mit $B := \{n+1, \ldots, n+m\}$ ist $z := (0, b)^T$ eine zulässige Basislösung von $\tilde{A}z = b$ mit den Basisindizes B. In diesem Falle ist es also völlig trivial eine zulässige Ausgangsbasislösung anzugeben und auf die Anwendung der sogenannten Phase I kann verzichtet werden. □

2.2.2 Die Phase II des primalen Simplexverfahrens

Gegeben sei das lineare Programm in Normalform[15]

$$\text{(P)} \qquad \text{Minimiere} \quad c^T z \quad \text{auf} \quad M := \{z \in \mathbb{R}^n : z \geq 0,\ Az = b\},$$

wobei $A = \begin{pmatrix} a_1 & \cdots & a_n \end{pmatrix} \in \mathbb{R}^{m \times n}$, $b \in \mathbb{R}^m$, $c \in \mathbb{R}^n$ und die Rang-Voraussetzung

$$\text{(V)} \qquad \text{Rang } (A) = m$$

erfüllt ist. Wir beschreiben in diesem Unterabschnitt die Phase II des Simplexverfahrens. In dieser wird vorausgesetzt, eine Ecke x des Polyeders M bzw. eine zulässige Basislösung zur Basis B sei bekannt. Dagegen dient die Phase I dazu, Widersprüche in den Nebenbedingungen bzw. $M = \emptyset$ zu entdecken, die Rang-Voraussetzung (V) zu überprüfen, gegebenenfalls redundante Gleichungen zu entfernen und eine zulässige Ausgangsbasislösung zu bestimmen. Da die Phase I darin besteht, die Phase II auf ein geeignetes Hilfsproblem anzuwenden, für welches trivialerweise die Rang-Voraussetzung (V) erfüllt und eine zulässige Basislösung bekannt ist, beginnen wir mit der Beschreibung der Phase II.

Sei also x eine zulässige Basislösung zur m-elementigen Basis B. Wir benutzen die oben eingeführten Bezeichnungen, also etwa A_B für die aus den Spalten von A mit einem Index aus B gebildete $m \times m$-Matrix. Entsprechende Bezeichnungen benutzen wir für Vektoren sowie andere Indexmengen als B, etwa die Menge der *Nichtbasisindizes* $N := \{1, \ldots, n\} \setminus B$. Wir werden uns auf das sogenannte *revidierte* Simplexverfahrens beschränken, von dem ein Schritt im folgenden Satz beschrieben wird.

[15] Die Variablen in dem linearen Programm bezeichnen wir mit z statt x, weil x gleich die Rolle einer aktuellen, zulässigen Basislösung spielen wird.

Satz 2.7 *Gegeben sei das lineare Programm in Normalform*

$$\text{(P)} \qquad \textit{Minimiere} \quad c^T z \quad \textit{auf} \quad M := \{z \in \mathbb{R}^n : z \geq 0,\ Az = b\}.$$

Hierbei ist $A = (\ a_1 \ \cdots \ a_n\ ) \in \mathbb{R}^{m\times n}$ *mit* $\operatorname{Rang}(A) = m$, $b \in \mathbb{R}^m$, *und* $c \in \mathbb{R}^n$. *Sei* $x \in M$ *eine zulässige Basislösung zur Basis* $B = \{j(1), \ldots, j(m)\}$, *der Basisanteil von* x *also* $x_B = A_B^{-1}b$, *mit zugehörigen Kosten* $c_0 := c_B^T A_B^{-1} b$. *Ferner sei* $y := A_B^{-T} c_B$ *und* $N := \{1, \ldots, n\} \setminus B$. *Dann gilt:*

1. *Ist* $c_N - A_N^T y \geq 0$, *so ist* x *eine Lösung von (P) und* y *eine Lösung des zu (P) dualen Programms (D). Ist sogar* $c_N - A_N^T y > 0$, *so ist* x *die eindeutige Lösung von (P).*

2. *Ist* $c_s - a_s^T y < 0$ *und* $w := A_B^{-1} a_s \leq 0$ *mit einem* $s \in N$, *so ist* $\inf(\text{P}) = -\infty$, *die Zielfunktion von (P) also auf der Menge der zulässigen Lösungen nicht nach unten beschränkt.*

3. *Sei* $c_s - a_s^T y < 0$ *und* $w := A_B^{-1} a_s \not\leq 0$ *mit einem* $s \in N$. *Man bestimme ein* $r \in \{1, \ldots, m\}$ *mit* $w_r > 0$ *und*

$$\frac{(A_B^{-1}b)_r}{w_r} = \min_{i=1,\ldots,m} \left\{ \frac{(A_B^{-1}b)_i}{w_i} : w_i > 0 \right\} =: \theta^*.$$

Definiert man $x^+ \in \mathbb{R}^n$ *durch*

$$x_j^+ := \begin{cases} (A_B^{-1}b)_i - \theta^* w_i & \text{für} \quad j = j(i) \in B, \\ \theta^* & \text{für} \quad j = s, \\ 0 & \text{für} \quad j \neq s,\ j \notin B, \end{cases}$$

so ist x^+ *eine zulässige Basislösung zur Basis*

$$B^+ := \{j(1), \ldots, j(r-1), s, j(r+1), \ldots, j(m)\}$$

mit den Kosten

$$c_0^+ := c^T x^+ = c^T x + \theta^*(c_s - a_s^T y) \leq c^T x = c_0.$$

Ferner ist

$$A_{B^+}^{-1} = \Big(I - \frac{(w - e_r)e_r^T}{w_r}\Big) A_B^{-1}.$$

Beweis: Sei $c_N - A_N^T y \geq 0$ und $z \in M$ beliebig. Wegen $Az = b$ ist $z_B = x_B - A_B^{-1} A_N z_N$ und daher

$$c^T z = c_B^T z_B + c_N^T z_N = c_B^T x_B + \underbrace{[c_N - A_N^T y]^T z_N}_{\geq 0} \geq c_B^T x_B = c^T x,$$

also x eine Lösung von (P). Auch die Eindeutigkeitsaussage liest man hieraus ab. Wegen

$$A^T y = \begin{pmatrix} A_B^T \\ A_N^T \end{pmatrix} y = \begin{pmatrix} c_B \\ A_N^T y \end{pmatrix} \leq \begin{pmatrix} c_B \\ c_N \end{pmatrix} = c$$

ist y dual zulässig. Weiter ist offensichtlich $b^T y = c^T x$, womit die Optimalität von y aus dem schwachen Dualitätssatz folgt.

Sei nun $c_s - a_s^T y < 0$ und $w := A_B^{-1} a_s \leq 0$ für ein $s \in N$. Definiert man $x(\theta) \in \mathbb{R}^n$ für $\theta \geq 0$ durch

$$x_j(\theta) := \begin{cases} (A_B^{-1}b)_i - \theta w_i & \text{für} \quad j = j(i) \in B, \\ \theta & \text{für} \quad j = s, \\ 0 & \text{für} \quad j \neq s,\ j \notin B, \end{cases}$$

so ist $x(\theta) \geq 0$ und

$$Ax(\theta) = A_B(x_B - \theta A_B^{-1} a_s) + \theta a_s = A_B x_B = b,$$

also $x(\theta) \in M$ für alle $\theta \geq 0$. Wegen

$$c^T x(\theta) = c_B^T(x_B - \theta A_B^{-1} a_s) + \theta c_s = c^T x + \theta(\underbrace{c_s - a_s^T y}_{<0})$$

folgt mit $\theta \to +\infty$, dass $\inf(\mathrm{P}) = -\infty$.

Sei nun $c_s - a_s^T y < 0$ und $w := A_B^{-1} a_s \not\leq 0$ mit einem $s \in N$. Ferner sei ein $r \in \{1, \ldots, m\}$ mit $w_r > 0$ und

$$\frac{(A_B^{-1}b)_r}{w_r} = \min_{i=1,\ldots,m} \left\{ \frac{(A_B^{-1}b)_i}{w_i} : w_i > 0 \right\} =: \theta^*$$

bestimmt. Dann ist $x^+ = x(\theta^*) \geq 0$ und $x^+_{j(r)} = 0$ nach Wahl von r. Ferner ist $Ax^+ = b$ und

$$c_0^+ := c^T x^+ = c^T x + \underbrace{\theta^*}_{\geq 0}(\underbrace{c_s - a_s^T y}_{<0}) \leq c^T x = c_0.$$

Erhält man B^+, wie angegeben, aus B dadurch, dass man $j(r)$ gegen s austauscht, so ist

$$A_{B^+} = (\ a_{j(1)} \ \cdots \ a_{j(r-1)} \ \ a_s \ \ a_{j(r+1)} \ \cdots \ a_{j(m)}\ ) = A_B + (a_s - a_{j(r)})e_r^T.$$

Wegen

$$1 + e_r^T A_B^{-1}(a_s - a_{j(r)}) = 1 + e_r^T(w - e_r) = w_r \neq 0$$

folgt aus dem Sherman-Morrison-Lemma[16], dass A_{B^+} nichtsingulär ist und

$$A_{B^+}^{-1} = A_B^{-1} - \frac{1}{w_r} A_B^{-1}(a_s - a_{j(r)}) e_r^T A_B^{-1} = \left(I - \frac{(w - e_r)e_r^T}{w_r}\right) A_B^{-1}$$

[16]Dieses sagt folgendes aus: Sei $A \in \mathbb{R}^{n\times n}$ nichtsingulär, seien $u, v \in \mathbb{R}^n$. Dann gilt:

1. Die Matrix $A + uv^T$ ist genau dann nichtsingulär, wenn $1 + v^T A^{-1} u \neq 0$.
2. Ist $1 + v^T A^{-1} u \neq 0$, so ist

$$(A + uv^T)^{-1} = A^{-1} - \frac{A^{-1}uv^T A^{-1}}{1 + v^T A^{-1} u}.$$

gilt. Wegen $x_j^+ = 0$ für alle $j \notin B^+$ und $x^+ \in M$ ist daher x^+ eine zulässige Basislösung zur Basis B^+. Insgesamt ist der Satz bewiesen. □

Bemerkung: Die Hauptarbeit beim revidierten Simplexverfahren besteht darin, die m-Vektoren $y := A_B^{-T} c_B$ und $w := A_B^{-1} a_s$ zu berechnen. Es wäre nicht gescheit, diese Berechnung jedesmal "ad hoc" zu machen und nicht zu berücksichtigen, dass sich die Koeffizientenmatrix von Schritt zu Schritt nur in einer Spalte verändert. Daher gewinnt[17] man $A_{B^+}^{-1}$ dadurch, dass man A_B^{-1} von links mit der *Gauß-Jordan-Matrix* (hierunter versteht man eine Matrix, die nur in einer Spalte von der Identität abweicht) $E := I - (w - e_r)e_r^T/w_r$ multipliziert. Dies erreicht man durch den folgenden einfachen Prozess:

- $e_r^T A_{B^+}^{-1} := (1/w_r) e_r^T A_B^{-1}$.

  Die r-te Zeile von $A_{B^+}^{-1}$ erhält man also dadurch, dass man die r-te Zeile von A_B^{-1} durch w_r dividiert.

- Für $i = 1, \ldots, m$, $i \neq r$:

  $$e_i^T A_{B^+}^{-1} := e_i^T A_B^{-1} - w_i \, e_r^T A_{B^+}^{-1}.$$

  Für $i \neq r$ gewinnt man daher die i-te Zeile von $A_{B^+}^{-1}$, indem man von der i-ten Zeile von A_B^{-1} das w_i-fache der r-ten Zeile von $A_{B^+}^{-1}$ subtrahiert.

Hat man daher für die Ausgangsbasis B_0 die Inverse $A_{B_0}^{-1}$ berechnet (häufig wird $A_{B_0} = I$ sein), so erhält man

$$A_{B_k}^{-1} = E_k \cdots E_1 A_{B_0}^{-1}$$

mit Gauß-Jordan-Matrizen $E_1, \ldots, E_k$. Um das Aufsummieren von Rundungsfehlern zu vermeiden, wird man nach einer gewissen Zahl von Schritten die Matrix A_B^{-1} neu berechnen. □

Wir geben ein MATLAB-Programm für die Durchführung eines Schrittes des revidierten primalen Simplexverfahrens an.

```
%Es wird ein Schritt des revidierten Simplexverfahrens programmiert.
%Gegeben ist die Aufgabe

%c'z=min, Az=b, z>=0, A eine m x n-Matrix mit Rang A=m<=n

%bei bekannter Basisindexmenge B. Das Programm hat die Form

%function [B_p,x_p,wert_p,in_p,y,info]=simplex(A,b,c,B,x,wert,in)
%Eingabe: Daten (A,b,c), Basisindexmenge B, Basisanteil x=x_B einer
%         Basisl"osung mit Zielfunktionswert wert und in=A_B^{-1}
%In info gibt es die folgenden Informationen:
%info= 1: B_p=B definiert eine optimale L"osung. wert_p=wert gibt den
%         zugeh"origen Optimalwert, x_p=x den Basisanteil einer
%         L"osung, y die L"osung  des dualen Problems  an, ferner ist
%         in_p=in die Inverse der Basismatrix.
```

[17] Eine weitere Möglichkeit besteht darin, *LR*-Zerlegungen "upzudaten".

```
%info=-1: Das Problem hat keine L"osung.
%info= 0: B definiert noch keine optimale L"osung. Es wird eine neue
%         Basisindexmenge B_p berechnet. Ausgegeben werden ferner
%         der neue Basisanteil x_p einer Basisl"osung, die neuen
%         Kosten wert_p, die neue Inverse in_p der Basismatrix und
%         und die duale N"aherung y.
%=====================================================================
function [B_p,x_p,wert_p,in_p,y,info]=simplex(A,b,c,B,x,wert,in);
[m,n]=size(A);N=1:n;N(B)=[];
A_N=A(:,N);  y=in'*c(B);
redu=c(N)-A_N'*y; [mi,j]=min(redu);  %j kann mehrere Indizes enthalten
%=====================================================================
                                                  %L"osung schon erreicht?
if mi>=0, info=1; B_p=B; x_p=x; wert_p=wert; in_p=in;
                                                  %L"osung erreicht
%=====================================================================
                                                   %L"osung nicht erreicht
else
  s=N(j(1)); a_s=A(:,s); w=in*a_s;
%=====================================================================
                                                 %Existiert keine L"osung?
  if max(w)<=0, info=-1; return, end;            %L"osung existiert nicht
%=====================================================================
                                                 %weitermachen
  P=find(w>0); [theta,i]=min(x(P)./w(P));
  r=P(i(1)); B_p=B; B_p(r)=s;
  info=0; x_p=x-theta*w; x_p(r)=theta; wert_p=wert+theta*mi;
  in_p(r,:)=(1/w(r))*in(r,:);
  for k=1:m
    if k~=r in_p(k,:)=in(k,:)-w(k)*in_p(r,:); end;
  end;
end; %if
```

Beispiel: Wir betrachten die lineare Optimierungsaufgabe

$$\text{(P)} \qquad \text{Minimiere} \quad c^T x \quad \text{auf} \quad M := \{x \in \mathbb{R}^n : x \geq 0,\ Ax = b\},$$

wobei

$$\begin{array}{|c|c|}\hline c^T & \\ \hline A & b \\ \hline \end{array} := \begin{array}{|rrrrr|r|}\hline -1 & -1 & 0 & 0 & 0 & \\ \hline 1 & 3 & 1 & 0 & 0 & 13 \\ 3 & 1 & 0 & 1 & 0 & 15 \\ -1 & 1 & 0 & 0 & 1 & 3 \\ \hline \end{array}$$

Nach

```
A=[1 3 1 0 0;3 1 0 1 0;-1 1 0 0 1];
b=[13;15;3];c=[-1;-1;0;0;0];
B=[3 4 5];A_B=A(:,B);in=inv(A_B);x=in*b;
wert=c(B)'*x;
```

```
info=0;
while info==0
   [B,x,wert,in,y,info]=simplex(A,b,c,B,x,wert,in);
end;
[m,n]=size(A);xx=zeros(n,1);xx(B)=x;x=xx;
```

erhalten wir die Lösung $x^* = (4, 3, 0, 0, 4)^T$, die duale Lösung $y^* = (-0.25, -0.25), 0)^T$ und den Optimalwert $c^T x^* = b^T y^* = -7$. □

Beispiel: Das folgende Beispiel wird auch V. Klee, G. J. Minty (1972) zugeschrieben (auch wenn ich es in dem Aufsatz nicht gefunden habe). Auch dieses soll demonstrieren, dass das Simplexverfahren exponentiell viele Schritte benötigen kann. Es lautet:

$$\text{Maximiere} \quad c^T x \quad \text{unter den Nebenbedingungen} \quad x \geq 0,\ Ax \leq b,$$

wobei

$$A := \begin{pmatrix} 1 & 0 & 0 & \cdots & 0 \\ 4 & 1 & 0 & \cdots & 0 \\ 8 & 4 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 2^n & 2^{n-1} & 2^{n-2} & \cdots & 1 \end{pmatrix}, \qquad b := \begin{pmatrix} 5 \\ 25 \\ 125 \\ \vdots \\ 5^n \end{pmatrix}, \qquad c := \begin{pmatrix} 2^{n-1} \\ 2^{n-2} \\ 2^{n-3} \\ \vdots \\ 1 \end{pmatrix}.$$

Wir wenden die Funktion `simplex` für $n = 4$ an und benötigen in der Tat $2^4 = 16$ Schritte, um zu der Lösung zu gelangen. In

B				x				$c^T x$
5	6	7	8	0	0	0	0	0
1	6	7	8	5	0	0	0	40
1	2	7	8	5	5	0	0	60
5	2	7	8	0	25	0	0	100
5	2	3	8	0	25	25	0	150
1	2	3	8	5	5	65	0	190
1	6	3	8	5	0	85	0	210
5	6	3	8	0	0	125	0	250
5	6	3	4	0	0	125	125	375
1	6	3	4	5	5	85	205	415
1	2	3	4	5	5	65	245	435
5	2	3	4	0	25	25	325	475
5	2	7	4	0	25	0	425	525
1	2	7	4	5	5	0	505	565
1	6	7	4	5	0	0	545	585
5	6	7	4	0	0	0	625	625

sind die Basisindizes und die zulässigen Näherungslösungen angegeben. □

Bemerkung: Ist x eine *nichtentartete* zulässige Basislösung zur Basis B, so werden die Kosten der neuen Basislösung x^+ echt vermindert. Da in jedem Schritt die Kosten zumindestens nicht vergrößert werden, kann man im Verlauf des Simplexverfahrens nicht zu x zurückkehren. Insbesondere ergibt sich hieraus, dass das Simplexverfashren nach endlich vielen Schritten abbrechen muss (mit einer Lösung oder der Information, dass die Zielfunktion auf der Menge der zulässigen Lösungen nicht nach unten beschränkt ist), wenn alle berechneten zulässigen Basislösungen nichtentartet sind, da es ja nur endlich viele zulässige Basislösungen gibt.

Ist dagegen x eine *entartete* zulässige Basislösung zur Basis $B = \{j(1), \ldots, j(m)\}$ und ist

$$\{i \in \{1, \ldots, m\} : w_i > 0,\ (A_b^{-1}b)_i = 0\} \neq \emptyset,$$

so erhält man im (revidierten) Simplexverfahren

$$0 = \frac{(A_B^{-1}b)_r}{w_r} = \min_{i=1,\ldots,m} \Big\{ \frac{(A_B^{-1}b)_i}{w_i} : w_i > 0 \Big\}.$$

In diesem Falle ist $x^+ = x$, man bleibt also in der Ecke x stehen, lediglich die Basisdarstellung von x^+ ist eine andere. Denn B^+ entsteht aus B dadurch, dass $s \notin B$ aufgenommen und $j(r) \in B$ entfernt wird. Kritisch wird es dann, wenn man in ein und derselben Ecke stehen bleibt, lediglich von Schritt zu Schritt die Basis austauscht, und nach endlich vielen Schritten zur Ausgangsbasis zurückkehrt. Man spricht dann von einem *Zyklus* im Simplexverfahren. Ein solcher Zyklus kann *theoretisch* auftreten. Ein bekanntes Beispiel stammt von Beale. Es handelt sich um ein lineares Programm in Normalform, dessen Daten in der üblichen Weise in

$-\frac{3}{4}$	20	$-\frac{1}{2}$	6	0	0	0	
$\frac{1}{4}$	-8	-1	9	1	0	0	0
$\frac{1}{2}$	-12	$-\frac{1}{2}$	3	0	1	0	0
0	0	1	0	0	0	1	1

eingetragen sind. Wählt man den in die Basis aufzunehmenden Index $s \notin B$ nach der sogenannten Kleinste-Kosten-Regel, also so, dass $\overline{c}_s = \min_{j \in N} \overline{c}_j$, und r kleinstmöglich, so wird man (genau das trifft z. B. für die obige MATLAB-Implementation des Simplexverfahrens zu) den folgenden Zyklus von Basisindizes erhalten:

$$\{5,6,7\},\ \{1,6,7\},\ \{1,2,7\},\ \{3,2,7\},\ \{3,4,7\},\ \{5,4,7\},\ \{5,6,7\}, \cdots.$$

Daher ist das Simplexverfahren ohne Zusatzregel kein endliches Verfahren. Eine "Anti-Zyklen-Regel" präzisiert, wie der aufzunehmende Index $s \notin B$ und der zu enfernende Index $j(r) \in B$ bzw. $r \in \{1, \ldots, m\}$ zu wählen sind, um die Endlichkeit des Simplexverfahrens zu sichern. Die einfachste Zusatzregel zur Vermeidung von Zyklen stammt von R. G. Bland (1977). Diese sogenannte Bland-Regel (oder auch Kleinste-Index-Regel) besagt, dass man s und r stets als *kleinstmöglichen* Index wählen sollte. Zum Beweis, dass das Simplexverfahren mit der Bland-Regel ein endliches Verfahren ist, verweisen

wir z. B. auf C. H. Papadimitriou, K. Steiglitz (1982, S. 54)[18]. Eine weitere bekannte Zusatzregel benutzt die lexikographische Ordnung zwischen Vektoren gleicher Länge, siehe z. B. J. Werner (1992, S. 98) oder C. H. Papadimitriou, K. Steiglitz (1982, S. 334). □

2.2.3 Die Phase I des Simplexverfahrens

Gegeben sei wiederum das lineare Programm in Normalform

$$\text{(P)} \qquad \text{Minimiere} \quad c^T x \quad \text{auf} \quad M := \{x \in \mathbb{R}^n : x \geq 0,\ Ax = b\},$$

wobei $A = (a_{ij}) = (\ a_1 \ \cdots \ a_n\ ) \in \mathbb{R}^{m \times n}$, $b = (b_i) \in \mathbb{R}^m$ und $c = (c_j) \in \mathbb{R}^n$. Da man notfalls eine Gleichungsrestriktion mit -1 multiplizieren kann, ist o. B. d. A. $b \geq 0$. Ziel der Phase I des Simplexverfahrens ist es, eine zulässige Basislösung für (P) zu berechnen (mit der dann die Phase II gestartet werden kann) bzw. zu entdecken, dass (P) nicht zulässig, also $M = \emptyset$, ist, oder A nicht vollen Rang hat und in diesem Falle redundante Gleichungen zu entfernen. Dieses Ziel wird durch die Anwendung der Phase II des Simplexverfahrens auf ein geeignetes Hilfsproblem erreicht.

Enthält A in den Spalten schon die m Einheitsvektoren des $\mathbb{R}^m$, so kann die Phase II sofort gestartet werden. Wir nehmen an, das sei nicht der Fall, führen einen Vektor y von sogenannten *künstlichen Variablen* ein und betrachten das lineare Programm in Normalform

$$\text{(}\hat{\text{P}}\text{)} \qquad \begin{cases} \text{Minimiere} \quad \begin{pmatrix} 0 \\ e \end{pmatrix}^T \begin{pmatrix} x \\ y \end{pmatrix} \quad \text{auf} \\ \hat{M} := \left\{ \begin{pmatrix} x \\ y \end{pmatrix} \in \mathbb{R}^{n+m} : \begin{pmatrix} x \\ y \end{pmatrix} \geq 0,\ (\ A \quad I\ ) \begin{pmatrix} x \\ y \end{pmatrix} = b \right\}. \end{cases}$$

Hierbei ist $e := (1, \ldots, 1)^T \in \mathbb{R}^m$. Zur Abkürzung setzen wir

$$\hat{A} := (\ A \quad I\ ) = (\ a_1 \ \cdots \ a_n \quad e_1 \ \cdots \ e_m\ ) \in \mathbb{R}^{m \times (n+m)}.$$

Mit der Basis $B := \{n+1, \ldots, n+m\}$ kann die Phase II des Simplexverfahrens gestartet werden. Da die Zielfunktion von $(\hat{\text{P}})$ auf der (nichtleeren) Menge der zulässigen Lösungen (durch 0) nach unten beschränkt ist, besitzt $(\hat{\text{P}})$ eine optimale Basislösung zu einer Basis $B = \{j(1), \ldots, j(m)\} \subset \{1, \ldots, n+m\}$, die wir mit der Phase II des Simplexverfahrens berechnen können. Wir unterscheiden zwei Fälle.

- Es ist $\min(\hat{\text{P}}) > 0$.

Dann besitzt (P) keine zulässige Lösung, man breche mit einer entsprechenden Meldung ab.

[18] C. H. Papadimitriou, K. Steiglitz (1982) *Combinatorial Optimization: Algorithms and Complexity*. Prentice-Hall, Englewood Cliffs.

Beispiel: Gegeben sei eine lineare Optimierungsaufgabe in Standardform, wobei die Daten

$$\left[\begin{array}{c|c} c^T & \\ \hline A & b \end{array}\right] := \left[\begin{array}{cccc|c} 0 & -1 & 0 & 0 & \\ \hline -1 & -1 & -1 & 0 & 3 \\ -1 & 2 & 0 & -1 & 1 \end{array}\right]$$

sind. Es werden zwei künstliche Variable eingeführt und die Phase II des Simplexverfahrens auf die Daten

$$\left[\begin{array}{c|c} \hat{c}^T & \\ \hline \hat{A} & b \end{array}\right] := \left[\begin{array}{cccccc|c} 0 & 0 & 0 & 0 & 1 & 1 & \\ \hline -1 & -1 & -1 & 0 & 1 & 0 & 3 \\ -1 & 2 & 0 & -1 & 0 & 1 & 1 \end{array}\right]$$

angewandt. Als zugehörigen Wert erhält man $\min(\hat{\mathrm{P}}) = 3.5$, das Ausgangsproblem hat also keine zulässige Lösung. □

- Es ist $\min(\hat{\mathrm{P}}) = 0$.

Dann ist gesichert, dass (P) eine zulässige Lösung besitzt. Nun berechne man $r \in \{1, \ldots, m\}$ mit $j(r) = \max_{i=1,\ldots,m} j(i)$. Ein angenehmer Fall liegt vor, wenn $j(r) \leq n$ bzw. die optimale Basis keinen zu einer künstlichen Variablen gehörenden Index enthält. Dann ist durch die Anwendung der Phase II des Simplexverfahrens auf $(\hat{\mathrm{P}})$ eine zulässige Basislösung von $Ax = b$ mit dem Basisanteil $x_B = \hat{A}_B^{-1} b = A_B^{-1} b$ sowie die Matrix $\hat{A}_B^{-1} = A_B^{-1}$ berechnet worden. Hiermit kann die Phase II des (revidierten) Simplexverfahrens zur Lösung des eigentlich interessierenden Problems (P) gestartet werden.

Beispiel: Gegeben sei eine lineare Optimierungsaufgabe in Standardform, wobei die Daten

$$\left[\begin{array}{c|c} c^T & \\ \hline A & b \end{array}\right] := \left[\begin{array}{cccccccc|c} \frac{6}{5} & \frac{9}{5} & 1 & 0 & 0 & 0 & 0 & 0 & \\ \hline 1 & 0 & -2 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & -2 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ -1 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & -2 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 3 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 1 \\ 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 1 \end{array}\right]$$

sind (siehe J. Werner (1992, S. 104)). Die ersten vier Einheitsvektoren sind als Spalten in der Koeffizientenmatrix A schon enthalten, so dass es genügt, zwei künstliche Variable einzuführen. Die Daten des in der Phase I zu lösenden Hilfsproblems sind

daher

0	0	0	0	0	0	0	0	1	1	
1	0	−2	1	0	0	0	0	0	0	0
1	−2	0	0	1	0	0	0	0	0	0
−1	1	0	0	0	1	0	0	0	0	0
0	−2	1	0	0	0	1	0	0	0	0
3	0	0	0	0	0	0	−1	1	0	1
1	1	1	0	0	0	0	0	0	1	1

Wir wenden die in MATLAB geschriebene Funktion `simplex` an, wobei wir mit der Indexmenge $B = \{4, 5, 6, 7, 9, 10\}$ starten. Nach

```
hatA=[1 0 -2 1 0 0 0 0 0 0;1 -2 0 0 1 0 0 0 0 0;-1 1 0 0 0 1 0 0 0 0;
0 -2 1 0 0 0 1 0 0 0;3 0 0 0 0 0 0 -1 1 0;1 1 1 0 0 0 0 0 0 1];
b=[0;0;0;0;1;1];hatc=[0;0;0;0;0;0;0;0;1;1];
B=[4 5 6 7 9 10];hatA_B=hatA(:,B);in=inv(hatA_B);x=in*b;
wert=hatc(B)'*x;info=0;
while info==0
   [B,x,wert,in,y,info]=simplex(hatA,b,hatc,B,x,wert,in);
end;
```

erhalten wir eine zulässige Basislösung x (bzw. genauer den Basisanteil x_B) zur keine künstlichen Indizes enthaltenden Basis $B = \{1, 3, 6, 7, 2, 8\}$. Es liegt also der eben angesprochene angenehme Fall vor. Hiermit kann die Phase II des Simplexverfahrens gestartet werden. Nach

```
A=hatA(:,1:8);c=[6/5;9/5;1;0;0;0;0;0];B=[1 3 6 7 2 8];
wert=c(B)'*x;info=0;
while info==0
   [B,x,wert,in,y,info]=simplex(A,b,c,B,x,wert,in);
end;
```

hat man die Lösung berechnet. Man erhält

$$B = \{1, 3, 6, 4, 2, 8\}, \qquad x_B = \begin{pmatrix} 0.4 \\ 0.4 \\ 0.2 \\ 0.4 \\ 0.2 \\ 0.2 \end{pmatrix}, \qquad y = \begin{pmatrix} 0 \\ -0.04 \\ 0 \\ -0.24 \\ 0 \\ 1.24 \end{pmatrix}$$

als optimale Basisindexmenge, optimalen Basisanteil der primalen Lösung und optimale Lösung des dualen Problems. □

Ist dagegen $j(r) = \min_{i=1,\dots,m} j(i) \in B$ ein künstlicher Basisindex, also $j(r) > n$, so ist die gewonnene optimale Lösung von $(\hat{\mathrm{P}})$ notwendig entartet, da alle Komponenten zu künstlichen Indizes wegen $\min(\hat{\mathrm{P}}) = 0$ verschwinden müssen. Insbesondere ist

$(\hat{A}_B^{-1}b)_r = 0$. Die Idee besteht nun darin, den künstlichen Basisindex $j(r)$ gegen ein $s \in \{1,\dots,n\} \setminus B$ auszutauschen oder festzustellen, dass eine Gleichung in $Ax = b$ redundant ist und folglich gestrichen werden kann. Genauer sind die folgenden beiden Fälle möglich.

(a) Es existiert ein $s \in \{1,\dots,n\} \setminus B$ mit $e_r^T \hat{A}_B^{-1} a_s \neq 0$.

Man setze $w := \hat{A}_B^{-1} a_s$ und $B^+ := \{j(1),\dots,j(r-1), s, j(r+1),\dots,j(m)\}$. Anschließend berechne man

$$\hat{A}_{B^+}^{-1} := \Big(I - \frac{(w - e_r)e_r^T}{w_r}\Big)\hat{A}_B^{-1}.$$

An der gewonnenen Basislösung ändert sich hierbei natürlich nichts, d. h. es ist $\hat{A}_{B^+}^{-1} b = \hat{A}_B^{-1} b$, was auch wegen $e_r^T A_B^{-1} b = (A_B^{-1}b)_r = 0$ offensichtlich ist. Zum Schluss setze man $B := B^+$ und prüfe erneut, ob $B \subset \{1,\dots,n\}$, also die künstlichen Indizes aus der Basis vertrieben sind.

(b) Für alle $s \in \{1,\dots,n\} \setminus B$ ist $e_r^T \hat{A}_B^{-1} a_s = 0$.

Wir werden erkennen, dass in diesem Falle die Gleichungen redundant sind und daher eine Gleichung gestrichen werden kann.

Für $j = j(i) \in \{1,\dots,n\} \cap B$ ist $\hat{A}_B^{-1} a_j = e_i$ und daher $e_r^T \hat{A}_B^{-1} a_j = 0$. Insgesamt ist daher die r-te Zeile von $\hat{A}_B^{-1} A$ eine Nullzeile. Folglich ist

$$A^T(\hat{A}_B^{-T} e_r) = 0 \qquad \text{bzw.} \qquad \sum_{i=1}^{m} (e_i^T \hat{A}_B^{-T} e_r) A^T e_i = 0,$$

die Zeilen von A sind also linear abhängig. Der künstliche Basisindex $j(r)$ sei durch $j(r) = n + q$ mit $q \in \{1,\dots,m\}$ gegeben. Der Koeffizient von $A^T e_q$, also der q-ten Spalte von A^T bzw. der q-ten Zeile von A, ist

$$e_q^T \hat{A}_B^{-T} e_r = e_r^T \hat{A}_B^{-1} e_q = e_r^T e_r = 1,$$

so dass

$$A^T e_q = -\sum_{\substack{i=1 \\ i \neq q}}^{m} (e_i^T \hat{A}_B^{-T} e_r) A^T e_i.$$

Daher ist die q-te Zeile von A eine Linearkombination der übrigen Zeilen. Die q-te Gleichung in $Ax = b$ ist also redundant und wird daher gestrichen. Außerdem streicht man in dem Basisanteil $\hat{A}_B^{-1} b$ der aktuellen Basislösung die r-te Komponente (hier stand eine Null) und in $\hat{A}_B^{-1}$ die r-te Zeile und die q-te Spalte (dies wird in dem folgenden Lemma gerechtfertigt). Anschließend setze man

$$B := \{j(1),\dots,j(r), j(r+1),\dots,j(m)\}, \qquad m := m - 1.$$

Da $j(r)$ der *größte* künstliche Basisindex war, ist auch nach dieser Reduktion $B \subset \{1,\dots,n+m\}$. Dann wird erneut geprüft, ob $B \subset \{1,\dots,n\}$, ob also die künstlichen Indizes aus der Basis vertrieben sind.

Beispiel: Wir betrachten ein lineares Programm in Standardform mit den Daten

$$\begin{array}{|c|c|}\hline c^T & \\ \hline A & b \\ \hline \end{array} := \begin{array}{|rrrrr|c|}\hline 2 & 4 & 3 & 0 & 0 & \\ \hline -1 & 1 & 1 & -1 & 0 & 2 \\ 2 & 1 & 0 & 0 & -1 & 1 \\ 1 & 2 & 1 & -1 & -1 & 3 \\ \hline \end{array}$$

Die dritte Gleichung entsteht offenbar durch Addition der ersten beiden Gleichungen, die Rangvoraussetzung ist also nicht erfüllt und das sollte erkannt werden. Nach Einführung künstlicher Variabler hat man ein lineares Programm in Standardform mit den Daten

$$\begin{array}{|c|c|}\hline \hat{c}^T & \\ \hline \hat{A} & b \\ \hline \end{array} := \begin{array}{|rrrrrrrr|c|}\hline 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & \\ \hline -1 & 1 & 1 & -1 & 0 & 1 & 0 & 0 & 2 \\ 2 & 1 & 0 & 0 & -1 & 0 & 1 & 0 & 1 \\ 1 & 2 & 1 & -1 & -1 & 0 & 0 & 1 & 3 \\ \hline \end{array}$$

Wendet man die Phase II des Simplexverfahrens auf dieses Problem an, so erhält man 0 als Wert dieses linearen Programms, das Ausgangsproblem ist also zulässig, weiter ist $B = \{3, 2, 8\}$ die optimale Basisindexmenge. Die künstliche Variable

$$j(\underbrace{3}_{=r}) = 8 = 5 + \underbrace{3}_{=q}$$

ist noch in der Basis. Es ist

$$\hat{A}_B^{-1}(\ a_1 \quad a_4 \quad a_5\ ) = \begin{pmatrix} -3 & -1 & 1 \\ 2 & 0 & -1 \\ 0 & 0 & 0 \end{pmatrix},$$

die dritte Zeile ist eine Nullzeile. Nach Streichen der dritten Zeile im Ausgangsproblem, der dritten Zeile und dritten Spalte in $\hat{A}_B^{-1}$, der dritten Komponente im Basisanteil und $B = \{3, 2\}$ kann die Phase II des Simplexverfahrens gestartet werden. Der erste Schritt stellt schon fest, dass man eine Lösung erreicht hat. Es ist $x_B = (1, 1)^T$, der optimale Wert ist 7. □

Den Beweis des nächsten Lemmas stellen wir als Aufgabe.

Lemma 2.8 *Sei $A \in \mathbb{R}^{m\times m}$ eine nichtsinguläre Matrix, deren r-te Spalte der q-te Einheitsvektor im $\mathbb{R}^m$ ist. Die Matrix $A^{qr} \in \mathbb{R}^{(m-1)\times(m-1)}$ entstehe aus A durch Streichen der q-ten Zeile und der r-ten Spalte. Entsprechend entstehe $b^q \in \mathbb{R}^{m-1}$ aus $b \in \mathbb{R}^m$ durch Streichen der b-ten Komponente. Dann gilt:*

(a) A^{qr} ist nichtsingulär und $(A^{qr})^{-1} = (A^{-1})^{rq}$.

Die Inverse von A^{qr} erhält man also dadurch, dass man in der Inversen A^{-1} von A die r-te Zeile und die q-te Spalte streicht.

(b) $(A^{qr})^{-1}b^q = (A^{-1}b)^r$.

Die Lösung y des linearen Gleichungssystems $A^{qr}y = b^q$ erhält man also dadurch, dass man in der Lösung x von $Ax = b$ die r-te Komponente streicht.

2.2.4 Das duale Simplexverfahren

In diesem Unterabschnitt betrachten wir wieder das lineare Programm in Normalform

$$\text{(P)} \qquad \text{Minimiere} \quad c^T x \quad \text{auf} \quad M := \{x \in \mathbb{R}^n : x \geq 0,\ Ax = b\},$$

wobei $A = (\ a_1\ \cdots\ a_n\ ) \in \mathbb{R}^{m\times n}$, $b \in \mathbb{R}^m$ und $c \in \mathbb{R}^n$. Es wird $\text{Rang}\,(A) = m$ vorausgesetzt.

Beim (primalen) revidierten Simplexverfahren ging man von einer zulässigen Basislösung x zu der Basis B aus und berechnete $y := A_B^{-T} c_B$. Dann ist $b^T y = c^T x$. Ist also y zulässig für das duale Problem

$$\text{(D)} \qquad \text{Maximiere} \quad b^T y \quad \text{auf} \quad N := \{y \in \mathbb{R}^m : A^T y \leq c\},$$

bzw. der Optimalitätstest

$$\bar{c}_N := c_N - (A_B^{-1} A_N)^T c_B = c_N - A_N^T y \geq 0$$

erfüllt, so ist x eine Lösung von (P) und y eine Lösung von (D).

Dagegen geht man beim *dualen Simplexverfahren* von einer (i. allg. nicht zulässigen) Basislösung x zu einer Basis B aus (der Basisanteil $x_B := A_B^{-1} b$ ist also i. allg. kein nichtnegativer Vektor), für die $y := A_B^{-T} c_B$ dual zulässig ist, also $c_N - A_N^T y \geq 0$ erfüllt ist. Nach wie vor ist $c^T x = b^T y$. Genau diese Situation bleibt im Algorithmus erhalten und der duale Zielfunktionswert wird von Schritt zu Schritt zumindestens nicht verkleinert. Sobald man zu einer primal zulässigen Basislösung kommt, endet das Verfahren.

Beispiel: Das duale Simplexverfahren ist sofort anwendbar auf ein lineares Programm der Form

$$\text{Minimiere} \quad c^T x \quad \text{unter den Nebenbedingungen} \quad x \geq 0,\ Ax \leq b$$

mit einem *nichtnegativen* Kostenvektor c. Denn nach Überführung auf Normalform können die Schlupfvariablen als Basisvariablen genommen werden. I. allg. wird hier b kein nichtnegativer Vektor sein (sonst wäre nämlich $x^* = 0$ schon eine Lösung), aber $y = 0$ ist dual zulässig. Durch die Anwendung des dualen Simplexverfahrens auf eine Aufgabe vom obigen Typ erspart man sich daher die Anwendung der Phase I des primalen Simplexverfahrens. □

Satz 2.7 ist Grundlage für das (primale) revidierte Simplexverfahren. Entsprechend werden im folgenden Satz der Optimalitäts- und der Unlösbarkeitstest sowie ein Schritt beim revidierten dualen Simplexverfahren beschrieben. Den Beweis überlassen wir als Aufgabe.

Satz 2.9 *Gegeben sei das lineare Programm in Normalform*

$$\text{(P)} \qquad \textit{Minimiere} \quad c^T z \quad \textit{auf} \quad M := \{z \in \mathbb{R}^n : z \geq 0,\ Az = b\}.$$

Hierbei sei $A = (\ a_1\ \cdots\ a_n\ )$ *mit* $\text{Rang}\,(A) = m$, $b \in \mathbb{R}^m$ *und* $c \in \mathbb{R}^n$. *Sei* x *eine (i. allg. nicht zulässige) Basislösung zur Basis* $B = \{j(1), \ldots, j(m)\}$, *der Basisanteil von* x *also* $x_B := A_B^{-1} b$. *Mit* $N := \{1, \ldots, n\} \setminus B$ *werde die Menge der Nichtbasisindizes bezeichnet. Sei* $y := A_B^{-T} c_B$ *dual zulässig, also* $\bar{c}_N := c_N - A_N^T y \geq 0$. *Dann gilt:*

1. *Ist $A_B^{-1}b \geq 0$, also x eine zulässige Basislösung, so ist x eine Lösung von (P) und y eine Lösung des zu (P) dualen Programms (D). Ist sogar $A_B^{-1}b > 0$, so ist y eindeutige Lösung von (D).*

2. *Ist $(A_B^{-1}b)_r < 0$, $u := A_B^{-T}e_r$ und $A_N^T u \geq 0$ mit einem $r \in \{1, \ldots, m\}$, so ist* $\sup(\mathrm{D}) = +\infty$ *und daher (P) nicht zulässig.*

3. *Sei $(A_B^{-1}b)_r < 0$, $u := A_B^{-T}e_r$ und $v_N := A_N^T u \not\geq 0$. Man bestimme ein $s \in N$ mit $v_s < 0$ und*
$$\frac{\overline{c}_s}{v_s} = \max_{j \in N}\left\{\frac{\overline{c}_j}{v_j} : v_j < 0\right\} =: \gamma^*,$$
anschließend berechne man $w := A_B^{-1}a_s$. Definiert man $x^+ \in \mathbb{R}^n$ durch
$$x_j^+ := \begin{cases} (A_B^{-1}b)_i - \dfrac{(A_B^{-1}b)_r}{w_r}w_i & \text{für} \quad j = j(i) \in B, \\ \dfrac{(A_B^{-1}b)_r}{w_r} & \text{für} \quad j = s, \\ 0 & \text{für} \quad j \neq s,\ j \notin B, \end{cases}$$
so ist x^+ eine Basislösung zur Basis
$$B^+ := \{j(1), \ldots, j(r-1), s, j(r+1), \ldots, j(m)\}.$$
Ferner ist
$$y^+ := A_{B^+}^{-T}c_{B^+} = y + \gamma^* u$$
dual zulässig und
$$c_0^+ := c^T x^+ = b^T y^+ = b^T y + \gamma^*(A_B^{-1}b)_r \geq b^T y = c^T x =: c_0.$$

Wir geben ein MATLAB-Programm zur Durchführung eines Schrittes des revidierten dualen Simplexverfahrens an.

```
%Es wird ein Schritt des revidierten dualen Simplexverfahrens
%programmiert. Gegeben ist die Aufgabe

%c'z=min, Az=b, z>=0, A eine m x n-Matrix mit Rang A=m<=n

%bei bekannter Basisindexmenge B. Das Programm hat die Form
%function [B_p,x_p,wert_p,in_p,y_p,info]=dualsimplex(A,b,c,B,x,wert,in,y)
%Eingabe: Daten (A,b,c), Basisindexmenge B, Basisanteil x=x_B einer
%         Basisl"osung mit Zielfunktionswert wert, in=A_B^{-1} und
%         dual zul"assiges y=in'*c(B)
%In info gibt es die folgenden Informationen:
%info= 1: B_p=B definiert eine optimale L"osung. wert_p=wert gibt den
%         zugeh"origen Optimalwert, x_p=x den Basisanteil einer
%         L"osung, y_p die L"osung  des dualen Problems  an, ferner ist
%         in_p=in die Inverse der Basismatrix.
```

```
%info=-1: Das Problem hat keine L"osung, da Ausgangsproblem
%         unzul"assig.
%info= 0: B definiert noch keine optimale L"osung. Es wird eine neue
%         Basisindexmenge B_p berechnet. Ausgegeben werden ferner
%         der neue Basisanteil x_p einer Basisl"osung, die neuen
%         Kosten wert_p, die neue Inverse in_p der Basismatrix und
%         und die duale N"aherung y_p.
%=====================================================================
function [B_p,x_p,wert_p,in_p,y_p,info]=dualsimplex(A,b,c,B,x,wert,in,y);
[m,n]=size(A);N=1:n;N(B)=[];
A_N=A(:,N);redu=c(N)-A_N'*y;
[mi,i]=min(x);                        %i kann mehrere Indizes enthalten
%=====================================================================
                                                  %L"osung schon erreicht?
if mi>=0, info=1; B_p=B; x_p=x; wert_p=wert; in_p=in; y_p=y;
                                                  %L"osung erreicht
%=====================================================================
                                                  %L"osung nicht erreicht
else
  r=i(1); u=in(r,:); u=u'; v_N=A_N'*u;
%=====================================================================
                                                 %Existiert keine L"osung?
  if min(v_N)>=0, info=-1; return, end;           %L"osung existiert nicht
%=====================================================================
                                                 %weitermachen
  P=find(v_N<0); [gamma,j]=max(redu(P)./v_N(P));
  s=N(P(j(1)));
  B_p=B; B_p(r)=s;
  a_s=A(:,s); w=in*a_s; theta=x(r)/w(r);
  info=0; x_p=x-theta*w; x_p(r)=theta; wert_p=wert+gamma*x(r);
  in_p(r,:)=(1/w(r))*in(r,:);
  for k=1:m
    if k~=r in_p(k,:)=in(k,:)-w(k)*in_p(r,:); end;
  end;
  y_p=y+gamma*u;
end; %if
```

Bemerkung: Ein Nachteil des dualen Simplexverfahrens ist natürlich, dass primal unzulässige "Näherungen" für eine Lösung bestimmt werden und erst am Schluss die primale Zulässigkeit gesichert ist. Dafür hat das duale Simplexverfahren aber auch einige Vorteile. Wie wir oben schon bemerkt haben, erspart man sich bei einigen Problemen die Phase I. Wichtiger noch ist, dass das Simplexverfahren mit "nachträglichen", zusätzlichen Restriktionen "fertig wird". Genauer gehen wir aus von einem linearen Programm in Normalform

$$\text{(P)} \qquad \text{Minimiere} \quad c^T x \quad \text{auf} \quad M := \{x \in \mathbb{R}^n : x \geq 0,\ Ax = b\}.$$

Hierbei sei Rang $(A) = m$. Mit dem (primalen oder dualen) Simplexverfahren sei eine optimale, zulässige Basislösung x zur Basis B bestimmt. Nun betrachte man das gleiche

Problem mit einer zusätzlichen Ungleichungsrestriktion, also

$$(\hat{\mathrm{P}}) \qquad \text{Minimiere} \quad c^T x \quad \text{auf} \quad \hat{M} := \{x \in \mathbb{R}^n : x \geq 0,\ Ax = b,\ a^T x \leq \beta\}.$$

Da das zu $(\hat{\mathrm{P}})$ duale Programm wegen der Existenz einer Lösung von (P) auf alle Fälle zulässig ist, ist $(\hat{\mathrm{P}})$ entweder unzulässig oder lösbar. Nach Einführung einer Schlupfvariablen hat man das zu $(\hat{\mathrm{P}})$ äquivalente Problem in Normalform mit den Daten

$$\begin{array}{|c|c|}\hline \hat{c}^T & \\ \hline \hat{A} & \hat{b} \\ \hline \end{array} := \begin{array}{|ccc|}\hline c^T & 0 & \\ \hline A & 0 & b \\ a^T & 1 & \beta \\ \hline \end{array}$$

Dann ist

$$\hat{x} := \begin{pmatrix} x \\ \beta - a^T x \end{pmatrix}$$

eine zugehörige Basislösung zur Basis $\hat{B} := B \cup \{n+1\}$, da einerseits die Gleichungen erfüllt sind und andererseits die Matrix

$$\hat{A}_{\hat{B}} := \begin{pmatrix} A_B & 0 \\ a_B^T & 1 \end{pmatrix}$$

nichtsingulär ist. Mit Hilfe der Matrix A_B^{-1}, die bei der Lösung von (P) abfiel, kann man die Inverse von $\hat{A}_{\hat{B}}$ sogar einfach angeben, es ist nämlich

$$\hat{A}_{\hat{B}}^{-1} = \begin{pmatrix} A_B^{-1} & 0 \\ -a_B^T A_B^{-1} & 1 \end{pmatrix}.$$

Ist $\beta \geq a^T x$, so ist x zulässig für $(\hat{\mathrm{P}})$ und daher sogar optimal. Andernfalls ist $(\hat{x}, \hat{B})$ eine unzulässige Basislösung, mit der wir aber das duale Simplexverfahren starten können. Denn durch

$$\hat{y} := \hat{A}_{\hat{B}}^{-T} \hat{c}_{\hat{B}} = \begin{pmatrix} A_B^{-T} & -A_B^{-T} a \\ 0^T & 1 \end{pmatrix} \begin{pmatrix} c_B \\ 0 \end{pmatrix} = \begin{pmatrix} y \\ 0 \end{pmatrix} \quad \text{mit} \quad y := A_B^{-T} c_B$$

ist eine dual zulässige Lösung gefunden, da die zugehörigen reduzierten Kosten wegen der Optimalität von x (bzw. y) nichtnegativ sind. Die eben angegebene Eigenschaft des dualen Simplexverfahrens, nach Einführung einer zusätzlichen Ungleichungsrestriktion sofort starten zu können, ist insbesondere bei Schnitt-Verfahren für ganzzahlige lineare Optimierungsaufgaben von Bedeutung. □

Beispiel: Gegeben sei ein lineares Programm in Standardform mit den Daten

$$\begin{array}{|c|c|}\hline c^T & \\ \hline A & b \\ \hline \end{array} := \begin{array}{|rrrrr|}\hline -2 & -1 & 0 & 0 & 0 & \\ \hline 1 & 1 & 1 & 0 & 0 & 5 \\ -1 & 1 & 0 & 1 & 0 & 0 \\ 6 & 2 & 0 & 0 & 1 & 21 \\ \hline \end{array}$$

Eine Lösung mit Hilfe des (primalen) revidierten Simplexverfahren liefert die optimale Basisindexmenge, den zugehörigen Basisanteil sowie die Lösung des dualen Problems, nämlich

$$B = \{2, 4, 1\}, \qquad x_B = (2.25, 0.50, 2.75)^T, \qquad y = (-0.50, 0, -0.25)^T,$$

ferner

$$A_B^{-1} = \begin{pmatrix} 1.5 & 0 & -0.25 \\ -2.0 & 1.0 & 0.50 \\ -0.5 & 0 & 0.25 \end{pmatrix}.$$

Nachträglich wird die zusätzliche Restriktion (wie in obiger Bemerkung sei es eine $\leq$-Bedingung) $-x_5 \leq -1$ gefordert. Jetzt haben wir ein lineares Programm in Standardform mit den Daten

$$\begin{array}{|c|c|}\hline \hat{c}^T & \\ \hline \hat{A} & \hat{b} \\ \hline \end{array} := \begin{array}{|rrrrrr|r|}\hline -2 & -1 & 0 & 0 & 0 & 0 & \\ \hline 1 & 1 & 1 & 0 & 0 & 0 & 5 \\ -1 & 1 & 0 & 1 & 0 & 0 & 0 \\ 6 & 2 & 0 & 0 & 1 & 0 & 21 \\ 0 & 0 & 0 & 0 & -1 & 1 & -1 \\ \hline \end{array}$$

zu lösen. Wir hatten uns überlegt, dass durch

$$\hat{B} = \{B, 6\}, \qquad x_{\hat{B}} = \begin{pmatrix} x_B \\ -1 \end{pmatrix}$$

eine Basislösung für das neue Problem gegeben ist. Man kann das duale Simplexverfahren mit den vorhandenen Informationen starten. Nach

```
A=[1 1 1 0 0; -1 1 0 1 0;6 2 0 0 1];
b=[5;0;21];c=[-2;-1;0;0;0];B=[3 4 5];
A_B=A(:,B);in=inv(A_B);x=in*b;wert=c(B)'*x;
info=0;
while info==0
   [B,x,wert,in,y,info]=simplex(A,b,c,B,x,wert,in);
end;
A=[A zeros(3,1);zeros(1,5) 1];A(4,5)=-1;c=[c;0];b=[b;-1];
B=[B 6];x=[x;-1];y=[y;0];in=[in zeros(3,1);zeros(1,3) 1];
info=0;
while info==0
   [B,x,wert,in,y,info]=dualsimplex(A,b,c,B,x,wert,in,y);
end;
```

erhalten wir

$$B = \{2, 4, 1, 5\}, \qquad x_B = (2.5, 0, 2.5, 1)^T, \qquad y = (-0.5, 0, -0.25, -0.25)^T$$

als optimale Basisindizes, zugehörigen Basisanteil und duale Lösung. □

Beispiel: Bei gegebenen $A \in \mathbb{R}^{m\times n}$ und $b \in \mathbb{R}^m$ betrachte man die sogenannte diskrete, lineare Tschebyscheffsche Approximationsaufgabe,

$$f(x) := \|Ax - b\|_\infty$$

auf dem $\mathbb{R}^n$ zu minimieren. Hierbei ist $\|\cdot\|_\infty$ die Maximum- oder auch Tschebyscheff-Norm auf dem $\mathbb{R}^m$, definiert durch

$$\|y\|_\infty := \max_{i=1,\ldots,m} |y_i|.$$

Diese Aufgabe ist äquivalent dem linearen Programm

(P) Minimiere δ unter den Nebenbedingungen $\delta \geq 0, \quad -\delta e \leq Ax - b \leq \delta e,$

wobei $e = (1,\ldots,1)^T \in \mathbb{R}^m$. Wegen des Existenzsatzes der linearen Optimierung (die Aufgabe (P) ist zulässig und $\inf(\mathrm{P}) \geq 0$) besitzt (P) eine Lösung. Nach Einführung von Schlupfvariablen und einer Darstellung der freien Variablen x als Differenz von vorzeichenbeschränkten Variablen erhält man das äquivalente Problem in Normalform

$$(\mathrm{P}) \quad \begin{cases} \text{Minimiere} \quad \begin{pmatrix} 0\\0\\1\\0\\0 \end{pmatrix}^T \begin{pmatrix} x_+\\x_-\\\delta\\y\\z \end{pmatrix} \quad \text{unter den Nebenbedingungen} \\ \begin{pmatrix} x_+\\x_-\\\delta\\y\\z \end{pmatrix} \geq 0, \quad \begin{pmatrix} A & -A & e & -I & -0\\ -A & A & e & 0 & -I \end{pmatrix} \begin{pmatrix} x_+\\x_-\\\delta\\y\\z \end{pmatrix} = \begin{pmatrix} b\\-b \end{pmatrix}. \end{cases}$$

Stellt man die Daten dieses linearen Programms in der üblichen Weise dar, so erhält man

0^T	0^T	1	0^T	0^T	
A	$-A$	e	$-I$	0	b
$-A$	A	e	0	$-I$	$-b$

Offenbar kann das duale Simplexverfahren sofort gestartet werden. Da i. allg. aber $m \gg n$, ist es allerdings meistens besser, das primale Simplexverfahren auf das zu (P) duale Programm anzuwenden, und aus einem optimalen Tableau die eigentlich interessierende Lösung zu erhalten. Das zu (P) duale lineare Programm ist nämlich

$$(\mathrm{D}) \quad \begin{cases} \text{Maximiere} \quad \begin{pmatrix} b\\-b \end{pmatrix}^T \begin{pmatrix} u\\v \end{pmatrix} \quad \text{unter den Nebenbedingungen} \\ \begin{pmatrix} A^T & -A^T\\ -A^T & A\\ e^T & e^T\\ -I & 0\\ 0 & -I \end{pmatrix} \begin{pmatrix} u\\v \end{pmatrix} \leq \begin{pmatrix} 0\\0\\1\\0\\0 \end{pmatrix}. \end{cases}$$

Offenbar sind die Variablen dieses dualen Problems sozusagen automatisch vorzeichenbeschränkt, außerdem hat man nur eine skalare Ungleichung und sonst nur Gleichungen. Nach Überführung in Normalform (mache aus der Maximierungs- eine Minimierungsaufgabe, führe ferner eine skalare Schlupfvariable ein) sind die Daten dieses Problems in der üblichen Weise durch

$-b^T$	b^T	0	
A^T	$-A^T$	0	0
e^T	e^T	1	1

gegeben. Auf dieses Problem kann die Phase I des Simplexverfahrens angewandt werden, wobei berücksichtigt werden sollte, dass ein Einheitsvektor in der Koeffizientenmatrix schon vorhanden ist. Ferner beachte man, dass

$$\text{Rang} \begin{pmatrix} A^T & -A^T \\ e^T & e^T \end{pmatrix} = n+1,$$

wenn Rang $(A) = n$. Denn die $n+1$ Spalten von

$$\begin{pmatrix} A & e \\ -A & e \end{pmatrix}$$

sind offensichtlich linear unabhängig. In diesem Falle brauchen in der Phase I keine Gleichungen gestrichen zu werden. □

2.2.5 Das Simplexverfahren bei Box-Constraints

Bisher sind wir bei der Anwendung des Simplexverfahrens stets von einem linearen Programm in Normalform ausgegangen. Zwar kann ein allgemeines lineares Programm auf äquivalente Normalform gebracht werden. Dies hat aber den Nachteil, dass sich die Anzahl der Variablen beträchtlich erhöhen kann. Es stellt sich daher die Frage, ob diese explizite Überführung in Normalform wirklich nötig ist, insbesondere dann, wenn besonders "einfache" Restriktionen vorkommen. Exemplarisch wollen wir den Fall betrachten, dass für gewisse Variable untere und obere obere Schranken gegeben sind (man spricht von Box-Constraints), sonst aber nur Gleichungen in den Restriktionen vorkommen.

Gegeben sei die lineare Optimierungsaufgabe

$$\text{(P)} \qquad \text{Minimiere} \quad c^T x \quad \text{auf} \quad M := \{x \in \mathbb{R}^n : l \le x \le u,\ Ax = b\}.$$

Wie üblich sei hier $A = (\ a_1 \ \cdots \ a_n\ ) \in \mathbb{R}^{m\times n}$, $b \in \mathbb{R}^m$ und $c \in \mathbb{R}^n$, ferner seien l, u zwei Vektoren des $\mathbb{R}^n$ mit $l < u$, wobei allerdings die Komponenten von l auch gleich $-\infty$ und die von u gleich $+\infty$ sein können. Eine Variable x_j (oder einfacher nur j) heißt *unbeschränkt* oder *frei*, wenn $l_j = -\infty$ und $u_j = +\infty$, andernfalls *beschränkt*.

Zunächst übertragen wir die Definition einer Basislösung bzw. einer zulässigen Basislösung auf den hier vorliegenden Fall.

Definition 2.10 Ein $x \in \mathbb{R}^n$ mit $Ax = b$ heißt *Basislösung* zu (P), wenn eine Indexmenge $B \subset \{1, \ldots, n\}$ mit $\#(B) = m$ derart existiert, dass

(a) Die Matrix A_B ist nichtsingulär (bzw. die zur Indexmenge B gehörenden m Spalten von A sind linear unabhängig),

(b) Für alle beschränkten $j \in N := \{1, \ldots, n\} \setminus B$ ist $x_j = l_j$ oder $x_j = u_j$.

Eine Basislösung x heißt *zulässig*, wenn $l \leq x \leq u$.

Bemerkung: Im Gegensatz zum rein vorzeichenbeschränkten Fall ist hier eine Basislösung nicht durch die Menge der Basisindizes festgelegt. Eine zulässige Basislösung x ist offenbar eine Ecke der Menge M der zulässigen Lösungen im linearen Programm (P), wenn alle Nichtbasisindizes beschränkt sind. Denn angenommen, es ist $x = (1-\lambda)x^1 + \lambda x^2$ mit x^1, $x^2 \in M$ und $\lambda \in (0,1)$. Für alle beschränkten $j \in N$ ist $x_j = x_j^1 = x_j^2$ bzw. $x_N = x_N^1 = x_N^2$. Dann ist aber auch $x_B = x_B^1 = x_B^2$. □

Die Phase II des revidierten Simplexverfahrens kann nun leicht übertragen werden. Wir wollen nicht in die Details gehen, aber den nächsten Satz formulieren und beweisen.

Satz 2.11 *Gegeben sei das lineare Programm (P) mit beschränkten Variablen. Sei $x \in M$ eine zulässige Basislösung mit Basisindexmenge B und Nichtbasisindizes $N := \{1, \ldots, n\} \setminus B$. Sei weiter $y := A_B^{-T} c_B$ und $\overline{c}_N := c_N - A_N^T y$. Dann gilt:*

(a) Ist ($\overline{c}_j \geq 0$ oder $x_j = u_j$) und ($\overline{c}_j \leq 0$ oder $x_j = l_j$) für alle $j \in N$ bzw. gilt

$$\overline{c}_j < 0 \Longrightarrow x_j = u_j, \qquad \overline{c}_j > 0 \Longrightarrow x_j = l_j$$

für alle $j \in N$, so ist x eine Lösung von (P).

(b) Sei $s \in N$ ein Nichtbasisindex mit ($\overline{c}_s < 0$ und $x_s < u_s$) oder ($\overline{c}_s > 0$ und $x_s > l_s$)[19]. Mit $w := A_B^{-1} a_s$ definiere man

$$x_s(t) := \begin{cases} x_s - t & \text{für } \overline{c}_s > 0, \\ x_s + t & \text{für } \overline{c}_s < 0, \end{cases} \qquad x_B(t) := \begin{cases} x_B + tw & \text{für } \overline{c}_s > 0, \\ x_B - tw & \text{für } \overline{c}_s < 0 \end{cases}$$

und schließlich $x_j(t) := x_j$ für alle $j \in N \setminus \{s\}$. Dann ist $Ax(t) = b$ und $c^T x(t) = c^T x - t\,|\overline{c}_s|$ für alle t.

(c) Sei

$$t^* := \sup\{t \geq 0 : l_s \leq x_s(t) \leq u_s,\ l_B \leq x_B(t) \leq u_B\}.$$

Ist $t^* = +\infty$, so ist $\inf(\mathrm{P}) = -\infty$, also (P) nicht lösbar.

Beweis: Sei $z \in M$ beliebig. Insbesondere ist $z_B = A_B^{-1} b - A_B^{-1} A_N z_N$, entsprechend natürlich auch $x_B = A_B^{-1} b - A_B^{-1} A_N x_N$. Dann ist

$$\begin{aligned} c^T z - c^T x &= c_B^T z_B + c_N^T z_N - (c_B^T x_B + c_N^T x_N) \\ &= (c_N - A_N^T y)^T (z_N - x_N) \\ &= \sum_{j \in N} \overline{c}_j (z_j - x_j) \\ &\geq 0. \end{aligned}$$

[19] Sind alle Variablen beschränkt, was z. B. für $l := 0$ der Fall ist, so ist ($\overline{c}_s < 0$ und $x_s = 0$) oder ($\overline{c}_s > 0$ und $x_s = u_s$).

Denn ist $\overline{c}_j \geq 0$ und $x_j = l_j$, so ist natürlich $\overline{c}_j(z_j - x_j) \geq 0$. Ist dagegen $x_j = u_j$ und $\overline{c}_j \leq 0$, so ist ebenfalls $\overline{c}_j(z_j - x_j) \geq 0$, so dass die Optimalitätsbedingung (a) bewiesen ist.

Sei $s \in N$ wie angegeben gewählt. Es ist

$$\begin{aligned} Ax(t) &= A_B x_B(t) + x_s(t)a_s + A_{N\setminus\{s\}} x_{N\setminus\{s\}}(t) \\ &= A_B(x_B + \operatorname{sign}(\overline{c}_s)tw) + (x_s - \operatorname{sign}(\overline{c}_s)t)a_s + A_{N\setminus\{s\}} x_{N\setminus\{s\}} \\ &= A_B x_B + x_s a_s + A_{N\setminus\{s\}} x_{N\setminus\{s\}} \\ &= Ax \\ &= b \end{aligned}$$

für alle t. Entsprechend ist

$$\begin{aligned} c^T x(t) &= c_B^T x_B(t) + c_s x_s(t) + c_{N\setminus\{s\}}^T x_{N\setminus\{s\}} \\ &= c_B^T(x_B + \operatorname{sign}(\overline{c}_s)tw) + c_s(x_s - \operatorname{sign}(\overline{c}_s)t) + c_{N\setminus\{s\}}^T x_{N\setminus\{s\}} \\ &= c^T x + t\operatorname{sign}(\overline{c}_s)c_B^T w - t\operatorname{sign}(\overline{c}_s)c_s \\ &= c^T x - t\operatorname{sign}(\overline{c}_s)(c_s - a_s^T y) \\ &= c^T x - t\,|\overline{c}_s|. \end{aligned}$$

Teil (c) ist nun trivial. Ist nämlich $t^* = +\infty$, so ist $x(t) \in M$ für alle $t \geq 0$, aus $c^T x(t) = c^T x - |\overline{c}_s|\, t$ folgt mit $t \to +\infty$, dass $\inf(\mathrm{P}) = -\infty$. Damit ist der Satz bewiesen. □

Nun ist es ziemlich klar, wie es weiter geht. Wir nehmen an, es sei ein Nichtbasisindex $s \in N$ wie in (b) gewählt (wenn ein solcher Index nicht existiert, so sind wir fertig). Man definiere t^* wie in (c), nehme $t^* \in [0, \infty)$ an (andernfalls ist $\inf(\mathrm{P}) = -\infty$) und berechne anschließend die neue Näherung $x^+ := x(t^*)$. Jetzt muss man sich überlegen, dass x^+ eine zulässige Basislösung zu einer geeigneten Menge B^+ von Basisindizes ist. Klar ist, dass $Ax^+ = b$ (wegen Teil (b) in Satz 2.11) und $l \leq x^+ \leq u$ (nach Definition von t^*), also x^+ zulässig für (P) ist. Sei $B = \{j(1), \ldots, j(m)\}$. Wir machen eine Fallunterscheidung:

- Es ist

$$t^* = \sup\{t \geq 0 : l_B \leq x_B(t) \leq u_B\} < \sup\{t \geq 0 : l_s \leq x_s(t) \leq u_s\}.$$

Dann bestimme $j(r) \in B$ derart, dass

$$\sup\{t \geq 0 : l_{j(r)} \leq x_{j(r)}(t) \leq u_{j(r)}\} = t^*.$$

Wir wollen uns überlegen, dass dann x^+ eine zulässige Basislösung zur Basisindexmenge

$$B^+ := \{j(1), \ldots, j(r-1), s, j(r+1), \ldots, j(m)\}$$

ist. Da nämlich $w_r \neq 0$ (andernfalls wäre $t^* = +\infty$) ist

$$A_{B^+} = \begin{pmatrix} a_{j(1)} & \cdots & a_{j(r-1)} & a_s & a_{j(r+1)} & \cdots & a_{j(m)} \end{pmatrix} = A_B + (a_s - a_{j(r)})e_r^T$$

wegen des Lemmas von Sherman-Morrison nichtsingulär. Mit

$$N^+ := \{1, \ldots, n\} \setminus B^+ = (N \setminus \{s\}) \cup \{j(r)\}$$

ist offensichtlich $x_j^+ = l_j$ oder $x_j^+ = u_j$ für alle beschränkten $j \in N^+$, also insgesamt (x^+, B^+) eine zulässige Basislösung.

- Es ist

$$t^* = \sup\{t \geq 0 : l_s \leq x_s(t) \leq u_s\} \leq \sup\{t \geq 0 : l_B \leq x_B(t) \leq u_B\}.$$

Dieser Fall kann nur eintreten, wenn s eine beschränkte Variable ist und $x_s^+ = l_s$ oder $x_s^+ = u_s$. Man setze $B^+ := B$. Ganz offensichtlich ist x^+ eine zulässige Basislösung zur Basis B^+.

2.2.6 Der primal-duale Algorithmus

Als eine Verallgemeinerung ähnlicher Verfahren bei Netzwerkfluss- und Transportproblemen entwickelten Dantzig, Ford und Fulkerson 1956 den primal-dualen Algorithmus. Die Idee hierzu ist die folgende. Man starte mit einer dual zulässigen Lösung y und suche eine primal zulässige Lösung x, welche der Gleichgewichtsbedingung genügt. Hat man eine solche gefunden, so ist man fertig, da x und y primal bzw. dual optimal sind. War dies nicht möglich, so gibt der primal-duale Algorithmus eine Modifikation von y an, mit der neu gestartet wird.

Gegeben sei das lineare Programm in Normalform

$$\text{(P)} \qquad \text{Minimiere} \quad c^T x \quad \text{auf} \quad M := \{x \in \mathbb{R}^n : x \geq 0,\ Ax = b\},$$

wobei, wie üblich, $A = (\ a_1 \ \cdots \ a_n\ ) \in \mathbb{R}^{m \times n}$, $b = (b_i) \in \mathbb{R}^m$ und $c = (c_j) \in \mathbb{R}^n$. Wir setzen voraus, daß $b \geq 0$, was durch eine eventuelle Multiplikation entsprechender Gleichungen mit -1 erreicht werden kann. Das zu (P) duale lineare Programm ist

$$\text{(D)} \qquad \text{Maximiere} \quad b^T y \quad \text{auf} \quad N := \{y \in \mathbb{R}^m : A^T y \leq c\}.$$

Ist $x \in M$ und $y \in N$, so sind x und y primal bzw. dual optimal, wenn $c^T x = b^T y$ bzw. die Gleichgewichtsbedingung $(c - A^T y)^T x = 0$ erfüllt ist. Mit einem gegebenen $y \in N$ wird man also versuchen, ein $x \in M$ zu bestimmen, das der Gleichgewichtsbedingung genügt, und zwar dadurch, dass wir ein gewisses Hilfsproblem lösen. Selbst wenn dieser Versuch nicht erfolgreich war (erfolgversprechend ist er schließlich nur, wenn y dual optimal ist), werden wir Informationen gewinnen, mit denen wir die dual zulässige Lösung y verbessern können. Im Prinzip sieht der primal-duale Algorithmus folgendermaßen aus:

1. Sei $y \in N$, also y dual zulässig.

2. Sei $J := \{j \in \{1,\dots,n\} : a_j^T y = c_j\}$, $K := \{1,\dots,n\} \setminus J$. Betrachte die Optimierungsaufgabe

$$(\hat{\mathrm{P}}) \qquad \begin{cases} \text{Minimiere } \begin{pmatrix} 0 \\ e \end{pmatrix}^T \begin{pmatrix} x_J \\ z \end{pmatrix} \quad \text{unter den Nebenbedingungen} \\ \begin{pmatrix} x_J \\ z \end{pmatrix} \geq 0, \quad \begin{pmatrix} A_J & I \end{pmatrix} \begin{pmatrix} x_J \\ z \end{pmatrix} = b. \end{cases}$$

   Wegen des Existenzsatzes der linearen Optimierung besitzt diese Aufgabe eine Lösung (x_J, z), welche mit der Phase II des Simplexverfahrens berechnet werden kann.

   Also versucht man hier, ganz ähnlich wie bei der Phase I des primalen Simplexverfahrens, mit Hilfe künstlicher Variabler eine Lösung von $A_J x_J = b$, $x_J \geq 0$, zu bestimmen.

3. Falls $\min(\hat{\mathrm{P}}) = 0$, dann: STOP. Setzt man $x_j := 0$ für $j \notin J$ bzw. $x_K := 0$, so sind x primal und y dual optimal.

4. Falls $\min(\hat{\mathrm{P}}) > 0$, so betrachte die zu $(\hat{\mathrm{P}})$ duale Aufgabe, nämlich

$$(\hat{\mathrm{D}}) \qquad \begin{cases} \text{Maximiere} \quad b^T u \quad \text{unter den Nebenbedingungen} \\ \begin{pmatrix} A_J^T \\ I \end{pmatrix} u \leq \begin{pmatrix} 0 \\ e \end{pmatrix}. \end{cases}$$

   Sei $\overline{u}$ eine Lösung von $(\hat{\mathrm{D}})$ (normalerweise wird $\overline{u}$ bei der Lösung von $(\hat{P})$ abfallen).

   (a) Falls $A_K^T \overline{u} \leq 0$, dann: STOP, da $\sup(\mathrm{D}) = +\infty$ bzw. (P) nicht zulässig ist. Denn: Es ist $b^T \overline{u} = \min(\hat{\mathrm{P}}) > 0$. Man definiere $y(t) := y + t\overline{u}$. Wegen $A^T \overline{u} \leq 0$ und $y \in N$ ist $y(t) \in N$ für alle $t \geq 0$. Außerdem ist $b^T y(t) = b^T y + t b^T \overline{u} \to +\infty$ mit $t \to +\infty$.

   (b) Falls $A_K^T \overline{u} \not\leq 0$, so berechne

$$t^* := \min_{j \in K} \Big\{ \frac{c_j - a_j^T y}{a_j^T \overline{u}} : a_j^T \overline{u} > 0 \Big\}, \qquad y^+ := y + t^* \overline{u}.$$

   Nach Definition von t^* ist $y^+ \in N$, ferner ist $t^* > 0$ und

$$b^T y^+ = b^T y + \underbrace{t^* b^T \overline{u}}_{>0} > b^T y,$$

   also y^+ eine "bessere" dual zulässige Lösung.

Bemerkungen: Natürlich ist das primal-duale Verfahren nur dann sinnvoll einsetzbar, wenn das Hilfsproblem $(\hat{\mathrm{P}})$ bzw. das dazu duale Programm $(\hat{\mathrm{D}})$ verhältnismäßig einfach zu lösen ist. In Spezialfällen werden wir uns hierüber Gedanken machen. Gedanken sollte man sich auch darüber machen, wie eine dual zulässige Lösung bestimmt werden

kann, was hier allerdings nicht geschehen soll. Die Hauptarbeit im primal-dualen Algorithmus besteht in der Lösung von $(\hat{\mathrm{P}})$ (und dem dazu dualen Programm $(\hat{\mathrm{D}})$). Die Aufgabe $(\hat{\mathrm{P}})$ ist ein lineares Programm mit m Gleichungen und $\#(J) + m$ Variablen. Sei B eine optimale Basisindexmenge und $j \in J \cap B$. Wir wollen uns überlegen, daß dann auch $j \in J^+$, wobei $J^+ \subset \{1, \ldots, n\}$ die Indexmenge derjenigen Ungleichungsrestriktionen im dualen Problem ist, die für die neue Iterierte y^+ aktiv sind. Denn für $j \in J \cap B$ ist $0 = a_j^T \overline{u}$ (im revidierten Simplexverfahren ist $\overline{u} = A_B^{-T} c_B$ bzw. $c_B = A_B^T \overline{u}$) und daher

$$a_j^T y^+ = \underbrace{a_j^T y}_{=c_j} + t^* \underbrace{a_j^T \overline{u}}_{=0} = c_j.$$

Das hat die angenehme Konsequenz, daß man zur Lösung des neuen Problems $(\hat{\mathrm{P}}^+)$ mit dem optimalen Tableau von $(\hat{\mathrm{P}})$ starten kann. □

2.2.7 Aufgaben

1. Man löse mit Hilfe des revidierten Simplexverfahrens ein lineares Programm in Standardform, bei dem die Daten durch

$$\begin{array}{|c|c|}\hline c^T & \\ \hline A & b \\ \hline \end{array} := \begin{array}{|cccccc|c|}\hline 14 & -19 & 0 & 21 & 52 & 0 & \\ \hline 1 & 0 & 0 & -1 & 1 & 1 & 3 \\ 1 & 1 & 0 & -1 & 3 & 0 & 4 \\ 1 & 1 & 1 & -3 & 0 & 1 & 6 \\ \hline \end{array}$$

   gegeben sind, wobei man mit der Basis $B := \{1, 2, 6\}$ starte.

2. Man beweise Lemma 2.8.

3. Man beweise Satz 2.9.

4. In einer Reifenfabrik[20] werden Sommer- und Winterreifen hergestellt. Die Fabrik hat Verträge, bis zu bestimmten Daten eine gewisse Zahl von Reifen mindestens herzustellen, nämlich

Zeitpunkt	Sommerreifen	Winterreifen
30. Juni	5 000	3 000
31. Juli	6 000	3 000
31. August	4 000	5 000

   Zur Produktion stehen zwei Typen von Maschinen zur Verfügung. Die Anzahl der zur Verfügung stehenden Produktionsstunden für die beiden Maschinen während der Sommermonate sind:

Monat	Maschine A	Maschine B
Juni	700	1 500
Juli	300	400
August	1 000	300

[20] Die Aufgabe ist

M. Asghar Bhatti (2000) *Practical Optimization Methods. With Mathematica Applications*. Springer-Verlag, New York-Berlin-Heidelberg

entnommen.

Die Produktionsraten (Stunden pro Reifen) auf den beiden Typen von Maschinen sind

Typ	Maschine A	Maschine B
Sommerreifen	0.15	0.16
Winterreifen	0.12	0.14

Unabhängig von den benutzten Typen und den produzierten Reifen kostet eine Arbeitsstunde 100 DM. Das Material für einen Sommerreifen kostet 52.50 DM, das für einen Winterreifen 41.50 DM. Pro Reifen kommen noch 4 DM hinzu. Überschüssige Reifen können in den nächsten Monat (also von Juni in den Juli und von Juli in den August) übernommen werden, die Lagerkosten sind 1.50 DM pro Reifen. Die produzierten Reifen werden für 200 DM (Sommerreifen) bzw. 150 DM (Winterreifen) verkauft. Wie[21] sollte die Produktion organisiert werden, um einerseits den Lieferbedingungen nachzukommen und andererseits den Gewinn der Fabrik zu maximieren?

5. Man löse das lineare Programm[22] in Standardform

   (P) Minimiere $c^T x$ unter den Nebenbedingungen $x \geq 0,\ Ax = b,$

   wobei die Daten durch

$$\left[\begin{array}{c|c} c^T & \\ \hline A & b \end{array}\right] := \left[\begin{array}{rrrrrrr|r} 5 & 3 & 3 & 6 & 0 & 0 & 0 & \\ \hline -6 & 1 & 2 & 4 & 1 & 0 & 0 & 14 \\ 3 & -2 & -1 & -5 & 0 & 1 & 0 & -25 \\ -2 & 1 & 0 & 2 & 0 & 0 & 1 & 14 \end{array}\right]$$

   gegeben sind, mit dem (revidierten) dualen Simplexverfahren.

6. Man schreibe ein Programm, das zu vorgegebenen $A \in \mathbb{R}^{m\times n}$ und $b \in \mathbb{R}^m$ eine Lösung der diskreten, linearen Tschebyscheffschen Approximationsaufgabe, $f(x) := \|Ax - b\|_\infty$ auf dem $\mathbb{R}^n$ zu minimieren, berechnet. Der Einfachheit halber werde Rang $(A) = n$ vorausgesetzt. Anschließend teste man das Programm an den Daten $A = (a_{ij}) \in \mathbb{R}^{11\times 5}$, $b = (b_i) \in \mathbb{R}^{11}$, wobei die Einträge mit $t_i := (i-1)\frac{1}{10}$, $i = 1, \ldots, 11$, durch

$$a_{ij} := t_i^{j-1}, \qquad b_i := \exp(t_i) \qquad (i = 1, \ldots, 11,\ j = 1, \ldots, 5)$$

   gegeben sind (wobei $0^0 = 1$).

7. Man benutze[23] das Simplexverfahren um nachzuweisen, dass für alle $t \in (-3, -\frac{1}{2})$ die eindeutige Lösung von

$$(\mathrm{P}_t) \quad \text{Minimiere} \quad tx_1 - x_2 \quad \text{unter den Nebenbedingungen} \quad \begin{array}{l} x_1 + 2x_2 \leq 4, \\ 6x_1 + 2x_2 \leq 9, \end{array} \quad x \geq 0$$

   in einem von t unabhängigen Punkt x^* angenommen wird. Man berechne $x^* = (x_1^*, x_2^*)$.

[21] Eigentlich handelt es sich hier um ein ganzzahliges lineares Programm, wovon wir aber absehen wollen.

[22] Siehe V. Chvátal (1983, S. 155).

[23] Diese Aufgabe wurde

R. Fletcher (1987, S. 190) *Practical Methods of Optimization*. J. Wiley, Chichester

entnommen.

8. Als Verallgemeinerung von Aufgabe 7 überlege man sich: Gegeben sei das lineare Programm

   $$(\mathrm{P}_t) \qquad \text{Minimiere} \quad (c+td)^T x \quad \text{auf} \quad M := \{x \in \mathbb{R}^n : x \geq 0,\ Ax = b\}.$$

   Mit Hilfe des Simplexverfahrens sei eine optimale Basisindexmenge B und die zugehörige optimale Basislösung x^* von (P_0) bestimmt. Es sei $\overline{c}_N := c_N - A_N^T A_B^{-T} c_B > 0$ mit $N := \{1, \ldots, n\} \setminus B$. Für welche t ist x^* auch Lösung von (P_t)?

9. Gegeben sei das lineare Programm in Standardform

   $$(\mathrm{P}) \qquad \text{Minimiere} \quad c^T x \quad \text{auf} \quad M := \{x \in \mathbb{R}^n : x \geq 0,\ Ax = b\}.$$

   Sei Rang $(A) = m$ und $x^* \in M$ eine nichtentartete, optimale Ecke von M, d. h. x^* ist eine Lösung von (P), die Indexmenge $B := \{j \in \{1, \ldots, n\} : x_j^* > 0\}$ enthält genau m Elemente und die Matrix $A_B \in \mathbb{R}^{m \times m}$ ist nichtsingulär. Man zeige:

   (a) Es ist $y^* := A_B^{-T} c_B$ die eindeutige Lösung des zu (P) dualen linearen Programms

   $$(\mathrm{D}) \qquad \text{Maximiere} \quad b^T y \quad \text{auf} \quad N := \{y \in \mathbb{R}^n : A^T y \leq c\}.$$

   (b) Für beliebiges $d \in \mathbb{R}^m$ ist das gestörte Problem

   $$(\mathrm{P}_t) \qquad \text{Minimiere} \quad c^T x \quad \text{auf} \quad M_t := \{x \in \mathbb{R}^n : x \geq 0,\ Ax = b + td\}.$$

   für alle hinreichend kleinen $|t|$ lösbar und es ist

   $$\min(\mathrm{P}_t) = \min(\mathrm{P}) + td^T y^* \qquad \text{für alle hinreichend kleinen } |t|.$$

10. Mit Hilfe des Simplexverfahrens sei eine (optimale Basis-) Lösung x^* von

    $$(\mathrm{P}) \qquad \text{Minimiere} \quad c^T x \quad \text{auf} \quad M := \{x \in \mathbb{R}^n : x \geq 0,\ Ax = b\}$$

    berechnet. Hierbei sei $A \in \mathbb{R}^{m \times n}$ und Rang $(A) = m$. Wie würden Sie das Simplexverfahren für die erweiterte Aufgabe

    $$(\hat{\mathrm{P}}) \quad \begin{cases} \text{Minimiere} \quad \begin{pmatrix} c \\ c_{n+1} \end{pmatrix}^T \begin{pmatrix} x \\ x_{n+1} \end{pmatrix} \quad \text{auf} \\ \hat{M} := \left\{ \begin{pmatrix} x \\ x_{n+1} \end{pmatrix} \in \mathbb{R}^{n+1} : \begin{pmatrix} x \\ x_{n+1} \end{pmatrix} \geq 0,\ \begin{pmatrix} A & a_{n+1} \end{pmatrix} \begin{pmatrix} x \\ x_{n+1} \end{pmatrix} = b \right\} \end{cases}$$

    starten? Unter welcher Voraussetzung ist

    $$\hat{x} := \begin{pmatrix} x^* \\ 0 \end{pmatrix}$$

    eine Lösung von $(\hat{\mathrm{P}})$?

11. Gegeben sei die lineare Optimierungsaufgabe mit beschränkten Variablen

    $$(\mathrm{P}) \qquad \text{Minimiere} \quad \text{auf} \quad M := \{x \in \mathbb{R}^n : l \leq x \leq u,\ Ax = b\}.$$

Wie üblich sei hier $A \in \mathbb{R}^{m\times n}$, $b \in \mathbb{R}^m$ und $c \in \mathbb{R}^n$, ferner seien l, u zwei Vektoren des $\mathbb{R}^n$ mit $l < u$, wobei allerdings die Komponenten von l auch gleich $-\infty$ und die von u gleich $+\infty$ sein können. Eine Variable x_j (oder einfacher nur j) heißt *unbeschränkt* oder *frei*, wenn $l_j = -\infty$ und $u_j = +\infty$, andernfalls *beschränkt*. Sei $x \in M$ eine zulässige Basislösung von (P), es existiere also eine Indexmenge $B \subset \{1,\dots,n\}$ mit $\#(B) = m$ und der Eigenschaft, dass A_B nichtsingulär ist und für alle beschränkten $j \in N := \{1,\dots,n\} \setminus B$ entweder $x_j = l_j$ oder $x_j = u_j$ gilt. Man zeige, dass x eine Ecke von M ist, wenn alle Nichtbasisindizes beschränkt sind.

12. Man löse das lineare Programm[24]

$$\text{Minimiere} \quad -x_1 - x_2 \quad \text{unter den Nebenbedingungen} \quad \begin{aligned} x_1 \qquad + 2x_3 &\le 1 \\ x_2 - \ x_3 &\le 1 \\ x_1 + x_2 + \ x_3 &= 2, \end{aligned} \qquad x \ge 0,$$

indem man es zunächst auf Standardform bringt und mit Hilfe der Methode der künstlichen Variablen eine Anfangsbasislösung bestimmt.

13. Gegeben sei die lineare Optimierungsaufgabe

$$\text{Minimiere} \quad -4x_1 - x_2 \quad \text{unter den Nebenbedingungen} \quad \begin{aligned} 2x_1 + x_2 &\le 9 \\ -2x_1 + x_2 &\le 1 \\ x_1 &\le 3 \\ 1 \le x_2 &\le 5. \end{aligned}$$

Nach Einführung von Schlupfvariablen führt dies auf ein lineares Programm in Standardform (mit beschränkten Variablen)

$$\text{(P)} \qquad \text{Minimiere} \quad c^T x \quad \text{auf} \quad M := \{x \in \mathbb{R}^n : l \le x \le u,\ Ax = b\},$$

wobei die Daten durch

$$\left[\begin{array}{c|c} c^T & \\ \hline A & b \\ \hline \begin{matrix} l^T \\ u^T \end{matrix} & \end{array}\right] := \left[\begin{array}{cccc|c} -4 & -1 & 0 & 0 & \\ \hline 2 & 1 & 1 & 0 & 9 \\ -2 & 1 & 0 & 1 & 1 \\ \hline -\infty & 1 & 0 & 0 & \\ 3 & 5 & +\infty & +\infty & \end{array}\right]$$

gegeben sind. Durch $x := (0,1,8,0)^T$ ist eine zulässige Basislösung zur Basis $B := \{1,3\}$ gegeben. Hiervon ausgehend bestimme man mit dem (für beschränkte Variable) modifizierten Simplexverfahren die Lösung.

2.3 Innere-Punkt-Verfahren

Seit der Aufsehen erregenden Arbeit von N. KARMARKAR (1984)[25] sind Innere-Punkt-Verfahren, insbesondere bei linearen Optimierungsproblemen, außerordentlich gründlich untersucht worden. Inzwischen bilden Innere-Punkt-Verfahren eine ernste Konkurrenz für das Simplexverfahren, gerade bei hochdimensionalen Problemen. In der Optimization Toolbox von MATLAB gibt es zur Lösung linearer Programme die Funktion

[24] Siehe K. NEUMANN, M. MORLOCK (1993, S. 95).

[25] N. KARMARKAR (1984) "A new polynomial time algorithm for linear programming". Combinatorica 4, 373–395.

`linprog`. Gerade für hochdimensionale (large-scale) Probleme ist dort ein sogenanntes primal-duales Innere-Punkt-Verfahren implementiert. Den entsprechenden Technical Report von Y. ZHANG[26] kann man unter der Adresse `http://www.caam.rice.edu/~zhang/lipsol/` finden. Zu Beginn dieses Aufsatzes kann man nachlesen:

- After over a decade of extraordinarily active research triggered by the seminal work of Karmarkar, the field of interior-point methods has finally come to maturity as far as linear programming is concerned. Not only do we have a solid theoretical foundation for interior-point methods for linear programming, but also a rather comprehensive understanding on their practical efficiency. Among many general algorithmic approaches, the most effective one in practice has proven to be the primal-dual infeasible-interior-point approach, ...

  Recent experiments indicate that as the problem size increases, so does the frequency of interior-point methods outperforming the classic simplex method. This trend has been observed for a while but is becoming more pronounced recently. In our opinion, this marks the beginning of a new era in which interior-point methods coexist with the simplex method but gradually assume the role of dominant computational engine for general large-scale linear programming.

Wir werden versuchen, die theoretischen Grundlagen von Innere-Punkt-Verfahren bei linearen Optimierungsaufgaben zu legen und etwas zur Implementation zu sagen.

2.3.1 Grundlagen

Gegeben sei das lineare Programm

$$\text{(P)} \qquad \text{Minimiere} \quad c^T \quad \text{auf} \quad M := \{x \in \mathbb{R}^n : x \geq 0,\ Ax = b\}.$$

Wie üblich seien hierbei $A \in \mathbb{R}^{m\times n}$, $b \in \mathbb{R}^m$ und $c \in \mathbb{R}^n$. Das zu (P) duale lineare Programm[27] ist

$$\text{(D)} \qquad \text{Maximiere} \quad b^T y \quad \text{auf} \quad N := \{y \in \mathbb{R}^m : A^T y \leq c\}.$$

Häufig wird vorausgesetzt, dass

(A1) Es ist $M_0 := \{x \in \mathbb{R}^n : x > 0,\ Ax = b\} \neq \emptyset$.

(A2) Es ist $N_0 := \{y \in \mathbb{R}^m : A^T y < c\} \neq \emptyset$.

(A3) Es ist Rang $(A) = m$.

Dann gilt:

[26] Y. ZHANG (1997) "Solving large-scale linear programs by interior-point methods under the Matlab environment".

[27] In der Literatur wird das duale Programm häufig auch in der Form

$$\text{(D)} \qquad \text{Maximiere} \quad b^T y \quad \text{auf} \quad \{(y,z) \in \mathbb{R}^m \times \mathbb{R}^n : z \geq 0,\ A^T y + z = c\}$$

geschrieben. Für $x \in M$ und $(y,z) \in N$ ist dann durch $x^T z = c^T x - b^T y$ die Dualitätslücke gegeben.

Lemma 3.1 *Sind die Voraussetzungen (A1), (A2) und (A3) erfüllt, so sind die Mengen M_{opt} bzw. N_{opt} der Lösungen von (P) bzw. (D) nichtleer und kompakt.*

Beweis: Da insbesondere wegen (A1) und (A2) die Probleme (P) und (D) zulässig sind, folgt aus dem starken Dualitätssatz, dass (P) und (D) lösbar sind, bzw. $M_{\text{opt}} \neq \emptyset$ und $N_{\text{opt}} \neq \emptyset$. Da die Lösungsmengen trivialerweise abgeschlossen sind, ist ihre Beschränktheit zu zeigen.

Angenommen,

$$M_{\text{opt}} = \{x \in \mathbb{R}^n : x \geq 0,\ Ax = b,\ c^T x = \min (\text{P})\}$$

wäre nicht beschränkt. Dann existiert eine Folge $\{x_k\} \subset M_{\text{opt}}$ mit $\|x_k\| \to \infty$. Man definiere $p_k := x_k/\|x_k\|$ und beachte, dass man wegen der Kompaktheit der Einheitssphäre aus $\{p_k\}$ eine gegen ein $p \neq 0$ konvergente Teilfolge auswählen kann. Notwendigerweise ist offenbar

$$p \geq 0, \quad Ap = 0, \quad c^T p = 0.$$

Nach Voraussetzung (A2) existiert ein $y_0 \in N_0$. Aus

$$0 < p^T(c - A^T y_0) = \underbrace{c^T p}_{=0} - \underbrace{(Ap)^T y_0}_{=0} \leq 0$$

erhalten wir einen Widerspruch.

Nun nehmen wir an,

$$N_{\text{opt}} = \{y \in \mathbb{R}^m : A^T y \leq c,\ b^T y = \max (\text{D})\}$$

sei unbeschränkt. Analog der eben angegebenen Argumentation existiert ein $q \neq 0$ mit

$$A^T q \leq 0, \quad b^T q = 0.$$

Dann ist $A^T q = 0$, da andernfalls mit einem nach Voraussetzung (A1) existierenden $x_0 \in M_0$ die Ungleichung

$$0 > x_0^T (A^T q) = (Ax_0)^T q = b^T q = 0$$

gelten würde, ein Widerspruch. Wegen Voraussetzung (A3) folgt aus $A^T q = 0$, dass $q = 0$, ein Widerspruch. Damit ist das Lemma bewiesen. □

Entscheidend für die Idee der primal-dualen Innere-Punkt-Verfahren sind die folgenden drei Sätze, die für eine wesentlich größere Klasse von Optimierungsaufgaben als lineare Programme gelten (worauf aber nicht eingegangen werden soll). Mit einem $\mu > 0$ ordnen wir nämlich den beiden linearen Programmen (P) und (D) die beiden (im wesentlichen unrestringierten) Optimierungsaufgaben

$$(\text{P}_\mu) \qquad \text{Minimiere} \quad f_\mu(x) := c^T x - \mu \sum_{j=1}^{n} \log x_j, \quad x \in M_0$$

und

$$(\text{D}_\mu) \qquad \text{Maximiere} \quad g_\mu(y) := b^T y + \mu \sum_{j=1}^{n} \log(c - A^T y)_j, \quad y \in N_0$$

zu, wobei wir natürlich zumindestens (A1) und (A2) voraussetzen.

Satz 3.2 *Unter den Voraussetzungen (A1)–(A3) besitzen die Optimierungsaufgaben* (P_μ) *und* (D_μ) *für jedes* $\mu > 0$ *genau eine Lösung* $x_\mu \in M_0$ *bzw.* $y_\mu \in N_0$.

Beweis: Sei $\mu > 0$ vorgegeben. Zunächst zeigen wir die Existenz einer Lösung von (P_μ), danach deren Eindeutigkeit.

Sei $x_0 \in M_0$ beliebig. Wir definieren die Niveaumenge

$$L_\mu(x_0) := \{x \in M_0 : f_\mu(x) \leq f_\mu(x_0)\}.$$

Wir zeigen, dass $L_\mu(x_0)$ beschränkt ist. Ist dies nicht der Fall, so existiert eine Folge $\{x_k\} \subset L_\mu(x_0)$ mit $\|x_k\| \to \infty$ und $x_k/\|x_k\| \to p$. Wir werden nachweisen, dass dann $x^* + tp \in M_{\mathrm{opt}}$ für alle $t \geq 0$ mit einem beliebigen $x^* \in M_{\mathrm{opt}}$, was wegen $p \neq 0$ einen Widerspruch zu der Beschränktheit von M_{opt} ergibt (siehe Lemma 3.1). Da $p \geq 0$ und $Ap = 0$, ist natürlich $x^* + tp \in M$ für alle $t \geq 0$. Nun beachte man, dass mit einer geeigneten Konstanten $c_0 > 0$ für alle k gilt, dass

$$(x_k)_j \leq \|x_k\|_\infty \leq c_0 \|x_k\|, \qquad j = 1, \ldots, n,$$

so dass

$$c^T x_k - \mu n \log(c_0 \|x_k\|) \leq c^T x_k - \mu \sum_{j=1}^{n} \log(x_k)_j = f_\mu(x_k) \leq f_\mu(x_0).$$

Nach Division durch $\|x_k\|$ und den Grenzübergang $k \to \infty$ erhält man unter Berücksichtigung von $\log(c_0\|x_k\|)/\|x_k\| \to 0$, dass $c^T p \leq 0$. Dann ist aber $c^T(x^* + tp) \leq c^T x^*$ für alle $t \geq 0$, insgesamt also $x^* + tp \in M_{\mathrm{opt}}$ für alle $t \geq 0$. Damit haben wir einen Widerspruch erhalten und die Beschränktheit der Niveaumenge $L_\mu(x_0)$ nachgewiesen. Die Niveaumenge $L_\mu(x_0)$ ist aber auch abgeschlossen, insgesamt also kompakt. Denn sei $\{x_k\} \subset L_\mu(x_0)$ eine gegen ein $x \in \mathbb{R}^n$ konvergente Folge. Wegen $\{x_k\} \subset M_0$ ist natürlich $x \in M$. Da $\{f_\mu(x_k)\}$ nach oben beschränkt ist, ist natürlich sogar $x > 0$ bzw. $x \in M_0$. Da f_μ auf M_0 stetig ist, ist auch $x \in L_\mu(x_0)$, womit insgesamt die Kompaktheit von $L_\mu(x_0)$ nachgewiesen ist. Da außerdem f_μ auf $L_\mu(x_0)$ stetig ist, ist die Menge der Lösungen von (P_μ) nichtleer. Oder einfacher gesagt: (P_μ) besitzt für jedes $\mu > 0$ mindestens eine Lösung.

Nun zur Eindeutigkeit einer Lösung von (P_μ). Dies ist aber eine einfache Folgerung aus der Tatsache, dass die Zielfunktion f_μ von (P_μ) auf M_0 *strikt konvex* ist, also aus $x^*, x^{**} \in M_0$, $x^* \neq x^{**}$ und $t \in (0,1)$ die Ungleichung

$$f_\mu((1-t)x^* + tx^{**}) < (1-t)f_\mu(x^*) + tf_\mu(x^{**})$$

folgt, was wiederum eine einfache Folgerung der strikten Konkavität des Logarithmus auf $\mathbb{R}_+$ ist. Damit ist insgesamt die Existenz genau einer Lösung von (P_μ) nachgewiesen.

Der Nachweis der entsprechenden Aussagen für (D_μ) verläuft weitgehend analog. Daher werden wir uns kurz fassen und nur auf die (geringfügigen) Unterschiede hinweisen. Der Existenzbeweis verläuft (fast) völlig analog. Für den Eindeutigkeitsbeweis nehmen wir an, y^* und y^{**} seien zwei Lösungen von (D_μ). Aus Konvexitätsgründen ist auch $\frac{1}{2}(y^* + y^{**})$ eine Lösung von (D_μ), wegen der strikten Konkavität des Logarithmus auf $\mathbb{R}_+$ folgt $A^T y^* = A^T y^{**}$. Voraussetzung (A3) impliziert daher $y^* = y^{**}$, die Eindeutigkeit einer Lösung von (D_μ). Insgesamt ist der Satz bewiesen. □

Satz 3.3 *Die Voraussetzungen (A1)–(A3) seien erfüllt. Für $\mu > 0$ sei $x_\mu \in M_0$ bzw. $y_\mu \in N_0$ die nach Satz 3.2 eindeutige Lösung von (P_μ) bzw. (D_μ). Dann existieren $x^* := \lim_{\mu\to 0+} x_\mu$ und $y^* := \lim_{\mu\to 0+} y_\mu$ und sind (gewisse) Lösungen von (P) bzw. (D).*

Beweis: Im ersten Teil des Beweises zeigen wir:

- Ist $\{\mu_k\} \subset \mathbb{R}_+$ eine Nullfolge, so ist die Folge $\{x_k\}$, wobei wir zur Abkürzung $x_k := x_{\mu_k}$ gesetzt haben, beschränkt und jeder ihrer Häufungspunkte eine Lösung von (P).

Denn: Man wähle μ so groß, dass $\mu_k < \mu$ für alle k und $x_\mu \in M_0$ die Lösung von (P_μ). Aus $f_{\mu_k}(x_k) \le f_{\mu_k}(x_\mu)$ und $f_\mu(x_\mu) \le f_\mu(x_k)$ folgt nach Multiplikation der zweiten Ungleichung mit $\mu_k/\mu \in (0,1)$ und anschließender Addition (wodurch die logarithmischen Terme wegfallen), dass $(1-\mu_k/\mu)c^T x_k \le (1-\mu_k/\mu)c^T x_\mu$ bzw. $c^T x_k \le c^T x_\mu$. Niveaumengen zu (P) sind aber kompakt (Beweis?), daher ist $\{x_k\}$ beschränkt. Nun sei $x^* \in M_{\text{opt}}$ beliebig, also eine gewisse Lösung von (P). Da x_k eine Lösung von (P_{μ_k}) ist und $x_k + t(x^* - x_k) \in M_0$ für alle $t \in (0,1)$, ist

$$\begin{aligned}
0 &\le \lim_{k\to\infty} \frac{f_{\mu_k}(x_k + t(x^* - x_k)) - f_{\mu_k}(x_k)}{t} \\
&= \nabla f_{\mu_k}(x_k)^T (x^* - x_k) \\
&= c^T(x^* - x_k) - \mu_k \sum_{j=1}^{n} \frac{(x^*)_j - (x_k)_j}{(x_k)_j} \\
&\le c^T x^* - c^T x_k + \mu_k n \\
&= \min(\mathrm{P}) - c^T x_k + \mu_k n,
\end{aligned}$$

woraus offenbar folgt, dass jeder Häufungspunkt der Folge $\{x_k\}$ eine Lösung von (P) ist.

Analog gilt:

- Ist $\{\mu_k\} \subset \mathbb{R}_+$ eine Nullfolge, so ist die Folge $\{y_k\}$, wobei wir zur Abkürzung $y_k := y_{\mu_k}$ gesetzt haben, beschränkt und jeder ihrer Häufungspunkte eine Lösung von (D).

Der Beweis hierfür verläuft völlig analog dem vorigen (wovon man sichüberzeugen sollte). Als ummittelbare Folgerung erhalten wir:

- Sind (P) bzw. (D) eindeutig lösbar, so existieren $x^* = \lim_{\mu\to 0+} x_\mu$ bzw. $y^* = \lim_{\mu\to 0+} y_\mu$ und sind die eindeutigen Lösungen von (P) bzw. (D).

Denn: Eine beschränkte Folge, die genau einen Häufungspunkt besitzt, ist konvergent.

Im nächsten Schritt zeigen wir die behauptete Aussage für das Problem (P).

- Sei

$$I := \{i \in \{1,\dots,n\} : x_i = 0 \text{ für alle } x \in M_{\text{opt}}\}, \qquad J := \{1,\dots,n\} \setminus I.$$

Dann gilt:

– Die Optimierungsaufgabe

$$(\mathrm{P}_0) \qquad \begin{cases} \text{Minimiere} \quad f_0(x) := -\sum_{j\in J} \log x_j \quad \text{unter den NBen} \\ x \in M_{\mathrm{opt}}, \qquad x_j > 0 \quad (j \in J) \end{cases}$$

besitzt eine eindeutige Lösung x_0.

– Es ist $\lim_{\mu\to 0+} x_\mu = x_0$.

Denn: Wir können o. B. d. A. annehmen, daß $J \neq \emptyset$, denn andernfalls ist $I = \{1, \ldots, n\}$ und folglich $M_{\mathrm{opt}} = \{0\}$, in welchem Fall beide Aussagen trivialerweise richtig sind. Nun zeigen wir, dass (P_0) eindeutig lösbar ist. Zunächst ist (P_0) zulässig. Denn zu $j \in J$ existiert ein $x^{(j)} \in M_{\mathrm{opt}}$ mit $x_j^{(j)} > 0$. Offenbar ist dann $x := (1/\#(J)) \sum_{j\in J} x^{(j)}$ zulässig für (P_0). Die Existenz eine Lösung von (P_0) folgt aus der Beobachtung, dass zu (P_0) gehörende Niveaumengen kompakt sind. Die Eindeutigkeit einer Lösung von (P_0) folgt schließlich wieder aus der strikten Konvexität der Zielfunktion f_0 auf der Menge der zulässigen Lösungen. Damit ist der erste Teil bewiesen.

Um den zweiten Teil zu beweisen, geben wir uns eine Nullfolge $\{\mu_k\} \subset \mathbb{R}_+$ und einen Häufungspunkt x^* der Folge $\{x_k\}$ vor (wieder setzen wir zur Abkürzung $x_k := x_{\mu_k}$). Wir wissen, dass jeder Häufungspunkt von $\{x_k\}$, und damit auch x^*, eine Lösung von (P) ist und folglich in M_{opt} gehört. Da wir notfalls zu einer Teilfolge übergehen können, kann $\lim_{k\to\infty} x_k = x^*$ angenommen werden. Wir werden zeigen, dass x^* eine Lösung von (P_0 ist, wegen der Eindeutigkeit einer Lösung von (P_0) also mit x_0 übereinstimmt, woraus dann die Behauptung folgt. Hierzu beweisen wir zuerst, dass x^* zulässig für (P_0) ist. Für vorgegebenes $\mu > 0$ und $x \in M_{\mathrm{opt}}$ ist $x_\mu + t(x - x_\mu) \in M_0$ für alle hinreichend kleinen $|t|$. Da x_μ die Lösung von (P_μ) ist, erhält man

$$\begin{aligned} 0 &= \nabla f_\mu(x_\mu)^T (x - x_\mu) \\ &= c^T(x - x_\mu) - \mu \sum_{j=1}^{n} \frac{x_j - (x_\mu)_j}{(x_\mu)_j} \\ &= \underbrace{\min(\mathrm{P}) - c^T x_\mu}_{\leq 0} + \mu n - \mu \sum_{j\in J} \frac{x_j}{(x_\mu)_j} \end{aligned}$$

und daher

$$\sum_{j\in J} \frac{x_j}{(x_\mu)_j} \leq n \qquad \text{für alle } \mu > 0 \text{ und alle } x \in M_{\mathrm{opt}}.$$

Setzt man hier insbesondere $x := x_0$ (es ist $(x_0)_j > 0$ für alle $j \in J$) und $\mu := \mu_k$, so erhält man nach dem Grenzübergang $\mu_k \to \infty$, dass $x_j^* > 0$ für alle $j \in J$, also x^* zulässig für (P_0) ist. Nun zeigen wir, dass x^* Lösung von (P_0) ist, woraus dann $x^* = x_0$ und damit die Behauptung folgt. Wie wir gerade eben gesehen haben, ist

$$\frac{c^T(x - x_\mu)}{\mu} = \sum_{j=1}^{n} \frac{x_j - (x_\mu)_j}{(x_\mu)_j} \qquad \text{für alle } \mu > 0 \text{ und alle } x \in M_{\mathrm{opt}}.$$

Setzt man hier einmal $x := x_0$, einmal $x := x^*$ und subtrahiert die beiden Gleichungen von einander, so erhält man

$$0 = \sum_{j=1}^{n} \frac{(x_0)_j - x_j^*}{(x_\mu)_j} = \sum_{j\in J} \frac{(x_0)_j - x_j^*}{(x_\mu)_j}.$$

Setzt man hier wiederum $\mu := \mu_k$ und macht den Grenzübergang $k \to \infty$, so erhält man

$$\sum_{j\in J} \frac{(x_0)_j}{x_j^*} = \#(J).$$

Hieraus wiederum folgt mit Hilfe der Ungleichung vom geometrisch-arithmetischen Mittel, dass

$$\begin{aligned} f_0(x^*) - f_0(x_0) &= -\sum_{j\in J} \log x_j^* + \sum_{j\in J} \log(x_0)_j \\ &= \log\Big(\prod_{j\in J} \frac{(x_0)_j}{x_j^*}\Big) \\ &\le \log\Big(\underbrace{\frac{1}{\#(J)} \sum_{j\in J} \frac{(x_0)_j}{x_j^*}}_{=1}\Big)^{\#(J)} \\ &= 0. \end{aligned}$$

Damit ist die Behauptung bewiesen.

Die entsprechende Aussage für (D) ist:

- Sei

  $$I := \{i \in \{1,\dots,n\} : (c - A^T y)_i = 0 \text{ für alle } y \in N_{\text{opt}}\}, \qquad J := \{1,\dots,n\} \setminus I.$$

  Dann gilt:

  - Die Optimierungsaufgabe

    $$(\mathrm{D}_0) \qquad \begin{cases} \text{Maximiere} \quad g_0(y) := \displaystyle\sum_{j\in J} \log(c - A^T y)_j \quad \text{unter den NBen} \\ \qquad\qquad y \in N_{\text{opt}}, \qquad (c - A^T y)_j > 0 \quad (j \in J) \end{cases}$$

    besitzt eine eindeutige Lösung y_0.
  - Es ist $\lim_{\mu\to 0+} y_\mu = y_0$.

Der Beweis verläuft völlig analog, wir übergehen ihn daher. □

Im folgenden benutzen wir die folgende Bezeichnung. Ist $u \in \mathbb{R}^p$, so sei $U \in \mathbb{R}^{p\times p}$ diejenige Diagonalmatrix, die die Komponenten von u der Reihe nach in der Diagonalen als Einträge hat. Weiter sei e stets ein Vektor passender Länge, dessen Komponenten alle gleich 1 sind. Also[28] ist $u = Ue$.

[28] Konsequenterweise müsste man die Einheitsmatrix dann allerdings mit E bezeichnen. Wir bleiben bei der (für uns) üblichen Bezeichnung I.

Satz 3.4 *Die Voraussetzungen (A1)–(A3) seien erfüllt. Dann besitzt bei vorgegebenem $\mu > 0$ das nichtlineare "Gleichungssystem"*

$$(*) \qquad F_\mu(x,y,z) := \begin{pmatrix} Xz - \mu e \\ Ax - b \\ A^T y + z - c \end{pmatrix} = 0, \qquad \begin{array}{rcl} x & > & 0, \\ z & > & 0 \end{array}$$

genau eine Lösung $(x_\mu, y_\mu, z_\mu) \in \mathbb{R}^n \times \mathbb{R}^m \times \mathbb{R}^n$. Hierbei ist $x_\mu \in M_0$ die Lösung von (P_μ), $y_\mu \in N_0$ die Lösung von (D_μ) und $c^T x_\mu - b^T y_\mu = \mu n$.

Beweis: Im ersten Teil des Beweises zeigen wir die Existenz einer Lösung von $(*)$. Genauer gilt:

- Sei $x_\mu \in M_0$ die Lösung von (P_μ), $y_\mu \in N_0$ die Lösung von (D_μ) und $z_\mu := c - A^T y_\mu$. Dann ist (x_μ, y_μ, z_μ) eine Lösung von $(*)$.

Denn: Offensichtlich ist $x_\mu > 0$, $z_\mu > 0$, $Ax_\mu - b = 0$ und $A^T y_\mu + z_\mu - c = 0$. Zu zeigen bleibt also nur noch $X_\mu z_\mu - \mu e = 0$ bzw. $c - \mu X_\mu^{-1} e - A^T y_\mu = 0$. Da x_μ Lösung von (P_μ) ist, folgt aus der Lagrangeschen Multiplikatorenregel die Existenz von $\tilde{y}_\mu \in \mathbb{R}^m$ mit $c - \mu X_\mu^{-1} e - A^T \tilde{y}_\mu = 0$. Wir haben zu zeigen, dass $\tilde{y}_\mu = y_\mu$. Offensichtlich ist $\tilde{y}_\mu \in N_0$. Mit $\tilde{z}_\mu := c - A^T \tilde{y}_\mu$ ist ferner

$$\nabla g_\mu(\tilde{y}_\mu) - b - \mu A \tilde{Z}_\mu^{-1} e = b - A X_\mu e = b - A x_\mu = 0.$$

Da g_μ auf N_0 konkav ist, ist $\tilde{y}_\mu$ Lösung von (D_μ). Nun ist aber y_μ die eindeutige Lösung von (D_μ), folglich ist $\tilde{y}_\mu = y_\mu$ und nachgewiesen, dass (x_μ, y_μ, z_μ) Lösung von $(*)$ ist.

Nachdem eben die Existenz eine Lösung von $(*)$ nachgewiesen wurde, folgt jetzt die Eindeutigkeit.

- Sei (x, y, z) eine Lösung von $(*)$. Dann ist $x = x_\mu$ die Lösung von (P_μ), $y = y_\mu$ die Lösung von (D_μ) und $z = c - A^T y_\mu$. Insbesondere ist also eine Lösung von $(*)$ eindeutig bestimmt.

Denn: Wegen $x > 0$ und $Ax = b$ ist $x \in M_0$. Weiter ist

$$0 = c - A^T y - z = c - \mu X^{-1} e - A^T y = \nabla f_\mu(x) - A^T y.$$

Hieraus folgt, dass x Lösung von (P_μ) ist. Denn ist $z \in M_0$ beliebig, so ist

$$\begin{aligned} f_\mu(z) - f_\mu(x) &= c^T(z-x) - \mu \sum_{j=1}^n [\log z_j - \log x_j] \\ &\geq c^T(z-x) - \mu \sum_{j=1}^n \frac{z_j - x_j}{x_j} \\ &\qquad \text{(wegen } \log t \leq t-1 \text{ für } t > 0\text{)} \\ &= (c - \mu X^{-1} e)^T (z-x) \\ &= (A^T y)^T (z-x) \\ &= y^T A (z-x) \\ &= 0. \end{aligned}$$

Wegen $z = c - A^T y > 0$ ist $y \in N_0$. Weiter ist

$$\nabla g_\mu(y) = b - \mu A Z^{-1} e = b - AXe = b - Ax = 0,$$

woraus (ähnlich wie eben) folgt, dass y eine Lösung von (D_μ) ist, also $y = y_\mu$ gilt. Damit ist auch der zweite Beweisschritt abgeschlossen.

Nun kommt der letzte Schritt.

- Ist (x_μ, y_μ, z_μ) die eindeutige Lösung von $(*)$, so ist $c^T x_\mu - b^T y_\mu = \mu n$.

Denn:

$$c^T x_\mu = (A^T y_\mu + z_\mu)^T x_\mu = b^T y_\mu + z_\mu^T x_\mu = b^T y_\mu + \mu e^T e = b^T y_\mu + \mu n.$$

Damit ist der Bewies abgeschlossen. □

Als Korollar zum letzten Satz könnten wir formulieren, dass das nichtlineare Gleichungssystem $(*)$ für jedes $\mu > 0$ eine eindeutige Lösung (x_μ, y_μ, z_μ) besitzt und für $\mu \to 0$ in den ersten beiden Komponenten Konvergenz gegen eine Lösung von (P) bzw. (D) vorliegt. Die Menge

$$\mathcal{C} := \{(x_\mu, y_\mu, z_\mu) : \mu > 0\}$$

heißt der zu (P) und (D) gehörende *zentrale Pfad*.

Bemerkung: Ist $(x^*, y^*) \in M_{\text{opt}} \times N_{\text{opt}}$ ein Paar von Lösungen von (P) bzw. (D) und $z^* := c - A^T y^*$, so ist (x^*, y^*, z^*) offenbar eine Lösung von $F_0(x, y, z) = 0$ mit nichtnegativer erster und dritter Komponente. □

2.3.2 Das primal-duale Innere-Punkt-Verfahren

Als Literatur in diesem Unterabschnitt sei S. J. Wright (1997)[29] empfohlen. Wir benutzen die im vorigen Abschnitt benutzten Bezeichnungen. Seien also weiter (P) und (D) die gegebenen zueinander dualen linearen Programme, die Voraussetzungen (A1)–(A3) werden auch weiterhin wichtig sein. Nach wie vor sei bei gegebem $x \in \mathbb{R}^n$ die Matrix $X \in \mathbb{R}^{n \times n}$ durch $X := \operatorname{diag}(x)$ definiert[30].

Wenn man ein nichtlineares Gleichungssystem lösen will, denkt man zunächst meistens an das Newton-Verfahren. Das gilt insbesondere für das nichtlineare Gleichungssystem $(*)$ in Satz 3.4. Um das Newton-Verfahren anwenden zu können, ist es wichtig, die Nichtsingularität der Funktionalmatrix zu sichern. Dies geschieht im folgenden Lemma.

Lemma 3.5 *Die Voraussetzung (A3) sei erfüllt. Bei vorgegebenem $\mu > 0$ sei die Abbildung $F_\mu\colon \mathbb{R}^n \times \mathbb{R}^m \times \mathbb{R}^n \longrightarrow \mathbb{R}^n \times \mathbb{R}^m \times \mathbb{R}^n$ durch*

$$F_\mu(x, y, z) := \begin{pmatrix} Xz - \mu e \\ Ax - b \\ A^T y + z - c \end{pmatrix}$$

[29] S. J. Wright (1997) *Primal-Dual Interior-Point Methods*. SIAM, Philadelphia.

Auf S. 35 findet man in diesem Buch übrigens auch einen Beweis von Satz 1.11 über die Existenz eines strikt komplementären optimale Paares. Informationen kann man unter der Adresse `http://www.siam.org/books/swright/` erhalten

[30] In Matlab ist `diag` ein Befehl, der genau dieser Beschreibung entspricht.

definiert. Dann ist die (von dem Parameter μ unabhängige) Funktionalmatrix

$$F'_\mu(x,y,z) = \begin{pmatrix} Z & 0 & X \\ A & 0 & 0 \\ 0 & A^T & I \end{pmatrix}$$

in jedem Tripel $(x,y,z) \in \mathbb{R}^n \times \mathbb{R}^m \times \mathbb{R}^n$ mit $x > 0$ und $z > 0$ nichtsingulär.

Beweis: Sei (p,q,r) aus dem Kern von $F'_\mu(x,y,z)$, also

$$Zp + Xr = 0, \qquad Ap = 0, \qquad A^T q + r = 0.$$

Nun ist

$$0 = q^T A(\underbrace{p + Z^{-1}Xr}_{=0}) = q^T \underbrace{Ap}_{=0} - q^T AZ^{-1}XA^T q = -q^T AZ^{-1}XA^T q.$$

Da $Z^{-1}X$ eine posiv definite (Diagonal-)Matrix und wegen (A3) Rang $(A) = m$, ist $AZ^{-1}XA^T$ positiv definit. Daher ist $q = 0$, folglich auch $r = 0$ und $p = 0$. Das einfache Lemma ist bewiesen. □

Ein Schritt des primal-dualen Innere-Punkt-Verfahrens besteht darin, zu einer aktuellen Näherung (x,y,z) (hier ist x Näherung für eine Lösung des primalen Problems, y Näherung für eine Lösung des dualen Problems und z eine Näherung für den optimalen Schlupf) mit $x > 0$ und $z > 0$ (hieran liegt es, dass man von *Innere-Punkt*-Verfahren spricht) ein $\mu > 0$ zu bestimmen, i. Allg. wird

$$\mu := \sigma \frac{x^T z}{n}$$

mit einem $\sigma \in (0,1)$ gesetzt, dann einen Schritt des Newton-Verfahrens[31] zur Lösung von $F_\mu = 0$ durchzuführen und diesen Schritt so zu dämpfen, dass auch für die neue Näherung (x^+, y^+, z^+) die Positivitätsbedingungen $x^+ > 0$ und $z^+ > 0$ erhalten bleiben. Wir gehen von einem *unzulässigen* primal-dualen Innere-Punkt-Verfahren aus und geben danach erst in einer Bemerkung die Vereinfachungen an, die sich bei einem *zulässigen* primal-dualen Innere-Punkt-Verfahren ergeben.

- Gegeben sei ein Tripel $(x,y,z) \in \mathbb{R}^n \times \mathbb{R}^m \times \mathbb{R}^n$ mit $x > 0$ und $z > 0$ sowie ein $\mu > 0$, etwa $\mu := \sigma x^T z/n$ mit $\sigma \in (0,1)$.
- Berechne die Newton-Richtung (p,q,r) durch

$$\begin{pmatrix} p \\ q \\ r \end{pmatrix} := -F'_\mu(x,y,z)^{-1} F_\mu(x,y,z)$$

  bzw. als Lösung von

$$\begin{pmatrix} Z & 0 & X \\ A & 0 & 0 \\ 0 & A^T & I \end{pmatrix} \begin{pmatrix} p \\ q \\ r \end{pmatrix} = - \begin{pmatrix} Xz - \mu e \\ Ax - b \\ A^T y + z - c \end{pmatrix}.$$

[31] Die Vorgehensweise bei den sogenannten Prädiktor-Korrektor Verfahren ist ein wenig anders. Wir werden hierauf später noch eingehen.

- Mit Schrittweiten $\alpha_p > 0$ und $\alpha_d > 0$ berechne man das neue Tripel (x^+, y^+, z^+) durch
$$x^+ := x + \alpha_p p, \qquad \begin{pmatrix} y^+ \\ z^+ \end{pmatrix} := \begin{pmatrix} y \\ z \end{pmatrix} + \alpha_d \begin{pmatrix} q \\ r \end{pmatrix}.$$
Hierbei sind $\alpha_p > 0$ und $\alpha_d > 0$ zumindestens so zu wählen, dass $x^+ > 0$ und $z^+ > 0$. Häufig wird man im Prinzip folgendermaßen vorgehen. Man wähle $\tau_p, \tau_d \in (0,1)$ und bestimme
$$\alpha_{p,\max} := \sup\{\alpha > 0 : x + \alpha p > 0\}, \qquad \alpha_{d,\max} := \sup\{\alpha > 0 : z + \alpha r > 0\}$$
und setze anschließend $\alpha_p := \min(1, \tau_p \alpha_{p,\max})$ sowie $\alpha_d := \min(1, \tau_d \alpha_{d,\max})$. Offenbar ist
$$\alpha_{p,\max} = \min_{j: p_j < 0} \Big(-\frac{x_j}{p_j}\Big), \qquad \alpha_{d,\max} = \min_{j: r_j < 0} \Big(-\frac{z_j}{r_j}\Big).$$

Hierdurch ist auch ein Schritt des primal-dualen Verfahrens gegeben. Die Hauptarbeit besteht natürlich in der Berechnung der Newton-Richtung (p, q, r), welche aus den Gleichungen

$$Zp + Xr = -(Xz - \mu e), \qquad Ap = -(Ax - b), \qquad A^T q + r = -(A^T y + z - c)$$

zu erhalten ist. Dies erlaubt die sukzessive Berechnung von q, danach von r und schließlich von p. Zur Abkürzung definiere man die positive Diagonalmatrix $D := Z^{-1}X$. Dann ist

$$\begin{aligned} ADA^T q &= -AD(A^T y + z - c) - ADr \\ &= -AD(A^T y + z - c) - AZ^{-1}Xr \\ &= -AD(A^T y + z - c) - AZ^{-1}[-Zp - (Xz - \mu e)] \\ &= -AD(A^T y + z - c) - (Ax - b) + ADX^{-1}(Xz - \mu e) \\ &= -[Ax - b + AD((A^T y + z - c) - X^{-1}(Xz - \mu e))] \end{aligned}$$

Also ist q aus dem linearen Gleichungssystem

$$(*) \qquad ADA^T q = -[Ax - b + AD((A^T y + z - c) - X^{-1}(Xz - \mu e))]$$

zu berechnen. Anschließend berechnet man p und r durch

$$\begin{aligned} r &= -A^T q - (A^T y + z - c) \\ p &= -Dr - Z^{-1}(Xz - \mu e). \end{aligned}$$

Unter der Rangvoraussetzung (A3) an A ist die Koeffizientenmatrix ADA^T des linearen Gleichungssystems $(*)$ symmetrisch und positiv definit, was natürlich bei einer effizienten Lösung zu beachten ist. Leider folgt aus einer Dünnbesetztheit von A i. Allg. nicht die von ADA^T. Denn ist etwa $A = \begin{pmatrix} a_1 & \cdots & a_n \end{pmatrix}$ und $D = \operatorname{diag}(d_1, \ldots, d_n)$, so ist

$$ADA^T = \sum_{j=1}^{n} d_j a_j a_j^T.$$

Wenn A also nur eine voll besetzte Spalte hat, so ist ADA^T voll besetzt. Allerdings wird meistens nur ein kleiner Teil der Spalten dicht besetzt sein, was ausgenutzt werden kann, wie man in dem Report von Y. Zhang, siehe `http://www.caam.rice.edu/~zhang/lipsol/`, nachlesen kann. Die Güte einer aktuellen Näherung wird bei Zhang gemessen durch den Defekt

$$d(x,y,z) := \frac{\|Ax-b\|}{\max(1,\|b\|)} + \frac{\|A^Ty+z-c\|}{\max(1,\|c\|)} + \frac{|c^Tx-b^Ty|}{\max(1,|c^Tx|,|b^Ty|)}$$

und die Rechnung abgebrochen, wenn dieser hinreichend klein (etwa 10^{-8}) ist. Man beachte, dass $A(x+\alpha_p p)-b = (1-\alpha_p)(Ax-b)$. Da i. Allg. $\alpha_p \in (0,1]$, wird der Defekt im Gleichungssystem sich echt verkleinern. Wegen

$$A^T(y+\alpha_d q)+(z+\alpha_d r)-c = (1-\alpha_d)(A^Ty+z-c)$$

gilt das entsprechende auch für die letzte zu lösende Gleichung. Weiter ist (hier gehen wir davon aus, dass die primale und die duale Schrittweite gleich sind)

$$\begin{aligned}(x+\alpha p)^T(z+\alpha r) &= x^Tz+\alpha(p^Tz+x^Tr)+\alpha^2p^Tr\\ &= x^Tz+\alpha e^T(Zp+Xr)+\alpha^2p^Tr\\ &= x^Tz-\alpha e^T(Xz-\mu e)+\alpha^2p^Tr\\ &= x^Tz-\alpha(x^Tz-\mu n)+\alpha^2p^Tr\\ &= [1-\alpha(1-\sigma)]x^Tz+\alpha^2p^Tr,\end{aligned}$$

wobei $\mu = \sigma x^Tz/n$ angenommen wird. Wir versuchen, den letzten Term noch weiter abzuschätzen und benutzen hierbei weiter die Abkürzung $D := Z^{-1}X$, außerdem die einfach zu beweisende Ungleichung $u^Tv \le \frac{1}{4}\|u+v\|^2$ (hier ist $\|\cdot\|$ natürlich die euklidische Norm). Dann ist

$$\begin{aligned}p^Tr &= (D^{-1/2}p)^T(D^{1/2}r)\\ &\le \frac{1}{4}\|D^{-1/2}p+D^{1/2}r\|^2\\ &= \frac{1}{4}\|X^{-1/2}Z^{-1/2}(XZ-\mu I)e\|^2\\ &= \frac{1}{4}\|(X^{1/2}Z^{1/2}e-\mu X^{-1/2}Z^{-1/2}e\|^2\\ &= \frac{1}{4}[x^Tz-2\mu n+\mu^2e^T(XZ)^{-1}e]\\ &= \frac{1}{4}\Big[(1-2\sigma)+\frac{\sigma^2}{n^2}x^Tz\,e^T(XZ)^{-1}e)\Big]x^Tz\\ &\le \frac{1}{4}\Big[(1-2\sigma)+\sigma^2\frac{x^Tz/n}{\min(XZe)}\Big]x^Tz.\end{aligned}$$

Hierbei ist für einen Vektor $u \in \mathbb{R}^n$ natürlich $\min(u) = \min_{j=1,\dots,n} u_j$.

Wir haben nicht den Ehrgeiz, in die Nähe von Konvergenzresultaten für unzulässige primal-duale Innere-Punkt-Verfahren zu gelangen und begnügen uns daher mit diesen Aussagen.

Bemerkung: Ein Schritt eines *zulässigen* primal-dualen Innere-Punkt Verfahrens sieht im Prinzip folgendermaßen aus.

- Gegeben sei ein Paar $(x, y) \in M_0 \times N_0$, setze $z := c - A^T y$, $D := Z^{-1}X$ und setze $\mu := \sigma x^T z/n$ mit $\sigma \in (0, 1)$.
- Berechne q als Lösung von
$$ADA^T q = X^{-1}(Xz - \mu e),$$
anschließend p durch
$$p = DA^T[q - X^{-1}(Xz - \mu e)].$$
- Berechne Schrittweite[32] $\alpha > 0$ mit $x + \alpha p > 0$ (wegen $Ap = 0$ ist dann auch $x + \alpha p \in M_0$) und $z - \alpha A^T q > 0$ (dann ist auch $y + \alpha q \in N_0$). Etwa wähle man $\tau \in (0, 1)$ und setze $\alpha := \min(1, \tau\alpha_{\max})$, wobei
$$\alpha_{\max} := \min\Big[\min_{j:p_j<0}\Big(-\frac{x_j}{p_j}\Big), \min_{j:(A^Tq)_j>0}\Big(\frac{z_j}{(A^Tq)_j}\Big)\Big].$$
- Setze $(x^+, y^+) := (x, y) + \alpha(p, q)$.

Wegen der Zulässigkeit von x und y für (P) bzw. (D) wird die Güte der Näherung (x, y) im wesentlichen durch die Dualitätslücke $c^T x - b^T y = x^T z$ gemessen. Daher ist es interessant, wie sich die Dualitätslücke von Schritt zu Schritt verändert. Es ist

$$\begin{aligned} c^T(x + \alpha p) - b^T(y + \alpha q) &= c^T x - b^T y + \alpha(c - A^T y)^T p - \alpha(Ax)^T q \\ &\qquad \text{(wegen } Ap = 0 \text{ und } Ax = b\text{)} \\ &= c^T x - b^T y + \alpha e^T(Zp - XA^T q) \\ &= c^T x - b^T y + \alpha e^T(\mu e - XZe) \\ &= c^T x - b^T y + \alpha\mu n - \alpha x^T z \\ &= [1 - \alpha(1 - \sigma)](c^T x - b^T y), \end{aligned}$$

wobei wir davon ausgehen, dass $\mu = \sigma x^T z/n$ mit einem $\sigma \in (0, 1)$. Daher verkleinert sich die Dualitätslücke von Schritt zu Schritt. Eigentlich hängen die Dualitätslücke, die Schrittweite und auch σ von dem Iterationsindex ab. Es ist also

$$\frac{x_{k+1}^T z_{k+1}}{x_k^T z_k} = 1 - (1 - \sigma_k)\alpha_k.$$

Wird also die Dualitätslücke gleichmäßig verkleinert, existiert also ein $c_0 \in (0, 1)$ mit $1-(1-\sigma_k)\alpha_k \le c_0$, so konvergiert die Dualitätslücke Q-linear gegen Null. Weite konvergiert die Folge der Dualitätslücken sogar superlinear gegen Null, wenn[33] z. B. $\sigma_k \to 0$ und $\alpha_k \to 1$. Nun wollen wir noch einige wenige Aussagen zur Konvergenz eines *zulässigen* primal-dualen Verfahrens machen. Hierbei wird die Gültigkeit der Voraussetzungen (A1)–(A3) vorausgesetzt.

[32] In der Praxis wird man mit einer primalen *und* einer dualen Schrittweite arbeiten

[33] Numerische Experimente zeigen, dass es in der Tat keine gute Idee ist, die σ_k konstant zu halten. Das gilt auch für das unzulässige Verfahren.

- Die Folgen $\{x_k\}$ und $\{z_k\}$ sind beschränkt.

Denn: Es ist $(x_k - x_0)^T(z_k - z_0) = 0$, da $x_k - x_0 \in \text{Kern}\,(A)$ und $z_k - z_0 \in \text{Bild}\,(A^T)$. Für alle k ist daher

$$0 = (x_k - x_0)^T(z_k - z_0) = x_k^T z_k + x_0^T z_0 - [x_0^T z_k + x_k^T z_0]$$

und folglich

$$x_0^T z_k + x_k^T z_0 = x_k^T z_k + x_0^T z_0 \leq 2x_0^T z_0.$$

Da x_0 und z_0 sowie x_k und z_k positive Vektoren sind, folgt die Beschränktheit der Folgen $\{x_k\}$ und $\{z_k\}$.

- Sei $\lim_{k\to\infty} x_k^T z_k = 0$. Ist dann (x^*, z^*) ein Häufungspunkt der beschränkten Folge $\{(x_k, z_k)\}$, so ist $x^* \in M$ eine Lösung von (P) und es existiert eine Lösung $y^* \in N$ von (D) mit $z^* = c - A^T y^*$.

Denn: Für den Häufungspunkt (x^*, z^*) von $\{(x_k, z_k)\}$ ist natürlich $x^* \in M$, also x^* zulässig für (P), weiter $z^* \geq 0$. Wegen $z_k \in c - \text{Bild}\,(A^T)$ existiert ein $y^* \in \mathbb{R}^m$ mit $z^* = c - A^T y^*$, folglich ist $y^* \in N$ dual zulässig. Da $\lim_{k\to\infty} x_k^T z_k = 0$ vorausgesetzt wurde, ist $(x^*)^T z^* = 0$ bzw. $c^T x^* = b^T y^*$. Aus dem schwachen Dualitätssatz folgt die Optimalität von x^* bzw. y^*.

- Sei

$$\mathcal{L} := \{(x^*, z^*) : x^* \in M \text{ löst (P)}, z^* = c - A^T y^* \text{ mit Lösung } y^* \in N \text{ von (D)}\}.$$

Ist dann $\lim_{k\to\infty} x_k^T z_k = 0$, so konvergiert $\{(x_k, z_k)\}$ gegen $\mathcal{L}$, d. h. es ist

$$\lim_{k\to\infty} \min_{(x^*,z^*)\in\mathcal{L}} \|(x_k, z_k) - (x^*, z^*)\| = 0.$$

Wegen des Projektionssatzes können wir hierbei mit gutem Gewissen min statt inf schreiben, da die Lösungsmenge $\mathcal{L}$ natürlich konvex und abgeschlossen ist.

Denn: Der wohl einfachste Beweis benutzt ein auf A. J. Hoffman (1952)[34] zurückgehendes Ergebnis, das wir ohne Beweis angeben.

Gegeben sei ein nichtleerer Polyeder $P := \{z \in \mathbb{R}^n : Az \leq b\}$ *mit* $A \in \mathbb{R}^{m\times n}$ *und* $b \in \mathbb{R}^m$. *Dann existiert eine positive Konstante* c *derart, dass*

$$d(x, P) := \min_{z\in P} \|x - z\| \leq c\,\|(Ax - b)_+\| \qquad \textit{für alle } x \in \mathbb{R}^n.$$

Hierbei ist $y_+ := (\max(y_i, 0))$.

[34] A. J. Hoffman (1952) "On approximate solutions of systems of linear inequalities". Journal of Research of the National Bureau of Standards 49, 263–265.

Die Idee besteht jetzt darin, die Lösungsmenge $\mathcal{L}$ als einen Polyeder darzustellen. Dies kann auf verschiedene Weise geschehen. Wir beachten, dass $\mathcal{L}$ wegen des starken Dualitätssatzes der linearen Optimierung als Polyeder und zwar als Menge aller $(x,z) \in \mathbb{R}^n \times \mathbb{R}^n$ mit

$$Ax = b, \quad Bz = Bc, \quad x \geq 0, \quad z \geq 0, \quad c^T x - b^T (AA^T)^{-1} A(c-z) \leq 0.$$

Hierbei ist $B \in \mathbb{R}^{(n-m)\times n}$ eine Matrix mit der Eigenschaft, dass die Spalten von B^T eine Basis des Kerns von A bilden. Dann ist $Bz = Bc$ äquivalent zu

$$c - z \in \operatorname{Kern}(B) = \operatorname{Bild}(B^T)^\perp = \operatorname{Kern}(A)^\perp = \operatorname{Bild}(A^T).$$

Benutzt man nur $\leq$-Ungleichungen bei der Beschreibung von $\mathcal{L}$, so ist also

$$\mathcal{L} = \left\{ (x,z) \in \mathbb{R}^n \times \mathbb{R}^n : \begin{pmatrix} A & 0 \\ -A & 0 \\ 0 & B \\ 0 & -B \\ -I & 0 \\ 0 & -I \\ c^T & b^T(AA^T)^{-1}A \end{pmatrix} \begin{pmatrix} x \\ z \end{pmatrix} \leq \begin{pmatrix} b \\ -b \\ Bc \\ -Bc \\ 0 \\ 0 \\ b^T(AA^T)^{-1}Ac \end{pmatrix} \right\}.$$

Nun beachten wir, dass

$$\left(\begin{pmatrix} A & 0 \\ -A & 0 \\ 0 & B \\ 0 & -B \\ -I & 0 \\ 0 & -I \\ c^T & b^T(AA^T)^{-1}A \end{pmatrix} \begin{pmatrix} x_k \\ z_k \end{pmatrix} - \begin{pmatrix} b \\ -b \\ Bc \\ -Bc \\ 0 \\ 0 \\ b^T(AA^T)^{-1}Ac \end{pmatrix} \right)_+ = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ x_k^T z_k \end{pmatrix}.$$

Aus dem Hoffman-Theorem folgt die Existenz einer positiven Konstanten c mit

$$\min_{(x^*,z^*)\in\mathcal{L}} \|(x_k,z_k) - (x^*,z^*)\| \leq c\, x_k^T z_k,$$

woraus dann wegen $\lim_{k\to\infty} x_k^T z_k = 0$ die Behauptung folgt.

- Sei $\mathcal{L}_0 := \{(x^*,z^*) \in \mathcal{L} : x^* + z^* > 0\}$. Wegen Satz 1.11 ist $\mathcal{L}_0 \neq \emptyset$. Für ein beliebiges Paar $(x^*,z^*) \in \mathcal{L}_0$ sei

  $$J_x^+ := \{j \in \{1,\ldots,n\} : x_j^* > 0\}, \qquad J_z^+ := \{j \in \{1,\ldots,n\} : z_j^* > 0\}.$$

  Dann gilt:

  – Die Indexmengen J_x^+ und J_z^+ sind von der Wahl von $(x^*,z^*) \in \mathcal{L}_0$ unabhängig.

Ist ferner die Folge $\{\eta_k\}$, wobei

$$\eta_k := \frac{x_k^T z_k / n}{\min(X_k Z_k e)},$$

(nach oben) beschränkt, so gilt:

- Es ist $\liminf_{k\to\infty}(x_k)_j > 0$ für $j \in J_x^+$ und entsprechend $\liminf_{k\to\infty}(z_k)_j > 0$ für $j \in J_z^+$.
- Ist $\lim_{k\to\infty} x_k^T z_k = 0$ und (x^∞, z^∞) ein Häufungspunkt der (beschränkten) Folge $\{(x_k, z_k)\}$, so ist $(x^\infty, z^\infty) \in \mathcal{L}_0$.

Denn: Seien $(x^*, z^*), (x^{**}, z^{**}) \in \mathcal{L}_0$ beliebig. Ist $j \in J_x^+$, also $x_j^* > 0$, so ist $z_j^{**} = 0$ wegen $(x^*)^T z^{**} = 0$. Da aber (x^{**}, z^{**}) ein strikt komplementäres Paar ist, ist $x_j^{**} > 0$. Hieraus, und einem entsprechenden Argument für J_z^+ folgt die Behauptung.

Nun sei die Folge $\{\eta_k\}$ beschränkt. Sei $(x^*, z^*) \in \mathcal{L}_0$ beliebig. Wegen

$$(x_k - x^*)^T (z_k - z^*) = 0$$

(wieder wegen $x_k - x^* \in \text{Kern}\,(A)$ und $z_k - z^* \in \text{Bild}\,(A^T)$) und $(x^*)^T z^* = 0$ ist

$$(x^*)^T z_k + (z^*)^T x_k = x_k^T z_k.$$

Wegen der vorausgesetzten Beschränktheit der Folge $\{\eta_k\}$ existiert eine positive Konstante c_0 mit $1/\eta_k \geq c_0$ für alle k. Sei $j \in J_x^+$ beliebig. Dann ist

$$(x_k)_j (z_k)_j \geq \min(X_k Z_k e) = \frac{1}{\eta_k} \frac{x_k^T z_k}{n} \geq c_0 \frac{x_k^T z_k}{n}$$

und folglich

$$(x_k)_j \geq \frac{c_0}{n} \frac{x_k^T z_k}{(z_k)_j} = \frac{c_0}{n} \frac{(x^*)^T z_k + (z^*)^T x_k}{(z_k)_j} \geq \frac{c_0}{n} \frac{(x^*)_j (z_k)_j}{(z_k)_j} = \frac{c_0}{n} (x^*)_j.$$

Daher ist $\liminf_{k\to\infty}(x_k)_j \geq (c_0/n) x_j^* > 0$, $j \in J_x^+$. Die entsprechende Aussage kann entsprechend bewiesen werden.

Ist $\lim_{k\to\infty} x_k^T z_k = 0$ und (x^∞, z^∞) ein Häufungspunkt von $\{(x_k, z_k)\}$, so ist natürlich $(x^\infty)^T z^\infty = 0$ und daher $(x^\infty, z^\infty) \in \mathcal{L}$. Wegen der gerade eben bewiesenen Aussagen ist $(x^\infty)_j > 0$ für $j \in J_x^+$ und $(z^\infty)_j > 0$ für $j \in J_z^+$, womit auch $(x^\infty, z^\infty) \in \mathcal{L}_0$ bewiesen ist. □

Die zulässigen primal-dualen Innere-Punkt-Verfahren sind von eher historischer Bedeutung, was dann auch für die eben gemachten Aussagen gilt[35] Die *unzulässigen* primal-dualen Innere-Punkt-Verfahren haben sich in der Praxis durchgesetzt. Andererseits sind Konvergenzanalysen nach wie vor eher unerfreulich. Einen kleinen Eindruck hiervon erhält man in Chapter 6 bei S. J. Wright (1997).

[35] Siehe

R. A. Tapia, Y. Zhang, Y. Ye (1995) "On the convergence of the iteration sequence in primal-dual interior-point methods". Mathematical Programming 68, 141–154.

Zum Schluss wollen wir noch die Prädiktor-Korrektor Version des primal-dualen Innere-Punkt-Verfahrens angeben, welche von Y. Zhang als das bei weitem beste Verfahren angegeben wird. Wieder geben wir einen Schritt des Verfahrens an, wobei wir für Feinheiten auf den Report von Zhang verweisen. Hinweisen wollen wir auch auf I. J. Lustig, R. E. Marsten, D. F. Shanno (1992)[36]

- Gegeben sei ein Tripel $(x, y, z) \in \mathbb{R}^n \times \mathbb{R}^m \times \mathbb{R}^n$ mit $x > 0$ und $z > 0$.
- Berechne eine Prädiktor-Richtung $(\hat{p}, \hat{q}, \hat{r})$ durch

$$\begin{pmatrix} \hat{p} \\ \hat{q} \\ \hat{r} \end{pmatrix} := -F_0'(x, y, z)^{-1} F_0(x, y, z)$$

  bzw. als Lösung von

$$\begin{pmatrix} Z & 0 & X \\ A & 0 & 0 \\ 0 & A^T & I \end{pmatrix} \begin{pmatrix} \hat{p} \\ \hat{q} \\ \hat{r} \end{pmatrix} = - \begin{pmatrix} Xz \\ Ax - b \\ A^T y + z - c \end{pmatrix}.$$

- Berechne

$$\begin{pmatrix} \hat{x} \\ \hat{y} \\ \hat{z} \end{pmatrix} := \begin{pmatrix} x \\ y \\ z \end{pmatrix} + \begin{pmatrix} \hat{p} \\ \hat{q} \\ \hat{r} \end{pmatrix}.$$

  Eventuell wird dieser Schritt noch durch eine Schrittweite $\hat{\alpha} > 0$ gedämpft.
- Berechne mit einem geeigneten $\mu > 0$ die Korrektor-Richtung (p, q, r) durch

$$\begin{pmatrix} p \\ q \\ r \end{pmatrix} := -F_0'(x, y, z)^{-1} F_\mu(\hat{x}, \hat{y}, \hat{z})$$

  bzw. als Lösung von

$$\begin{pmatrix} Z & 0 & X \\ A & 0 & 0 \\ 0 & A^T & I \end{pmatrix} \begin{pmatrix} p \\ q \\ r \end{pmatrix} = - \begin{pmatrix} \hat{X}\hat{z} - \mu e \\ A\hat{x} - b \\ A^T \hat{y} + \hat{z} - c \end{pmatrix}.$$

- Mit einer geeigneten Schrittweite[37] $\alpha > 0$ berechne man die neue Näherung

$$\begin{pmatrix} x^+ \\ y^+ \\ z^+ \end{pmatrix} := \begin{pmatrix} x \\ y \\ z \end{pmatrix} + \alpha \begin{pmatrix} \hat{p} + p \\ \hat{q} + q \\ \hat{r} + r \end{pmatrix}.$$

Hier werden also in jedem Schritt *zwei* lineare Gleichungssysteme gelöst, die aber die selbe Koeffizientenmatrix besitzen.

[36] I. J. Lustig, R. E. Marsten, D. F. Shanno (1992) "On implementing Mehrotra's predictor-corrector interior-point method for linear programming". SIAM J. Optimization 2, 435–449.

[37] Auch hier wird man in der Praxis eine primale und eine duale Schrittweite wählen.

2.3.3 Aufgaben

1. Seien $x, z \in \mathbb{R}^n$ gegeben und $X := \operatorname{diag}(x)$, $\|\cdot\|$ bezeichne die euklidische Norm. Dann ist
$$\left\| Xz - \frac{x^T z}{n} e \right\| = \min_{\lambda \in \mathbb{R}} \|Xz - \lambda e\|,$$
d. h. $(x^T z/n)e$ ist die orthogonale Projektion von Xz auf $\operatorname{span}\{e\}$.

2. Gegeben sei ein lineares Programm in Karmarkar-Standardform, also
$$\text{(P)} \qquad \text{Minimiere} \quad c^T x \quad \text{auf} \quad M := \left\{ x \in \mathbb{R}^n : x \geq 0, \begin{pmatrix} A \\ e^T \end{pmatrix} = \begin{pmatrix} 0 \\ n \end{pmatrix} \right\}.$$
Hierbei seien $A \in \mathbb{R}^{m \times n}$ und $c \in \mathbb{R}^n$ mit $n \geq 2$ gegeben, weiter sei $e := (1, \ldots, 1)^T \in \mathbb{R}^n$. Es werde vorausgesetzt:

   (V) Es ist $Ae = 0$ und $\operatorname{Rang}(A) = m$.

   Man zeige, dass die Voraussetzungen (A1), (A2), (A3) aus Unterabschnitt 2.3.1 erfüllt sind.

3. Man betrachte[38] ein lineares Programm der Form
$$\text{Minimiere} \quad c^T x \quad \text{unter den Nebenbedingungen} \quad Ax \leq b, \quad Cx = d.$$
Man führe dieses Programm in Standardform über. Hat das zu diesem Programm in Standardform duale Programm strikt zulässige Lösungen?

4. Gegeben sei die Nullstellenaufgabe $f(x) = 0$ mit einer in einer Umgebung U^* von $x^* \in \mathbb{R}$ vier mal stetig differenzierbaren reellwertigen Funktion f. Sei $f(x^*) = 0$ und $f'(x^*) \neq 0$. Man zeige, dass das *zusammengesetzte Newton-Verfahren*
$$x_{k+1} := x_k - \frac{f(x_k) + f(x_k - f(x_k)/f'(x_k))}{f'(x_k)}$$
lokal von mindestens dritter Ordnung konvergiert. An hand von $f(x) := x - \cos^2 x$ mit den Startwerten $x_0 := 1$, $x_0 := 2$, $x_0 := 5$ vergleiche man die durch das Newton- und das zusammengesetzte Newton-Verfahren erhaltenen Werte. Weshalb wird diese Aufgabe in dem Abschnitt über Innere-Punkt-Verfahren gestellt?

   Hinweis: Man wende einen aus der Numerischen Mathematik her bekannten Satz über die Konvergenzgeschwindigkeit von Iterationsvererfahren bei nichtlinearen Gleichungen an, siehe z. B. J. Werner (1992, S. 98)[39]. Zum Nachweis der Voraussetzungen dieses Satzes kann man *Mathematica* benutzen.

5. Man programmiere einen Schritt des unzulässigen primal dualen Verfahrens für ein Problem in Standardform mit den Daten (A, b, c). Anschließend wende man das Verfahren auf ein Beispiel mit den Daten

c^T	
A	b

$:=$

5	3	3	6	0	0	0	
−6	1	2	4	1	0	0	14
3	−2	−1	−5	0	1	0	−25
−2	1	0	2	0	0	1	14

[38] Siehe S. J. Wright (1997, S. 124).
[39] J. Werner (1992) *Numerische Mathematik 1*. Vieweg, Braunschweig-Wiesbaden.

an, wobei man mit $(x, y, z) := (e, 0, e)$ starte. Insbesondere beobachte man, wie sich der Defekt

$$d(x,y,z) := \frac{\|Ax-b\|}{\max(1,\|b\|)} + \frac{\|A^T y + z - c\|}{\max(1,\|c\|)} + \frac{|c^T x - b^T y|}{\max(1,|c^T x|,|b^T y|)}$$

verändert. Man sollte verschiedene Strategien bei der Wahl von σ ausprobieren, etwa $\sigma_k := 0.5$, $\sigma_k := 1/(k+1)$ und $\sigma_k := 1/(k+1)^2$.

6. Man wende das primal-duale Innere-Punkt-Verfahren auf die linearen Programme in Standardform an, wie sie durch Aufgabe 4 in Unterabschnitt 2.2.7 gegeben sind.

7. Gegeben seien das lineare Programm

   (P) $\qquad$ Minimiere $\quad c^T x \quad$ auf $\quad M := \{x \in \mathbb{R}^n : x \geq 0,\ Ax = b\}$

   und das hierzu duale lineare Programm

   (D) $\qquad$ Maximiere $\quad b^T y \quad$ auf $\quad N := \{y \in \mathbb{R}^m : A^T y \leq c\},$

   wobei $A = (\ a_1\ \cdots\ a_n\ ) \in \mathbb{R}^{m\times n}$ eine Matrix mit Rang $(A) = m$ ist, $b \in \mathbb{R}^m$ und $c \in \mathbb{R}^n$. Sei $x^* \in M$ eine *nichtentartete, zulässige Basislösung* von $Ax = b$, die Indexmenge $B^* := \{j \in \{1,\ldots,n\} : x_j^* > 0\}$ besitze also genau m Elemente und die Spalten $\{a_j\}_{j\in B^*}$ sind linear unabhängig. Ferner existiere ein $y^* \in \mathbb{R}^m$ mit $(A^T y^*)_j = c_j$ für alle $j \in B^*$ und $(A^T y^*)_j < c_j$ für alle $j \notin B^*$. Man zeige:

   (a) Die linearen Programme (P) bzw. (D) besitzen $x^* \in M$ bzw. $y^* \in N$ als eindeutige Lösungen.

   (b) Es ist

   $$M_0 := \{x \in \mathbb{R}^n : x > 0,\ Ax = b\} \neq \emptyset, \qquad N_0 := \{y \in \mathbb{R}^m : A^T y < c\} \neq \emptyset.$$

   (c) Für jedes $\mu > 0$ besitzen die Programme

   (P_μ) $\qquad$ Minimiere $\quad f_\mu(x) := c^T x - \mu \sum_{j=1}^{n} \log x_j, \quad x \in M_0$

   und

   (D_μ) $\qquad$ Maximiere $\quad g_\mu(y) := b^T y + \mu \sum_{j=1}^{n} \log(c - A^T y)_j, \quad y \in N_0$

   eindeutige Lösungen $x_\mu \in M_0$ bzw. $y_\mu \in N_0$. Ferner gilt

   $$c - \mu\, X_\mu^{-1} e = A^T y_\mu \qquad \text{mit} \quad X_\mu := \operatorname{diag}(x_\mu)$$

   und

   $$\lim_{\mu\to 0+} x_\mu = x^*, \qquad \lim_{\mu\to 0+} y_\mu = y^*.$$

   (d) Man definiere $z^* := c - A^T y^*$ und anschließend

   $$X^* := \operatorname{diag}(x^*), \qquad Z^* := \operatorname{diag}(z^*).$$

   Für $\mu > 0$ seien $x_\mu \in M_0$ bzw. $y_\mu \in N_0$ die eindeutigen Lösungen von (P_μ) bzw. (D_μ), ferner sei $z_\mu := c - A^T y_\mu$. Man zeige:

i. Die Matrix

$$\begin{pmatrix} Z^* & 0 & X^* \\ A & 0 & 0 \\ 0 & A^T & I \end{pmatrix}$$

ist nichtsingulär.

ii. Definiert man $(\dot{x}_0, \dot{y}_0, \dot{z}_0) \in \mathbb{R}^n \times \mathbb{R}^m \times \mathbb{R}^n$ durch

$$\begin{pmatrix} Z^* & 0 & X^* \\ A & 0 & 0 \\ 0 & A^T & I \end{pmatrix} \begin{pmatrix} \dot{x}_0 \\ \dot{y}_0 \\ \dot{z}_0 \end{pmatrix} = \begin{pmatrix} e \\ 0 \\ 0 \end{pmatrix}$$

(hierbei ist $e \in \mathbb{R}^n$ einmal wieder der Vektor, dessen Komponenten alle gleich Eins sind), so gilt

$$\lim_{\mu \to 0+} \frac{(x_\mu, y_\mu, z_\mu)) - (x^*, y^*, z^*)}{\mu} = (\dot{x}_0, \dot{y}_0, \dot{z}_0).$$

Kapitel 3

Graphentheorie

Wir wollen in diesem Kapitel eine relativ kurze Einführung in die Graphentheorie geben, da deren Begriffe und grundlegenden Aussagen auch im Operations Research von Interesse sind.

3.1 Grundlegende Begriffe

Ein[1] *Graph* $G = (V, E)$ besteht aus einer endlichen Menge $V = V(G)$, der Menge der *Ecken*, (hier steht V für **V**ertices) und einer Teilmenge $E = E(G)$ von (ungeordneten) Paaren aus V, der Menge der *Kanten* (wobei hier E für **E**dges steht, und nicht etwa für Ecken!). Die Anzahl der Ecken in einem Graphen heißt die *Ordnung* des Graphen. Ein Graph ist also dadurch gegeben, dass auf einer Menge von Ecken eine binäre Relation gegeben ist, welche gerade aussagt, ob zwei Ecken durch eine Kante verbunden ist. Hierdurch sind z. B. Schlingen (Verbindung einer Ecke mit sich selbst) und Mehrfachkanten ausgeschlossen[2]. Wir sprechen von einem *Multigraphen*, wenn Schlingen und Mehrfachkanten vorkommen können. Der Deutlichkeit halber nennt man gelegentlich einen Graphen (ohne Schlingen und Mehrfachkanten) auch einen *einfachen* Graphen.

Beispiele: Sei

$$V = \{1, 2, 3, 4, 5\}, \qquad E = \{\{1, 2\}, \{1, 3\}, \{2, 3\}, \{2, 4\}, \{2, 5\}, \{3, 5\}\}.$$

Wir zeichnen G in Abbildung 3.1 links. Wie wir schon sagten, kann jede Menge V, auf der eine binäre Relation erklärt ist, als ein Graph interpretiert werden. Sei z. B.

[1] Als Literatur nennen wir M. Aigner (1996) *Diskrete Mathematik*. 2., durchgesehene Auflage. Friedrich Vieweg & Sohn, Braunschweig-Wiesbaden.

M. Aigner (1984) *Graphentheorie. Eine Entwicklung aus dem 4-Farben Problem*. Teubner, Stuttgart.

B. Bollababás (1998) *Modern Graph Theory*. Springer, New York-Berlin-Heidelberg.

R. Diestel (1997) *Graph Theory*. Springer, New York-Berlin-Heidelberg.

J. M. Aldous, R. J. Wilson (2000) *Graph Theory and Applications. An Introductory Approach*. Springer, London-Berlin-Heidelberg-New York.

[2] Die Definition eines Graphen in der Literatur ist keineswegs einheitlich. Z. B. sind bei M. Aigner (1984) Mehrfachkanten in einem Graphen zugelassen, bei M. Aigner (1996) ist dies nicht der Fall.

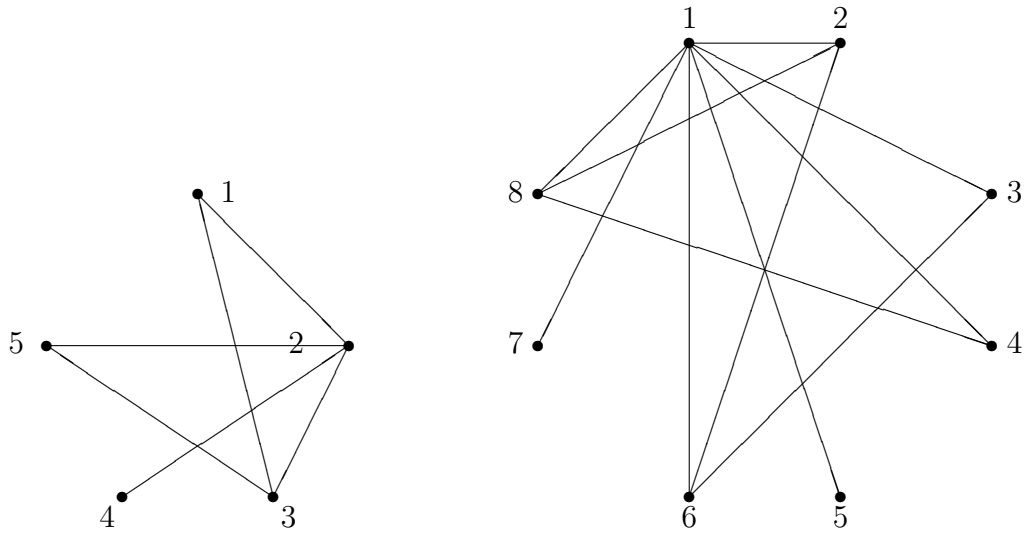


Abbildung 3.1: Zwei Beispiele von Graphen

$V = \{1, 2, \ldots, 8\}$ und $\{i, j\} \in E$, falls i, j in der Teilerrelation $i|j$ stehen. Der zugehörige Graph wird in Abbildung 3.1 rechts dargestellt. Wir haben diese Graphen, sozusagen per Hand, mit dem Picture-Environment von LaTeX gezeichnet. Wir werden später auch die Möglichkeiten von *Mathematica* ausloten. Hier sei nur erwähnt, dass wir nach `«DiscreteMath‘Combinatorica‘` (Laden eines Zusatzpakets) durch

```
ShowLabeledGraph[MakeGraph[Range[8],(Mod[#1,#2]==0)&]]
```

die Abbildung 3.2 erzeugt haben. □

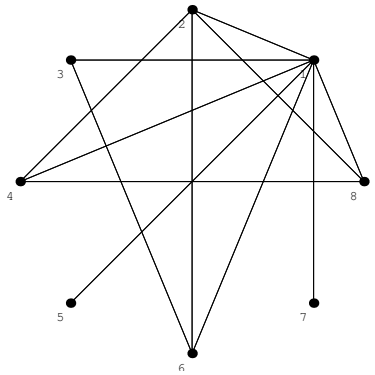


Abbildung 3.2: Derselbe Graph wie in Abbildung 3.1 rechts

Ein Graph $G' = (V', E')$ heißt ein *Untergraph* von $G = (V, E)$, wenn $V' \subset V$, $E' \subset E$. Ferner heißt $G' = (V', E')$ ein *induzierter Untergraph* wenn G' alle Kanten zwischen den Ecken von E' enthält, die auch in G enthalten sind. Die Kante $\{x, y\}$ bezeichnen wir auch mit xy, x und y sind ihre *Endecken*. Einige Ausdruckweisen sind dann völlig klar: Die Kante xy *verbindet* die Ecken x und y, *benachbarte Ecken* sind durch eine Kante verbunden, ein *Weg* im Graphen ist eine Folge von paarweise verschiedenen Ecken, die jeweils durch eine Kante verbunden werden können. Dagegen ist ein *Kantenzug* im Graphen eine Folge von paarweise verschiedenen Kanten, wobei aufeinanderfolgende

Kanten jeweils eine Ecke gemein haben. Ein *Kreis* ist ein geschlossener Weg, also eine Folge von paarweise verschiedenen Ecken $x_1, x_2, \ldots, x_n$ mit $x_i x_{i+1} \in E$, $i = 1, \ldots, n-1$, und $x_n x_1 \in E$, die Anzahl n der Ecken bzw. Kanten ist die *Länge* des Kreises. Man beachte, dass wir zwischen einem Kreis und einem geschlossenen Kantenzug unterscheiden: Jeder Kreis ist auch ein geschlossener Kantenzug, aber nicht umgekehrt. Ein Kreis der Länge 3 ist ein *Dreieck*, usw. Wir sagen, $y \in V$ sei *erreichbar* von $x \in V$, falls es einen Weg mit Anfangsecke x und Endecke y gibt. Offenbar ist Erreichbarkeit eine Äquivalenzrelation auf der Eckenmenge V. Die auf den einzelnen Äquivalenzklassen induzierten Untergraphen heißen die (zusammenhängenden) Komponenten von G. Eine Kante heißt eine *Brücke*, falls die Entfernung der Kante die Anzahl der Komponenten erhöht. Ein Graph heißt *zusammenhängend*, falls er nur eine Komponente hat. Auf der Menge V der Ecken eines Graphen kann ein "Abstandsbegriff" eingeführt werden. Der *Abstand* $d(x, y)$ zweier Ecken x und y ist die Länge eines kürzesten Weges von x nach y, wobei wir $d(x, x) := 0$ setzen. Falls ein solcher Weg nicht existiert (dann liegen x und y in verschiedenen Komponenten), setzen wir $d(x, y) := \infty$.

Einige weitere Definitionen sind wichtig. Ein Graph $G = (V, E)$ heißt *vollständig*, wenn je zwei Ecken durch eine Kante verbunden sind. Ist n die Anzahl der Ecken, so ist also

$$\sum_{j=1}^{n}(n-j) = \frac{n(n-1)}{2} = \binom{n}{2}$$

die Anzahl der Kanten in einem vollständigen Graphen mit n Ecken. Dieser vollständige Graph wird mit K_n bezeichnet. Weiter heißt ein Graph *bipartit*, wenn die Menge

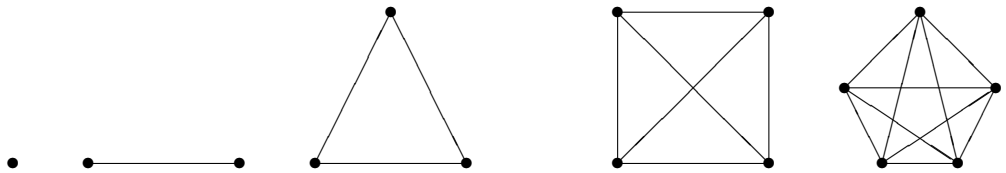


Abbildung 3.3: Vollständige Graphen

der Ecken V disjunkte Vereinigung von zwei Mengen U und W ist und jede Kante eine Ecke in U und eine in W besitzt. Bipartite Graphen treten z. B. beim Transportproblem auf, wo die Menge der Ecken die disjunkte Vereinigung der "Lagerecken" und der "Kundenecken" ist und ein Transport stets von einem Lager zu einem Kunden erfolgt. Man spricht von einem *vollständigen bipartiten Graphen*, wenn alle Kanten zwischen U und W vorhanden sind. Ist $|U| = m$ und $|W| = n$, so wird er mit $K_{m,n}$ bezeichnet.

Beispiel: Durch `«DiscreteMath‘Combinatorica‘` wird in *Mathematica* ein Zusatzpaket geladen[3], wodurch Hilfsmittel zur Kombinatorik und zur Graphentheorie zur

[3]Genaueres zum Zusatzpaket `DiscreteMath‘Combinatorica‘` findet man bei

S. Skiena (1990) *Implementing Discrete Mathematics. Combinatorics and Graph Theory with Mathematica*. Addison-Wesley, New York.

Verfügung gestellt werden. Hier wollen wir uns zunächst mit der *Darstellungsmöglichkeit spezieller Graphen* begnügen. Z. B. wird durch `ShowGraph[CompleteGraph[17]]` der K_{17} dargestellt, siehe Abbildung 3.4 links. Auch die Darstellung vollständiger bi-

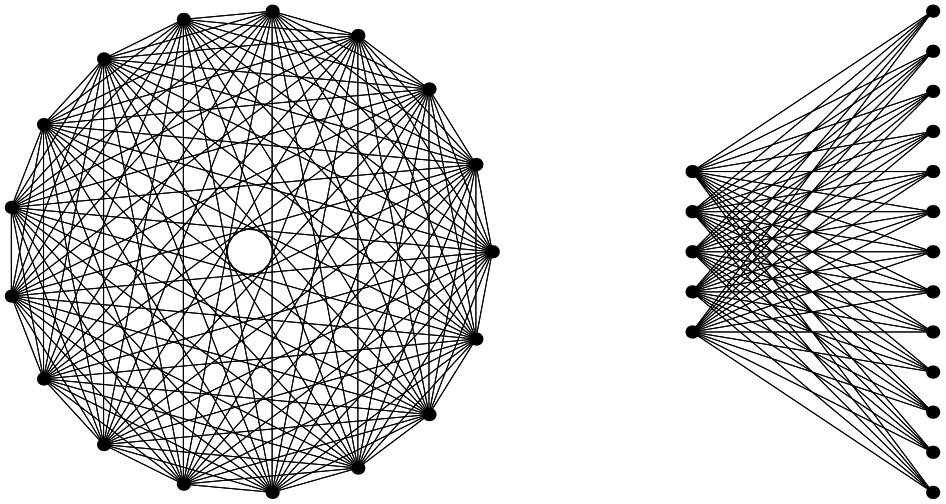


Abbildung 3.4: Die vollständigen Graphen K_{17} und $K_{5,13}$

partiter Graphen ist entsprechend möglich. Z. B. ist rechts in Abbildung 3.4 der $K_{5,13}$ durch `ShowGraph[CompleteGraph[5,13]]` dargestellt worden. Dagegen haben wir die Graphen in Abbildung 3.5 durch `ShowGraph[Wheel[17]]; ShowGraph[Star[17]]` erzeugt. Weitere interessante Graphen sind in Abbildung 3.6 angegeben. Sie wurden

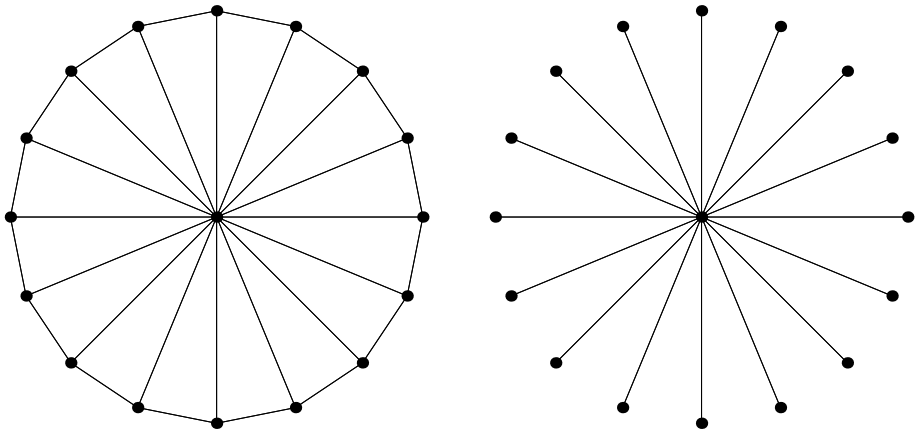


Abbildung 3.5: Ein Rad und ein Stern

durch `ShowGraph[Turan[13,5]]; ShowGraph[Hypercube[5]]` erzeugt. Hierbei liefert der Aufruf `Turan[n,p]` einen Graphen mit n Ecken und einer maximalen Anzahl von Kanten, der aber nicht einen K_p enthält, also keinen vollständigen Graphen mit p Ecken. Durch `Hypercube[n]` wird ein Graph Q_n erzeugt, der sogenannte *Hyperwürfel*, dessen Eckenmenge V geben ist durch alle 0,1-Folgen der Länge n. Offenbar ist dann

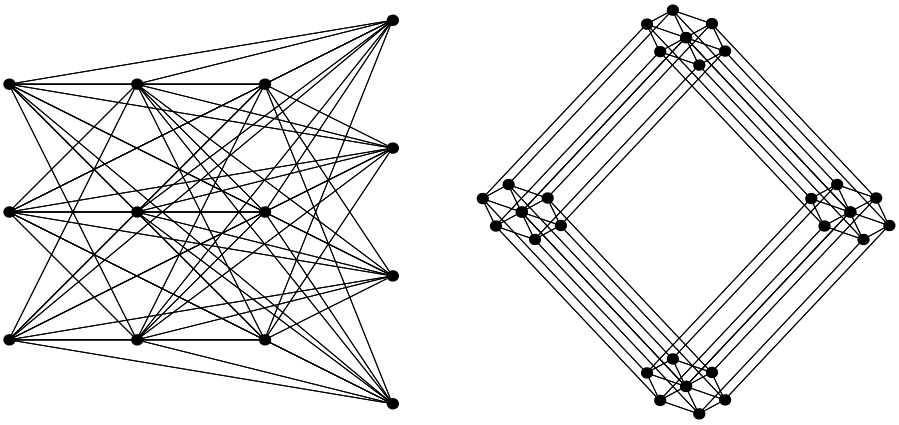


Abbildung 3.6: Weitere spezielle Graphen

$|V| = 2^n$. Zwei Ecken $x, y \in V$ werden durch eine Kante xy verbunden, wenn sich die beiden 0,1-Folgen x und y an genau einer Stelle unterscheiden. Man kann zeigen, dass $|E(Q_n)| = n2^{n-1}$ (siehe Aufgabe 1). □

Zwei Graphen $G = (V, E)$ und $G' = (V', E')$ heißen *isomorph*, falls es eine Bijektion $\phi: V \longrightarrow V'$ gibt, so dass $xy \in E$ genau dann, wenn $\phi(x)\phi(y) \in E'$. Welche *notwendigen* Bedingungen müssen erfüllt sein, damit zwei Graphen isomorph sind? Es genügt doch wohl kaum, dass die Anzahl der Ecken und der Kanten übereinstimmt. Sind z. B. die beiden Graphen in Abbildung 3.7 isomorph? Bevor diese Frage beantwortet wird,

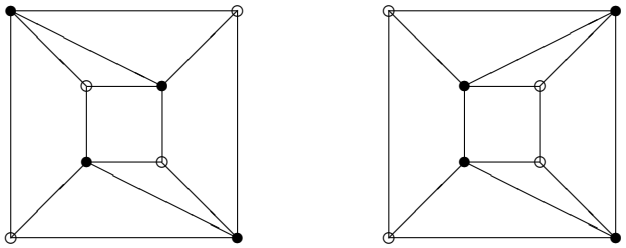


Abbildung 3.7: Sind diese Graphen isomorph?

werden weitere Begriffe eingeführt. Der *Grad* $d(x)$ einer Ecke $x \in V$ ist die Anzahl der benachbarten Ecken bzw. *inzidierenden* Kanten. Ein Graph heißt *regulär*, wenn alle seine Ecken den gleichen Grad haben. Beide Graphen in Abbildung 3.7 haben vier Ecken vom Grad 4 (gekennzeicnet durch $\bullet$) und vier Ecken vom Grad 3 (gekennzeichnet durch $\circ$). Die sogenannte *Gradfolge* (hier schreibt man die Grade der Knoten in fallender Reihenfolge auf) ist also in beiden Graphen dieselbe, nämlich $4, 4, 4, 4, 3, 3, 3, 3$, sicher eine notwendige Bedingung für Isomorphie. Trotzdem sind die beiden Graphen nicht isomorph. Denn der rechtsstehende Graph besitzt je zwei benachbarte Ecken vom Grad 3, während es im linken Graphen keine benachbarten Ecken vom Grad 3 gibt. Da aber ein Isomorphismus zwischen Graphen jede Ecke auf eine Ecke vom selben Grad abbildet, sind die beiden obigen Graphen nicht isomorph.

Wir wollen ein erstes kleines Lemma formulieren und beweisen.

Lemma 1.1 *Sei $G = (V, E)$ ein Graph und $|E|$ die Anzahl der Kanten. Dann gilt:*

1. *Es ist*
$$\sum_{x \in V} d(x) = 2|E|.$$
2. *G besitzt eine gerade Anzahl von Ecken ungeraden Grades.*

Beweis: Wir betrachten die Menge aller Paare

$$P := \{(x, e) \in V \times E : x \in e\}$$

und berechnen die Anzahl ihrer Elemente auf zweierlei Weise. Da jede Kante $e \in E$ zwei Endecken besitzt, ist $|P| = 2|E|$. Andererseits gibt es zu jeder Ecke $x \in V$ genau $d(x)$ benachbarte Ecken, so dass $|P| = \sum_{x \in V} d(x)$. Zum Beweis des zweiten Teils des Satzes bezeichnen wir mit V_g die Menge der Ecken geraden Grades und mit V_u die Menge der Ecken ungeraden Grades. Dann ist

$$\sum_{x \in V_u} d(x) = 2|E| - \sum_{x \in V_g} d(x)$$

eine gerade Zahl und folglich ist $|V_u|$ gerade. Damit ist der einfache Beweis abgeschlossen. □

Insbesondere ist die Summe der Grade in einem Graphen eine gerade Zahl. Diese Aussage heißt auch manchmal das "Handshaking Lemma", da es die Tatsache ausdrückt, dass bei einer Party die Gesamtzahl der geschüttelten Hände eine gerade Zahl ist.

Um die bisher eingeführten Begriffe zu üben, geben wir noch zwei interessante Sätze an und beweisen sie.

Satz 1.2 [4]*Die Kantenmenge eines Graphen kann genau dann in Kreise partitioniert werden, wenn jede Ecke geraden Grad besitzt.*

Beweis: Zunächst nehmen wir an, die Kantenmenge eines Graphen sei die disjunkte Vereinigung von Kreisen. Eine beliebige Ecke des Graphen ist entweder isoliert (dann hat sie den Grad 0) oder es führen Kanten zu ihr hin und von ihr fort, folglich ist sie in einer bestimmten Zahl k von Kreisen enthalten. Die Ordnung ist dann $2k$, also gerade.

Umgekehrt nehmen wir nun an, jede Ecke des Graphen habe geraden Grad. Außerdem können wir natürlich annehmen, dass nicht alle Ecken isoliert sind, es also überhaupt Kanten im Graphen gibt. Wie kann man einen Kreis im Graphen finden? Sei $x_0 x_1 \dots x_l$ ein Weg maximaler Länge l im Graphen. Da x_0 nicht isoliert ist und geraden Grad besitzt, hat x_0 außer x_1 noch einen weiteren Nachbarn, etwa y. Es ist $y = x_i$ für ein $i \in \{2, \dots, l\}$, denn andernfalls wäre $y x_0 \dots x_l$ ein Weg der Länge $l + 1$. Damit hat man zunächst wenigstens einen Kreis gefunden, nämlich $x_0 x_1 \dots x_i$. Nun entferne man aus dem Graphen alle Kanten des gerade gefundenen Kreises und wende auf den so entstandenen Graphen dieselbe Argumentation an. Nach endlich vielen Schritten ist die Behauptung bewiesen. □

[4]Siehe B. Bollobás (1998, S. 5).

Satz 1.3 [5]*Ein Graph $G = (V, E)$ mit $|V| \geq 2$ Ecken ist genau dann bipartit, wenn alle Kreise gerade Länge haben. Insbesondere ist G bipartit, wenn keine Kreise existieren.*

Beweis: Es wird o. B. d. A. angenommen, dass G zusammenhängend ist. Zunächst nehmen wir an, G sei bipartit mit der Eckenmenge $V = U \cup W$. Jeder Kreis muss abwechselnd zwischen U und W verlaufen, also gerade Länge haben. Umgekehrt nehmen wir nun an, dass alle Kreise gerade Länge haben. Wir wählen $x \in V$ beliebig und setzen

$$U := \{y \in V : d(x, y) \text{ gerade}\}, \qquad W := \{y \in V : d(x, y) \text{ ungerade}\}.$$

Insbesondere ist $x \in U$. Zu zeigen bleibt, dass Kanten nur zwischen Ecken aus U und W verlaufen bzw. Ecken aus U oder W nicht benachbart sind. Wir nehmen das Gegenteil an. Es seien also $y, z \in U$ und $yz \in E$ (der Fall $y, z \in W$ kann analog behandelt werden). Dann ist $|d(x, y) - d(x, z)| \leq 1$, da yz eine Kante ist. Da $d(x, y)$ und $d(x, z)$ beide gerade sind, ist $d(x, y) = d(x, z)$. Sei P ein Wege von x nach y der Länge $d(x, y)$ und entsprechend P' ein Wege von x nach z der Länge $d(x, z)$. Mit v werde die letzte gemeinsame Ecke von P und P' bezeichnet (es ist $v = x$ möglich, dass also P und P' nur die Startecke gemein haben). Da $d(x, y) = d(x, z)$, ist auch $d(v, y) = d(v, z)$, d. h. die Wege von v nach y und v nach z haben dieselbe Länge. Dann ist aber der Weg von v nach y, die Kante yz und der Wege von z nach v ein Kreis der ungeraden Länge $2d(v, y) + 1$, ein Widerspruch. □

3.2 Darstellung von Graphen

Die geometrische Darstellung von Graphen, wie wir sie in obigen Abbildungen benutzt haben, ist als Eingabe in einen Computer natürlich nicht geeignet. Hierzu muss man sich andere Möglichkeiten überlegen, worauf wir jetzt eingehen wollen.

Sei $G = (V, E)$ ein Graph mit $V = \{v_1, \ldots, v_n\}$ und $E = \{e_1, \ldots, e_m\}$. Dann ist die *Adjazenzmatrix* $A = (a_{ij}) \in \mathbb{R}^{n \times n}$ definiert durch

$$a_{ij} := \begin{cases} 1 & \text{falls} \quad v_i v_j \in E, \\ 0 & \text{sonst.} \end{cases}$$

Daher ist A eine symmetrische Matrix mit Nullen in der Hauptdiagonalen (da ja keine Schleifen auftreten), weiter ist offenbar

$$d(v_i) = \sum_{j=1}^{n} a_{ij}, \qquad d(v_j) = \sum_{i=1}^{n} a_{ij}.$$

Die *Inzidenzmatrix* $B = (b_{ij}) \in \mathbb{R}^{n \times m}$ ist definiert durch

$$b_{ij} := \begin{cases} 1 & \text{falls} \quad v_i \in e_j, \\ 0 & \text{sonst.} \end{cases}$$

Einen einfachen Zusammenhang zwischen Adjazenz- und Inzidenzmatrix eines Graphen gibt das folgende Lemma an.

[5]Siehe M. Aigner (1996, S. 98), B. Bollobás (1998, S. 9) und R. Diestel (1998, S. 14).

Lemma 2.1 *Sei $G = (V, E)$ ein Graph mit $V = \{v_1, \ldots, v_n\}$ und $E = \{e_1, \ldots, e_m\}$. Sei $A \in \mathbb{R}^{n\times n}$ die Ajazenz- und $B \in \mathbb{R}^{n\times m}$ die Inzidenzmatrix von G. Weiter sei $D := \text{diag}\,(d(v_1), \ldots, d(v_n))$. Dann ist $BB^T = D + A$.*

Beweis: Der Beweis wird als (einfache) Übungsaufgabe überlassen. □

Beispiel: Die Adjazenz- und die Inzidenzmatrix eines Graphen hängt jeweils von der Nummerierung der Ecken bzw. der Ecken und Kanten ab. Z. B. betrachte man die beiden Graphen in Abbildung 3.8. Den ersten Graphen haben wir durch

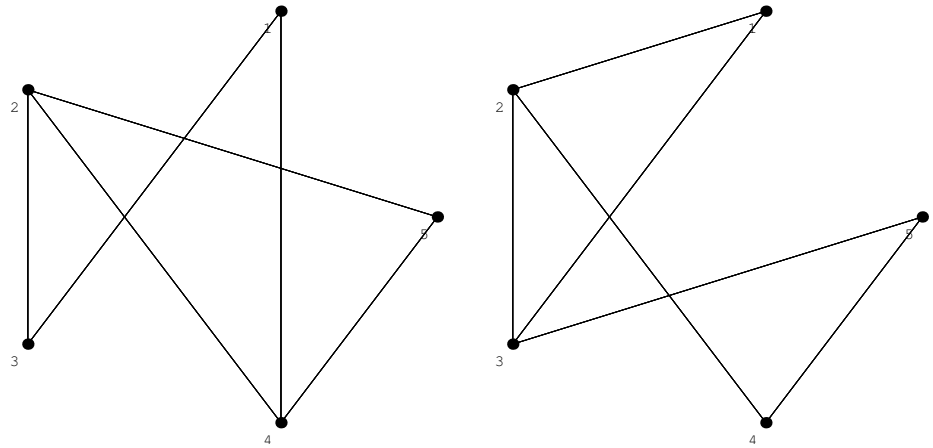


Abbildung 3.8: Zwei unterschiedlich nummerierte Graphen

```
adjdata1={{3,4},{3,4,5},{1,2},{1,2,5},{2,4}}
gr1=FromAdjacencyLists[adjdata1]
labgr1=ShowLabeledGraph[gr1]
```

dargestellt, entsprechend der zweite durch

```
adjdata2={{2,3},{1,3,4},{1,2,5},{2,5},{3,4}}
gr2=FromAdjacencyLists[adjdata2]
labgr2=ShowLabeledGraph[gr2]
```

Natürlich sind die beiden Graphen isomorph. Nach `IsomorphicQ[gr1,gr2]` erhält man `True` als Output, nach `Isomorphism[gr1,gr2]` die Liste $\{4, 3, 5, 2, 1\}$, welche gerade die Permutation der Ecken angibt, welche den Isomorphismus der beiden Graphen bestimmt. Durch `Isomorphism[gr1,gr2,All]` erhalten wir $\{\{4, 3, 5, 2, 1\}, \{5, 2, 4, 3, 1\}\}$. Wir haben die Graphen durch sogenannte Adjazenzlisten erzeugt. Dies ist eine Liste von Listen. In der i-ten Liste sind die Ecken aufgeführt, die zur i-ten Ecke benachbart sind. Aus den Graphen kann man in einfacher Weise die Adjazenz- und Inzidenzmatrizen erhalten. Durch

```
MatrixForm[Edges[gr1]]
MatrixForm[Edges[gr2]]
```

erhält man die Adjazenzmatrizen

$$A_1 = \begin{pmatrix} 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 \end{pmatrix}, \qquad A_2 = \begin{pmatrix} 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 \\ 1 & 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 1 & 0 \end{pmatrix}.$$

Im zweiten Fall erhält man eine Bandmatrix kleinerer Breite. Durch[6]

```
MatrixForm[Transpose[IncidenceMatrix[gr1]]]
MatrixForm[Transpose[IncidenceMatrix[gr2]]]
```

erhält man die Inzidenzmatrizen

$$B_1 = \begin{pmatrix} 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 \end{pmatrix}, \qquad B_2 = \begin{pmatrix} 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 \end{pmatrix}.$$

Hierbei sind im ersten Fall die Kanten durch $E = \{13, 14, 23, 24, 25, 45\}$ sortiert, im zweiten Fall durch $E = \{12, 13, 23, 24, 35, 45\}$. Die Kanten in den Graphen in Abbildung 3.8 überkreuzen sich. Man nennt einen Graphen *planar*, wenn man einen isomorphen Graphen in der Ebene angeben kann, bei dem sich zwei Kanten höchstens in einer Ecke schneiden. Zur Entscheidung, ob ein Graph planar ist, stellt *Mathematica* den Befehl `PlanarQ` zur Verfügung. Hierdurch stellen wir fest, dass beide Graphen planar sind. Dies erkennt man deutlich, wenn man sie wie in Abbildung 3.9 darstellt. Wie die

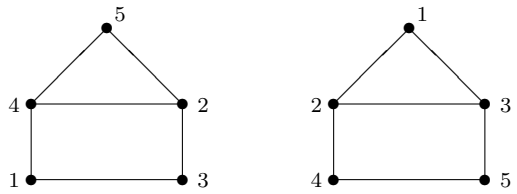


Abbildung 3.9: Die Graphen in Abbildung 3.8 als planare Graphen

Planarität eines Graphen festgestellt werden kann, darauf wollen wir nicht eingehen. □

3.3 Bäume

Weitere in der Graphentheorie wichtige Begriffe folgen. Ein Graph, der keinen Kreis enthält, heißt ein *Wald*. Ein zusammenhängender Wald heißt ein *Baum*. Sei $T = (V, E)$ ein Baum und $v_0 \in V$ beliebig. Auf die folgende Weise kann man T als einen Baum mit der *Wurzel* v_0 auffassen (oder einen *Wurzelbaum*, man denke etwa an Inhaltsverzeichnisse für die Dateien auf der Festplatte eines Computers).

[6] Weshalb hier erst durch `Transpose` die richtige Inzidenzmatrix gewonnen wird, verstehe ich nicht.

- Setze $V_0 := \{v_0\}$.
- Für $k = 1, \ldots,$:
  - Falls $V_0 \cup \cdots \cup V_{k-1} \neq V$, so sei $V_k \subset V$ die Menge aller Ecken, die einen Nachbarn in V_{k-1} haben.

Dieses "Verfahren" muss natürlich nach endlich vielen Schritten abbrechen. Auf diese Weise ist es möglich, die Knoten in endlich vielen Schichten so zu arrangieren, dass es Kanten nur zu Knoten benachbarter Schichten gibt. Ein *Wurzelbaum* (T, v_0) ist also ein Baum T zusammen mit einem ausgezeichneten Knoten v_0, der *Wurzel*. Als *Tiefe* eines Knotens von T wird sein Abstand von der Wurzel bezeichnet. Die *Tiefe* von T ist die Tiefe des Knotens von T mit dem größten Abstand zur Wurzel. Alle Knoten der gleichen Tiefe bilden ein *Knotenniveau*. Als *Kinder* eines Knotens v von T werden sämtliche Knoten bezeichnet, die zu v benachbart sind und deren Tiefe die von v um eins übersteigt. v heißt *Vater* seiner Kinder. Die Knoten eines Baumes vom Grad 1 werden *Blätter* genannt, die Knoten vom Grad größer als 1 heißen *innere Knoten*.

Ist $G = (V, E)$ ein zusammenhängender Graph, so heißt ein Untergraph H der selben Ordnung $|V|$ wie G, der ein Baum ist, ein (den Graphen G) aufspannender Baum (spanning tree). Offenbar besitzt jeder zusammenhängende Graph G (mindestens) einen aufspannenden Baum. Entweder ist G bereits ein Baum, dann sind wir fertig, oder G besitzt einen Kreis C. Entfernen wir aus C eine beliebige Kante e, so ist $G_1 = (V, E \setminus \{e\})$ nach wie vor zusammenhängend. Auf G_1 kann dasselbe Verfahren angewandt werden und nach endlich vielen Schritten erhält man einen aufspannenden Baum.

Im folgenden Satz werden notwendige und hinreichende Bedingungen dafür angegeben, dass ein Graph ein Baum ist.

Satz 3.1 *Sei $G = (V, E)$ ein Graph. Die folgenden Bedingungen sind äquivalent:*

1. *G ist ein Baum.*
2. *Je zwei Ecken in G sind durch genau einen Weg verbunden.*
3. *G ist zusammenhängend und es gilt $|E| = |V| - 1$.*

Beweis: Sei G ein Baum, insbesondere also zusammenhängend. Zwei beliebige Ecken $x, y \in V$ können durch mindestens einen Weg verbunden werden. Würde es zwei Wege von x nach y geben, so hätte man einen Kreis, was in einem Baum nicht möglich ist. Sind umgekehrt je zwei Ecken in G durch genau einen Weg verbunden, so ist G insbesondere zusammenhängend. Gäbe es in G einen Kreis, so sind je zwei Ecken dieses Kreises durch zwei verschiedene Wege verbunden. Damit ist die Äquivalenz der ersten beiden Aussagen bewiesen.

Wir nehmen an, $G = (V, E)$ sei ein Baum. Dann ist G insbesondere zusammenhängend und es bleibt zu zeigen, dass $|E| = |V| - 1$. Hierzu überlegen wir uns, dass es in einem Baum Ecken vom Grad 1 gibt. Ist nämlich $P = x, x_1, \ldots, y$ ein längster Weg in G, so sind alle Nachbarn von x (und von y) in P, also $d(x) = d(y) = 1$ (andernfalls könnte der Weg verlängert werden, da G keinen Kreis enthält). Man setze

$G_1 := (V_1, E_1) := (V \setminus \{x\}, E \setminus \{xx_1\})$. Dann ist auch G_1 ein Baum und $|V_1| - |E_1| = |V| - |E|$. Nach $n-2$ Schritten (hierbei bezeichne $n := |V|$ die Ordnung von G) erhalten wir einen Baum $G_{n-2} = (V_{n-2}, E_{n-2})$ auf 2 Ecken, d. h. es ist $G_{n-2} = K_2$ und es gilt $|V| - |E| = |V_{n-2}| - |E_{n-2}| = 1$. Umgekehrt sei $G = (V, E)$ ein zusammenhängender Graph mit $|V| = |E| - 1$. Sei T ein aufspannender Baum von G. Wegen $V(T) = V(G)$ und $E(T) \leq E(G)$ ist

$$1 = |V(G)| - |E(G)| \leq |V(T)| - |E(T)| = 1,$$

wobei wir für die letzte Gleichung die gerade eben bewiesene Aussage auf T anwenden. Folglich ist $|E(T)| = |E(G)|$ und daher $G = T$ selbst schon ein Baum. □

Ein besonders interessantes Problem besteht darin, zu einem gewichteten oder bewerteten zusammenhängenden Graphen einen minimalen aufspannenden Baum zu bestimmen. Genauer handelt es sich um das folgende Problem. Gegeben sei ein zusammenhängender Graph $G = (V, E)$ und eine auf den Kanten definierte Kostenfunktion $f: E \longrightarrow \mathbb{R}$. Unter allen G aufspannenden Bäumen $T = (V, E')$ finde man einen mit minimalen Kosten $f(T) := \sum_{xy \in E'} f(xy)$. Man kann sich das Problem etwa folgendermaßen vorstellen: Die Ecken im Graphen sind gewisse Dörfer. Zwischen zwei Ecken bzw. Dörfern gibt es eine zugehörige Kante, wenn es möglich ist, zwischen diesen Dörfern eine Leitung (Strom, Wasser usw.) zu bauen. Bekannt seien die Kosten zum Bau einer Leitung zwischen zwei Dörfern, also eine Kantenbewertung. Gesucht ist ein möglichst günstiges Leitungssystem zwischen den Dörfern. Man spricht auch von einem minimalen aufspannenden Baum (MST). Z. B. betrachte man den in Abbildung 3.10 angegebenen Graphen mit Kantenbewertung. Zur Bestimmung eines minimalen

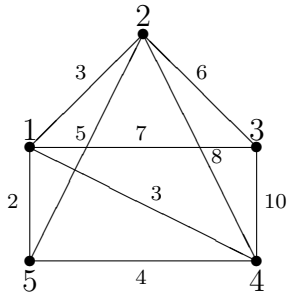


Abbildung 3.10: Was ist der kürzeste aufspannende Baum?

$G = (V, E)$ aufspannenden Baumes $T' = (V, E')$ geben wir den folgenden Algorithmus an:

1. Sei $E' := \emptyset$, $F := E$.
2. Bestimme $e' \in F$ mit $f(e') = \min_{e \in F} f(e)$. Setze $F := F \setminus \{e'\}$.
3. Besitzt der Graph $(V, E' \cup \{e'\})$ einen Kreis, so gehe zu 2.
4. Besitzt der Graph $(V, E' \cup \{e'\})$ keinen Kreis, so setze $E' := E' \cup \{e'\}$. Ist $|E'| = |V| - 1$, dann STOP, andernfalls gehe zu 2.

Zunächst bestimme man also die kürzeste Kante, d. h. diejenige mit der niedrigsten Bewertung. Jede danach zu wählende Kante wird als eine kürzeste unter den verbleibenden Kanten gewählt, es sei denn, man würde dadurch einen Kreis erzeugen. Der Prozess endet, wenn man keine Kante hinzufügen kann, ohne einen Kreis zu erzeugen. Dies ist der Fall, wenn $|V|-1$ Kanten hinzugefügt sind[7]. Man nennt diesen Algorithmus einen *Greedy-Algorithmus* (greedy=gefräßig), weil in jedem Schritt die zu dieser Zeit lokal beste Möglichkeit ausgewählt wird.

Wir wollen uns das Verfahren an dem obigen Graphen klar machen. Die einzelnen Schritte findet man in Abbildung 3.11. Im letzten Schritt haben wir nicht die Kante

Abbildung 3.11: Konstruktion des minimalen aufspannenden Baumes

54 mit der Bewertung 4 wählen dürfen, weil wir dann den Kreis (genauer ein Dreieck) 145 erzeugt hätten. Insgesamt haben wir einen G aufspannenden Baum der Länge 14 erzeugt.

Nun kann man beweisen (gar nicht so ganz einfach!)[8]:

Satz 3.2 *Der durch den obigen Algorithmus (Verfahren von Kruskal) zum zusammenhängenden Graphen $G = (V, E)$ erzeugte Graph $T' = (V, E')$ ist ein kürzester G aufspannender Baum.*

Beweis: Nach Konstruktion ist $T' = (V, E')$ ein maximaler kreisfreier Untergraph von G und als solcher ein G aufspannender Baum, wenn noch gezeigt werden kann, dass T' zusammenhängend ist. Angenommen, $x, y \in V$ seien zwei beliebige Ecken. Wir wollen zeigen, dass man auf einem Weg in T', der also nur Kanten aus E' enthält, von x nach y kommen kann. Da G als zusammenhängend vorausgesetzt wurde, gibt es in G einen Weg von x nach y, etwa $xz_1 \cdots z_k y$. Mit $z_0 := x$, $z_{k+1} := y$ ist also $z_i z_{i+1} \in E$, $i = 0, \ldots, k$. Es genügt zu zeigen, dass man von z_i nach z_{i+1} auf einem Weg in T' gelangen kann, $i = 0, \ldots, k$. Ist $z_i z_{i+1} \in E'$, so ist dies trivialerweise der Fall. Ist

[7] Jeder Graph $G = (V, E)$ mit $|E| \geq |V|$ enthält einen Kreis. Denn angenommen, das wäre nicht der Fall. Dann wäre G ein Wald und seine Zusammenhangskomponenten daher Bäume. Ist t die Anzahl der Zusammenhangskomponenten von G, so $|E| = |V| - t < |V|$, ein Widerspruch.

[8] Siehe z. B.

W. J. Cook, W. H. Cunningham, W. R. Pulleyblank, A. Schrijver (1998, S. 13) *Combinatorial Optimization*. J. Wiley, New York

und

B. Bollobás (1998, S. 13) *Modern Graph Theory*. Springer, New York-Berlin-Heidelberg.

dagegen $z_i z_{i+1} \notin E'$, so existiert in $(V, E' \cup \{z_i z_{i+1}\})$ ein Kreis, was impliziert, dass man auch in diesem Fall von z_i nach z_{i+1} über Kanten aus E' kommen kann. Insgesamt gibt es also von x nach y einen Weg in T', d. h. T' ist zusammenhängend.

Nachdem wir nun gezeigt haben, dass der Kruskal-Algorithmus mit einem G aufspannenden Baum T' endet, wollen wir zeigen, dass T' ein MST ist. Hierzu nehmen wir an, $T^* = (V, E^*)$ sei ein MST, der maximal viele Kanten mit $T' = (V, E')$ gemeinsam hat. Wir werden die Annahme $E' \neq E^*$ zum Widerspruch führen. Die Kanten in E' werden Schritt für Schritt bestimmt. Da wir $E' \neq E^*$ annehmen, gibt es eine erste Kante xy in E', die keine Kante in E^* ist. In T^* gibt es genau einen Weg P von x nach y (wegen des Zusammenhangs von T^* ist die Existenz eines Weges klar, die Eindeutigkeit folgt wegen der Kreisfreiheit von T^*). Es gibt wenigstens eine Kante in P, die nicht zu E' gehört, denn andernfals gibt es einen Kreis in T' (da $xy \in E'$). Sei etwa uv eine solche Kante. Wir überlegen uns, dass $f(xy) \leq f(uv)$. Dies ist sicherlich richtig, wenn man sich überlegt, dass zu dem Zeitpunkt, als xy als Kante gewählt wurde, auch durch Hinzufügen der Kante uv kein Kreis entstanden wäre. Denn alle Kanten, die vor xy erzeugt sind, liegen in T^*. Wenn also durch Hinzufügen von $uv \in E^*$ ein Kreis entstanden wäre, so hätte man einen Widerspruch zur Kreisfreiheit von T^*. Also ist $f(xy) \leq f(uv)$. Definiert man $T := (V, (E^* \setminus \{uv\}) \cup \{xy\})$, so ist T ein G aufspannender Baum (denn T ist zusammenhängend, da es einen Weg von u nach v in T gibt, ferner gibt es auch in T keinen Kreis) mit

$$f(T) = f(T^*) - f(uv) + f(xy) \leq f(T^*).$$

Da T^* ein kürzester aufspannender Baum ist, ist es auch T. Da T mehr Kanten mit T' gemein hat als T^*, haben wir einen Widerspruch zur Wahl von T^* erhalten. Folglich ist $E' = E^*$ bzw. $T' = T^*$, also auch T' ein kürzester aufspannender Baum. □

Beispiel: Wir wollen untersuchen, welche Hilfsmittel *Mathematica* bereitstellt, um minimale aufspannende Bäume zu bestimmen. Wir kehren zu dem Beispiel aus Abbildung 3.10 zurück. Durch

```
adjdata={{2,3,4,5},{1,3,4,5},{1,2,4},{1,2,3,5},{1,2,4}}
gr=FromAdjacencyLists[adjdata]
```

erzeugen wir zunächst den in Abbildung 3.12 links stehenden Graphen. In der vorliegen-

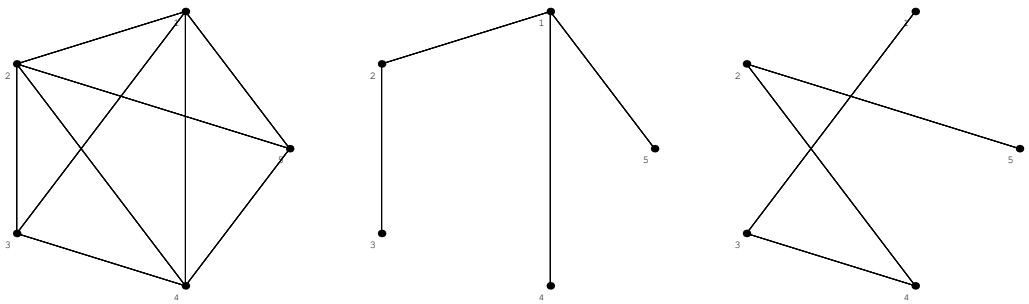


Abbildung 3.12: Minimaler und maximaler aufspannender Baum

den Form ist das ein ungewichteter Graph. Als Antwort auf `UnweightedQ[gr]` erhält

man daher `True`. Um zu einem gewichteten Graphen überzugehen, muss die Adjazenzmatrix verändert und dadurch ein gewichteter Graph definiert werden. Wir definieren eine (gewichtete) Adjazenzmatrix, übergeben die Liste der Ecken und erhalten dadurch einen neuen Graphen. Auf diesen wird durch den Befehl `MinimumSpanningTree` ein minimaler aufspannender Baum erzeugt (und durch `MaximumSpanningTree` ein maximaler aufspannender Baum):

```
g={{0,3,7,3,2},{3,0,6,8,5},{7,6,0,10,0},{3,8,10,0,4},{2,5,0,4,0}};
l=Vertices[gr];
gra=Graph[g,l];
trmi=MinimumSpanningTree[gra];
trma=MaximumSpanningTree[gra];
```

Die entsprechenden aufspannenden Bäume sind ebenfalls in Abbildung 3.12 angegeben. Die Darstellung erfolgt jeweils mit `ShowLabeledGraph`. □

3.4 Kürzeste Wege in Graphen

Gegeben sei ein zusammenhängender Graph $G = (V, E)$ und eine nichtnegative[9] Gewichtsfunktion auf der Kantenmenge E, also $f\colon E \longrightarrow \mathbb{R}_{\geq 0}$. Für $x, y \in V$ sei $P = P(x, y)$ ein x und y verbindender Weg und $E(P)$ die dabei (als Verbindung der paarweise verschiedenen Ecken) auftretenden Kanten. Mit $l(P) := \sum_{e \in E(P)} f(e)$ wird die zugehörige (gewichtete) Länge des Weges P bezeichnet. Beim *Problem des kürzesten Weges* sucht man zu gegebenen Ecken x, y einen x und y verbindenden Weg P mit minimaler (gewichteter) Länge $l(P) = d(x, y)$.

Wir geben nun den Algorithmus von Dijkstra an. Dieser gibt zu einem zusammenhängenden gewichteten Graphen $G = (V, E)$ und einer vorgegebenen Ecke $x \in V$ einen G aufspannenden Baum T an mit der Eigenschaft, dass für *jedes* $y \in V$ der eindeutige Weg von x nach y in T (gäbe es zwei Wege, so enthielte T einen Kreis) kürzester Weg von x nach y in G ist.

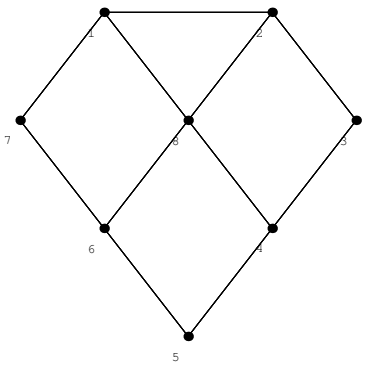
- Gegeben ist ein zusammenhängender Graph $G = (V, E)$ mit $n := |V|$, eine nichtnegative Gewichtsfunktion $f\colon E \longrightarrow \mathbb{R}_{\geq 0}$ und $x \in V$.
- Setze $x_0 := x$, $V_0 := \{x_0\}$, $E_0 := \emptyset$ und $l(x_0) := 0$.
- Für $i = 0, \ldots, n-2$:
  - Betrachte für alle Kanten $e = uv$ mit $u \in V_i$, $v \in V \setminus V_i$ den Ausdruck $g(e) := l(u) + f(e)$ und wähle unter diesen Kanten $e' = u'v'$ mit $g(e') = \min g(e)$.
  - Setze $x_{i+1} := v'$, $e_{i+1} := e'$ und $V_{i+1} := V_i \cup \{x_{i+1}\}$, $E_{i+1} := E_i \cup \{e_{i+1}\}$ sowie $l(x_{i+1}) := g(e')$.

[9] Wir werden uns von dieser Voraussetzung später in Unterabschnitt 4.1.2 befreien.

- Dann ist $T := (V, E_{n-1})$ ein G aufspannender Baum mit der Eigenschaft, dass für jedes $y \in V$ der (in T) eindeutige Weg von x nach y ein minimaler Weg von x nach y in G ist.

Bevor wir beweisen, dass der Algorithmus von Dijkstra wirklich das Verlangte tut, geben wir ein Beispiel an.

Beispiel: Gegeben sei der in Abbildung 3.13 angegebene Graph, wobei die Gewichtung



$$A := \begin{pmatrix} 0 & 3 & 0 & 0 & 0 & 0 & 1 & 1 \\ 3 & 0 & 1 & 0 & 0 & 0 & 0 & 3 \\ 0 & 1 & 0 & 4 & 0 & 0 & 0 & 0 \\ 0 & 0 & 4 & 0 & 2 & 0 & 0 & 2 \\ 0 & 0 & 0 & 2 & 0 & 3 & 0 & 0 \\ 0 & 0 & 0 & 0 & 3 & 0 & 2 & 2 \\ 1 & 0 & 0 & 0 & 0 & 2 & 0 & 0 \\ 1 & 3 & 0 & 2 & 0 & 2 & 0 & 0 \end{pmatrix}$$

Abbildung 3.13: Ein gewichteter Graph und seine Adjazenzmatrix

aus der gewichteten Adjazenzmatrix A zu ersehen ist. Es sei $x := \{1\}$. Wir erhalten dann die folgenden Schritte.

- $V_0 = \{1\}$, $E_0 = \emptyset$.
- $V_1 = \{1, 7\}$, $E_1 = \{17\}$ (statt 7 hätte auch die Ecke 8 aufgenommen werden können).
- $V_2 = \{1, 7, 8\}$, $E_2 = \{17, 18\}$.
- $V_3 = \{1, 7, 8, 2\}$, $E_3 = \{17, 18, 12\}$.
- $V_4 = \{1, 7, 8, 2, 6\}$, $E_4 = \{17, 18, 12, 86\}$.
- $V_5 = \{1, 7, 8, 2, 6, 4\}$, $E_5 = \{17, 18, 12, 86, 84\}$.
- $V_6 = \{1, 7, 8, 2, 6, 4, 3\}$, $E_6 = \{17, 18, 12, 86, 84, 23\}$.
- $V_7 = \{1, 7, 8, 2, 6, 4, 3, 5\}$, $E_7 = \{17, 18, 12, 86, 84, 23, 45\}$.

Der resultierende aufspannende Baum ist in Abbildung 3.14 angegeben. Hieraus liest man z. B. ab, dass der kürzeste Wege von 1 nach 5 durch $\{1, 8, 4, 5\}$ gegeben ist. Bevor wir die Korrektheit des Dijkstra-Verfahrens nachweisen, wollen wir noch die Möglichkeiten von *Mathematica* zur Lösung des Problems der kürzesten Wege testen. Hierzu erzeugen wir zunächst durch

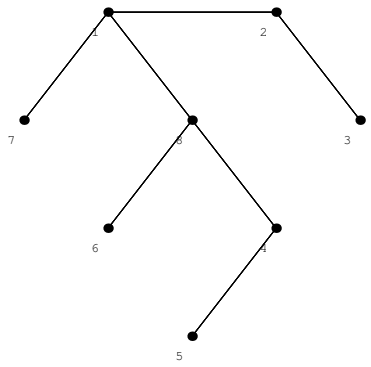


Abbildung 3.14: Resultat des Algorithmus von Dijkstra

```
g={{0,3,0,0,0,0,1,1},{3,0,1,0,0,0,0,3},{0,1,0,4,0,0,0,0},
{0,0,4,0,2,0,0,2},{0,0,0,2,0,3,0,0},{0,0,0,0,3,0,2,2},
{1,0,0,0,0,2,0,0},{1,3,0,2,0,2,0,0}};
l={{-1,3},{1,3},{2,2},{1,1},{0,0},{-1,1},{-2,2},{0,2}};
gra=Graph[g,l];
```

den in 3.13 angegebenen gewichteten Graphen. `ShortestPathSpanningTree[gra,1]` erzeugt den Baum in Abbildung 3.15, eine (geringfügig) andere Lösung als die in Ab-

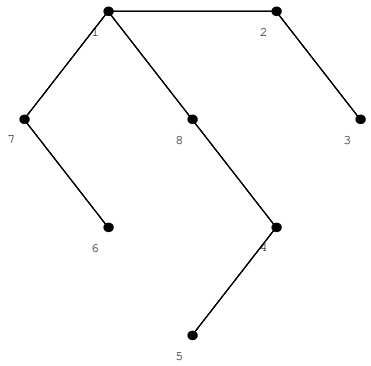


Abbildung 3.15: Ein aufspannender Baum

bildung 3.14 dargestellte. Das Resultat von `ShortestPath[gra,1,5]` ist $\{1, 8, 4, 5\}$. Nach `AllPairsShortestPath[gra]` wird die Matrix

$$\begin{pmatrix} 0 & 3 & 4 & 3 & 5 & 3 & 1 & 1 \\ 3 & 0 & 1 & 5 & 7 & 5 & 4 & 3 \\ 4 & 1 & 0 & 4 & 6 & 6 & 5 & 4 \\ 3 & 5 & 4 & 0 & 2 & 4 & 4 & 2 \\ 5 & 7 & 6 & 2 & 0 & 3 & 5 & 4 \\ 3 & 5 & 6 & 4 & 3 & 0 & 2 & 2 \\ 1 & 4 & 5 & 4 & 5 & 2 & 0 & 2 \\ 1 & 3 & 4 & 2 & 4 & 2 & 2 & 0 \end{pmatrix}$$

ausgegeben. Das (i, j)-Element in dieser Matrix gibt die (gewichtete) Länge eine Weges von der Ecke i zur Ecke j an. □

Satz 4.1 *Sei $G = (V, E)$ ein zusammenhängender Graph, $f: E \longrightarrow \mathbb{R}_{\geq 0}$ eine nichtnegative Kantenbewertung und $x \in V$. Der Algorithmus von Dijkstra ergibt einen G aufspannenden Baum T mit der Eigenschaft, dass für jedes $y \in V$ der in T eindeutige Weg von x nach y ein minimaler (x, y)-Weg mit $d(x, y) = l(y)$ ist.*

Beweis: Offenbar ist $T_i = (V_i, E_i)$ zusammenhängend (leicht durch vollständige Induktion zu zeigen) und $|E_i| = |V_i| - 1$, $i = 1, \ldots, n-1$. Nach Satz 3.1 ist T_i ein Baum, insbesondere also $T_{n-1} = (V, E_{n-1})$ ein G aufspannender Baum. Durch vollständige Induktion nach i zeigen wir, dass für $i = 1, \ldots, n-1$ der in T_i eindeutige Weg von $x = x_0$ nach $y \in V_i \setminus \{x\}$ ein kürzester Weg in G und $d(x, y) = l(y)$ seine Länge ist.

Für den Induktionsanfang ist $i = 1$. Im ersten Schritt wird $x_1 \in V \setminus \{x\}$ so bestimmt, dass die Kante $e_1 = x_0 x_1$ minimales Gewicht hat. Daher ist durch $x_0 x_1$, den einzigen Weg in T_1, ein kürzester Weg von $x = x_0$ nach $y = x_1$ gegeben, ferner ist $l(x_1) = f(e_1) = d(x_0, x_1)$.

Die Aussage sei für i richtig. Zu zeigen ist, dass sie auch für $i+1$ richtig ist. Hierfür genügt es, $y = x_{i+1}$ zu betrachten. Die Kante $e' = u' x_{i+1}$ wurde unter allen Kanten $e = uv$ mit $u \in V_i$, $v \in V \setminus V_i$, so gewählt, dass $l(u) + f(e)$ minimal ist. Mit P_0 werde der Weg von $x = x_0$ über $u' \in E_i$ nach $y = x_{i+1}$ bezeichnet. Seine Länge ist nach Induktionsvoraussetzung

$$l(P_0) = d(x_0, u') + f(e') = l(u') + f(e') = l(x_{i+1}).$$

Wir wollen zeigen, dass P_0 ein kürzester Weg von $x = x_0$ nach $y = x_{i+1}$ ist. Hierzu nehmen wir an, P sei ein kürzester Weg von $x = x_0$ nach $y = x_{i+1}$. Sei u die letzte Ecke aus V_i im Weg P (eine solche muss es geben, denn die erste Ecke $x = x_0$ liegt in V_i), weiter sei $v \in V \setminus V_i$ der Nachfolger (auch diesen muss es geben, denn $y = x_{i+1} \in V \setminus V_i$) im Weg P, ferner sei $e := uv$. Teilwege eines kürzesten Weges sind ebenfalls kürzeste Wege, d. h. in $P = P(x_0, u)P(v, x_{i+1})$ sind auch $P(x_0, u)$ und $P(v, x_{i+1})$ kürzeste Wege von x_0 nach $u \in V_i$ bzw. von v nach x_{i+1}. Nach Induktionsannahme ist ferner

$$l(P(x_0, u)) = d(x_0, u) = l(u).$$

Daher ist

$$\begin{aligned}
d(x_0, x_{i+1}) &= l(P) \\
&= l(P(x_0, u)) + f(e) + l(P(v, x_{i+1})) \\
&= l(u) + f(e) + \underbrace{l(P(v, x_{i+1}))}_{\geq 0} \\
&\geq l(u) + f(e) \\
&\geq l(u') + f(e') \\
&= l(x_{i+1}) \\
&= l(P_0).
\end{aligned}$$

Also ist P_0 ein kürzester Weg und $l(P_0) = l(x_{i+1})$. Damit ist die Induktionsbehauptung und der ganze Satz bewiesen. □

Bemerkung: Es wurde im Beweis entscheidend benutzt, dass die (gewichtete) Länge von Wegen nichtnegativ ist, was natürlich insbesondere für nichtnegative Kantenbewertungen der Fall ist. □

3.5 Eulersche Graphen

Schon als Kind versuchte man, "das Haus des Nikolaus" in einem Zug zu zeichnen, siehe Abbildung 3.16. Bekanntlich ist dies möglich, etwa durch den Kantenzug

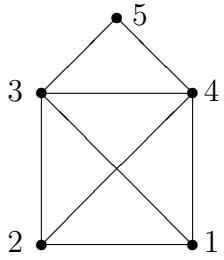


Abbildung 3.16: Das ist das Haus des Nikolaus

$$12, 23, 34, 45, 53, 31, 14, 42.$$

Wie kann man es dem Haus des Nikolaus oder einem anderen Graphen "ansehen", dass dies möglich ist? Noch berühmter ist wohl das Königsberger Brückenproblem[10] von Euler (1736). Links in Abbildung 3.17 haben wir schematisch die 7 Brücken über die Pregel in Königsberg angegeben. Euler stellte 1736 die Frage: Ist es möglich, einen Spaziergang durch Königsberg zu machen, wieder zum Anfangspunkt zurückzukehren und jede Brücke genau einmal zu überqueren? Diese Aufgabe wird am natürlichsten in einem *Multigraphen* formuliert, also in einem Graphen, in dem auch Schlingen und vor allem Mehrfachkanten zugelassen sind. Die Ecken in dem Graphen sind die vier Teile A, B, C, D von Königsberg, die Brücken bilden die Kanten. Da es zwei Brücken von A nach B (und von A nach C) gibt, gibt es zwei Kanten von A nach B (und von A nach C), weiter gibt es jeweils eine Kante von D nach A, B und C. Natürlich könnte man auch zu einem (einfachen) Graphen übergehen, indem man die 7 Brücken a, b, c, d, e, f, g als Hilfsknoten einführt, siehe Abbildung 3.18.

Die folgenden Aussagen könnnten auch für Multigraphen $G = (V, E)$ gemacht werden, in welchem Fall auch Schlingen und Mehrfachkanten erlaubt wären. Wir wollen uns aber nach wie vor auf einfache Graphen beschränken. Sei $q := |E|$ die Anzahl der Kanten in G. Ein geschlossener Kantenzug in G, der jede der q Kanten genau einmal

[10] Aus der Abhandlung von Euler wird bei

L. Volkmann (1991, S. 57 ff.) *Graphen und Digraphen. Eine Einführung in die Graphentheorie*. Springer-Verlag, Wien-New York

zitiert.

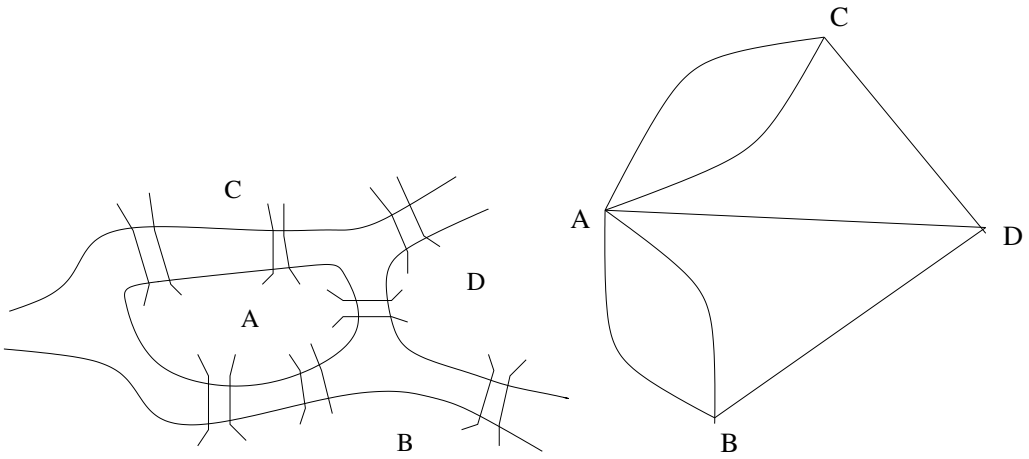


Abbildung 3.17: Die sieben Brücken von Königsberg

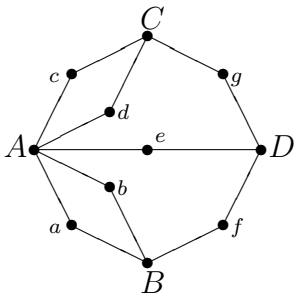


Abbildung 3.18: Das Königsberger Brückenproblem in einem einfachen Graphen

enthält, heißt ein *Euler-Zug*. Ein nicht notwendig geschlossener Kantenzug mit der selben Eigenschaft heißt ein *offener Euler-Zug*. Ein Graph heißt *Eulersch*, wenn er einen Euler-Zug besitzt. Interessant sind natürlich die beiden folgenden Fragen:

1. Wodurch sind Eulersche Graphen charakterisiert?
2. Wie findet man einen Euler-Zug?

Wir beantworten nur die erste Frage und verweisen für die zweite auf M. Aigner (1996, S. 138).

Satz 5.1 *Ein Graph $G = (V, E)$ mit keinen isolierten Ecken und $E \neq \emptyset$ ist genau dann Eulersch, wenn G zusammenhängend ist und alle Ecken geraden Grad haben.*

Beweis: Sei G ein Eulerscher Graph ohne isolierte Ecken. Dann ist G zusammenhängend, denn in einem Euler-Zug kommt jede Ecke wenigstens einmal vor. Da der Euler-Zug jede Ecke über eine gewisse Kante erreicht und über eine andere verlässt, hat jede Ecke geraden Grad. Umgekehrt nehmen wir nun an, G sei zusammenhängend und jede Ecke habe geraden Grad. Da jede Ecke geraden Grad hat, existieren in G Kreise, insbesondere also geschlossene Kantenzüge. Dies folgt aus Satz 1.2, denn nach diesem Satz ist die Kantenmenge in einem Graphen genau dann disjunkte Vereinigung von Kreisen, wenn der Grad jeder Ecke gerade ist. Sei nun C ein geschlossener Kantenzug

mit einer maximalen Anzahl von Kanten $E(C)$. Ist $E = E(C)$, so sind wir fertig. Andernfalls sei $G' := (V, E \setminus E(C))$. Da G zusammenhängend ist, muss es eine Ecke u auf dem geschlossenen Kantenzug C geben, die mit einer Kante aus G' inzidiert. In G' hat wieder jede Ecke geraden Grad, daher gibt es einen geschlossenen Kantenzug C' in G', der u enthält. Schiebt man C' beim Durchlaufen von C an der Stelle u ein, so erhält man einen Widerspruch zur Maximalität von C. Insgesamt ist der Satz bewiesen. □

Beispiele: In Abbildung 3.18 erkennt man, dass die Ecken A, B, C, D jeweils ungeraden Grad, nämlich $5, 3, 3, 3$, besitzen, daher ist der Graph nicht Eulersch bzw. ein Spaziergang durch Königsberg, bei dem jede der 7 Brücken genau einmal überquert wird, nicht möglich. Dagegen ist das in Abbildung 3.19 angegebene erweiterte Haus des Nikolaus ein Eulerscher Graph. Denn der Grad jeder Ecke in dem Graphen ist 2

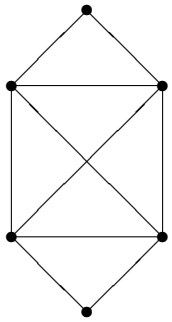


Abbildung 3.19: Das (erweiterte) Haus des Nikolaus

oder 4. □

Im nächsten Satz werden Bedingungen dafür angegeben, dass ein offener Euler-Zug von einer Ecke x zu einer Ecke $y \neq x$ existiert. In Abbildung 3.16 konnten wir für das Haus des Nikolaus einen offenen Euler-Zug von der Ecke 1 zur Ecke 2 eintragen. Wie wir sehen werden, liegt das daran, dass die Ecken 1 und 2 die beiden einzigen Ecken mit ungeradem Grad sind.

Satz 5.2 *Sei $G = (V, E)$ zusammenhängend. Dann existiert genau dann von der Ecke $x \in V$ ein offener Euler-Zug zu der Ecke $y \in V \setminus \{x\}$, wenn x und y die einzigen Ecken mit ungeradem Grad sind.*

Beweis: Seien x und y die beiden einzigen Ecken im Graphen mit ungeradem Grad. Man definiere $G^* := (V \cup \{u\}, E \cup \{ux\} \cup \{uy\})$, wobei $u \notin V$. Dann ist G^* zusammenhängend, ferner besitzen alle Ecken in G^* (also auch x, y und u) geraden Grad. Daher gibt es wegen des gerade ebenen bewiesenen Satz 5.1 in G^* einen Euler-Zug C^*. Läßt man in diesem Euler-Zug die Ecke u und die inzidierenden Kanten ux und uy fort, so erhält man einen offenen Euler-Zug von x nach y. Umgekehrt gebe es einen offenen Euler-Zug von x nach $y \neq x$. Da in jeder anderen Ecke außer x, y ebenso viele Kanten anfangen wie enden, ist der Grad einer jeden solchen Ecke gerade. Aus dem entsprechenden Grund ist der Grad von x und y ungerade. □

Bemerkung: Wir wollen noch einmal bemerken, dass die beiden letzten Sätze entsprechend auch für Multigraphen gelten. Auch eine Übertragung auf *gerichtete Multigraphen* (hier sind die Kanten gerichtet, naheliegenderweise spricht man häufig statt von

(gerichteten) Kanten auch von *Pfeilen*) ist leicht möglich. Hier muss man für eine Ecke x zwischen dem In-Grad $d^+(x)$ (Anzahl der bei x endenden Pfeile) und dem Aus-Grad $d^-(x)$ (Anzahl der bei x startenden Pfeile) unterscheiden. Man kann dann z. B. zeigen, dass ein zusammenhängender gerichteter Graph genau dann einen (gerichteten) Euler-Zug besitzt, wenn $d^-(x) = d^+(x)$ für alle Ecken x. Man versuche, die entsprechenden Aussagen zu präzisieren und zu beweisen. □

Beispiel: Wir wollen *Mathematica* benutzen, um im erweiterten Haus des Nikolaus einen Euler-Zug zu bestimmen. Zunächst erhält man durch

```
adjdata={{2,6},{1,3,5,6},{2,4,6},{3,5},{2,3,4,6},{1,2,3,5}};
gr=FromAdjacencyLists[adjdata];
ShowLabeledGraph[gr]
```

den in Abbildung 3.20 angegebenen Graphen. Dieser Graph ist Eulersch, da man auf

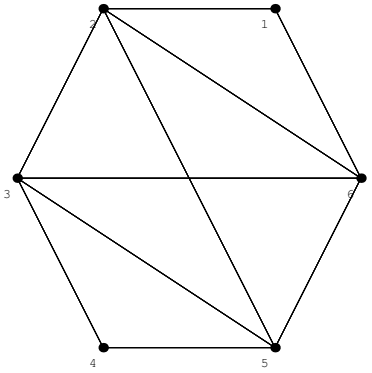


Abbildung 3.20: Das (erweiterte) Haus des Nikolaus

`EulerianQ[gr]` die Antwort `True` erhält, was wegen Satz 5.1 auch kein Wunder ist, da `ConnectedQ[gr]` ebenfalls die Antwort `True` ergibt und `DegreeSequence[gr]` den Output $\{4, 4, 4, 2, 2\}$ zur Folge hat, also anzeigt, dass jede Ecke geraden Grad besitzt. Nach `EulerianCycle[gr]` ergibt sich $\{4, 5, 6, 3, 5, 2, 6, 1, 2, 3, 4\}$ als eine Euler-Tour. □

3.6 Hamiltonsche Kreise

Ein Euler-Zug ist ein geschlossener Kantenzug in einem Graphen, der jede *Kante* des Graphen genau einmal enthält. Analog kann man nach Kreisen fragen, die alle *Ecken* genau einmal enthalten. Die entsprechenden Kreise heißen *Hamilton-Kreise*, und ein Graph heißt *Hamiltonsch*, falls er einen solchen Kreis enthält. Mit Satz 5.1 haben wir eine befriedigende notwendige und hinreichende Bedingung dafür angegeben, dass ein Graph Eulersch ist. Entsprechendes ist für Hamiltonsche Graphen viel schwerer. Ein triviales Beispiel eines Hamiltonschen Graphen ist der vollständige Graph K_n. Auch der vollständige bipartite Graph $K_{m,m}$ ist Hamiltonsch. Einige einfache notwendige Bedingungen bzw. hinreichende Bedingungen dafür, dass ein Graph Hamiltonsch ist, kommen in den Aufgaben 16 und 17 vor.

Zunächst wollen wir zwei auch historisch interessante Beispiele angeben.

Beispiel: In Abbildung 3.21 geben wir einen Graphen an, von dem Sir William Ha-

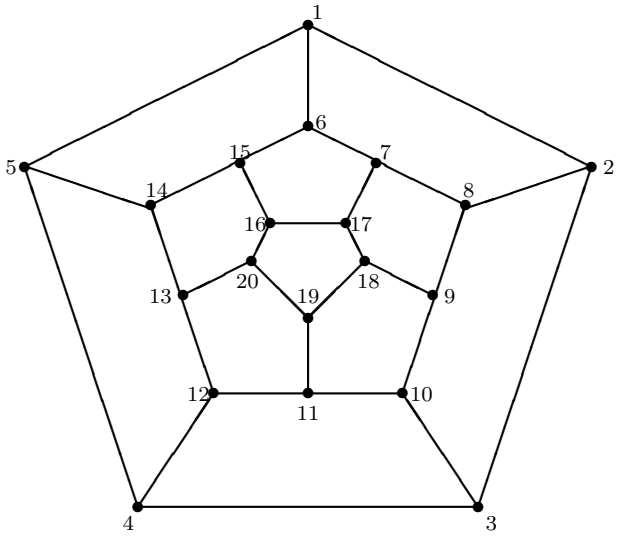


Abbildung 3.21: Ist dieser Graph Hamiltonsch?

milton im Jahre 1859 zeigte, dass er Hamiltonsch ist. Wer findet in diesem Graphen einen Hamilton-Kreis?

Zur Lösung benutzen wir *Mathematica*. Zunächst wird durch Angabe der entsprechenden Adjazenzliste ein Graph definiert.

```
adjdata={{2,5,6},{1,3,8},{2,4,10},{3,5,12},{1,4,14},{1,7,15},{6,8,17},
{2,7,9},{8,10,18},{3,9,11},{10,12,19},{4,11,13},{12,14,20},{5,13,15},
{6,14,16},{15,17,20},{7,16,18},{9,17,19},{11,18,20},{13,16,19}};
gr=FromAdjacencyLists[adjdata];
```

Mittels `ShowLabeledGraph[gr]` kann man sich den Graphen ansehen, siehe Abbildung 3.22. Im durch `gr=FromAdjacencyLists[adjdata]` definierten Graphen sind die Ecken per default auf einem Kreis angeordnet. Will man mit *Mathematica* ein ähnliches (wahscheinlich besseres) Bild wie in Abbildung 3.21 (dieses haben wir ziemlich mühsam im `picture`-environment von LaTeX erstellt), so hätte man in einer Liste noch die Koordinaten der Ecken angeben müssen. So erkennt man nicht unmittelbar, dass es sich hierbei um den selben Graphen wie in Abbildung 3.21 handelt. Nach `HamiltonianCycle[gr]` erhält man die Folge der Ecken

$$\{1, 2, 3, 4, 5, 14, 13, 12, 11, 10, 9, 8, 7, 17, 18, 19, 20, 16, 15, 6, 1\}$$

als Output. Den entsprechenden Hamilton-Kreis (bezüglich des ursprünglichen Graphen) geben wir in Abbildung 3.23 graphisch an. □

Beispiel: Beim Problem des Rösselsprungs auf dem Schachbrett geht es um folgendes: Mit dem Springer sollen alle 64 Felder des Schachbretts genau einmal in einem kontinuierlichen Zug erreicht und zum Ausgangsfeld zurückgekehrt werden. Die 64 Felder des

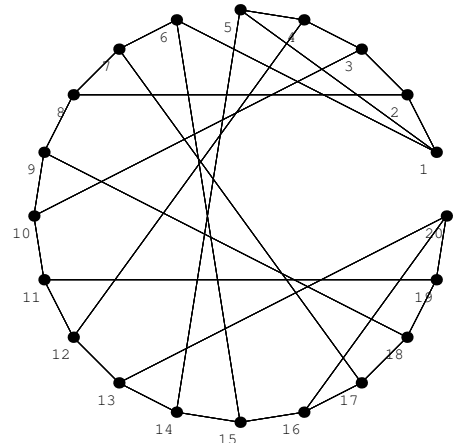


Abbildung 3.22: Derselbe Graph wie in Abbildung 3.21

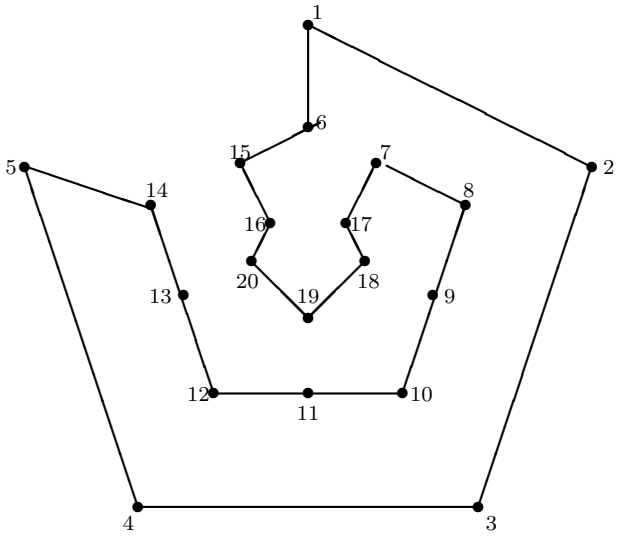


Abbildung 3.23: Ein Hamilton-Kreis zum Graphen in Abbildung 3.21

Schachbretts seien die Ecken eines Graphen. Weiter sind zwei Ecken genau dann durch eine Kante verbunden, wenn zwischen den entsprechenden Feldern ein Rösselsprung möglich ist. Das Rösselsprungproblem ist dann offenbar äquivalent dazu, in dem definierten Graphen einen Hamilton-Kreis zu bestimmen. Bei L. Volkmann ist eine auf Euler (1759) zurückgehende Lösung angegeben, die wir hier links reproduzieren:

58	43	60	37	52	41	62	35
49	46	57	42	61	36	53	40
44	59	48	51	38	55	34	63
47	50	45	56	33	64	39	54
22	7	32	1	24	13	18	15
31	2	23	6	19	16	27	12
8	21	4	29	10	25	14	17
3	30	9	20	5	28	11	26

63	22	15	40	1	42	59	18
14	39	64	21	60	17	2	43
37	62	23	16	41	4	19	58
24	13	38	61	20	57	44	3
11	36	25	52	29	46	5	56
26	51	12	33	8	55	30	45
35	10	49	28	53	32	47	6
50	27	34	9	48	7	54	31

In Abbildung 3.24 ist diese Lösung als Hamilton-Kreis in einem Graphen veranschaulicht. Noch erstaunlicher ist ein Springerkreis, der gleichzeitig ein magisches Quadrat

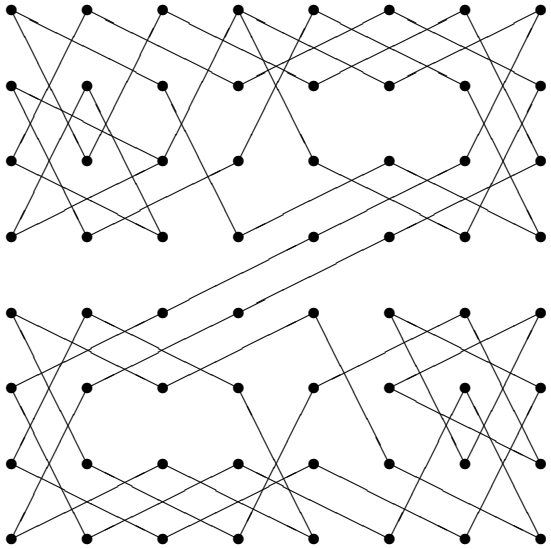


Abbildung 3.24: Eine Lösung des Rösselsprungproblems

ist (alle Zeilen- und Spaltensummen ergeben 260). Man findet ihn bei M. Löbbing, I. Wegener (1996)[11]. Er ist oben rechts angegeben. Bei Löbbing-Wegener findet man weitere interessante Bemerkungen zum Problem der Springerkreise, z. B. dass dieses in einer Folge von "Wetten, daß. . . ?" schon eine Rolle spielte. Man kann dort nachlesen:

- Der Kandidat behauptete, dass er für ein beliebiges Feld den Springer so über das Schachbrett bewegen kann, dass jedes Feld genau einmal erreicht wird. Und der Kandidat hat die Wette gewonnen. Diese Leistung hat bei vielen Menschen Bewunderung hervorgerufen. Immerhin gibt es 64 Startfelder, und 64 Springerwege enthalten 4032 Springerzüge. Diese Vorgehensweise ist jedoch sträflich naiv. Es genügt doch, einen Springerkreis auswendig zu lernen, da jeder Springerkreis für jedes Startfeld einen Springerweg enthält. Diese Gedächtnisleisung wird noch geringer, wenn ein Springerkreis mit vielen Symmetrien gewählt wird.

□

Die Hamilton-Kreise spielen im Zusammenhang mit dem Traveling Salesman Problem (TSP) in natürlicher Weise eine Rolle. Hier sind n Städte bzw. der vollständige Graph K_n gegeben, die Kanten tragen Kosten $c_{ij} \geq 0$, $1 \leq i, j \leq n$, (beim symmetrischen TSP ist $c_{ij} = c_{ji}$, beim metrischen TSP gilt darüberhinaus die Dreiecksungleichung, also $c_{ij} \leq c_{ik} + c_{kj}$), gesucht ist ein Hamilton-Kreis in K_n mit minimalen Kosten. Wir haben schon wiederholt bemerkt, dass es "sehr schwer" ist, eine optimale Lösung des

[11] M. Löbbing, I. Wegener (1996) "Knight moves–was macht der Springer allein auf dem Schachbrett?" In: Highlights aus der Informatik (I. Wegener, Hrsg.). Springer, Berlin-Heidelberg-New York.

TSP bei großem n zu bestimmen. Es gibt aber einige Heuristiken zur Konstruktion einer kostengünstigen Rundtour. Auf solche, die eine i. Allg. suboptimale Tour sukzessive aufbauen, wollen wir aber hier noch nicht eingehen. Die einfachste und naheliegendste ist ein Greedy-Algorithmus, bei dem man jeweils die nächste noch nicht besuchte Stadt anfährt (Nächster Nachbar), nur unwesentlich subtiler ist die Strategie, an beiden "Enden" nach dem nächsten Nachbarn zu sehen und von beiden den näheren zu wählen (Doppelter Nächster Nachbar). In ungünstigen Fällen können beide Verfahren wegen weiter Wege am Ende sehr schlecht sein.

Stattdessen wollen wir zwei globale Methoden angeben. Die erste Methode heißt *Minimum Spanning Tree Heuristik*. Sie besteht aus folgenden Schritten.

- Zu dem bewerteten vollständigen Graphen K_n bestimme man (z. B. mit dem Verfahren von Kruskal) einen minimalen aufspannenden Baum T.
- Verdopple alle Kanten in T. Hierdurch erhält man einen Eulerschen Multigraphen T_D. Sei C ein Euler-Zug in T_D.
- In C ist ein Hamilton-Kreis enthalten, den man durch Überspringen schon durchlaufener Ecken erhalten kann.

Beispiel: Wir reproduzieren ein Beispiel bei M. Aigner (1996, S. 141). Die 6 Städte mit ihren jeweiligen Entfernungen sind in der folgenden Tabelle angegeben.

	A	B	D	F	K	W
Aachen	–	91	80	259	70	121
Bonn	91	–	77	175	27	84
Düsseldorf	80	77	–	232	47	29
Frankfurt	259	175	232	–	189	236
Köln	70	27	47	189	–	55
Wuppertal	121	84	29	236	55	–

In Abbildung 3.25 geben wir zunächst den mit dem Verfahren von Kruskal gewonnenen minimalen aufspannenden Baum an (der Reihe nach werden die Kanten KB, DW, KD, AK (die Kante KW darf nicht genommen werden, weil sonst ein Kreis gebildet wäre), BF konstruiert). Wenn dann alle Kanten verdoppelt werden, gibt es natürlich

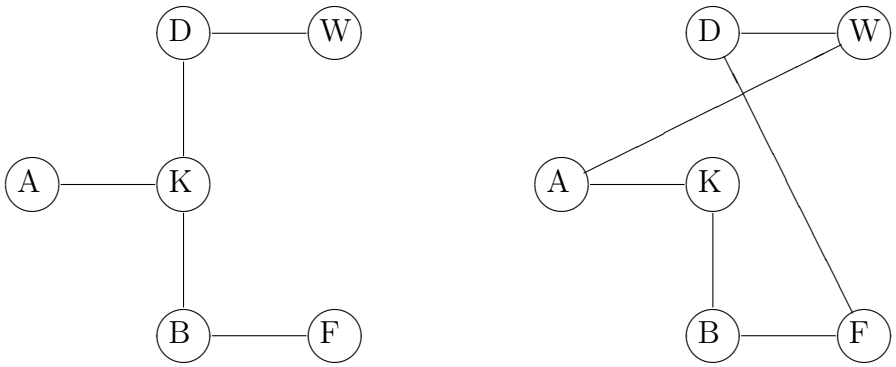


Abbildung 3.25: Minimum Spanning Tree Heuristik

eine Euler-Tour in dem so entstandenen Multigraphen. Z.B werden der Reihe nach die

Kanten KB, BF, FB, BK, KD, DW, WD, DK, KA, AK durchlaufen. Hieraus erhält man den Hamilton-Kreis K, B, F, D, W, A, K, den wir in Abbildung 3.25 rechts eingetragen haben. Die Länge dieser Tour ist 654. □

Eine weitere wichtige Heuristik ist die *Christofides*-Heuristik.

- Zu dem bewerteten vollständigen Graphen K_n bestimme man (z. B. mit dem Verfahren von Kruskal) einen minimalen aufspannenden Baum T.
- Sei U die Menge der Ecken ungeraden Grades in T. Wegen des Handshaking Lemmas 1.1 ist $|U| = 2m$ gerade. Zu dieser Eckenmenge bilde man den vollständigen (bewerteten) Graphen (mit $m(2m-1)$ Kanten). Unter einem *Maximum Matching* M in einem Graphen verstehen wir eine Menge maximal vieler nicht-inzidenter (also keine Ecken gemein habender) Kanten. Wegen $|U| = 2m$ besteht ein Maximum Matching hier aus m Kanten. Unter allen Maximum Matchings bestimme man eines mit minimalen Kosten. Diese Kanten M füge man zu T hinzu (ist eine Kante schon in T enthalten, so erhält man eine Mehrfachkante). Man erhält einen Eulerschen Multigraphen T_D (denn der Graph ist zusammenhängend und jede Ecke hat geraden Grad). Sei C ein Euler-Zug in T_D.
- In C ist ein Hamilton-Kreis enthalten, den man durch Überspringen schon durchlaufener Ecken erhalten kann.

Beispiel: Wir kehren zum letzten Beispiel zurück. Genau wie bei der Minimum Spanning Tree Heuristik wird zunächst der minimale aufspannende Baum T konstruiert. Dann ist $U = \{A, K, W, F\}$ die Menge der Ecken ungeraden Grades in T. In Abbildung 3.26 geben wir noch einmal den (den vollständigen Graphen) aufspannenden Baum T an, danach den zu U gehörenden vollständigen Graphen. Durch die Kantenmengen

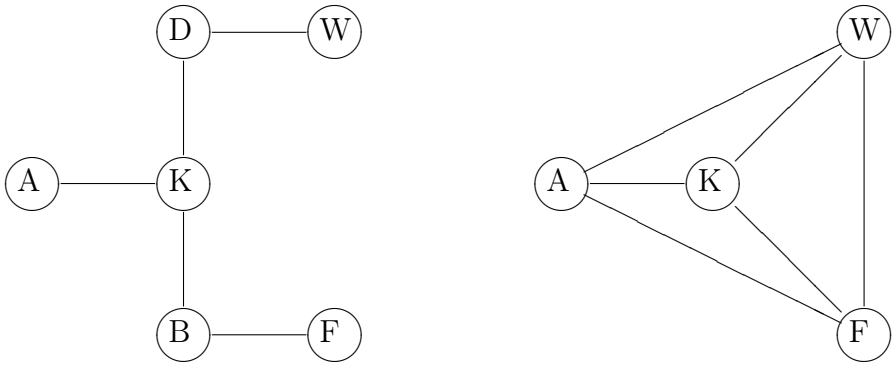


Abbildung 3.26: Christofides Heuristik

$\{AK, WF\}$, $\{AW, KF\}$ und $\{AF, KW\}$ erhält man Maximum Matchings mit Kosten 306, 310 und 314. Daher ist $M = \{AK, WF\}$ ein Maximum Matching mit minimalen Kosten. Diese beiden Kanten werden zu T hinzugenommen. Der hierdurch entstehende Multigraph T_D ist Eulersch. Durch BK, KA, AK, KD, DW, WF, FB ist ein Euler-Zug gegeben. Durch Überspringen schon benutzter Ecken erhalten wir hieraus den in Abbildung 3.27 angegebenen Hamilton-Kreis. Die Länge dieses Hamilton-Kreises ist 617. Es kann gezeigt werden, dass diese Rundtour unter allen 120 Hamilton-Kreisen optimal ist. □

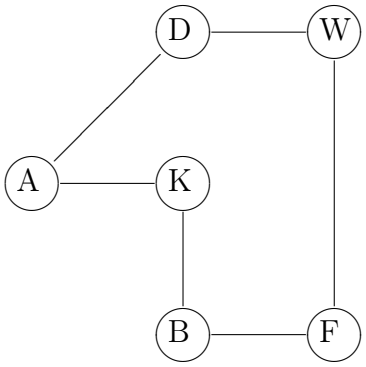


Abbildung 3.27: Hamilton-Kreis

Beispiel: Im durch «DiscreteMath‘Combinatorica‘ geladenen Mathematica-Zusatzpaket findet sich auch ein Befehl TravelingSalesman. Wir wollen diesen an dem obigen Beispiel ausprobieren. Nach

```
g={{0,91,80,259,70,121},{91,0,77,175,27,84},{80,77,0,232,47,29},
{259,175,232,0,189,236},{70,27,47,189,0,55},{121,84,29,236,55,0}};
l=Vertices[CompleteGraph[6]];
gr=Graph[g,l];
TravelingSalesman[gr]
```

erhält man $\{1, 3, 6, 4, 2, 5, 1\}$ als Output, also die Tour

Aachen → Düsseldorf → Wuppertal → Frankfurt → Bonn → Köln → Aachen.

Dies stimmt mit der Tour in Abbildung 3.27 überein. Durch

```
TravelingSalesmanBounds[gr]
```

erhält man untere und obere Schranken für die optimalen Kosten beim TSP. In unserem Fall wird $\{357, 654\}$ ausgegeben. □

3.7 Gerichtete Graphen (Digraphen)

Eine Kante in einem Graphen ist ein ungeordnetes Paar von Ecken. Dagegen ist ein *gerichteter Graph* oder *Digraph* ein Paar $(\mathcal{N}, \mathcal{A})$, wobei $\mathcal{N}$ die endliche Menge der *Knoten* (oder auch Ecken) und $\mathcal{A} \subset \mathcal{N} \times \mathcal{N}$ die Menge der *Pfeile* (oder auch Bögen, gerichtete Kanten) ist. Jeder Pfeil ist also ein *geordnetes* Paar von Knoten. Stellt man sich einen Graphen als ein Straßennetz zwischen Orten vor, so kann man durch den Übergang zu Digraphen auch Einbahnstraßen modellieren. Ist $a = (x, y)$ ein Pfeil, so heißt $x = a^-$ der Anfangsknoten, $y = a^+$ der Endknoten von $a \in \mathcal{A}$. Jedes geordnete Paar (x, y) von Knoten kommt höchstens einmal in $\mathcal{A}$ vor (also keine Mehrfachpfeile), ferner wird $x \neq y$ verlangt (keine Schleifen). Andernfalls spricht man von gerichteten Multigraphen. Für zwei verschiedene Knoten können aber natürlich sowohl (x, y) als auch (y, x) Pfeile in $\mathcal{A}$ sein. Für einen Knoten $x \in \mathcal{N}$ heißt $d^+(x) := |\{a \in \mathcal{A} : a^+ = x\}|$ der *In-Grad* von x, also die Anzahl der in x endenden Pfeile. Entsprechend heißt

$d^-(x) := |\{a \in \mathcal{A} : a^- = x\}|$, also die Anzahl der bei x startenden Pfeile, der *Aus-Grad* von x. Ein Knoten heißt eine *Quelle*, wenn sein In-Grad verschwindet, entsprechend heißt ein Knoten eine *Senke*, wenn sein Aus-Grad gleich Null ist. Einige Begriffe übertragen sich direkt von ungerichteten auf gerichtete Graphen. So dürfte klar sein, was man unter einem *gerichteten Weg* oder einem *gerichteten Kreis* versteht. Man nennt einen gerichteten Graphen *stark zusammenhängend*, wenn es zwischen je zwei Knoten einen gerichteten Weg gibt, dagegen heißt er *(schwach) zusammenhängend*, wenn der zugrundeliegende ungerichtete Graph (ersetze Pfeile durch ungerichtete Kanten) zusammenhängend ist. Auch die Begriffe (gerichteter) Euler-Zug und Eulerscher Digraph sowie (gerichteter) Hamilton-Kreis und Hamiltonscher Digraph können in naheliegender Weise erklärt werden. Wir hatten schon erwähnt, dass ein Digraph genau dann Eulersch ist, wenn für jeden Knoten In-Grad und Aus-Grad übereinstimmen.

Sind $\mathcal{N} = \{x_1, \ldots, x_n\}$ und $\mathcal{A} = \{a_1, \ldots, a_m\}$ die Knoten bzw. Pfeile (in einer gewissen Nummerierung) in dem gerichteten Graphen $(\mathcal{N}, \mathcal{A})$, so heißt die durch

$$b_{ij} := \begin{cases} 1 & \text{falls} \quad x_i = a_j^+, \\ -1 & \text{falls} \quad x_i = a_j^-, \\ 0 & \text{falls} \quad x_i \notin a_j, \end{cases}$$

definierte Matrix $B = (b_{ij}) \in \mathbb{R}^{n \times m}$ die *Inzidenzmatrix*[12] des Digraphen $(\mathcal{N}, \mathcal{A})$. Dagegen heißt die Matrix $A = (a_{ij}) \in \mathbb{R}^{n \times n}$, die durch

$$a_{ij} := \begin{cases} 1 & \text{falls} \quad (x_i, x_j) \in \mathcal{A}, \\ 0 & \text{sonst} \end{cases}$$

definiert ist, die *Adjazenzmatrix* des Digraphen $(\mathcal{N}, \mathcal{A})$. Einen *gewichteten* Digraphen erhält man, indem man (für $(x_i, x_j) \in \mathcal{A}$) in der (i, j)-Position der Adjazenzmatrix das entsprechende Gewicht einträgt.

Beispiel: Wir wollen einen ersten Eindruck darüber gewinnen, welche Möglichkeiten *Mathematica* zur Darstellung und Bearbeitung von gerichteten Graphen zur Verfügung stellt. Z. B. haben wir den in Abbildung 3.28 dargestellten gerichteten Graphen durch

```
adj={{6,7},{1},{2,4},{5},{6,8},{7},{2,8},{3,4}};
gra=FromAdjacencyLists[adj];
dirgra=Show[ShowGraph[gra,Directed],ShowLabeledGraph[gra]]
```

erhalten[13]. Die Adjazenzmatrix ist im Gegensatz zu ungerichteten Graphen i. Allg. unsymmetrisch. Für den in 3.28 abgebildeten Graphen erhält man z. B. durch `Edges[gra]`

[12] Hier muss man sehr aufpassen. Obige Definition stimmt mit der bei M. Aigner (1996, S. 99) überein. Hiernach ist der (i, j)-Eintrag der Inzidenzmatrix 1 bzw. -1 , wenn der i-te Knoten *Endpunkt* bzw. *Anfangspunkt* des j-ten Pfeils ist, 0 in allen anderen Fällen. Gelegentlich (und daran werden wir uns später halten) nimmt man genau das Negative als Einträge (siehe z. B. die Definition der Knoten-Pfeil-Inzidenzmatrix bei C. H. Papadimitriou, K. Steiglitz (1982, S. 75)).

[13] Leider wissen wir nicht, wie Größe und Position der Pfeile verändert werden können. Ferner scheint eine Nummerierung der Knoten eines gerichteten Graphen mit Hilfe von `ShowLabeledGraph` nicht möglich zu sein, weshalb die obige Hilfskonstruktion gewählt wurde.

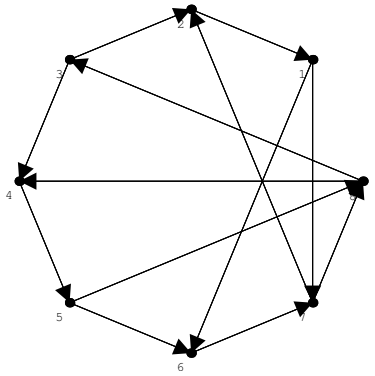


Abbildung 3.28: Ein gerichteter Graph

die Adjazenzmatrix

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

Als Antwort auf `ConnectedQ[gra,Undirected]` erhält man `True`, obiger Graph ist also (schwach) zusammenhängend. Er ist sogar stark zusammenhängend (was man nicht sofort sieht), denn auch auf `ConnectedQ[gra,Directed]` erhält man die Antwort `True`. Der angegebene Digraph ist nicht Eulersch (z. B. ist $d^-(1) \neq d^+(1)$), was man auch durch die Antwort `False` auf die Frage `EulerianQ[gra,Directed]` erfahren kann[14] □

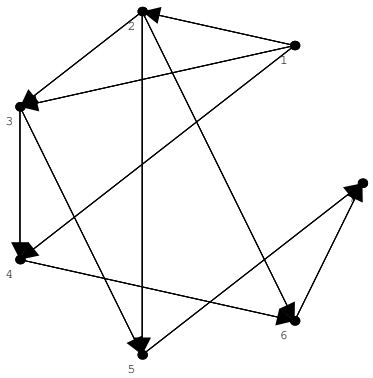
Das Problem der kürzesten (gerichteten) Wege stellt sich natürlich ganz entsprechend in einem bewerteten Digraphen. Das Verfahren von Dijkstra lässt sich in kanonischer Weise erweitern. Wir machen uns dies durch ein Beispiel klar.

Beispiel: Gegeben sei der in 3.29 angegebene Digraph, dessen (bewertete) Adjazenzmatrix man ebenfalls dort findet. Den gerichteten Graphen haben wir durch

```
gewadj={{0,7,13,28,0,0,0},{0,0,4,0,25,10,0},{0,0,0,5,6,0,0},
{0,0,0,0,0,3,0},{0,0,0,0,0,0,5},{0,0,0,0,0,0,12},{0,0,0,0,0,0,0}};
gra=Graph[gewadj,Vertices[CompleteGraph[7]]];
```

erzeugt. Es seien die vom Knoten 1 ausgehenden kürzesten Wege zu bestimmen. Der Knoten 1 erhält den Wert 0. Danach wird der Pfeil $(1, 2)$ erzeugt, der Knoten 2 bekommt den Wert 7. Weiter kommt $(2, 3)$ dazu, 3 wird mit 11 bewertet. Weiter erhält man $(3, 4)$, Knoten 4 bekommt den Wert 16. Im nächsten Schritt wird der Pfeil $(2, 6)$ aufgenommen, der Knoten 6 mit 17 bewertet. Danach $(3, 5)$, 19 wird der Wert

[14] Eine entsprechende Möglichkeit festzustellen, ob ein Digraph Hamiltonsch ist, scheint es im Paket `DiscreteMath‘Combinatorica‘` nicht zu geben.



$$A = \begin{pmatrix} 0 & 7 & 13 & 28 & 0 & 0 & 0 \\ 0 & 0 & 4 & 0 & 25 & 10 & 0 \\ 0 & 0 & 0 & 5 & 6 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 3 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 5 \\ 0 & 0 & 0 & 0 & 0 & 0 & 12 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

Abbildung 3.29: Ein gewichteter Digraph und seine Adjazenzmatrix

von 5. Zum Schluss kommt der Pfeil $(5, 7)$ hinzu, 7 hat den Wert 24. Definiert man `tree=ShortestPathSpanningTree[gra,1]`, so erkennt man durch die mit `True` beantwortete Frage `UndirectedQ[tree]`, dass der entsprechende Baum ungerichtet ist (was eigentlich auch kein Problem ist). In Abbildung 3.30 links ist er abgebildet, rechts findet sich die gerichtete Version mit den jeweiligen Kosten. Als Antwort auf `ShortestPath[gra,1,7]` erhält man $\{1, 2, 3, 5, 7\}$. □

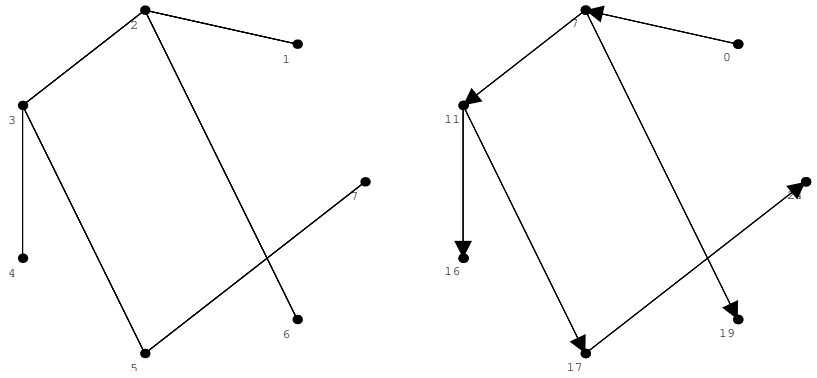


Abbildung 3.30: Vom Knoten 1 ausgehende kürzeste gerichtete Wege

Wichtige weitere Optimierungsaufgaben auf Digraphen sind *Netzwerkflussprobleme*, auf die wir ausführlich im nächsten Kapitel eingehen werden. Wir wollen hier nur an einem Beispiel die (begrenzten) Fähigkeiten von *Mathematica* zur Behandlung von Netzwerkflussproblemen erkunden. Dieses bezieht sich auf das Problem des maximalen Flusses. Unter Wahrung der durch die Gewichte auf den Pfeilen angegebenen Kapazitätsgrenzen soll möglichst viel von der Quelle zur Senke gelangen, wobei außer in der Quelle und der Senke in jedem Knoten die Flussbedingung zu gelten hat, dass nämlich alles was ankommt auch wegtransportiert wird. Das wichtigste theoretische Ergebnis zum Maximalflussproblem ist das *Max Flow-Min Cut* Theorem von Ford-Fulkerson,

von denen auch das wichtigste Verfahren zur Lösung stammt. Hierauf gehen wir im nächsten Kapitel ausführlich ein.

Beispiel: In Abbildung 3.31 geben wir einen gewichteten Digraphen an. Mit Hilfe

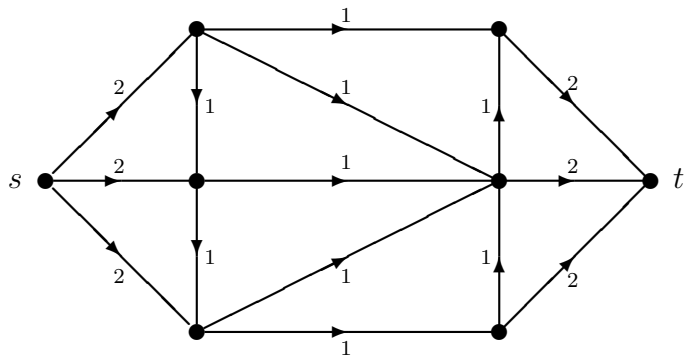


Abbildung 3.31: Ein Netzwerk mit 8 Knoten und 14 Pfeilen

von *Mathematica* soll ein maximaler Fluss bestimmt werden. Durch Angabe der Adjazenzmatrix definieren wir zunächst den gewichteten Digraphen. Da es uns auf die Einbettung in der Ebene nicht ankommt, wählen wir die von `Cycle[8]`. Dies geschieht durch

```
adj={{0,2,2,2,0,0,0,0},{0,0,1,0,1,1,0,0},{0,0,0,1,0,1,0,0},
{0,0,0,0,0,1,1,0},{0,0,0,0,0,0,0,2},{0,0,0,0,1,0,0,2},
{0,0,0,0,0,1,0,2},{0,0,0,0,0,0,0,0}};
gra=Graph[adj,Vertices[Cycle[8]]];
```

Hierbei haben wir die Knoten von links nach rechts und oben nach unten nummeriert. Die Quelle ist der Knoten 1, die Senke der Knoten 8. Will man den *Wert* des maximalen Flusses erhalten, so gibt man `NetworkFlow[gra,1,8]` ein. Die Antwort ist, wie zu erwarten, 5. Zur Berechnung des maximalen Flusses selber gibt es den Befehl `NetworkFlowEdges`. Durch `NetworkFlowEdges[gra,1,8]` wird eine Adjazenzmatrix ausgegeben, aus der die Verteilung des maximalen Flusses zu ersehen ist. In unserem Falle erhält man

$$\begin{pmatrix} 0 & 2 & 1 & 2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 2 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}.$$

Das entspricht dem in Abbildung 3.32 angegebenen maximalen Fluss. □

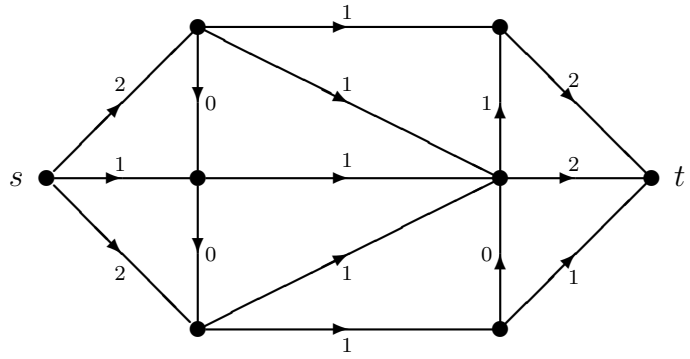


Abbildung 3.32: Ein maximaler Fluss

3.8 Matchings

Beispiel: Sei $U = \{u_1, \ldots, u_m\}$ eine Menge von Damen, $W = \{w_1, \ldots, w_n\}$ eine Menge von Herren. Wir sagen, ein Paar $(u, w) \in U \times W$ *mag* sich, wenn beide für eine gegenseitige langfristige Beziehung, z. B. eine Heirat, sind. Unter welchen Bedingungen gibt es zu jeder Dame $u \in U$ genau einen Herrn $w \in W$ derart, dass das Paar (u, w) sich mag? Hierbei darf natürlich jeder Herr höchstens eine Dame als Partnerin erhalten. □

Wir formulieren dieses Problem als eine graphentheoretische Aufgabe. Hierzu führen wir einen bipartiten Graphen $G = (V, E)$ mit der Eckenmenge $V := U \cup W$ der Damen und Herren ein. Wenn das Paar $(u, w) \in U \times W$ sich mag, so sei $e = uv$ eine Kante im Graphen. Ein "zulässiger Heiratsplan" (Matching) ist eine Teilmenge $F \subset E$ der Kantenmenge mit der Eigenschaft, dass zwei verschiedene Elemente von F keine Ecke gemeinsam haben, es also zu keiner Bigamie kommt. Allgemeiner formulieren wir:

Definition 8.1 Ein *Matching* in einem Graphen $G = (V, E)$ ist eine Kantenmenge $F \subset E$ mit der Eigenschaft, dass zwei verschiedene Elemente von F keine Ecke gemeinsam haben. Die *Matching-Zahl* $m(G)$ ist die Anzahl der Kanten in einem maximal großen Matching. Ein Matching F heißt ein *Maximum-Matching*, wenn $|F| = m(G)$. Ein Matching F heißt *gesättigt*, wenn es kein Matching F' mit $F \subset F'$ und $F \neq F'$ gibt. Weiter heißt ein Matching F *perfekt*, wenn jede Ecke aus V Anfangs- oder Endecke (genau einer) der Kanten aus F ist.

Uns wird vor allem das maximale bipartite Matchingproblem interessieren, in welchem es darauf ankommt, in einem bipartiten Graphen ein Maximum-Matching zu bestimmen. Allerdings gibt es im Zusatzpaket `DiscreteMath‘Combinatorica‘` von *Mathematica* auch den Befehl `MaximalMatching`, mit dem ein Matching in einem Graphen ausgegeben wird, das zwar nicht notwendigerweise ein Maximum-Matching ist, aber jedenfalls die Eigenschaft hat, dass man bei Hinzunahme einer Kante kein Matching mehr hat. Als Antwort auf `MaximalMatching[CompleteGraph[7]]` erhält man z. B. $\{\{1, 2\}, \{3, 4\}, \{5, 6\}\}$ und in der Tat ist die Matching-Zahl des vollständigen Graphen mit 7 Ecken offenbar 3.

Beispiel: In Abbildung 3.33 links geben wir einen Graphen an, von dem ein Maximum-

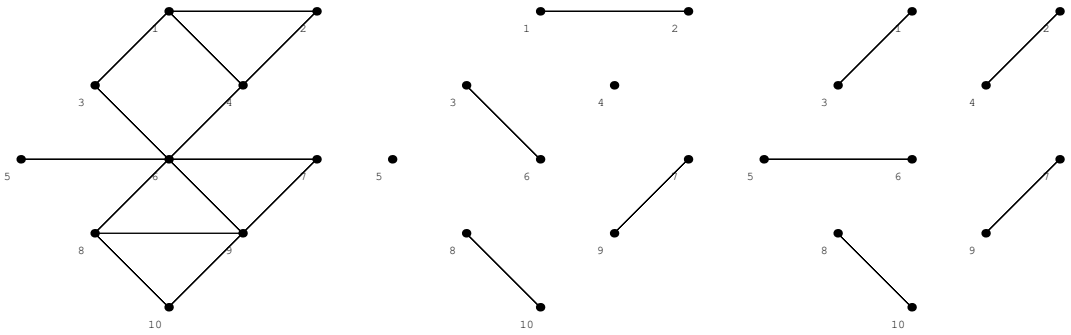


Abbildung 3.33: Ein Graph und sein Maximum-Matching

Matching bestimmt werden soll[15]. Als Resultat von `MaximalMatching` erhalten wir die Kanten $\{1,2\}$, $\{3,6\}$, $\{7,9\}$ und $\{8,10\}$, was wir in Abbildung 3.33 in der Mitte darstellen. Dies ist aber kein Maximum-Matching, denn es gibt auch ein aus 5 Kanten bestehendes Matching, welches wir in Abbildung 3.33 rechts darstellen. □

Wir werden uns nur noch mit Matching in einem bipartiten Graphen beschäftigen.

Als erstes Ergebnis formulieren und beweisen wir den auf P. Hall (1935) zurückgehenden *Heiratssatz*.

Satz 8.2 *Gegeben sei ein bipartiter Graph $G = (U \cup W, E)$. Für $T \subset U$ sei*

$$N(T) := \{w \in W : \text{Es existiert ein } u \in T \text{ mit } uw \in E\}.$$

(Die Menge $N(T) \subset W$ kann als Menge der Nachbarn von $T \subset U$ aufgefasst werden.) Dann ist $m(G) = |U|$ genau dann, wenn $|T| \leq |N(T)|$ für alle $T \subset U$.

Beweis: Angenommen[16], es existiert ein $T \subset U$ mit $|T| > |N(T)|$. Dann können nicht alle Ecken aus T gematcht werden oder anders ausgedrückt: Die Anzahl $|T|$ der Damen in T ist größer als die Anzahl der möglichen Heiratskandidaten, so dass kein Matching aller Damen möglich ist. Ist also $m(G) = |U|$, so ist notwendig $|T| \leq |N(T)|$ für alle $T \subset U$.

Nun nehmen wir an, es sei $|T| \leq |N(T)|$ für alle $T \subset U$. Sei $F \subset E$ ein Matching mit $|F| < |U|$. Wir werden zeigen, dass F kein Maximum-Matching ist, da es ein Matching F' mit $|F'| = |F|+1$ gibt. Wegen $|F| < |U|$ gibt es eine "ungematchte" Ecke $u_0 \in U$, also eine Ecke, die auf keiner der Kanten aus F liegt. Wegen $|N(\{u_0\})| \geq |\{u_0\}| = 1$ gibt es einen Nachbarn $w_1 \in W$ (es ist also $u_0w_1 \in E$). Ist w_1 nicht gematcht in F, liegt also w_1 auf keiner Kante aus F, so setze man $F' := F \cup \{u_0w_1\}$, wodurch ein Matching F' mit $|F'| = |F|+1$ gewonnen ist. Daher nehmen wir jetzt an, es sei w_1 durch F gematcht, d. h. es existiere ein $u_1 \in U \setminus \{u_0\}$ mit $u_1w_1 \in F$. Nach Voraussetzung ist $|N(\{u_0, u_1\})| \geq |\{u_0, u_1\}| = 2$, d. h. es gibt mindestens zwei Nachbarn von $\{u_0, u_1\}$, also neben w_1 noch eine weitere Ecke $w_2 \in W \setminus \{w_1\}$ mit der Eigenschaft, dass u_0w_2 oder u_1w_2 eine Kante in G ist. Ist w_2 nicht gematcht, dann: STOP (Wir werden zeigen, dass es dann ein

[15] Siehe M. Aigner (1996, S. 124).

[16] Wir folgen hier dem Beweis bei M. Aigner (1996, S. 120), weitere Beweise findet man z. B. bei B. Bollobás (1998, S. 77 ff.) und R. Diestel (1998, S. 31).

Matching F' mit $|F'| = |F| + 1$ gibt). Wenn w_2 dagegen gematcht ist, so gibt es ein $u_2 \in U \setminus \{u_0, u_1\}$ mit $u_2w_2 \in F$. In dieser Weise können wir fortfahren und erhalten schließlich eine ungematchte Ecke $w_r \in W$. Die Elemente in $\{u_0, u_1, \ldots, u_{r-1}\} \subset U$ bzw. $\{w_1, w_2, \ldots, w_r\} \subset W$ sind paarweise verschieden und haben die folgende Eigenschaft: Es ist $u_iw_i \in F$, $i = 1, \ldots, r-1$, ferner ist jedes $w_j \in \{w_1, \ldots, w_r\}$ benachbart zu mindestens einer Ecke $u_i \in \{u_0, \ldots, u_{j-1}\}$.

Man bestimme nun, von w_r rückwärtsgehend, einen Weg P von w_r nach u_0. Zu w_r gibt es ein $u_a \in \{u_0, \ldots, u_{r-1}\}$ mit $w_ru_a \in E \setminus F$ (denn einerseits ist w_r benachbart zu einer Ecke aus $\{u_0, \ldots, u_{r-1}\}$, andererseits ist w_r nicht durch F gematcht). Nach Konstruktion ist $u_aw_a \in F$. Zu w_a gibt es ein benachbartes $u_b \in \{u_0, \ldots, u_{a-1}\}$, so dass also $w_au_b \in E$ eine Kante in G ist. Weiter ist $w_au_b \notin F$, da w_a ja schon mit u_a verheiratet ist. Insgesamt erhält man einen Weg

$$P = w_r, u_a, w_a, u_b, w_b, \ldots, u_h, w_h, u_0$$

mit $r > a > b > \cdots > h > 0$. Hierbei ist $u_aw_a, u_bw_b, \ldots, u_hw_h \in F$, während $w_ru_a, w_au_b, \ldots, w_hu_0 \in E \setminus F$. Nun setze man

$$F' := (F \setminus \{u_aw_a, u_bw_b, \ldots, u_hw_h\}) \cup \{w_ru_a, w_au_b, \ldots, w_hu_0\}.$$

Dann ist F' ein Matching mit $|F'| = |F| + 1$. Damit ist die Behauptung bewiesen. □

Bemerkung: Der Heiratssatz 8.2 gibt eine notwendige und hinreichende Bedingung dafür an, dass alle Damen in U einen Herrn aus W finden derart, dass die Paare sich mögen und keine Dame und kein Herr Bigamist wird. Die Bedingung besteht darin, dass für jede Teilmenge der Damen ihre Anzahl nicht größer ist als die Anzahl derjenigen Herren, mit denen für diese Damen eine langfristige Beziehung möglich ist. □

Wir wollen eine Reihe von Folgerungen und Varianten des Heiratssatzes angeben und beweisen.

Satz 8.3 *Sei der bipartite Graph $G = (U \cup W, E)$ gegeben. Für $T \subset U$ sei wieder $N(T) \subset W$ die Menge der Nachbarn von T. Dann ist die Matching-Zahl gegeben durch*

$$m(G) = |U| - \max_{T \subset U}(|T| - |N(T)|).$$

Beweis: Sei $\delta := \max_{T \subset U}(|T| - |N(T)|)$. Man beachte, dass $\delta \geq 0$, da $\delta \geq |\emptyset| - |N(\emptyset)| = 0$. Sei T eine Teilmenge von U mit $\delta = |T| - |N(T)|$. In T gibt es $\delta = |T| - |N(T)|$ Damen ohne geeigneten Partner, d. h. es ist $m(G) \leq |U| - \delta$. Wir haben zu zeigen, dass hier Gleichheit gilt. Sei D eine beliebige Menge mit $|D| = \delta$ und $W \cap D = \emptyset$. Man definiere den bipartiten Graphen $G^* := (U \cup (W \cup D), E^*)$, wobei $E^* := E \cup \{ud : u \in U, d \in D\}$. Für eine beliebige Teilmenge $T \subset U$ ist dann $N^*(T) = N(T) \cup D$ die Menge aller Nachbarn von Elementen aus T in G^*. Für ein beliebiges $T \subset U$ ist daher $|N^*(T)| = |N(T)| + \delta \geq |T|$, d. h. in G^* gibt es wegen des Heiratssatzes 8.2 ein Matching $F^* \subset E^*$ mit $|F^*| = |U|$. Aus dem Matching F^* in G^* erhalte man ein Matching F, indem man alle Kanten, die nach D führen, entfernt. Deren Anzahl ist höchstens gleich der Anzahl der Elemente in D, also δ, und folglich $|F| \geq |U| - \delta$. Folglich ist $m(G) \geq |U| - \delta$ und insgesamt $m(G) = |U| - \delta$. Damit ist der Satz bewiesen. □

Eine andere Interpretation von Satz 8.3 basiert auf der folgenden Definition.

Definition 8.4 Eine Eckenmenge $D \subset U \cup W$ heißt *Träger* des bipartiten Graphen $G = (U \cup W, E)$, falls D jede Kante aus G trifft bzw. bedeckt. Genauer heißt dies: Ist $e \in E$ beliebig, so existiert ein $d \in D$, welches entweder Anfangs- oder Endecke von e ist.

Der folgende Satz (er geht auf D. König (1931) zurück) hat eine ähnliche Struktur wie der starke Dualitätssatz der linearen Optimierung (oder das Max Flow-Min Cut Theorem von Ford-Fulkerson).

Satz 8.5 *Für einen bipartiten Graphen $G = (U \cup W, E)$ gilt*

$$\max\{|F| : F \subset E \textit{ Matching}\} = \min\{|D| : D \subset U \cup W \textit{ Träger}\}.$$

Beweis: Wie so oft ist bei dem Nachweis, dass eine gewisse Gleichung gilt, der Beweis einer Ungleichung mehr oder weniger trivial, der Beweis der umgekehrten Ungleichung aber nicht trivial (man denke etwa an die Aussage des starken Dualitätssatzes). Das ist hier genau so.

Sei $F \subset E$ ein Matching und $D \subset U \cup W$ ein Träger von G. Da Kanten aus F keine gemeinsame Ecke haben, andererseits die Ecken auf D alle Kanten bedecken, ist trivialerweise $|D| \geq |F|$ und daher

$$\max\{|F| : F \subset E \text{ Matching}\} \leq \min\{|D| : D \subset U \cup W \text{ Träger}\}.$$

Wegen Satz 8.3 existiert ein $T \subset U$ mit

$$m(G) = \max\{|F| : F \subset E \text{ Matching}\} = |U| - |T| + |N(T)| = |U \setminus T| + |N(T)|.$$

Wir überlegen uns, dass $D := (U \setminus T) \cup N(T)$ ein Träger von G ist. Hierzu sei $e = uw \in E$ mit $(u, w) \in U \times W$ eine beliebige Kante in G. Ist $u \in U \setminus T$, so wird e natürlich von D bedeckt, während für $u \in T$ nach Definition $w \in N(T)$ ein Nachbar von T ist. In jedem Fall ist D ein Träger von G. Wegen $|D| = |U \setminus T| + |N(T)|$ ist daher auch

$$\max\{|F| : F \subset E \text{ Matching}\} \geq \min\{|D| : D \subset U \cup W \text{ Träger}\}.$$

Insgesamt ist der Satz bewiesen. □

Wir wollen gleich noch ein wenig auf die Berechnung eines Maximum-Matching in einem bipartiten Graphen eingehen. Zunächst wollen wir aber die Möglichkeiten von *Mathematica* bei dieser Fragestellung an einem Beispiel untersuchen.

Beispiel: Gegeben sei der in Abbildung 3.34 links dargestellte bipartite Graph. Wendet man auf den Graphen `BipartiteMatching` an, so erhält man eine Liste von Kanten, die in Abbildung 3.34 rechts dargestellt sind. Wendet man `MinimumVertexCover` auf den in 3.34 links angegebenen bipartiten Graphen an, so erhält man die sechs Ecken $\{5, 6, A, B, E, G\}$. Der Träger mit einer minimalen Zahl von Elementen ist i. Allg. nicht eindeutig bestimmt. Z. B. ist auch durch $\{5, 6, 7, A, B, E\}$ ein Träger mit einer minimalen Zahl von Elementen gegeben, wie man auch mit Hilfe von `VertexCoverQ` feststellen kann. Die hinreichende Bedingung im Heiratssatz ist nicht erfüllt, da $N(\{1, 2, 4\}) =$

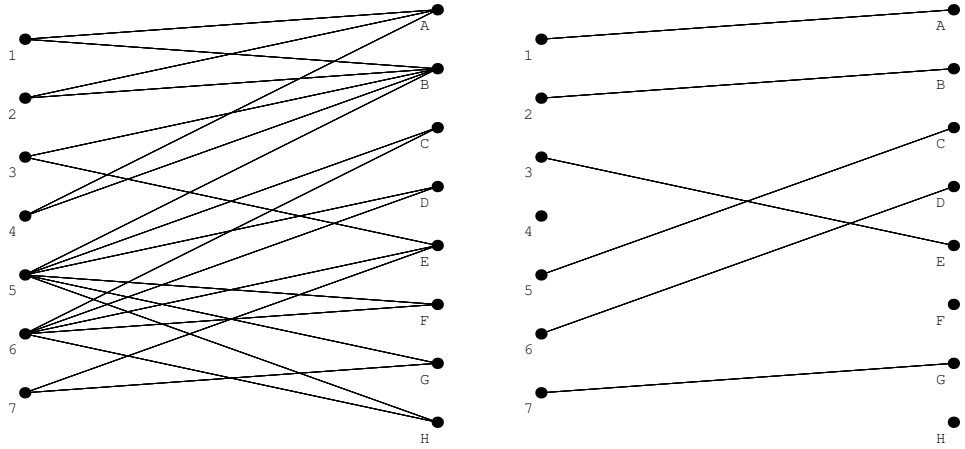


Abbildung 3.34: Maximum-Matching in einem bipartiten Graphen

$\{A, B\}$. Es ist daher kein Wunder, dass eine der sieben Damen bei dem angegebenen Maximum-Matching keinen Partner erhält. □

Nun kommen wir zur Konstruktion optimaler Matchings, z. B. von Maximum-Matchings. Die folgende Definition und der anschließende Satz beziehen sich auf Graphen, die nicht notwendig bipartit sind.

Definition 8.6 Sei $G = (V, E)$ ein Graph und $F \subset E$ ein Matching. Ist die Ecke $u \in V$ Anfangs- oder Endecke einer Kante aus F, so heißt sie *F-saturiert*, andernfalls *F-unsaturiert*. Ein *F-alternierender Weg* P ist ein Weg in G, der abwechselnd Kanten aus F und $E \setminus F$ verwendet, wobei die beiden Endecken von P jeweils F-unsaturiert sind.

Die Anzahl der Kanten in einem F-alternierenden Weg ist stets ungerade, ferner liegt stets eine gerade Anzahl von Ecken auf einem F-alternierenden Weg.

Beispiel: Gegeben sei der in Abbildung 3.35 dargestellte Graph. Durch etwa fettere

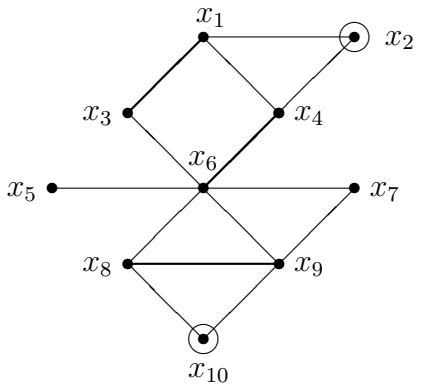


Abbildung 3.35: Ein Matching F und ein F-alternierender Weg

Linien haben wir ein Matching $F = \{x_1x_3, x_4x_6, x_8x_9\}$ eingetragen. Durch

$$P = x_2, x_4, x_6, x_8, x_9, x_{10}$$

ist ein F-alternierender Weg von x_2 nach x_{10} gegeben. □

Satz 8.7 *In einem Graphen $G = (V, E)$ ist ein Matching $F \subset E$ genau dann ein Maximum-Matching, wenn es keinen F-alternierenden Weg in G gibt.*

Beweis: Sei[17] F ein Matching und $x_0, x_1, \ldots, x_{2k-1}, x_{2k}, x_{2k+1}$ ein F-alternierender Weg. Dann ist

$$F' := (F \setminus \{x_1x_2, x_3x_4, \ldots, x_{2k-1}x_{2k}\}) \cup \{x_0x_1, x_2x_3, \ldots, x_{2k}x_{2k+1}\}$$

ein Matching mit $|F'| = |F| + 1$. Ist also F ein Maximum-Matching, so existiert kein F-alternierender Weg.

Jetzt nehmen wir an, F sei kein Maximum-Matching und zeigen, dass es dann einen F alternierenden Weg gibt. Es existiere also ein Matching F' mit $|F'| > |F|$. Wir setzen $E' := (F \setminus F') \cup (F' \setminus F)$ (symmetrische Differenz von F und F'). Sei $G' = (V', E')$ der von der Kantenmenge E' aufgespannte Untergraph von G. Kanten aus G' liegen also entweder in F oder in F'. Jede Ecke in G' hat den Grad 1 oder 2, denn sie liegt höchstens auf genau einer Kante aus F und aus F'. Wir betrachten eine Zusammenhangskomponente von G'. Eine soche Komponente ist ein (geschlossener oder nicht geschlossener) Weg, bei dem sich die Kanten von F und F' abwechseln (da nämlich beide Matchings sind, können nicht zwei Kanten aus F oder F' ein und dieselbe Ecke enthalten). Da $|F'| > |F|$, gibt es unter diesen Weg-Komponenten einen Weg P, der mit einer Kante aus F' beginnt und endet, insbesondere also nicht geschlossen ist. Da die Endecken von P nicht F-saturiert sein können (denn sie liegen auf einer Kante von F', also nicht auf einer Kante aus F), ist P ein F-alternierender Weg und der Satz ist bewiesen. □

Beispiel: Wir betrachten noch einmal das vorige Beispiel und insbesondere Abbildung 3.35. In diesem Graphen mit dem Matching $F = \{x_1x_3, x_4x_6, x_8x_9\}$ hatten wir den F-alternierenden Weg $P = x_2, x_4, x_6, x_8, x_9, x_{10}$ gefunden. Wir benutzen den ersten Teil des obigen Beweises und erhalten mit

$$\begin{aligned} F' &:= (\{x_1x_3, x_4x_6, x_8x_9\} \setminus \{x_4x_6, x_8x_9\}) \cup \{x_2x_4, x_6x_8, x_9x_{10}\} \\ &= \{x_1x_3, x_2x_4, x_6x_8, x_9x_{10}\} \end{aligned}$$

ein Matching mit $|F'| = 4$. Das entsprechende Matching geben wir in Abbildung 3.36 links an. Hier entdeckt man sehr schnell den F'-alternierenden Weg

$$P' = x_5, x_6, x_8, x_{10}, x_9, x_7$$

von x_5 nach x_7. Setzt man daher

$$\begin{aligned} F'' &:= (\{x_1x_3, x_2x_4, x_6x_8, x_9x_{10}\} \setminus \{x_6x_8, x_9x_{10}\}) \cup \{x_5x_6, x_8x_{10}, x_7x_9\} \\ &= \{x_1x_3, x_2x_4, x_5x_6, x_8x_{10}, x_7x_9\}, \end{aligned}$$

so erhält man das in Abbildung 3.36 rechts angegebene Maximum-Matching F''. □

[17] Wir folgen einmal wieder fast wörtlich der Darstellung bei M. Aigner (1996, S. 124 ff.).

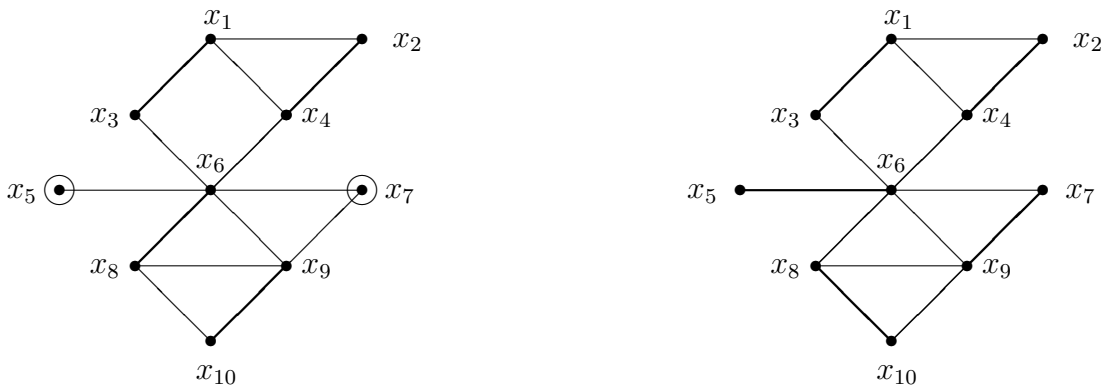


Abbildung 3.36: Der Weg zum Maximum-Matching

Nun wollen wir uns auf bipartite Graphen konzentrieren. Gegeben sei also ein bipartiter Graph $G = (U \cup W, E)$, gesucht sei ein Maximum-Matching. Bei gegebenem Matching F kommt es darauf an, einen F-alternierenden Weg (oft spricht man auch von einem F-vermehrenden (augmenting) Weg[18]) zu finden. Ein F-alternierender Weg startet in einer F-unsaturierten Ecke und endet in einer solchen. Die Anzahl der Kanten in einem F-alternierenden Weg ist ungerade, daher ist ein Endpunkt eines F-alternierenden Weges in U und der andere in W. Daher versucht man o. B. d. A. von einer F-unsaturierten Ecke $u \in U$ aus einen F-alternierenden Weg zu konstruieren[19]. Wir konstruieren einen *F-alternierenden Baum T mit der Wurzel u*, d. h. $T \subset G$ ist ein Baum, u ist Ecke in T und jede Ecke in T ist mit u durch einen (eindeutigen) Weg in T verbunden, in dem aufeinander folgende Kanten abwechselnd aus F und $E \setminus F$ sind. Ein F-alternierender Baum T mit Wurzel u heißt *gesättigt*, wenn man T (etwas lax formuliert) durch keine Kante aus G vergrößern kann. Wichtig ist nun die folgende Aussage.

Satz 8.8 *Sei $G = (U \cup W, E)$ ein bipartiter Graph, $F \subset E$ ein Matching von G und $u \in U$ eine F-unsaturierte Ecke. Weiter sei $T \subset G$ ein gesättigter F-alternierender Baum mit Wurzel u. Gibt es in T keinen bei u startenden F-alternierenden Weg, so gibt es auch in G keinen F-alternierenden Weg mit der Anfangsecke u*[20].

Beweis: Wir setzen $U_T := V(T) \cap U$, $W_T := V(T) \cap W$. Also ist U_T z. B. die Menge der Ecken in $T \subset G$, die zu U gehören. Dann ist $W_T \subset N(U_T)$, wobei $N(U_T)$ die Menge der Nachbarn der Elemente aus U_T ist. Denn zu jedem $w_0 \in W_T$ gibt es einen Weg $w_0, u_0, \ldots, w_n, u$ in T nach u, der abwechselnd Kanten aus F und $E \setminus F$ benutzt. Wegen $u_0 \in U_T$ ist $w_0 \in N(U_T)$ und folglich $W_T \subset N(U_T)$. Da T gesättigt ist, gilt hier

[18] Wir folgen bei der Definition eines F-alternierenden Weges M. Aigner (1996, S. 124). Insgesamt gebräuchlicher scheint es zu sein, einen F-alternierenden Weg als einen Weg zu definieren, bei dem abwechselnd Kanten aus F und $E \setminus F$ benutzt werden und erst einen F-vermehrenden Weg als einen F-alternierenden Weg zu definieren, bei dem die Anfangs- und die Endecke F-unsaturiert sind.

[19] Falls keine F-unsaturierte Ecke existiert, so ist F ein Maximum-Matching und $m(G) = |U|$.

[20] Hat man also für alle F-unsaturierten Ecken u aus U festgestellt, dass der zugehörige gesättigte F-alternierende Baum mit Wurzel u keinen F-alternierenden Weg enthält, so ist F wegen Satz 8.7 ein Maximum-Matching.

sogar Gleichheit. Angenommen, es gibt in G einen F-alternierenden Weg

$$P: \quad u = u_0, w_1, u_1, \ldots, u_{p-1}, w_p = w$$

von u nach einen $w \in W$. Ist $u_j \in U_T$ (dies ist für $j = 0$ der Fall), so ist w_{j+1} als Nachbar von u_j in $N(U_T) = W_T$. Es ist $w_{j+1}u_{j+1} \in F$ und daher $u_{j+1} \in U_T$. Insgesamt wäre P also im Widerspruch zur Annahme ein F-alternierender Weg in T. □

Es kommt also darauf an, zu vorgegebenem Matching F und F-unsaturiertem $u \in U$ sukzessive einen F-alternierenden Baum mit Wurzel u aufzubauen und in ihm F-alternierende Wege zu finden. Wir machen uns dies durch ein Beispiel klar.

Beispiel: Gegeben sei der bipartite Graph aus Abbildung 3.34, den wir in Abbildung 3.37 links noch einmal angeben, daneben wird das Matching $F_0 := \{2A, 4B, 5G, 6E\}$ dargestellt. Es gibt drei F_0-unsaturierte Ecken in $U := \{1, 2, 3, 4, 5, 6, 7\}$, nämlich 1,

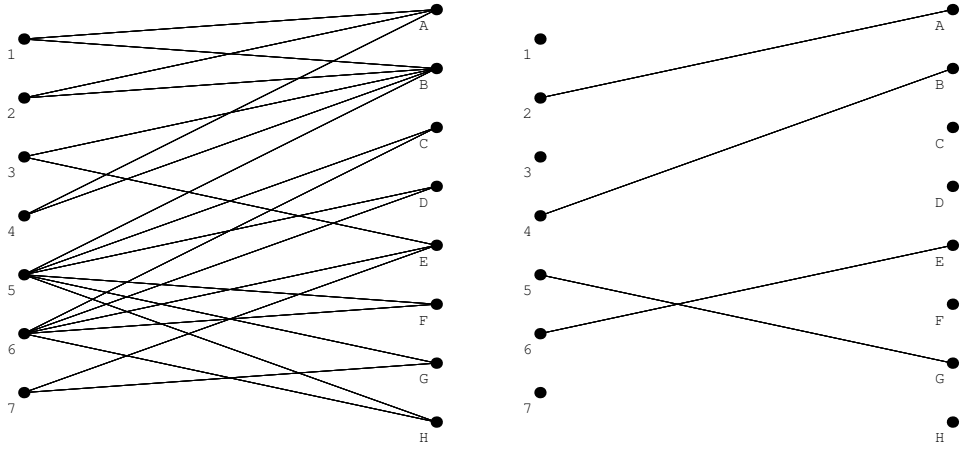


Abbildung 3.37: Ein bipartiter Graph und ein Matching

3 und 7. Wir geben in Abbildung 3.38 links den gesättigten F_0-alternierenden Baum T_0 mit der Wurzel 1 an. Offensichtlich gibt es hierin keinen F_0 alternierenden Weg. Daher gehen wir jetzt zur Ecke 3 über und geben in Abbildung 3.38 in der Mitte den F_0-alternierenden Baum T_1 mit der Wurzel 3 an. Hier sind die Ecken C, D, F, H jeweils F_0-unsaturiert und man erhält den F_0-alternierenden Weg $3, E, 6, C$, wobei man statt nach C auch nach D, F oder H gehen kann. Damit erhält man das neue Matching

$$F_1 := (\{2A, 4B, 5G, 6E\} \setminus \{6E\}) \cup \{3E, 6C\} = \{2A, 3E, 4B, 5G, 6C\}.$$

Dieses Matching geben wir in Abbildung 3.39 links an. Die Ecken 1 und 7 sind F_1-unsaturiert. Der von 1 ausgehende F_1-alternierende Baum enthält wieder keinen F_1-alternierenden Weg. Daher geben wir in Abbildung 3.38 rechts den F_1-alternierenden Baum mit der Wurzel 7 wieder. Hierin erkennt man den F_1-alternierenden Weg von 7 über G, 5 nach D, F oder H. Durch

$$F_2 := (\{2A, 3E, 4B, 5G, 6C\} \setminus \{5G\}) \cup \{7G, 5D\} = \{2A, 3E, 4B, 5D, 6C, 7G\}$$

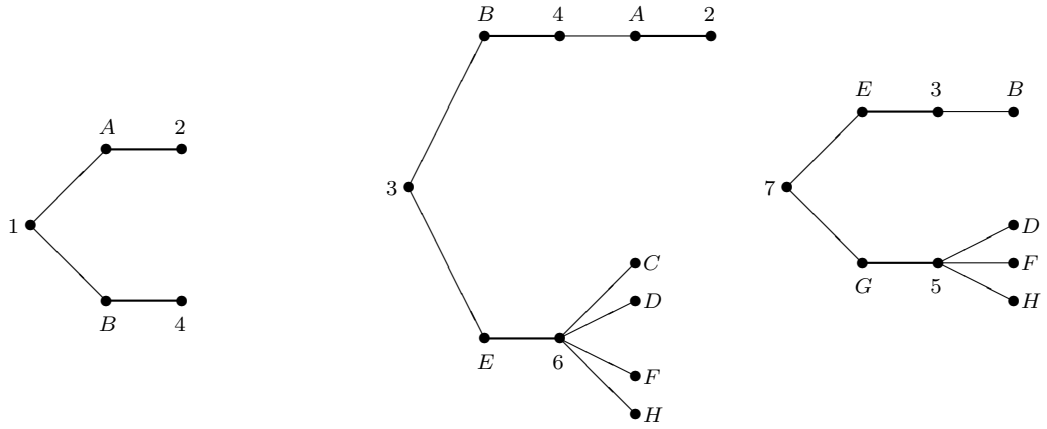


Abbildung 3.38: Alternierende Bäume

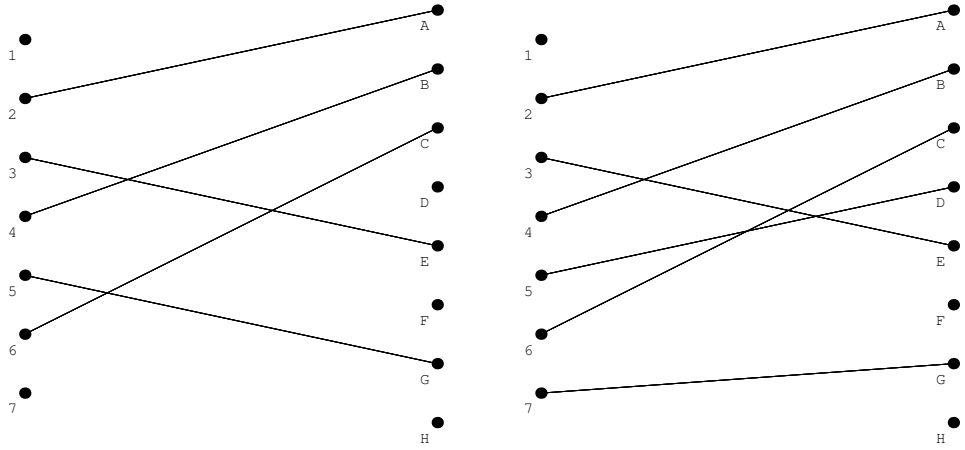


Abbildung 3.39: Ein verbessertes Matching und ein Maximum-Matching

erhält man ein vergrößertes Matching (siehe Abbildung 3.39 rechts), wobei man in der Kante $5D$ die Ecke D auch durch F oder H ersetzen könnte. Bezüglich dieses Matchings gibt es keinen alternierenden Weg, es ist also ein Maximum-Matching. □

Es ist nun nicht schwierig, von dem obigen Beispiel zu abstrahieren und die sogenannte *Ungarische Methode* zur Berechnung eines Maximum-Matching in einem bipartiten Graphen formal darzustellen. Das könnte folgendermaßen aussehen, wobei wir davon ausgehen, dass $G = (U \cup W, E)$ ein bipartiter Graph mit (o. B. d. A.) $|U| \leq |W|$ ist. Im Folgenden bezeichnen wir mit $V(F)$ die Menge der Ecken, die mit einem Matching inzidieren, ferner sei $V(T)$ die Eckenmenge und $E(T)$ die Kantenmenge eines Baumes T.

1. Sei $F \subset E$ ein Matching in G.
2. Falls $V(F) \cap U = U$, dann: STOP, F ist ein Maximum-Matching.

   Andernfalls wähle $u \in U \setminus V(F)$ (also eine F-unsaturierte Ecke in U) und setze

$$T := (\{u\}, \emptyset), \qquad U_T := \{u\}, \qquad W_T := \emptyset.$$

3. Ist $W_T = N(U_T)$, so setze $U := U \setminus \{u\}$ und gehe nach 2.

   Andernfalls wähle $w_0 \in N(U_T) \setminus W_T$ und eine Kante $e_0 = u_0 w_0$ mit $u_0 \in U_T$.

4. Inzidiert w_0 mit F, ist also F-saturiert, so existiert (genau) eine Kante $e = w_0 u_1 \in F$, wobei $u_1 \in U \setminus U_T$. Setze

   $$U_T := U_T \cup \{u_1\}, \qquad W_T := W_T \cup \{w_0\}, \qquad T := (U_T \cup W_T, E(T) \cup \{u_0 w_0, w_0 u_1\})$$

   und gehe nach 3.

   Andernfalls (wenn also w_0 unsaturiert bezüglich F ist) ist der im Baum T eindeutig bestimmte Weg von u über u_0 nach w_0 ein F-alternierender Weg. Nimmt man aus diesem Weg die Kanten zum Matching F hinzu, die nicht zu F gehören, lässt man dagegen im Matching diejenigen Kanten fort, die im Weg zu F gehören, so erhält man ein neues Matching, das eine Kante mehr als F enthält. Dieses Matching wird wieder F genannt, danach wird nach 2. gegangen.

Nun betrachten wir ein etwas anderes Problem, nämlich das *Zuordnungsproblem*. In unserer jetzt benutzten Sprache besteht dieses in folgendem. Gegeben ist ein bipartiter Graph $G = (U \cup W, E)$ und eine Gewichtung der Kanten durch eine Abbildung $w\colon E \longrightarrow \mathbb{R}_{\geq 0}$ (gleich werden wir uns allerdings aus bestimmten Gründen auf *ganzzahlige* Gewichte zurückziehen). Gesucht ist ein Matching $F \subset E$ mit maximalem Gewicht $w(F) := \sum_{e \in F} w(e)$. Haben alle Kanten dasselbe positive Gewicht, etwa Eins, so suchen wir nach einem Maximum-Matching. Indem wir notfalls Kanten das Gewicht 0 geben, können wir o. B. d. A. annehmen, dass G alle Verbindungen von U und W enthält und $|U| = |W|$ gilt, es sich also bei G um den vollständigen bipartiten Graphen $K_{n,n}$ handelt. Weiter ist es einfach, von einem Maximum- zu einem Minimum-Problem überzugehen. Man setze nämlich $W := \max_{e \in E} w(e)$ und anschließend $c(e) := W - w(e)$ für alle Kanten e in $K_{n,n}$. Die gegebene Aufgabe ist dann äquivalent dazu, in $K_{n,n}$ ein Matching F mit minimalen Kosten $c(F) := \sum_{e \in F} c(e)$ zu bestimmen. Ein Matching in $K_{n,n}$ entspricht offensichtlich einer $n \times n$-Permutationsmatrix, also einer $n \times n$-Matrix, die in jeder Zeile und jeder Spalte genau eine Eins und sonst nur Nullen enthält. Daher kann das Problem auch folgendermaßen formuliert werden:

- Gegeben sei eine $n \times n$-Matrix nichtnegativer, ganzzahliger Kosten $(c_{ij}) \in \mathbb{N}_0^{n \times n}$. Gesucht ist eine Permutationsmatrix (x_{ij}) mit minimalen Kosten $\sum_{i,j=1}^{n} c_{ij} x_{ij}$.

Wir schildern die *Ungarische Methode* zur Lösung des Zuordnungsproblems. Der erste Schritt ist ein Initialisierungsschritt. In ihm wird die Kostenmatrix (c_{ij}) verändert, ohne dass sich hierbei die Lösungsmenge ändert.

- Input: $(c_{ij}) \in \mathbb{N}_0^{n \times n}$.
  - Subtrahiere $u_i := \min_{j=1,\ldots,n} c_{ij}$ von Zeile i in (c_{ij}) für $i = 1, \ldots, n$. Die neue Matrix sei wieder mit (c_{ij}) bezeichnet.
  - Subtrahiere $v_j := \min_{i=1,\ldots,n} c_{ij}$ von Spalte j in (c_{ij}) für $i = 1, \ldots, n$. Die neue Matrix sei wieder mit (c_{ij}) bezeichnet.

Beispiel: Sei

$$(c_{ij}) := \begin{pmatrix} 9 & 11 & 12 & 11 \\ 6 & 3 & 8 & 5 \\ 7 & 6 & 13 & 11 \\ 9 & 10 & 10 & 7 \end{pmatrix}.$$

Subtraktion der Zeilenminima und anschließende Subtraktion der (nur in der dritten Spalte positiven) Spaltenminima liefert

$$(c_{ij}) := \begin{pmatrix} 0 & 2 & 3 & 2 \\ 3 & 0 & 5 & 2 \\ 1 & 0 & 7 & 5 \\ 2 & 3 & 3 & 0 \end{pmatrix}, \qquad (c_{ij}) := \begin{pmatrix} 0 & 2 & 0 & 2 \\ 3 & 0 & 2 & 2 \\ 1 & 0 & 4 & 5 \\ 2 & 3 & 0 & 0 \end{pmatrix}.$$

Wäre in der resultierenden Matrix (c_{ij}) (zufällig) das Element in der Position $(2, 4)$ eine Null, so hätten wir schon eine optimale Zuordnung erhalten, nämlich die durch die Permutationsmatrix

$$(x_{ij}) = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix}$$

gegebene. □

Schlüssel für das weitere Verfahren ist die folgende Aussage, welche man als Korollar zum Heiratssatz 8.2 bzw. von Satz 8.5 auffassen kann. Eine nur geringfügig andere Formulierung findet man in Aufgabe 20.

- Sei $C = (c_{ij}) \in \mathbb{N}_0^{n \times n}$ gegeben. Unter einer *Linie* in C verstehen wir eine Zeile oder eine Spalte in C. Dann ist die Minimalzahl der alle Nullen enthaltenden Linien (wir sprechen von einer *minimalen Bedeckung*) gleich der Maximalzahl der Nullen, von denen nicht zwei auf einer Linie liegen.

Im obigen Beispiel hat man durch die erste und vierte Zeile sowie die zweite Spalte eine minimale Bedeckung der Nullen, eine sogenannte 0-Diagonale der Länge 3 steht in den Positionen $(1, 1)$, $(3, 2)$ und $(4, 3)$. Allgemein nennen wir eine Menge von m Null-Einträgen in C, von denen nicht zwei auf einer Linie liegen, eine 0-Diagonale der Länge m in C. Der Zusammenhang der gemachten Aussage mit Satz 8.5 dürfte klar sein: Eine 0-Diagonale entspricht einem Matching, eine Bedeckung der Nullen einem Träger.

Nach Schilderung des Initialisierungsschrittes fahren wir mit der Beschreibung der ungarischen Methode fort.

- Man bestimme in C eine minimale Bedeckung der Nullen. Falls die minimale Bedeckung die Länge n hat, so ist durch eine maximale 0-Diagonale (ebenfalls der Länge n) die gesuchte Permutationsmatrix bzw. das Matching mit minimalen Kosten bestimmt.

Wir gehen daher jetzt davon aus, die minimale Bedeckung bestehe aus weniger als n Linien, etwa aus z Zeilen und s Spalten, und es sei $z + s < n$. Sei d das Minimum der unbedeckten Einträge in C. Es ist $d > 0$, denn nach Definition einer Bedeckung müssen *alle* Null-Einträge bedeckt sein. Wir ziehen zunächst d von den $n - z$ Zeilen von C ab, die nicht bedeckt sind. Anschließend addieren wir d zu den s Spalten der Bedeckung. Die neue Matrix nennen wir vorübergehend C'. Die Menge der Lösungen hat sich hierbei nicht verändert (Beweis?). Wir setzen also

$$c'_{ij} := \begin{cases} c_{ij} - d, & \text{falls } c_{ij} \text{ unbedeckt,} \\ c_{ij}, & \text{falls } c_{ij} \text{ von einer Zeile oder einer Spalte bedeckt ist,} \\ c_{ij} + d, & \text{falls } c_{ij} \text{ von einer Zeile und einer Spalte bedeckt ist.} \end{cases}$$

Insbesondere ist auch $(c'_{ij}) \in \mathbb{N}_0^{n \times n}$. Die Anzahl der doppelt bedeckten (also durch eine Zeile *und* eine Spalte) Einträge ist sz, die der unbedeckten Einträge ist $(n-z)(n-s)$. Nun vergleichen wir die Summe aller Einträge in den Matrizen C' und C. Wir erhalten

$$\begin{aligned} \sum_{i,j=1}^{n} c'_{ij} - \sum_{i,j=1}^{n} c_{ij} &= \sum_{i,j=1}^{n} (c'_{ij} - c_{ij}) \\ &= -d(n-z)(n-s) + dzs \\ &= dn(z + s - n) \\ &< 0. \end{aligned}$$

Die neue Matrix C', die nichtnegative ganze Einträge besitzt, hat also echt kleinere Gesamtkosten. Insgesamt lautet dieser Schritt also:

- Sei d der kleinste unbedeckte Eintrag in C. Ziehe von C von allen unbedeckten Einträgen d ab und addiere d zu allen doppelt bedeckten Einträgen. Die entstehende Matrix werde wieder mit C bezeichnet. Hiermit gehe man zu dem vorigen Schritt zurück.

Nach endlich vielen Schritten bricht dieses Verfahren ab. Genau hier geht die Voraussetzung ein, dass die Einträge der Kostenmatrix nichtnegativ und *ganzzahlig* sind.

Beispiel: Wir kehren zum vorigen Beispiel zurück. Sei also

$$(c_{ij}) = \begin{pmatrix} 0 & 2 & 0 & 2 \\ 3 & 0 & 2 & 2 \\ 1 & 0 & 4 & 5 \\ 2 & 3 & 0 & 0 \end{pmatrix}.$$

Eine Bedeckung ist durch $\{1, 4\} \cup \{2\}$ gegeben (die erste Menge ist eine Menge von Zeilen, die zweite eine von Spalten). Dann ist $d = 1$ und man erhält die neue Matrix

$$(c_{ij}) = \begin{pmatrix} 0 & 3 & \mathbf{0} & 2 \\ 2 & \mathbf{0} & 1 & 1 \\ \mathbf{0} & 0 & 3 & 4 \\ 2 & 4 & 0 & \mathbf{0} \end{pmatrix}.$$

Eine 0-Diagonale haben wir fett gedruckt angedeutet. Eine Lösung ist also gegeben durch

$$(x_{ij}) = \begin{pmatrix} 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}.$$

Die Gesamtkosten sind 12+3+7+7 = 29 (Summe der Einträge in der *Ausgangsmatrix*, die zur optimalen Zuordnung gehören). Oder auch 9+3+6+7 (Reduktion der Zeilen)+3 (Reduktion der Spalten)+1 (weiterer Schritt) = 29. Dies hat den Vorteil, dass man sich die Ausgangsmatrix nicht zu merken braucht. Übrigens scheint `BipartiteMatching` nicht für gewichtete bipartite Graphen zu funktionieren. □

3.9 Planare Graphen

Wir hatten früher schon erwähnt, wann wir einen Graphen $G = (V, E)$ *planar* nennen. Grob gesagt sind dies Graphen, die sich ohne Kantenüberschneidungen in die Ebene einbetten lassen. Genauer definieren wir (siehe L. Volkmann (1991, S. 179)):

Definition 9.1 Eine stetige Abbildung $e\colon [0,1] \longrightarrow \mathbb{R}^p$ ist ein *Jordanbogen* im $\mathbb{R}^p$, wenn $e(t) \neq e(s)$ für alle $s, t \in [0,1]$ mit $s \neq t$.

Definition 9.2 Ein Graph $G = (V, E)$ heißt ein *Euklidischer Graph im* $\mathbb{R}^p$, wenn folgende drei Bedingungen erfüllt sind:

1. Es ist $V = \{x_1, \ldots, x_n\} \subset \mathbb{R}^p$.
2. Es ist $E = \{e_1, \ldots, e_m\}$ mit Jordanbögen $e_1, \ldots, e_m$ im $\mathbb{R}^p$, wobei $e_i(0), e_i(1) \in V$ und $e_i(t) \notin V$ für alle $t \in (0,1)$, $i = 1, \ldots, m$. Die Ecke x_i und die Kante e_j heißen *inzident*, wenn $x_i \in \{e_j(0), e_j(1)\}$.
3. Die Kanten von G haben (außer Ecken) keine Schnittpunkte. Sind also e_i und e_j zwei verschiedene Kanten, so ist $e_i(s) \neq e_j(t)$ für alle $s, t \in (0,1)$. Zwei verschiedene Kanten heißen *inzident*, wenn sie mit einer gemeinsamen Ecke inzidieren.

Definition 9.3 Ist $G' = (V', E')$ ein Graph, der zu einem Euklidischen Graphen $G = (V, E)$ im $\mathbb{R}^p$ isomorph ist, so heißt G eine *Einbettung* von G' in den $\mathbb{R}^p$. Ein Euklidischer Graph im $\mathbb{R}^2$ heißt *ebener Graph*. Ein Graph, der zu einem ebenen Graphen isomorph ist, heißt ein *planarer Graph*.

Bemerkung: Jeder Graph G lässt sich in den $\mathbb{R}^3$ einbetten. Das kann man folgendermaßen einsehen. Man ordne den n Ecken von G paarweise verschiedene Punkte der x-Achse im $\mathbb{R}^3$ zu. Danach wähle man für die m Kanten von G paarweise verschiedene Ebenen im $\mathbb{R}^3$, die die x-Achse enthalten. Zu jeder Kante in G (also Verbindung zweier Ecken in G) kann man in der hierdurch festgelegten Ebene die beiden zugehörigen Punkte auf der x-Achse etwa durch einen Halbkreis verbinden. Da alle diese Halbkreise auf verschiedenen Ebenen liegen, die sich nur in der x-Achse schneiden, können sie sich

außerhalb derjenigen Punkte, die den Ecken des Graphen entsprechen, nicht schneiden. Damit ist eine Einbettung von G in den $\mathbb{R}^3$ gefunden. □

Beispiel: Die vollständigen Graphen K_1, K_2, K_3 sind selbstverständlich planar. Auch K_4 und der vollständige bipartite Graph $K_{2,3}$ sind planar (Beweis?). Die einfachsten nichtplanaren Graphen sind K_5 und $K_{3,3}$. Bei J. M. Aldous, R. B. Wilson (2000, S. 245) findet man eine Begründung, weshalb $K_{3,3}$ nicht planar ist. In Abbildung 3.40 geben wir $K_{3,3}$ und den darin enthaltenen Kreis $1, a, 2, b, 3, c$ an. Wenn $K_{3,3}$ planar wäre,

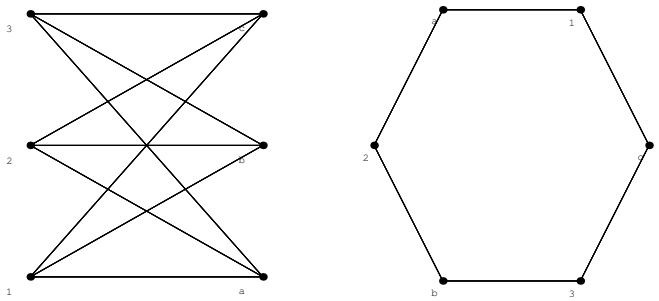


Abbildung 3.40: Der $K_{3,3}$ und ein darin enthaltener Kreis

müssten die noch fehlenden Kanten $1b$, $2c$ und $3a$ noch überschneidungsfrei in den Kreis eingezeichnet werden können, was aber offenbar[21] nicht möglich ist. Daher ist es nicht möglich, drei Häuser "kreuzungsfrei" mit drei Brünnen zu verbinden. □

Sei $G = (V, E)$ ein ebener Graph. Fassen wir G als Vereinigung der Ecken und Kanten auf, so ist $\mathbb{R}^2 \setminus G$ disjunkte Vereinigung zusammenhängender Gebiete, von denen genau eines nicht beschränkt ist. Diese Zusammenhangskomponenten von $\mathbb{R}^2 \setminus G$ nennen wir *Länder* (auch "faces" genannt, man denke an Seitenflächen eines Polyeders). Die Anzahl der Länder eines planaren Graphen ist natürlich die Anzahl der Länder einer ebenen Einbettung.

Nun folgt schon die *Eulersche Polyederformel*.

Satz 9.4 *Sei $G = (V, E)$ ein zusammenhängender planarer Graph mit n Ecken, m Kanten und f Ländern. Dann ist $n - m + f = 2$.*

Beweis: Wir halten n fest und beweisen die Behauptung durch vollständige Induktion nach m, der Anzahl der Kanten von G. Ist $m = 1$, so ist $n = 2$ und $f = 1$, die Aussage also richtig. Wir nehmen nun an, die Aussage sei für Graphen mit $\leq m - 1$ Kanten richtig. Angenommen, $e \in E$ sei keine Brücke. Nach Definition einer Brücke ist $G' := (V, E \setminus \{e\})$ zusammenhängend und natürlich planar. Dann hat G' genau so viel Ecken wie G, eine Kante und ein Land weniger als G, denn nach dem Entfernen der Grenze e wird aus zwei Ländern eines. Nach Induktionsvoraussetzung ist also $n-(m-1)+(f-1) = 2$ bzw. $n-m+f = 2$. Zum Schluss müssen wir noch den Fall betrachten, dass jede Kante von G eine Brücke ist. Dann ist G ein Baum, also kreisfrei (denn wenn es einen Kreis geben würde, so wäre jede Kante in diesem Kreis keine Brücke). Nach

[21] Einen genaueren Beweis werden wir mit einer Folgerung zur Eulerschen Polyederformel erhalten.

Satz 3.1 ist $m = n - 1$. Ferner ist $f = 1$, da ein Baum genau ein (nicht beschränktes) Land besitzt. Also gilt auch hier $n - m + f = 2$. Damit ist der Satz bewiesen. □

Als Folgerung formulieren wir:

Korollar 9.5 *Sei $G = (V, E)$ ein zusammenhängender planarer Graph mit $|V| \geq 3$. Dann gilt:*

1. *Es ist $|E| \leq 3|V| - 6$.*
2. *Enthält G kein Dreieck, so ist $|E| \leq 2|V| - 4$.*

Beweis: Natürlich können wir annehmen, dass G ein ebener Graph ist. Sei $n := |V|$ die Anzahl der Ecken, $m := |E|$ die Anzahl der Kanten und f die Anzahl der Länder. Für $m = 2$ ist die Aussage richtig, so dass $m \geq 3$ angenommen werden kann. Jedes Land wird von mindestens drei Kanten begrenzt, jede Kante gehört zum Rand von höchstens zwei Ländern[22]. Daher ist $3f \leq 2m$. Wegen der Eulerschen Polyederformel ist $3m - 3n + 6 \leq 2m$ bzw. $m \leq 3n - 6$, was schon die erste Behauptung ist. Enthält G kein Dreieck, so wird jedes Land von mindestens vier Kanten begrenzt, was $4f \leq 2m$ nach sich zieht. Einsetzen in die Eulersche Polyederformel liefert $m \leq 2n - 4$, die zweite Behauptung. □

Beispiel: Der K_5 hat $n = 5$ Ecken und $m = 10$ Kanten. Wegen der ersten der beiden obigen Aussagen ist K_5 nicht planar. Der vollständige bipartite Graph $K_{3,3}$ hat $n = 6$ Ecken und $m = 9$ Kanten, weiter enthält $K_{3,3}$ natürlich kein Dreieck. Aus der zweiten der beiden obigen Aussagen folgt, dass $K_{3,3}$ nicht planar ist. □

Wir gehen nicht ein auf weitere notwendige und/oder hinreichende Bedingungen dafür, dass ein Graph planar ist, oder auf Verfahren für einen "Planaritätstest". Hierzu verweisen wir auf die angegebene Literatur. Stattdessen wollen wir uns mit dem Färben von Landkarten beschäftigen.

Gegeben sei also ein zusammenhängender, ebener Graph $G = (V, E)$. Die Zusammenhangskomponenten von $\mathbb{R}^2 \setminus G$ hatten wir Länder genannt. Zur Veranschaulichung haben wir in Abbildung 3.41 den afrikanischen Kontinent und seine Länder aufgetragen. Zwei Länder heißen *benachbart*, wenn es eine Kante gibt, die zum Rand beider Länder gehört[23]. Man sagt, dass sich G (oder auch die durch G bestimmte *Landkarte*, map) durch p Farben färben lässt, wenn jedes Land in G so durch eine von p Farben gefärbt werden kann, dass zwei benachbarte Länder unterschiedlich gefärbt sind. Einer der berühmtesten Sätze der Mathematik und damit auch der Graphentheorie ist der *Vierfarbensatz*, dass sich nämlich jeder zusammenhängende ebene Graph (bzw. die zugehörige Landkarte) durch vier Farben färben lässt. Bekanntlich gibt es für diesen Satz bisher nur einen Computerbeweis (K. Appel, W. Haken (1976)), daher wird gelegentlich immer noch von der *Vierfarbenvermutung* gesprochen. Wir wollen uns hier nur mit einfachen Resultaten begnügen. Zunächst ist klar, dass es einen zusammenhängenden, ebenen Graphen gibt, der nicht mit weniger als vier Farben gefärbt werden kann.

[22] Es folgt eine Art Handshaking-Argument. Man betrachte die Menge von Paaren, bestehend aus einem Land und einer dieses Land begrenzenden Kante. Die Anzahl der Elemente dieser Menge ist mindestens $3f$ und höchstens $2m$.

[23] Zwei Länder, die sich nur in einer Ecke treffen, sind also nicht benachbart.

Abbildung 3.41: Die Länder Afrikas

In Abbildung 3.42 geben wir ein Beispiel, wobei man natürlich immer daran denken muss, dass auch das unbeschränkte Land gefärbt werden muss. Der erste Trick bei der

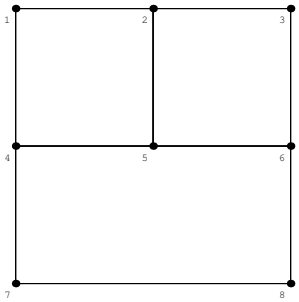


Abbildung 3.42: Vier Farben sind notwendig

Färbung einer Landkarte besteht i. Allg. darin, das "Länderfärbungsproblem" als ein "Eckenfärbungsproblem" zu formulieren. Hierzu führen wir den zu G *dualen Graphen* G^* ein. Jedem Land (auch dem nicht beschränkten) wird eine Ecke zugeordnet. Je zwei dieser Ecken werden durch eine Kante verbunden, wenn die zugrundeliegenden Länder eine gemeinsame Grenze haben. Es kommt nun darauf an, die Ecken des dualen Graphen G^* so zu färben, dass zwei benachbarte Ecken verschiedene Farben haben.

Beispiel: Als Beispiel betrachten wir eine Landkarte mit einem Teil Europas, siehe Abbildung 3.43. Genauer haben wir hier links 10 Länder eingetragen, nämlich die in Tabelle 3.1 angegebenen. Hinzu kommt noch das "Außenland", dem wir den Knoten 11 zuordnen. Bemerkt sei, dass wir (Entschuldigung!) Andorra, Liechtenstein, Monaco und San Marino fortgelassen haben. Rechts geben wir den dualen Graphen an, wobei wir auf geographische Belange zunächst keine Rücksicht nehmen. So erkennt man auch noch nicht, dass der duale Graph ebenfalls planar ist. Um eine ebene Version desselben Graphen herzustellen, gehen wir zunächst von einem Hamilton-Kreis im dualen Graphen aus (siehe Abbildung 3.44 links). Hier fehlen noch die Kanten

$$\{2,11\},\{3,6\},\{3,7\},\{3,8\},\{3,10\},\{3,11\},\{4,9\},\{4,11\},\{5,10\},\{6,10\},\{6,11\}$$

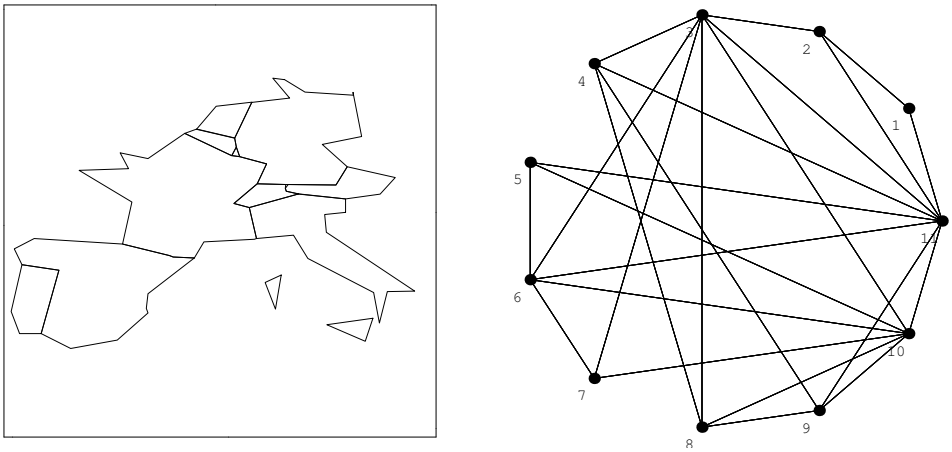


Abbildung 3.43: Einige Länder Europas und der duale Graph

.

Tabelle 3.1: Einige Länder Europas

1	Portugal
2	Spanien
3	Frankreich
4	Italien
5	Niederlande
6	Belgien
7	Luxemburg
8	Schweiz
9	Österreich
10	Deutschland

und

$$\{8,10\},\{9,11\},\{10,11\}.$$

In Abbildung 3.44 rechts haben wir die Kanten

$$\{2,11\},\{3,6\},\{3,7\},\{3,8\},\{3,10\},\{3,11\},\{6,11\},\{8,10\}$$

schon eingetragen. Die noch fehlenden Kanten

$$\{4,9\},\{4,11\},\{5,10\},\{6,10\},\{9,11\},\{10,11\}$$

können mühelos "außen herum" überschneidungsfrei gezeichnet werden. In Abbildung 3.45 geben wir eine Färbung der Landkarte Westeuropas bzw. des dualen Graphen mit vier Farben an. Diese haben wir durch eine eine Eckenfärbung des dualen Graphen mit Hilfe des *Mathematica*-Befehls `VertexColoring` erhalten. Etwas unglücklich (aber nicht verboten) ist, dass Korsika und Sizilien die gleiche Farbe haben. Man muss sich noch vorstellen, dass die "Außenwelt" rot gefärbt ist. □

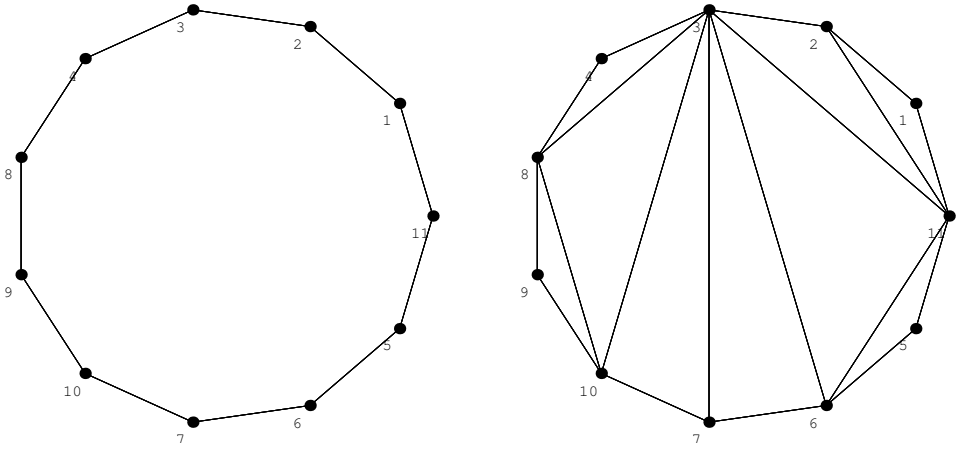


Abbildung 3.44: Der duale Graph ist planar

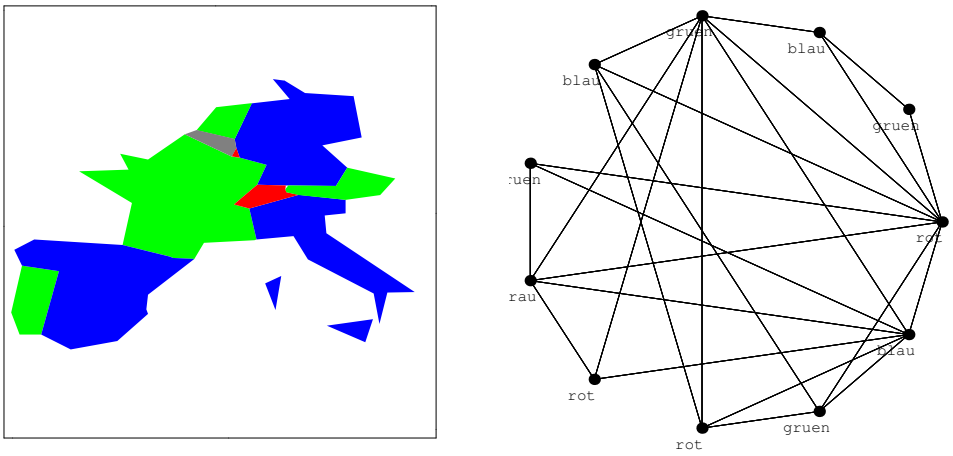


Abbildung 3.45: Vier Farben genügen

Bemerkung: Gegeben sei ein zusammenhängender, ebener Graph $G = (V, E)$ mit $n := |V|$ Ecken, $m := |E|$ Kanten und f Ländern. Dann ist der duale Graph zusammenhängend und planar[24]. Der Zusammenhang ist hierbei klar, denn (siehe L. Volkmann (1991, S. 187)) aus jedem beschränkten Gebiet von G gelangt man über Kanten benachbarter Gebiete zum äußeren Gebiet von G. Zu je zwei Ecken im dualen Graphen gibt es also einen verbindenden Weg, d. h. auch G^* ist zusammenhängend. Auf einen formalen Beweis für die Planarität des dualen Graphen wollen wir verzichten (wer weiß einen einfachen?). Sei n^* die Anzahl der Ecken, m^* die Anzahl der Kanten und f^* die Anzahl der Länder im dualen Graphen. Wegen der Eulerschen Polyederformel (angewandt auf G^*) ist $n^* - m^* + f^* = 2$. Nach Konstruktion ist $n^* = f$ und $m^* = m$. Wegen der Eulerschen Polyederformel (angewandt auf G) ist daher $f^* = 2 - f + m = n$. □

[24]Bei L. Volkmann (1991, S. 186) kann man lesen, dass es "anschaulich recht einleuchtend" ist, dass mit G auch der duale Graph G^* planar ist.

Nun wollen wir den *Sechsfarbensatz* formulieren und beweisen[25].

Satz 9.6 *Ein zusammenhängender, ebener Graph bzw. die zugehörige Landkarte lässt sich durch sechs Farben färben.*

Beweis: Wir zeigen, dass eine Eckenfärbung eines zusammenhängenden, planaren Graphen $G = (V, E)$ mit sechs Farben möglich ist (wobei natürlich zwei benachbarte Ecken unterschiedlich gefärbt sind). Eine Anwendung dieses Resultats auf den dualen Graphen liefert dann die Behauptung[26]. Wir beweisen die Aussage durch vollständige Induktion nach der Anzahl n der Ecken von G. Die Aussage ist trivialerweise richtig für Graphen mit $n \leq 6$ Ecken. Wir nehmen an, sie sei richtig für Graphen mit weniger als n Ecken und zeigen, dass eine Eckenfärbung eines zusammenhängenden, planaren Graphen mit sechs (oder weniger) Farben möglich ist. Wegen Aufgabe 23 (mit dem Hinweis ist die Lösung eine einfache Folgerung aus einem Korollar zu der Eulerschen Polyederformel) gibt es eine Ecke v in G mit $d(v) \leq 5$. Also gibt es eine Ecke v mit höchstens fünf Nachbarn. Diese Ecke mitsamt der Kanten, auf denen sie liegt, entferne man aus dem Graphen G und erhalte so den (nach wie vor planaren, möglicherweise aber nicht zusammenhängenden) Graphen H mit $n-1$ Ecken. Diesen Prozess veranschaulichen wir uns in Abbildung 3.46. Nach Induktionsannahme können die Ecken von H (bzw. der

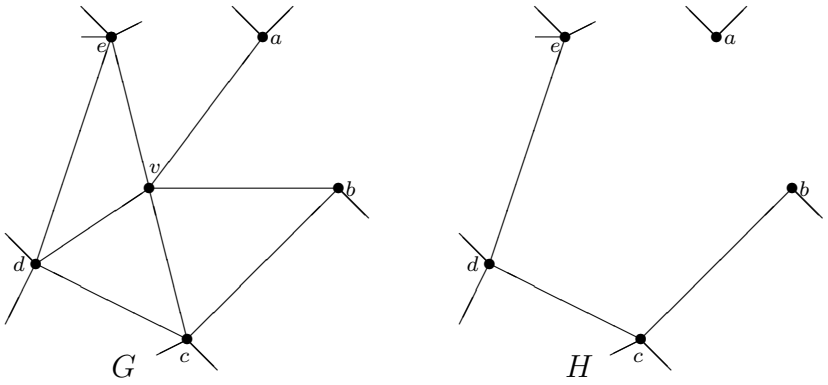


Abbildung 3.46: Gewinne H aus G durch Entfernen der Ecke v

Komponenten von H, wenn H nicht zusammenhängend ist) mit sechs Farben so gefärbt werden, dass benachbarte Ecken unterschiedlich gefärbt sind. Die höchstens fünf Nachbarn von v sind dadurch gefärbt, hierzu wurden natürlich höchstens fünf Farben benötigt. Zur Färbung von v steht noch eine Farbe zur Verfügung, so dass eine Eckenfärbung von G mit höchstens sechs Farben gelungen ist. Der Beweis ist damit abgeschlossen. □

Mit etwas mehr Aufwand können wir auch den *Fünffarbensatz* beweisen.

Satz 9.7 *Ein zusammenhängender, ebener Graph bzw. die zugehörige Landkarte lässt sich durch fünf Farben färben.*

[25] Wir folgen hier und auch beim entsprechenden Fünffarbensatz der Darstellung bei J. M. Aldous, R. J. Wilson (2000, S. 284 ff.).

[26] Hierbei benutzen wir, dass der zu einem zusammenhängenden, planaren Graphen duale Graph ebenfalls zusammenhängend und planar ist.

Beweis: Der Beginn des Beweises ist wie beim Sechsfarbensatz. Entsprechend zeigen wir, dass die Ecken eines zusammenhängenden, planaren Graphen G mit fünf Farben so gefärbt werden können, dass benachbarte Ecken unterschiedliche Farben haben. Dies geschieht wieder durch vollständige Induktion nach n, der Anzahl der Ecken von G. Der Induktionsanfang kann für $n = 1, 2, 3, 4, 5$ gemacht werden, die Annahme ist, die Aussage sei für zusammenhängende, planare Graphen mit weniger als n Ecken richtig. Auch der Beginn des Induktionsbeweises ist wie beim Beweis des Sechsfarbensatz. Es gibt eine Ecke v in G mit höchstens fünf Nachbarn. Diese Ecke mitsamt der Kanten, auf denen sie liegt, entferne man wieder aus dem Graphen G und erhalte so den Graphen H mit $n - 1$ Ecken (siehe Abbildung 3.46). Nach Induktionsvoraussetzung können die Ecken von H (bzw. der Komponenten von H, wenn H nicht zusammenhängend ist) mit fünf Farben so gefärbt werden, dass benachbarte Ecken unterschiedlich gefärbt sind. Wir können annehmen, dass v fünf Nachbarn besitzt, die mit fünf verschiedenen Farben gefärbt sind, da man andernfalls noch eine Farbe zur Färbung von v übrig hätte. Die fünf Farben seien etwa (wir benutzen englische Vokabeln, da sich die Anfangsbuchstaben der Farben besser unterscheiden) blue (b), green (g), purple (p), red (r) und amber (a), siehe Abbildung 3.47.Wir greifen uns zwei beliebige Nachbarn von v heraus,

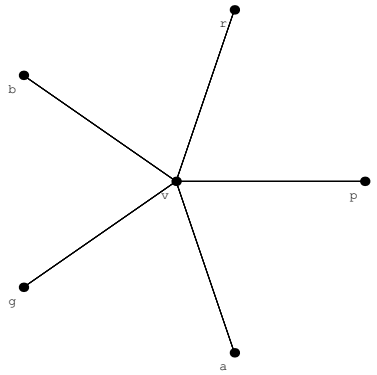


Abbildung 3.47: Die Ecke v und ihre fünf Nachbarn

etwa die red (r) und amber (a) gefärbten Nachbarn. Wir betrachten Wege, die nur aus roten (r) oder bernsteinfarbenen (a) Ecken bestehen, beginnend bei den Nachbarn r bzw. a und machen eine Fallunterscheidung. Im ersten Fall sind alle r- und a-Ecken, die vom r-Nachbarn erreicht werden können, verschieden von denen, die vom a-Nachbarn aus erreicht werden können. Insbesondere gibt es dann keinen Weg vom r-Nachbarn zum a-Nachbarn, der aus (natürlich abwechselnd) roten und bernsteinfarbenen Nachbarn besteht. In diesem Fall vertausche man einfach die Farben derjenigen Ecken, die vom r-Nachbarn aus auf einem r-a-Weg erreicht werden kann und färbe anschließend v rot. Diesen Fall veranschaulichen wir in Abbildung 3.48. Im zweiten Teil gibt es einen r-a-Weg vom r-Nachbarn zum a-Nachbarn. Ein Vertauschen der Farben würde nichts nützen, da v dann nach wie vor einen r- und einen a-Nachbarn hätte. Andererseits kann es keinen Weg aus violetten (p) und grünen (g) Ecken vom p-Nachbarn zum g-Nachbarn geben, da sonst der r-a-Weg gekreuzt werden müsste. Daher kann man im p-g-Weg von v über den p-Nachbarn die Farben purple (p) und green (g) vertauschen und anschließend die Ecke v mit der Farbe purple (p) färben, man hat den zweiten

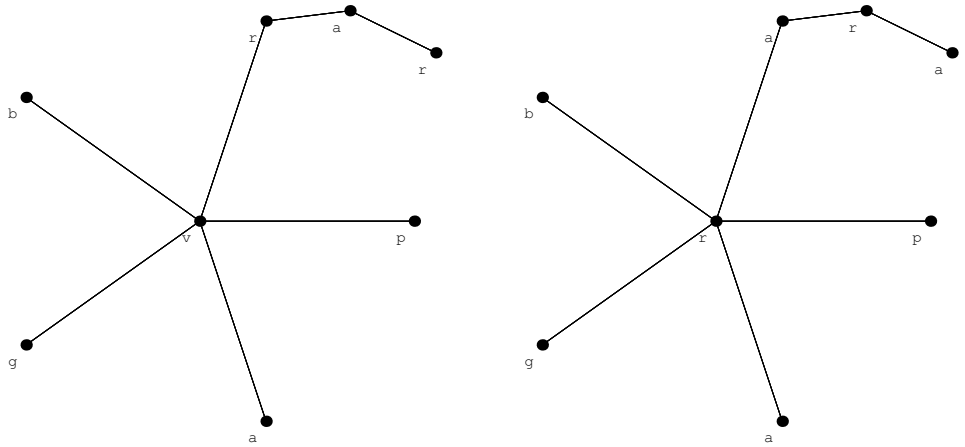


Abbildung 3.48: Vertausche Farben r und a für vom r-Nachbarn ausgehende r-a-Wege

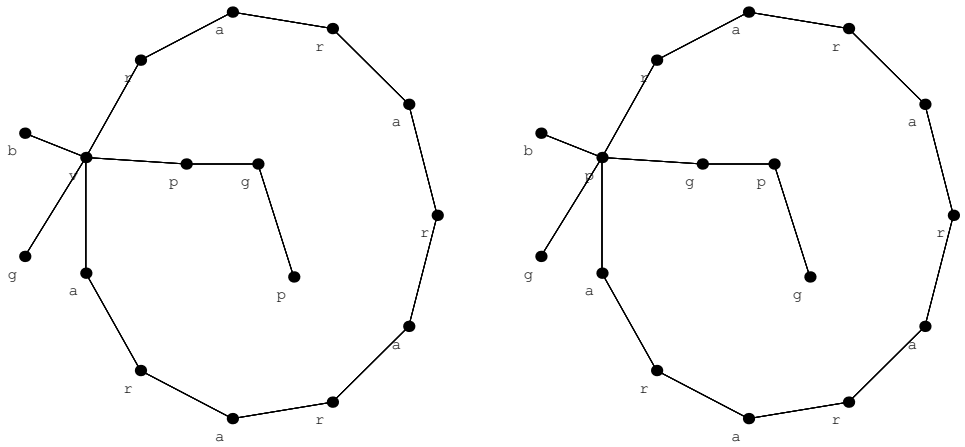


Abbildung 3.49: Vertausche green (g) und purple (p) rechts, färbe v mit p

Fall also auf den ersten bezüglich eines anderen Farbpaares zurückgeführt. Dies ist in Abbildung 3.49 veranschaulicht. Insgesamt ist der Satz bewiesen. □

Beispiel: Ein[27] Unternehmen muss gewisse Chemikalien, nämlich genau sieben verschiedene Arten, lagern. Gewisse Chemikalien müssen von anderen getrennt werden, können aber durchaus mit anderen zusammen gelagert werden. In der folgenden Tabelle ist durch $*$ gekennzeichnet, dass entsprechende Chemikalien nur getrennt voneinander

[27] Auch hier folgen wir J. M. Aldous, R. J. Wilson (2000, S. 276).

gelagert werden können.

	a	b	c	d	e	f	g
a	–	*	*	*	–	–	*
b	*	–	*	*	*	–	*
c	*	*	–	*	–	*	–
d	*	*	*	–	–	*	–
e	–	*	–	–	–	–	–
f	–	–	*	*	–	–	*
g	*	*	–	–	–	*	–

Die Frage ist, wie man die Chemikalien lagern sollte, wobei man mit möglichst wenig Sonderlagern auskommen möchte. Wir definieren einen Graphen, dessen Ecken die sieben Chemikalien sind. Diese Ecken werden durch eine Kante verbunden, wenn die entsprechenden Chemikalien getrennt gelagert werden müssen, siehe Abbildung 3.50.Dies

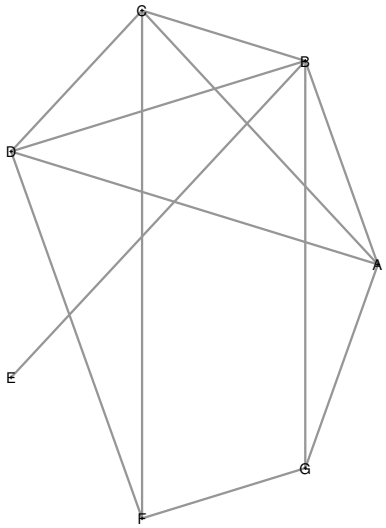


Abbildung 3.50: Unverträgliche Chemikalien

ist kein planarer Graph, wie wir durch Anwenden von `PlanarQ` durch die Antwort `False` erhalten haben (natürlich muss der Graph zunächst, z. B. durch eine Adjazenzliste bzw. im wesentlichen die obige Tabelle, erzeugt werden). Nach dem Befehl `VertexColoring` erhält man die mögliche Aufteilung $\{a, e\}$, $\{b, f\}$, $\{c, g\}$, $\{d\}$. Bei J. M. Aldous, R. J. Wilson ist die weitere mögliche Aufteilung $\{a, e\}$, $\{b, f\}$, $\{c\}$, $\{d, g\}$ angegeben. Hier haben wir also einen nichtplanaren Graphen, dessen Ecken durch vier Farben so gefärbt werden können, dass benachbarte Ecken unterschiedliche Farben haben. Da eine entsprechende Färbung mit drei Farben nicht möglich ist, sagt man, der Graph habe die chromatische Zahl 4. Die Aussage des *Vierfarbensatzes* ist es, dass jeder zusammenhängende, planare Graph eine chromatische Zahl ≤ 4 besitzt. Für nichtplanare Graphen ist das natürlich i. Allg. nicht richtig, denn z. B. hat der vollständige Graph K_n die chromatische Zahl n. □

3.10 Aufgaben

1. Man zeige, dass der Hyperwürfel Q_n genau $n2^{n-1}$ Kanten besitzt und bipartit ist.

2. Sei $G = (V, E)$ ein Graph mit $V = \{v_1, \ldots, v_n\}$ und $E = \{e_1, \ldots, e_q\}$. Sei $A \in \mathbb{R}^{n \times n}$ die Adjazenz- und $B \in \mathbb{R}^{n \times q}$ die Inzidenzmatrix von G, sei $D := \text{diag}\,(d(v_1), \ldots, d(v_n))$. Dann ist $BB^T = D + A$.

3. Ein Quickie: Man zeige, dass es keinen Graphen mit der Gradfolge $(5, 4, 3, 2, 2, 1)$ oder $(5, 5, 4, 4, 0)$ geben kann. Weiter konstruiere man einen Graphen mit der Gradfolge $(4, 3, 3, 3, 2, 2, 2, 1)$.

4. Sei $G = (V, E)$ ein zusammenhängender Graph mit $n := |V|$, $m := |E|$. Dann ist $n \leq m + 1$.

5. Sei G ein Graph mit 10 Ecken, der nicht zusammenhängend ist. Man zeige, dass G höchstens 36 Kanten besitzt.

6. Ein Graph $G = (V, E)$ mit $|E| > \lfloor |V|^2/4 \rfloor$ enthält[28] ein Dreieck.

7. Sei $G = (V, E)$ ein zusammenhängender Graph. Mit $d(x, y)$ bezeichnen wir den Abstand zweier Ecken $x, y \in V$, also die Länge eines kürzesten Weges von x nach y. Für $x \in V$ definieren wir $r(x) := \max_{y \in V \setminus \{x\}} d(x, y)$ als die *Exzentrizität* in x, also die Länge eines längsten von x ausgehenden kürzesten Weges. Dann heißt $r(G) := \min_{x \in V} r(x)$ der *Radius* von G und $Z(G) := \{x \in V : r(x) = r(G)\}$ das *Zentrum* von G.

   (a) Was ist der Radius und was das Zentrum zu dem in Abbildung 3.51 dargestellten Graphen G?

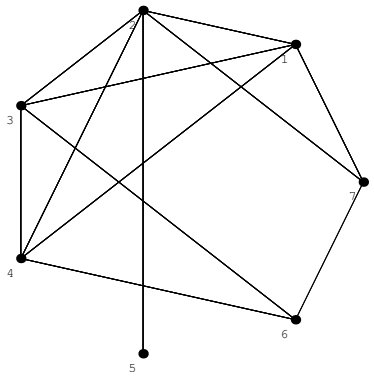


Abbildung 3.51: Was ist der Radius, was ist das Zentrum?

   (b) Man zeige, dass das Zentrum eines Baumes entweder aus einer Ecke oder zwei benachbarten Ecken besteht.

8. Man zeige, dass die in Abbildung 3.52 dargestellten Graphen isomorph sind.

9. Man zeichne alle Bäume mit höchstens 5 Ecken.

[28] Hierbei bedeutet $\lfloor x \rfloor$ für reelles x die größte ganze Zahl $\leq x$.

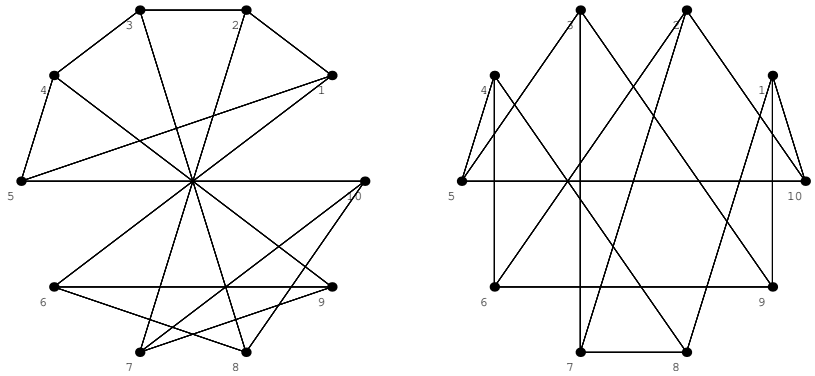


Abbildung 3.52: Zwei isomorphe Graphen

10. Ein Wurzelbaum (T, v_0) heißt ein *binärer Wurzelbaum*, wenn jeder innere Knoten höchstens zwei Kinder hat. Man zeige: Besitzt ein binärer Wurzelbaum t Blätter und hat er die Tiefe d, so ist $t \leq 2^d$.

11. In der folgenden Tabelle sind die Entfernungen (in Hunderten von Meilen) von sechs Städten angegeben.

	Berlin	London	Moskau	Paris	Rom	Sevilla
Berlin	–	7	11	7	10	15
London	7	–	18	3	12	11
Moskau	11	18	–	18	20	27
Paris	7	3	18	–	9	8
Rom	10	12	20	9	–	13
Sevilla	15	11	27	8	13	–

In dem zugehörigen gewichteten vollständigen Graphen bestimme man einen minimalen aufspannenden Baum.

12. Eine Firma hat sich zu überlegen, welches zusammenhängende Pipelinenetzwerk sie zwischen 7 Quellen $A, B, \ldots, G$ und einer Fabrik H bauen sollte. Die möglichen Pipelines und ihre Konstruktionskosten (in gewissen Geldeinheiten) sind gegeben durch

Pipeline	Kosten	Pipeline	Kosten
AB	23	CG	10
AE	17	DE	14
AD	19	DF	20
BC	15	EH	28
BE	30	FG	11
BF	27	FH	35

Welches Pipelinenetzwerk sollte gebaut werden und was sind seine Konstruktionskosten? Was ist die kostengünstigste Verbindung von der Quelle A zur Fabrik H?

13. Ein zusammenhängender Graph $G = (V, E)$ habe paarweise verschiedene Gewichte auf den Kanten. Man zeige, dass G einen *eindeutigen* minimalen aufspannenden Baum besitzt.

14. Man zeige, dass ein bipartiter Graph mit einer ungeraden Zahl von Ecken nicht Hamiltonsch ist. Hiermit zeige man, dass der in Abbildung 3.53 angegebene Graph nicht

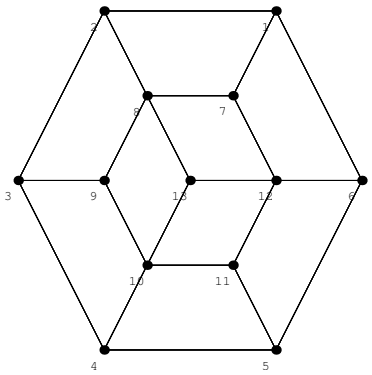


Abbildung 3.53: Bipartiter Graph mit ungerader Eckenzahl

Hamiltonsch ist.

15. Man zeige, dass der Hyperwürfel Q_n, $n \geq 2$, ein Hamiltonscher Graph ist.

16. Man zeige: Kann der Zusammenhang eines Graphen G durch die Entnahme einer einzigen Ecke und sämtlicher mit dieser Ecke inzidierender Kanten zerstört werden, so ist G kein Hamiltonscher Graph.

17. Sei G ein Graph mit n Ecken und der Eigenschaft, dass der Grad jeder Ecke mindestens $n/2$ ist. Man zeige, dass G Hamiltonsch ist.

18. Gegeben sei der in Abbildung 3.54 dargestellte gewichtete Digraph mit der Quelle q und

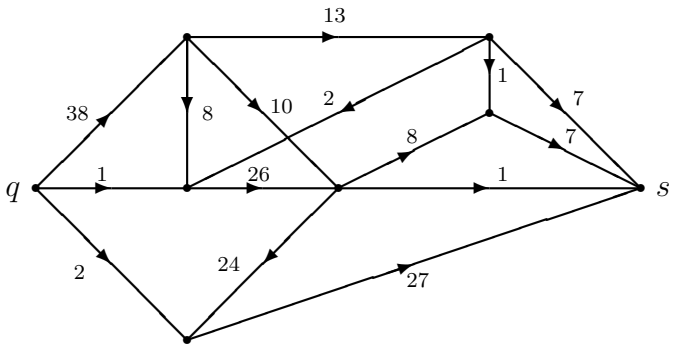


Abbildung 3.54: Ein gewichteter Digraph

der Senke s. Mit Hilfe von *Mathematica* bestimme man einen maximalen Fluss sowie einen kürzesten Weg von der Quelle $q = 1$ zu den übrigen Knoten.

19. Zeige[29], dass ein bipartiter Graph $G = (U \cup W, E)$ mit $|U| = |W| = n$ und $|E| > (m-1)n$ ein Matching der Größe m enthält.

[29]Diese Aufgabe haben wir M. Aigner (1996, S. 147) entnommen.

20. Sei $A = (a_{ij}) \in \mathbb{R}^{n\times n}$ eine Matrix mit $a_{ij} \in \{0,1\}$, $1 \le i,j \le n$. Unter einer *Linie* in einer Matrix verstehe man eine Zeile oder eine Spalte. Man zeige, dass die Minimalzahl der alle Einsen enthaltenden Linien gleich der Maximahlzahl der Einsen ist, von denen nicht zwei auf einer Linie liegen.

    Beispiel: In
    $$\begin{pmatrix} 1 & 1 & 0 & 1 \\ 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix}$$
    sind alle Einsen in drei Linien (etwa erste beide Zeilen, zweite Spalte) enthalten, während es maximal drei Einsen gibt, von denen keine zwei auf einer Linie liegen, z. B. die Einsen in den Positionen $(1,4)$, $(2,1)$ und $(4,2)$.

    Hinweis: Man wende Satz 8.5 auf einen geeignet definierten bipartiten Graphen an.

21. In einem Obstkorb liegen ein Apfel, eine Banane, ein Pfirsich und eine Mango. Vier Kindern, nämlich Erika, Frank, Gisela und Hubert, soll jeweils eine Frucht gegeben werden. Erika mag Pfirsiche und Mangos, Frank mag Äpfel und Pfirsiche, Gisela Bananen und Pfirsiche, Hubert mag Äpfel, Bananen und Mangos. Wie sollte das Obst verteilt werden?

22. Man löse das Zuordnungsproblem mit der Kostenmatrix
    $$C := \begin{pmatrix} 6 & 3 & 2 & 5 & 2 & 2 \\ 8 & 4 & 3 & 2 & 3 & 4 \\ 4 & 4 & 3 & 4 & 4 & 5 \\ 2 & 7 & 4 & 5 & 4 & 3 \\ 8 & 6 & 4 & 6 & 6 & 6 \\ 0 & 2 & 3 & 3 & 2 & 2 \end{pmatrix}.$$

23. Sei $G = (V,E)$ ein zusammenhängender, planarer Graph mit $n := |V| \ge 3$. Dann existiert eine Ecke mit einem Grad ≤ 5.

    Hinweis: Mache einen Widerspruchsbeweis und zähle (wie im Handshaking-Lemma) die Anzahl der Elemente in $P := \{(x,e) \in V \times E : x \in e\}$.

24. Ist der in Abbildung 3.55 angegebene Graph planar?

25. Sei $G = (V,E)$ ein zusammenhängender, planarer Graph mit $n := |V| \ge 5$ Ecken, $m := |E|$ Kanten, ferner habe jeder Kreis eine Länge ≥ 5. Man zeige:

    (a) Es ist $m \le \frac{5}{3}(n-2)$.

    (b) Der in Abbildung 3.56 dargestellte Petersen-Graph ist nicht planar.

    (c) Man bestimme eine Eckenfärbung des Petersen-Graphen mit möglichst wenig Farben, wobei natürlich benachbarte Ecken unterschiedliche Farben haben sollen.

26. Sei $G = (V,E)$ ein Graph mit $d := \max_{x\in V} d(x)$, wobei $d(x)$ den Grad der Ecke $x \in V$ bedeutet. Dann ist die chromatische Zahl von G kleiner oder gleich $d+1$. D. h. es ist eine Färbung der Ecken von G durch höchstens $d+1$ Farben derart möglich, dass benachbarte Ecken verschieden gefärbt sind.

    Hinweis: Man mache einen Induktionsbeweis nach der Anzahl n der Ecken von G und schließe wie beim Beweis des Sechsfarbensatzes.

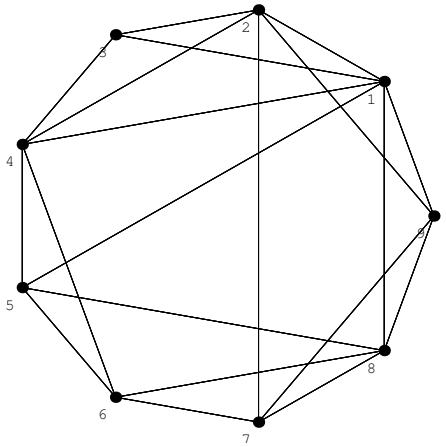


Abbildung 3.55: Ist dieser Graph planar?

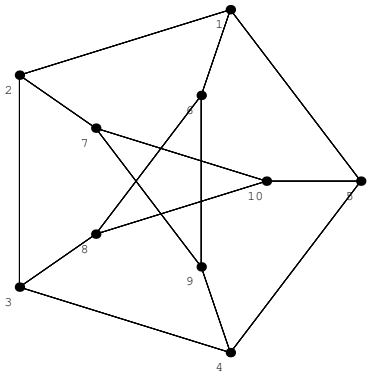


Abbildung 3.56: Der Petersen-Graph ist nicht planar

27. Ein Zoo muss geeignete Umgebungen für acht Tierarten $A, B, \ldots, H$ bestimmen, wobei aus Sicherheitsgründen gewisse Tiere von anderen getrennt werden müssen. Es soll die minimale Anzahl der "Umgebungen" bestimmt werden, in denen die Tiere sicher leben können. In der folgenden Tabelle gibt ein $*$ eine notwendige Trennung an.

	A	B	C	D	E	F	G	H
A	–	$*$	–	–	$*$	$*$	–	$*$
B	$*$	–	$*$	–	–	$*$	–	$*$
C	–	$*$	–	$*$	–	$*$	$*$	$*$
D	–	–	$*$	–	$*$	$*$	$*$	–
E	$*$	–	–	$*$	–	$*$	$*$	–
F	$*$	$*$	$*$	$*$	$*$	–	–	–
G	–	–	$*$	$*$	$*$	–	–	$*$
H	$*$	$*$	$*$	–	–	–	$*$	–

Man stelle den zugehörigen Graphen auf und bestimme dessen chromatische Zahl und eine geeignete Verteilung der Tiere. Hierbei kann *Mathematica* und das Zusatzpaket `DiscreteMath‘Combinatorica‘` benutzt werden.

Kapitel 4

Netzwerkflussprobleme

In Abschnitt 3.7 hatten wir definiert, was wir unter einem gerichteten Graphen bzw. einem Digraphen $(\mathcal{N}, \mathcal{A})$ verstehen. Die Definition eines *Netzwerkes* ist in der Literatur leider nicht einheitlich. Bei K. Neumann, M. Morlock (1993, S. 187) ist ein Netzwerk ein bewerteter Digraph ohne isolierte Knoten, bei R. Fletcher (1987, S. 345) ein (schwach oder stark: nicht ganz klar) zusammenhängender Digraph. Bei I. M. Bomze, W. Grossmann (1993, S. 59)[1] ist ein Netzwerk ein 5-Tupel bestehend aus einem Digraphen $(\mathcal{N}, \mathcal{A})$, speziellen Knoten $s, t \in \mathcal{N}$, nämlich der Quelle s mit $d^+(s) = 0$ (kein Pfeil endet in s) und der *Senke* t mit $d^-(t) = 0$ (kein Pfeil startet in t) und schließlich Kapazitätschranken auf den Pfeilen, also einer Funktion $b\colon \mathcal{A} \longrightarrow \mathbb{R}_{\geq 0}$. Genau diese Definition findet man auch bei C. H. Papadimitriou, K. Steiglitz (1982, S. 23) und M. Aigner (1996, S. 130). Wir werden der pragmatischen Vorgehensweise von D. P. Bertskas (1998)[2] folgen und gar nicht definieren, was ein Netzwerk ist, sondern nur erklären, was ein Netzwerkfluss-Problem ist.

4.1 Netzwerkfluss-Probleme: Beispiele

4.1.1 Das Minimale-Kosten-Fluss-Problem

Die zugrundeliegende Aufgabe kann man sich folgendermaßen vorstellen: Ein Produkt wird in gewissen Orten in einer bestimmten Menge angeboten und an anderen Orten verlangt. Schließlich gibt es Orte, die nichts anbieten und nichts verlangen, in denen aber umgeladen werden darf. Gewisse Orte sind miteinander durch Verkehrswege miteinander verbunden. Die Kosten für den Transport einer Mengeneinheit des Gutes längs eines Verkehrsweges sind bekannt, ferner ist die Kapazität eines jeden möglichen Transportweges vorgegeben. Diese gibt Obergrenzen für die zu transportierende Menge auf dem Weg an. Gesucht ist ein kostenminimaler Transportplan. Wir werden gleich ein mathematisches Modell für diese Aufgabenstellung angeben und in den folgenden Unterabschnitten auf Spezialfälle eingehen.

[1] I. M. Bomze, W. Grossmann (1993) *Optimierung–Theorie und Algorithmen. Eine Einführung in Operations Research für Wirtschaftsinformatiker*. BI Wissenschaftsverlag, Mannheim.

[2] D. P. Bertskas (1998) *Network Optimization. Continuous and Discrete Models*. Athena Scientific, Belmont, Massachusetts.

Gegeben sei ein gerichteter Graph bzw. Digraph $(\mathcal{N}, \mathcal{A})$. Mit jedem Knoten $k \in \mathcal{N}$ ist eine (i. allg. ganzzahlige) Mengenangabe b_k des im Digraphen zu transportierenden Gutes verbunden. Ist $b_k > 0$, so sind b_k Mengeneinheiten dieses Gutes im Knoten k vorhanden und Knoten k wird ein *Angebotsknoten* genannt. Ist dagegen $b_k < 0$, so werden dort $|b_k|$ Mengeneinheiten benötigt, man spricht von einem *Bedarfsknoten*. Im Fall $b_k = 0$ handelt es sich um einen *Umladeknoten*.

Zu jedem Pfeil $(i, j) \in \mathcal{A}$ des Digraphen gehören die Kosten c_{ij} für den Fluss einer Mengeneinheit auf ihm. Mit x_{ij} wird der Fluss auf diesem Pfeil bezeichnet, die *Kapazität* des Pfeils wird durch (i. allg. ganzzahliges) $u_{ij} > 0$ angegeben. Gesucht wird ein Fluss im Digraphen, der unter Berücksichtigung der Kapazitätsbeschränkungen die Angebote und "Bedarfe" mengenmäßig ausgleicht und die dafür erforderlichen Kosten minimiert. Dabei ist in jedem Knoten der Fluss zu erhalten. Dies bedeutet für den Knoten $k \in \mathcal{N}$, dass die Summe der Flüsse auf seinen eingehenden Pfeilen plus der in ihm verfügbaren (wenn k ein Angebotsknoten) beziehungsweise minus der von ihm benötigten (wenn k ein Bedarfsknoten) Menge $|b_k|$ gleich der Summe der Flüsse auf seinen ausgehenden Pfeilen ist. Die Flusserhaltungsbedingung für den Knoten k lautet daher

$$\sum_{i:(i,k)\in\mathcal{A}} x_{ik} + b_k = \sum_{j:(k,j)\in\mathcal{A}} x_{kj}.$$

Das kapazitierte lineare Netzwerkflussproblem lässt sich daher wie folgt formulieren:

$$\begin{cases} \text{Minimiere} \quad \displaystyle\sum_{(i,j)\in\mathcal{A}} c_{ij}x_{ij} \\ \text{unter den Nebenbedingungen} \\ \displaystyle\sum_{j:(k,j)\in\mathcal{A}} x_{kj} - \sum_{i:(i,k)\in\mathcal{A}} x_{ik} = b_k \qquad (k \in \mathcal{N}), \qquad 0 \le x_{ij} \le u_{ij} \qquad ((i,j) \in \mathcal{A}). \end{cases}$$

Diese Aufgabe wollen wir nun in Matrix-Vektorschreibweise formulieren, wobei wir die Bezeichnungen gegenüber denen in Abschnitt 3.7 geringfügig verändern. Dies kann folgendermaßen geschehen. Der Fluss $x = (x_{ij})$ hat so viele Komponenten wie es Pfeile gibt, ihre Anzahl sei $n := |\mathcal{A}|$. Es liegt also nahe, $\mathcal{A}$ durchzunummerieren. Es sei etwa $\mathcal{A} = \{l_1, \ldots, l_n\}$ mit $l_p = (i_p, j_p)$, $p = 1, \ldots, n$. Dann kann $x = (x_{ij})_{(i,j)\in\mathcal{A}}$ als Vektor $x = (x_1, \ldots, x_n)^T$ mit $x_p = x_{i_p j_p}$, $p = 1, \ldots, n$, geschrieben werden, entsprechendes gilt für die Kosten $c = (c_{ij})$ und Kapazitäten $u = (u_{ij})$. Ist ferner $m := |\mathcal{N}|$ die Anzahl der Knoten, so kann man $(b_k)_{k\in\mathcal{N}}$ zu einem Vektor $b = (b_1, \ldots, b_m)^T$ zusammenfassen. Definiert[3] man die *Knoten-Pfeil-Inzidenzmatrix* $A = (a_{kp}) \in \mathbb{R}^{m\times n}$ durch

$$a_{kp} := \begin{cases} +1, & \text{falls} \quad k = i_p, \\ -1, & \text{falls} \quad k = j_p, \\ 0 & \text{sonst}, \end{cases}$$

[3]Wir haben früher schon darauf hingewiesen, dass man bei der Definition der Inzidenzmatrix zu einem gerichteten Graphen aufpassen muss, weil gelegentlich genau das Negative unserer Definition als Inzidenzmatrix bezeichnet wird.

so erkennt man, dass obiges Netzwerkflussproblem, das sogenannte (kapazitierte) Minimale-Kosten-Fluss-Problem, in der Form

$$\text{(P)} \qquad \text{Minimiere} \quad c^T x \quad \text{auf} \quad M := \{x \in \mathbb{R}^n : 0 \le x \le u,\ Ax = b\}$$

geschrieben werden kann.

Beispiel: Wir betrachten einen Digraphen, der in Abbildung 4.1 gegeben ist. Wir

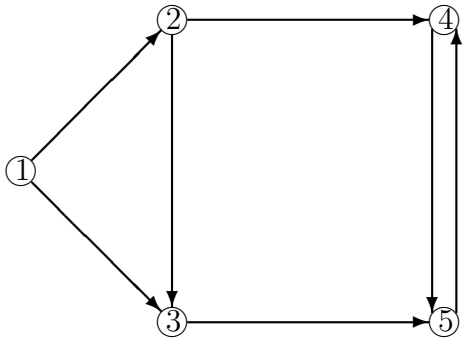


Abbildung 4.1: Ein Digraph mit 5 Knoten und 7 Pfeilen

wollen die zugehörige Knoten-Pfeil-Inzidenzmatrix aufstellen. Die Nummerierung der Knoten sei dabei wie angegeben, während die Nummerierung der Pfeile der folgenden Tabelle zu entnehmen ist.

	$(1,2)$	$(1,3)$	$(2,3)$	$(2,4)$	$(3,5)$	$(4,5)$	$(5,4)$
1	1	1	0	0	0	0	0
2	−1	0	1	1	0	0	0
3	0	−1	−1	0	1	0	0
4	0	0	0	−1	0	1	−1
5	0	0	0	0	−1	−1	1

□

Am letzten Beispiel erkennt man, dass die Summe der Zeilen der (Knoten-Pfeil-) Inzidenzmatrix die Nullzeile ergibt. Dies gilt auch ganz allgemein, also nicht nur in diesem speziellen Beispiel. Denn jeder Pfeil startet und endet in genau einem Knoten. In jeder Spalte stehen also genau eine 1 und eine -1, sonst nur Nullen. Daher ist $A^T e = 0$ bzw. die Summe der Zeilen von A eine Nullzeile. Der Rang der Knoten-Pfeil-Inzidenzmatrix ist also $\le |\mathcal{N}| - 1$. Uns wird später interessieren, wann hier Gleichheit gilt und welche Spalten linear unabhängig sind. Siehe hierzu auch Aufgabe 2. Es stellt sich heraus, dass für einen (schwach) zusammenhängenden Digraphen die Knoten-Pfeil-Inzidenzmatrix den maximalen Rang $|\mathcal{N}| - 1$ besitzt und die Spalten, die zu einem $(\mathcal{N}, \mathcal{A})$ aufspannenden Baum gehören, linear unabhängig sind.

Notwendig für die Existenz einer Lösung von (P) ist natürlich, dass das Optimierungsproblem zulässig bzw. die Menge M der zulässigen Flüsse nichtleer ist. Angenommen, es ist $x \in M$ ein zulässiger Fluss. Dann ist

$$\sum_{k\in\mathcal{N}} b_k = \sum_{k\in\mathcal{N}} \Big(\sum_{j:(k,j)\in\mathcal{A}} x_{kj} - \sum_{i:(i,k)\in\mathcal{A}} x_{ik} \Big) = \sum_{(k,j)\in\mathcal{A}} x_{kj} - \sum_{(i,k)\in\mathcal{A}} x_{ik} = 0,$$

denn beide Summanden sind die Summe der Flüsse über alle Pfeile. Aus der Matrixschreibweise von (P) erkennt man das sogar noch schneller. Denn aus $Ax = b$ und $A^T e = 0$ folgt sofort $b^T e = 0$. Notwendig für die Existenz einer Lösung ist also, dass das Gesamtangebot gleich dem Gesamtbedarf ist. Ähnlich wie beim Transportproblem kann man den Fall, dass das Gesamtangebot größer als der Gesamtbedarf ist, dadurch auf den obigen Fall zurückführen, dass man einen Knoten und Pfeile von allen Angebotsknoten zu diesem neuen Knoten einführt. Dieser hat als Bedarf den Überschuss, ferner entstehen beim Transport zu ihm längs der neu eingefügten Pfeile keine Kosten.

Ist ein (kapazitiertes) Minimale-Kosten-Fluss-Problem (oder einfacher: Netzwerkflussproblem) zulässig, so ist es auch lösbar, da die Menge der zulässigen Flüsse kompakt ist (jedenfalls dann, wenn alle Kapazitätsschranken endlich sind). Sind andererseits die Kosten nichtnegativ und das Problem zulässig mit möglicherweise unendlichen Kapazitätsgrenzen, so folgt schon aus dem Existenzsatz der linearen Optimierung die Existenz einer Lösung. Interessanter werden Aussagen darüber sein, ob die Ganzzahligkeit der Vektoren b und u die Existenz einer ganzzahligen Lösung impliziert. Auch hierüber werden wir uns später Gedanken machen.

Beispiel: Natürlich kann auf ein gegebenes Minimale-Kosten-Fluss-Problem einfach ein zur Verfügung stehender "Solver" für lineare Optimierungsprobleme angewandt werden. Dabei wird aber natürlich die Struktur des Problems nicht berücksichtigt. Wir wollen das mit dem folgenden Beispiel verdeutlichen.

Gegeben sei der in Abbildung 4.2 angegebene Digraph, wobei rechts angegeben

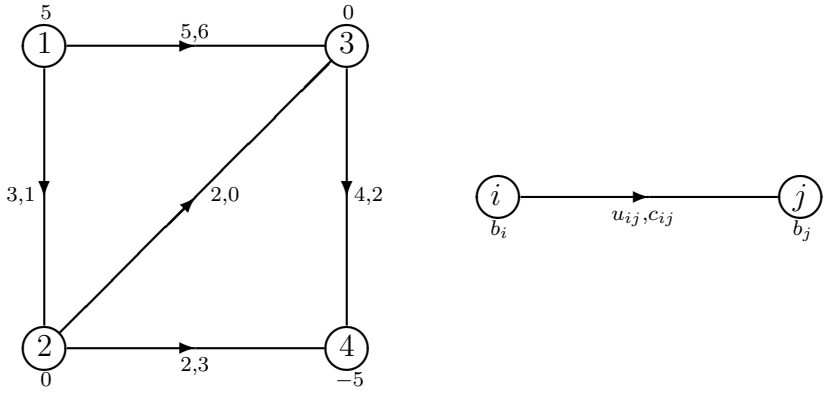


Abbildung 4.2: Ein Umladeproblem

ist, welche Bedeutung die angegebenen Zahlen haben. Z. B. sind die Knoten 2 und 3 Umladeknoten, der Knoten 1 ein Angebots- und der Knoten 4 ein Bedarfsknoten. Als Knoten-Pfeil-Inzidenzmatrix hat man

	$(1,2)$	$(1,3)$	$(2,3)$	$(2,4)$	$(3,4)$
1	1	1	0	0	0
2	−1	0	1	1	0
3	0	−1	−1	0	1
4	0	0	0	−1	−1

Das zugehörige lineare Programm lautet also

$$\text{Minimiere} \quad \begin{pmatrix} 1 \\ 6 \\ 0 \\ 3 \\ 2 \end{pmatrix}^T \begin{pmatrix} x_{12} \\ x_{13} \\ x_{23} \\ x_{24} \\ x_{34} \end{pmatrix} \quad \text{unter den Nebenbedingungen}$$

$$\begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} \le \begin{pmatrix} x_{12} \\ x_{13} \\ x_{23} \\ x_{24} \\ x_{34} \end{pmatrix} \le \begin{pmatrix} 3 \\ 5 \\ 2 \\ 2 \\ 4 \end{pmatrix}, \quad \begin{pmatrix} 1 & 1 & 0 & 0 & 0 \\ -1 & 0 & 1 & 1 & 0 \\ 0 & -1 & -1 & 0 & 1 \\ 0 & 0 & 0 & -1 & -1 \end{pmatrix} \begin{pmatrix} x_{12} \\ x_{13} \\ x_{23} \\ x_{24} \\ x_{34} \end{pmatrix} = \begin{pmatrix} 5 \\ 0 \\ 0 \\ -5 \end{pmatrix}.$$

Wir haben die `linprog` Funktion der Optimization-Toolbox von MATLAB benutzt. Nach

```
A=[1 1 0 0 0;-1 0 1 1 0;0 -1 -1 0 1;0 0 0 -1 -1];
b=[5;0;0;-5];c=[1;6;0;3;2];l=zeros(5,1);u=[3;5;2;2;4];
[x,minkosten]=linprog(c,[],[],A,b,l,u);
```

erhalten wir als Lösung (nach `format long`)

$$x = \begin{pmatrix} 2.99999999996060 \\ 2.00000000003940 \\ 1.99999999710308 \\ 1.00000000285752 \\ 3.99999999714248 \end{pmatrix}$$

mit den minimalen Kosten 26.00000000305450. Das Ergebnis (und das wird sich auch bei anderen Beispielen zeigen) ist enttäuschend ungenau, was daran liegen mag, dass per default in `linprog` eine primal-duale Innere-Punkt Methode angewandt wird. Erstaunlicherweise ist das Ergebnis wesentlich besser, wenn man am Schluss den Aufruf `[x,minkosten]=linprog(c,A,b,[],[],l,u)` macht, also eigentlich die Aufgabe

$$\text{Minimiere} \quad c^T x \quad \text{auf} \quad M := \{x : l \le x \le u,\ Ax \le b\}$$

löst. Die obige Vermutung wird bestätigt, wenn man

```
options=optimset('largescale','off')
```

setzt und anschließend z. B. den Aufruf

```
[x,minkosten]=linprog(c,[],[],A,b,l,u,zeros(5,1),options)
```

macht. Als Ergebnis erhält man in diesem Falle

$$x = \begin{pmatrix} 3.00000000000000 \\ 2.00000000000000 \\ 2.00000000000000 \\ 1.00000000000000 \\ 4.00000000000000 \end{pmatrix}$$

mit zugehörigen Kosten 25.99999999999998. □

4.1.2 Das Kürzester-Pfad-Problem

In einem Verkehrsnetz sei die kürzeste Verbindung zwischen zwei Orten zu bestimmen. Dieser Aufgabenstellung kann das folgende mathematische Modell zugeordnet werden. Gegeben sei ein Digraph $(\mathcal{N}, \mathcal{A})$, zwei Knoten $s, t \in \mathcal{N}$ mit $s \neq t$ seien ausgezeichnet. Jeder Pfeil $(i, j) \in \mathcal{A}$ trage Kosten (oder eine Länge) $c_{ij} \in \mathbb{R}$, die nicht notwendig nichtnegativ oder ganzzahlig ist. Gesucht sei ein kostengünstigster (oder kürzester) (gerichteter Vorwärts-) Pfad von s nach t. Hierbei verstehen wir unter einem *Pfad*[4] (gelegentlich der Deutlichkeit halber auch Vorwärts-Pfad genannt) von s nach t eine Folge von Pfeilen, bei denen der erste in s startet, der letzte in t endet und der Endknoten eines Pfeils Startknoten des nächsten Pfeils ist. Die Länge eines Pfades ist dabei natürlich die Summe der Längen (bzw. Kosten) der zugehörigen Pfeile.

Wir zitieren einige Sätze bei K. Neumann, M. Morlock (1993, S. 19): "Unfallrettung und Feuerwehr stehen ständig vor dem Problem, so schnell wie möglich zum jeweiligen Einsatzort zu gelangen. Für besonders wichtige oder gefährdete Objekte sind von den Rettungsdiensten vorsorglich geeignete Wege in dem vorliegenden Verkehrsnetz zu bestimmen, auf denen das Ziel in kürzest möglicher Zeit erreicht werden kann. An Stelle der Zeit kann auch die Weglänge als Zielkriterium verwendet werden. In diesem Verkehrsnetz sind neben Einbahnstraßenregelungen und Abbiegeverboten vor allem die Fahrzeiten auf den einzelnen Streckenabschnitten zu berücksichtigen, die verkehrsbedingt im Laufe des Tages stark schwanken können."

Sei P ein (Vorwärts-) Pfad von s nach t. Wir ordnen diesem Pfad durch

$$(*) \qquad x_{ij} := \begin{cases} 1, & \text{falls} \quad (i,j) \in P, \\ 0, & \text{falls} \quad (i,j) \notin P, \end{cases} \qquad (i,j) \in \mathcal{A},$$

einen Vektor $x \in \mathbb{R}^{|\mathcal{A}|}$ zu. Die Kosten (oder die Länge) des Pfades P sind dann gegeben durch $c(P) := \sum_{(i,j)\in\mathcal{A}} c_{ij} x_{ij}$. Weiter ist

$$\sum_{j:(k,j)\in\mathcal{A}} x_{kj} - \sum_{i:(i,k)\in\mathcal{A}} x_{ik} = \begin{cases} 1, & \text{falls} \quad k = s, \\ -1, & \text{falls} \quad k = t, \\ 0, & \text{sonst}, \end{cases} \qquad k \in \mathcal{N},$$

und

$$0 \leq x_{ij}, \qquad (i,j) \in \mathcal{A}.$$

[4] Auch bei dieser Definition müssen wir aufpassen. Wir werden später der Definition von D. P. Bertsekas (1998, S. 4) folgen. Hiernach ist ein *Pfad* P in einem gerichteten Graphen eine Folge von Knoten $(n_1, n_2, \ldots, n_k)$ mit $k \geq 2$ und eine zugehörige Folge von $k-1$ Pfeilen derart, dass der i-te Pfeil in der Folge entweder (n_i, n_{i+1}) (in diesem Fall wird er ein *Vorwärtspfeil* im Pfad genannt) oder (n_{i+1}, n_i) (in diesem Fall spricht man von einem *Rückwärtspfeil*) ist. Ein Pfad heißt ein *Vorwärtspfad*, wenn alle seine Pfeile Vorwärtspfeile sind. Entsprechendes gilt für einen Rückwärtspfeil. Ein *Zyklus* ist ein geschlossener Pfad, also ein Pfad, bei dem Start und Endknoten übereinstimmen. Ein Pfad heißt *einfach*, wenn er nur paarweise verschiedene Pfeile enthält und auch die Knoten paarweise verschieden sind mit der einzigen möglichen Ausnahme, dass Start- und Endknoten zusammenfallen (in welchem Fall der einfache Pfad ein *einfacher Zyklus* oder ein *Kreis* heißt).

Beim Kürzesten-Pfad-Problem spielen nur Vorwärtspfade eine Rolle, so dass wir hier und in Abschnitt 4.3 auf die explizite Nennung des Wortes "Vorwärts" i. Allg. verzichten werden.

Mit der Knoten-Pfeil-Inzidenzmatrix $A \in \mathbb{R}^{|\mathcal{N}|\times|\mathcal{A}|}$, dem Vektor $b \in \mathbb{R}^{|\mathcal{N}|}$ mit

$$b_k := \begin{cases} 1, & \text{falls} \quad k = s, \\ -1, & \text{falls} \quad k = t, \\ 0, & \text{sonst} \end{cases}$$

und dem Kostenvektor $c \in \mathbb{R}^{|\mathcal{A}|}$ (nach entsprechender Nummerierung der Knoten und Pfeile) kann dem Problem des kürzesten Pfades (KP) in $(\mathcal{N}, \mathcal{A})$ zwischen s und t das lineare Programm

(P) Minimiere $c^T x$ auf $M := \{x \in \mathbb{R}^{|\mathcal{A}|} : x \geq 0,\ Ax = b\}$

zugeordnet werden. Man beachte, dass dieses ein spezielles Minimale-Kosten-Fluss-Problem ist. Den Zusammenhang zwischen (KP) und (P) kann man folgendermaßen zusammenfassen:

- Besitzt (P) eine Lösung x der Form $(*)$, so ist der zugehörige Pfad P ein kürzester Pfad in $(\mathcal{N}, \mathcal{A})$ von s nach t.
- Besitzt (P) eine Lösung, so besitzt (P) auch eine Lösung der Form $(*)$ und der zugehörige Pfad ist ein kürzester Pfad in $(\mathcal{N}, \mathcal{A})$ von s nach t. Dies ist keineswegs trivial, wir werden in allgemeinerem Zusammenhang darauf zurückkommen.

Beispiel: Wir betrachten den in Abbildung 4.3 angegebenen Digraphen mit den ausgezeichneten Knoten s und t. Das zugehörige lineare Programm

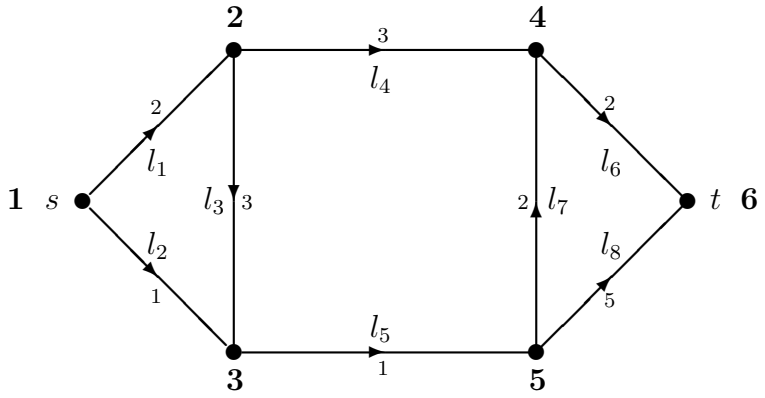


Abbildung 4.3: Was ist der kürzeste Pfad von s nach t?

(P) Minimiere $c^T x$ auf $M := \{x \in \mathbb{R}^8 : x \geq 0,\ Ax = b\}$

hat die Daten

$$\begin{array}{|c|c|} \hline c^T & \\ \hline A & b \\ \hline \end{array} :=$$

2	1	3	3	1	2	2	5	
1	1	0	0	0	0	0	0	1
−1	0	1	1	0	0	0	0	0
0	−1	−1	0	1	0	0	0	0
0	0	0	−1	0	1	−1	0	0
0	0	0	0	−1	0	1	1	0
0	0	0	0	0	−1	0	−1	−1

wobei die letzte Zeile noch gestrichen werden könnte, weil die entsprechende Gleichung redundant ist. Nach

```
A=[1 1 0 0 0 0 0 0 ;-1 0 1 1 0 0 0 0;0 -1 -1 0 1 0 0 0;
     0 0 0 -1 0 1 -1 0;0 0 0 0 -1 0 1 1;0 0 0 0 0 -1 0 -1];
b=[1;0;0;0;0;-1];c=[2;1;3;3;1;2;2;5];
l=zeros(8,1);u=Inf*ones(8,1);
[x,minkosten]=linprog(c,[],[],A,b,l,u);
```

erhalten wir als Lösung

$$x = \begin{pmatrix} 0.00000000000095 \\ 0.99999999999905 \\ 0.00000000000017 \\ 0.00000000000078 \\ 0.99999999999922 \\ 0.99999999999968 \\ 0.99999999999890 \\ 0.00000000000033 \end{pmatrix},$$

als zugehörige Kosten wird 6.00000000000180 ausgegeben. Sinngemäß gilt hier dasselbe wie in einem früheren Beispiel. Stellt man auf MediumScale um, so erhält man als Ergebnis

$$x = \begin{pmatrix} 0 \\ 1.00000000000000 \\ 0 \\ 0.00000000000000 \\ 1.00000000000000 \\ 1.00000000000000 \\ 1.00000000000000 \\ 0 \end{pmatrix}$$

mit den Kosten 6. Dies ergibt den kürzesten Pfad (l_2, l_5, l_7, l_6).

Das entsprechende Problem kann man auch mit *Mathematica* lösen. Nach dem Aufruf `«DiscreteMath‘Combinatorica‘` erhält man durch

```
adj={{0,2,1,0,0,0},{0,0,3,3,0,0},{0,0,0,0,1,0},
{0,0,0,0,0,2},{0,0,0,2,0,5},{0,0,0,0,0,0}};
gra=Graph[adj,Vertices[CompleteGraph[6]]];ShortestPath[gra,1,6]
```

den kürzesten Pfad $\{1, 3, 5, 4, 6\}$. Man kann natürlich auch den Algorithmus von Dijkstra anwenden und vom Knoten 1 ausgehend den zugehörigen minimalen aufspannenden (gerichteten) Baum berechnen. Mit *Mathematica* geschieht dies durch

```
tree=ShortestPathSpanningTree[gra,1];
ShowLabeledGraph[tree]
```

Das Ergebnis ist in Abbildung 4.4 angegeben. Beim Algorithmus von Dijkstra erhält der Knoten 1 den Wert 0. Danach wird der Pfeil $(1, 3)$ erzugt, der Knoten 3 bekommt

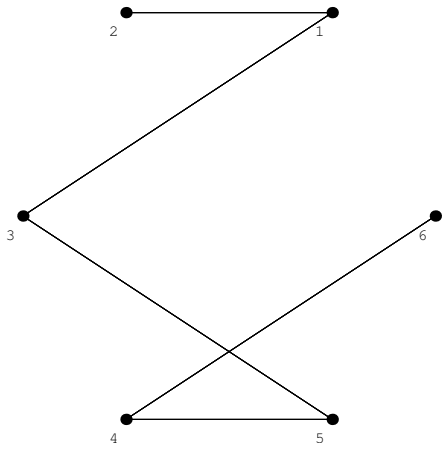


Abbildung 4.4: Ein minimaler aufspannender Baum

den Wert 1. Weiter kommt $(1,2)$ hinzu, der Knoten 2 wird mit 2 bewertet. Danach kommt der Pfeil $(3,5)$ hinzu, der Knoten 5 wird mit 2 bewertet. Im nächsten Schritt wird der Pfeil $(5,4)$ aufgenommen (es hätte auch $(2,4)$ gewählt werden können), der Knoten 4 erhält den Wert 4. Im letzten Schritt kommt $(4,6)$ hinzu, der Knoten 6 hat den Wert 6. □

4.1.3 Das Zuordnungsproblem

Die Aufgabe besteht darin, n Personen n Objekte eineindeutig zuzuordnen und zwar so, dass die Gesamtkosten der Zuordnung minimal sind. Also ist jeder Person genau ein Objekt, jedem Objekt genau eine Person zuzuordnen. Weiter sei $\mathcal{A} \subset \{1,\dots,n\} \times \{1,\dots,n\}$ die Menge der erlaubten Zuordnungen. D. h. Person i kann das Objekt j nur dann zugeordnet werden, wenn $(i,j) \in \mathcal{A}$. In diesem Falle seien c_{ij} die zugehörigen Kosten. Jeder (erlaubten) Zuordnung kann ein Vektor $x \in \mathbb{R}^{|\mathcal{A}|}$ dadurch zugeordnet werden, dass wir $x_{ij} := 1$ setzen, wenn Person i das Objekt j zugeordnet wir, andernfalls $x_{ij} := 0$, durch $\sum_{(i,j)\in\mathcal{A}} c_{ij}x_{ij}$ sind die Kosten dieser Zuordnung gegeben, diese sind unter allen erlaubten zu minimieren. Dieser Aufgabe kann mittels

$$\text{Minimiere} \quad \sum_{(i,j)\in\mathcal{A}} c_{ij}x_{ij}$$

unter den Nebenbedingungen

$$(*) \qquad \sum_{j:(i,j)\in\mathcal{A}} x_{ij} = 1 \quad (i=1,\dots,n), \qquad \sum_{i:(i,j)\in\mathcal{A}} x_{ij} = 1 \quad (j=1,\dots,n)$$

sowie

$$0 \le x_{ij} \le 1, \qquad (i,j) \in \mathcal{A},$$

eine lineare Optimierungsaufgabe zugeordnet werden, wobei natürlich die Kapazitätsbeschränkung $x_{ij} \le 1$, $(i,j) \in \mathcal{A}$, eigentlich redundant ist. Auch hier werden wir später sehen: Hat das lineare Programm eine zulässige Lösung, so hat es eine Lösung mit

$x_{ij} \in \{0,1\}$, $(i,j) \in \mathcal{A}$. Das Zuordnungsproblem kann als ein Minimale-Kosten-Fluss-Problem angesehen werden. Man hat einen bipartiten gerichteten Graphen mit $2n$ Knoten, die in zwei Gruppen von n Personen (Angebotsknoten) und n Objekten (Bedarfsknoten) zerfallen. Wenn der Person i das Objekt j zugeordnet werden kann, so ist $(i,j) \in \mathcal{A}$ ein Pfeil. Die Gleichungen $(*)$ können als Flussgleichungen interpretiert werden. Daher kann auch das Zuordnungsproblem als ein Minimale-Kosten-Fluss-Prroblem formuliert werden.

Beispiel: Es ist nicht sinnvoll, das Zuordnungsproblem als lineares Programm durch einen geeigneten Solver zu lösen. Gegeben sei z. B. ein Zuordnungsproblem[5] mit 5 Personen und 5 Objekten, bei der alle 5! Zuordnungen erlaubt sind und die Kostenmatrix durch

$$C := \begin{pmatrix} 4 & 2 & 6 & 1 & 3 \\ 5 & 8 & 3 & 7 & 1 \\ 9 & 4 & 6 & 3 & 7 \\ 2 & 5 & 4 & 2 & 3 \\ 3 & 3 & 7 & 4 & 4 \end{pmatrix}$$

gegeben ist. Nach

```
e=ones(5,1);o=zeros(5,1);
A=[e' o' o' o' o';o' e' o' o' o';o' o' e' o' o';
o' o' o' e' o';o' o' o' o' e';eye(5) eye(5) eye(5) eye(5) eye(5)];
b=[e;e];
c=[4;2;6;1;3;5;8;3;7;1;9;4;6;3;7;2;5;4;2;3;3;3;7;4;4];
l=zeros(25,1);u=ones(25,1);options=optimset('largescale','off');
[x,minkosten]=linprog(c,[],[],A,b,l,u,l,options);
```

erhält man die Lösung

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

mit den Kosten 13. Hier ist es sogar wichtig, dass man durch die Optionen kein Inneres-Punkt-Verfahren benutzt, weil man andernfalls keine Ecke und daher auch keine Permutationsmatrix als Lösung erhält (wie wir sehen werden, ist das Zuordnungsproblem zu obiger Kostenmatrix C nämlich nicht eindeutig lösbar). Aber wie schon gesagt, ist es sowieso nicht besonders gescheit, so wie eben beschrieben vorzugehen. Besser ist es, die ungarische Methode anzuwenden. Im Initialisierungsschritt erhält man die veränderten Kostenmatrizen

$$C = \begin{pmatrix} 3 & 1 & 5 & 0 & 2 \\ 4 & 7 & 2 & 6 & 0 \\ 6 & 1 & 3 & 0 & 4 \\ 0 & 3 & 2 & 0 & 1 \\ 0 & 0 & 4 & 1 & 1 \end{pmatrix}, \qquad C = \begin{pmatrix} 3 & 1 & 3 & 0 & 2 \\ 4 & 7 & 0 & 6 & 0 \\ 6 & 1 & 1 & 0 & 4 \\ 0 & 3 & 0 & 0 & 1 \\ 0 & 0 & 2 & 1 & 1 \end{pmatrix}.$$

[5] Siehe I. M. Bomze, W. Grossmann (1993, S. 377).

Hierbei sind schon Kosten von $12 = 1 + 1 + 3 + 2 + 3 + 2$ aufgelaufen. Eine minimale Überdeckung der Nullen in C ist durch $\{2, 4, 5\} \cup \{4\}$ gegeben (erstere beziehen sich auf die Zeilen, letztere auf die Spalten), das minimale unbedeckte Element ist $d = 1$. Von allen unbedeckten Elementen ist d zu substrahieren, zu allen doppelt bedeckten Elementen ist d zu addieren. Man erhält

$$C = \begin{pmatrix} 2 & 0 & 2 & \mathbf{0} & 1 \\ 4 & 7 & 0 & 7 & \mathbf{0} \\ 5 & 0 & \mathbf{0} & 0 & 3 \\ \mathbf{0} & 3 & 0 & 1 & 1 \\ 0 & \mathbf{0} & 2 & 2 & 1 \end{pmatrix} = \begin{pmatrix} 2 & \mathbf{0} & 2 & 0 & 1 \\ 4 & 7 & 0 & 7 & \mathbf{0} \\ 5 & 0 & 0 & \mathbf{0} & 3 \\ 0 & 3 & \mathbf{0} & 1 & 1 \\ \mathbf{0} & 0 & 2 & 2 & 1 \end{pmatrix},$$

wobei wir zwei maximale 0-Diagonale fett geduckt angedeutet haben. Die linke 0-Diagonale führt auf die oben angegebene Lösung, die rechte auf eine entsprechende andere Permutationsmatrix, beide haben die Kosten 13. Jede Konvexkombination der beiden Permutationsmatrizen ist natürlich auch eine Lösung. □

4.1.4 Das Maximaler-Fluss-Problem

Beim *Maximaler-Fluss-Problem* (*maximum flow problem*) ist ein gerichteter Graph $(\mathcal{N}, \mathcal{A})$ gegeben, in dem zwei Knoten s (Quelle, **s**ource, kein Pfeil endet in s) und t (Senke, **t**erminal, kein Pfeil startet in t) ausgezeichnet sind. Längs der Pfeile sind wieder Kapazitäten festgelegt. Es wird angenommen, dass es eine die Quelle s und die Senke t verbindenden (Vorwärts-) Pfad gibt. Es wird nach dem maximalen Fluss von s nach t gefragt, also nach der maximalen Anzahl der Mengeneinheiten, die bei s losgeschickt werden können und in t ankommen, wobei natürlich die Kapazitätsbeschränkungen zu berücksichtigen sind.

Wie kann nun das Maximaler-Fluss-Problem als ein Minimale-Kosten-Fluss-Problem geschrieben werden? Hierzu fassen wir alle Knoten als reine Umladeknoten auf, die Kosten längs jeden Pfeils werden auf Null gesetzt, es ist also $c_{ij} := 0$ für alle $(i, j) \in \mathcal{A}$. Ferner führe man einen Pfeil (t, s) von der Senke zur Quelle ein und definiere $c_{ts} := -1$ und $u_{ts} := +\infty$. Dieser Pfeil kann also beliebig viel aufnehmen. Die Aufgabe könnte also formuliert werden als:

$$\text{Minimiere} \quad -x_{ts}$$

unter den Nebenbedingungen

$$\sum_{j:(k,j)\in\mathcal{A}} x_{kj} - \sum_{i:(i,k)\in\mathcal{A}} x_{ik} = 0, \qquad k \in \mathcal{N} \setminus \{s, t\},$$

(alles was in einem Knoten $k \in \mathcal{N} \setminus \{s, t\}$ ankommt, fließt dort auch weg),

$$\sum_{j:(s,j)\in\mathcal{A}} x_{sj} = \sum_{i:(i,t)\in\mathcal{A}} x_{it} = x_{ts},$$

(alles was bei s wegfließt, kommt bei t an und fließt schließlich wieder (über den künstlichen Pfeil) nach t) sowie

$$0 \le x_{ij} \le u_{ij}, \qquad (i, j) \in \mathcal{A}.$$

Offenbar ist dann ein kostenminimaler Fluss ein Maximalfluss.

Etwas natürlicher ist es aber vielleicht, direkt die zum Maximaler-Fluss-Problem gehörende lineare Optimierungsaufgabe aufzustellen. Die Knoten seien so nummeriert, dass die Quelle s der erste und die Senke t der letzte bzw. der m-te Knoten ist. Mit $A = (a_{kp}) \in \mathbb{R}^{m\times n}$ bezeichnen wir wieder die Knoten-Pfeil-Inzidenzmatrix. Es ist also $a_{kp} = +1$, wenn der k-te Knoten Startknoten für den p-ten Pfeil ist, $a_{kp} = -1$, wenn der k-te Knoten Endknoten für den p-ten Pfeil ist, und $a_{kp} = 0$ in allen anderen Fällen. Ist $x = (x_1, \ldots, x_n)^T$ (hierbei bedeutet x_p den Fluss auf dem p-ten Pfeil), so ist $(Ax)_1 = \sum_{p=1}^{n} a_{1p}x_p$ der an der Quelle austretende Fluss. Diesen gilt es unter Berücksichtigung der Kapazitätsbeschränkungen auf den Pfeilen und der Flussbedingung $(Ax)_k = 0$, $k = 2, \ldots, m-1$, zu minimieren. Weiter ist $(Ax)_m = -(Ax)_1$ (Beweis?). Definiert man daher noch den Vektor $d \in \mathbb{R}^m$ durch

$$d_k := \begin{cases} -1 & \text{für} \quad k = 1, \\ 0 & \text{für} \quad k = 2, \ldots, m-1, \\ +1 & \text{für} \quad k = m, \end{cases}$$

so erkennt man, dass das Maximaler-Fluss-Problem als lineare Optimierungsaufgabe

$$\text{Maximiere} \quad v \quad \text{auf} \quad \{(x, v) \in \mathbb{R}^n \times \mathbb{R} : Ax + dv = 0,\ 0 \le x \le u\}$$

formuliert werden kann. Wieder ist es einleuchtend, dass zur Lösung dieses linearen Programms die spezielle Struktur ausgenutzt werden sollte.

Beispiel: In der folgenden Abbildung (siehe auch Abbildung 3.31) geben wir einen Digraphen mit 8 Knoten und 14 Pfeilen an, eingetragen sind ferner die Kapazitäten längs der Pfeile. Was ist der maximale Fluss? Klar ist, dass dieser nicht größer als 6 sein

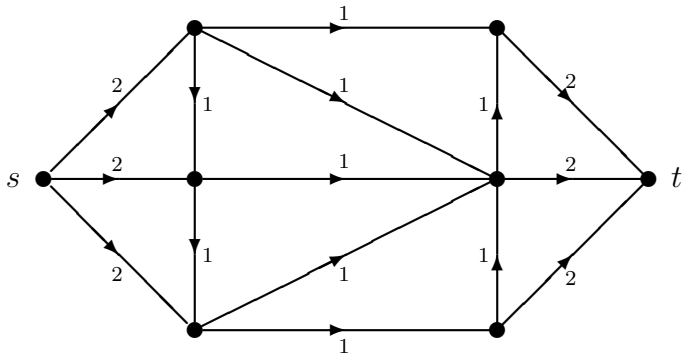


Abbildung 4.5: Ein Digraph mit 8 Knoten und 14 Pfeilen

kann, da die drei Pfeile weg von der Quelle nur eine Gesamtkapazität von 6 besitzen.

In der Abbildung 4.6 (siehe auch Abbildung 3.32) geben wir einen Fluss mit dem Wert 5 an. Gibt es auch einen mit dem Wert 6? Obiges lineares Programm wäre in diesem Falle eine Aufgabe mit 16 Variablen und 8 Gleichungsrestriktionen. Wenn man es mit einem normalen Solver, z. B. `linprog` in MATLAB, löst, erhält man nicht notwendig eine ganzzahlige Lösung. Z. B. erhält man mit `linprog` die in Abbildung 4.7

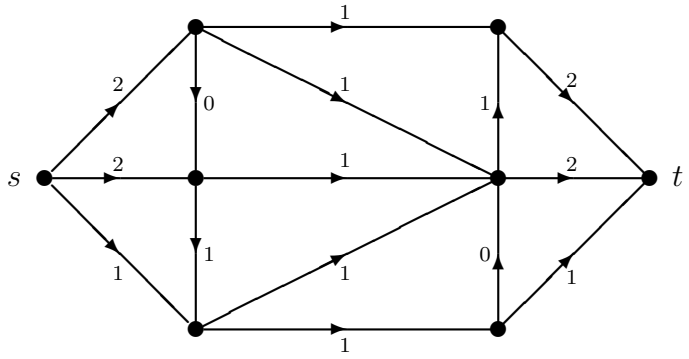


Abbildung 4.6: Ein Fluss mit dem Wert 5

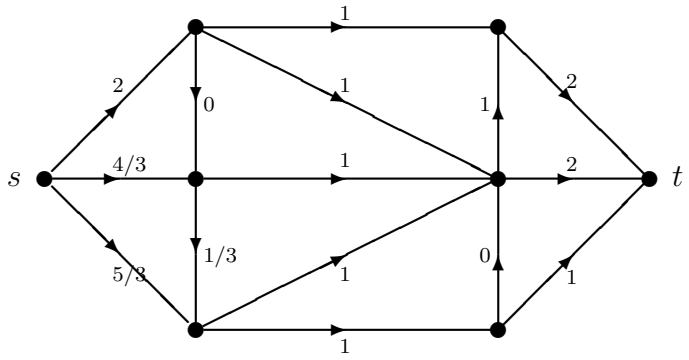


Abbildung 4.7: Ebenfalls ein Fluss mit dem Wert 5

Wieder liegt es an der Nichteindeutigkeit einer Lösung, dass es auch nicht-ganzzahlige Lösungen gibt (was allerdings noch bewiesen werden muss). Das obige Ergebnis haben wir durch

```
c=zeros(16,1);c(16)=-1;
A=[1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 -1;-1 0 0 1 0 1 1 0 0 0 0 0 0 0 0 0;
0 -1 0 -1 1 0 0 1 0 0 0 0 0 0 0 0;0 0 -1 0 -1 0 0 0 1 1 0 0 0 0 0 0;
0 0 0 0 0 -1 0 0 0 0 -1 0 1 0 0 0;0 0 0 0 0 0 -1 -1 -1 0 1 -1 0 1 0 0;
0 0 0 0 0 0 0 0 0 -1 0 1 0 0 1 0;0 0 0 0 0 0 0 0 0 0 0 0 -1 -1 -1 1];
b=zeros(8,1);l=zeros(16,1);l(16)=-inf;
u=[2;2;2;1;1;1;1;1;1;1;1;1;2;2;2;+inf];
options=optimset('LargeScale','off');
[x,minkosten]=linprog(c,[],[],A,b,l,u,zeros(16,1),options);
```

gewonnen. □

4.1.5 Das Transportproblem

Das Transportproblem haben wir früher schon geschildert. Der zu Grunde liegende Graph ist bipartit. Die Knoten zerfallen in m Lager (Angebotsknoten) und n Kunden (Bedarfsknoten), Pfeile gibt es nur von den Lagern zu den Kunden. Es ist nicht schwierig, das Transportproblem (ähnlich wie das Zuordnungsproblem) als ein Minimale-Kosten-Fluss-Problem zu schreiben. Die Flussgleichungen zerfallen, entsprechend dem Zerfall der Knoten, in zwei Anteile, solche für die Lager und solche für die Kunden:

$$\sum_{j:(i,j)\in\mathcal{A}} x_{ij} = l_i \quad (i = 1,\ldots,m), \qquad -\sum_{i:(i,j)\in\mathcal{A}} x_{ij} = -k_j \quad (j = 1,\ldots,n).$$

Den zweiten Block von Gleichungen werden wir i. Allg. mit -1 multiplizieren. Schreiben wir das Transportproblem also als lineares Optimierungsproblem in Normalform:

$$\text{Minimiere} \quad c^T x \quad \text{auf} \quad M := \{x \in \mathbb{R}^{mn} : x \geq 0,\ Ax = b\},$$

so ist A i. Allg. nicht genau die Knoten-Pfeil-Inzidenzmatrix zum bipartiten gerichteten Graphen. Ersetzt man die Restriktion $x \geq 0$ durch $0 \leq x \leq u$, so erhält man das *kapazitierte Transportproblem*.

Beispiel: Sei $m = 3$, $n = 4$. In Abbildung 4.8 haben wir den entsprechend Graphen

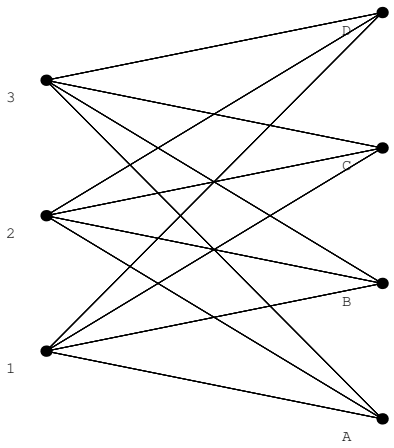


Abbildung 4.8: Ein bipartiter Graph

angegeben (weil es einfacher ist: als ungerichteten Graphen gezeichnet). Nummeriert man die Knoten in der Reihenfolge $1, 2, 3, A, B, C, D$ (als ert die Lager, dann die Kunden) und die Pfeile in der Reihenfolge $(1, A), (1, B), (1, C), \ldots, (3, D)$, so erhält man als Knoten-Pfeil-Inzidenzmatrix zu diesem gerichteten bipartiten Graphen die Matrix

$$A = \left(\begin{array}{cccc|cccc|cccc} 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ \hline -1 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & -1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & -1 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & -1 \end{array}\right).$$

Wie oben schon gesagt, werden wir -1 durch 1 ersetzen, wobei sich der Rang von A oder einer Untermatrix A_B nicht verändert, da es sich hierbei nur um eine Multiplikation mit der nichtsingulären Block-Diagonalmatrix diag $(I_m, -I_n)$ handelt. Es stellt sich die Frage, was der Rang von A ist und welche Spalten von A linear unabhängig sind. Dies wollen wir außerhalb dieses Beispiels genauer beantworten. □

Bei gebenen natürlichen Zahlen m und n sei

$$A := \begin{pmatrix} e^T & 0^T & \cdots & 0^T \\ 0^T & e^T & & 0^T \\ \vdots & \vdots & \ddots & \vdots \\ 0^T & 0^T & \cdots & e^T \\ I & I & \cdots & I \end{pmatrix}$$

(im wesentlichen) die Knoten-Pfeil-Inzidenzmatrix des vollständigen bipartiten Graphen $K_{m,n}$. Da $K_{m,n}$ zusammenhängend ist, ist der Rang von A gegeben durch Knotenzahl minus 1, also durch $m+n-1$, ferner sind die zu den Kanten (es kommt hier nicht auf die Richtung an) eines aufspannenden Baumes gehörenden Spalten von A linear unabhängig (siehe Aufgabe 2). In Abbildung 4.9 geben wir zu $K_{3,4}$ zwei aufspannende Bäume an. Die zugehörigen "Inzidenzmatrizen" sind

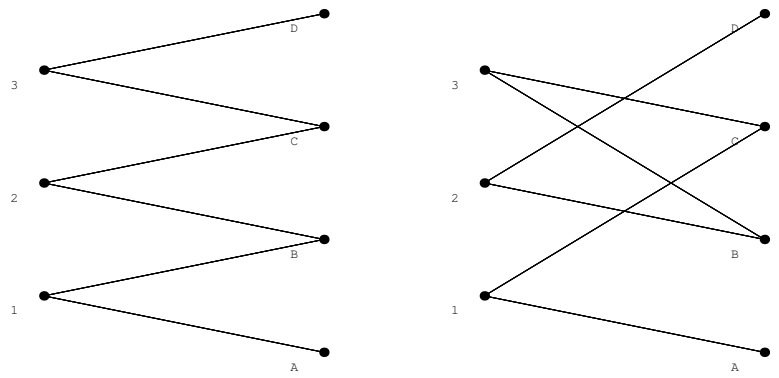


Abbildung 4.9: Zwei $K_{3,4}$ aufspannende Bäume

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

4.1.6 Aufgaben

1. Man betrachte den in Abbildung 4.10 dargestellten Digraphen $(\mathcal{N}, \mathcal{A})$. Nach einer geeigneten Nummerierung der Pfeile stelle man seine Knoten-Pfeil-Inzidenzmatrix auf. Man

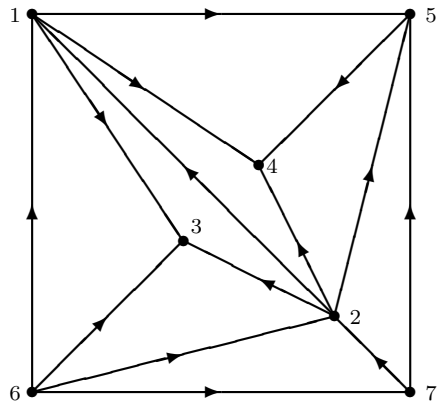


Abbildung 4.10: Ein gerichteter Graph

bestimme wenigstens zwei $(\mathcal{N}, \mathcal{A})$ (als ungerichteten Graphen betrachtet) aufspannende Bäume und deren Knoten-Pfeil-Inzidenzmatrix. Man zeige, dass diese nach Weglassen der letzten Zeile nichtsingulär sind.

2. Sei $(\mathcal{N}, \mathcal{A})$ ein (schwach) zusammenhängender Digraph. Sei $A \in \mathbb{R}^{m\times n}$ mit $m := |\mathcal{N}|$ und $n := |\mathcal{A}|$ (nach geeigneter Nummerierung der Knoten und Pfeile) die zugehörige Knoten-Pfeil-Inzidenzmatrix. Man zeige, dass Rang $(A) = m - 1$. Genauer zeige man, dass die Spalten von A, die zu einem $(\mathcal{N}, \mathcal{A})$ aufspannenden Baum gehören, linear unabhängig sind.

3. Man[6] betrachte das Zuordnungsproblem, bei dem die Kostenmatrix $C = (c_{ij})$ durch $c_{ij} = a_i b_j$ (also $C = ab^T$) gegeben ist, wobei

$$a_1 \le \cdots \le a_n, \qquad b_n \le \cdots \le b_1.$$

Man zeige, dass durch die Identität $X = I$ (bzw. $x_{ij} = \delta_{ij}$) eine optimale Zuordnung gegeben ist. Unter welcher zusätzlichen Bedingung ist diese eindeutig?

4.2 (Total) Unimodulare Matrizen

In diesem Abschnitt wollen wir untersuchen, unter welchen Voraussetzungen aus der Ganzzahligkeit (gewisser) der Daten einer linearen Optimierungsaufgabe die Existenz[7] einer ganzzahligen Lösung folgt. Sei z. B. ein lineares Programm in Standardform gegeben, also

(P) Minimiere $c^T x$ auf $M := \{x \in \mathbb{R}^n : x \ge 0,\ Ax = b\}$.

[6] Bei

D. P. Bertsekas (1998) *Network Optimization. Continuous and Discrete Models*. Athena Scientific, Belmont

wird die Aussage dieser Aufgabe Hardy zugeschrieben.

[7] Man kann nicht erwarten, dass *jede* Lösung ganzzahlig ist. Denn ist die Aufgabe nicht eindeutig lösbar, so ist eine Konvexkombination von zwei ganzzahligen Lösungen eine i. Allg. nicht ganzzahlige Lösung.

Wenn man weiß, dass jede Ecke von M ganzzahlig ist, so folgt aus der Existenz einer Lösung von (P) auch die Existenz einer ganzzahligen Lösung. Daher interessiert insbesondere, welche Eigenschaften einer Matrix A garantieren, dass für alle $b \in \mathbb{Z}^m$ jede Ecke von

$$M(b) := \{x \in \mathbb{R}^n : x \geq 0,\ Ax = b\}$$

ganzzahlig ist.

Wir beginnen mit einem einfachen Lemma.

Lemma 2.1 *Sei $A \in \mathbb{Z}^{m\times n}$ nichtsingulär. Dann ist $A^{-1}b \in \mathbb{Z}^m$ für alle $b \in \mathbb{Z}^m$ genau dann, wenn* $\det(A) = 1$ *oder* $\det(A) = -1$.

Beweis: Angenommen, es sei $\det(A) = \pm 1$. Aus der Cramerschen Regel folgt $A^{-1} \in \mathbb{Z}^{m\times m}$ woraus trivialerweise $A^{-1}b \in \mathbb{Z}^m$ für jedes $b \in \mathbb{Z}^m$ folgt. Umgekehrt nehmen wir an, es sei $A^{-1}b \in \mathbb{Z}^m$ für alle $b \in \mathbb{Z}^m$. Insbesondere ist $A^{-1}e_i \in \mathbb{Z}^m$, $i = 1, \ldots, m$, folglich $A^{-1} \in \mathbb{Z}^{m\times m}$. Daher sind $\det(A)$ und $\det(A^{-1})$ beides ganze Zahlen. Wegen $\det(A)\det(A^{-1}) = 1$ folgt $\det(A) = \pm 1$. □

Nun stellt sich naheliegenderweise die folgende Frage: Welche Eigenschaften von $A \in \mathbb{Z}^{m\times n}$ garantieren, dass alle Ecken von $M(b) := \{x \in \mathbb{R}^n : x \geq 0,\ Ax = b\}$ für alle $b \in \mathbb{Z}^m$ ganzzahlig sind? Da man notfalls redundante Gleichungen streichen kann, können wir annehmen, dass Rang $(A) = m$. Die entscheidende Definition ist

Definition 2.2 Eine Matrix $A \in \mathbb{Z}^{m\times n}$ mit Rang $(A) = m$ heißt *unimodular*, wenn die Determinante jeder nichtsingulären $m \times m$-Untermatrix 1 oder -1 ist.

Der folgende Satz ist eine einfache Folgerung früherer Ergebnisse zu Ecken von $M := \{x \in \mathbb{R}^n : x \geq 0,\ Ax = b\}$ bzw. zulässigen Basislösungen.

Satz 2.3 *Sei $A \in \mathbb{Z}^{m\times n}$ mit* Rang $(A) = m$ *gegeben. Für jedes $b \in \mathbb{Z}^m$ ist jede Ecke von $M(b) := \{x \in \mathbb{R}^n : x \geq 0,\ Ax = b\}$ ganzzahlig genau dann, wenn A unimodular ist.*

Beweis: Sei A unimodular, $b \in \mathbb{Z}^m$ und $x \in M(b)$ eine Ecke. Dann gibt es eine Basis-Indexmenge $B \subset \{1, \ldots, n\}$ mit $\#(B) = m$ derart, dass $A_B \in \mathbb{Z}^{m\times m}$ nichtsingulär ist, der Basisanteil x_B durch $x_B = A_B^{-1}b$ gegeben ist und der Nichtbasisanteil x_N verschwindet. Da A unimodular ist wegen des obigen Lemmas $x_B \in \mathbb{Z}^m$ und insgesamt $x \in \mathbb{Z}^n$.

Umgekehrt sei jede Ecke von $M(b)$ für alle $b \in \mathbb{Z}^m$ ganzzahlig. Sei $B \subset \{1, \ldots, n\}$ eine Indexmenge mit $\#(B) = m$ und der Eigenschaft, dass A_B nichtsingulär ist. Um die Unimodularität von A zu zeigen, genügt wegen obigen Lemmas der Nachweis dafür, dass $A_B^{-1}\hat{b} \in \mathbb{Z}^m$ für alle $\hat{b} \in \mathbb{Z}^m$. Sei daher $\hat{b} \in \mathbb{Z}^m$ beliebig vorgegeben. Man bestimme $u \in \mathbb{Z}^m$ (hinreichend groß) mit $u + A_B^{-1}\hat{b} \geq 0$ und setze $b := A_B(u + A_B^{-1}\hat{b})$. Man beachte, dass $b \in \mathbb{Z}^m$. Definiert man $z = (z_B, z_N)$ durch $z_B := u + A_B^{-1}\hat{b}$, $z_N := 0$, so ist z eine Ecke von $M(b)$ und daher ganzzahlig. Da u nach Voraussetzung ganzzahlig ist, ist es auch $A_B^{-1}\hat{b}$. Wegen des letzten Lemmas ist $\det(A_B) = \pm 1$ und damit A unimodular. Das einfache Lemma ist bewiesen. □

Die folgende Definition ist wichtig.

Definition 2.4 Eine Matrix $A \in \mathbb{Z}^{m\times n}$ heißt *total unimodular*, wenn die Determinante jeder quadratischen Untermatrix von A gleich 1, -1 oder 0 ist.

Insbesondere sind die Einträge einer total unimodularen Matrix 1, -1 oder 0. Ist weiter $A \in \mathbb{Z}^{m\times n}$ total unimodular und Rang $(A) = m$, so ist A erst recht unimodular (sonst wäre die Bezeichnung *total* auch nicht gerechtfertigt). Ist umgekehrt $A \in \mathbb{Z}^{m\times n}$ unimodular und Rang $(A) = m$, so ist A nicht notwendig total unimodular. Denn

$$A := \begin{pmatrix} 1 & 0 & 0 & -1 & 0 & 0 \\ 2 & 1 & 0 & -2 & -1 & 0 \\ 2 & 2 & 1 & -2 & -2 & -1 \end{pmatrix}$$

ist z. B. sicher nicht total unimodular. Aber A hat den Rang 3 und die Determinante jeder nichtsingulären 3×3-Untermatrix ist ± 1. Auch die Matrix

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

(sie hat die Determinante -1) ist ein Beispiel einer unimodularen, aber nicht total unimodularen Matrix (man sehe sich den linken oberen 2×2-Block an).

Es gilt der folgende Satz, der oft nach Hoffman-Kruskal benannt wird (siehe z. B. W. J. Cook, W. H. Cunningham, W. R. Pulleyblank, A. Schrijver (1998, S. 219)[8]). Vorher beweisen wir aber ein Lemma, in welchem der Kern des Beweises des Satzes von Hoffman-Kruskal enthalten ist.

Lemma 2.5 *Sei $A \in \mathbb{Z}^{m\times n}$. Dann ist A genau dann total unimodular, wenn $\hat{A} := (\ A \quad I\ ) \in \mathbb{Z}^{m\times(n+m)}$ unimodular ist.*

Beweis: Sei A total unimodular und $\hat{A}_B$ eine $m \times m$ Untermatrix von $\hat{A}$, welche aus $\hat{A}$ durch Zusammensetzen der Spalten zu Indizes aus $B \subset \{1, \ldots, n+m\}$ entsteht, wobei B genau m Elemente enthält. Dann ist $\hat{A}_B = (\ A_C \quad I_D\ )$ mit $C \subset \{1, \ldots, n\}$, $D \subset \{1, \ldots, m\}$ und $B = C \cup (n + D)$. Es existiert eine $m \times m$-Permutationsmatrix P derart, dass

$$P\hat{A}_B = \begin{pmatrix} A_{11} & 0 \\ A_{21} & I \end{pmatrix},$$

wobei A_{11} eine (eventuell in den Zeilen permutierte) quadratische Untermatrix von A ist. Da A nach Voraussetzung total unimodular ist, ist $\det(\hat{A}_B) = \pm\det(A_{11}) \in \{0, 1, -1\}$. Wenn also $\hat{A}_B$ nichtsingulär ist, so ist $\det(\hat{A}_B) = \pm 1$ und folglich $\hat{A} = (\ A \quad I\ )$ unimodular[9].

[8] W. J. Cook, W. H. Cunningham, W. R. Pulleyblank, A. Schrijver (1998) *Combinatorial Optimization*. J. Wiley, New York.

[9] Es gilt sogar mehr: Ist $A \in \mathbb{Z}^{m\times n}$ total unimodular, so ist $(\ A \quad I\ )$ total unimodular. Denn man überlegt sich leicht: Ist $A \in \mathbb{Z}^{m\times n}$ und e_j der j-te Einheitsvektor im $\mathbb{R}^m$, so ist $(\ A \quad e_j\ )$ total unimodular. Sukzessives Anfügen der Einheitsvektoren liefert also, dass $(\ A \quad I\ )$ total unimodular ist. Wegen Rang $(\ A \quad I\ ) = m$, ist $(\ A \quad I\ )$ natürlich auch unimodular.

Umgekehrt sei $\hat{A} = (\ A \quad I\ )$ unimodular und A_{11} eine nichtsinguläre, quadratische Untermatrix von A. Dann ist

$$\begin{pmatrix} A_{11} & 0 \\ * & I \end{pmatrix}$$

eine $m \times m$-Untermatrix von $\hat{A}$ und folglich $\det(A_{11}) = \pm 1$. Damit ist das Lemma bewiesen. □

Satz 2.6 *Sei $A \in \mathbb{Z}^{m\times n}$. Dann ist A genau dann total unimodular, wenn für jedes $b \in \mathbb{Z}^m$ jede Ecke von*

$$M'(b) := \{x \in \mathbb{R}^n : x \geq 0,\ Ax \leq b\}$$

ganzzahlig ist.

Beweis: Sei A total unimodular, $b \in \mathbb{Z}^m$ und x eine Ecke von $M'(b)$. Eine einfache Überlegung (siehe Aufgabe 1) zeigt, dass

$$\begin{pmatrix} x \\ b - Ax \end{pmatrix}$$

eine Ecke von

$$M(b) := \{z \in \mathbb{R}^{n+m} : z \geq 0,\ (\ A \quad I\ )z = b\}$$

ist. Da $(\ A \quad I\ )$ wegen Lemma 2.5 unimodular ist, folgt aus Satz 2.3, dass diese Ecke von $M'(b)$ und daher erst recht x ganzzahlig ist.

Umgekehrt sei für jedes $b \in \mathbb{Z}^m$ jede Ecke von $M'(b)$ ganzzahlig. Mit obigen Bezeichnungen ist für jedes $b \in \mathbb{Z}^m$ jede Ecke von $M(b)$ ganzzahlig. Wiederum aus Satz 2.3 folgt, dass $\hat{A} := (\ A \quad I\ )$ unimodular ist. Aus Lemma 2.5 erhält man, dass A total unimodular ist. Der Satz ist damit bewiesen. □

Die Bedeutung total unimodularer Matrizen wird auch durch den folgenden Satz unterstrichen.

Satz 2.7 *Sei $A \in \mathbb{Z}^{m\times n}$ total unimodular und $b \in \mathbb{Z}^m$. Dann ist jede Ecke des Polyeders $P := \{x \in \mathbb{R}^n : Ax \leq b\}$ ganzzahlig.*

Beweis: Siehe W. J. Cook et al. (1998, S. 221), A. Schrijver (1986, S. 266). □

Natürlich interessieren hinreichende Bedingungen dafür, dass eine Matrix $A \in \mathbb{Z}^{m\times n}$ (total) unimodular ist. Eine *notwendige* Bedingung ist natürlich, dass bei einer total unimodularen Matrix nur die Einträge $0, +1$ und -1 auftreten können. Diese Bedingung ist aber sicher nicht hinreichend, denn die Matrix

$$A = \left(\begin{array}{ccc|c} 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 \\ 0 & 1 & 1 & 0 \\ \hline 1 & 1 & 0 & 1 \end{array}\right)$$

ist nicht total unimodular, da die angegebene 3×3-Untermatrix die Determinante -2 besitzt.

Wir geben den folgenden Satz (siehe C. H. Papadimitriou, K. Steiglitz (1982, S. 317)) an, in dem einfache hinreichende Bedingungen für totale Unimodularität angegeben werden.

Satz 2.8 *Sei $A = (a_{ij}) \in \mathbb{Z}^{m\times n}$ mit $a_{ij} \in \{0, +1, -1\}$ für alle $(i, j) \in \{1, \ldots, m\} \times \{1, \ldots, n\}$. Dann ist A total unimodular, wenn in jeder Spalte von A nicht mehr als zwei von 0 verschiedene Einträge enthalten sind und die Zeilenindexmenge $I := \{1, \ldots, m\}$ so in zwei (nicht notwendig nichtleere) Indexmengen I_1 und I_2 partitioniert werden kann, dass die folgenden beiden Aussagen gelten:*

1. *Enthält eine Spalte zwei Einträge vom gleichen Vorzeichen, so sind die zugehörigen Zeilen in verschiedenenen Indexmengen. Für jede Spalte $j \in \{1, \ldots, n\}$ mit $a_{ij} = a_{kj} = +1$ oder $a_{ij} = a_{kj} = -1$ (hierbei sei natürlich $i, k \in \{1, \ldots, m\}$, $i \neq k$) ist also ($i \in I_1$ und $k \in I_2$) oder ($i \in I_2$ und $k \in I_1$).*

2. *Enthält eine Spalte zwei Einträge mit unterschiedlichem Vorzeichen, so gehören die zugehörigen Zeilen zur gleichen Indexmenge. Ist also $a_{ij} = +1$ und $a_{kj} = -1$, so sind $i, k \in I_1$ oder $i, k \in I_2$.*

Beweis: Wir haben zu zeigen, dass jede quadratische, nichtsinguläre Untermatrix C von A die Determinante $+1$ oder -1 besitzt. Dies geschieht durch vollständige Induktion nach der Zeilen- (oder Spalten) Zahl p der quadratischen Untermatrix C. Ist $p = 1$, so ist trivialerweise $\det(C) = 0$ oder $\det(C) = \pm 1$. Sei nun $C \in \mathbb{Z}^{p\times p}$ eine Untermatrix von A. Hat C eine Nullspalte, so ist C singulär. Besitzt C eine Spalte mit genau einem von Null verschiedenen (also $+1$ oder -1) Eintrag, so kann man zur Berechnung von $\det(C)$ nach dieser Spalte entwickeln und erhält das gewünschte Ergebnis aus der Induktionsvoraussetzung. Es bleibt der Fall, dass C in jeder Spalte (genau) zwei von 0 verschiedene Einträge besitzt. Die beiden Bedingungen 1. und 2. sichern, dass für alle für C relevanten Spaltenindizes j gilt

$$\sum_{i\in I_1} a_{ij} = \sum_{i\in I_2} a_{ij},$$

eine Linearkombination der Zeilen von C verschwindet also und folglich ist $\det(C) = 0$. Der Satz ist bewiesen. □

Als Folgerung aus Satz 2.8 zeigen wir:

Satz 2.9 *Sei $G = (V, E)$ ein (ungerichteter) bipartiter Graph und $A \in \mathbb{R}^{|V|\times|E|}$ seine Inzidenzmatrix. Dann ist A total unimodular.*

Beweis: Nach Voraussetzung lässt sich die Eckenmenge V als $V = U \cup W$ partitionieren mit nichtleeren Mengen U und W und der Eigenschaft, dass es zu jedem $e \in E$ Ecken $u \in U$, $w \in W$ mit $e = uw$ gibt. Man unterteile die Zeilenindexmenge $I := \{1, \ldots, |V|\}$ in die Menge I_1 zu Knoten aus U und die Menge I_2 zu Knoten aus W. Da jede Kante Endknoten aus verschiedenen Klassen hat, ist die erste Bedingung in Satz 2.8 erfüllt, während die zweite irrelevant ist. □

Bemerkung: Die Inzidenzmatrix A eines bipartiten Graphen $G = (U \cup W, E)$ hat einen Rang $\leq |U| + |W| - 1$. Denn summiert man die zur Knotenmenge U gehörenden Zeilen so erhält man einen Vektor aus lauter Einsen, genau wie bei der entsprechenden Summation über die zur Knotenmenge W gehörenden Zeilen. Daher sind die $|U| + |W|$ Zeilen der Inzidenzmatrix linear abhängig und daher Rang $(A) \leq |U| + |W| - 1$

(natürlich auch $\leq |E|$). Ist ein (nicht notwendig bipartiter) Graph $G = (V, E)$ aber zusammenhängend, so sind die zu einem G auspannenden Baum gehörenden $|V| - 1$ Spalten der Inzidenzmatrix linear unabhängig. Dies ist praktisch in Aufgabe 2 in Abschnitt 4.1 gezeigt worden, der Vollständigkeit halber wiederholen wir das Argument.

- Sei etwa $T = (V, E')$ mit $|E'| = |V| - 1$ ein G aufspannender Baum. In diesem gibt es mindestens eine[10] Ecke mit einem Grad 1, also eine Ecke, die mit nur einer Kante inzidiert. In der zu $T = (V, E')$ gehörenden Inzidenzmatrix bedeutet dies, dass es eine Zeile mit genau einer 1, sonst nur Nullen, gibt. Bei einer zu Null sich addierenden Linearkombination der Spalten muss also der Koeffizient verschwinden, der zu der Spalte gehört, in der die Eins steht. Streichen der Ecke vom Grad 1 und der einzigen inzidierenden Kante liefert wieder einen Baum, auf den das selbe Argument sinngemäß angewandt werden kann.

Insbesondere ist in einem zusammenhängenden bipartiten Graphen $(U \cup W, E)$ der Rang der zugehörigen Inzidenzmatrix genau $|U| + |W| - 1$. Die Inzidenzmatrix ist total unimodular, nach Weglassen einer beliebigen Zeile ist die entstehende gestutzte Inzidenzmatrix ebenfalls total unimodular und (da vollständiger Rang) unimodular. Insbesondere gelten diese Aussagen für den vollständigen bipartiten Graphen $K_{m,n}$. Bei ganzzahligen Lagerbeständen und ganzzahligen Kundenwünschen besitzt das (unkapazitierte) Transportproblem daher eine ganzzahlige Lösung (wobei wir ausnutzen, dass ein lineares Programm in Standardform, wenn es lösbar ist, auch eine Ecke als Lösung hat). □

Beispiel: In Abbildung 4.11 geben wir einen speziellen Graphen an. Die zugehöri-

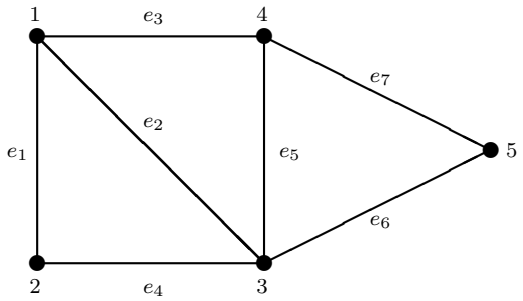


Abbildung 4.11: Ein Graph mit 5 Ecken und 7 Kanten

ge Ecken-Kanten-Inzidenzmatrix (bei der angegebenen Nummerierung der Ecken und

[10]Man kann sogar zeigen, dass es mindestens *zwei* Ecken vom Grad 1 im Baum $T = (V, E')$ gibt. Denn andernfalls wäre wegen des Handshaking-Lemmas

$$1 + 2(|V| - 1) \leq \sum_{x \in V} d(x) = 2\,|E'| = 2(|V| - 1),$$

ein Widerspruch.

Kanten) ist gegeben durch

$$A = \left(\begin{array}{ccccccc} 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 & 0 \\ \hline 0 & 1 & 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 \end{array}\right).$$

Die Matrix A ist *nicht* total unimodular, da z. B. der rechte untere Block die Determinante -2 besitzt. Es ist Rang $(A) = 5$, also maximal, aber A auch nicht unimodular, da z. B. mit $B = \{3, 4, 5, 6, 7\}$ die Inverse von A_B durch

$$A_B^{-1} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ -\frac{1}{2} & -\frac{1}{2} & \frac{1}{2} & \frac{1}{2} & -\frac{1}{2} \\ \frac{1}{2} & -\frac{1}{2} & \frac{1}{2} & -\frac{1}{2} & \frac{1}{2} \\ -\frac{1}{2} & \frac{1}{2} & -\frac{1}{2} & \frac{1}{2} & \frac{1}{2} \end{pmatrix}$$

gegeben ist. Dies ist natürlich kein Widerspruch zu Satz 2.9, da der obige Graph nicht bipartit ist. □

Eine weitere Folgerung aus Satz 2.8 ist

Satz 2.10 *Die Knoten-Pfeil-Inzidenzmatrix zu einem gerichteten Graphen $(\mathcal{N}, \mathcal{A})$ ist total unimodular.*

Beweis: In Satz 2.8 setze man $I_1 := \{1, \ldots, |\mathcal{N}|\}$ und $I_2 := \emptyset$. Da in jeder Spalte von A genau eine 1 und eine -1 steht und sonst nur Nullen, liefert eine Anwendung von Satz 2.8 die Behauptung. □

Bemerkung: Sinngemäß gilt die selbe Bemerkung wie die im Anschluss zu Satz 2.9. Die Knoten-Pfeil-Inzidenzmatrix A eines zusammenhängenden gerichteten Graphen $(\mathcal{N}, \mathcal{A})$ hat den Rang $|\mathcal{N}| - 1$. Gewinnt man die gestutzte Knoten-Pfeil-Inzidenzmatrix $\tilde{A}$ aus A durch Streichen einer beliebigen Zeile, so ist $\tilde{A}$ total unimodular, hat vollen Rang $|\mathcal{N}| - 1$ und ist daher auch unimodular. □

Das Transportproblem mit ganzzahligem Angebot und Bedarf besitzt eine ganzzahlige Lösung. Wie steht es aber mit dem kapazitierten Transportproblem oder dem Minimale-Kosten-Fluss-Problem mit ganzzahligen Kapazitätsschranken auf den Pfeilen? Diese Frage wird in dem folgenden Satz beantwortet.

Satz 2.11 *Man betrachte die lineare Optimierungsaufgabe*

(P) $\qquad$ *Minimiere* $\quad c^T x \quad$ *auf* $\quad M := \{x \in \mathbb{R}^n : 0 \leq x \leq u,\ Ax = b\},$

wobei u in gewissen Komponenten den Wert $+\infty$ haben kann, ferner sei Rang $(A) = m$. *Dann gilt:*

1. *Besitzt (P) eine Lösung, dann gibt es auch eine optimale zulässige Basislösung $x^* \in M$. Zu dieser gibt es also eine m-elementige Menge $B^* \subset \{1, \ldots, n\}$ mit der Eigenschaft, dass A_B^* nichtsingulär ist, und $x_j^* \in \{0, u_j\}$ für alle $j \in N^* := \{1, \ldots, n\} \setminus B^*$.*

2. *Ist $A \in \mathbb{Z}^{m\times n}$ unimodular und $b \in \mathbb{Z}^m$ sowie $u \in \mathbb{Z}^n$ (eventuell mit gewissen Komponenten gleich $+\infty$), so ist jede zulässige Basislösung von (P) ganzzahlig.*

Beweis: [11] Zu einer Lösung x^* von (P) definiere man die Indexmenge

$$J(x^*) := \{j \in \{1,\dots,n\} : 0 < x_j^* < u_j\}$$

und wähle unter allen Lösungen x^* von (P) eine mit minimalem $|J(x^*)|$. Ist $J(x^*) = \emptyset$, so ist x^* eine Basislösung mit beliebigem B^*, für welches A_{B^*} nichtsingulär ist. Daher können wir im weiteren annehmen, dass $|J(x^*)| \geq 1$. Wir wollen uns überlegen, dass $A_{J(x^*)}$ den Rang $|J(x^*)|$ hat bzw. die Spalten von A, die zu Indizes aus $J(x^*)$ gehören, linear unabhängig sind. Ist dies nicht der Fall, so existiert ein Vektor $a_{J(x^*)} \neq 0$ mit $A_{J(x^*)}a_{J(x^*)} = 0$. Man definiere $x^+(\delta)$ und $x^-(\delta)$ in Abhängigkeit von $\delta > 0$ durch

$$x_j^+(\delta) := \begin{cases} x_j^* + \delta a_j, & \text{falls} \quad j \in J(x^*), \\ x_j^*, & \text{falls} \quad j \notin J(x^*), \end{cases}$$

und

$$x_j^-(\delta) := \begin{cases} x_j^* - \delta a_j, & \text{falls} \quad j \in J(x^*), \\ x_j^*, & \text{falls} \quad j \notin J(x^*). \end{cases}$$

Dann kann man $\delta > 0$ so klein wählen, dass $x^+(\delta)$ und $x^-(\delta)$ in M liegen, also zulässig für (P) sind, andererseits aber auch so größ, dass $\min(|J(x^+(\delta))|, |J(x^-(\delta))|) < |J(x^*)|$. Sei etwa $|J(x^+(\delta))| < |J(x^*)|$. Dann ist $x^* = \frac{1}{2}(x^+(\delta) + x^-(\delta))$. Da x^* eine Lösung von (P) ist, folgt daher

$$c^T x^* = \frac{1}{2}c^T x^+(\delta) + \frac{1}{2}c^T x^-(\delta) \geq \frac{1}{2}c^T x^+(\delta) + \frac{1}{2}c^T x^*,$$

folglich $c^T x^* \geq c^T x^+(\delta)$. Daher ist $x^+(\delta)$ eine Lösung von (P) mit einer kleineren Anzahl von Komponenten in $(0, u_j)$, ein Widerspruch. Ist $|J(x^*)| = m$, so setze man $B^* := J(x^*)$. Andernfalls ergänze man $J(x^*)$ durch weitere linear unabhängige Spalten zu einer m-elementigen Menge B^*. Insgesamt ist der erste Teil des Satzes bewiesen.

Der zweite Teil der Aussage ist eine einfache Folge der Unimodularität von A und der vorausgesetzten Ganzzahligkeit von b und u. Denn ist (x^*, B^*) eine Basislösung von (P) und $N^* := \{1,\dots,n\} \setminus B^*$ die Menge der Nichtbasisindizes, so ist $x_{N^*}^*$ als ein Vektor, dessen Komponenten gleich 0 oder einem u_j sind, ganzzahlig. Aus $b = Ax^* = A_{B^*}x_{B^*}^* + A_{N^*}x_{N^*}$ folgt $x_{B^*} = A_{B^*}^{-1}(b - A_{N^*}x_{N^*}^*) \in \mathbb{Z}^m$, da $\det(A_{B^*}) = \pm 1$ und $A_{B^*} \in \mathbb{Z}^{m\times m}$ (siehe Lemma 2.1). Damit ist der Satz schließlich bewiesen. □

Bemerkung: Stutzt man also die Ecken-Kanten-Inzidenzmatrix eines zusammenhängenden bipartiten Graphen oder die Knoten-Pfeil-Inzidenzmatrix eines zusammenhängenden gerichteten Graphen, so erhält man eine unimodulare Matrix mit vollem Rang (sogar jeweils eine total unimodulare Matrix) und jede zulässige Basislösung des entsprechenden kapazitierten Problems ist ganzzahlig, ferner existiert eine Lösung, die eine optimale Basislösung, also ganzzahlig ist. Für das Minimale-Kosten-Fluss-Problem und insbesondere das Transportproblem mit ganzzahligen Kapazitätsschranken gilt also: Jede

[11] Der Beweis ist eine Adaption des Beweises der entsprechenden Aussage für lineare Programme in Standardform.

zulässige Basislösung ist ganzzahlig, insbesondere existiert eine ganzzahlige Lösung, wenn es überhaupt eine Lösung gibt. Da sich jedes Maximaler-Fluss-Problem als ein Minimale-Kosten-Fluss-Problem schreiben lässt, ist die entsprechende Aussage auch für diese Klasse von Aufgaben richtig. □

Bemerkung: Der letzte Satz 2.11 hätte natürlich ohne Schwierigkeiten auch für eine Nebenbedingung $l \leq x \leq u$ statt $0 \leq x \leq u$ formuliert werden können. Hier könnten dann gewisse der Komponenten von l auch gleich $-\infty$ sein. □

4.2.1 Aufgaben

1. Sei $A \in \mathbb{R}^{m\times n}$, $b \in \mathbb{R}^m$. Hiermit definiere man

   $$M' := \{x \in \mathbb{R}^n : x \geq 0,\ Ax \leq b\}, \qquad M := \{z \in \mathbb{R}^{n+m} : z \geq 0,\ (\ A \quad I\ )z = b\}.$$

   Man zeige, dass $x \in M$ genau dann eine Ecke von M' ist, wenn

   $$z := \begin{pmatrix} x \\ b - Ax \end{pmatrix}$$

   eine Ecke von M ist.

2. Sei

   $$A = \begin{pmatrix} a_1^T \\ \vdots \\ a_m^T \end{pmatrix} \in \mathbb{R}^{m\times n}, \qquad b = \begin{pmatrix} b_1 \\ \vdots \\ b_m \end{pmatrix} \in \mathbb{R}^m$$

   und

   $$M := \{x \in \mathbb{R}^n : Ax \leq b\}.$$

   Man zeige, dass $x \in M$ genau dann eine Ecke von M ist, wenn mit

   $$I(x) := \{i \in \{1, \ldots, m\} : a_i^T x = b_i\}$$

   die Implikation

   $$a_i^T p = 0 \quad (i \in I(x)) \Longrightarrow p = 0$$

   gilt.

3. Sei $a := (1, 2, 4)^T$ und $b := 4$. Man zeige, dass jede Ecke von

   $$M := \{x \in \mathbb{R}^3 : x \geq 0,\ a^T x \leq b\}$$

   ganzzahlig ist.

4. Mit $A \in \mathbb{Z}^{m\times n}$ sind auch A^T, $-A$ und $(\ A \quad I\ )$ total unimodular.

4.3 Das Kürzester-Pfad-Problem

Gegeben sei ein (schwach) zusammenhängender gerichteter Graph $(\mathcal{N}, \mathcal{A})$. Jeder Pfeil $(i, j) \in \mathcal{A}$ trägt eine Länge bzw. Kosten $c_{ij} \in \mathbb{R}$ (wenn nichts anderes gesagt wird, können diese auch negativ und nichtganzzahlig sein). Ist $P = \{(i_1, i_2), \ldots, (i_{k-1}, i_k)\}$ ein (Vorwärts-) Pfad von i_1 nach i_k, so ist seine *Länge* durch $c(P) = \sum_{j=1}^{k-1} c_{i_j i_{j+1}}$ gegeben. Gesucht ist ein kürzester Pfad von einem Anfangsknoten s zu einem Endknoten t (oder kürzeste Pfade vom Anfangsknoten s zu allen übrigen Knoten). Wir folgen im wesentlichen der Darstellung bei D. P. Bertsekas (1998, S. 51 ff.).

4.3.1 Ein Modellalgorithmus

Es wird ein Modellalgorithmus dafür entwickelt, einen kürzesten Pfad vom Knoten 1 zu den übrigen $|\mathcal{N}| - 1$ Knoten zu bestimmen. In jedem Schritt wird ein Vektor $(d_1, \ldots, d_{|\mathcal{N}|})$ aktualisiert, wobei $d_j \in \mathbb{R} \cup \{\infty\}$ die *Marke* des Knotens j genannt wird. Am Schluss ist d_j die Länge eines kürzesten Pfades vom Knoten 1 zum Knoten j. Ist daher am Schluss $d_j = \infty$, so bedeutet dies, dass es keinen (gerichteten) Pfad von 1 nach j gibt. Im folgenden Lemma wird eine hinreichende Optimalitätsbedingung für einen kürzesten Pfad angegeben.

Lemma 3.1 *Sei $d = (d_1, \ldots, d_{|\mathcal{N}|})^T$ ein Vektor mir $d_j \leq d_i + c_{ij}$, $(i,j) \in \mathcal{A}$, und P ein Pfad von s nach t mit $d_j = d_i + c_{ij}$ für alle Pfeile (i,j) aus P. Dann ist P ein kürzester Pfad von s nach t.*

Beweis: Sei $P = \{(i_1, i_2), (i_2, i_3), \ldots, (i_{k-1}, i_k)\}$ mit $s = i_1$ und $t = i_k$ ein s und t verbindender Pfad. Die Länge des Pfades P ist dann

$$c(P) = \sum_{j=1}^{k-1} c_{i_j i_{j+1}} = \sum_{j=1}^{k-1} (d_{i_{j+1}} - d_{i_j}) = d_{i_k} - d_{i_1} = d_t - d_s$$

und diese kann wegen $d_j \leq d_i + c_{ij}$ durch keinen anderen Pfad von s nach t unterboten werden. □

Bemerkung: Man vermutet zu Recht, dass so etwas wie der schwache Dualitätssatz hinter der Aussage des letzten Lemmas steckt. Das wollen wir genauer erläutern. Der Aufgabe, einen kürzesten Pfad von s nach t zu bestimmen, hatten wir früher in Unterabschnitt 4.1.2 schon die lineare Optimierungsaufgabe

$$\text{(P)} \qquad \text{Minimiere} \quad c^T x \quad \text{auf} \quad M := \{x \in \mathbb{R}^{|\mathcal{A}|} : x \geq 0,\ Ax = b\}$$

zugeordnet, wobei A die Knoten-Pfeil-Inzidenzmatrix von $(\mathcal{N}, \mathcal{A})$ bedeutet und

$$b_k := \begin{cases} 1, & \text{falls} \quad k = s, \\ -1, & \text{falls} \quad k = t, \\ 0, & \text{sonst}, \end{cases} \qquad k \in \mathcal{N}.$$

Das zu (P) duale lineare Programm ist

$$\text{(D)} \qquad \text{Maximiere} \quad b^T u \quad \text{auf} \quad N := \{u \in \mathbb{R}^{|\mathcal{N}|} : A^T u \leq c\},$$

bzw.

$$\text{(D)} \qquad \begin{cases} \text{Maximiere} & u_s - u_t \quad \text{unter den Nebenbedingungen} \\ & u_i - u_j \leq c_{ij}, \quad (i,j) \in \mathcal{A}. \end{cases}$$

Man sieht, dass das Negative des obigen Markenvektors $u := -d$ dual zulässig ist und $b^T u = c^T x$, wobei Komponenten von $x \in \mathbb{R}^{|\mathcal{A}|}$ gleich 1 oder 0 sind, je nach dem ob ein Pfeil im Pfad P vorkommt oder nicht. Die obige Aussage hätte also auch mit dem schwachen Dualitätssatz bewiesen werden können. □

Jetzt können wir schon den Modellalgorithmus angeben. Wir nehmen hierbei o. B. d. A. an, es sei $s = 1$.

- Setze $V := \{1\}$, $d_1 := 0$ und $d_i := \infty$, $i = 2, \ldots, |\mathcal{N}|$.
- Solange $V \neq \emptyset$:
  - Wähle $i \in V$ und setze $V := V \setminus \{i\}$.
  - Für alle $(i, j) \in \mathcal{A}$:
    - Falls $d_j > d_i + c_{ij}$, dann: Setze $d_j := d_i + c_{ij}$ und[12] $V := V \cup \{j\}$.

Wir sprechen von einem *Modellalgorithmus*, weil nicht spezifiziert ist, welcher der Knoten aus V entfernt wird. Wir sagen, der Knoten i, der V verlässt, werde *gescant*.

Beispiel: In Abbildung 4.12 geben wir einen gewichteten, gerichteten Graphen an. Wir erhalten die folgenden Iterationen, wobei k die Iterationsstufe, V die aktuelle

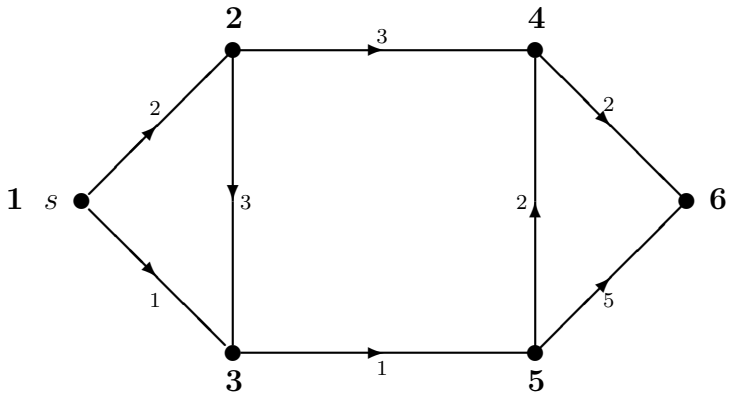


Abbildung 4.12: Ein gerichteter Graph mit Quelle $s = 1$

Kandidatenliste, d den Markenvektor und i den zu entfernenden Knoten bedeutet.

k	i	d	V
0		$(0, \infty, \infty, \infty, \infty, \infty)$	$\{1\}$
1	1	$(0, 2, 1, \infty, \infty, \infty)$	$\{2, 3\}$
2	2	$(0, 2, 1, 5, \infty, \infty)$	$\{3, 4\}$
3	3	$(0, 2, 1, 5, 2, \infty)$	$\{4, 5\}$
4	4	$(0, 2, 1, 5, 2, 7)$	$\{5, 6\}$
5	5	$(0, 2, 1, 4, 2, 7)$	$\{4, 6\}$
6	4	$(0, 2, 1, 4, 2, 6)$	$\{6\}$
7	6	$(0, 2, 1, 4, 2, 6)$	$\emptyset$

□

Im folgenden Satz werden die wesentlichen Eigenschaften des obigen Modellalgorithmus angegeben.

Satz 3.2 *Man betrachte den obigen Modellalgorithmus. Dann gilt:*

[12] Der Knoten j ist natürlich nur dann in V aufzunehmen, wenn er nicht vorher schon in V enthalten war.

1. *Am Ende jeder Iteration gelten die folgenden Bedingungen:*

   (a) *Ist $d_j < \infty$, so ist d_j die Länge eines (gewissen) Pfades vom Knoten 1 zum Knoten j.*

   (b) *Ist $i \notin V$, so ist $d_i = \infty$ oder $d_j \leq d_i + c_{ij}$ für alle j mit $(i,j) \in \mathcal{A}$.*

2. *Terminiert der Algorithmus, so ist d_j für alle j mit $d_j < \infty$ die minimale Länge von Knoten 1 zu Knoten j und*

$$d_j = \begin{cases} \min_{(i,j)\in\mathcal{A}}(d_i + c_{ij}), & j \neq 1, \\ 0, & j = 1. \end{cases}$$

   Ferner ist $d_j = \infty$ genau dann, wenn es keinen (Vorwärts-) Pfad von 1 nach j gibt.

3. *Terminiert der Algorithmus nicht, so gibt es einen Knoten j und eine Folge von Pfaden von 1 nach j mit einer Länge, die gegen $-\infty$ divergiert.*

4. *Der Algorithmus terminiert genau dann, wenn es keinen bei 1 startenden Pfad gibt, der einen Zyklus (geschlossener Pfad, dessen Start- und Endknoten also übereinstimmen) negativer Länge enthält.*

Beweis: Den ersten Teil beweisen wir durch Induktion nach der Induktionsstufe. Nach der ersten Iteration (in dieser musste $i = 1$ gewählt werden) gibt es für alle Knoten $j \neq 1$ mit $d_j < \infty$ einen Pfeil $(1,j) \in \mathcal{A}$ mit der Marke $d_j = c_{1j}$, während $d_1 = 0$, was als Abstand des Knotens 1 zu sich selbst interpretiert wird. Angenommen dies gelte auch zu Beginn einer Iteration, bei der der Knoten i aus V entfernt wird. Dann ist $d_i < \infty$ (denn i muss einmal in V aufgenommen worden sein) und nach Induktionsvoraussetzung ist d_i Länge eines gewissen Pfades P_i von 1 nach i. Wenn eine Marke d_j in der Iteration verändert wird, so wird $d_j := d_i + c_{ij}$ gesetzt, was die Länge von 1 nach j auf einem Pfad ist, der zunächst aus dem Pfad P_i und dann aus dem Pfeil (i,j) besteht. Damit ist Teil (a) des ersten Teiles bewiesen. Nun zum Teil (b). Wenn in einer Iteration i aus V entfernt wird, so ist $d_j \leq d_i + c_{ij}$ für alle j mit $(i,j) \in \mathcal{A}$. Bis zur nächsten Aufnahme von i in V bleibt d_i konstant, während die Marken d_j für alle j mit $(i,j) \in \mathcal{A}$ nicht ansteigen können, so dass die Ungleichung $d_j \leq d_i + c_{ij}$ erhalten bleibt.

Zum Beweis des zweiten Teils führen wir die Indexmengen

$$I := \{i : d_i < \infty \text{ am Schluss}\}, \qquad \overline{I} := \{i : d_i = \infty \text{ am Schluss}\}$$

ein und zeigen, dass $j \in \overline{I}$ genau dann, wenn es keinen (Vorwärts-) Pfad von 1 nach j gibt. Ist $i \in I$, so ist $d_i < \infty$ und daher $d_j < \infty$ für alle j, für die $(i,j) \in \mathcal{A}$, diese j liegen also in I. Daher gibt es keinen Pfad von einem Knoten aus I (z. B. dem Knoten 1) zu einem Knoten aus $\overline{I}$. Gibt es umgekehrt keinen Pfad von 1 nach j, so ist $d_j = \infty$, weil man andernfalls einen Widerspruch zum ersten Teil des Satzes hätte. Nun zeigen wir: Bricht der Algorithmus ab, so ist d_j für alle $j \in I$ der kürzeste Abstand von Knoten 1 zum Knoten j. Aus dem ersten Teil des Satzes erhalten wir: Für alle $i \in I$ ist $d_j \leq d_i + c_{ij}$ für alle j mit $(i,j) \in \mathcal{A}$, weiterhin d_i die Länge eines gewissen Pfades

P_i von 1 nach i. Nun sei $m \in I$ beliebig und P ein beliebiger Pfad von 1 nach m. Für jeden Pfeil (i,j) dieses Pfades gilt $d_j \leq d_i + c_{ij}$, daher gilt für die Länge des Pfades $c(P) \geq d_m - d_1 \geq d_m$, wobei wir $d_1 \leq 0$ benutzt haben (weil am Anfang $d_1 = 0$ und keine Marke sich vergrößert). Die Länge des beliebigen Pfades ist also nicht kleiner als die des Pfades P_m, dieser ist also ein kürzester Pfad von 1 nach m und d_m seine Länge. Ferner ist $d_j = d_i + c_{ij}$ für alle Pfeile (i,j) auf dem kürzesten Pfad P_m, $m \in I$, woraus

$$d_j = \begin{cases} \min_{(i,j)\in\mathcal{A}}(d_i + c_{ij}), & j \neq 1, \\ 0, & j = 1 \end{cases}$$

folgt. Damit ist der zweite Teil des Satzes bewiesen.

Bricht der Algorithmus nicht ab, so muss eine Marke d_j unendlich oft verkleinert werden. Dazu gehört eine Folge von Pfaden P_j von 1 nach j mit sich verkleinernder Länge d_j. Den Pfad P_j denke man sich jeweils dargestellt als einen einfachen Pfad von 1 nach j und Zyklen. Da die Anzahl der einfachen Pfade von 1 nach j endlich ist und die Länge der Pfade P_j monoton fällt, muss der Pfad P_j letztendlich einen Zyklus negativer Länge enthalten. Indem man diesen Zyklus negativer Länge hinreichend oft durchläuft, erhält man Pfade von 1 nach j beliebig kleiner Länge.

Der Algorithmus terminiert genau dann, wenn die Länge aller Pfade, die bei 1 starten, nach unten beschränkt ist. Dies wiederum ist genau dann der Fall, wenn es keinen bei 1 startenden Pfad gibt, der einen (einfachen) Zyklus negativer Länge enthält. Der Satz ist damit bewiesen. □

Bemerkung: Wenn alle Zyklen im gerichteten Graphen $(\mathcal{N},\mathcal{A})$ nichtnegative Länge haben (das ist natürlich insbesondere dann der Fall, wenn $c_{ij} \geq 0$ für alle $(i,j) \in \mathcal{A}$) und es vom Knoten 1 zu jedem Knoten j einen Pfad gibt, so terminiert obiger Modellalgorithmus, am Schluss sind alle Marken gleich dem entsprechenden kürzesten Abstand und es ist $d_1 = 0$ und

$$d_j = \min_{(i,j)\in\mathcal{A}}(d_i + c_{ij}), \qquad j \neq 1.$$

Diese Gleichung heißt *Bellmansche Gleichung*. Ihre intuitive Bedeutung ist die folgende. Die kürzeste Entfernung von 1 nach j erhält man, indem man den Vorganger i des Knotens j so wählt, dass die Summe aus Abstand d_i (von 1 nach i) und der Länge des Pfeiles (i,j) minimal ist.

Wenn alle den Knoten 1 nicht enthaltenden Zyklen in $(\mathcal{N},\mathcal{A})$ positive Länge haben, so kann man wegen der Bellmanschen Gleichung für jeden Knoten $j \neq 1$ nicht nur den kürzesten Abstand sondern auch den kürzesten Pfad bestimmen. Hierzu bestimme man zu jedem Knoten $j \neq 1$ einen Pfeil (i,j) mit $d_j = d_i + c_{ij}$ und betrachte den Untergraphen mit diesen $|\mathcal{N}| - 1$ Pfeilen. In Abbildung 4.13 haben wir dies für den gewichteten Graphen aus Abbildung 4.12 zum Markenvektor $d = (0,2,1,4,2,6)$ durchgeführt. Da der Untergraph zusammenhängend ist und $|\mathcal{N}| - 1$ Pfeile bzw. Kanten besitzt, ist er ein aufspannender Baum. Von jedem Knoten j kommt man, indem man die Pfeile rückwärts verfolgt, sozusagen rückwärts auf dem kürzesten Pfad zum Knoten 1. Man kann hierbei nicht zu einem schon vorher erreichten Knoten kommen, weil dies die Existenz eines Kreises der Länge 0 bedeuten würde. □

Bemerkung: Der Modellalgorithmus muss nicht mit

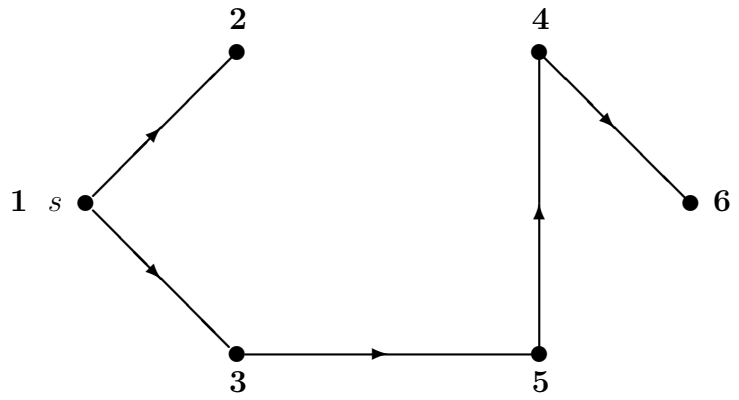


Abbildung 4.13: "Shortest Path Spanning Tree" mit Quelle $s = 1$

- Setze $V := \{1\}$, $d_1 := 0$ und $d_i := \infty$, $i = 2, \ldots, |\mathcal{N}|$

gestartet werden. Jede Liste V mit einem Markenvektor d kann genommen werden, für welche die Bedingungen des ersten Teiles von Satz 3.2 gelten, also:

- Ist $d_j < \infty$, so ist d_j die Länge eines (gewissen) Pfades vom Knoten 1 zum Knoten j.
- Ist $i \notin V$, so ist $d_i = \infty$ oder $d_j \leq d_i + c_{ij}$ für alle j mit $(i, j) \in \mathcal{A}$.

Insbesondere kann man den Markenvektor mit $d_1 = 0$ so setzen, dass

- Für jeden Knoten $i \neq 1$ ist $d_i = \infty$ oder d_i die Länge eines Pfades von 1 bis i.

Anschließend kann man V so bestimmen, dass

- Alle Knoten i mit $d_i + c_{ij} < d_j$ für einen Pfeil (i, j) sind in V enthalten.

Diese Initialisierungstechniken können nützlich sein, wenn eine große Zahl ähnlicher Probleme zu lösen ist, bei denen sich z. B. nur einige der Pfeillängen ändern. □

Wir werden in den beiden kurzen folgenden Unterabschnitten zwei spezielle Verfahren, die sich dem Modellalgorithmus unterordnen lassen, kennen lernen.

4.3.2 Label-Setting-Verfahren (Dijkstra)

Genau wie bei K. Neumann, M. Morlock (1993, S. 212) verzichten wir auf eine Übersetzung von "Label-Setting". Spezifiziert werden muss nur die Wahl des Knoten i, der die Liste V verlässt. Beim Label-Setting-Verfahren von Dijkstra ist der Knoten aus V mit einer minimalen Marke zu entfernen. Insgesamt hat man also das folgende Verfahren:

- Setze $V := \{1\}$, $d_1 := 0$ und $d_i := \infty$, $i = 2, \ldots, |\mathcal{N}|$.
- Solange $V \neq \emptyset$:

- Bestimme $i \in V$ mit $d_i = \min_{j \in V} d_j$ und setze $V := V \setminus \{i\}$.
- Für alle $(i, j) \in \mathcal{A}$:
  - Falls $d_j > d_i + c_{ij}$, dann: Setze $d_j := d_i + c_{ij}$ und[13] $V := V \cup \{j\}$.

Beispiel: Wir betrachten erneut den gerichteten, bewerteten Graphen aus Abbildung 4.12. Wir erhalten diesmal die folgende Tabelle:

k	i	d	V
0		$(0, \infty, \infty, \infty, \infty, \infty)$	$\{1\}$
1	1	$(0, 2, 1, \infty, \infty, \infty)$	$\{2, 3\}$
2	3	$(0, 2, 1, \infty, 2, \infty)$	$\{2, 5\}$
3	5	$(0, 2, 1, 4, 2, 7)$	$\{2, 4, 6\}$
4	2	$(0, 2, 1, 4, 2, 7)$	$\{4, 6\}$
5	4	$(0, 2, 1, 4, 2, 6)$	$\{6\}$
6	6	$(0, 2, 1, 4, 2, 6)$	$\emptyset$

□

Im folgenden Satz wird u. a. bewiesen, dass bei nichtnegativen Pfeillängen c_{ij} im obigen Algorithmus ein Knoten höchstens einmal der Liste hinzugefügt und daher auch nur einmal gescant werden kann. Die Marke ist endgültig, sobald der Knoten die Liste verlässt.

Satz 3.3 *Seien alle Pfeillängen im gewichteten, gerichteten Graphen $(\mathcal{N}, \mathcal{A})$ nichtnegativ, also $c_{ij} \geq 0$ für alle $(i, j) \in \mathcal{A}$. Dann gilt für das obige Label-Setting-Verfahren von Dijkstra:*

1. *In jeder Iterationsstufe gelten mit $W := \{i : d_i < \infty, i \notin V\}$ die folgenden Aussagen:*
   (a) *Kein Knoten, der zu W beim Start der Iteration gehört, wird zur Liste V im Verlauf der Iteration hinzugefügt.*
   (b) *Zum Schluss der Iteration ist $d_i \leq d_j$ für alle $i \in W$ und $j \notin W$.*
   (c) *Für jeden Knoten j betrachte man einfache Pfade, die beim Knoten 1 starten, im Knoten j enden und alle anderen Knoten aus W zum Ende der Iteration haben. Dann ist die Marke d_j zum Ende der Iteration gleich der Länge des kürzesten dieser Pfade ($d_j = \infty$ wenn kein solcher Pfad existiert).*
2. *Das Label-Setting-Verfahren terminiert. Am Schluss sind alle Knoten mit einer endlichen Marke genau einmal aus der Liste V entfernt worden, und zwar in der Reihenfolge wachsenden kürzesten Abstands zum Knoten 1. Falls also die letzten Marken der Knoten i und j endlich sind und $d_i < d_j$ genügen, so wird i vor j aus V entfernt.*

[13] Der Knoten j ist natürlich nur dann in V aufzunehmen, wenn er nicht vorher schon in V enthalten war.

Beweis: Im ersten Teil des Satzes werden (a) und (b) zusammen durch Induktion nach der Iterationsstufe bewiesen. Offensichtlich gelten (a) und (b) in der ersten Iteration, in der 1 die Liste V verlässt und nach W kommt. Angenommen, (a) und (b) gelten in einer Iterationsstufe $k-1$, in der Iterationsstufe k sei $d_i = \min_{j\in V} d_j$, so dass i die Liste V verlässt (also ist i zu Beginn der k-ten Iterationsstufe in V, am Schluss aber nicht). Sei W die Menge aller Knoten mit endlicher Marke, die zu Beginn (also vor dem Entfernen von i) der k-ten Iterationsstufe nicht in V sind, $\overline{W}$ die entsprechende Menge zum Schluss der k-ten Iteration (also nach dem eventuellen Hinzufügen von Knoten zu V). Entsprechend sei d der Markenvektor zu Beginn und $\overline{d}$ der Markenvektor zum Schluss der k-ten Iteration. Nach Induktionsvoraussetzung ist $d_j \le d_i$ für alle $j \in W$, weiter ist $c_{ij} \ge 0$ für alle $(i,j) \in \mathcal{A}$. Daher ist $d_j \le d_i + c_{ij}$ für alle Pfeile (i,j) mit $j \in W$. Ein Knoten $j \in W$ kann folglich der Liste V in der Iteration k nicht hinzugefügt werden. Das beendet den Induktionsbeweis für (a) und zeigt, dass $\overline{W} = W \cup \{i\}$. In der Iterationsstufe k sind die einzigen sich eventuell verändernden Marken diejenigen zu Knoten $j \notin \overline{W}$ mit $(i,j) \in \mathcal{A}$, für diese ist $\overline{d}_j = \min(d_j, d_i + c_{ij})$. Wegen $c_{ij} \ge 0$, $d_i \le d_j$ für alle $j \notin W$ (hier geht $d_i = \min_{j\in V} d_j$ ein) und $\overline{d}_i = d_i$ ist $\overline{d}_i \le \overline{d}_j$ für alle $j \notin \overline{W}$. Nach Induktionsvoraussetzung ist $d_m \le d_i$ für alle $m \in \overline{W}$, für diese verändert sich die Marke nicht, es ist also $d_m = \overline{d}_m$ für alle $m \in \overline{W}$. Für beliebige $m \in \overline{W}$, $j \notin \overline{W}$, ist daher

$$\overline{d}_m = d_m \le d_i \le \overline{d}_i \le \overline{d}_j,$$

was den Induktionsbeweis für (b) beschließt. Um (c) zu beweisen, wähle man einen Knoten j und betrachte einen Untergraphen, der aus den Knoten $W \cup \{j\}$ sowie Pfeilen, deren Anfangs- und Endknoten in $W \cup \{j\}$ liegen, besteht. Ferner betrachte man das auf diesem Untergraphen induzierte Kürzester-Pfad-Problem (gleicher Ausgangsknoten 1, gleiche Pfeillängen). Wegen der schon bewiesenen Aussagen (a) und (b) ergibt das auf das modifizierte Kürzester-Pfad-Problem angewandte Label-Setting-Verfahren dieselbe Folge der die Liste V verlassenden Knoten und dieselbe Folge von Marken wie die Anwendung des Verfahrens auf das Originalproblem bis zur aktuellen Iteration. Wegen Satz 3.2 terminiert das auf das modifizierte Problem angewandte Label-Setting-Verfahren (es gibt keine Zyklen negativer Länge, da ja sogar alle Pfeillängen nichtnegativ sind), die Marken sind die kürzesten Abstände im modifizierten Problem. Daher haben die Marken zum Ende der Iteration die behaupteten Eigenschaften.

Nun zum Beweis des zweiten Teils des Satzes. Das Verfahren terminiert, da im Digraphen keine Zyklen negativer Länge existieren. In jeder Iteration wird der die Liste V verlassende Knoten der Liste W hinzugefügt, und wegen (a) im ersten Teil des Satzes kehrt kein Knoten aus W nach V zurück. Daher wird jeder Knoten, dessen Marke am Schluss endlich ist, aus V genau einmal entfernt und nach W überführt, danach wird seine Marke nicht mehr verändert. Eigenschaft (b) im ersten Teil des Satzes zeigt, dass jeder Knoten, der zu W hinzugefügt wird, eine Marke hat, die mindestens so groß wie die Marken der schon in W enthaltenen Knoten ist. Das beweist den zweiten Teil des Satzes. □

Bemerkung: Auch ohne die Voraussetzung, dass die Pfeillängen alle nichtnegativ sind, kann das Label-Setting-Verfahren angewandt werden, wenn es keine Zyklen negativer Länge gibt. Die Aussagen des Satzes müssen dann aber natürlich nicht notwendig

gelten. □

Bemerkung: Bei D. P. BERTSEKAS (1998, S. 68 ff.) wird kurz auf eine Komplexitätsanalyse und auf Implementationsfragen eingegangen. Zunächst benötigt man i. Allg. genau $|\mathcal{N}|$ Iterationen (am Anfang ist $W = \emptyset$, in jedem Iterationsschritt kommt ein Knoten zu W hinzu) und in jedem dieser Iterationsschritte wird die Liste nach einem minimalen Knoten durchsucht. Durchsucht man die Knoten der Reihe nach, um zu überprüfen, ob sie zu V gehören und unter den dazugehörenden einen mit minimaler Marke zu bestimmen, benötigt $O(|\mathcal{N}|)$ Operationen per Iteration, insgesamt also $O(|\mathcal{N}|^2)$ Operationen. Im Verlauf des Algorithmus muss ferner jeder Pfeil (i, j) genau einmal darauf hin überprüft werden, ob er der Bedingung $d_j > d_i + c_{ij}$ genügt und, wenn die Bedingung erfüllt ist, $d_j := d_i + c_{ij}$ gesetzt werden. Dies erfordert noch einmal $O(|\mathcal{A}|)$ Operationen, was durch $O(|\mathcal{N}|^2)$ dominiert wird. □

4.3.3 Label-Correcting-Verfahren

Wir beschränken uns auf die Beschreibung des Bellman-Ford-Verfahrens (siehe z. B. D. P. BERTSEKAS (1998, S. 73 ff.), K. NEUMANN, M. MORLOCK (1993, S. 208 ff.)). Die Regeln für das Entfernen eines Knotens aus V sind einfacher, dafür kann ein Knoten mehrfach in V aufgenommen werden. Jetzt ist es wesentlich, V als eine *Liste* und nicht nur als eine Menge aufzufassen. Es wird stets das erste Element der Liste entfernt. Eventuell hinzukommende Knoten kommen ans Ende der Liste. Aufeinanderfolgende Iterationen werden zu einem *Zyklus* zusammengefasst. Der erste Zyklus besteht darin, den Knoten 1 aus der Liste V zu entfernen. In jedem folgenden Zyklus werden die Knoten, die im vorhergehenden Zyklus in V aufgenommen wurden, in der Reihenfolge ihres Eintritts aus der Liste entfernt.

Eine wesentliche Eigenschaft des Bellman-Ford-Verfahrens wird in dem folgenden Satz formuliert.

Satz 3.4 *Für einen Knoten j und $k \in \mathbb{N}$ sei d_j^k der kürzeste Abstand zwischen 1 und j, wobei nur $\leq k$ Pfeile benutzt werden, und $d_j^k := \infty$, wenn j von 1 nicht über einen Pfad mit $\leq k$ Pfeilen erreicht werden kann. Dann ist $d_j \leq d_j^k$, wobei d_j die Marke des Knotens j nach dem k-ten Zyklus des Bellman-Ford-Verfahrens ist.*

Beweis: Der Beweis erfolgt durch vollständige Induktion nach k. Nach dem ersten Zyklus ist

$$d_j = \begin{cases} 0, & \text{falls} \quad j = 1, \\ c_{1j}, & \text{falls} \quad j \neq 1,\ (1,j) \in \mathcal{A}, \\ \infty, & \text{falls} \quad j \neq 1,\ (1,j) \notin \mathcal{A}, \end{cases}$$

während

$$d_j^1 = \begin{cases} c_{1j}, & \text{falls} \quad (1,j) \in \mathcal{A}, \\ \infty, & \text{falls} \quad (1,j) \notin \mathcal{A}. \end{cases}$$

Der Induktionsanfang für $k = 1$ ist also gelegt. Sei nun d der Markenvektor und V die Kandidaten-Liste nach Abschluss des k-ten Zyklus, ferner sei $\overline{d}$ der Markenvektor nach Abschluss des $(k+1)$-ten Zyklus. Die Induktionsannahme ist, dass $d_j \leq d_j^k$ für alle Knoten j. Im Induktionsschluss besteht im Nachweis von $\overline{d}_j \leq d_j^{k+1}$ für alle Knoten j.

Sei nun j ein fester Knoten. Wegen Aussage 1 (a) in Satz 3.2 ist $d_j \le d_i + c_{ij}$ für alle $i \notin V$ mit $(i,j) \in \mathcal{A}$. Da weiter $\overline{d}_j \le d_j$ (Marken können sich nicht vergrößern), ist $\overline{d}_j \le d_i + c_{ij}$ für alle $i \notin V$ mit $(i,j) \in \mathcal{A}$. Es ist aber auch $\overline{d}_j \le d_i + c_{ij}$ für alle $i \in V$ mit $(i,j) \in \mathcal{A}$. Denn sei i ein solcher Knoten. In der ersten zukünftigen Iteration, in der i aus V entfernt wird, sei $\tilde{d}_i$ die Marke von i. Diese Iteration gehört zum $(k+1)$-ten Zyklus. Dann ist $\tilde{d}_i \le d_i$ (Marken vergrößern sich nicht), ferner wird in dieser Iteration die Marke von j auf $\tilde{d}_i + c_{ij}$ gesetzt, wenn sie $\tilde{d}_i + c_{ij}$ übersteigt. Daher ist

$$\overline{d}_j \le \tilde{d}_i + c_{ij} \le d_i + c_{ij}.$$

Insgesamt ist also $\overline{d}_j \le d_i + c_{ij}$ für alle i mit $(i,j) \in \mathcal{A}$. Daher ist

$$\overline{d}_j \le \min_{(i,j)\in\mathcal{A}} (d_i + c_{ij}) \le \min_{(i,j)\in\mathcal{A}} (d_i^k + c_{ij}),$$

wobei bei der zweiten Ungleichung die Induktionsannahme eingeht. Folglich ist

$$\overline{d}_j \le \min\Big(d_j^k, \min_{(i,j)\in\mathcal{A}} (d_i^k + c_{ij})\Big), \qquad j = 1, \ldots, |\mathcal{N}|.$$

Der Induktionsschluss ist also abgeschlossen, wenn wir noch

$$(*) \qquad d_j^{k+1} = \min\Big(d_j^k, \min_{(i,j)\in\mathcal{A}} (d_i^k + c_{ij})\Big), \qquad j = 1, \ldots, |\mathcal{N}|,$$

zeigen können. Dies ist aber klar. Kann der Knoten j von 1 aus über einen Pfad mit $\le k$ Pfeilen erreicht werden, so ist $d_j^{k+1} = d_j^k$. Benötigt man dagegen einen Pfad mit $k+1$ Pfeilen, so existiert ein Knoten i mit $(i,j) \in \mathcal{A}$ und der Eigenschaft, dass man von 1 nach i auf einem Pfad mit $\le k$ Pfeilen kommen kann. Die minimale Länge eines solchen Pfades von 1 nach j ist $\min_{(i,j)\in\mathcal{A}}(d_i^k + c_{ij})$. Insgesamt gilt also $(*)$ und der Satz ist bewiesen. □

Beispiel: Der Modellalgorithmus terminiert wegen Satz 3.2 genau dann, wenn im bewerteten, gerichteten $(\mathcal{N}, \mathcal{A})$ ein Zyklus (im Sinne von gerichteten Kreis) negativer Länge existiert. In Abbildung 4.14 geben wir ein Beispiel hierfür an. Hier gibt es Pfade,

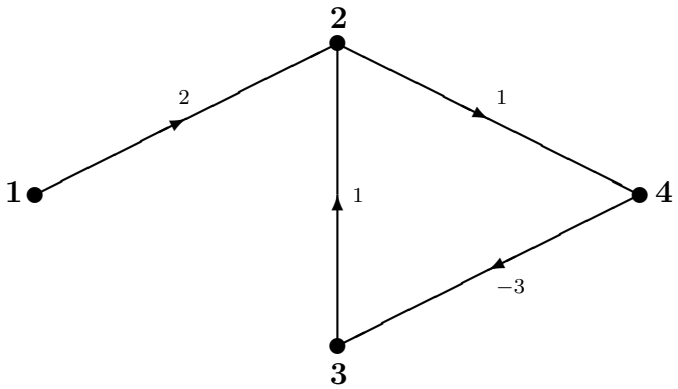


Abbildung 4.14: Es existiert ein Zyklus negativer Länge

z. B. von 1 nach 2, beliebig kurzer Länge, da man ja den gerichteten Kreis negativer Länge beliebig oft durchlaufen kann. Das spiegelt sich darin wieder, dass nach dem kleinen MATLAB-Programm

```
A=[1 0 0 0;-1 1 0 -1;0 0 -1 1;0 -1 1 0];
b=[1;0;0;-1];c=[2;1;-3;1];
options=optimset('largescale','off');
l=zeros(4,1);u=inf*ones(4,1);
[x,minkosten]=linprog(c,[],[],A,b,l,u,l,options);
```

die Antwort

```
Exiting: The solution is unbounded and at infinity;
         the constraints are not restrictive enough
```

gegeben wird. □

4.3.4 Der primal-duale Algorithmus

In Unterabschnitt 2.2.6 wurde der primal-duale Algorithmus vorgestellt. In Kurzform wollen wir das wichtigste wiederholen, wobei wir gleich die Beziehung zum Kürzester-Pfad-Problem herstellen.

Gegeben sei das lineare Programm in Standardform

$$\text{(P)} \qquad \text{Minimiere} \quad c^T x \quad \text{auf} \quad M := \{x \in \mathbb{R}^n : x \geq 0,\ Ax = b\},$$

wobei $A \in \mathbb{R}^{m \times n}$ den Rang m hat, weiter natürlich $b \in \mathbb{R}^m$ und $c \in \mathbb{R}^n$. Im konkreten Fall sei das Kürzester-Pfad-Problem in dem (schwach) zusammenhängenden gerichteten Graphen $(\mathcal{N}, \mathcal{A})$, dessen Kanten bewertet sind, zwischen Knoten $s, t \in \mathcal{N}$ gegeben. Wie wir wissen, lässt sich diese Aufgabe als ein lineares Programm in Standardform formulieren, bei dem $A \in \mathbb{Z}^{(|\mathcal{N}|-1) \times |\mathcal{A}|}$ die in der Zeile t gestutzte Knoten-Pfeil-Inzidenzmatrix, $b \in \mathbb{Z}^{|\mathcal{N}|-1}$ der Vektor, der nur in der Komponente s eine Eins und sonst nur Nullen hat (die in 2.2.6 gemachte Voraussetzung $b \geq 0$ ist dann erfüllt) und schließlich $c \in \mathbb{R}^{|\mathcal{A}|}$ der Vektor der Kantenbewertungen ist.

Das zu (P) duale Programm ist natürlich

$$\text{(D)} \qquad \text{Maximiere} \quad b^T y \quad \text{auf} \quad N := \{y \in \mathbb{R}^m : A^T y \leq c\}.$$

In unserem konkreten Fall lautet das duale Programm

$$\text{(D)} \qquad \text{Maximiere} \quad y_s \quad \text{auf} \quad N := \{y \in \mathbb{R}^{|\mathcal{N}|-1} : y_i - y_j \leq c_{ij},\ (i,j) \in \mathcal{A}\},$$

wobei wir in N formal $y_t := 0$ setzen.

Wir wiederholen (um das Blättern zu ersparen) die Schritte des primal-dualen Algorithmus.

1. Sei $y \in N$, also y dual zulässig.
2. Sei $J := \{j \in \{1, \ldots, n\} : a_j^T y = c_j\}$, $K := \{1, \ldots, n\} \setminus J$. Berechne Lösung (x_J, z) von

$$\text{(}\hat{\text{P}}\text{)} \qquad \begin{cases} \text{Minimiere} \begin{pmatrix} 0 \\ e \end{pmatrix}^T \begin{pmatrix} x_J \\ z \end{pmatrix} \quad \text{unter den Nebenbedingungen} \\ \begin{pmatrix} x_J \\ z \end{pmatrix} \geq 0, \quad \begin{pmatrix} A_J & I \end{pmatrix} \begin{pmatrix} x_J \\ z \end{pmatrix} = b. \end{cases}$$

3. Falls $\min(\hat{\mathrm{P}}) = 0$, dann: STOP. Setzt man $x_j := 0$ für $j \notin J$ bzw. $x_K := 0$, so sind x primal und y dual optimal.

4. Falls $\min(\hat{\mathrm{P}}) > 0$, so betrachte die zu $(\hat{\mathrm{P}})$ duale Aufgabe, nämlich

$$(\hat{\mathrm{D}}) \qquad \left\{ \begin{array}{c} \text{Maximiere} \quad b^T u \quad \text{unter den Nebenbedingungen} \\ \begin{pmatrix} A_J^T \\ I \end{pmatrix} u \le \begin{pmatrix} 0 \\ e \end{pmatrix}. \end{array} \right.$$

   Sei $\overline{u}$ eine Lösung von $(\hat{\mathrm{D}})$ (normalerweise wird $\overline{u}$ bei der Lösung von $(\hat{P})$ abfallen).

   (a) Falls $A_K^T \overline{u} \le 0$, dann: STOP, da $\sup(\mathrm{D}) = +\infty$ bzw. (P) nicht zulässig ist.

   (b) Falls $A_K^T \overline{u} \not\le 0$, so berechne

$$t^* := \min_{j \in K} \Big\{ \frac{c_j - a_j^T y}{a_j^T \overline{u}} : a_j^T \overline{u} > 0 \Big\}, \qquad y^+ := y + t^* \overline{u}.$$

   (Dann ist $y^+ \in N$, $t^* > 0$ und $b^T y^+ > b^T y$, also y^+ eine "bessere" dual zulässige Lösung.)

Nun spezialisieren wir diesen Algorithmus auf die Aufgabe, einen kürzesten Pfad von s nach t zu bestimmen. Sind alle Kantenbewertungen nichtnegativ, wovon wir jetzt ausgehen wollen, so ist $y := 0$ dual zulässig. Am Anfang kann also hiermit gestartet werden. In einem aktuellen Schritt bestimmt man die Indexmenge der in der aktuellen, dual zulässigen Lösung y aktiven Ungleichungsrestriktionen, also

$$J := \{(i,j) \in \mathcal{A} : y_i - y_j = c_{ij}\},$$

wobei weiter $y_t := 0$ gesetzt wird (das rührt her vom Streichen der Gleichung zum Knoten t). Also ist J eine Teilmenge der $|\mathcal{A}|$ Pfeile. Das zu dem restringierten primalen Problem

$$(\hat{\mathrm{P}}) \quad \text{Minimiere} \quad e^T z \quad \text{auf} \quad \hat{M} := \left\{ \begin{pmatrix} x_J \\ z \end{pmatrix} : \begin{pmatrix} x_J \\ z \end{pmatrix} \ge 0, \ \begin{pmatrix} A_J & I \end{pmatrix} \begin{pmatrix} x_J \\ z \end{pmatrix} = b \right\}$$

duale Programm ist

$$(\hat{\mathrm{D}}) \qquad \text{Maximiere} \quad b^T u \quad \text{auf} \quad \hat{N} := \{u \in \mathbb{R}^{|\mathcal{N}|-1} : A_J^T u \le 0, \ u \le e\}$$

bzw.

$$(\hat{\mathrm{D}}) \qquad \left\{ \begin{array}{c} \text{Maximiere} \quad u_s \quad \text{unter den Nebenbedingungen} \\ u_i - u_j \le 0 \quad ((i,j) \in J), \qquad u_i \le 1 \quad (i \in \mathcal{N} \setminus \{t\}), \end{array} \right.$$

wobei wieder $u_t := 0$ gesetzt ist. Nun kann man eine Lösung von $(\hat{\mathrm{D}})$ aber unmittelbar angeben. Hierzu unterscheiden wir zwei Fälle.

- Es gibt einen Pfad von s nach t mit Pfeilen aus J.

Dann ist $\overline{u} := 0$ eine Lösung von $(\hat{\mathrm{D}})$. Denn sei u zulässig. Nach Voraussetzung gibt es einen Pfad $(s, i_2), (i_2, i_3), \ldots, (i_{k-1}, t)$ von s nach t, der aus Pfeilen aus J besteht. Durch Aufsummieren der Ungleichungen

$$u_s - u_{i_2} \leq 0, \quad u_{i_2} - u_{i_3} \leq 0, \quad \cdots \quad , u_{i_{k-1}} - \underbrace{u_t}_{=0} \leq 0$$

erhält man $u_s \leq 0$, so dass $\overline{u} := 0$ in der Tat eine Lösung von $(\hat{\mathrm{D}})$ ist. Also ist $\max(\hat{\mathrm{D}}) = \min(\hat{\mathrm{P}}) = 0$. Jeder Pfad von s nach t mit Pfeilen aus J ist dann ein kürzester Pfad.

- Es gibt keinen Pfad von s nach t mit Pfeilen aus J.

Dann ist $\overline{u}$, definiert durch

$$\overline{u}_i := \begin{cases} 0, & \text{wenn } t \text{ von } i \text{ auf einem Pfad mit Pfeilen aus } J \text{ erreichbar ist,} \\ 1, & \text{sonst,} \end{cases}$$

eine Lösung von $(\hat{\mathrm{D}})$. Denn einerseits ist $\overline{u}_s = 1$ (ein größerer Wert ist wegen der Nebenbedingung $u_i \leq 1$ nicht möglich), andererseits ist $\overline{u}_i - \overline{u}_j \leq 0$ für alle $(i, j) \in J$ (mit $\overline{u}_t := 0$). Denn angenommen, es existiert ein $(i, j) \in J$ mit $\overline{u}_i - \overline{u}_j > 0$. Dann ist notwendig $\overline{u}_i = 1$ und $\overline{u}_j = 0$, d. h. t ist von j durch Pfeile aus J erreichbar, von i aber nicht. Da aber $(i, j) \in J$, ist das natürlich ein Widerspruch.

Anschließend definiere man

$$t^* := \min_{(i,j) \in \mathcal{A} \setminus J} \{c_{ij} - (y_i - y_j) : \overline{u}_i - \overline{u}_j > 0\}, \qquad y^+ := y + t^* \overline{u},$$

betrachte das neue Problem $(\hat{\mathrm{D}}^+)$ und fahre so fort, bis dieses den Wert Null besitzt bzw. die Gleichgewichtsbedingung erfüllt ist.

Beispiel: Wir betrachten noch einmal den in Abbildung 4.12 angegebenen bewerteten Digraphen, siehe auch Abbildung 4.15. Mit dem primal-dualen Algorithmus wollen wir

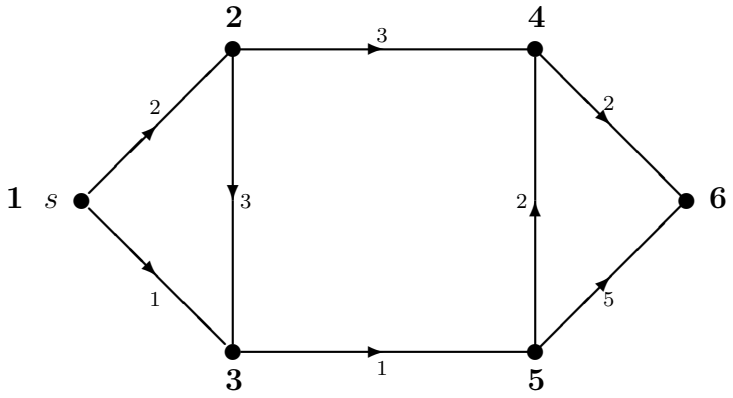


Abbildung 4.15: Ein gerichteter Graph mit Quelle $s = 1$

einen kürzesten Pfad von $s = 1$ nach $t = 6$ bestimmen. Wir erhalten die folgenden Iterationsschritte.

1. Sei $y := (0,0,0,0,0)^T$ und daher $J = \emptyset$ (alle Restriktionen sind inaktiv). Dann ist $\overline{u} = (1,1,1,1,1)^T$ und $t^* = 2$ (man beachte, dass $\overline{u}_t = 0$, so dass hier im ersten Schritt $t^* = \min_{(i,t)\in\mathcal{A}} c_{it}$, dieses Minimum wird durch den Pfeil $(4,6)$ mit den Kosten $c_{46} = 2$ angenommen).

2. Es ist $y = (2,2,2,2,2)^T$ und $J = \{(4,6)\}$. Dann ist $\overline{u} = (1,1,1,0,1)^T$ und $t^* = 2$ (hier braucht das Minimum nur über die Pfeile genommen zu werden, die im Knoten 4 enden, und das sind $(2,4)$ und $(5,4)$, wobei das Minimum auf $(5,4)$ angenommen wird).

3. Es ist $y = (4,4,4,2,4)^T$ und $J = \{(4,6),(5,4)\}$. Es ist $\overline{u} = (1,1,1,0,0)^T$ und $t^* = 1$. Denn: Hier braucht das Minimum nur über die Pfeile genommen zu werden, die nicht zu J gehören und im Knoten 4 oder 5 enden. Das sind $(2,4)$ und $(3,5)$. Also ist $t^* = \min[3-(4-2), 1-(4-4)] = 1$.

4. Es ist $y = (5,5,5,2,4)$ und $J = \{(4,6),(5,4),(3,5),(2,4)\}$. Die i-te Komponente von $\overline{u}$ ist genau dann gleich Null, wenn es vom Knoten i einen Pfad mit Pfeilen aus J zur Senke t gibt. Also ist $\overline{u} = (1,0,0,0,0)^T$ und $t^* = 1$.

5. Es ist $y = (6,5,5,2,4)$ und $J = \{(4,6),(5,4),(3,1),(2,4),(1,3)\}$. Jetzt kann man von der Quelle s zur Senke t mit einem Pfad aus Pfeilen in J gelangen, so dass $\overline{u} = (0,0,0,0,0)^T$. Ein Pfad von der Quelle zur Senke mit Pfeilen aus J ist ein kürzester Pfad, das ist offenbar $\{(1,3),(3,5),(5,4),(4,6)\}$.

□

4.3.5 Aufgaben

1. Man betrachte den bewerteten, gerichteten Graphen in Abbildung 4.16 und bestimme

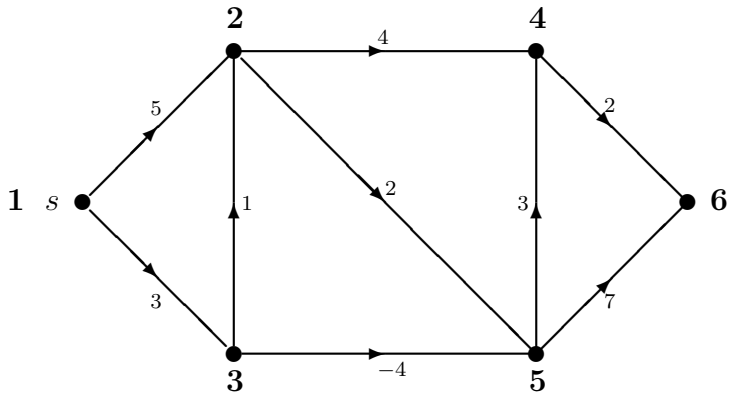


Abbildung 4.16: Ein gerichteter Graph mit Quelle $s = 1$

die kürzesten Entfernungen und die kürzesten Pfade vom Knoten 1 zu allen anderen Knoten. Hierbei wende man zunächst das Verfahren von Dijkstra (obwohl nicht alle Längen nichtnegativ sind), danach das Verfahren von Bellman-Ford an.

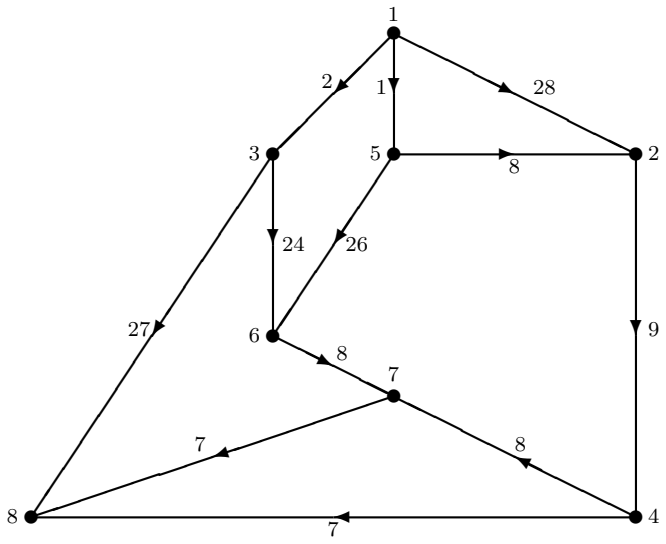


Abbildung 4.17: Was ist der kürzeste Pfad von 1 zu den übrigen Knoten?

2. Gegeben sei der Digraph in Abbildung 4.17. Die Kosten zwischen zwei Knoten sind eingetragen. Mit Hilfe des Dijkstra-Verfahrens bestimme man einen kürzesten Pfad zwischen 1 und den übrigen Knoten.

3. Gegeben sei der Digraph in Abbildung 4.17. Man bestimme einen kürzesten Pfad vom Knoten 1 zum Knoten 8 mit Hilfe des primal-dualen Algorithmus.

4.4 Das Maximaler-Fluss-Problem

Sei $(\mathcal{N}, \mathcal{A})$ ein gerichteter Graph, für jeden Pfeil $(i, j) \in \mathcal{A}$ seien Kapazitätsschranken u_{ij} für einen zulässigen Fluss $x = (x_{ij})_{(i,j)\in\mathcal{A}}$ gegeben. Schließlich seien zwei spezielle Knoten s und t, die Quelle und die Senke, gegeben, wobei wir davon ausgehen, dass kein Pfeil in s endet und keiner in t startet. Der bei s austretende Fluss ist

$$v = \sum_{j:(s,j)\in\mathcal{A}} x_{sj}.$$

Dieser ist beim Maximal-Fluss-Problem zu maximieren unter der Kapazitätsbeschränkung

$$0 \le x_{ij} \le u_{ij}, \qquad (i, j) \in \mathcal{A},$$

und der Flussbedingung ($\pi\alpha\nu\tau\alpha\ \rho\epsilon\iota$)

$$\sum_{j:(k,j)\in\mathcal{A}} x_{kj} - \sum_{i:(i,k)\in\mathcal{A}} x_{ik} = 0, \qquad k \in \mathcal{N} \setminus \{s, t\},$$

dass nämlich in jedem Knoten außer Quelle und Senke alles abfließt, was ankommt. Hierbei wird u_{ij} als nichtnegativ, aber zunächst nicht notwendig als ganzzahlig vorausgesetzt. Wir hatten früher schon bemerkt, dass man ein Maximaler-Fluss-Problem

als ein Minimale-Kosten-Problem schreiben kann. Zur Erinnerung: Man führe einen zusätzlichen Pfeil (t, s) mit der Kapazität $u_{ts} := \infty$ ein, fasse alle Knoten als Umladeknoten auf, definiere die Kosten $c_{ij} := 0$, $(i, j) \in \mathcal{A}$, und $c_{ts} := -1$. Dann kann das Maximaler-Fluss-Problem auch als

$$\left\{\begin{array}{l} \text{Minimiere} \quad -x_{ts} \quad \text{unter den Nebenbedingungen} \\ \displaystyle\sum_{j:(s,j)\in\mathcal{A}} x_{sj} - x_{ts} = 0, \\ \displaystyle\sum_{j:(k,j)\in\mathcal{A}} x_{kj} - \sum_{i:(i,k)\in\mathcal{A}} x_{ik} = 0, \quad k \in \mathcal{N} \setminus \{s,t\}, \\ \displaystyle x_{ts} - \sum_{i:(i,t)\in\mathcal{A}} x_{it} = 0, \\ 0 \le x_{ij} \le u_{ij}, \quad ((i,j) \in \mathcal{A}), \quad 0 \le x_{ts}. \end{array}\right.$$

Dies hat den Vorteil, dass wir jetzt auch für das Maximaler-Fluss-Problem wissen, dass ganzzahlige Kapazitäten (wenn eine Lösung existiert) die Existenz einer ganzzahligen Lösung sichern.

Bemerkung: Denkbar wäre auch eine beidseitige Kapazitätsbeschränkung der Form

$$l_{ij} \le x_{ij} \le u_{ij}, \qquad (i,j) \in \mathcal{A},$$

für einen zulässigen Fluss x. Hierbei können gewisse der l_{ij} und u_{ij} auch gleich $-\infty$ bzw. $+\infty$ sein. Man überlege sich selber, wie die folgenden Schlüsse zu modifizieren sind. □

Bemerkung: Bevor wir auf das berühmte Max-Flow/Min-Cut Theorem von Ford-Fulkerson eingehen, wollen wir zeigen, dass man das Matching-Problem in einem bipartiten Graphen auf ein Maximaler-Fluss-Problem zurückführen kann.

Gegeben sei ein bipartiter Graph $G = (U \cup W, E)$, gesucht sei ein Maximum-Matching in G. Auf die folgende Weise erhalte man einen gerichteten Graphen $(\mathcal{N}, \mathcal{A})$. Man wähle zwei Knoten $s \neq t$, die nicht in $U \cup W$ enthalten sind und setze $\mathcal{N} := \{s\} \cup U \cup W \cup \{t\}$. Die Pfeilmenge $\mathcal{A}$ bestehe aus (s, u) für alle $u \in U$, den gerichteten Kanten (u, w) mit $u \in U$, $w \in W$, für die $(u, w) \in E$, und (w, t) für alle $w \in W$. Als Kapazitätsschranken auf den Pfeilen aus $\mathcal{A}$ setze man $u_{su} := 1$ und $u_{wt} := 1$, während die Kapazitätsschranken auf den Pfeilen (u, w) gleich $+\infty$ sei. Wir wollen uns überlegen:

- Die Matchingzahl $m(G)$ des bipartiten Graphen $G = (U \cup W, E)$ ist gleich dem Wert des maximalen Flusses in dem oben definierten gerichteten Graphen $(\mathcal{N}, \mathcal{A})$.

Denn: Zu einem Matching F in G kann man auf die folgende Weise einen zulässigen Fluss x in $(\mathcal{N}, \mathcal{A})$ bestimmen: Für jedes $u \in U$, welches durch F gematcht ist, zu dem es also ein $w \in W$ mit $(u, w) \in F$ gibt, setze man $x_{su} := 1$. Entsprechend setze man $x_{wt} := 1$ für alle durch F gematchten $w \in W$, und weiter $x_{uw} := 1$ für alle $(u, w) \in F$. Die Flüsse auf allen anderen Pfeilen seien gleich Null. Der hierdurch definierte Fluss ist offensichtlich zulässig (die Kapazitätsbedingung ist trivialerweise

erfüllt, die Flussbedingung ebenfalls), sein Wert v ist gleich $|F|$. Daher existiert in $(\mathcal{N}, \mathcal{A})$ ein Fluss x^* mit dem Wert $v^* = m(G)$ und es ist $\max(\mathrm{D}) \geq m(G)$.

Umgekehrt beachte man, dass es in $(\mathcal{N}, \mathcal{A})$ einen optimalen ganzzahligen Fluss x^* mit Wert $v^* = \max(\mathrm{D})$ gibt (siehe Bemerkung im Anschluss an Satz 2.11), der Fluss auf jedem Pfeil ist also 0 oder 1. Sei F^* die Menge aller Kanten $(u, w) \in E$ mit $x^*_{uw} = 1$. Dann ist F^* ein Matching in dem bipartiten Graphen $G = (U \cup W, E)$, ferner ist $|F^*| = v^*$. Denn hätten zwei verschiedene Kanten in F eine Ecke gemein, so würde man in dieser Ecke (bzw. Knoten) einen Verstoß der Flussgleichung feststellen. Folglich ist $v^* = \max(\mathrm{D}) \leq m(G)$. Insgesamt ist die Behauptung bewiesen. □

Beispiel: In Abbildung 4.18 links haben wir einen bipartiten Graphen und einen zu-

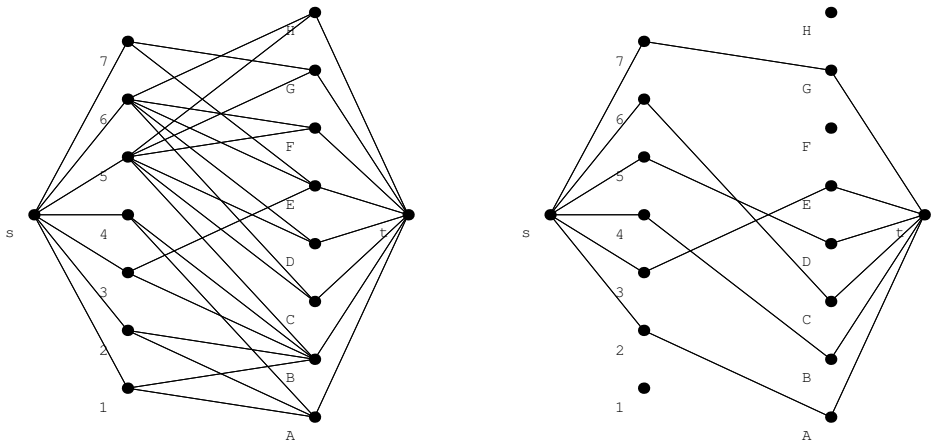


Abbildung 4.18: Matching und maximaler Fluss

gehörigen Digraphen mit Quelle s und Senke t angegeben (aus technischen Gründen verzichten wir auf eine Darstellung der Pfeilrichtungen, diese gehen aber sämtlich von links nach rechts). Der bipartite Graph stimmt (bis auf die Reihenfolge der Ecken) mit dem in Abbildung 3.37 überein. In Abbildung 3.39 hatten wir ein Maximum-Matching angegeben, siehe auch Abbildung 4.18 rechts, wo wir auch einen maximalen Fluss eintragen: Auf angezeigten Kanten ist der Fluss 1, auf den übrigen ist er 0. Die Matching Zahl $m(G)$ und der maximale Fluss v^* in dem Digraphen $(\mathcal{N}, \mathcal{A})$ ist jeweils 6. Außerdem gibt es sechs Pfade von s nach t, die keinen Pfeil gemein haben, was offenbar die Maximalzahl von Pfaden mit dieser Eigenschaft ist. Dass diese Zahlen jeweils übereinstimmen ist kein Zufall. Wir kommen hierauf im Zusammenhang mit einem Satz von Menger in allgemeinen Digraphen im Anschluss an das Max-Flow Min-Cut Theorem zurück. □

4.4.1 Das Max-Flow Min-Cut Theorem

Bezeichnet man mit $A \in \mathbb{R}^{|\mathcal{N}| \times |\mathcal{A}|}$ die Knoten-Pfeil-Inzidenzmatrix von $(\mathcal{N}, \mathcal{A})$ (bei einer gewissen Nummerierung der Knoten und Pfeile), definiert man ferner $d \in \mathbb{R}^{|\mathcal{A}|}$

durch

$$d_k := \begin{cases} -1, & k = s, \\ 0, & k \neq s, t \\ 1, & k = t, \end{cases}$$

so kann das Maximaler-Fluss-Problem als ein lineares Programm in der Form

$$\text{(D)} \qquad \text{Maximiere} \quad v \quad \text{auf} \quad \{(x, v) \in \mathbb{R}^{|\mathcal{A}|} \times \mathbb{R} : Ax + dv = 0,\ 0 \leq x \leq u\}$$

geschrieben werden. Nun erkennt man leicht, dass (D) gerade das duale Problem zu

$$\text{(P)} \qquad \begin{cases} \text{Minimiere} \quad u^T w \quad \text{auf} \\ M := \{(y, w) \in \mathbb{R}^{|\mathcal{N}|} \times \mathbb{R}^{|\mathcal{A}|} : w \geq 0,\ A^T y + w \geq 0,\ d^T y = 1\} \end{cases}$$

ist. Wir wollen beide Probleme sozusagen auch noch komponentenweise formulieren. Äquivalente Formulierungen sind

$$\text{(P)} \qquad \begin{cases} \text{Minimiere} \quad \sum_{(i,j)\in\mathcal{A}} u_{ij} w_{ij} \quad \text{unter den Nebenbedingungen} \\ \qquad w_{ij} \geq 0, \qquad (i,j) \in \mathcal{A}, \\ \qquad y_i - y_j + w_{ij} \geq 0, \qquad (i,j) \in \mathcal{A}, \\ \qquad -y_s + y_t = 1 \end{cases}$$

und

$$\text{(D)} \qquad \begin{cases} \text{Maximiere} \quad \sum_{j:(s,j)\in\mathcal{A}} x_{sj} \quad \text{unter den Nebenbedingungen} \\ \qquad \sum_{i:(i,k)\in\mathcal{A}} x_{ik} = \sum_{j:(k,j)\in\mathcal{A}} x_{kj}, \qquad k \in \mathcal{N} \setminus \{s, t\}, \\ \qquad 0 \leq x_{ij} \leq u_{ij}, \qquad (i,j) \in \mathcal{A}. \end{cases}$$

Ist in dieser Notation $x = (x_{ij})_{(i,j)\in\mathcal{A}}$ ein zulässiger Fluss, so ist sein *Wert* durch $v := \sum_{j:(s,j)\in\mathcal{A}} x_{sj}$ gegeben.

Bisher haben wir das Maximaler-Fluss-Problem (max-flow problem) geschildert und gezeigt, dass es sich in natürlicher Weise als eine lineare Optimierungsaufgabe formulieren lässt. Diese besitzt eine Lösung, denn die zulässige Menge in (D) ist nichtleer (der triviale Fluss $x = 0$ genügt allen Nebenbedingungen) und kompakt (jedenfalls dann, wenn alle Kapazitätsschranken endlich sind).

Nun kommen wir zum Minimaler-Schnitt-Problem (min-cut problem). Wieder ist der gerichtete Graph $(\mathcal{N}, \mathcal{A})$ mit den zwei ausgezeichneten Knoten s (Quelle) und t (Senke) sowie (i. allg. positive) Kapazitäten u_{ij} längs der Pfeile $(i,j) \in \mathcal{A}$ gegeben. Ein *Schnitt* ist eine Partition der Knotenmenge $\mathcal{N}$ in zwei (disjunkte) Mengen $\mathcal{N}_1$ und $\mathcal{N}_2$ mit $s \in \mathcal{N}_1$ und $t \in \mathcal{N}_2$. Zu einem Schnitt $(\mathcal{N}_1, \mathcal{N}_2)$ definieren wir die zugehörige *Kapazität* $C(\mathcal{N}_1, \mathcal{N}_2)$ als die Summe aller Kapazitätsschranken über Pfeilen, die in $\mathcal{N}_1$ starten und in $\mathcal{N}_2$ enden, also in der oben eingeführten Notation durch

$$C(\mathcal{N}_1, \mathcal{N}_2) := \sum_{\substack{(i,j)\in\mathcal{A} \\ i\in\mathcal{N}_1,\, j\in\mathcal{N}_2}} u_{ij}.$$

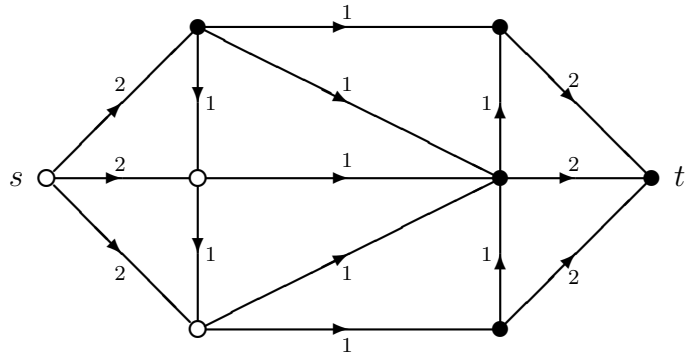


Abbildung 4.19: Ein Schnitt mit der Kapazität 5

Unter dem *Minimaler-Schnitt-Problem* (min-cut problem) versteht man die Aufgabe, einen Schnitt mit minimaler Kapazität zu bestimmen.

In Abbildung 4.19 geben wir einen Schnitt an. Die zu $\mathcal{N}_1$ gehörenden Knoten sind durch $\circ$, solche zu $\mathcal{N}_2$ durch $\bullet$ gekennzeichnet. Hier gibt es vier Pfeile, die Knoten aus $\mathcal{N}_1$ mit Knoten aus $\mathcal{N}_2$ verbinden, die zugehörige Kapazität ist 5.

Unser Ziel ist es, das Max-Flow Min-Cut Theorem von Ford-Fulkerson zu beweisen. Dieses sagt unter Benutzung obiger Bezeichnungen aus:

Satz 4.1 *Es gilt:*

(a) Ist $x = (x_{ij})_{(i,j)\in\mathcal{A}}$ ein zulässiger Fluss mit dem Wert $v = \sum_{j:(s,j)\in\mathcal{A}} x_{sj}$ und ist $(\mathcal{N}_1, \mathcal{N}_2)$ ein Schnitt mit Kapazität $C(\mathcal{N}_1, \mathcal{N}_2)$, so ist $v \leq C(\mathcal{N}_1, \mathcal{N}_2)$.

(b) Ist x^* eine Lösung des Maximaler-Fluss-Problems mit dem Wert

$$v^* = \sum_{j:(s,j)\in\mathcal{A}} x^*_{sj},$$

so existiert ein Schnitt $(\mathcal{N}_1^*, \mathcal{N}_2^*)$ mit der Kapazität $C(\mathcal{N}_1^*, \mathcal{N}_2^*) = v^*$. Dieser Schnitt ist eine Lösung des Minimaler-Schnitt-Problems.

(c) Ist $(\mathcal{N}_1^*, \mathcal{N}_2^*)$ eine Lösung des Minimaler-Schnitt-Problems (da es nur endlich viele Schnitte gibt, und mindestens einen, wenn es eine die Quelle und die Senke verbindende Pfeilfolge gibt, hat das Minimaler-Schnitt-Problem trivialerweise eine Lösung), so gibt es einen zulässigen Fluss x^* mit dem Wert $v^* = C(\mathcal{N}_1^*, \mathcal{N}_2^*)$. Dieser Fluss ist eine Lösung des Maximaler-Fluss-Problems.

(d) Ein zulässiger Fluss x^* und ein Schnitt $(\mathcal{N}_1^*, \mathcal{N}_2^*)$ sind genau dann optimal für das Maximaler-Fluss- bzw. das Minimaler-Schnitt-Problem, wenn

$$x^*_{ij} = \begin{cases} 0 & \text{falls } (i,j) \in \mathcal{A} \text{ mit } i \in \mathcal{N}_2^* \text{ und } j \in \mathcal{N}_1^*, \\ u_{ij} & \text{falls } (i,j) \in \mathcal{A} \text{ mit } i \in \mathcal{N}_1^* \text{ und } j \in \mathcal{N}_2^*. \end{cases}$$

Beweis: Zum Beweis von (a) definieren wir zu dem Schnitt $(\mathcal{N}_1, \mathcal{N}_2)$ ein Paar $(y, w) \in \mathbb{R}^{|\mathcal{N}|} \times \mathbb{R}^{|\mathcal{A}|}$ durch

$$y_k := \begin{cases} 0 & \text{für alle } k \in \mathcal{N}_1, \\ 1 & \text{für alle } k \in \mathcal{N}_2. \end{cases}$$

$$w_{ij} := \begin{cases} 1 & \text{für alle } (i,j) \in \mathcal{A} \text{ mit } i \in \mathcal{N}_1 \text{ und } j \in \mathcal{N}_2, \\ 0 & \text{für alle anderen } (i,j) \in \mathcal{A}, \end{cases}$$

Offenbar ist (y, w) für das obige lineare Programm (P) zulässig und

$$C(\mathcal{N}_1, \mathcal{N}_2) = \sum_{(i,j)\in\mathcal{A}} u_{ij} w_{ij}.$$

Der schwache Dualitätssatz der linearen Optimierung liefert[14] (a).

Zum Beweis von (b) nehmen wir an, x^* sei Lösung des Maximaler-Fluss-Problems mit zugehörigem Wert v^*. Dann ist (x^*, v^*) auch eine Lösung des linearen Programms (D). Wegen des starken Dualitätssatzes der linearen Optimierung besitzt das lineare Programm (P) eine Lösung (y^*, w^*) mit

$$\sum_{(i,j)\in\mathcal{A}} u_{ij} w^*_{ij} = v^*.$$

[14]Natürlich kann man den Beweis auch direkt, ohne den Umweg über den auf (P) und (D) angewandten schwachen Dualitätssatz führen. Sei hierzu x ein zulässiger Fluss und $(\mathcal{N}_1, \mathcal{N}_2)$ ein Schnitt. Dann gilt:

$$\begin{aligned}
v &= \sum_{j:(s,j)\in\mathcal{A}} x_{sj} - \underbrace{\sum_{i:(i,s)\in\mathcal{A}} x_{is}}_{=0} + \sum_{k\in\mathcal{N}_1\setminus\{s\}} \Bigg(\underbrace{\sum_{j:(k,j)\in\mathcal{A}} x_{kj} - \sum_{i:(i,k)\in\mathcal{A}} x_{ik}}_{=0} \Bigg) \\
&\qquad \text{(Definition von } v \text{ und Aufsummieren der Flussgleichung für alle } k \in \mathcal{N}_1 \setminus \{s\}.) \\
&= \sum_{k\in\mathcal{N}_1} \Bigg(\sum_{j:(k,j)\in\mathcal{A}} x_{kj} - \sum_{i:(i,k)\in\mathcal{A}} x_{ik} \Bigg) \\
&= \sum_{k\in\mathcal{N}_1} \Bigg(\sum_{\substack{j\in\mathcal{N}_2 \\ (k,j)\in\mathcal{A}}} x_{kj} - \sum_{\substack{i\in\mathcal{N}_2 \\ (i,k)\in\mathcal{A}}} x_{ik} \Bigg) + \underbrace{\sum_{k\in\mathcal{N}_1} \Bigg(\sum_{\substack{j\in\mathcal{N}_1 \\ (k,j)\in\mathcal{A}}} x_{kj} - \sum_{\substack{i\in\mathcal{N}_1 \\ (i,k)\in\mathcal{A}}} x_{ik} \Bigg)}_{=0} \\
&\leq \sum_{k\in\mathcal{N}_1} \sum_{\substack{j\in\mathcal{N}_2 \\ (k,j)\in\mathcal{A}}} x_{kj} \\
&= \sum_{i\in\mathcal{N}_1} \sum_{\substack{j\in\mathcal{N}_2 \\ (i,j)\in\mathcal{A}}} x_{ij} \\
&\leq \sum_{\substack{(i,j)\in\mathcal{A} \\ i\in\mathcal{N}_1,\, j\in\mathcal{N}_2}} u_{ij} \\
&= C(\mathcal{N}_1, \mathcal{N}_2).
\end{aligned}$$

Diese Argumentation wird auch beim Beweis von (b) und in (d) eine Rolle spielen.

Wir konstruieren einen Schnitt $(\mathcal{N}_1^*, \mathcal{N}_2^*)$ mit

$$v^* = C(\mathcal{N}_1^*, \mathcal{N}_2^*),$$

womit wir dann auch (b) bewiesen haben, da ein solcher Schnitt wegen (a) notwendigerweise eine Lösung des Minimaler-Schnitt-Problems ist. O. B. d. A. können wir annehmen, dass $y_s^* = 0$ (und dann $y_t^* = 1$), da man notfalls y^* durch $y^* - y_s^* e$ ersetzen kann, ohne etwas an der Optimalität von (y^*, w^*) zu ändern. Wir definieren

$$\mathcal{N}_1^* := \{k \in \mathcal{N} : y_k^* < 1\}, \qquad \mathcal{N}_2^* := \{k \in \mathcal{N} : y_k^* \geq 1\},$$

wodurch offenbar ein Schnitt im Digraphen $(\mathcal{N}, \mathcal{A})$, gegeben ist. Zunächst beachten wir, dass die Gleichgewichtsbedingungen zwischen den beiden zueinander dualen linearen Programmen die Beziehungen

$$w_{ij}^*[u_{ij} - x_{ij}^*] = 0, \qquad x_{ij}^*[y_i^* - y_j^* + w_{ij}^*] = 0, \qquad ((i,j) \in \mathcal{A})$$

liefern. Um diese Beziehungen, die ja für alle Pfeile aus $\mathcal{A}$ gelten, auszunutzen, unterscheiden wir zwischen "Vorwärts-Pfeilen" und "Rückwärts-Pfeilen" bezüglich des Schnittes $(\mathcal{N}_1^*, \mathcal{N}_2^*)$. Hierbei nennen wir ein $(i,j) \in \mathcal{A}$ einen *Vorwärts-Pfeil* (bezüglich $(\mathcal{N}_1^*, \mathcal{N}_2^*)$), wenn $i \in \mathcal{N}_1^*$ und $j \in \mathcal{N}_2^*$. Für einen solchen ist

$$w_{ij}^* \geq \underbrace{y_j^*}_{\geq 1} - \underbrace{y_i^*}_{<1} > 0$$

und daher $x_{ij}^* = u_{ij}$. Ist dagegen $(i,j) \in \mathcal{A}$ ein "Rückwärts-Pfeil", also $i \in \mathcal{N}_2^*$ und $j \in \mathcal{N}_1^*$, so ist

$$y_i^* - y_j^* + \underbrace{w_{ij}^*}_{\geq 0} \geq \underbrace{y_i^*}_{\geq 1} - \underbrace{y_j^*}_{<1} > 0,$$

und folglich $x_{ij}^* = 0$. Der Trick besteht jetzt darin, den maximalen Fluss als Summe der Flüsse auf den Vorwärts-Pfeilen bzw. (wegen $x_{ij}^* = u_{ij}$ auf den Vorwärts-Pfeilen) als Kapazität des Schnittes $(\mathcal{N}_1^*, \mathcal{N}_2^*)$ zu "entlarven". Es ist nämlich

$$\begin{aligned}
v^* &= \sum_{j:(s,j)\in\mathcal{A}} x_{sj}^* - \underbrace{\sum_{i:(i,s)\in\mathcal{A}} x_{is}^*}_{=0} \\
&\qquad \text{(In der Quelle } s \text{ endet kein Pfeil)} \\
&= \sum_{j:(s,j)\in\mathcal{A}} x_{sj}^* - \sum_{i:(i,s)\in\mathcal{A}} x_{is}^* + \sum_{k\in\mathcal{N}_1^*\setminus\{s\}} \underbrace{\Big(\sum_{j:(k,j)\in\mathcal{A}} x_{kj}^* - \sum_{i:(i,k)\in\mathcal{A}} x_{ik}^* \Big)}_{=0} \\
&\qquad \text{(Flussbedingung)} \\
&= \sum_{k\in\mathcal{N}_1^*} \Big(\sum_{j:(k,j)\in\mathcal{A}} x_{kj}^* - \sum_{i:(i,k)\in\mathcal{A}} x_{ik}^* \Big) \\
&= \sum_{k\in\mathcal{N}_1^*} \Big(\sum_{\substack{j\in\mathcal{N}_2^*: \\ (k,j)\in\mathcal{A}}} \underbrace{x_{kj}^*}_{=u_{kj}} - \sum_{\substack{i\in\mathcal{N}_2^*: \\ (i,k)\in\mathcal{A}}} \underbrace{x_{ik}^*}_{=0} \Big) + \underbrace{\sum_{k\in\mathcal{N}_1^*} \Big(\sum_{\substack{j\in\mathcal{N}_1^*: \\ (k,j)\in\mathcal{A}}} x_{kj}^* - \sum_{\substack{i\in\mathcal{N}_1^*: \\ (i,k)\in\mathcal{A}}} x_{ik}^* \Big)}_{=0}
\end{aligned}$$

$$
\begin{aligned}
&= \sum_{i\in\mathcal{N}_1^*} \sum_{\substack{j\in\mathcal{N}_2^*:\\(i,j)\in\mathcal{A}}} u_{ij} \\
&= C(\mathcal{N}_1^*, \mathcal{N}_2^*),
\end{aligned}
$$

womit schließlich auch (b) bewiesen ist.

In (c) wird angenommen, $(\mathcal{N}_1^*, \mathcal{N}_2^*)$ sei eine Lösung des Minimaler-Schnitt-Problems. Sei x^* eine Lösung des Maximaler-Fluss-Problems mit dem Fluss v^*. Nach (b) existiert zu x^* ein Schnitt $(\mathcal{N}_1^{**}, \mathcal{N}_2^{**})$ mit $v^* = C(\mathcal{N}_1^{**}, \mathcal{N}_2^{**})$. Wegen (a) ist

$$C(\mathcal{N}_1^{**}, \mathcal{N}_2^{**}) = v^* \leq C(\mathcal{N}_1^*, \mathcal{N}_2^*).$$

Da weiter $(\mathcal{N}_1^*, \mathcal{N}_2^*)$ eine Lösung des Minimalschnittproblems ist, ist $v^* = C(\mathcal{N}_1^*, \mathcal{N}_2^*)$, also mit x^* der gesuchte Fluss gefunden.

In (d) haben wir nur noch einmal das zusammengefasst, was wir wegen der Gleichgewichtsbedingung schon wissen. Ist nämlich x^* ein Fluss, v^* sein Wert und $(\mathcal{N}_1^*, \mathcal{N}_2^*)$ ein Schnitt mit

$$x_{ij}^* = \begin{cases} 0 & \text{falls } (i,j) \in \mathcal{A} \text{ mit } i \in \mathcal{N}_2^* \text{ und } j \in \mathcal{N}_1^*, \\ u_{ij} & \text{falls } (i,j) \in \mathcal{A} \text{ mit } i \in \mathcal{N}_1^* \text{ und } j \in \mathcal{N}_2^*, \end{cases}$$

so ist (siehe obige Rechnung)

$$v^* = \sum_{k\in\mathcal{N}_1^*} \Big(\sum_{\substack{j\in\mathcal{N}_2^*:\\(k,j)\in\mathcal{A}}} \underbrace{x_{kj}^*}_{=u_{kj}} - \sum_{\substack{i\in\mathcal{N}_2^*:\\(i,k)\in\mathcal{A}}} \underbrace{x_{ik}^*}_{=0} \Big) = C(\mathcal{N}_1^*, \mathcal{N}_2^*),$$

also x^* und $(\mathcal{N}_1^*, \mathcal{N}_2^*)$ optimal für das Maximaler-Fluss- bzw. das Minimaler-Schnitt-Problem. Sind umgekehrt x^* und $(\mathcal{N}_1^*, \mathcal{N}_2^*)$ optimal, so ist $v^* = C(\mathcal{N}_1^*, \mathcal{N}_2^*)$ und die Behauptung folgt (siehe einmal wieder obige Rechnung) aus

$$v^* = \sum_{i\in\mathcal{N}_1^*} \Big(\sum_{\substack{j\in\mathcal{N}_2^*:\\(i,j)\in\mathcal{A}}} x_{ij}^* - \sum_{\substack{j\in\mathcal{N}_2^*:\\(j,i)\in\mathcal{A}}} x_{ji}^* \Big) = \sum_{i\in\mathcal{N}_1^*} \sum_{\substack{j\in\mathcal{N}_2^*:\\(i,j)\in\mathcal{A}}} u_{ij} = C(\mathcal{N}_1^*, \mathcal{N}_2^*),$$

dass

$$0 = \sum_{i\in\mathcal{N}_1^*} \sum_{\substack{j\in\mathcal{N}_2^*:\\(i,j)\in\mathcal{A}}} \underbrace{(u_{ij} - x_{ij}^*)}_{\geq 0} + \sum_{\substack{j\in\mathcal{N}_2^*:\\(j,i)\in\mathcal{A}}} \underbrace{x_{ji}^*}_{\geq 0}$$

und daher $x_{ij}^* = u_{ij}$ für alle $(i,j) \in \mathcal{A}$ mit $i \in \mathcal{N}_1^*$, $j \in \mathcal{N}_2^*$ und $x_{ij}^* = 0$ für alle $(i,j) \in \mathcal{A}$ mit $i \in \mathcal{N}_2^*$, $j \in \mathcal{N}_1^*$. Damit ist der Satz schließlich bewiesen. □

Bemerkung: Sind die oberen Kapazitätsschranken u_{ij}, $(i,j) \in \mathcal{A}$, ganzzahlig, so besitzt das Maximaler-Fluss-Problem eine ganzzahlige Lösung. Man sehe sich hierzu noch einmal Satz 2.11 und die anschließende Bemerkung an. Nun nehmen wir an, es sei speziell $u_{ij} := 1$, $(i,j) \in \mathcal{A}$. Wie wir eben bemerkt haben, besitzt das Maximaler-Fluss-Problem eine ganzzahlige Lösung x^*, es ist also $x^* \in \{0,1\}^{|\mathcal{A}|}$. Der Wert $v^* = \sum_{j:(s,j)\in\mathcal{A}} x_{sj}^*$ ist gleich der Anzahl der Pfade von s nach t, die paarweise

keinen Pfeil gemein haben. Nach dem Max-Flow/Min-Cut Theorem ist andererseits v^* gleich der minimalen Kapazität $C(\mathcal{N}_1, \mathcal{N}_2)$ eines Schnittes bzw. der minimalen Länge $|\{(i,j) \in \mathcal{A} : i \in \mathcal{N}_1,\ j \in \mathcal{N}_2\}|$ aller Schnitte $(\mathcal{N}_1, \mathcal{N}_2)$. Dieses ist eine Version eines Satzes von Menger (siehe A. SCHRIJVER (1986, S. 275)):

- Sei $(\mathcal{N}, \mathcal{A})$ ein Digraph mit Quelle $s \in \mathcal{N}$ und Senke $t \in \mathcal{N}$. Dann ist die maximale Zahl der Pfade von s nach t, welche paarweise keinen Pfeil gemein haben, gleich der minimalen Länge $|\{(i,j) \in \mathcal{A} : i \in \mathcal{N}_1,\ j \in \mathcal{N}_2\}|$ aller Schnitte $(\mathcal{N}_1, \mathcal{N}_2)$ (natürlich bezüglich der ausgezeichneten Knoten s und t).

Als Beispiel zum Satz von Menger betrachten wir den in Abbildung 4.20 angegebenen

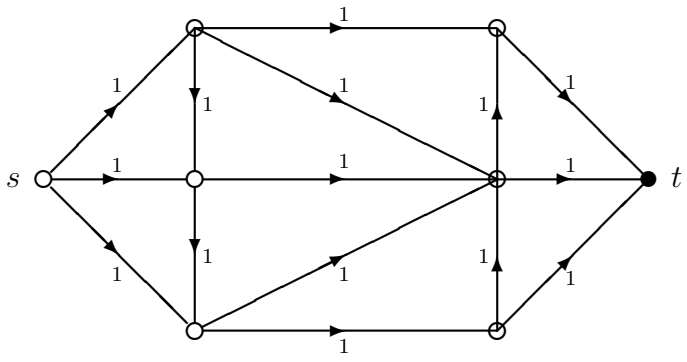


Abbildung 4.20: Beispiel zum Satz von Menger

Digraphen, bei dem alle Pfeile die Kapazität 1 tragen. Hier gibt es offenbar drei Pfade von s nach t, die keinen Pfeil gemein haben, und das ist die Maximalzahl. Fasst man in $\mathcal{N}_1$ alle Knoten bis aus t zusammen und setzt $\mathcal{N}_2 := \{t\}$, so hat der Schnitt $(\mathcal{N}_1, \mathcal{N}_2)$ die Kapazität bzw. Länge 3. □

Bemerkung: In einer Bemerkung unmittelbar vor Beginn dieses Unterabschnitts hatten wir uns überlegt, dass die Matchingzahl $m(G)$ eines bipartiten Graphen $G = (U \cup W, E)$ gleich dem maximalen Fluss in einem gerichteten Graphen $(\mathcal{N}, \mathcal{A})$ ist, den man aus G dadurch erhält, dass man $\mathcal{N} := U \cup W \cup \{s,t\}$ mit neuen Knoten s, t setzt und als Pfeilmenge $\mathcal{A}$ gerade (s,u) für alle $u \in U$, (u,w) für $u \in U$, $w \in W$ mit $(u,w) \in E$ und (w,t) für alle $w \in W$ nimmt. Die Kapazitäten auf Pfeilen (s,u) bzw. (w,t) seien gleich 1, die auf Pfeile (u,w) gleich $+\infty$. Sei $(\mathcal{N}_1^*, \mathcal{N}_2^*)$ mit $s \in \mathcal{N}_1^*$, $t \in \mathcal{N}_2^*$ ein minimaler Schnitt mit der Kapazität $C(\mathcal{N}_1^*, \mathcal{N}_2^*)$. Da die Kapazität endlich ist und Pfeile (u,w) mit $u \in U$, $w \in W$, die Kapazitätsschranke $+\infty$ haben, gibt es keine Kante in G mit einer Ecke in $(\mathcal{N}_1^* \setminus \{s\}) \cap U$ und der anderen in $W \setminus (\mathcal{N}_1^* \setminus \{s\})$. Daher inzidiert jede Kante in G mit einer Ecke in

$$D^* := (U \setminus (\mathcal{N}_1^* \setminus \{s\}) \cup (W \cap (\mathcal{N}_1^* \setminus \{s\})),$$

mit anderen Worten ist D^* ein Träger von G. Offensichtlich ist

$$|D^*| = |U \setminus (\mathcal{N}_1^* \setminus \{s\})| + |W \cap (\mathcal{N}_1^* \setminus \{s\})| = C(\mathcal{N}_1^*, \mathcal{N}_2^*).$$

Da andererseits die Matching-Zahl $m(G)$ gleich dem maximalen Fluss v^* in $(\mathcal{N}, \mathcal{A})$ von s nach t ist, erhält man aus dem Max-Flow/Min-Cut Theorem noch einmal den Satz von König (siehe Satz 8.5 in Abschnitt 3.8):

- *Für einen bipartiten Graphen $G = (U \cup W, E)$ gilt*

$$\max\{|F| : F \subset E \text{ } \textit{Matching}\} = \min\{|D| : D \subset U \cup W \text{ } \textit{Träger}\}.$$

Als Beispiel betrachten wir erneut den bipartiten Graphen $(U \cup W, E)$ und den zugehörigen Digraphen $(\mathcal{N}, \mathcal{A})$ aus Abbildung 4.18 links, wobei wir rechts auch einen maximalen Fluss eingezeichnet hatten. Zur Bequemlichkeit des Lesers wiederholen wir diese Abbildung. Darunter in Abbildung 4.22 geben wir einen minimalen s,t-Schnitt

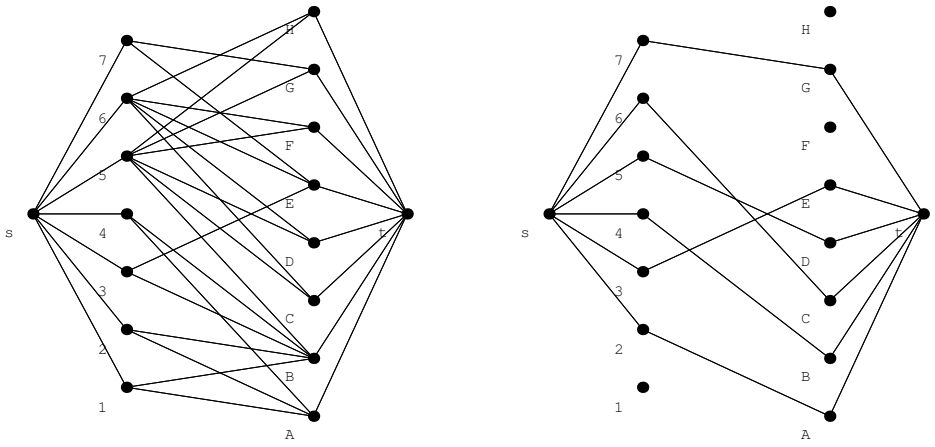


Abbildung 4.21: Matching und maximaler Fluss

in $(\mathcal{N}, \mathcal{A})$ an, wobei wir Knoten aus $\mathcal{N}_1^*$ durch $\circ$ und Knoten aus $\mathcal{N}_2^* = \mathcal{N}^* \setminus \mathcal{N}_1^*$ durch

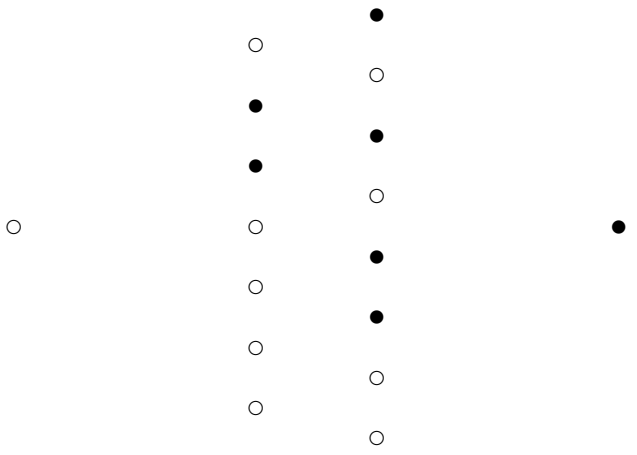


Abbildung 4.22: Ein minimaler Schnitt

$\bullet$ kennzeichnen. Durch die Ecken $\{5, 6, A, B, E, G\}$ hat man eine minimale Eckenüberdeckung gefunden. $\square$

4.4.2 Das Verfahren von Ford-Fulkerson

Gegeben sei wieder der Digraph $(\mathcal{N}, \mathcal{A})$ mit den beiden ausgezeichneten Knoten s und t, einer Quelle und einer Senke, weiter (nichtnegative, eventuell unendliche) Kapazitätsschranken u_{ij} auf den Pfeilen $(i,j) \in \mathcal{A}$. Sei $x \in \mathbb{R}^{|\mathcal{A}|}$ ein beliebiger zulässiger Fluss (am Anfang sei etwa $x := 0$) mit zugehörigem Wert v. Wir wollen eine Methode angeben, mit der ein zulässiger Fluss x^+ mit einem Wert $v^+ > v$ bestimmt werden kann (jedenfalls dann, wenn x nicht schon optimal ist).

Im folgenden unterscheiden wir zwischen einem *gerichteten* und einem *ungerichteten Pfad* im gegebenen Digraphen $(\mathcal{N}, \mathcal{A})$. Ein gerichteter Pfad benutzt nur aufeinanderfolgende Pfeile aus $\mathcal{A}$ (auch *Vorwärtspfeile* genannt). Dagegen kann ein ungerichteter Pfad (statt "ungerichteten Pfad" sagen wir auch einfach nur "Pfad") auch einen *Rückwärtspfeil* (i,j) benutzen, wobei $(j,i) \in \mathcal{A}$.

Definition 4.2 In dem Digraphen $(\mathcal{N}, \mathcal{A})$ mit einer Quelle s und einer Senke t sowie Kapazitäten u_{ij} auf den Pfeilen $(i,j) \in \mathcal{A}$ heißt ein (ungerichteter) Pfad P von s nach t ein *x-vermehrender Pfad* (x-augmenting path) bezüglich eines zulässigen Flusses $x = (x_{ij})_{(i,j)\in\mathcal{A}}$, wenn

(a) Ist $(i,j) \in P$ ein Vorwärtspfeil, also $(i,j) \in \mathcal{A}$, so ist $x_{ij} < u_{ij}$. Vorwärtspfeile im Pfad sind also nicht gesättigt (unsaturiert).

(b) Ist $(j,i) \in P$ ein Rückwärtspfeil, also $(i,j) \in \mathcal{A}$, so ist $x_{ij} > 0$. Rückwärtspfeile im Pfad tragen also einen positiven Fluss.

Mit anderen Worten ist ein Pfad von s nach t offenbar x-vermehrend, wenn Pfeile im Pfad mit Nullfluss Vorwärtspfeile, gesättigte (Kapazitätsschranken werden angenommen) Pfeile Rückwärtspfeile sind.

Beispiel: In Abbildung 4.23 geben wir einen kapazitierten Digraphen mit Quelle s und Senke t an. Längs der Pfeile sind die entsprechenden Kapazitäten aufgetragen. Die

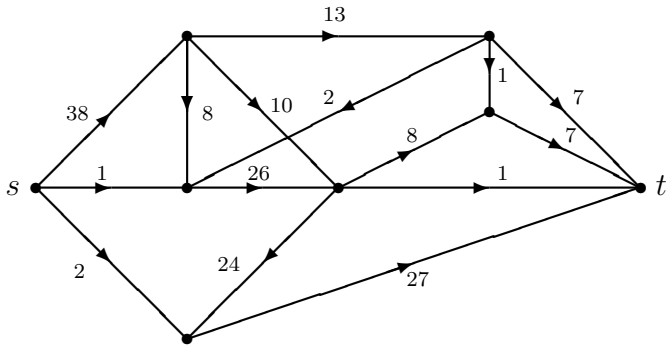


Abbildung 4.23: Ein kapazitierter Digraph

nächste Abbildung 4.24 zeigt einen zulässigen Fluss x von der Quelle zur Senke mit dem Wert $v = 28$. In der danach folgenden Abbildung 4.25 geben wir gestrichelt einen

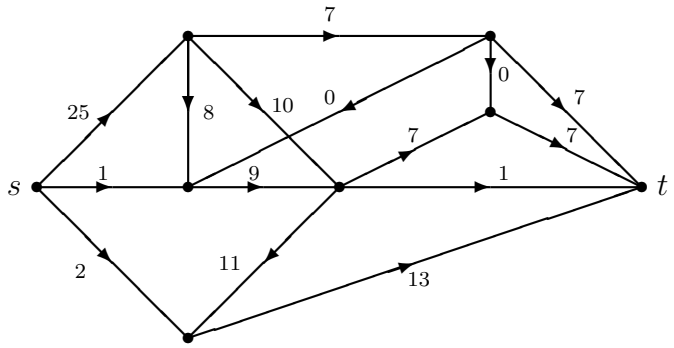


Abbildung 4.24: Ein zulässiger Fluss im kapazitierten Digraphen

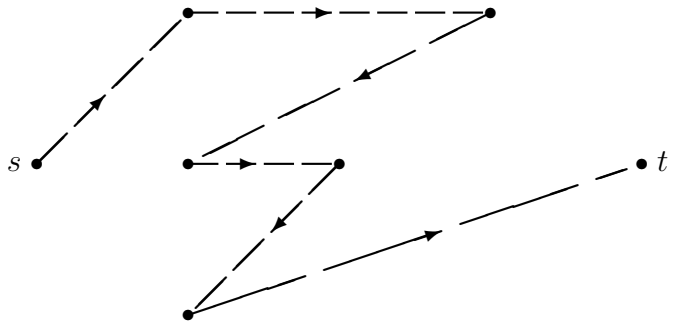


Abbildung 4.25: Ein x-vermehrender Pfad

x-vermehrenden Pfad an. Wenn man genau hinsieht, so erkennt man, dass dies sogar der einzige x-vermehrende Pfad ist. Er besteht nur aus Vorwärtspfeilen. □

Ist P ein x-vermehrender Pfad, so existiert ein $\epsilon > 0$ mit der folgenden Eigenschaft: Auf jedem Vorwärtspfeil in P kann x um ϵ vergrößert, auf jedem Rückwärtspfeil um ϵ vermindert werden, wobei die Flussbedingungen und die Kapazitätsschranken respektiert bleiben. Genauer definiere man

$$\epsilon_1 := \min\{u_{ij} - x_{ij} : (i,j) \in P \text{ ist Vorwärtspfeil}\}$$

und

$$\epsilon_2 := \min\{x_{ij} : (j,i) \in P \text{ ist Rückwärtspfeil}\},$$

anschließend

$$\epsilon := \min(\epsilon_1, \epsilon_2).$$

Man beachte, dass ϵ eine (positive) natürliche Zahl ist, wenn die Kapazitäten u_{ij}, $(i,j) \in \mathcal{A}$, und der Fluss x ganzzahlig sind. Nun definiere man $x^+ \in \mathbb{R}^{|\mathcal{A}|}$ durch

$$x_{ij}^+ := \begin{cases} x_{ij} + \epsilon, & (i,j) \in P \text{ ist Vorwärtspfeil}, \\ x_{ij} - \epsilon, & (j,i) \in P \text{ ist Rückwärtspfeil}, \\ x_{ij}, & \text{sonst}, \end{cases} \qquad (i,j) \in \mathcal{A}.$$

Offensichlich ist x^+ ein zulässiger Fluss, für den zugehörigen Wert v^+ gilt $v^+ \geq v + \epsilon$. Dieses Vorgehen kann fortgesetzt werden, solange es einen x-vermehrenden Fluss bezüglich des aktuellen Flusses x gibt. Das resultierende Verfahren ist der augmenting path algorithm von Ford-Fulkerson. Gerechtfertigt wird dieser durch den folgenden Satz, das sogenannte Augmenting Path Theorem.

Satz 4.3 *Ein zulässiger Fluss x ist genau dann ein maximaler Fluss, wenn es keinen x-vermehrenden Pfad gibt.*

Beweis: Durch die obige Konstruktion haben wir gezeigt: Gibt es einen x-vermehrenden Pfad, so ist x kein maximaler Fluss. Zu einem maxilen Fluss x gibt es also keinen x-vermehrenden Pfad. Nun nehmen wir umgekehrt an, dass x ein zulässiger Fluss ist und es keinen x-vermehrenden Pfad gibt. Es sei $\mathcal{N}_1$ die Menge aller Knoten j aus $\mathcal{N}$, für die es einen vermehrenden Pfad von der Quelle s nach j gibt (einschließlich s selber) und $\mathcal{N}_2 := \mathcal{N} \setminus \mathcal{N}_1$. Dann ist $(\mathcal{N}_1, \mathcal{N}_2)$ ein Schnitt im Digraphen, denn $\mathcal{N}_2$ und $\mathcal{N}_2$ sind nichtleer ($\mathcal{N}_1$ enthält die Quelle s, da es keinen vermehrenden Pfad von s nach t gibt ist t in $\mathcal{N}_2$ enthalten), disjunkt und ihre Vereinigung ist die Menge aller Knoten $\mathcal{N}$. Wir zeigen die Optimalität von x mit Hilfe der Aussage (d) im Max-Flow Min-Cut Theorem (Satz 4.1) von Ford-Fulkerson:

- *Ein zulässiger Fluss x^* und ein Schnitt $(\mathcal{N}_1^*, \mathcal{N}_2^*)$ sind genau dann optimal für das Maximaler-Fluss- bzw. das Minimaler-Schnitt-Problem, wenn*

$$x_{ij}^* = \begin{cases} 0 & \text{falls } (i,j) \in \mathcal{A} \text{ mit } i \in \mathcal{N}_2^* \text{ und } j \in \mathcal{N}_1^*, \\ u_{ij} & \text{falls } (i,j) \in \mathcal{A} \text{ mit } i \in \mathcal{N}_1^* \text{ und } j \in \mathcal{N}_2^*. \end{cases}$$

Wir benutzen, dass die eben angegebene Optimalitätsbedingung insbesondere hinreichend ist. Sei also $(i,j) \in \mathcal{A}$ ein Pfeil mit $i \in \mathcal{N}_2$ und $j \in \mathcal{N}_1$. Nach Definition von $\mathcal{N}_1$ bzw. $\mathcal{N}_2$ als Komplement von $\mathcal{N}_1$ in $\mathcal{N}$ gibt es einen vermehrenden Pfad von der Quelle s nach j, nicht aber nach i. Auf dem Rückwärtspfeil (j,i) verschwindet also der Fluss, es ist also $x_{ij} = 0$, denn andernfalls gäbe es einen vermehrenden Fluss von s nach i. Ist entsprechend $(i,j) \in \mathcal{A}$ ein Pfeil mit $i \in \mathcal{N}_1$ und $j \in \mathcal{N}_2$, so ist $x_{ij} = u_{ij}$ auf dem Vorwärtspfeil (i,j), denn andernfalls gäbe es einen vermehrenden Pfad von s nach j. Daher genügen x und der Schnitt $(\mathcal{N}_1, \mathcal{N}_2)$ der obigen (notwendigen und) hinreichenden Optimalitätsbedingung. Insbesondere ist x ein maximaler Fluss, der Satz ist bewiesen. □

Beispiel: Wir setzen obiges Beispiel fort. Im in Abbildung 4.25 eingetragenen x-vermehrenden Pfades (bezüglich des in Abbildung 4.25 angegebenen zulässigen Flusses x) sind alle Pfeile Vorwärtspfeile, ferner ist

$$\epsilon_1 = \min(13, 6, 2, 17, 13, 14) = 2, \qquad \epsilon_2 = +\infty.$$

Daher hat man $\epsilon = 2$ auf die Pfeile des Pfades zu addieren, man erhält den in Abbildung 4.26 angegebenen zulässigen Fluss, den wir wieder mit x bezeichnen. Sein Wert ist $v = 30$. Auch dieser verbesserte Fluss ist noch nicht optimal, denn man kann immer noch einen x-vermehrenden Pfad von der Quelle s zur Senke t finden. Dieser ist in

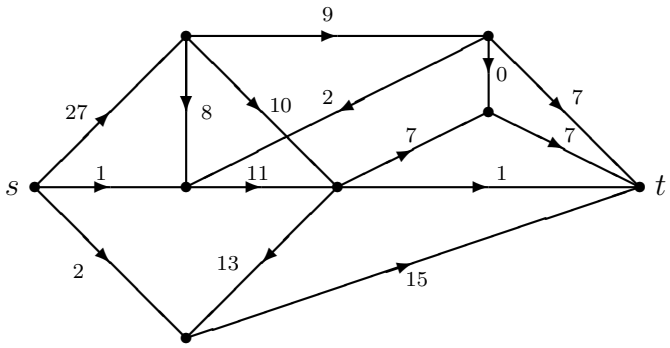


Abbildung 4.26: Ein verbesserter, zulässiger Fluss x

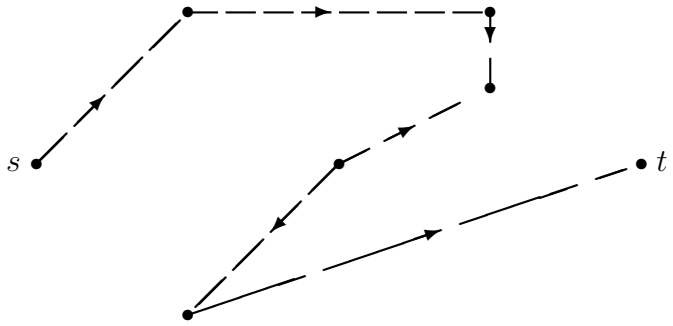


Abbildung 4.27: Ein x-vermehrender Pfad

Abbildung 4.27 gestrichelt eingetragen. In diesem Pfad gibt es einen Rückwärtspfeil. Man berechnet

$$\epsilon_1 = \min(11, 4, 1, 11, 12) = 1, \qquad \epsilon_2 = 7,$$

so dass $\epsilon = 1$ auf den Fluss der Vorwärtspfeile addiert und des Rückwärtspfeils subtrahiert wird. Man erhält schließlich den in Abbildung 4.28 angegebenen Fluss. Hier geben wir gleich auch noch eine Partition der Knoten an, die zu einem minimalen Schnitt gehört. Die drei mit $\circ$ gekennzeichneten Knoten gehören zur Knotenmenge $\mathcal{N}_1^*$, während die übrigen fünf Knoten, die zu $\mathcal{N}_2^*$ gehören, mit $\bullet$ bezeichnet sind. Offenbar ist der Wert des Flusses gleich 31, und das ist auch die Kapazität $C(\mathcal{N}_1^*, \mathcal{N}_2^*)$ des Schnittes $(\mathcal{N}_1^*, \mathcal{N}_2^*)$. □

Natürlich benötigt man man noch eine Methode, wie man einen x-vermehrenden Pfad zu einem zulässigen Fluss x bestimmen kann (oder feststellen kann, dass ein solcher nicht existiert, also x ein maximaler Fluss ist). Startet man mit dem trivialen Fluss $x = 0$ mit dem Wert 0, so brauchen keine Kapazitäten oder Flüsse untersucht zu werden. In einem zum trivialen Fluss gehörenden vermehrenden Pfad treten nur Vorwärtspfeile auf. Daher ist jeder gerichtete Pfad (also aufeinander folgende Vorwärtspfeile) im Digraphen $(\mathcal{N}, \mathcal{A})$ von der Quelle s zur Senke t ein vermehrender Pfad. Im nächsten Schritt können in einem vermehrenden Pfad auch Rückwärtspfeile auftreten (nämlich solche mit einem positiven Fluss), weiter kommen gesättigte Pfeile nicht als

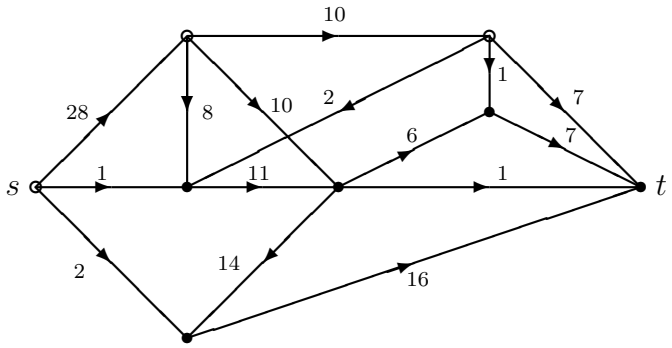


Abbildung 4.28: Ein optimaler Fluss

Vorwärtspfeile in Frage. Es wäre aber natürlich schön, wenn man die Aufgabe, einen vermehrenden Pfad zu finden, darauf zurückführen könnte, dass man in einem dem aktuellen Fluss x und dem Digraphen $(\mathcal{N}, \mathcal{A})$ zugeordneten Digraphen $(\mathcal{N}, \mathcal{A}(x))$ (es ändert sich also nur die Menge der Pfeile) einen gerichteten Pfad von der Quelle zur Senke findet. Hierbei sei

$$\mathcal{A}(x) := \{(i,j) \in \mathcal{N} \times \mathcal{N} : (i,j) \in \mathcal{A},\ x_{ij} < u_{ij}\} \cup \{(j,i) \in \mathcal{N} \times \mathcal{N} : (i,j) \in \mathcal{A},\ x_{ij} > 0\}.$$

Sei nun P ein gerichteter Pfad in $(\mathcal{N}, \mathcal{A}(x))$ von der Quelle s zur Senke t. Dann ist P offenbar ein x-vermehrender Pfad in $(\mathcal{N}, \mathcal{A})$, die gewünschte Zurückführung auf die Bestimmung eines gerichteten Pfades von s nach t in $(\mathcal{N}, \mathcal{A}(x))$ ist also gelungen.

Beispiel: Wir betrachten den in Abbildung 4.29 dargestellten kapazitierten Digraphen $(\mathcal{N}, \mathcal{A})$. Eingekreist sind die Kapazitäten u_{ij} auf den Pfeilen $(i,j) \in \mathcal{A}$, daneben ist

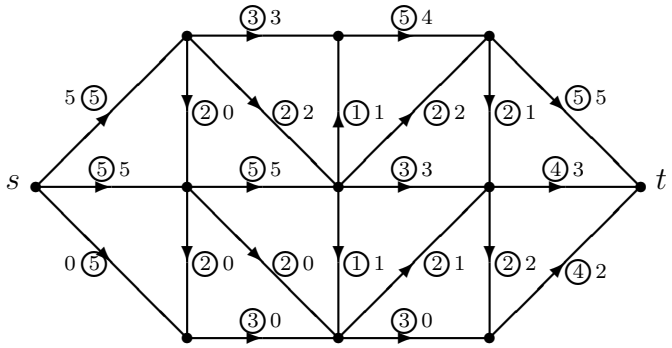


Abbildung 4.29: Ein kapazitierter Digraph

ein zulässiger Fluss x eingetragen. Der hieraus abgeleitete Digraph $(\mathcal{N}, \mathcal{A}(x))$ ist in Abbildung 4.30 angegeben. In $(\mathcal{N}, \mathcal{A}(x))$ gibt es gerichtete Pfade von s nach t, z. B. $(s, 4, 7, 9, t)$, $(s, 4, 7, 10, t)$, aber auch $(s, 4, 7, 10, 9, 8, t)$. □

Jetzt stellt sich die Frage: Wie bestimmt man in einem Digraphen $(\mathcal{N}, \mathcal{A})$ einen gerichteten Pfad von einem Knoten s zu einem anderen Knoten t? Wie bestimmt man einen

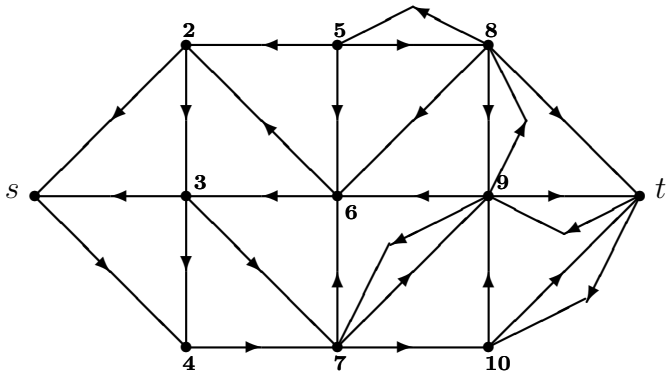


Abbildung 4.30: Abgeleiteter Digraph

gerichteten Pfad von s nach t mit einer minimalen Zahl von Pfeilen? Das Problem haben wir bei der Behandlung des Problems der kürzesten Wege schon gelöst. Wir setzen einfach für die Pfeillänge $c_{ij} := 1$, $(i,j) \in \mathcal{A}$, und wenden das Verfahren von Dijkstra an. Das Verfahren vereinfacht sich, da jetzt der endgültige Wert von d_j (Anzahl der Pfeile in einem kürzesten gerichteten Pfad von s nach j) der erste endliche Wert ist, der ihm zugeteilt wird (Beweis? Siehe W. J. Cook et al. (1998, S. 33)). Weiter kann man zeigen (dies ist ein Ergebnis von Edmonds-Karp (1972), siehe z. B. W. J. Cook et al. (1998, S. 44)), dass nach Addition von höchstens $|\mathcal{N}|\,|\mathcal{A}|$ *kürzesten* vermehrenden Pfaden der Augmenting Path Algorithm terminiert. Die Gesamtkomplexität dieses Verfahrens ist durch $O(|\mathcal{N}|\,|\mathcal{A}|^2)$ gegeben.

Beispiel: Wir wollen mit dem Verfahren von Dijkstra die kürzesten von s ausgehenden gerichteten Pfade in dem in Abbildung 4.30 angegebenen Digraphen bestimmen. Wir erhalten die folgenden Iterationen, wobei k die Iterationsstufe, V die aktuelle Kandidatenliste, d den Markenvektor und i den zu entfernenden Knoten bedeutet.

k	i	d	V
0		$(0,\infty,\infty,\infty,\infty,\infty,\infty,\infty,\infty,\infty,\infty)$	$\{s\}$
1	s	$(0,\infty,\infty,1,\infty,\infty,\infty,\infty,\infty,\infty,\infty)$	$\{4\}$
2	4	$(0,\infty,\infty,1,\infty,\infty,2,\infty,\infty,\infty,\infty)$	$\{7\}$
3	7	$(0,\infty,\infty,1,\infty,3,2,\infty,3,3,\infty)$	$\{6,9,10\}$
4	6	$(0,4,4,1,\infty,3,2,\infty,3,3,\infty)$	$\{2,3,9,10\}$
5	9	$(0,4,4,1,\infty,3,2,4,3,3,4)$	$\{2,3,8,10,t\}$
6	10	$(0,4,4,1,\infty,3,2,4,3,3,4)$	$\{2,3,8,t\}$
7	2	$(0,4,4,1,\infty,3,2,4,3,3,4)$	$\{3,8,t\}$
8	3	$(0,4,4,1,\infty,3,2,4,3,3,4)$	$\{8,t\}$
9	8	$(0,4,4,1,5,3,2,4,3,3,4)$	$\{5,t\}$
10	t	$(0,4,4,1,5,3,2,4,3,3,4)$	$\{5\}$
11	5	$(0,4,4,1,5,3,2,4,3,3,4)$	$\emptyset$

Man erhält den folgenden Shortest Path Spanning Tree, den wir in Abbildung 4.31 angeben. Hieraus liest man von der Quelle s ausgehende kürzeste Pfade zu den übrigen Knoten ab. □

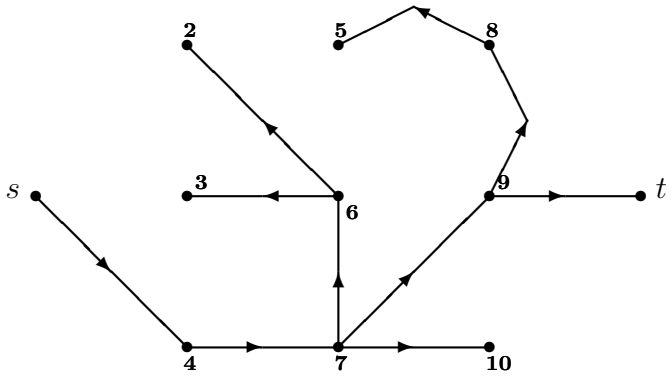


Abbildung 4.31: Abgeleiteter Digraph

Trivialerweise gilt die folgende Aussage (bei W. J. Cook at al. (1998, S. 43) als Satz formuliert):

- Die Kapazitätsschranken u_{ij}, $(i,j) \in \mathcal{A}$, seien (nichtnegativ und) ganzzahlig. Ist $K < \infty$ der Wert eines maximalen Flusses, so endet der augmenting path algorithm (gestartet mit dem trivialen Fluss $x = 0$) nach maximal K Durchläufen (in jedem Durchlauf wird zum aktuellen Fluss x ein geeignetes Vielfaches eines x-vermehrenden Pfades hinzuaddiert).

Eine entsprechende Aussage ist bei irrationalen Kapazitäten i. Allg. nicht richtig, wie ein Gegenbeispiel von Ford-Fulkerson zeigt (siehe Papadimitriou-Steiglitz, S. 126 ff.). In diesem Beispiel bricht das Verfahren nicht nur nicht ab, sondern erzeugt eine Folge von Flüssen, deren Wert gegen einen nichtoptimalen Wert konvergiert.

Beispiel: Man betrachte den kapazitierten Digraphen in Abbildung 4.32, die Kapazitäten sind angegeben. Der maximale Fluss ist offensichtlich 2000. Der Augmenting Path

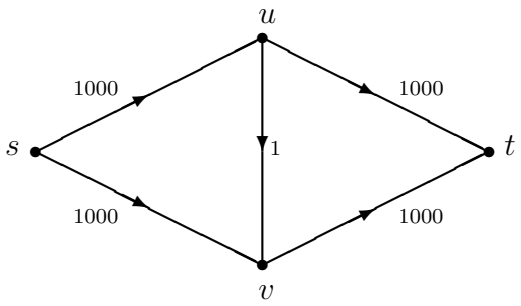


Abbildung 4.32: Ein Digraph mit 4 Knoten und 5 Pfeilen

Algorithm werde mit dem trivialen Fluss $x = 0$ als Startfluss angewandt. Im ersten Iterationsschritt ist (s,u,v,t) ein legaler vermehrender Pfad, durch den ein Zuwachs an Fluss um 1 erreicht wird. Im zweiten Schritt ist (s,v,u,t) ein vermehrender Pfad, der zu einem neuen Fluss mit dem Wert 2 führt. Daher können vermehrende Pfade zwischen (s,u,v,t) und (s,v,u,t) alternieren, der Fluss erhöht sich jedesmal um 1 und man benötigt 2000 Iterationen. Hier kann natürlich 1000 durch eine beliebige andere

große Zahl ersetzt werden. Wählt man die vermehrenden Pfade also ungeschickt, so kann die oben angebene Methode beliebig schlecht sein.

Die obige Wahl des vermehrenden Pfades ist ungünstig, da dieser länger als nötig ist. Hätten wir z. B. (s, v, t) gewählt, so hätten wir schon nach einer Iteration einen Fluss mit dem Wert 1000 erreicht. □

4.4.3 Aufgaben

1. Eine Gruppe von 11 Personen trifft sich in San Francisco. Möglichst viele von ihnen sollen nach New York geschickt werden. Es gibt keine Direktflüge, sondern es muss umgestiegen werden, wobei der Anschluss jeweils gesichert ist. In der folgenden Tabelle sind diese Flüge und die jeweils noch vorhandenen freien Sitze aufgelistet.

Von	Nach	Zahl freier Sitze
San Francisco	Denver	5
San Francisco	Houston	6
Denver	Atlanta	4
Denver	Chicago	2
Houston	Atlanta	5
Atlanta	New York	7
Chicago	New York	4

   (a) Man formuliere die Aufgabe als Maximalflussproblem in einem geeigneten Digraphen.

   (b) Man stelle die Knoten-Pfeil-Inzidenzmatrix auf und beweise, dass die gestutzte (letzte Zeile weggelassen) 5×7-Knoten-Pfeil-Inzidenzmatrix maximalen Rang hat. Welche 5×5-Untermatrizen sind nichtsingulär? Wie viele gibt es? Man veranschauliche sich diese im Digraphen!

   (c) Man rate einen maximalen Fluss und beweise seine Optimalität mit dem Max-Flow Min-Cut Theorem.

   (d) Ausgehend vom trivalen Fluss bestimme man jeweils zu einem aktuellen zulässigen Fluss x einen x-vermehrenden Pfad, addiere diesen auf den Fluss, bis man eine Lösung bestimmt hat.

2. In einem Maximaler-Fluss-Problem seien alle (endlichen) Kapazitäten rationale Zahlen. Man zeige: Besitzt das Problem eine Lösung, so besitzt es auch eine rationale Lösung.

3. Gegeben sei ein Digraph $(\mathcal{N}, \mathcal{A})$ mit einer Quelle s und einer Senke t sowie Kapazitäten u_{ij} auf den Pfeilen $(i, j) \in \mathcal{A}$. Sei x ein zulässiger Fluss. Man bilde den abgeleiteten kapazitierten Digraphen $(\mathcal{N}, \mathcal{A}(x))$, wobei die Pfeile $\mathcal{A}(x)$ und Kapazitäten $u(x)$ gegeben sind durch:

   - Ist $(i, j) \in \mathcal{A}$ und $x_{ij} < u_{ij}$, so ist $(i, j) \in \mathcal{A}(x)$ und $u(x)_{ij} := u_{ij} - x_{ij}$.
   - Ist $(i, j) \in \mathcal{A}$ und $x_{ij} > 0$, so ist $(j, i) \in \mathcal{A}(x)$ und $u(x)_{ji} := x_{ij}$.

   Man zeige: Ist v^* der Wert eines maximalen Flusses in $(\mathcal{N}, \mathcal{A})$ und v der Wert des zulässigen Flusses, so ist $v^* - v$ der Wert eines maximalen Flusses in dem abgeleiteten kapazitierten Digraphen $(\mathcal{N}, \mathcal{A}(x))$.

4. Man betrachte ein zulässiges Maximaler-Fluss-Problem im kapazitierten Digraphen $(\mathcal{N}, \mathcal{A})$. Man zeige: Werden die Kapazitätsschranken auf jedem Pfeil um $\alpha > 0$ vergrößert, so wird der maximale Fluss um höchstens $\alpha\,|\mathcal{A}|$ vergrößert.

4.5 Das Minimale-Kosten-Fluss-Problem

4.5.1 Formulierung des Problems, Dualität

Als Minimale-Kosten-Fluss-Problem hatten wir die folgende Aufgabe bezeichnet: Sei $(\mathcal{N}, \mathcal{A})$ ein gerichteter Graph mit der Knoten-Pfeil-Inzidenzmatrix $A \in \mathbb{R}^{|\mathcal{N}|\times|\mathcal{A}|}$, einem nichtnegativen Kapazitätsvektor $u \in \mathbb{R}^{|\mathcal{A}|}$ (Komponenten können auch gleich $+\infty$ sein), einem Kostenvektor $c \in \mathbb{R}^{|\mathcal{A}|}$ und einem Knotenvektor $b \in \mathbb{R}^{|\mathcal{N}|}$ (ein Knoten $k \in \mathcal{N}$ mit $b_k > 0$ ist ein *Angebotsknoten*, mit $b_k < 0$ ein *Bedarfsknoten*, mit $b_k = 0$ ein *Umladeknoten*). Zu lösen ist die Aufgabe

$$\text{(P)} \quad \begin{cases} \text{Minimiere} \displaystyle\sum_{(i,j)\in\mathcal{A}} c_{ij}x_{ij} \quad \text{unter den Nebendingungen} \\ 0 \le x_{ij} \le u_{ij} \quad ((i,j)\in\mathcal{A}), \qquad \displaystyle\sum_{j:(k,j)\in\mathcal{A}} x_{kj} - \sum_{i:(i,k)\in\mathcal{A}} x_{ik} = b_k \quad (k \in \mathcal{N}) \end{cases}$$

bzw.

$$\text{(P)} \qquad \text{Minimiere} \quad c^T x \quad \text{auf} \quad M := \{x \in \mathbb{R}^{|\mathcal{A}|} : 0 \le x \le u,\ Ax = b\}.$$

Für $k \in \mathcal{N}$ ist

$$(Ax)_k = \sum_{j:(k,j)\in\mathcal{A}} x_{kj} - \sum_{i:(i,k)\in\mathcal{A}} x_{ik}$$

der Fluss im Knoten k, d. h. die Differenz zwischen dem Fluss, der von k wegfliesst, und dem der in k ankommt. Denkbar wäre ein allgemeineres Problem der Form

$$\text{Minimiere} \quad c^T x \quad \text{unter den Nebenbedingungen} \quad l \le x \le u,\ a \le Ax \le b.$$

Hier sind also auf den Pfeilen untere und obere Kapazitätsschranken gegeben (wobei die untere Kapazitätsschranke auf gewissen Pfeilen auch gleich $-\infty$ und die obere Kapazitätsschranke auf gewissen Pfeilen auch gleich $+\infty$ sein kann), ferner der Fluss in allen Knoten nach unten und oben beschränkt. Ohne große Schwierigkeiten kann gezeigt werden, dass sich dieses scheinbar allgemeinere Problem auf eine Aufgabe der Form (P) zurückführen lässt (siehe W. J. Cook et al. (1998, S. 98)). Daher gehen wir im weiteren von der obigen Standardform (P) aus.

Zunächst wollen wir das zu (P) duale lineare Programm aufstellen, wobei wir die Restriktion $x \le u$ als $-x \ge -u$ schreiben. Das duale Problem ist

$$\text{(D)} \qquad \begin{cases} \text{Maximiere} \quad b^T y - u^T z \quad \text{auf} \\ N := \{(y,z) \in \mathbb{R}^{|\mathcal{N}|} \times \mathbb{R}^{|\mathcal{A}|} : z \ge 0,\ A^T y - z \le c\}. \end{cases}$$

Hierbei vereinbaren wir, dass $z_{ij} = 0$ für alle $(i,j) \in \mathcal{A}$ mit $u_{ij} = \infty$ (und $\infty \cdot 0 = 0$). Mit

$$\mathcal{A}' := \{(i,j) \in \mathcal{A} : u_{ij} < \infty\}$$

kann also das duale Problem in der Form

$$\text{(D)} \quad \begin{cases} \text{Maximiere} \quad \displaystyle\sum_{k \in \mathcal{N}} b_k y_k - \sum_{(i,j) \in \mathcal{A}'} u_{ij} z_{ij} \quad \text{unter den Nebenbedingungen} \\ \\ z_{ij} \geq 0 \quad ((i,j) \in \mathcal{A}'), \qquad \begin{array}{lll} y_i - y_j & \leq & c_{ij}, \quad ((i,j) \in \mathcal{A} \setminus \mathcal{A}'), \\ y_i - y_j - z_{ij} & \leq & c_{ij}, \quad ((i,j) \in \mathcal{A}') \end{array} \end{cases}$$

geschrieben werden. Für einen Vektor $y = (y_k) \in \mathbb{R}^{|\mathcal{N}|}$ setzen wir

$$\overline{c}_{ij} := c_{ij} - y_i + y_j, \qquad (i,j) \in \mathcal{A},$$

und nennen $\overline{c}_{ij}$ die *reduzierten Kosten* von (i,j). Für ein dual zulässiges Paar $(y,z) \in N$ ist daher $\overline{c}_{ij} \geq 0$, falls $u_{ij} = \infty$ bzw. $(i,j) \notin \mathcal{A}'$. Ist dagegen $u_{ij} < \infty$ bzw. $(i,j) \in \mathcal{A}'$, so muss z_{ij} sowohl $z_{ij} \geq 0$ als auch $z_{ij} \geq -\overline{c}_{ij}$ genügen, so dass die beste Wahl für z_{ij} gerade $\max(0, -\overline{c}_{ij})$ ist. In einem gewissen Sinne sind die Variablen z_{ij}, $(i,j) \in \mathcal{A}'$, daher überflüssig: Ein $y \in \mathbb{R}^{|\mathcal{N}|}$ nennen wir *zulässig* für (D), wenn die zu y gehörenden reduzierten Kosten $\overline{c}_{ij}$ für alle $(i,j) \notin \mathcal{A}'$ nichtnegativ sind. Durch

$$z_{ij} := \begin{cases} \max(0, -\overline{c}_{ij}), & (i,j) \in \mathcal{A}', \\ 0, & (i,j) \in \mathcal{A} \setminus \mathcal{A}' \end{cases}$$

erhält man dann nämlich ein dual zulässiges Paar (y,z). Entsprechend nennen wir $y^* \in \mathbb{R}^{|\mathcal{N}|}$ *optimal* für (D), wenn y^* zulässig für (D) ist und das Paar (y^*, z^*) mit

$$z^*_{ij} := \begin{cases} \max(0, -\overline{c}^*_{ij}), & (i,j) \in \mathcal{A}', \\ 0, & (i,j) \in \mathcal{A} \setminus \mathcal{A}' \end{cases}$$

(wobei c^*_{ij} die zu y^* gehörenden reduzierten Kosten sind) eine Lösung von (D) ist. Als Folgerung aus dem Satz von Kuhn-Tucker erhält man:

Satz 5.1 *Ein für das Minimale-Kosten-Fluss-Problem zulässiger Fluss x ist genau dann optimal, wenn es einen Vektor $y \in \mathbb{R}^{|\mathcal{N}|}$ gibt derart, dass mit den zugehörigen reduzierten Kosten $\overline{c}_{ij} := c_{ij} - y_i + y_j$ für alle $(i,j) \in \mathcal{A}$ gilt:*

$$\overline{c}_{ij} < 0 \Longrightarrow x_{ij} = u_{ij} \ (< \infty), \qquad \overline{c}_{ij} > 0 \Longrightarrow x_{ij} = 0.$$

Ist ferner x eine Lösung von (P) und y (im obigen Sinne) optimal für (D), so ist diese Bedingung erfüllt.

Gegeben sei ein nicht notwendig gerichteter Pfad P in $(\mathcal{N}, \mathcal{A})$, also eine Folge

$$(i_1, i_2), (i_2, i_3), \ldots, (i_{k-1}, i_k),$$

wobei $(i_j, i_{j+1}) \in \mathcal{A}$ oder $(i_{j+1}, i_j) \in \mathcal{A}$, $j = 1, \ldots, k-1$. Im ersten Fall spricht man (siehe oben) von einem Vorwärtspfeil, im zweiten Fall von einem Rückwärtspfeil. Mit

P^+ bezeichnen wir die Menge der Vorwärtspfeile in P, entsprechend mit P^- die Menge der Rückwärtspfeile in P. Als Kosten des Pfades P definieren wir

$$c(P) := \sum_{(i,j)\in P^+} c_{ij} - \sum_{(j,i)\in P^-} c_{ij}.$$

In Definition 4.2 haben wir festgelegt, wann wir einen Pfad P *vermehrend* bezüglich eines zulässigen Flusses x nennen. Dies war dann der Fall, wenn durch x die Kapazitäten auf den Vorwärtspfeilen des Pfades nicht ausgeschöpft werden und der Fluss positiv auf den Rückwärtspfeilen ist. Jetzt betrachten wir den Fall, dass P ein x-vermehrender Zyklus ist (also ein geschlossener x-vermehrender Pfad). Genau wie früher berechnen wir

$$\epsilon_1 := \min\{u_{ij} - x_{ij} : (i,j) \in P \text{ ist Vorwärtspfeil}\}$$

und

$$\epsilon_2 := \min\{x_{ij} : (j,i) \in P \text{ ist Rückwärtspfeil}\},$$

anschließend

$$\epsilon := \min(\epsilon_1, \epsilon_2)$$

und definieren $x^+ \in \mathbb{R}^{|\mathcal{A}|}$ durch

$$x^+_{ij} := \begin{cases} x_{ij} + \epsilon, & (i,j) \in P \text{ ist Vorwärtspfeil,} \\ x_{ij} - \epsilon, & (j,i) \in P \text{ ist Rückwärtspfeil,} \\ x_{ij}, & \text{sonst,} \end{cases} \qquad (i,j) \in \mathcal{A}.$$

Offensichtlich ist x^+ dann ein zulässiger Fluss mit den Kosten

$$c^T x^+ = c^T x + \epsilon c(P).$$

Daher erhalten wir:

- Ist x ein zulässiger Fluss und ist P ein x-vermehrender Zyklus mit negativen Kosten $c(P)$, so ist x^+ ein zulässiger Fluss mit kleineren Kosten als x.

Die Suche nach einem x-vermehrenden (ungerichteten) Zyklus kann man wieder auf die Suche nach einem gerichteten Zyklus in einem abgeleiteten Digraphen zurückführen. Bezüglich eines zulässigen Flusses x definieren wir die Pfeilmenge $\mathcal{A}(x)$ wie früher, definieren ferner noch Kosten $c(x)$ auf den Pfeilen aus $\mathcal{A}(x)$. Und zwar sei $(i,j) \in \mathcal{A}(x)$, wenn $(i,j) \in \mathcal{A}$ und $x_{ij} < u_{ij}$, die Kosten seien $c(x)_{ij} := c_{ij}$. Weiter sei $(j,i) \in \mathcal{A}(x)$, wenn $(i,j) \in \mathcal{A}$ und $x_{ij} > 0$, in diesem Fall seien die Kosten $c(x)_{ji} := -c_{ij}$. Jeder x-vermehrende Pfad in $(\mathcal{N}, \mathcal{A})$ entspricht dann einem gerichteten Pfad in $(\mathcal{N}, \mathcal{A}(x))$ mit den selben Kosten. Insbesondere entspricht ein x-vermehrender Zyklus mit negativen Kosten in $(\mathcal{N}, \mathcal{A})$ einem gerichteten Zyklus in $(\mathcal{N}, \mathcal{A}(x))$ mit negativen Kosten.

Beispiel: In Abbildung 4.33 geben wir $(\mathcal{N}, \mathcal{A})$ und $(\mathcal{N}, \mathcal{A}(x))$ an. Links steht $(\mathcal{N}, \mathcal{A})$, die drei Zahlen neben dem Pfeil $(i,j) \in \mathcal{A}$ sind c_{ij}, u_{ij}, x_{ij}. Die Kosten des Flusses sind 25. Auf der rechten Seite steht der Digraph $(\mathcal{N}, \mathcal{A}(x))$, eingetragen sind ferner auf den Pfeilen aus $\mathcal{A}(x)$ die entsprechenden Kosten. In $(\mathcal{N}, \mathcal{A}(x))$ gibt es den gerichteten Zyklus (p,b), (b,q), (q,p) mit den Kosten $-4+1+1=-2$. Einen verbesserten Fluss geben wir in Abbildung 4.34 an. Der verbesserte Fluss hat die Kosten 16. □

Der folgende Satz ist ein Analogon zu Satz 4.3, in dem ausgesagt wurde:

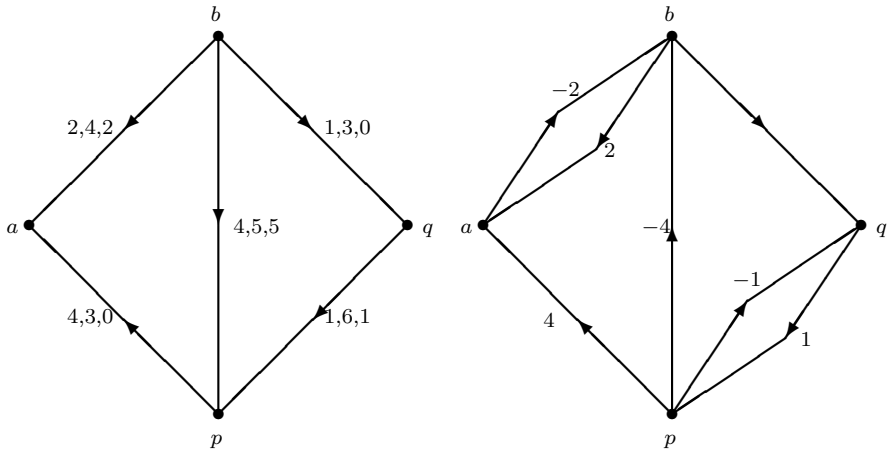


Abbildung 4.33: $(\mathcal{N}, \mathcal{A})$ und $(\mathcal{N}, \mathcal{A}(x))$

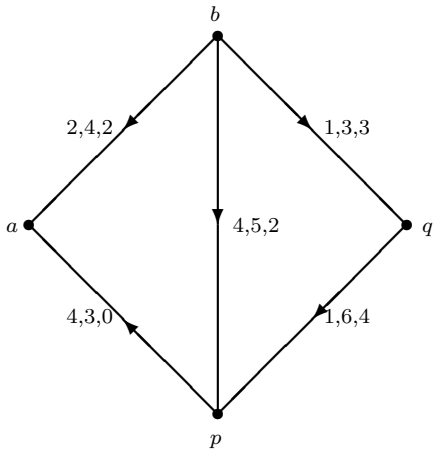


Abbildung 4.34: Ein verbesserter Fluss

- Ein zulässiger Fluss x ist genau dann ein maximaler Fluss, wenn es keinen x-vermehrenden Pfad gibt.

Satz 5.2 *Ein für das Minimale-Kosten-Fluss-Problem (P) zulässiger Fluss ist genau dann ein optimaler Fluss, wenn es keinen (ungerichteten) x-vermehrenden Zyklus mit negativen Kosten in $(\mathcal{N}, \mathcal{A})$ gibt.*

Beweis: Klar ist: Ist x ein optimaler Fluss, so gibt es keinen x-vermehrenden Zyklus mit negativen Kosten in $(\mathcal{N}, \mathcal{A})$. Daher nehmen wir nun umgekehrt an, dass es zu dem zulässigen Fluss x keinen x-vermehrenden Zyklus mit negativen Kosten in $(\mathcal{N}, \mathcal{A})$ gibt. Dann gibt es auch in $(\mathcal{N}, \mathcal{A}(x))$ keinen (gerichteten) Zyklus mit negativen Kosten. Zu $\mathcal{N}$ fügen wir einen weiteren Knoten r hinzu, ferner sei (r, j) für alle $j \in \mathcal{N}$ den Pfeilen $\mathcal{A}(x)$ hinzugefügt, die Kosten seien $c(x)_{rj} := 0$. Hierdurch erhalte man den neuen Digraphen $(\mathcal{N}', \mathcal{A}'(x))$. Das kürzester-Pfad-Problem in diesem neuen Digraphen ist lösbar (von r aus ist jeder Knoten j erreichbar und es gibt keinen Zyklus mit negativen Kosten), daher gibt es (Dualitätssatz) ein $y \in \mathbb{R}^{|\mathcal{N}|}$ mit $y_i - y_j \leq c(x)_{ij}$ für alle $(i, j) \in \mathcal{A}(x)$.

Nach Definition von $\mathcal{A}(x)$ ist also $y_i - y_j \leq c_{ij}$ für alle $(i,j) \in \mathcal{A}$ mit $x_{ij} < u_{ij}$, ferner $y_i - y_j \geq c_{ij}$ für alle $(i,j) \in \mathcal{A}$ mit $x_{ij} > 0$. Mit den zu y gehörenden reduzierten Kosten $\overline{c}_{ij} := c_{ij} - y_i + y_j$ ist also $\overline{c}_{ij} \geq 0$ für alle $(i,j) \in \mathcal{A}$ mit $x_{ij} < u_{ij}$ und $\overline{c}_{ij} \leq 0$ für alle $(i,j) \in \mathcal{A}$ mit $x_{ij} > 0$. Ist also $\overline{c}_{ij} < 0$, so ist $x_{ij} = u_{ij}$, während aus $\overline{c}_{ij} > 0$ folgt, dass $x_{ij} = 0$. Aus Satz 5.1 folgt, dass der zulässige Fluss x ein optimaler Fluss ist. Der Satz ist damit bewiesen. □

Bemerkung: Eine lineare (Minimierungs-) Optimierungsaufgabe ist lösbar, wenn sie zulässig ist und die Zielfunktion auf der Menge der zulässigen Lösungen nach unten beschränkt ist. Sind die Kosten nichtnegativ und existiert ein zulässiger Fluss, so ist das Minimale-Kosten-Fluss-Problem also lösbar. Diese Aussage wird in dem folgenden Satz noch wesentlich verallgemeinert. □

Satz 5.3 *Das gegebene Minimale-Kosten-Fluss-Problem sei zulässig. Dann besitzt das Problem genau dann eine Lösung, wenn es in $(\mathcal{N}, \mathcal{A})$ keinen gerichteten Zyklus mit negativen Kosten gibt derart, dass die Kapazitäten auf allen Pfeilen des Zyklus ∞ sind.*

Beweis: Siehe Aufgabe 2. □

4.5.2 Primale Minimale-Kosten-Fluss-Algorithmen

Wegen Satz 5.2 ist ein zulässiger Fluss eines Minimale-Kosten-Fluss-Problems genau dann optimal, wenn es keinen x-vermehrenden (ungerichteten) Zyklus mit negativen Kosten im Zyklus gibt. Ist weiter x nicht optimal, so erhält man durch einen x-vermehrenden Zyklus auf einfache Weise einen verbesserten zulässigen Fluss. Diese Idee ist Grundlage der primalen Minimale-Kosten-Fluss-Algorithmen. Um einen Eindruck zu bekommen, wieviele Iterationen bei diesem naiven Zugang möglich sind, fasse man das Maximaler-Fluss-Problem als einen Spezialfall des Minimale-Kosten-Fluss-Problems auf und betrachte das Maximale-Fluss-Problem aus Abbildung 4.32. Man füge einen Pfeil (t,s) mit Kosten $c_{ts} := -1$ und Kapazität $u_{ts} := \infty$ ein, gebe ferner den schon vorhandenen Pfeilen die Kosten 0.

Nun gehen wir auf das (primale) Netzwerk-Simplex-Verfahren zur Lösung des Minimale-Kosten-Fluss-Problems ein. Als lineares Programm geschrieben lautet dieses:

$$\text{(P)} \qquad \text{Minimiere} \quad c^T x \quad \text{auf} \quad M := \{x \in \mathbb{R}^{|\mathcal{A}|} : 0 \leq x \leq u,\ Ax = b\}.$$

Hierbei bedeute A die *gestutzte* Knoten-Pfeil-Inzidenzmatrix zum gerichteten Graphen $(\mathcal{N}, \mathcal{A})$, welchen wir stets als schwach zusammenhängend vorausgesetzt. Wir haben hier ein lineares Programm, das sich nur durch Box-Constraints von einem linearen Programm in Normalform unterscheidet. Die Adaption des Simplexverfahrens auf Probleme dieser Art haben wir in Unterabschnitt 2.2.5 untersucht, siehe Satz 2.11. Um das Blättern zu ersparen, geben wir diesen Satz hier noch einmal an.

Satz 5.4 *Gegeben sei das lineare Programm*

$$\text{(P)} \qquad \textit{Minimiere} \quad c^T x \quad \textit{auf} \quad M := \{x \in \mathbb{R}^n : l \leq x \leq u,\ Ax = b\}.$$

Sei $x \in M$ eine zulässige Basislösung mit Basisindexmenge B und Nichtbasisindizes $N := \{1, \ldots, n\} \setminus B$. Sei weiter $y := A_B^{-T} c_B$ und $\overline{c}_N := c_N - A_N^T y$. Dann gilt:

(a) Ist ($\overline{c}_j \geq 0$ *oder* $x_j = u_j$) *und* ($\overline{c}_j \leq 0$ *oder* $x_j = l_j$) *für alle* $j \in N$ *bzw. gilt*

$$\overline{c}_j < 0 \Longrightarrow x_j = u_j, \qquad \overline{c}_j > 0 \Longrightarrow x_j = l_j$$

für alle $j \in N$, *so ist* x *eine Lösung von (P).*

(b) Sei $s \in N$ *ein Nichtbasisindex mit* ($\overline{c}_s < 0$ *und* $x_s < u_s$) *oder* ($\overline{c}_s > 0$ *und* $x_s > l_s$)[15]. *Mit* $w := A_B^{-1} a_s$ *definiere man*

$$x_s(t) := \begin{cases} x_s - t & \text{für} \quad \overline{c}_s > 0, \\ x_s + t & \text{für} \quad \overline{c}_s < 0, \end{cases} \qquad x_B(t) := \begin{cases} x_B + tw & \text{für} \quad \overline{c}_s > 0, \\ x_B - tw & \text{für} \quad \overline{c}_s < 0 \end{cases}$$

und schließlich $x_j(t) := x_j$ *für alle* $j \in N \setminus \{s\}$. *Dann ist* $Ax(t) = b$ *und* $c^T x(t) = c^T x - t\,|\overline{c}_s|$ *für alle* t.

(c) Sei

$$t^* := \sup\{t \geq 0 : l_s \leq x_s(t) \leq u_s,\ l_B \leq x_B(t) \leq u_B\}.$$

Ist $t^* = +\infty$, *so ist* $\inf(\mathrm{P}) = -\infty$, *also (P) nicht lösbar.*

Sei $x \in M$ eine zulässige Basislösung, d. h. es existiere eine $(|\mathcal{N}|-1)$-elementige Menge $\mathcal{B} \subset \mathcal{A}$ von Basisindizes mit der Eigenschaft, dass die zu $\mathcal{B}$ gehörenden Spalten von A linear unabhängig sind (bzw. $(\mathcal{N}, \mathcal{B})$ ein $(\mathcal{N}, \mathcal{A})$ aufspannender Baum ist) bzw. die hieraus gebildete Matrix $A_B \in \mathbb{R}^{(|\mathcal{N}-1)\times(|\mathcal{N}|-1)}$ nichtsingulär ist und für alle Nichtbasisindizes $(i,j) \in \mathcal{A} \setminus \mathcal{B}$ gilt $x_{ij} = 0$ oder $x_{ij} = u_{ij}$ bzw.

$$x_{ij} = \begin{cases} 0 \text{ oder } u_{ij} & \text{für} \quad u_{ij} < \infty, \\ 0 & \text{für} \quad u_{ij} = \infty, \end{cases} \qquad (i,j) \in \mathcal{A} \setminus \mathcal{B}.$$

Die aus den Spalten von A mit einem Nichtbasisindex aus $\mathcal{A} \setminus \mathcal{B}$ (wir vermeiden aus begreiflichen Gründen die Bezeichnung $\mathcal{N}$ hierfür) gebildete Untermatrix werde mit A_N bezeichnet. Die Bezeichnungen c_B, c_N usw. sind in naheliegender Weise zu verstehen. Wir berechnen

$$y := A_B^{-T} c_B.$$

Hierzu ist das lineare Gleichungssystem $A_B^T y = c_B$ zu lösen. Ist $(i,j) \in \mathcal{B}$, so hat A_B in der Spalte (i,j) in der i-ten Komponente eine $+1$, in der j-ten eine -1 und sonst nur Nullen. Das Gleichungsystem $A_B^T y = c_B$ schreibt sich also komponentenweise in der Form

$$y_i - y_j = c_{ij}, \qquad (i,j) \in \mathcal{B},$$

wobei $y_{|\mathcal{N}|} := 0$ zu setzen ist, da die letzte Gleichung weggelassen wurde. Anschließend sind die reduzierten Kosten

$$\overline{c}_N := c_N - A_N^T y$$

oder, komponentenweise geschrieben,

$$\overline{c}_{ij} := c_{ij} - y_i + y_j, \qquad (i,j) \in \mathcal{A} \setminus \mathcal{B},$$

[15] Sind alle Variablen beschränkt, was z. B. für $l := 0$ der Fall ist, so ist ($\overline{c}_s < 0$ und $x_s = 0$) oder ($\overline{c}_s > 0$ und $x_s = u_s$).

zu berechnen. Nach Satz 2.11 ist es ziemlich klar, wie es weitergeht. Man bestimme nämlich die Indexmengen

$$\mathcal{N}_1 := \{(i,j) \in \mathcal{A} \setminus \mathcal{B} : x_{ij} = 0,\ \bar{c}_{ij} < 0\}, \quad \mathcal{N}_2 := \{(i,j) \in \mathcal{A} \setminus \mathcal{B} : x_{ij} = u_{ij},\ \bar{c}_{ij} > 0\}.$$

Ist $\mathcal{N}_1 \cup \mathcal{N}_2 = \emptyset$, sind also $\mathcal{N}_1$ und $\mathcal{N}_2$ leer, so ist die aktuelle Basislösung x optimal. Dies folgt einfach aus Satz 2.11 (a). Siehe auch die Optimalitätsbedingung in Satz 5.1, denn $\mathcal{N}_1 = \mathcal{N}_2 = \emptyset$ impliziert offenbar, dass

$$\bar{c}_{ij} < 0 \Longrightarrow x_{ij} = u_{ij}\ (< \infty), \qquad \bar{c}_{ij} > 0 \Longrightarrow x_{ij} = 0$$

für alle $(i,j) \in \mathcal{A} \setminus \mathcal{B}$ (da $\bar{c}_{ij} = 0$ für alle $(i,j) \in \mathcal{B}$, gilt dies trivialerweise für alle $(i,j) \in \mathcal{A}$). Andernfalls wähle man $(i_s, j_s) \in \mathcal{N}_1 \cup \mathcal{N}_2$. Nach dem Simplexverfahren (siehe Satz 2.11 (b)) hat man jetzt $w := A_B^{-1} a^{i_s j_s}$ zu berechnen, wobei $a^{i_s j_s}$ die Spalte von A ist, die zum Pfeil $(i_s, j_s) \in \mathcal{A} \setminus \mathcal{B}$ gehört. Nach Definition der Knoten-Pfeil-Inzidenzmatrix hat der Vektor $a^{i_s j_s}$ als Eintrag in der i_s-ten Komponente eine $+1$, in der j_s-ten Komponente eine -1 und sonst nur Nullen. Mit anderen Worten ist

$$a^{i_s j_s} = e_{i_s} - e_{j_s},$$

wobei $e_{|\mathcal{N}|} := 0$ zu setzen ist, da die Matrix A eine gestutzte Knoten-Pfeil-Inzidenzmatrix ist. Die Berechnung von w kann folgendermaßen erreicht werden: Sei P der (eindeutige) ungerichtete Pfad in dem aktuellen (ungerichteten) aufspannenden Baum $(\mathcal{N}, \mathcal{B})$, der vom Knoten j_s zum Knoten i_s führt. Anschließend definiere man $\lambda = (\lambda_{ij})_{(i,j)\in\mathcal{B}}$ durch

$$\lambda_{ij} := \begin{cases} +1, & \text{falls } (i,j) \text{ Vorwärtspfeil in } P, \\ -1, & \text{falls } (i,j) \text{ Rückwärtspfeil in } P, \\ 0, & \text{sonst} \end{cases} \qquad ((i,j) \in \mathcal{B}).$$

Wir wollen uns überlegen, dass dann

$$(*) \qquad \sum_{(i,j)\in\mathcal{B}} \lambda_{ij} a^{ij} = e_{j_s} - e_{i_s},$$

wobei e_i natürlich wieder den i-ten Einheitsvektor im $\mathbb{R}^{|\mathcal{N}|-1}$ bedeutet und $e_{|\mathcal{N}|} := 0$ gesetzt wird. Die Beziehung $(*)$ zeigen wir komponentenweise, wobei wir ausnutzen, dass

$$e_k^T \Big(\sum_{(i,j)\in\mathcal{B}} \lambda_{ij} a^{ij} \Big) = \sum_{(i,j)\in\mathcal{B}} \lambda_{ij} e_k^T a^{ij} = \sum_{j:(k,j)\in\mathcal{B}} \lambda_{kj} - \sum_{i:(i,k)\in\mathcal{B}} \lambda_{ik}.$$

Für die Komponenten k, für die der Knoten k nicht im Pfad P vorkommt, ist

$$e_k^T \Big(\sum_{(i,j)\in\mathcal{B}} \lambda_{ij} a^{ij} \Big) = \sum_{j:(k,j)\in\mathcal{B}} \lambda_{kj} - \sum_{i:(i,k)\in\mathcal{B}} \lambda_{ik} = 0 = e_k^T (e_{j_s} - e_{i_s}).$$

Nun nehmen wir an, der Knoten k komme im Pfad P vor. Zunächst sei k kein Endpunkt des Pfades, es sei also $k \notin \{i_s, j_s\}$. Dann hat k zwei Nachbarn p und q im Pfad P. Die Knotenfolge im Pfad P ist also $j_s, \ldots, p, k, q, \ldots, i_s$. Insgesamt vier Fälle sind möglich:

1. $(p,k),(q,k) \in \mathcal{B}$.

   Dann ist (p,k) ein Vorwärts- und (q,k) ein Rückwärtspfeil und daher

$$e_k^T\Big(\sum_{(i,j)\in\mathcal{B}} \lambda_{ij}a^{ij}\Big) = \sum_{j:(k,j)\in\mathcal{B}} \lambda_{kj} - \sum_{i:(i,k)\in\mathcal{B}} \lambda_{ik} = -(\underbrace{\lambda_{pk}}_{=1} + \underbrace{\lambda_{qk}}_{=-1}) = 0 = e_k^T(e_{j_s} - e_{i_s}).$$

2. $(k,p),(k,q) \in \mathcal{B}$.

   Dann ist (k,p) ein Rückwärts- und (k,q) ein Vorwärtspfeil, folglich

$$e_k^T\Big(\sum_{(i,j)\in\mathcal{B}} \lambda_{ij}a^{ij}\Big) = \sum_{j:(k,j)\in\mathcal{B}} \lambda_{kj} - \sum_{i:(i,k)\in\mathcal{B}} \lambda_{ik} = (\underbrace{\lambda_{kp}}_{=-1} + \underbrace{\lambda_{kq}}_{=1}) = 0 = e_k^T(e_{j_s} - e_{i_s}).$$

3. $(p,k),(k,q) \in \mathcal{B}$.

   Dann sind (p,k) und (k,q) Vorwärtspfeile und daher

$$e_k^T\Big(\sum_{(i,j)\in\mathcal{B}} \lambda_{ij}a^{ij}\Big) = \sum_{j:(k,j)\in\mathcal{B}} \lambda_{kj} - \sum_{i:(i,k)\in\mathcal{B}} \lambda_{ik} = \underbrace{\lambda_{kq}}_{=1} - \underbrace{\lambda_{pk}}_{=1} = 0 = e_k^T(e_{j_s} - e_{i_s}).$$

4. $(k,p),(q,k) \in \mathcal{B}$.

   Dann sind (k,p) und (q,k) Rückwärtspfeile und daher

$$e_k^T\Big(\sum_{(i,j)\in\mathcal{B}} \lambda_{ij}a^{ij}\Big) = \sum_{j:(k,j)\in\mathcal{B}} \lambda_{kj} - \sum_{i:(i,k)\in\mathcal{B}} \lambda_{ik} = \underbrace{\lambda_{kp}}_{=-1} - \underbrace{\lambda_{qk}}_{=-1} = 0 = e_k^T(e_{j_s} - e_{i_s}).$$

Es bleibt der Fall, dass $k = j_s$ oder $k = i_s$. Sei etwa $k = j_s$ und q der nächste Knoten im Pfad P. Diesmal sind zwei Fälle zu unterscheiden, nämlich ob $(k,q) \in \mathcal{B}$ (also (k,q) ein Vorwärtspfeil) oder $(q,k) \in \mathcal{B}$ (also (q,k) ein Rückwärtspfeil). Im ersten Fall ist

$$e_k^T\Big(\sum_{(i,j)\in\mathcal{B}} \lambda_{ij}a^{ij}\Big) = \sum_{j:(k,j)\in\mathcal{B}} \lambda_{kj} - \sum_{i:(i,k)\in\mathcal{B}} \lambda_{ik} = \underbrace{\lambda_{kq}}_{=1} = 1 = e_k^T(e_{j_s} - e_{i_s}).$$

Im zweiten Fall ist

$$e_k^T\Big(\sum_{(i,j)\in\mathcal{B}} \lambda_{ij}a^{ij}\Big) = \sum_{j:(k,j)\in\mathcal{B}} \lambda_{kj} - \sum_{i:(i,k)\in\mathcal{B}} \lambda_{ik} = -\underbrace{\lambda_{qk}}_{=-1} = 1 = e_k^T(e_{j_s} - e_{i_s}).$$

Entsprechendes zeigt man für $k = i_s$, so dass insgesamt $(*)$ bewiesen ist. Also ist

$$A_B w = a^{i_s j_s} = e_{i_s} - e_{j_s} = -A_B\lambda,$$

folglich $w = -\lambda$ bzw.

$$w_{ij} := \begin{cases} -1, & \text{falls } (i,j) \text{ Vorwärtspfeil in } P, \\ 1, & \text{falls } (i,j) \text{ Rückwärtspfeil in } P, \\ 0, & \text{sonst} \end{cases} \qquad ((i,j) \in \mathcal{B}).$$

Wir nehmen zunächst an, es sei $(i_s, j_s) \in \mathcal{N}_1$, also $\overline{c}_{i_s j_s} < 0$ und $x_{i_s j_s} = 0$. Nach Satz 2.11 definiere man $x(t) \in \mathbb{R}^n$ durch

$$x_{i_s j_s}(t) := t$$

sowie

$$x_{ij}(t) := \begin{cases} x_{ij} + t, & \text{falls } (i,j) \text{ Vorwärtspfeil in } P, \\ x_{ij} - t, & \text{falls } (i,j) \text{ Rückwärtspfeil in } P, \\ x_{ij}, & \text{sonst} \end{cases} \qquad ((i,j) \in \mathcal{B})$$

und

$$x_{ij}(t) := x_{ij} \quad ((i,j) \in (\mathcal{A} \setminus \mathcal{B}) \setminus \{(i_s, j_s)\}).$$

Dann ist (siehe Beweis von Satz 2.11)

$$Ax(t) = b, \qquad c^T x(t) = c^T x + t\overline{c}_{i_s j_s} \qquad \text{für alle } t \in \mathbb{R}.$$

Aus diesem Grunde wird man versuchen, $t^* \geq 0$ möglichst groß zu wählen, wobei aber die Zulässigkeit von $x(t^*)$ gesichert sein muss. Wegen $x_{i_s j_s}(t) = t$ muss $t^* \leq u_{i_s j_s}$ gelten. Weiter hat man offenbar die Bedingungen

$$t^* \leq \min_{\substack{(i,j) \in \mathcal{B} \\ w_{ij} = -1}} (u_{ij} - x_{ij}), \qquad t^* \leq \min_{\substack{(i,j) \in \mathcal{B} \\ w_{ij} = 1}} x_{ij},$$

insgesamt kann

$$t^* := \min\Big(\min_{\substack{(i,j) \in \mathcal{B} \\ w_{ij} = -1}} (u_{ij} - x_{ij}), \min_{\substack{(i,j) \in \mathcal{B} \\ w_{ij} = 1}} x_{ij}, u_{i_s j_s} \Big)$$

gesetzt werden. Ist $t^* = \infty$, so besitzt das gegebene Minimale-Kosten-Fluss-Problem keine Lösung, da die Zielfunktion auf der Menge der zulässigen Lösungen nicht nach unten beschränkt ist.

Die Unterschiede für den Fall, dass $(i_s, j_s) \in \mathcal{N}_2$ bzw. $\overline{c}_{i_s j_s} > 0$ und $x_{i s j_s} = u_{i_s j_s}$ ist, sind marginal. Wegen Satz 2.11 definiere man hier $x(t) \in \mathbb{R}^n$ durch

$$x_{i_s j_s}(t) := u_{i_s j_s} - t,$$

sowie

$$x_{ij}(t) := \begin{cases} x_{ij} - t, & \text{falls } (i,j) \text{ Vorwärtspfeil in } P, \\ x_{ij} + t, & \text{falls } (i,j) \text{ Rückwärtspfeil in } P, \\ x_{ij}, & \text{sonst} \end{cases} \qquad ((i,j) \in \mathcal{B})$$

und

$$x_{ij}(t) := x_{ij} \quad ((i,j) \in (\mathcal{A} \setminus \mathcal{B}) \setminus \{(i_s, j_s)\}).$$

Dann ist wie oben

$$Ax(t) = b, \qquad c^T x(t) = c^T x - t\overline{c}_{i_s j_s} \qquad \text{für alle } t \in \mathbb{R}.$$

Wieder wird man $t^* \geq 0$ unter der Restriktion, dass $x(t^*)$ zulässig ist, möglichst groß wählen. Dies führt auf

$$t^* := \min\Big(\min_{\substack{(i,j)\in\mathcal{B}\\ w_{ij}=-1}} x_{ij},\ \min_{\substack{(i,j)\in\mathcal{B}\\ w_{ij}=1}} (u_{ij}-x_{ij}), u_{i_s j_s}\Big).$$

Man erkennt, dass man beide Fälle dadurch zusammensetzen kann, dass man

$$\sigma := \begin{cases} 1, & \text{falls} \quad (i_s,j_s)\in\mathcal{N}_1,\\ -1, & \text{falls} \quad (i_s,j_s)\in\mathcal{N}_2 \end{cases}$$

definiert, ferner

$$\delta_1 := \min_{\substack{(i,j)\in\mathcal{B}\\ w_{ij}=\sigma}} x_{ij}, \qquad \delta_2 := \min_{\substack{(i,j)\in\mathcal{B}\\ w_{ij}=-\sigma}} (u_{ij}-x_{ij})$$

und anschließend

$$t^* := \min(\delta_1,\delta_2,u_{i_s j_s}), \qquad x^+ := x(t^*).$$

Ist $t^* = \infty$, so ist das Problem nicht lösbar.

Falls $t^* = u_{i_s j_s}$, so setze man $\mathcal{B}^+ := \mathcal{B}$, durch $(x^+,\mathcal{B}^+)$ hat man eine neue zulässige Basislösung gefunden. Denn $(\mathcal{N},\mathcal{B}^+) = (\mathcal{N},\mathcal{B})$ ist nachwievor ein $(\mathcal{N},\mathcal{A})$ aufspannender Baum, ferner ist $x^+_{ij} \in \{0,u_{ij}$ für alle $(i,j)\in\mathcal{A}\setminus\mathcal{B}^+ = \mathcal{A}\setminus\mathcal{B}$, da $x^+_{ij} = x_{ij} \in \{0,u_{ij}\}$ für alle $(i,j)\in(\mathcal{A}\setminus\mathcal{B}^+)\setminus\{(i_s,j_s)\}$ (da x Basislösung) und $x^+_{i_s j_s}\in\{0,u_{i_s j_s}\}$.

Ist dagegen $t^* < u_{i_s j_s}$, so bestimme man ein Paar $(i_r,j_r)\in\mathcal{B}$, für welches bei der Bestimmung von δ_1 oder δ_2 (je nach dem, welches kleiner ist) das Minimum angenommen wird. Insbesondere kommt dann $(i_r,j_r)\in\mathcal{B}$ als Vorwärts- oder Rückwärtspfeil in einem Pfad von j_s nach i_s vor. Die neue Indexmenge ist

$$\mathcal{B}^+ := (\mathcal{B}\setminus\{(i_r,j_r)\})\cup\{(i_s,j_s)\}.$$

Dann ist $x^+_{ij}\in\{0,u_{ij}\}$ für alle $(i,j)\in\mathcal{A}\setminus\mathcal{B}^+$. Um nachzuweisen, dass x^+ eine zulässige Basislösung für das gestellte Netzwerkflußproblem ist, muss daher nur noch nachgewiesen werden, dass die zu $\mathcal{B}^+$ gehörenden $|\mathcal{N}|-1$ Spalten der gestutzten Knoten-Pfeil-Inzidenzmatrix A linear unabhängig sind bzw. $(\mathcal{N},\mathcal{B}^+)$ ein $(\mathcal{N},\mathcal{A})$ (als ungerichtete Graphen betrachtet) aufspannender Baum ist. Wegen $|\mathcal{B}^+| = |\mathcal{N}|-1$ genügt der Nachweis dafür, dass $(\mathcal{N},\mathcal{B}^+)$ zusammenhängend ist, es also in $(\mathcal{N},\mathcal{B}^+)$ einen (ungerichteten) Pfad von $s\in\mathcal{N}$ nach $t\in\mathcal{N}$ gibt. Denn: $(\mathcal{N},\mathcal{B}\setminus\{(i_r,j_r)\})$ hat zwei Zusammenhangskomponenten. Wir bezeichnen mit $\mathcal{N}^-$ die Menge der Knoten $j\in\mathcal{N}$, für die der eindeutige (ungerichtete) Pfad von s nach j in $(\mathcal{N},\mathcal{B})$ den Pfeil $(i_r,j_r)\in\mathcal{B}$ (als Vorwärt- oder Rückwärtspfeil) *nicht* benutzt, mit $\mathcal{N}^+ := \mathcal{N}\setminus\mathcal{N}^-$ wird das Komplement bezeichnet. Natürlich liegen i_r und j_r in verschiedenen Zusammenhangskomponenten. Es sei etwa $i_r\in\mathcal{N}^-$ und $j_r\in\mathcal{N}^+$. Ist $t\in\mathcal{N}^-$, so liegen s und t in der selben Zusammenhangskomponente, so dass es von s nach t sogar einen Pfad in $(\mathcal{N},\mathcal{B}\setminus\{(i_r,j_r)\})$ gibt. Sei daher $t\in\mathcal{N}^+$. Zunächst sei $i_s\in\mathcal{N}^-$. Dann besteht ein Pfad von s nach t in $(\mathcal{N},\mathcal{B}^+)$ aus dem Pfad von s nach i_s in $(\mathcal{N},\mathcal{B}\setminus\{(i_r,j_r)\})$ zusammen mit (i_s,j_s) und dem Pfad von j_s nach t in $(\mathcal{N},\mathcal{B})$. Letzterer enthält nicht (i_r,j_r) als Vorwärts- oder Rückwärtspfeil, ist also sogar ein Pfad in $(\mathcal{N},\mathcal{B}\setminus\{(i_r,j_r)\})$, da $j_s\in\mathcal{N}^+$ (andernfalls wären i_s und j_s in der selben Zusammenhangskomponente wie i_r, aber einer anderen

als j_r, so dass ein Pfad von j_s nach i_s nicht (i_r, j_r) als Vorwärts- oder Rückwärtspfeil enthalten könnte). Ist $i_s \in \mathcal{N}^+$, so ist $j_s \in \mathcal{N}^-$, so dass ein Pfad von s nach t in $(\mathcal{N}, \mathcal{B}^+)$ aus dem Pfad von s nach j_s in $(\mathcal{N}, \mathcal{B} \setminus \{(i_r, j_r)\})$, dem Rückwärtspfeil (j_s, i_s) und dem Pfad von i_s nach t in $(\mathcal{N}, \mathcal{B} \setminus \{(i_r, j_r)\}$ besteht. Insgesamt ist gezeigt, dass $(\mathcal{N}, \mathcal{B}^+)$ zusammenhängend und damit x^+ eine Basislösung mit den Basispfeilen $\mathcal{B}^+$ ist.

Beispiel: Gegeben sei der in Abbildung 4.35 angegebene gerichtete Graph, wobei rechts

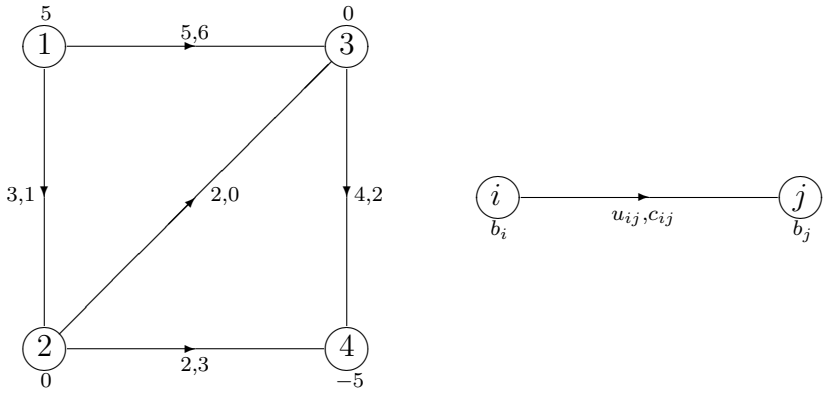


Abbildung 4.35: Ein Minimale-Kosten-Fluss-Problem

angegeben ist, welche Bedeutung die angegebenen Zahlen haben. Z. B. sind die Knoten 2 und 3 Umladeknoten, der Knoten 1 ein Angebots- und der Knoten 4 ein Bedarfsknoten. Man starte mit der zulässigen Basislösung

$$x = (x_{12}, x_{13}, x_{23}, x_{24}, x_{34})^T := (1, 4, 0, 1, 4)^T$$

und der zugehörigen Indexmenge

$$\mathcal{B} := \{(1, 2), (1, 3), (2, 4)\}.$$

Die zugehörigen Kosten sind $c_0 = 36$. Der zu dieser Startlösung gehörende aufspannende Baum ist in Abbildung 4.36 angegeben. Die Berechnung von $y := A_B^{-T} c_B$ führt auf

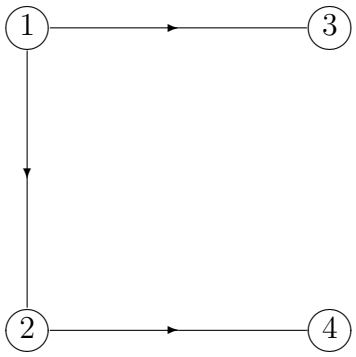


Abbildung 4.36: Aufspannender Baum zur Startlösung

das lineare Gleichungssystem

$$\begin{aligned} y_1 - y_2 &= 1 \\ y_1 - y_3 &= 6 \\ y_2 - y_4 &= 3, \end{aligned}$$

mit $y_4 = 0$ erhält man

$$y = (y_1, y_2, y_3)^T = (4, 3, -2).$$

Als reduzierte Kosten erhält man

$$\overline{c}_{23} = -5, \qquad \overline{c}_{34} = 4.$$

Die Nichtbasisindizes zerfallen in

$$\mathcal{N}_1 = \{(2,3)\}, \qquad \mathcal{N}_2 = \{(3,4)\}.$$

Nach der Kleinste-Kosten-Regel wählen wir $(i_s, j_s) = (2,3)$ als (eventuell aufzunehmendes) Indexpaar und setzen dem entsprechend $\sigma = 1$. Nun kommen wir zur Berechnung von $w := A_B^{-1} a^{i_s j_s}$. In Abbildung 4.37 geben wir den Pfad P im aktuellen Baum an, der von $j_s = 3$ zu $i_s = 2$ führt. Hierbei ist $(1,2)$ ein Vorwärts- und $(1,3)$ ein Rückwärtspfeil. Wir hatten nachgewiesen, dass

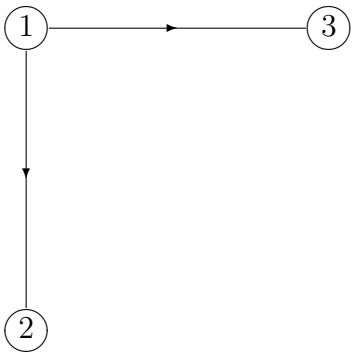


Abbildung 4.37: Pfad von 3 nach 2

$$w_{ij} := \begin{cases} -1, & \text{falls } (i,j) \text{ Vorwärtspfeil in } P, \\ 1, & \text{falls } (i,j) \text{ Rückwärtspfeil in } P, \\ 0, & \text{sonst} \end{cases} \qquad ((i,j) \in \mathcal{B}).$$

Folglich ist

$$w = (w_{12}, w_{13}, w_{24})^T = (-1, 1, 0)^T.$$

Dann ist

$$\delta_1 := \min_{\substack{(i,j)\in\mathcal{B} \\ w_{ij}=\sigma}} x_{ij} = x_{13} = 4, \qquad \delta_2 := \min_{\substack{(i,j)\in\mathcal{B} \\ w_{ij}=-\sigma}} (u_{ij} - x_{ij}) = u_{12} - x_{12} = 2,$$

folglich

$$t^* = \min(4, 2, 2).$$

Die Basisindexmenge wird nicht verändert, die neue Basislösung (wir nennen sie wieder x) ist

$$x = (x_{12}, x_{13}, x_{23}, x_{24}, x_{34})^T = (3, 2, 2, 1, 4)$$

mit den zugehörigen Kosten $c_0 = 26$. Da sich die Basisindexmenge nicht verändert hat, brauchen die reduzierten Kosten nicht neu berechnet zu werden. Diesmal ist

$$\mathcal{N}_1 = \emptyset, \qquad \mathcal{N}_2 = \{(3,4)\}, \qquad \sigma = -1.$$

Das (eventuell in die Basis aufzunehmende) Indexpaar (i_s, j_s) ist daher durch $(i_s, j_s) = (3, 4)$ gegeben. Diesmal ist

$$w = (w_{12}, w_{13}, w_{24})^T = (1, -1, 1)^T,$$

weiter

$$\delta_1 := \min_{\substack{(i,j)\in\mathcal{B}\\ w_{ij}=\sigma}} x_{ij} = x_{13} = 2, \qquad \delta_2 := \min_{\substack{(i,j)\in\mathcal{B}\\ w_{ij}=-\sigma}} (u_{ij} - x_{ij}) = \min(u_{12} - x_{12}, u_{24} - x_{24}) = 0,$$

folglich $t^* = \min(2, 0, 4) = 0$. Das Minimum bei der Berechnung von δ_2 wird für $(i_r, j_r) = (1, 2)$ angenommen. Wegen $t^* = 0$ bleibt man natürlich in dem selben x stehen, die neue Basisindexmenge ist aber $\mathcal{B} = \{(1,3), (2,4), (3,4)\}$. Als neue reduzierte Kosten berechnet man

$$\overline{c}_{12} = -4, \qquad \overline{c}_{23} = -1.$$

Folglich ist $\mathcal{N}_1 = \emptyset$, $\mathcal{N}_2 = \emptyset$, damit $(x, \mathcal{B})$ eine optimale zulässige Basislösung. □

4.5.3 Das klassische Transportproblem

Wir wollen uns überlegen, wie sich das (primale) Netzwerk-Simplex-Verfahren vereinfacht, wenn wir das klassische, also unkapazitierte, Transportproblem zugrunde legen. Es stellt sich heraus, dass die Vereinfachungen nur verhältnismäßig gering sind.

Die Problemstellung brauchen wir nicht noch einmal zu wiederholen, dies geschah schon oft genug. Um das klassische Transportproblem dem linearen Minimale-Kosten-Fluss-Problem unterzuordnen, betrachten wir einen bipartiten gerichteten Graphen $(\mathcal{N}, \mathcal{A})$, bei dem die Knotenmenge zerfällt in die m Lager (Angebotsknoten) $\{1, \ldots, m\}$ und die n Kunden (Bedarfsknoten) $\{1, \ldots, n\}$. Die Menge der Pfeile ist gegeben durch $\mathcal{A} := \{1, \ldots, m\} \times \{1, \ldots, n\}$, ferner seien dem Pfeil $(i, j) \in \mathcal{A}$ Kosten c_{ij} zugeordnet. Sei $l_i \geq 0$, $i = 1, \ldots, m$, das Angebot der Lager, $k_j \geq 0$, $j = 1, \ldots, n$, der Bedarf der Kunden und $\sum_{i=1}^m l_i = \sum_{j=1}^n k_j$, ferner

$$b := \begin{pmatrix} l \\ -k \end{pmatrix}.$$

Gesucht ist ein zulässiger Fluss mit minimalen Kosten. Wir wollen bei der im letzten Unterabschnitt benutzten Bezeichnung bleiben, dass A die (um die letzte Zeile) gestutzte Knoten-Pfeil-Inzidenzmatrix zum bipartiten Digraphen ist (andernfalls müsste man die letzten $n - 1$ Gleichungen mit -1 multiplizieren).

Sei $(x, \mathcal{B})$ eine zulässige Basislösung, also $\mathcal{B} \subset \mathcal{A} = \{1, \ldots, m\} \times \{1, \ldots, n\}$ eine $m + n - 1$-elementige Menge von Pfeilen von den Lagern zu den Kunden mit der Eigenschaft, dass $(\mathcal{N}, \mathcal{B})$ ein $(\mathcal{N}, \mathcal{A})$ aufspannender Baum ist. Zunächst berechnet man

$$y := \begin{pmatrix} u \\ v \end{pmatrix} = A_B^{-T} c_B$$

wie beim allgemeinen Netzwerk-Simplex-Verfahren aus

$$u_i - v_j = c_{ij}, \qquad ((i,j) \in \mathcal{B}),$$

wobei $v_n = 0$ gesetzt ist, anschließend berechnet man die reduzierten Kosten aus

$$\overline{c}_{ij} := c_{ij} - u_i + v_j, \qquad (i,j) \in \mathcal{A} \setminus \mathcal{B}.$$

Da wir das unkapazitierte Transportproblem betrachten (in der Terminologie des Minimale-Kosten-Fluss-Problems ist $u := +\infty\, e$) ist $\sigma = 1$. Die aktuelle zulässige Basislösung $(x, \mathcal{B})$ ist optimal, wenn $\overline{c}_{ij} \geq 0$ für alle $(i,j) \in \mathcal{A} \setminus \mathcal{B}$, die reduzierten Kosten also alle nichtnegativ sind. Wir nehmen an, das sei nicht der Fall und wählen ein $(i_s, j_s) \in \mathcal{A} \setminus \mathcal{B}$ mit $c_{i_s j_s} < 0$. Dies geschehe etwa nach der Kleinste-Kosten-Regel:

$$\overline{c}_{i_s j_s} = \min_{(i,j) \in \mathcal{A} \setminus \mathcal{B}} c_{ij}.$$

Die Berechnung von $w := A_B^{-1} a^{i_s j_s}$ ist im vorigen Abschnitt geschildert worden. Sei P ein Pfad im aktuellen aufspannenden Baum von j_s nach i_s. Da es ja nur Pfeile von den Lagern zu den Kunden gibt, wechseln sich in dem Pfad Rückwärts- und Vorwärtspfeile ab. Nach den Untersuchungen im letzten Unterabschnitt haben wir die Transportmenge auf den Vorwärtspfeilen um $t \geq 0$ zu erhöhen, auf den Rückwärtspfeilen um $t \geq 0$ zu erniedrigen. Genauer definiert man $x(t) \in \mathbb{R}^{m \times n}$ durch

$$x_{i_s j_s}(t) := t$$

sowie

$$x_{ij}(t) := \begin{cases} x_{ij} + t, & \text{falls } (i,j) \text{ Vorwärtspfeil in } P, \\ x_{ij} - t, & \text{falls } (i,j) \text{ Rückwärtspfeil in } P, \\ x_{ij}, & \text{sonst} \end{cases} \qquad ((i,j) \in \mathcal{B})$$

und

$$x_{ij}(t) := x_{ij} \quad ((i,j) \in (\mathcal{A} \setminus \mathcal{B}) \setminus \{(i_s, j_s)\}).$$

Die zugehörigen Transportkosten sind

$$c^T x(t) = c^T x + t \overline{c}_{i_s j_s}.$$

Natürlich wählt man $t^* \geq 0$ wieder möglichst groß, und zwar diesmal so, dass $x^+ := x(t^*) \geq 0$. Offenbar ist t^* das Minimum aller x_{ij}, wobei (i,j) über alle Rückwärtspfeile in P variiert. Dieses Minimum wird durch (mindestens) ein Paar $(i_r, j_r) \in \mathcal{B}$ angenommen. Man setzt $\mathcal{B}^+ := (\mathcal{B} \setminus \{(i_r, j_r)\}) \cup \{(i_s, j_s)\}$. Dann ist $(\mathcal{N}, \mathcal{B}^+)$ ein aufspannender Baum im gegebenen bipartiten Netzwerk, also $(x^+, \mathcal{B}^+)$ eine neue zulässige Basislösung.

Zum Schluss wollen wir wenigstens für das klassische Transportproblem auf die Frage eingehen, wie man eine zulässige Ausgangsbasislösung bestimmen kann. Wir geben zunächst eine Art Modellalgorithmus an.

1. Setze $I := \{1, \ldots, m\}$, $J := \{1, \ldots, n\}$, $\mathcal{B} := \emptyset$.

2. Bestimme $(i^+, j^+) \in I \times J$ nach einer bestimmten Auswahlregel und setze $d := l_{i^+} - k_{j^+}$.

3. Setze $x_{i^+j^+} := \min(l_{i^+}, k_{j^+})$, $\mathcal{B} := \mathcal{B} \cup \{(i^+, j^+)\}$.

4. Falls $d < 0$, dann: $I := I \setminus \{i^+\}$, $k_{j^+} := k_{j^+} - l_{i^+}$.

   Falls $d < 0$ bzw. $l_{i^+} < k_{j^+}$, so wird der Gesamtbestand im Lager i^+ zum Kunden j^+ transportiert. Das Lager i^+ kann für die weiteren Schritte eliminiert werden (weil es leer ist), d. h. man setzt $I := I \setminus \{i^+\}$ und ersetzt k_{j^+}, den Bedarf des Kunden j^+, durch den Restbedarf $k_{j^+} - l_{i^+}$.

5. Falls $d > 0$, dann: $J := J \setminus \{j^+\}$, $l_{i^+} := l_{i^+} - k_{j^+}$.

   Falls $d > 0$ bzw. $l_{i^+} > k_{j^+}$, so wird der Gesamtbedarf des Kunden j^+ durch das Lager i^+ gedeckt. Der Kunde j^+ braucht im weiteren nicht mehr berücksichtigt zu werden (weil er keine Wünsche mehr hat), d. h. man setzt $J := J \setminus \{j^+\}$ und ersetzt l_{i^+}, den Bestand des Lagers i^+, durch den Restbestand $l_{i^+} := l_{i^+} - k_{j^+}$.

6. Falls $d = 0$, dann:

   (a) Falls $(|I| = 1)$ und $(|J| > 1)$, dann: $J := J \setminus \{j^+\}$, $l_{i^+} := 0$.

   (b) Falls $(|I| > 1)$ und $(|J| = 1)$, dann: $I := I \setminus \{i^+\}$, $k_{j^+} := 0$.

   (c) Falls $((|I| = 1)$ und $(|J| = 1))$ oder $((|I| > 1)$ und $(|J| > 1))$, dann: Setze $I := I \setminus \{i^+\}$, $k_{j^+} := 0$ oder $J := J \setminus \{j^+\}$, $l_{i^+} := 0$.

   Falls $d = 0$ bzw. $l_{i^+} = k_{j^+}$, so wird der Gesamtbestand des Lagers i^+ zum Kunden j^+ transportiert und entweder das Lager i^+ oder der Kunde j^+ gestrichen, wobei danach der Kunde j^+ den Bedarf $k_{j^+} = 0$ bzw. das Lager i^+ den Bestand $l_{i^+} = 0$ besitzt. Gibt es nur noch ein (nicht gestrichenes) Lager aber mehr als einen (nicht gestrichenen) Kunden, so wird der Kunde j^+ gestrichen. Entsprechendes gilt in dem Fall, dass nur noch ein (nicht gestrichener) Kunde existiert.

7. Falls $(I \neq \emptyset)$ und $(J \neq \emptyset)$, dann: Gehe nach 2, andernfalls: STOP.

Jetzt wollen wir uns überlegen, dass durch die Nordwesteckenregel eine zulässige Basislösung zum Transportproblem bestimmt wird. Nach genau $n + m - 1$ Schritten terminiert das Verfahren, d. h. die Menge $\mathcal{B} \subset \{1, \ldots, m\} \times \{1, \ldots, n\}$ hat $m + n - 1$ Elemente. Setzt man $x_{ij} := 0$ für alle $(i, j) \in \mathcal{A} \setminus \mathcal{B}$, so ist x eine zulässige Basislösung von $Ax = b$ zur Indexmenge $\mathcal{B}$, wenn wir noch zeigen können, dass $(\mathcal{N}, \mathcal{B})$ ein $(\mathcal{N}, \mathcal{A})$ aufspannender Baum ist. Nun ist aber klar, dass es in $(\mathcal{N}, \mathcal{B})$ keinen Kreis geben kann. Denn Kanten bzw. Pfeile aus $\mathcal{B}$ existieren nur zwischen Lagern und Kunden, ferner können nicht Anfangs- *und* Endknoten eines Pfeiles aus $\mathcal{B}$ mit zwei Pfeilen inzidieren, da ja in jedem Iterationsschritt der Anfangs- oder der Endknoten eines aufzunehmenden Pfeiles aus der Liste der noch zur Verfügung stehenden Lager bzw. Kunden gestrichen wird. Nun wende man noch das folgende einfache Lemma an:

Lemma 5.5 *Sei $G = (V, E)$ ein kreisfreier Graph mit $|E| = |V| - 1$. Dann ist G zusammenhängend und folglich ein Baum.*

Beweis: Seien $(V_1, E_1), \ldots, (V_k, E_k)$ die Zusammenhangskomponenten von (V, E), diese sind natürlich ebenfalls kreisfrei und folglich Bäume. Nach Satz 3.1 in Abschnitt 3.3 ist $|E_i| = |V_i| - 1$, $i = 1, \ldots, k$. Folglich ist

$$|V| - 1 = |E| = \sum_{i=1}^{k} |E_i| = \sum_{i=1}^{k} (|V_i| - 1) = |V| - k,$$

also $k = 1$ und daher (V, E) zusammenhängend. □

Die wichtigsten Strategien zur Auswahl von $(i^+, j^+) \in I \times J$ in Schritt 2 sind:

1. Nordwestecken-Regel.

   Bestimme $(i^+, j^+) \in I \times J$ mit $i^+ := \min_{i \in I} i$, $j^+ := \min_{j \in J} j$.

2. Zeilenminimum-Regel.

   Setze $i^+ := \min_{i \in I} i$ und bestimme $j^+ \in J$ so, dass $c_{i^+ j^+} = \min_{j \in J} c_{i^+ j}$.

3. Spaltenminimum-Regel.

   Setze $j^+ := \min_{j \in J} j$ und bestimme $i^+ \in I$ so, dass $c_{i^+ j^+} = \min_{i \in I} c_{i j^+}$.

4. Matrixminimum-Regel.

   Bestimme $(i^+, j^+) \in I \times J$ so, dass $c_{i^+ j^+} = \min_{(i,j) \in I \times J} c_{ij}$.

Jedesmal wird, wie wir uns oben überlegt haben, eine zulässige Basislösung berechnet.

Beispiel: Wir betrachten ein Transportproblem, bei dem die Daten durch

	k^T
l	c

$=$

	10	3	2	3
5	6	4	3	2
9	5	2	4	4
4	4	2	2	4

gegeben sind. Mit Hilfe der Matrixminimumregel erhalten wir die Ausgangsbasislösung

$$x = \begin{pmatrix} 2 & & & 3 \\ 6 & 3 & & \\ 2 & & 2 & \end{pmatrix}, \qquad \mathcal{B} = \{(1,1), (1,4), (2,1), (2,2), (3,1), (3,3)\}$$

mit den Kosten $c_0 = 66$. Den zugehörigen aufspannenden Baum im Transportgraphen geben wir in Abbildung 4.38 an. Danach berechnet man u_i, $i = 1, \ldots, 3$, und v_j, $j = 1, \ldots, 4$ mit $v_4 = 0$ aus

$$u_i - v_j = c_{ij}, \qquad (i, j) \in \mathcal{B}.$$

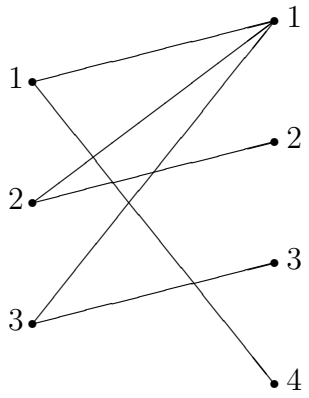


Abbildung 4.38: Ein aufspannender Baum zum Transportgraphen

Dies ist sehr einfach, wenn man die c_{ij}, $(i,j) \in \mathcal{B}$, eingerahmt in ein $m \times n$-Feld einträgt und die Ränder u, v auffüllt. Anschließend besetzt man die Positionen, die zu $(i,j) \in \mathcal{A} \setminus \mathcal{B}$ gehören mit den reduzierten Kosten $\overline{c}_{ij} := c_{ij} - u_i + v_j$. Dies liefert:

$$\begin{array}{|c|c|}\hline c & u \\ \hline v^T & \\ \hline \end{array} = \begin{array}{|cccc|c|}\hline \boxed{6} & 1 & -1 & \boxed{2} & 2 \\ \boxed{5} & \boxed{2} & 1 & 3 & 1 \\ \boxed{4} & 1 & \boxed{2} & 4 & 0 \\ \hline -4 & -1 & -2 & 0 & \\ \hline \end{array}$$

In die Basis aufzunehmen ist daher $(1,3)$. Damit ist $x(t)$ gegeben durch

$$x(t) = \begin{pmatrix} 2-t & & t & 3 \\ 6 & 3 & & \\ 2+t & & 2-t & \end{pmatrix},$$

ferner ist $t^* = 2$. Bei dem zu entfernenden Paar hat man die Wahl, wir nehmen $(1,1)$. Die neue Basislösung ist daher

$$x = \begin{pmatrix} & & 2 & 3 \\ 6 & 3 & & \\ 4 & & 0 & \end{pmatrix}, \qquad \mathcal{B} = \{(1,3),(1,4),(2,1),(2,2),(3,1),(3,3)\},$$

sie ist entartet und hat die Kosten $c_0 = 66 + 2(-1) = 64$. Im nächsten Schritt erhält man

$$\begin{array}{|c|c|}\hline c & u \\ \hline v^T & \\ \hline \end{array} = \begin{array}{|cccc|c|}\hline 7 & 2 & \boxed{3} & \boxed{2} & 2 \\ \boxed{5} & \boxed{2} & 3 & 2 & 2 \\ \boxed{4} & 1 & \boxed{2} & 4 & 1 \\ \hline -3 & 0 & -1 & 0 & \\ \hline \end{array},$$

alle reduzierten Kosten sind nichtnegativ, die erhaltene Basislösung daher optimal. □

4.5.4 Aufgaben

1. In Abbildung 4.39 ist ein gerichteter Graph dargestellt, ferner sind auf den Pfeilen Kosten, Kapazitäten und Flüsse angegeben. Die Zahlen an den Knoten sind die b_k, $k \in$

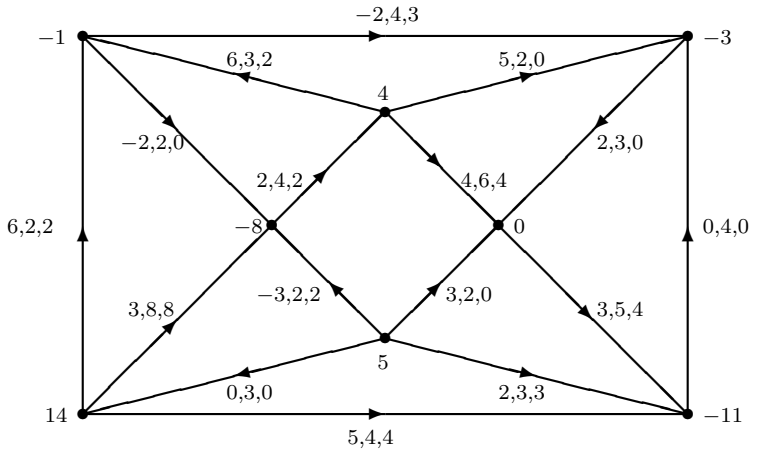


Abbildung 4.39: Ein Minimale-Kosten-Fluss-Problem

$\mathcal{N}$, die Zahlen an den Pfeilen sind in der Reihenfolge c_{ij}, u_{ij}, x_{ij} (Kosten, Kapazitäten, Flüsse) zu verstehen. Man zeige, dass der angegebene Fluss zulässig ist und eine Lösung des zugehörigen Minimale-Kosten-Fluss-Problems ist.

2. Man zeige: Das gegebene Minimale-Kosten-Fluss-Problem sei zulässig. Dann besitzt das Problem genau dann eine Lösung, wenn es in $(\mathcal{N}, \mathcal{A})$ keinen gerichteten Zyklus mit negativen Kosten gibt derart, dass die Kapazitäten auf allen Pfeilen des Zyklus ∞ sind.

3. Gegeben sei ein Minimale-Kosten-Fluss-Problem mit den in Abbildung 4.40 angegebenen Daten. In der Abbildung links sind die Knoten nummeriert und zu den Pfeilen sind die Kosten angegeben. Rechts ist in den Knoten das Angebot bzw. der Bedarf und zu den Pfeilen die entsprechenden Kapazitäten angegeben. Mit Hilfe des Netzwerk-

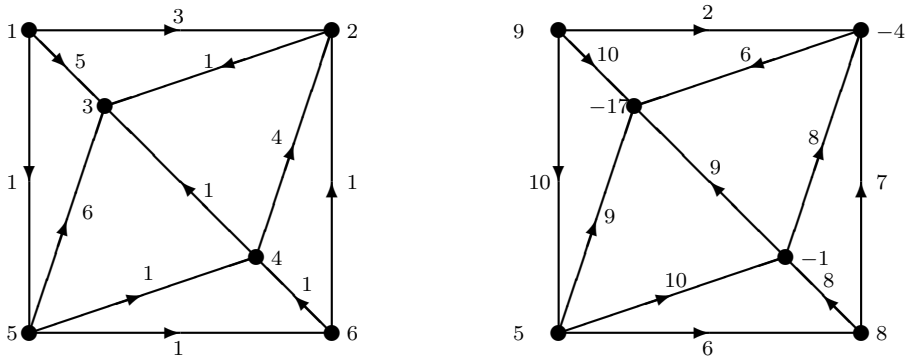


Abbildung 4.40: Ein Minimale-Kosten-Fluss-Problem

Simplex-Verfahren löse man dieses Problem. Hierbei starte man mit der zulässigen Basislösung

$$x = (x_{12}, x_{13}, x_{15}, x_{23}, x_{42}, x_{43}, x_{53}, x_{54}, x_{56}, x_{62}, x_{64})^T = (0, 0, 9, 6, 4, 3, 8, 0, 6, 6, 8)^T$$

zur Basisindexmenge

$$\mathcal{B} = \{(1, 5), (4, 2), (4, 3), (5, 3), (6, 2)\}.$$

Weshalb ist $(x, \mathcal{B})$ eine Basislösung?

4. In einer Stadt braucht man zu einer gewissen Zeit Ersatzbusse. Und zwar werden an Orten O_1, O_2, O_3 und O_4 genau 3, 3, 4 bzw. 5 Busse benötigt, welche aus Garagen G_1, G_2 bzw. G_3 kommen müssen, in denen 2, 6 bzw. 7 Busse für diesen Notfall bereitstehen. Die Gesamtzeit, die die Busse von ihrer Garage zu ihrem Bestimmungsort fahren, soll minimiert werden. Die Zeiten liest man aus der folgenden Tabelle ab.

	O_1	O_2	O_3	O_4
G_1	13	11	15	20
G_2	17	14	12	13
G_3	18	18	15	12

Man bestimme einen optimalen Transportplan.

Kapitel 5

Ganzzahlige und kombinatorische Optimierungsaufgaben

5.1 Beispiele

5.1.1 Ganzzahlige lineare Optimierungsaufgaben

Wir sprechen von einer gannzahligen linearen Optimierungsaufgabe, wenn an die Variablen einer linearen Optimierungsaufgabe zusätzlich die Forderung gestellt wird, dass sie ganzzahlig sind.

Beispiel: Eine Möbelfabrik stellt vier Arten von Schreibtischen her. Jeder Schreibtisch wird zuerst in der Schreinerei hergestellt und dann zur Veredelungswerkstatt geschickt, wo er lackiert, gewachst und poliert wird. Die Anzahl der Arbeitsstunden, die in jeder Werkstatt benötigt wird, sowie der Gewinn (Einnahmen minus Arbeitskosten) seien wie folgt:

	Schreibtisch 1	Schreibtisch 2	Schreibtisch 3	Schreibtisch 4
Schreinerei	4	9	7	10
Veredelungs− werkstatt	1	1	3	40
Gewinn	12	20	18	40

Infolge einer Kapazitätsbegrenzung der Fabrik können innerhalb der nächsten sechs Monate nicht mehr als 6000 Arbeitsstunden in der Schreinerei und 4000 in der Veredelungswerkstatt ausgeführt werden. Unter diesen Restriktionen soll der Gewinn maximiert werden. Wie lautet der optimale Produktionsplan?

Bezeichnet man mit x_j die Anzahl der vom Schreibtisch j herzustellenden Exemplare, so genügt der *Produktionsplan* $x = (x_1, x_2, x_3, x_4)^T$ den geforderten Restriktionen, wenn

$$\begin{pmatrix} 4 & 9 & 7 & 10 \\ 1 & 1 & 3 & 40 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{pmatrix} \leq \begin{pmatrix} 6000 \\ 4000 \end{pmatrix}, \qquad \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{pmatrix} \geq \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}.$$

Hinzu kommt die Ganzzahligkeitsforderung, dass nämlich $x_j \in \mathbb{N}_0$, $j = 1, \ldots, 4$. Unter diesen Bedingungen ist Gewinn

$$g(x) := \begin{pmatrix} 12 \\ 20 \\ 18 \\ 40 \end{pmatrix}^T \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{pmatrix}$$

zu maximieren. Verzichtet man auf die Forderung nach der Ganzzahligkeit der Variablen, so erhält man als Lösung

$$x_1^* = \frac{4000}{3}, \quad x_2^* = 0, \quad x_3^* = 0, \quad x_4^* = \frac{200}{3}$$

mit dem zugehörigen Zielfunktionswert $g(x^*) = 56000/3$. Rundet man (nach unten, damit die Zulässigkeit erhalten bleibt), so erhält man $x = (1\,333, 0, 0, 66)$ mit dem Gewinn $g(x) = 18\,636$. Dies ist aber jedenfalls nicht Lösung der ganzzahligen linearen Optimierungsaufgabe, da auch $x = (1\,333, 0, 1, 66)$ zulässig ist und den höheren Gewinn 18 654 erbringt. □

Allgemein nennt man z. B. die Aufgabe

$$\text{(ILP)} \qquad \begin{cases} \text{Minimiere} \quad c^T x \quad \text{unter den Nebenbedingungen} \\ x \geq 0, \quad Ax = b, \quad x \in \mathbb{Z}^n, \end{cases}$$

ein *ganzzahliges lineares Programm* in Standardform. Hierbei seien $A \in \mathbb{Z}^{m \times n}$ mit Rang $(A) = m$, $b \in \mathbb{Z}^m$ und (gewöhnlich auch) $c \in \mathbb{Z}^n$ gegeben. Ist A unimodular, so kann das Simplexverfahren angewandt werden, da wir wegen Satz 2.3 in Abschnitt 4.2 wissen, dass dann jede Ecke von $M := \{x \in \mathbb{R}^n : x \geq 0,\ Ax = b\}$ ganzzahlig ist. Wir werden daher davon ausgehen, dass A nicht unimodular ist. Dem ganzzahligen linearen Programm (ILP) wird das relaxierte Programm

$$\text{(LP)} \qquad \begin{cases} \text{Minimiere} \quad c^T x \quad \text{unter den Nebenbedingungen} \\ x \geq 0, \quad Ax = b \end{cases}$$

zugeordnet, das aus (ILP) entsteht, indem die Ganzzahligkeitsfordrung fallengelassen wird. Zwei einfache, aber wichtige Beobachtungen wollen wir hier schon notieren:

- Ist x^* eine Lösung von (LP) mit $x^* \in \mathbb{Z}^n$, so ist x^* auch eine Lösung von (ILP).
- Ist x^* eine Lösung von (LP), so ist $c^T x^* \leq \inf$ (ILP).

Ersetzt man in (ILP) die Forderung $x \in \mathbb{Z}^n$ durch $x \in \{0, 1\}^n$, so spricht man von einem *binären* linearen Programm. Eine naheliegende Relaxation eines binären linearen Programms besteht darin, die Bedingung $x \in \{0, 1\}^n$ durch $0 \leq x \leq e$ zu ersetzen (wobei e einmal wieder der Vektor ist, dessen Komponenten alle gleich 1 sind). Die obigen beiden Bemerkungen gelten dann entsprechend.

5.1.2 Das Rucksackproblem

Die folgende Modellsituation führt zum *Rucksackproblem* (knapsack problem). Ein Bergsteiger möchte auf einer Bergtour n Gegenstände in einem Rucksack mitnehmen, wobei sein Tragevermögen durch das Gesamtgewicht β beschränkt ist. Für $j = 1, \ldots, n$ ist das Gewicht a_j und der "Wert" (dieser soll die Bedeutung des Gegenstandes für die Bergtour widerspiegeln) $c_j > 0$ des j-ten Gegenstandes bekannt. Der Gesamtwert der mitgeführten Gegenstände ist zu maximieren, wobei das Tragevermögen zu berücksichtigen ist.

Definiert man

$$x_j := \begin{cases} 1, & j\text{-ter Gegenstand wird gewählt,} \\ 0, & \text{sonst,} \end{cases}$$

definiert man ferner $c = (c_j) \in \mathbb{R}^n$, $a = (a_j) \in \mathbb{R}^n$ und $x = (x_j) \in \mathbb{R}^n$, so lässt sich das (binäre) Rucksackproblem in der Form

$$\begin{cases} \text{Maximiere} \quad c^T x \quad \text{unter der Nebenbedingung} \\ a^T x \le \beta, \qquad x_j \in \{0,1\} \qquad (j = 1, \ldots, n) \end{cases}$$

schreiben. Es sei darauf hingewiesen, dass sich ein solches Problem nicht nur bei Bergtouren oder Antarktis-Expeditionen stellt. Wir stellen uns vor, dass eine Firma zur Einführung eines neuen Produktes eine Werbekampagne starten will. Hierzu stehen n verschiedene Medien und ein Werbeetat von β DM zur Verfügung. Eine Werbung im j-ten Medium koste a_j DM, c_j gebe die Anzahl potentieller Käufer im j-ten Medium an. Die Firma hat offenbar das Rucksackproblem zu lösen.

Wir wollen hier schon eine Methode zur Gewinnung einer i. allg. guten (im worst case Fall aber auch beliebig schlechten) Näherungslösung für das Rucksackproblem schildern. Der erste Schritt besteht darin, eine *Relaxation* des binären Rucksackproblems vorzunehmen, indem man die kombinatorische Nebenbedingung $x_j \in \{0,1\}$ durch $0 \le x \le e$ ersetzt. Nimmt man an, dass die n Gegenstände entsprechend ihrer relativen Bedeutung geordnet sind, so kann man diese Relaxation geschlossen lösen, wie die folgende Aussage zeigt:

- *Seien $a = (a_j) \in \mathbb{R}^n$, $c = (c_j) \in \mathbb{R}^n$ mit $a > 0$, $c > 0$ und $c_1/a_1 \ge \cdots \ge c_n/a_n$ gegeben. Mit $\beta > 0$ betrachte man die lineare Optimierungsaufgabe*

  (P) *Maximiere* $c^T x$ *unter den Nebenbedingungen* $a^T x \le \beta, \quad 0 \le x \le e.$

  O. B. d. A. sei $a^T e > \beta$ (andernfalls ist $x^* = e$ die Lösung von (P)). Dann ist eine Lösung $x = (x_j)$ von (P) gegeben durch

$$x_j := \begin{cases} 1 & \text{für} \quad j < p, \\ \dfrac{1}{a_p}\Big(\beta - \displaystyle\sum_{k<p} a_k\Big) & \text{für} \quad j = p, \\ 0 & \text{für} \quad j > p. \end{cases}$$

  Hierbei ist p, der sogenannte kritische Index, die kleinste natürliche Zahl mit $\sum_{j \le p} a_j \ge \beta$.

Denn: Zunächst ist der angegebene Vektor x zulässig für (P), da $(1/a_p)[\beta - \sum_{k<p} a_k] \in (0,1]$ nach Definition des kritischen Index p. Zu (P) formulieren wir eine weitere lineare (nämlich die duale) Optimierungsaufgabe

$$\text{(D)} \qquad \begin{cases} \text{Minimiere} & \beta\alpha + e^T y \quad \text{unter den Nebenbedingungen} \\ & \alpha a + y \geq c, \quad \alpha \geq 0, \quad y \geq 0. \end{cases}$$

Man definiere nun

$$\alpha := \frac{c_p}{a_p}, \qquad y_j := \begin{cases} c_j - \dfrac{c_p}{a_p} a_j, & j = 1, \ldots, p-1, \\ 0, & j = p, \ldots, n. \end{cases}$$

Man weist sehr leicht nach, dass (α, y) für (D) zulässig ist und $c^T x = \beta\alpha + e^T y$ gilt. Hieraus erhält man (schwacher Dualitätssatz), dass x Lösung von (P) (und (α, y) Lösung von (D)). Ist die p-te Komponente von x_p gleich 1, ist also $\sum_{j \leq p} a_j = \beta$, so x als Lösung des relaxierten Problems, welche zulässig für das kombinatorische Rucksackproblems ist, auch eine Lösung des kombinatorischen Problems. Andernfalls kann man die p-te Komponente auf 0 setzen, um eine Näherung zu erhalten.

Beispiel: Sei

$$a' = (3,2,1,4,4)^T, \qquad c' = (2,3,1,4,7), \qquad \beta = 7.$$

Zunächst müssen die Variablen nach ihrer Bedeutung geordnet werden. Man erhält etwa (die Anordnung ist nicht eindeutig)

$$a = (4,2,1,4,3)^T, \qquad c = (7,3,1,4,2)^T,$$

die erste und die fünfte Variable werden vertauscht. Mit der obigen naheliegenden Methode erhalten wir $x = (1,1,1,0,0)^T$ als zugehörige Lösung, bzw. $x' = (0,1,1,0,1)^T$ als Lösung des Ausgangsproblems, der zugehörige Wert ist $c^T x = (c')^T x' = 11$. Wir haben Glück, dies ist sogar die Optimallösung. Um nicht den Eindruck zu erwecken, das sei jedesmal der Fall, geben wir ein weiteres Zahlenbeispiel an, bei dem die Gegenstände schon nach ihrer Bedeutung geordnet sind. Es sei nämlich

$$a = (2,1,8,12,2,14,3,10,11)^T, \qquad c = (6,2,10,14,2,13,2,5,1)^T, \qquad \beta = 17.$$

Hier ist $p = 4$, mit der obigen Aussage erhält man als Lösung des relaxierten Problems

$$x = (1,1,1,\tfrac{1}{2},0,0,0,0,0)^T.$$

Der Zielfunktionswert ist 25. Das zugehörige binäre Rucksackproblem kann natürlich keinen größeren Zielfunktionswert liefern. Mit der obigen Methode erhalten wir

$$x = (1,1,1,0,1,0,0,0,0)^T, \qquad c^T x = 20.$$

Dagegen ist die optimale Lösung

$$x = (1,1,0,1,1,0,0,0,0)^T, \qquad c^T x = 24.$$

Hier hat man also nicht ganz soviel Glück gehabt. □

Im Gegensatz zum obigen binären Rucksackproblem spricht man bei

$$\left\{ \begin{array}{c} \text{Maximiere} \quad c^T x \quad \text{unter der Nebenbedingung} \\ a^T x \leq \beta, \qquad x \geq 0, \qquad x \in \mathbb{Z}^n \end{array} \right.$$

vom *ganzzahligen Rucksackproblem*. Zu einer solchen Formulierung kommt man beim Rucksackproblem, wenn es für die Expedition sinnvoll ist, einen Gegenstand auch in mehreren Exemplaren mitzunehmen. Wir verweisen noch auf S. MARTELLO, P. TOTH (1990)[1], die wohl im Augenblick ausführlichste Darstellung des Rucksackproblems nebst Varianten.

5.1.3 Verschnittprobleme

Wir zitieren K. NEUMANN, M. MORLOCK (1993, S. 423): "Bei vielen industriellen Fertigungsvorgängen ergeben sich Probleme beim Zuschneiden von Material für die weitere Verarbeitung oder den Verkauf. Aus vorhandenen Ausgangsmaterialien, wie z. B. Papier, Stahl, Holz oder Kunststoff, sind dabei Teile in der benötigten Größe zu schneiden. Gesucht ist ein sogenannter *Verschnittplan*, bei dem entweder der Verschnitt (Abfall) minimal ist oder die kleinstmögliche Menge des Ausgangsmaterials für das Zuschneiden der benötigten Teile erforderlich ist. Man spricht daher bei diesen Aufgabenstellungen von *Verschnittproblemen*."

Wir wollen uns damit begnügen, etwas zu *eindimensionalen* Verschnittproblemen zu sagen.

Beispiel: Eine Papiermühle produziere Papierrollen ("Ausgangsrollen") der Breite $L > 0$. Ein Kundenauftrag besteht darin, dass b_i Rollen der Breite $l_i \in (0, L]$, $i = 1, \ldots, m$, bestellt werden. Zusammenfassen ergibt den Vektor $b = (b_1, \ldots, b_m)^T$. Diese Rollen sollen von den Ausgangsrollen so abgeschnitten werden, dass möglichst wenig Rollen angeschnitten werden oder der Gesamtabfall möglichst klein ist, man spricht vom (eindimensionalen) Verschnittproblem. Um diese Aufgabe als (ganzzahlige) lineare Optimierungsaufgabe zu schreiben, muss man Bedingungen an mögliche Schnittpläne aufstellen bzw. alle sinnvollen Schnittpläne angeben. Ein zulässiger *Schnittplan* ist ein Vektor $a = (a_1, \ldots, a_m)^T \in \mathbb{Z}^m$ mit $a \geq 0$ und $\sum_{i=1}^m a_i l_i \leq L$. Sei n die Anzahl der Schnittpläne, diese seien $a^{(1)}, \ldots, a^{(n)}$. Hiermit definieren wir die Matrix

$$A := \left(\begin{array}{ccc} a^{(1)} & \cdots & a^{(n)} \end{array} \right) \in \mathbb{R}^{m \times n}.$$

Gesucht ist ein Vektor $x = (x_1, \ldots, x_n)^T \geq 0$, alle Komponenten ganzzahlig, welcher aussagt, dass der j-te Schnittplan x_j mal auszuführen ist. Dann ist $\sum_{j=1}^n x_j$ die Anzahl der benötigten Schnittpläne bzw. der benutzten Ausgangsrollen, diese gilt es zu minimieren. Denkbar wäre aber noch eine etwas andere Zielfunktion. Der Schnittplan $a = (a_1, \ldots, a_m)$ verursacht den Abfall $c(a) := L - \sum_{i=1}^m a_i l_i$. Ist c_j der Abfall des j-ten Schnittplans $a^{(j)}$ und entsteht $c = (c_j) \in \mathbb{R}^n$ durch Zusammenfügen, so ist durch $c^T x$

[1] S. Martello, P. Toth, Knapsack Problems. Algorithms and Computer Implementations. John Wiley & Sons, 1990.

der Gesamtabfall gegeben. Als Nebenbedingung hat man jeweils, dass der Kundenauftrag erfüllt wird bzw. $Ax \geq b$ gilt, außerdem natürlich, dass $x \geq 0$ und $x \in \mathbb{Z}^n$. Als Relaxation kann man jeweils die linearen Optimierungsaufgaben betrachten, die man durch Aufheben der Ganzzahligkeitsforderung erhält, also

(P_1) Minimiere $e^T x$ unter den Nebenbedingungen $Ax \geq b,\ x \geq 0$

bzw.

(P_2) Minimiere $c^T x$ unter den Nebenbedingungen $Ax \geq b,\ x \geq 0.$

Ist eine Lösung ganzzahlig, so ist sie Lösung des entsprechenden (eindimensionalen) Verschnittproblems, andernfalls muss man runden, um wenigstens eine Näherungslösung zu gewinnen.

Das folgende Zahlenbeispiel ist noch relativ harmlos, da die Anzahl der zulässigen Schnittpläne klein ist. Sei $L = 60$, $m = 3$ und $l = (28, 20, 15)^T$. In den sieben Spalten von

$$A = \begin{pmatrix} 2 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 3 & 2 & 1 & 0 \\ 0 & 0 & 2 & 0 & 1 & 2 & 4 \end{pmatrix}$$

stehen die zulässigen Schnittpläne. Wir nehmen an, es sei $b = (30, 60, 48)^T$, es werden also 30 Rollen der Länge 28 usw. geordert. Der "Abfallvektor" ist in diesem Fall $c = (4, 12, 2, 0, 5, 10, 0)^T$. Als gemeinsame (ist dies ein Zufall?) Lösung der beiden relaxierten Probleme berechnet man $x = (15, 0, 0, 20, 0, 0, 12)^T$ (auch $x = (3, 0, 24, 20, 0, 0, 0)^T$ ist eine gemeinsame Lösung) mit den Zielfunktionswerten $e^T x = 47$ bzw. $c^T x = 60$. Da diese Lösung der relaxierten Probleme ganzzahlig ist, ist sie Lösung des entsprechenden Verschnittproblems.

Bei V. Chvátal (1983, S. 196) wird ein Problem mit den Daten $L = 100$, $m = 4$ sowie $l = (45, 36, 31, 14)^T$ und $b = (97, 610, 395, 211)^T$ betrachtet. Es stellt sich heraus, dass es hier schon 37 zulässige Schnittpläne gibt. Die Lösung des relaxierten Problems ist nicht ganzzahlig. Durch Runden auf die nächst kleinere ganze Zahl erhält man einen Vektor, durch den der Kundenauftrag "fast" erfüllt wird. Jedenfalls in diesem Beispiel kann man überblicken, wieviele Rollen man noch für den Rest benötigt □

5.1.4 Partitions- und Überdeckungsprobleme

Die jetzt vorzustellenden Probleme gehören zu den wichtigsten speziell strukturierten binären bzw. kombinatorischen Optimierungsaufgaben.

Gegeben seien

(1) Eine endliche Menge M, etwa $M = \{1, \ldots, m\}$.

(2) Eine Menge $\mathcal{F}$ von nichtleeren Mengen $F \subset M$, etwa $\mathcal{F} = \{F_1, \ldots, F_n\}$ mit $F_j \subset M$, $j = 1, \ldots, n$,

(3) Eine Kostenfunktion $c\colon \mathcal{F} \longrightarrow \mathbb{R}_+$, d. h. jedem $F_j \in \mathcal{F}$ ist ein Kostenwert $c_j > 0$ zugeordnet, $j = 1, \ldots, n$. Diese seien zu dem Vektor $c = (c_j)$ zusammengefasst.

Aus $\mathcal{F}$ ist eine Teilmenge $\mathcal{F}^* \subset \mathcal{F}$ so auszuwählen, dass $\mathcal{F}^*$ eine kostenminimale Partition bzw. Überdeckung von M ist. Genauer:

(a) Partitionsproblem.

Gesucht ist eine kostenminimale Partition der Menge M durch Mengen aus $\mathcal{F}$, also eine Auswahl von Teilmengen $F_{i_1}, \ldots, F_{i_k}$ mit $\bigcup_{j=1}^{k} F_{i_j} = M$ und $F_{i_j} \cap F_{i_l} = \emptyset$, $1 \le j < l \le k$, die $\sum_{j=1}^{k} c_{i_j}$ minimiert.

(b) Überdeckungsproblem.

Gesucht ist eine kostenminimale Überdeckung der Menge M mit Mengen aus $\mathcal{F}$, also eine Auswahl $F_{i_1}, \ldots, F_{i_k}$ von Mengen aus $\mathcal{F}$ mit $\bigcup_{j=1}^{k} F_{i_j} = M$, für die $\sum_{j=1}^{k} c_{i_j}$ minimal ist.

Es ist nicht schwierig, diese Aufgaben als binäre lineare Optimierungsaufgaben zu schreiben. In Abhängigkeit von $\mathcal{F}^* \subset \mathcal{F}$ definiere man hierzu die binäre Variable $x \in \{0,1\}^n$ durch

$$x_j = \begin{cases} 1 & \text{falls} \quad F_j \in \mathcal{F}^*, \\ 0 & \text{sonst.} \end{cases}$$

Schließlich sei A die $m \times n$-Inzidenzmatrix der Elemente von $\mathcal{F}$ (Spalten von A) gegen die Elemente von M (Zeilen von A), d. h. $A = (a_{ij})$ ist gegeben durch

$$a_{ij} := \begin{cases} 1 & \text{falls} \quad i \in F_j, \\ 0 & \text{sonst.} \end{cases}$$

Bezeichnet man ferner wie üblich mit e den Vektor (hier im $\mathbb{R}^m$), dessen Komponenten alle gleich 1 sind, so erhält man für die Probleme (a)–(c):

(a) Partitionsproblem.

$$\text{Minimiere} \quad c^T x \quad \text{unter den Nebenbedingungen} \quad Ax = e, \quad x \in \{0,1\}^n.$$

(b) Überdeckungsproblem.

$$\text{Minimiere} \quad c^T x \quad \text{unter den Nebenbedingungen} \quad Ax \ge e, \quad x \in \{0,1\}^n.$$

Nun wollen wir andeuten, wo Probleme der obigen Form vorkommen. Hierzu betrachten wir ein *Projektplanungsproblem*. Zur Realisierung eines Projektes, etwa einer Expedition, seien m Leistungen $L_1, \ldots, L_m$ notwendig. Es stehen n Bewerber $B_1, \ldots, B_n$ zur Diskussion, wobei B_j eine Teilmenge $L(B_j)$ der Menge $\mathcal{L} = \{L_1, \ldots, L_m\}$ aller erforderlichen Leistungen erbringen kann und Personalkosten c_j verursacht, $j = 1, \ldots, n$. Unter den Bewerbern soll eine Auswahl getroffen werden, d. h. ein Team $\mathcal{B}^* \subset \mathcal{B} := \{B_1, \ldots, B_n\}$ bestimmt werden derart, dass jede notwendige Leistung von mindestens einem Mitglied aus $\mathcal{B}^*$ erbracht werden kann, also

$$\bigcup_{j:B_j \in \mathcal{B}^*} L(B_j) = \mathcal{L},$$

und die auftretenden Personalkosten minimal sind. Man erhält also ein Problem der kostenminimalen Überdeckung. Sind alle c_i gleich, so wird nach einem Team mit möglichst wenig Mitgliedern gefragt. Weitere Anwendungen (siehe I. M. Bomze, W. Grossmann (1993, S. 433)) betreffen die Routenplanung in einem Fahrzeugpark. Die Menge M entspricht den anzufahrenden Zielen, die Mengen F_j repräsentieren die möglichen Kombinationen von Zielen, die ein Fahrzeug anlaufen kann, und c_j die damit verbundenen Kosten. Das Partitionsproblem bestimmt dann den kostengünstigsten Fahrzeugeinsatz unter der Zusatzannahme, dass jedes Ziel nur von einem Fahrzeug angefahren wird, das Überdeckungsproblem jenen Einsatzplan, bei welchem auch mehrfaches Anfahren zulässig ist.

5.1.5 Das Problem des Handlungsreisenden

Auf das Problem des Handlungsreisenden (Traveling Salesman Problem, TSP) sind wir kurz schon in Abschnitt 3.6 über Hamiltonsche Kreise eingegangen. In einem vollständigen Graphen $G = (V, E)$ mit $V = \{1, \ldots, n\}$ tragen alle Kanten $(i, j) \in E$ gewisse Kosten c_{ij}. Da wir von einem *ungerichteten* vollständigen Graphen ausgehen, wird die Kostenmatrix (c_{ij}) also als *symmetrisch* vorausgesetzt. Der Deutlichkeit halber spricht man dann auch vom *symmetrischen TSP*. Ein Kreis in G, der alle Ecken genau einmal enthält, heißt bekanntlich ein Hamiltonscher Kreis, seine Kosten sind natürlich die Summe der entsprechenden Kantenkosten. Beim Problem des Handlungsreisenden kommt es darauf an, einen Hamiltonschen Kreis mit minimalen Kosten zu bestimmen. Historische Bemerkungen zum TSP finden sich in Chapter 1 des Buches von E. L. Lawler et al. (1985)[2]. Man kann das Problem des Handlungsreisenden als eine binäre lineare Optimierungsaufgabe formulieren (was aber wenig hilft). Hierzu liegt es nahe, die Variablen

$$x_{ij} = \begin{cases} 1 & \text{falls die Tour direkt von } i \text{ nach } j \text{ führt,} \\ 0 & \text{sonst} \end{cases}$$

einzuführen. Die Zielfunktion $\sum_{i,j=1}^{n} c_{ij} x_{ij}$ ist zu minimieren. Zu den Nebenbedingungen gehören auf alle Fälle $x_{ij} \in \{0, 1\}$ (Entscheidung ob von i direkt nach j gegangen wird oder nicht), $\sum_{i=1}^{n} x_{ij} = 1$, $j = 1, \ldots, n$, d. h. jede Stadt muss besucht werden, $\sum_{j=1}^{n} x_{ij} = 1$, $i = 1, \ldots, n$, d. h. jede Stadt muss verlassen werden. Bisher sind dies nur die Nebenbedingungen des Zuordnungsproblems, dessen Menge der zulässigen Lösungen die Menge aller Permutationen von $\{1, \ldots, n\}$ ist. Dagegen ist die Menge der zulässigen Lösungen des Traveling Salesman Problems die Menge aller zyklischen Permutationen von $\{1, \ldots, n\}$, also der Permutationen ohne Unterzyklen. Es stellt sich daher die Frage, durch welche Art von Nebenbedingungen an $x = (x_{ij})$ Unterzyklen

[2] E. L. Lawler, J. K. Lenstra, A. H. G. Rinnoy Kan, D. B. Shmoys (1985) *The Traveling Salesman Problem*. J. Wiley, New York.

bzw. Subtouren verboten werden können. Z. B. genügt

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

als Permutationsmatrix den Nebenbedingungen des Zuordnungsproblems, nicht aber denen des Traveling Salesman Problems, da die Tour in die Subtouren $1 \to 2 \to 3 \to 1$ und $4 \to 5 \to 6 \to 4$ zerfällt. Eine Möglichkeit diese beiden Soubtoren auszuschließen besteht in der Forderung, dass es mindestens eine Verbindung zwischen $\{1, 2, 3\}$ und $\{4, 5, 6\}$ gibt, dass also zu einer zulässigen Tour ein Paar $(i, j) \in \{1, 2, 3\} \times \{4, 5, 6\}$ mit $x_{ij} \geq 1$ existiert. Allgemein werden Subtouren verhindert, wenn

$$\sum_{\substack{i,j=1 \\ i \in I,\, j \in N \setminus I}}^{n} x_{ij} \geq 1 \qquad \text{für alle } I \subset N := \{1, \ldots, n\} \text{ mit } 2 \leq |I| \leq n-2.$$

Damit ist das Traveling Salesman Problem als ein binäres lineares Optimierungsproblem geschrieben (siehe G. L. Nemhauser, L. A. Wolsey (1988, S. 10)[3]). Die Anzahl der Restriktionen ist allerdings (schon bei moderatem n) sehr groß, ferner sind viele von ihnen redundant.

5.2 Schnittebenenverfahren für (ganzzahlige) lineare Optimierungsaufgaben

Wir wollen uns sehr kurz fassen und nur die einfachsten Ideen zur Aufstellung von Verfahren für rein ganzzahlige lineare Optimierungsaufgaben schildern.

5.2.1 Motivation für Schnittebenenverfahren von Gomory

Schnittebenenverfahren gehören wohl zu den ältesten und auch theoretisch am umfassendsten untersuchten Verfahren zur Lösung ganzzahliger linearer Optimierungsaufgaben. Ebenso wie das Simplexverfahren von G. B. Dantzig den Durchbruch bei ("gewöhnlichen") linearen Optimierungsaufgaben brachte, wurden durch das Schnittebenenverfahren von R. E. Gomory (1958) ganzzahlige lineare Optimierungsaufgaben behandelbar. Allerdings mit dem wesentlichen Unterschied, dass Gomory-Verfahren nur bei verhältnismäßig kleindimensionalen Problemen effizient sind.

Die Vorgehensweise bei Schnittebenenverfahren für ganzzahlige lineare Programme ist im Prinzip die folgende:

1. Sei x^* eine Lösung des relaxierten Programms (LP). Wir können annehmen, dass $x^* \notin \mathbb{Z}^n$, da man andernfalls fertig ist.

[3] G. L. Nemhauser, L. A. Wolsey, Integer and Combinatorial Optimization. Wiley, 1988.

2. Bestimme eine Hyperebene

$$H := \{x \in \mathbb{R}^n : a^T x = \beta\}$$

bzw. einen *Gomory-Schnitt* mit den Eigenschaften:

(a) Es ist

$$\{x \in \mathbb{Z}^n : x \geq 0,\ Ax = b\} \subset H^- := \{x \in \mathbb{R}^n : a^T x \leq \beta\},$$

d. h. die Menge der für (ILP) zulässigen Gitterpunkte liegt im nichtpositiven Halbraum, der von der Hyperebene H erzeugt ist.

(b) Es ist $x^* \notin H^-$ bzw. $a^T x^* > \beta$.

3. Dann ist das Ausgangsproblem äquivalent zu

$$\begin{cases} \text{Minimiere} \quad \begin{pmatrix} c \\ 0 \end{pmatrix}^T \begin{pmatrix} x \\ x_{n+1} \end{pmatrix} \quad \text{unter den Nebenbedingungen} \\ \begin{pmatrix} x \\ x_{n+1} \end{pmatrix} \geq 0, \quad \begin{pmatrix} A & 0 \\ a^T & 1 \end{pmatrix} \begin{pmatrix} x \\ x_{n+1} \end{pmatrix} = \begin{pmatrix} b \\ \beta \end{pmatrix}, \quad (x, x_{n+1}) \in \mathbb{Z}^n \times \mathbb{Z}. \end{cases}$$

Zu diesem Problem, welches natürlich x^* nicht wieder als Lösung besitzen kann, kann wieder die entsprechende Relaxation gebildet und gelöst werden.

Die Lösung einer linearen ganzzahligen Optimierungsaufgabe mittels eines Schnittebenenverfahrens wird also auf die Lösung einer Folge linearer Programme ohne Ganzzahligkeitsforderung zurückgeführt. Die einzelnen Schnittebenenverfahren unterscheiden sich in der Wahl der Gomory-Schnitte.

Hier wollen wir zunächst nur den sogenannten *fundamentalen Schnitt* (f-cut) herleiten und diesen an einem Beispiel studieren. Auf die Konvergenz (d. h. den Abbruch mit einer optimalen Lösung nach endlich vielen Schritten) des hierdurch entstehenden ersten Algorithmus von Gomory wollen wir nicht eingehen.

Gegeben sei also das ganzzahlige lineare Programm

$$\text{(ILP)} \qquad \begin{cases} \text{Minimiere} \quad c^T x \quad \text{unter den Nebenbedingungen} \\ x \geq 0, \quad Ax = b, \quad x \in \mathbb{Z}^n. \end{cases}$$

Hierbei seien $A \in \mathbb{Z}^{m \times n}$ mit Rang $(A) = m$, $b \in \mathbb{Z}^m$ und (gewöhnlich auch) $c \in \mathbb{Z}^n$ gegeben. Ferner sei (x, B) eine optimale Basislösung des relaxierten Programms

$$\text{(LP)} \qquad \begin{cases} \text{Minimiere} \quad c^T x \quad \text{unter den Nebenbedingungen} \\ x \geq 0, \quad Ax = b. \end{cases}$$

Wie üblich werden mit $N := \{1, \ldots, n\} \setminus B$ die Nichtbasisindizes bezeichnet. Der Basisanteil von x ist natürlich $x_B = A_B^{-1} b$, während der Nichtbasisanteil x_N verschwindet. Wir gehen davon aus, dass $x_B \notin \mathbb{Z}^m$, denn andernfalls ist x eine Lösung von (ILP). Weiter sei $B = \{j(1), \ldots, j(m)\}$.

Wir benutzen die folgende Bezeichnung. Ist $\alpha \in \mathbb{R}$, so sei $\lfloor \alpha \rfloor$ die größte ganze Zahl mit $\lfloor \alpha \rfloor \leq \alpha$. Dann ist $0 \leq \alpha - \lfloor \alpha \rfloor < 1$. Ist a ein Vektor, so ist $\lfloor a \rfloor$ entsprechend komponentenweise zu verstehen. Das gleiche gilt für eine Matrix.

Sei nun $z \in \mathbb{Z}^n$ zulässig für (ILP). Wegen $Az = b$ ist dann

$$\begin{aligned} z_B &= x_B - A_B^{-1} A_N z_N \\ &= \lfloor x_B \rfloor + (x_B - \lfloor x_B \rfloor) - [\lfloor A_B^{-1} A_N \rfloor + (A_B^{-1} A_N - \lfloor A_B^{-1} A_N \rfloor)] z_N \end{aligned}$$

und folglich

$$z_B - \lfloor x_B \rfloor + \lfloor A_B^{-1} A_N \rfloor z_N = \underbrace{x_B - \lfloor x_B \rfloor}_{<e} - \underbrace{(A_B^{-1} A_N - \lfloor A_B^{-1} A_N \rfloor) z_N}_{\geq 0} \in \mathbb{Z}^m.$$

Daher gilt für jedes z, das für das ganzzahlige lineare Programm (ILP) zulässig ist, daß

$$x_B - \lfloor x_B \rfloor - (A_B^{-1} A_N - \lfloor A_B^{-1} A_N \rfloor) z_N \leq 0.$$

Da wir davon ausgehen, dass x nicht ganzzahlig ist, ist $x_B - \lfloor x_B \rfloor \neq 0$ bzw. mindestens eine Komponente von $x_B - \lfloor x_B \rfloor$ positiv. Sei etwa die i-te Komponente von x_B nicht ganzzahlig, also $x_{j(i)} - \lfloor x_{j(i)} \rfloor > 0$. Dann ist

$$-e_i^T (A_B^{-1} A_N - \lfloor A_B^{-1} A_N \rfloor) z_N \leq -(x_{j(i)} - \lfloor x_{j(i)} \rfloor)$$

eine Restriktion, der jeder für (ILP) zulässige Vektor z, nicht aber eine optimale Basislösung des relaxierten Problems (LP) genügt. Damit sind die gewünschten Schnitte gefunden.

Wir haben in Unterabschnitt 2.2.4 geschildert, wie das duale Simplexverfahren eingesetzt werden kann, wenn nachträglich zusätzliche Restriktionen, wie hier etwa ein Schnitt, zu berücksichtigen sind.

Beispiel: Betrachte ein ganzzahliges lineares Programm mit den Daten

$$\begin{array}{|c|c|} \hline c^T & \\ \hline A & b \\ \hline \end{array} = \begin{array}{|rrrrr|r|} \hline -2 & -1 & 0 & 0 & 0 & \\ \hline 1 & 1 & 1 & 0 & 0 & 5 \\ -1 & 1 & 0 & 1 & 0 & 0 \\ 6 & 2 & 0 & 0 & 1 & 21 \\ \hline \end{array}$$

Zur Lösung des relaxierten Problems wenden wir das (primale) revidierte Simplexverfahren an und erhalten Tableaus der Form

$$\begin{array}{|c|c|c|} \hline & y^T & c_0 \\ \hline B & A_B^{-1} & x_B \\ \hline \end{array} \qquad \begin{array}{|c|} \hline N \\ \hline \overline{c}_N^T \\ \hline \end{array},$$

wobei die beim revidierten Simplexverfahren eingeführten Bezeichnungen benutzt wurden. Also ist z. B. $y := A_B^{-T} c_B$ und $\overline{c}_N := c_N - A_N^T y$. Nach zwei Schritten erhält man das optimale Tableau (die reduzierten Kosten sind nichtnegativ):

$$\begin{array}{|c|rrr|r|} \hline & -\frac{1}{2} & 0 & -\frac{1}{4} & -\frac{31}{4} \\ \hline 2 & \frac{3}{2} & 0 & -\frac{1}{4} & \frac{9}{4} \\ 4 & -2 & 1 & \frac{1}{2} & \frac{1}{2} \\ 1 & -\frac{1}{2} & 0 & \frac{1}{4} & \frac{11}{4} \\ \hline \end{array} \qquad \begin{array}{|cc|} \hline 3 & 5 \\ \hline \frac{1}{2} & \frac{1}{4} \\ \hline \end{array}.$$

Durch

$$x := (\tfrac{11}{4}, \tfrac{9}{4}, 0, \tfrac{1}{2}, 0)^T, \qquad B := \{2, 4, 1\}$$

hat man eine optimale Basislösung des relaxierten Problems gewonnen. Es ist

$$x_B - \lfloor x_B \rfloor = \begin{pmatrix} \frac{1}{4} \\ \frac{1}{2} \\ \frac{3}{4} \end{pmatrix}$$

und

$$A_B^{-1} A_N = \begin{pmatrix} \frac{3}{2} & 0 & -\frac{1}{4} \\ -2 & 1 & \frac{1}{2} \\ -\frac{1}{2} & 0 & \frac{1}{4} \end{pmatrix} \begin{pmatrix} 1 & 0 \\ 0 & 0 \\ 0 & 1 \end{pmatrix} = \begin{pmatrix} \frac{3}{2} & -\frac{1}{4} \\ -2 & \frac{1}{2} \\ -\frac{1}{2} & \frac{1}{4} \end{pmatrix}.$$

Daher ist

$$A_B^{-1} A_N - \lfloor A_B^{-1} A_N \rfloor = \begin{pmatrix} \frac{1}{2} & \frac{3}{4} \\ 0 & \frac{1}{2} \\ \frac{1}{2} & \frac{1}{4} \end{pmatrix}.$$

Mögliche Schnitte sind daher

$$\begin{array}{rcl} -\frac{1}{2} z_3 - \frac{3}{4} z_5 & \leq & -\frac{1}{4}, \\ -\frac{1}{2} z_5 & \leq & -\frac{1}{2}, \\ -\frac{1}{2} z_3 - \frac{1}{4} z_5 & \leq & -\frac{3}{4}. \end{array}$$

Wir wählen den zweiten Schnitt und haben jetzt eine Optimierungsaufgabe in Normalform mit den Daten

$$\begin{array}{|c|c|} \hline \hat{c}^T & \\ \hline \hat{A} & \hat{b} \\ \hline \end{array} = \begin{array}{|cccccc|c|} \hline -2 & -1 & 0 & 0 & 0 & 0 & \\ \hline 1 & 1 & 1 & 0 & 0 & 0 & 5 \\ -1 & 1 & 0 & 1 & 0 & 0 & 0 \\ 6 & 2 & 0 & 0 & 1 & 0 & 21 \\ 0 & 0 & 0 & 0 & -\frac{1}{2} & 1 & -\frac{1}{2} \\ \hline \end{array}$$

zu lösen. Wie beim dualen Simplexverfahren erläutert wurde, kann das optimale letzte Tableau hier sinnvoll eingesetzt werden. Wir hatten uns überlegt, dass durch

$$\hat{x} = (\tfrac{11}{4}, \tfrac{9}{4}, 0, \tfrac{1}{2}, 0, -\tfrac{1}{2})^T, \qquad \hat{B} = \{2, 4, 1, 6\}$$

eine Basislösung für das neue Problem gegeben ist. Das Starttableau für die Anwendung des dualen Simplexverfahrens ist

$$\begin{array}{|c|c|c|} \hline & \hat{y} & \hat{c}_0 \\ \hline \hat{B} & \hat{A}_{\hat{B}}^{-1} & \hat{x}_{\hat{B}} \\ \hline \end{array} = \begin{array}{|c|cccc|c|} \hline & -\frac{1}{2} & 0 & -\frac{1}{4} & 0 & -\frac{31}{4} \\ \hline 2 & \frac{3}{2} & 0 & -\frac{1}{4} & 0 & \frac{9}{4} \\ 4 & -2 & 1 & \frac{1}{2} & 0 & \frac{1}{2} \\ 1 & -\frac{1}{2} & 0 & \frac{1}{4} & 0 & \frac{11}{4} \\ 6 & 0 & 0 & 0 & 1 & -\frac{1}{2} \\ \hline \end{array} \qquad \begin{array}{|c|} \hline N \\ \hline \overline{c}_N^T \\ \hline \end{array} = \begin{array}{|cc|} \hline 3 & 5 \\ \hline \frac{1}{2} & \frac{1}{4} \\ \hline \end{array}.$$

Im nächsten Schritt erhält man das optimale Tableau

	$-\frac{1}{2}$	0	$-\frac{1}{4}$	$-\frac{1}{2}$	$-\frac{15}{2}$
2	$\frac{3}{2}$	0	$-\frac{1}{4}$	$-\frac{1}{2}$	$\frac{5}{2}$
4	-2	1	$\frac{1}{2}$	1	0
1	$-\frac{1}{2}$	0	$\frac{1}{4}$	$\frac{1}{2}$	$\frac{5}{2}$
5	0	0	0	-2	1

3	6
$\frac{1}{2}$	$\frac{1}{2}$

Die beiden jetzt noch möglichen Schnitte fallen zusammen und ergeben

$$-\tfrac{1}{2}z_3 - \tfrac{1}{2}z_6 \leq -\tfrac{1}{2}.$$

Die zugehörige Relaxation hat eine ganzzahlige Lösung, nämlich

$$x = (3, 1, 1, 2, 1)^T,$$

dies ist die Lösung der gegebenen ganzzahligen Optimierungsaufgabe. □

Spätestens dann, wenn man, wie wir gerade eben, ein Beispiel gerechnet hat, stellen sich die folgenden Fragen:

- Wie sollte man den jeweiligen fundamentalen Schnitt auswählen? Wie viele muss man maximal zu den Restriktionen hinzufügen? Man beachte hierbei, dass jeder Schnitt die Anzahl der Variablen und der Restriktionen um eine erhöht.

- Kann, wenigstens unter gewissen Auswahlregeln, die Endlichkeit eines Schnittebenenverfahrens bewiesen werden?

- Am obigen Beispiel erkennt man: Selbst wenn das Ausgangstableau ganzzahlig ist, so sind die weiteren Tableaus i. allg. nicht ganzzahlig, da das Pivotelement bei der Durchführung des dualen Simplexverfahrens i. allg. nicht -1 ist. Dies muss zwangsläufig zu numerischen Schwierigkeiten führen. Es wäre daher wünschenswert, einen Algorithmus zu entwickeln, der mit ganzzahligen Tableaus arbeitet.

5.2.2 Rein ganzzahlige Schnittebenenverfahren

Das im vorigen Abschnitt beschriebene Schnittebenenverfahren (Gomory I oder fractional dual algorithm) hat zwei Nachteile:

- Selbst bei ganzzahligen Daten A, b und c sind die auftretenden Tableaus i. allg. nicht ganzzahlig, was zwangsläufig zu Schwierigkeiten führen muß.

- Da mit dem dualen Simplexverfahren gerechnet wird, hat man keine zulässige Näherungslösung, wenn man das Verfahren vorzeitig abbricht.

Wir beschreiben nun zwei weitere Schnittebenenverfahren, die beide mit ganzzahligen Tableaus arbeiten, und von denen das erste ein duales und das zweite sogar ein primales Verfahren ist.

Gegeben sei jeweils das rein ganzzahlige Problem

(ILP) Minimiere $c^T x$ unter den Nebenbedingungen $x \in \mathbb{Z}^n$, $x \geq 0$, $Ax = b$,

wobei $A \in \mathbb{Z}^{m\times n}$ mit Rang $(A) = m$, $b \in \mathbb{Z}^m$ und $c \in \mathbb{Z}^n$ gegeben sind.

Es wird angenommen, $B \subset \{1, \ldots, n\}$ sei eine m-elementige Menge, für die A_B nichtsingulär und A_B^{-1} ganzzahlig ist. Dies impliziert wegen der Ganzzahligkeit von b, dass (x, B) mit $x_B = A_B^{-1}b$ und $x_N = 0$ (mit $N := \{1, \ldots, n\} \setminus B$) eine ganzzahlige Basislösung ist. Ferner sei $y := A_B^{-T} c_B$ dual zulässig, also $\overline{c}_N := c_N - A_N^T y \geq 0$. Ist $x_B = A_B^{-1}b \geq 0$, so hat man eine gesuchte Lösung von (ILP) gefunden. Andernfalls bestimme man ein $r \in \{1, \ldots, m\}$ mit $(x_B)_r = (A_B^{-1}b)_r < 0$, etwa durch

$$(A_B^{-1}b)_r = \min_{i=1,\ldots,m} (A_B^{-1}b)_i.$$

Sei nun z zulässig für das ganzzahlige lineare Programm (ILP). Dann ist

$$0 \leq z_B = x_B - A_B^{-1}A_N z_N,$$

mit einem $h \in (0, 1]$ ist daher

$$0 \leq hz_B = hx_B - \lfloor hA_B^{-1}A_N \rfloor z_N - \underbrace{[hA_B^{-1}A_N - \lfloor hA_B^{-1}A_N \rfloor]}_{\geq 0} \underbrace{z_N}_{\geq 0}$$

und daher

$$\lfloor hA_B^{-1}A_N \rfloor z_N \leq hx_B.$$

In der letzten Ungleichung steht links eine ganze Zahl, so dass für ein für (ILP) zulässiges z sogar die Ungleichung

$$\lfloor hA_B^{-1}A_N \rfloor z_N \leq \lfloor hx_B \rfloor$$

gilt. Betrachten wir hier die r-te Ungleichung, so ist durch

$$(*) \qquad \lfloor he_r^T A_B^{-1}A_N \rfloor z_N \leq \lfloor h(x_B)_r \rfloor$$

ein möglicher Schnitt gefunden (man beachte, dass x dieser Ungleichung *nicht* genügt, da $x_N = 0$ und $(x_B)_r < 0$). Wir werden versuchen, durch geeignete Wahl von h zu erreichen, dass dieser Schnitt als Pivotzeile genommen werden kann. Hierzu müssen wir dafür sorgen, dass das Pivotelement $\lfloor he_r^T A_B^{-1}a_s \rfloor$ bei geeigneter Wahl von $s \in N$ und $h \in (0, 1]$ gleich -1 ist und ferner die duale Zulässigkeit gewahrt bleibt.

Wie beim dualen Simplexverfahren berechnen wir (den ganzzahligen Vektor) $v_N := A_N^T A_B^{-T} e_r$ (man beachte, dass sich der Schnitt $(*)$ dann in der Form $\lfloor hv_N \rfloor^T z_N \leq \lfloor h(x_B)_r \rfloor$ schreiben lässt) und setzen

$$N_r := \{j \in N : v_j < 0\}.$$

Ist $N_r = \emptyset$, so ist die Relaxation zum ganzzahligen linearen Programm (ILP) nicht zulässig, erst recht also (ILP). Mit einer entsprechenden Meldung sollte dann ausgestiegen werden. Andernfalls bestimme man $s \in N_r$ so, dass

$$\overline{c}_s = \min_{j \in N_r} \overline{c}_j.$$

Wir machen eine Fallunterscheidung.

- Es ist $\overline{c}_s = 0$. Dann ist
$$0 = \frac{\overline{c}_s}{v_s} = \max_{j \in N_r} \frac{\overline{c}_j}{v_j},$$
die Wahl von s entspricht also der im dualen Simplexverfahren.

Definiere $h := -1/v_s$. Da $v_s < 0$ und folglich $v_s \leq -1$, ist $0 < h \leq 1$. Ist $h = 1$, so ist $v_s = -1$. Ohne den zusätzlichen Schnitt $(*)$ kann mit der Pivotzeile r und der Pivotspalte s ein Schritt des dualen Simplexverfahrens durchgeführt werden. Mit $w := A_B^{-1} a_s$ ist $w_r = e_r^T A_B^{-1} a_s = v_s = -1$, so dass die Ganzzahligkeit des neuen Tableaus (und der neuen Basislösung) erhalten bleibt. Ist dagegen $h < 1$, so benutze man den zusätzlichen Schnitt $(*)$ und wende auf das entsprechend erweiterte Tableau einen Schritt des dualen Simplexverfahrens mit der letzten Zeile (also der Schnitt-Restriktion) als Pivotzeile und s als Pivotspalte an.

- Es ist $\overline{c}_s > 0$, also $0 < \overline{c}_s \leq \overline{c}_j$ und daher $\lfloor \overline{c}_j / \overline{c}_s \rfloor \geq 1$ für alle $j \in N_r$.

Wir setzen

$$h := \min_{j \in N_r} \Big(-\frac{1}{v_j} \Big\lfloor \frac{\overline{c}_j}{\overline{c}_s} \Big\rfloor \Big).$$

Dann ist $0 < h \leq -1/v_s \leq 1$, insbesondere daher $-1 \leq h v_s < 0$ und folglich $\lfloor h v_s \rfloor = -1$. Ist $h = 1$, so ist $v_s = -1$. Wieder kann in diesem Falle ohne den zusätzlichen Schnitt $(*)$ mit der Pivotzeile r und der Pivotspalte s ein Schritt des dualen Simplexverfahrens durchgeführt werden. Ferner entspricht in diesem Falle die Wahl der Pivotspalte wieder der des dualen Simplexverfahrens. Denn aus $h = 1$ folgt $(-1/v_j) \lfloor \overline{c}_j / \overline{c}_s \rfloor \geq 1$ für alle $j \in N_r$ bzw. (benutze, dass $v_s = -1$)

$$-v_j \leq \Big\lfloor \frac{\overline{c}_j}{\overline{c}_s} \Big\rfloor \leq \frac{\overline{c}_j}{\overline{c}_s} = -v_s \frac{\overline{c}_j}{\overline{c}_s} \qquad \text{für alle } j \in N_r.$$

Daher ist

$$\frac{\overline{c}_s}{v_s} = \max_{j \in N_r} \frac{\overline{c}_j}{v_j},$$

d. h. auch hier genügt die Pivotspalte s der entsprechenden Forderung im dualen Simplexverfahren, insbesondere ist die duale Zulässigkeit im nächsten Schritt gesichert. Es bleibt der Fall zu untersuchen, dass $h < 1$. Wir haben zu zeigen, dass wir die dem Schnitt $(*)$ entsprechende Zeile als Pivotzeile nehmen können und damit die duale Zulässigkeit gewahrt bleibt. Genauer haben wir nachzuweisen, dass

$$-\overline{c}_s = \frac{\overline{c}_s}{\lfloor h v_s \rfloor} = \max_{j \in N_r} \frac{\overline{c}_j}{\lfloor h v_j \rfloor}.$$

Aus der Definition von h folgt $h \le -1/v_j \lfloor \overline{c}_j/\overline{c}_s \rfloor$, $j \in N_r$, bzw. $hv_j \ge -\lfloor \overline{c}_j/\overline{c}_s \rfloor$, $j \in N_r$. Weil hier rechts eine ganze Zahl steht, ist auch

$$\lfloor hv_j \rfloor \ge -\Big\lfloor \frac{\overline{c}_j}{\overline{c}_s} \Big\rfloor \ge -\frac{\overline{c}_j}{\overline{c}_s} = \lfloor hv_s \rfloor \frac{\overline{c}_j}{\overline{c}_s}, \qquad j \in N_r.$$

Hieraus wiederum folgt (für $j \in N_r$ ist $hv_j < 0$ und daher $\lfloor hv_j \rfloor \le -1$, ferner ist $c_j > 0$)

$$\frac{\overline{c}_s}{\lfloor hv_s \rfloor} = \max_{j \in N_r} \frac{\overline{c}_j}{\lfloor hv_j \rfloor},$$

das war zu zeigen.

Durch obige Betrachtungen ist die Durchführbarkeit des Verfahrens (auch Gomory II oder duales, rein ganzzahliges Verfahren genannt) gesichert.

Nun kommen wir zu einem rein ganzzahligen, primalen Verfahren, das wir nur sehr kurz beschreiben. Wir gehen von einem rein ganzzahligen linearen Programm in Standardform aus, also

(ILP) Minimiere $c^T x$ unter den Nebenbedingungen $x \in \mathbb{Z}^n$, $x \ge 0$, $Ax = b$,

wobei wieder $A \in \mathbb{Z}^{m \times n}$ mit Rang $(A) = m$, $b \in \mathbb{Z}^m$ und $c \in \mathbb{Z}^n$ gegeben sind. Diesmal wird wie eben angenommen, $B \subset \{1, \ldots, n\}$ sei eine m-elementige Menge, für die A_B nichtsingulär und A_B^{-1} ganzzahlig ist, weiter sei die zugehörige Basislösung x, also $x_B = A_B^{-1} b$, $x_N = 0$, sogar zulässig bzw. $A_B^{-1} b \ge 0$.

Wie beim primalen Simplexverfahren berechne man $y := A_B^{-T} c_B$ und anschließend die reduzierten Kosten $\overline{c}_N := c_N - A_N^T y$. Ist $\overline{c}_N \ge 0$, so ist x Lösung des relaxierten linearen Programms, erst recht also des ganzzahligen linearen Programms (ILP). Andernfalls wähle man wie beim gewöhnlichen primalen Simplexverfahren ein $s \in N$ mit $\overline{c}_s < 0$, etwa mit $\overline{c}_s = \min_{j \in N} \overline{c}_j$. Anschließend berechne man den ganzzahligen Vektor $w := A_B^{-1} a_s$. Ist $w \le 0$, so ist $\inf(\mathrm{ILP}) = -\infty$ (Beweis? Man sehe sich den entsprechenden Beweis beim Simplexverfahren an und benutze, dass w ganzzahlig ist). Wie beim normalen primalen Simplexverfahren bestimmt man die potentielle Pivotzeile r so, dass

$$\frac{(x_B)_r}{w_r} = \min_{i \in B_s} \frac{(x_B)_i}{w_i} \qquad \text{mit} \quad B_s := \{i \in \{1, \ldots, m\} : w_i > 0\}.$$

Da w ein ganzzahliger Vektor ist, ist natürlich $B_s = \{i \in \{1, \ldots, m\} : w_i \ge 1\}$. Ist $w_r = 1$, so wird man einen ganz normalen Simplexschritt ausführen und das Verfahren fortsetzen. Andernfalls sei $h := 1/w_r$, also $0 < h < 1$. Für ein für (ILP) zulässiges z ist $z_B = x_B - A_B^{-1} A_N z_N$, nach Multiplikation mit h ist also bei Betrachtung der r-ten Komponente

$$0 \le (hz_B)_r = h(x_B)_r - he_r^T A_B^{-1} A_N z_N \le h(x_B)_r - \lfloor he_r^T A_B^{-1} A_N \rfloor z_N$$

und folglich

$$\lfloor he_r^T A_B^{-1} A_N \rfloor z_N \le h(x_B)_r.$$

Links steht eine ganze Zahl, so dass ein für (ILP) zulässiges z sogar der Ungleichung

$$\lfloor he_r^T A_B^{-1} A_N \rfloor z_N \le \lfloor h(x_B)_r \rfloor \qquad \text{bzw.} \qquad \Big\lfloor \frac{e_r^T A_B^{-1} A_N}{w_r} \Big\rfloor z_N \le \Big\lfloor \frac{(x_B)_r}{w_r} \Big\rfloor$$

genügt. Diese Ungleichung füge man als letzte Zeile dem Tableau hinzu und nehme sie als Pivotzeile. Mit s als Pivotspalte mache man also einen Basisschritt.

5.3 Das Traveling Salesman Problem

5.3.1 Heuristiken für das TSP

Wir zitieren W. J. Cook et al. (1998, S. 242):

- *Heuristics* are methods which cannot guarenteed to produce optimal solutions, but which, we hope, produce fairly good solutions at least some of the time. For the TSP, there are two different types of heuristics. The first attempts to construct a "good" tour from scratch. The second tries to improve an existing tour, by means of "local" improvements. In practice it seems very difficult to get a really good tour construction method. It is the second type of method, in particular, an algorithm developed by Lin and Kernighan [1973], which usually results in the best solutions and forms the basis of the most effective computer codes.

Einige Heuristiken zur Konstruktion einer suboptimalen Tour haben wir früher schon erwähnt:

1. Nächster Nachbar.

   Beginne bei irgendeinem Knoten. Besuche jeweils den nächsten noch nicht besuchten Knoten und kehre zum Startknoten zurück, wenn alle anderen Knoten besucht sind.

2. Minimum Spanning Tree.

   Die Vorgehensweise bei dieser Heuristik besteht aus den folgenden Schritten.

   - Zu dem bewerteten vollständigen Graphen K_n bestimme man (z. B. mit dem Verfahren von Kruskal) einen minimalen aufspannenden Baum T.
   - Verdopple alle Kanten in T. Hierdurch erhält man einen Eulerschen Multigraphen T_D. Sei C ein Euler-Zug in T_D.
   - In C ist ein Hamilton-Kreis enthalten, den man durch Überspringen schon durchlaufener Ecken erhalten kann.

3. Christofides.

   Hier lauten die Schritte:

   - Zu dem bewerteten vollständigen Graphen K_n bestimme man (z. B. mit dem Verfahren von Kruskal) einen minimalen aufspannenden Baum T.
   - Sei U die Menge der Ecken ungeraden Grades in T. Wegen des Handshaking Lemmas 1.1 ist $|U| = 2m$ gerade. Zu dieser Eckenmenge bilde man den vollständigen (bewerteten) Graphen (mit $m(2m-1)$ Kanten). Unter einem *Maximum Matching* M in einem Graphen verstehen wir eine Menge maximal vieler nicht-inzidenter (also keine Ecken gemein habender) Kanten. Wegen $|U| = 2m$ besteht ein Maximum Matching hier aus m Kanten. Unter allen Maximum Matchings bestimme man eines mit minimalen Kosten.

Diese Kanten M füge man zu T hinzu (ist eine Kante schon in T enthalten, so erhält man eine Mehrfachkante). Man erhält einen Eulerschen Multigraphen T_D (denn der Graph ist zusammenhängend und jede Ecke hat geraden Grad). Sei C ein Euler-Zug in T_D.

- In C ist ein Hamilton-Kreis enthalten, den man durch Überspringen schon durchlaufener Ecken erhalten kann.

Mit T bezeichnen wir eine zulässige Tour des Handlungsreisenden, mit $c(T)$ ihre Kosten. Hat man allgemeiner einen Untergraphen des K_n, so werden dessen Kosten als die Summe der Kantenkosten definiert.

Es gilt die folgende Aussage.

Satz 1.1 *Die Kostenmatrix $C = (c_{ij}) \in \mathbb{R}^n$ im Traveling Salesman Problem sei symmetrisch, nichtnegativ (also $c_{ij} \geq 0$ für alle $i, j \in \{1, \ldots, n\}$), ihre Einträge mögen ferner der Dreiecksungleichung genügen (d. h. es sei $c_{ik} \leq c_{ij} + c_{jk}$ für alle $i, j, k \in \{1, \ldots, n\}$). Ist dann T_{opt} eine optimale Tour, T_{MST} eine nach der Minimum Spanning Tree Heuristik und T_{CH} eine nach der Christofides Heuristik bestimmte Tour, so gilt:*

1. *Es ist $c(T_{\text{MST}}) \leq 2c(T_{\text{opt}})$.*
2. *Es ist $c(T_{\text{CH}}) \leq \frac{3}{2}c(T_{\text{opt}})$.*

Beweis: Entfernt man aus dem optimalen Hamilton-Kreis T_{opt} eine Kante, so erhält man einen den vollständigen Graphen K_n aufspannenden Baum. Da die Kosten dieser weggelassenen Kante nichtnegativ ist, gilt für den in der ersten Phase der Minimum Spanning Tree Heuristik bestimmten Baum T, dass $c(T) \leq c(T_{\text{opt}})$. Durch Verdopplung aller Kanten in T erhält man den Eulerschen Multigraphen T_D, die Kosten sind $c(T_D) = 2c(T) \leq 2c(T_{\text{opt}})$. Sei $C = \{1, 2, \ldots\}$ ein Euler-Zug in T_D (wir beschreiben C durch die benachbarten Ecken), also ein geschlossener Kantenzug, der alle $2(n-1)$ Kanten von T_D genau einmal enthält. Sei weiter $T_{\text{MST}} = \{i_1, i_2, \ldots, i_n\}$ mit $i_1 < i_2 < \cdots < i_n$ der in C enthaltene Hamilton-Kreis. Die Kosten dieses Hamilton-Kreises sind gegeben durch

$$c(T_{\text{MST}}) = \sum_{k=1}^{n-1} c_{i_k i_{k+1}}.$$

Diese kann man wegen der Dreiecksungleichung abschätzen durch

$$\begin{aligned} c(T_{\text{MST}}) &\leq c_{i_1,i_1+1} + \cdots + c_{i_2-1,i_2} \\ &\quad + c_{i_2,i_2+1} + \cdots + c_{i_3-1,i_3} \\ &\quad + \cdots \\ &\quad + c_{i_{n-1},i_{n-1}+1} + \cdots + c_{i_n-1,i_n} \\ &\leq c(T_D) \\ &\leq 2c(T_{\text{opt}}). \end{aligned}$$

Damit ist der erste Teil des Satzes bewiesen.

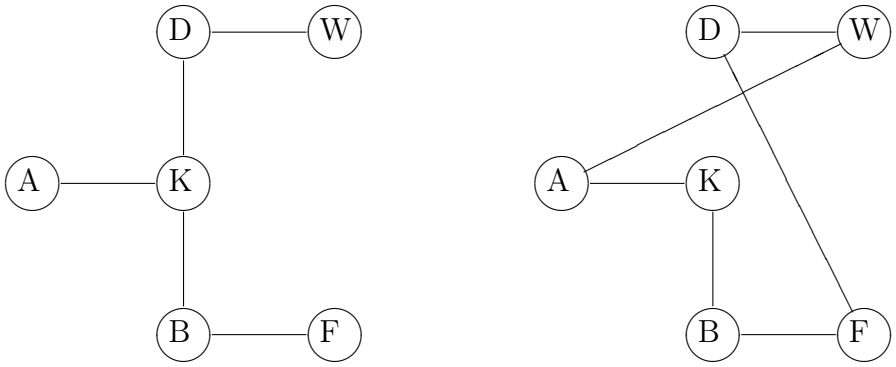


Abbildung 5.1: Minimum Spanning Tree Heuristik

Beispiel: Der obige Beweis ist eigentlich nur wegen der Bezeichnungsschwierigkeiten nicht sofort zu verstehen. Wir geben hier Abbildung 3.25 in Abschnitt 3.6 wieder, siehe Abbildung 5.1. Der Euler-Zug im "doppelten" minimalen aufspannenden Baum T_D ist z. B. (wieder geben wir nur die Ecken an, die Kanten ergeben sich hieraus)

$$C = \{K, B, F, B, K, D, W, D, K, A, K\},$$

seine Kosten sind

$$c(C) = c_{KB} + c_{BF} + c_{FB} + c_{BK} + c_{KD} + c_{DW} + c_{WD} + c_{DK} + c_{KA} + c_{AK}$$

Einen Hamilton-Kreis T_{MST} in T_D erhält man aus C, indem man schon besuchte Städte überspringt:

$$T_{\text{MST}} = \{K, B, F, D, W, A, K\}.$$

Seine Kosten sind

$$\begin{aligned} c(T_{\text{MST}}) &= c_{KB} + c_{BF} + c_{FD} + c_{DW} + c_{WA} + c_{AK} \\ &\leq c_{KB} + c_{BF} + \underbrace{c_{FB} + c_{BK} + c_{KD}}_{\geq c_{FD}} + c_{DW} \\ &\qquad + \underbrace{c_{WD} + c_{DK} + c_{KA}}_{\geq c_{WA}} + c_{AK} \\ &= c(C) \\ &= c(T_D) \\ &= 2c(T) \\ &\leq 2c(T_{\text{opt}}), \end{aligned}$$

wobei wir bei der Abschätzung die Dreiecksungleichung benutzt haben. In diesem Beispiel haben wir damit den obigen Beweis nachvollzogen. □

Nun zum zweiten Teil des Satzes und der entsprechenden Abschätzung für die Christofides-Heuristik. Wieder ist $c(T) \leq c(T_{\text{opt}})$. Die Knoten im Graphen seien o. B. d. A. so nummeriert, dass $T_{\text{opt}} = \{1, \ldots, n\}$ eine optimale Tour ist. Seien $i_1, i_2, \ldots, i_{2m}$ die Ecken von U (Menge der Ecken ungeraden Grades in T) in dieser Reihenfolge, also $i_1 < i_2 < \cdots < i_{2m}$. Wir definieren die Maximum-Matchings

$$M_1 := \{(i_1, i_2), (i_3, i_4), \ldots, (i_{2m-1}, i_{2m})\}, \qquad M_2 := \{(i_2, i_3), (i_4, i_5), \ldots, (i_{2m}, i_1)\}$$

in dem von der Eckenmenge U erzeugten vollständigen Graphen, ferner sei M unter allen Maximum-Matchings in diesem Graphen eines mit minimalen Kosten. Dann ist $c(M) \leq c(M_1)$ und $c(M) \leq c(M_2)$. Wegen der Dreiecksungleichung ist

$$\begin{aligned} c(M_1) + c(M_2) &= \sum_{k=1}^{m} c_{i_{2k-1},i_{2k}} + \sum_{k=1}^{m-1} c_{i_{2k},i_{2k+1}} + c_{i_{2m},i_1} \\ &= c_{i_1,i_2} + c_{i_2,i_3} + \cdots + c_{i_{2m-1},i_{2m}} + c_{i_{2m},i_1} \\ &\leq \sum_{i=1}^{n-1} c_{i,i+1} + c_{n,1} \\ &= c(H_{\text{opt}}). \end{aligned}$$

Hieraus erhalten wir

$$c(M) \leq \frac{1}{2}[c(M_1) + c(M_2)] \leq \frac{1}{2} c(T_{\text{opt}}).$$

Da T_{CH} sich als Hamilton-Kreis in dem Eulerschen Graphen T_D ergibt, den man dadurch erhält, dass man dem aufspannenden Baum T die Kanten aus M hinzufügt, ist schließlich

$$c(T_{\text{CH}}) \leq c(T_D) = c(T) + c(M) \leq \frac{3}{2} c(T_{\text{opt}}),$$

womit die Behauptung bewiesen ist. □

Beispiel: Auch den letzten Beweisteil wollen wir durch ein früheres Beispiel (Rheinlandproblem) illustrieren. Die Kostenmatrix ist gegeben durch

	A	B	D	F	K	W
Aachen	–	91	80	259	70	121
Bonn	91	–	77	175	27	84
Düsseldorf	80	77	–	232	47	29
Frankfurt	259	175	232	–	189	236
Köln	70	27	47	189	–	55
Wuppertal	121	84	29	236	55	–

Siehe Abbildung 3.26 in Abschnitt 3.6, welche wir in Abbildung 5.2 wiederholen. Links

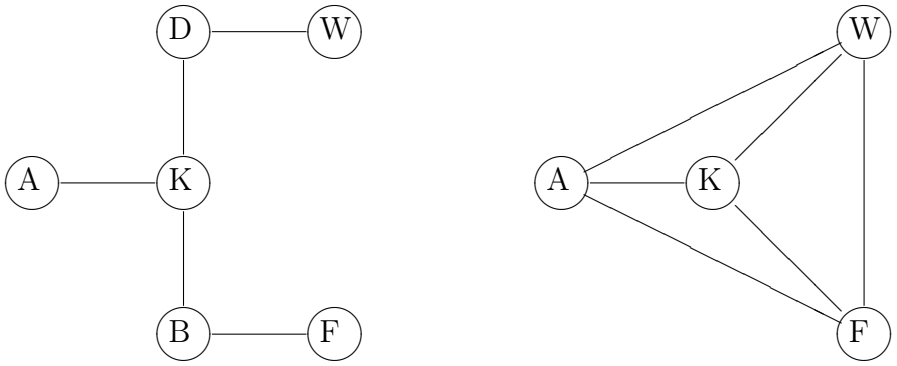


Abbildung 5.2: Christofides Heuristik I

sieht man einen minimalen aufspannenden Baum T zum vollständigen Graphen mit den Knoten $V = \{A, B, D, F, K, W\}$. Die Menge U der Knoten ungeraden Grades ist $U = \{A, F, K, W\}$, ein Maximum-Matching mit minimalen Kosten auf dem von U erzeugten vollständigen Graphen ist $M = \{AK, WF\}$ mit $c(M) = 306$, in Abbildung 5.2 rechts findet man den von U erzeugten vollständigen Graphen. In Abbildung 5.3 links findet man den Eulerschen Graphen T_D, den man erhält, indem man die Kantenmenge M

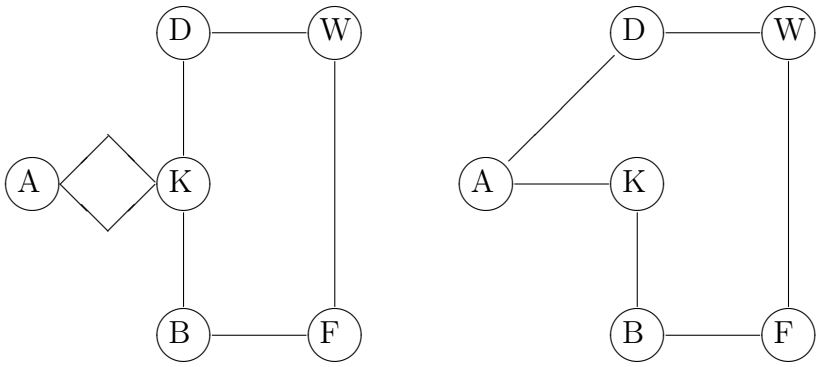


Abbildung 5.3: Christofides Heuristik II

zum Baum T hinzufügt. Durch Überspringen schon erreichter Städte erhält man den Hamilton-Kreis T_{CH} in Abbildung 5.3 rechts. Hier ist nun zufälligerweise

$$T_{\text{CH}} = T_{\text{opt}} = \{K, B, F, W, D, A\}.$$

Die Ecken U ungeraden Grades in dieser Reihenfolge sind $U = \{K, F, W, A\}$, die beiden Matchings M_1 und M_2 in dem von U erzeugten vollständigen Graphen sind daher

$$M_1 = \{KF, WA\}, \qquad M_2 = \{FW, AK\}$$

mit Kosten $c(M_1) = 310$, $c(M_2) = 306$ (hier ist $M_2 = M$ ein Maximum-Matching mit minimalen Kosten). Hier ist

$$\begin{aligned} c(M_1) + c(M_2) &= c_{KF} + c_{FW} + c_{WA} + c_{AK} \\ &\leq \underbrace{c_{KB} + c_{BF}}_{\geq c_{KF}} + c_{FW} + \underbrace{c_{WD} + c_{DA}}_{\geq c_{WA}} + c_{AK} \\ &= c(H_{\text{opt}}). \end{aligned}$$

□

Bemerkung: Umfangreiche Tests haben ergeben, dass die Christofides-Heuristik Touren liefert, deren Kosten im Mittel 1.14 mal der minimalen Kosten betragen. Die Schranke 1.5 ist die beste bekannte Schranke. Bei W. J. Cook et al. (1998, S. 245) kann man nachlesen: "It has the best worst case bound of any known method: It always produces a solution of cost at most $\frac{3}{2}$ times the optimum (assuming that the graph is complete and the costs are nonnegative and satisfy the triangle inequality)". Bei E. L. Lawler et al. (1985, S. 155) wird der erste Teil des obigen Satzes als "Folklore" bezeichnet. Ferner findet man dort Beispiele die zeigen, dass die Schranken 2 und $\frac{3}{2}$ best möglich sind. □

Es gibt einige Verfahren, um zu versuchen, eine aktuelle Tour T zu verbessern. Die einfachste wird *2-opt* genannt. Man betrachte ein Paar von nicht inzidierenden Kanten in T. Entfernt man diese beiden Kanten aus T, so zerfällt der Rest der Toor in zwei Pfade T_1 und T_2. Auf eindeutige Weise können diese beiden Pfade zu einer Tour T' zusammen gesetzt werden. Ist $c(T') < c(T)$, so ersetzt man T durch T' und wiederholt das Verfahren. Ist $c(T') \geq c(T)$ für jede Wahl von Paaren nicht inzidierender Kanten, so nennen wir T *2-optimal* und terminieren.

Beispiel: Die Nächste Nachbar Heuristik liefert beim Rheinlandproblem mit dem Startort Köln die Tour $T = \{K, B, D, W, A, F, K\}$ mit den Kosten $c(T) = 702$. Entfernen der beiden Kanten FK, AW und ersetzen durch FW, AK liefert die Tour $T' = \{K, B, D, W, F, A, K\}$ mit den Kosten $c(T') = 698$. □

Bei W. J. Cook (1998, S. 247 ff.) wird ausführlich auf raffiniertere Heuristiken eingegangen.

Kapitel 6

Lösungen zu den Aufgaben

6.1 Aufgaben zu Kapitel 2

6.1.1 Aufgaben zu Abschnitt 2.1

1. Eine nichtleere, abgeschlossene, konvexe Menge $C \subset \mathbb{R}^n$ ist der Durchschnitt aller abgeschlossenen Halbräume, die C enthalten.

   Hinweis: Man wende den starken Trennungssatz an.

   Lösung: Sei K der Durchnitt aller abgeschlossenen Halbräume, die C enthalten. Dann ist $C \subset K$. Angenommen, es existiert ein $x^* \in K \setminus C$. Wegen des starken Trennungssatzes existiert ein $y \in \mathbb{R}^n \setminus \{0\}$ mit $y^T x^* < \gamma := \inf_{x \in C} y^T x$. Dann ist $H := \{x \in \mathbb{R}^n : y^T x = \gamma\}$ eine Hyperebene, die C im zugehörigen Halbraum enthält. Dieser Halbraum enthält nicht $x^* \in K$, was ein Widerspruch zur Definition von K ist.

2. In[1] zwei Rangierbahnhöfen A und B stehen 18 bzw. 12 leere Güterwagen. In den drei Bahnhöfen R, S und T werden 11, 10 bzw. 9 Güterwagen zum Verladen von Waren benötigt. Die Distanzen in km von den Rangierbahnhöfen zu den Bahnhöfen sind durch

	R	S	T
A	5	4	9
B	7	8	10

   gegeben. Die Güterwagen sind so zu leiten, dass die totale Anzahl der durchfahrenen Leerkilometer minimal ist. Man formuliere diese Aufgabe als lineare Optimierungsaufgabe und löse sie mit einem mathematischen Anwendersystem.

   Lösung: Wir reproduzieren (ziemlich wörtlich) zunächst die Vorgehensweise bei H. R. Schwarz und behandeln danach die Aufgabe als einen Sonderfall des klassischen Transportproblems.

   Wir beachten, dass der Gesamtbedarf $(11 + 10 + 9 = 30)$ gleich dem Gesamtangebot $(18 + 12 = 30)$ ist. Als Unbekannte führen wir ein:

   $$\begin{aligned} x_1 &= \text{Anzahl der Wagen von A nach R,} \\ x_2 &= \text{Anzahl der Wagen von A nach S.} \end{aligned}$$

[1] Diese Aufgabe ist ein Beispiel in dem Lehrbuch über Numerische Mathematik von H. R. Schwarz.

Mit dieser Wahl der Unbekannten lässt sich die Zahl der Güterwagen, die auf den anderen Strecken zu fahren haben, ausdrücken. Die Anzahl der Leerkilometer ist dann gegeben durch

$$\begin{aligned} z &= 5x_1 + 4x_2 + 9(18 - x_1 - x_2) + 7(11 - x_1) + 8(10 - x_2) + 10(x_1 + x_2 - 9) \\ &= -x_1 - 3x_2 + 229, \end{aligned}$$

diese sind zu minimieren unter den Nebenbedingungen

$$x_1 + x_2 \leq 18, \quad x_1 \leq 11, \quad x_2 \leq 10, \quad x_1 + x_2 \geq 9$$

und natürlich der Nichtnegativitätsforderung

$$x_1, x_2 \geq 0.$$

Nach

```
A={{-1,-1},{-1,0},{0,-1},{1,1}};
b={-18,-11,-10,9};
c={-1,-3};
LinearProgramming[c,A,b]
```

erhält man $x_1 = 8$, $x_2 = 10$, der optimale Transportplan ist

	11	10	9
18	8	10	0
12	3	0	9

Ein bisschen weniger Nachdenken muss man, wenn man

```
A={{-1,-1,-1,0,0,0},{0,0,0,-1,-1,-1},{1,0,0,1,0,0},
{0,1,0,0,1,0},{0,0,1,0,0,1}};
b={-18,-12,11,10,9};
c={5,4,9,7,8,10};LinearProgramming[c,A,b]
```

eingibt und natürlich ebenfalls obiges Ergebnis erhält.

3. Seien α, β und γ Winkel in einem Dreieck mit $90° \geq \alpha \geq \beta \geq \gamma \geq 0$ und natürlich $\alpha + \beta + \gamma = 180°$. Unter allen solchen Winkeln bestimme man diejenigen, für die

$$g(\alpha, \beta, \gamma) := \min\{\gamma, \beta - \gamma, \alpha - \beta, 90° - \alpha\}$$

maximal ist. Hierzu formuliere man diese Aufgabe als ein lineares Programm.

Hinweis: Die obige Aufgabenstellung steht im engen Zusammenhang mit einem (sehr witzigen) Aufsatz von B. Tergan (1980), in dem gezeigt wird, dass es (bis auf Ähnlichkeit) genau ein allgemeines, spitzwinkliges Dreieck gibt, dessen Winkel durch $\alpha^* := 75°$, $\beta^* := 60°$ und $\gamma^* := 45°$ gegeben sind.

Lösung: Als lineares Programm kann man obige Aufgabe schreiben als

$$\begin{cases} \text{Maximiere} \quad \delta \quad \text{unter den Nebenbedingungen} \\ \delta \geq 0°, \quad \delta \leq \gamma, \quad \delta \leq \beta - \gamma, \quad \delta \leq \alpha - \beta, \quad \delta \leq 90° - \alpha, \quad \alpha + \beta + \gamma = 180°. \end{cases}$$

Eliminiert man hier γ durch die Gleichungsrestriktion, so erhält man das lineare Programm

$$\left\{\begin{array}{l} \text{Maximiere} \quad \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}^T \begin{pmatrix} \alpha \\ \beta \\ \delta \end{pmatrix} \quad \text{unter den Nebenbedingungen} \\ \begin{pmatrix} 0 & 0 & -1 \\ 1 & 1 & 1 \\ -1 & -2 & 1 \\ -1 & 1 & 1 \\ 1 & 0 & 1 \end{pmatrix} \begin{pmatrix} \alpha \\ \beta \\ \delta \end{pmatrix} \le \begin{pmatrix} 0 \\ 180 \\ -180 \\ 0 \\ 90 \end{pmatrix}. \end{array}\right.$$

Mit

```
A={{0,0,1},{-1,-1,-1},{1,2,-1},{1,-1,-1},{-1,0,-1}};
c={0,0,-1};b={0,-180,180,0,-90};
LinearProgramming[c,A,b]
```

erhält man als Lösung $(\alpha^*, \beta^*, \delta^*) = (75, 60, 15)$.

4. Die[2] Funktion $f(t) := \cos((\pi/2)t)$ soll im Intervall $[0,1]$ durch ein Polynom vierten Grades im Tschebyscheffschen Sinne approximiert werden bezüglich der elf äquidistanten Abszissen $t_i := (i-1)/10$, $i = 1, \ldots, 11$. Gesucht ist also eine Lösung der Aufgabe

$$\text{Minimiere} \quad f(a) := \max_{i=1,\ldots,11} \left| \sum_{j=1}^{5} a_j t_i^{j-1} - \cos((\pi/2)t_i) \right|, \quad a \in \mathbb{R}^5.$$

Man führe diese Aufgabe in eine lineare Optimierungsaufgabe über und löse sie mit einem mathematischen Anwendersystem. Anschließend plotte man den Fehler

$$d(t) := \sum_{j=1}^{5} a_j t^{j-1} - \cos((\pi/2)t)$$

über dem Intervall $[0,1]$.

Lösung: Wir setzen $T := (t_i^{j-1}) \in \mathbb{R}^{11\times 5}$ und $b := (\cos((\pi/2)t_i)) \in \mathbb{R}^{11}$. Das obige diskrete, lineare Tschebyscheffsche Approximationsproblem führt auf die lineare Optimierungsaufgabe

$$\text{Minimiere} \quad \delta \quad \text{auf} \quad M := \{(a,\delta) \in \mathbb{R}^5 \times \mathbb{R} : \delta \ge 0,\ -\delta e \le Tx - b \le \delta e\},$$

wobei $e := (1, \ldots, 1)^T \in \mathbb{R}^{11}$. Um den *Mathematica*-Befehl `LinearProgramming` anwenden zu können, schreiben wir dieses lineare Programm in die äquivalente Form

$$\left\{\begin{array}{l} \text{Minimiere} \quad \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}^T \begin{pmatrix} a_+ \\ a_- \\ \delta \end{pmatrix} \quad \text{unter den Nebenbedingungen} \\ \begin{pmatrix} -T & T & e \\ T & -T & e \end{pmatrix} \begin{pmatrix} a_+ \\ a_- \\ \delta \end{pmatrix} \ge \begin{pmatrix} -b \\ b \end{pmatrix}, \quad \begin{pmatrix} a_+ \\ a_- \\ \delta \end{pmatrix} \ge 0. \end{array}\right.$$

Nach

[2]Diese Aufgabe ist als Beispiel im Lehrbuch von H. R. Schwarz über Numerische Mathematik enthalten. Hierdurch ist es möglich, die erhaltenen Ergebnisse zu vergleichen.

```
<<LinearAlgebra‘MatrixManipulation‘
T=Table[If[j>1,((i-1)/10)^(j-1),1],{i,11},{j,5}];
e=Table[1,{i,11},{j,1}];
b=Table[Cos[(Pi/2)*(i-1)/10],{i,11}];
A=BlockMatrix[{{-T,T,e},{T,-T,e}}];
b=N[Flatten[{-b,b}]];
c=Table[If[j<11,0,1],{j,11}];
z=LinearProgramming[c,A,b];
a=Table[z[[i]]-z[[i+5]],{i,5}];
delta=z[[11]];
d[t_]:=(((a[[5]]*t+a[[4]])*t+a[[3]])*t+a[[2]])*t+a[[1]]-Cos[(Pi/2)*t];
Plot[d[t],{t,0,1}]
```

erhalten wir als Output (mit 12 Stellen)

$$a = \begin{pmatrix} 0.999896084731 \\ 0.00496010706648 \\ -1.27104452066 \\ 0.0933071199016 \\ 0.172985124229 \end{pmatrix}, \qquad \delta = 0.000103915268617, .$$

außerdem ist in Abbildung 6.1 die Fehlerfunktin d geplottet worden.

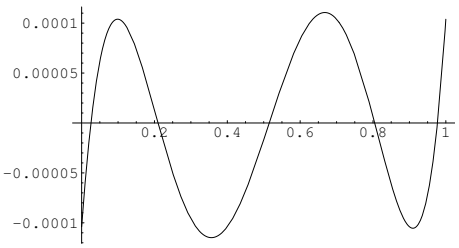


Abbildung 6.1: Fehler bei diskreter Tschebyscheff-Approximation

5. Sei $A \in \mathbb{R}^{m\times n}$. Man beweise den Alternativsatz von Gordan: Genau eine der beiden Aussagen

   (I) $\qquad Ax = 0, \quad x \geq 0, \quad x \neq 0 \qquad$ hat eine Lösung $x \in \mathbb{R}^n$

   bzw.

   (II) $\qquad A^T y > 0 \qquad$ hat eine Lösung $y \in \mathbb{R}^m$

   ist richtig.

Lösung: (I) und (II) sind nicht gleichzeitig lösbar, wie man leicht nachweist. Angenommen, (I) sei nicht lösbar. Mit $e := (1, \ldots, 1)^T \in \mathbb{R}^n$ ist auch

$$\begin{pmatrix} A \\ e^T \end{pmatrix} x = \begin{pmatrix} 0 \\ 1 \end{pmatrix}, \qquad x \geq 0$$

nicht lösbar. Aus dem Farkas-Lemma folgt die Lösbarkeit von

$$\begin{pmatrix} A^T & e \end{pmatrix} \begin{pmatrix} y \\ \delta \end{pmatrix} \geq 0, \qquad \begin{pmatrix} 0 \\ 1 \end{pmatrix}^T \begin{pmatrix} y \\ \delta \end{pmatrix} < 0.$$

Also ist $\delta < 0$ und folglich

$$A^T y \geq -\delta e > 0,$$

also (II) lösbar.

6. Sei $A \in \mathbb{R}^{m \times n}$, $b \in \mathbb{R}^m$. Man beweise den Alternativsatz von Gale: Genau eine der beiden Aussagen

(I) $$Ax \leq b \qquad \text{hat eine Lösung } x \in \mathbb{R}^n$$

bzw.

(II) $$A^T y = 0, \quad y \geq 0, \quad b^T y < 0 \qquad \text{hat eine Lösung } y \in \mathbb{R}^m$$

ist richtig.

Lösung: Im (verallgemeinerten) Farkas-Lemma sei K der nichtnegative Orthant im $\mathbb{R}^m$ und C der gesamte $\mathbb{R}^n$.

7. Gegeben sei das sogenannte *Quotientenprogramm*

(P) $$\text{Minimiere} \quad \frac{c^T x + c_0}{d^T x + d_0} \quad \text{auf} \quad M := \{x \in \mathbb{R}^n : x \geq 0,\ Ax = b\}.$$

Es wird vorausgesetzt, dass $d^T x + d_0 > 0$ für alle $x \in M$. Der Optimierungsaufgabe (P) ordne man das lineare Programm

$$(\hat{\mathrm{P}}) \quad \begin{cases} \text{Minimiere} \quad \begin{pmatrix} c \\ c_0 \end{pmatrix}^T \begin{pmatrix} z \\ z_0 \end{pmatrix} \quad \text{auf} \\ \hat{M} := \left\{(z, z_0) \in \mathbb{R}^n \times \mathbb{R} : \begin{pmatrix} z \\ z_0 \end{pmatrix} \geq 0,\ \begin{pmatrix} A & -b \\ d^T & d_0 \end{pmatrix} \begin{pmatrix} z \\ z_0 \end{pmatrix} = \begin{pmatrix} 0 \\ 1 \end{pmatrix}\right\}. \end{cases}$$

Man zeige: Ist (z^*, z_0^*) eine Lösung von $(\hat{\mathrm{P}})$ mit $z_0^* > 0$, so ist $x^* := (1/z_0^*) z^*$ eine Lösung von (P).

Lösung: Zunächst ist x^* offensichtlich zulässig für (P). Nun sei $x \in M$ beliebig. Mit $v(x) := d^T x + d_0 > 0$ ist $(x/v(x), 1/v(x)) \in \hat{M}$ zulässig für $(\hat{\mathrm{P}})$. Da (z^*, z_0^*) eine Lösung von (P) ist, ist daher

$$\frac{c^T x^* + c_0}{d^T x^* + d_0} = c^T z^* + c_0 z_0^* \leq \frac{c^T x + c_0}{v(x)} = \frac{c^T x + c_0}{d^T x + d_0},$$

womit die Behauptung schon bewiesen ist.

8. Sei $A \in \mathbb{R}^{m\times n}$, $b \in \mathbb{R}^m$. Die Menge $M := \{x \in \mathbb{R}^n : x \geq 0,\ Ax \geq b\}$ sei nichtleer. Man zeige, dass M genau dann beschränkt ist, wenn es ein $u \in \mathbb{R}^m$ mit $u \geq 0$ und $A^T u < 0$ gibt. Wie lautet die entsprechende Aussage, wenn $M := \{x \in \mathbb{R}^n : x \geq 0,\ Ax = b\}$?

   Lösung: Wir nehmen zunächst an, es sei M beschränkt. Dann gibt es kein $p \in \mathbb{R}^n \setminus \{0\}$ mit $Ap \geq 0$, $p \geq 0$ (da mit einem $x \in M$ andernfalls $x + tp \in M$ für alle $t \geq 0$ wäre, ein Widerspruch zur Beschränktheit von M). Dies impliziert, dass das System $Ap \geq 0$, $p \geq 0$, $-e^T p < 0$ (hierbei ist e der Vektor, dessen Komponenten alle gleich 1 sind) nicht lösbar ist. Das Farkas-Lemma wiederum zeigt, dass es ein $u \in \mathbb{R}^m$ mit $-e - A^T u \geq 0$, $u \geq 0$ gibt, was zu zeigen war.

   Nun existiere ein $u \in \mathbb{R}^m$ mit $u \geq 0$ und $A^T u < 0$. Im Widerspruch zur Behauptung nehmen wir an, die Menge M sei nicht beschränkt. Dann existiert eine Folge $\{x_k\} \subset M$ mit $\|x_k\| \to \infty$. Die Folge $\{p_k\}$ mit $p_k := x_k/\|x_k\|$ besitzt einen Häufungspunkt p, für den offenbar $p \neq 0$ und $p \geq 0$, $Ap \geq 0$ gilt. Dann wäre $(A^T u)^T p$ als Skalarprodukt eines negativen Vektors mit einem nichtverschwindenden nichtnegativen Vektor einerseits negativ, andererseits ist $(A^T u)^T p = u^T Ap$ als Skalarprodukt zweier nichtnegativer Vektoren nichtnegativ. Das ist der gewünschte Widerspruch.

   Sei nun $M := \{x \in \mathbb{R}^n : x \geq 0,\ Ax = b\}$. Ist M beschränkt, so gibt es kein $p \in \mathbb{R}^n \setminus \{0\}$ mit $Ap = 0$, $p \geq 0$. Dies impliziert, dass das System

$$\begin{pmatrix} A \\ e^T \end{pmatrix} p = \begin{pmatrix} 0 \\ 1 \end{pmatrix}, \quad p \geq 0$$

   nicht lösbar ist. Das Farkas-Lemma liefert die Existenz von $(u, \alpha) \in \mathbb{R}^m \times \mathbb{R}$ mit

$$\begin{pmatrix} A^T & e \end{pmatrix} \begin{pmatrix} u \\ \alpha \end{pmatrix} \geq 0, \quad \begin{pmatrix} 0 \\ 1 \end{pmatrix}^T \begin{pmatrix} u \\ \alpha \end{pmatrix} < 0.$$

   Offensichtlich ist $A^T u > 0$. Umgekehrt existiere $u \in \mathbb{R}^m$ mit $A^T u > 0$. Ist M nicht beschränkt, so existiert (Routineschluss) $p \in \mathbb{R}^n \setminus \{0\}$ mit $Ap = 0$, $p \geq 0$. Dann ist $0 = u^T Ap = (A^T u)^T p > 0$, ein Widerspruch.

6.1.2 Aufgaben zu Abschnitt 2.2

1. Man löse mit Hilfe des revidierten Simplexverfahrens ein lineares Programm in Standardform, bei dem die Daten durch

$$\begin{array}{|c|c|}\hline c^T & \\ \hline A & b \\ \hline \end{array} := \begin{array}{|cccccc|c|}\hline 14 & -19 & 0 & 21 & 52 & 0 & \\ \hline 1 & 0 & 0 & -1 & 1 & 1 & 3 \\ 1 & 1 & 0 & -1 & 3 & 0 & 4 \\ 1 & 1 & 1 & -3 & 0 & 1 & 6 \\ \hline \end{array}$$

   gegeben sind, wobei man mit der Basis $B := \{1, 2, 6\}$ starte.

   Lösung: Wir benutzen die in MATLAB geschriebene Funktion `simplex`. Das MATLAB-Programm

```
A=[1 0 0 -1 1 1;1 1 0 -1 3 0;1 1 1 -3 0 1];
b=[13;15;3];c=[14;-19;0;21;52;0];
B=[1 2 6];A_B=A(:,B);in=inv(A_B);x=in*b;
```

```
wert=c(B)'*x;
info=0;
while info==0
   [B,x,wert,in,y,info]=simplex(A,b,c,B,x,wert,in);
end;
[m,n]=size(A);xx=zeros(n,1);xx(B)=x;x=xx;
```

berechnet die optimale Basisindexmenge $B^* = \{4, 2, 6\}$, die Lösungen

$$x^* = \begin{pmatrix} 0 \\ 40 \\ 0 \\ 25 \\ 0 \\ 38 \end{pmatrix}, \qquad y^* = \begin{pmatrix} 2 \\ -17 \\ -2 \end{pmatrix}$$

des primalen bzw. des hierzu dualen Programms, der Wert ist $c^T x^* = b^T y^* = -235$.

2. Man beweise Lemma 2.8.

   Lösung: Wir definieren die Matrix $T_q \in \mathbb{R}^{(m-1)\times m}$ durch

$$T_q := \begin{pmatrix} e_1^T \\ \vdots \\ e_{q-1}^T \\ e_{q+1}^T \\ \vdots \\ e_m^T \end{pmatrix}.$$

   Dann ist

$$T_q^T T_q = I - e_q e_q^T \in \mathbb{R}^{m\times m}, \qquad T_q T_q^T = I \in \mathbb{R}^{(m-1)\times(m-1)}.$$

   Mit dieser Notation ist $A^{qr} = T_q A T_r^T$, $b^q = T_q b$. Folglich ist

$$\begin{aligned} A^{qr}(A^{-1})^{rq} &= T_q A T_r^T T_r A^{-1} T_q^T \\ &= T_q A (I - e_r e_r^T) A^{-1} T_q^T \\ &= T_q (A - e_q e_r^T) A^{-1} T_q^T \\ &= \underbrace{T_q T_q^T}_{=I} - \underbrace{T_q e_q}_{=0} e_r^T A^{-1} T_q^T \\ &= I, \end{aligned}$$

   womit (a) bewiesen ist. Unter Benutzung von (a) ist weiter

$$\begin{aligned} (A^{qr})^{-1} b^q &= T_r A^{-1} T_q^T T_q b \\ &= T_r A^{-1} (I - e_q e_q^T) b \\ &= T_r A^{-1} b - e_q^T b \underbrace{T_r e_r}_{=0} \\ &= (A^{-1} b)^r, \end{aligned}$$

   womit auch (b) bewiesen ist.

3. Man beweise Satz 2.9.

Lösung: Ist $A_B^{-1}b \geq 0$, so ist die Basislösung x mit dem Basisanteil $x_B := A_B^{-1}b$ primal zulässig. Nach Voraussetzung ist $y := A_B^{-T}c_B$ dual zulässig. Da ferner

$$b^T y = c_B^T x_B = c^T x$$

gilt, folgt aus dem schwachen Dualitätssatz, dass x eine Lösung von (P) und y eine Lösung des dualen Programms (D) ist. Nun sei $x_B = A_B^{-1}b > 0$. Ist $u \in \mathbb{R}^m$ dual zulässig und $u \neq y$, so ist $A_B^T u \leq c_B$ und $A_B^T u \neq c$. Dann folgt aber (nach Multiplikation von links mit x_B^T unter Benutzung von $c_B^T x_B = b^T y$) $b^T u < b^T y$, also die Eindeutigkeit von y.

Besitzt $A_B^{-1}b$ eine negative Komponente $(A_B^{-1}b)_r 0$, ist $u := A_B^{-T}e_r$ und $A_N^T u \geq 0$, so ist einerseits $y(\gamma) := y + \gamma u$ für alle $\gamma \leq 0$ wegen

$$A^T y(\gamma) = \begin{pmatrix} c_B + \gamma e_r \\ A_N^T y + \gamma A_N^T u \end{pmatrix} \leq \begin{pmatrix} c_B \\ c_N \end{pmatrix} = c$$

dual zulässig, andererseits ist

$$b^T y(\gamma) = b^T y + \gamma b^T A_B^{-T} e_r = b^T y + \gamma \underbrace{(A_B^{-1}b)_r}_{<0} \to +\infty \qquad \text{für} \quad \gamma \to -\infty,$$

also $\sup(\mathrm{D}) = +\infty$ und daher (P) nicht zulässig.

Sei nun $(A_B^{-1}b)_r < 0$, $u := A_B^{-T}e_r$ und $v_N := A_N^T u \not\geq 0$ mit $r \in \{1, \ldots, m\}$. Wählt bzw. bestimmt man $s \in N$ auf die angegebene Weise mit $v_s < 0$ und

$$\frac{\overline{c}_s}{v_s} = \max_{j \in N}\left\{ \frac{\overline{c}_j}{v_j} : v_j < 0 \right\} =: \gamma^*,$$

so ist die r-te Komponente des Vektors $w := A_B^{-1}a_s$ negativ wegen

$$w_r = e_r^T A_B^{-1} a_s = a_s^T u = v_s < 0.$$

Die neuen Basisindizes B^+ gewinnt man aus B, indem man $j(r)$ durch s ersetzt. Daher erhält man A_{B^+} aus A_B dadurch, dass die r-te Spalte $a_{j(r)}$ durch a_s ersetzt wird. Genau wie im Beweis von Satz 2.7 folgt wegen $w_r \neq 0$, dass mit A_B auch A_{B^+} nichtsingulär und die Inverse durch

$$A_{B^+}^{-1} = \left(I - \frac{(w - e_r)e_r^T}{w_r} \right) A_B^{-1}$$

gegeben ist. Insbesondere ist x^+ mit dem Basisanteil $x_{B^+}^+ := A_{B^+}^{-1}b$ eine Basislösung. Den neuen Vektor $y^+ := A_{B^+}^{-T}c_{B^+}$ berechnet man aus

$$\begin{aligned} y^+ &= A_B^{-T}\left(I - \frac{e_r(w - e_r)^T}{w_r} \right)[c_B + (c_s - c_{j(r)})e_r] \\ &= A_B^{-T}\left(c_B - \frac{(w - e_r)^T c_B}{w_r} e_r + (c_s - c_{j(r)})e_r - \frac{(w_r - 1)(c_s - c_{j(r)})}{w_r} e_r \right) \\ &= A_B^{-T}\left(c_B - \frac{a_s^T y - c_{j(r)}}{w_r} e_r + (c_s - c_{j(r)})e_r - \frac{(w_r - 1)(c_s - c_{j(r)})}{w_r} e_r \right) \end{aligned}$$

$$
\begin{aligned}
&= A_B^{-T}\left(c_B + \frac{c_s - a_s^T y}{v_s} e_r\right) \\
&= y + \frac{c_s - a_s^T y}{v_s} A_B^{-T} e_r \\
&= y + \gamma^* A_B^{-T} e_r.
\end{aligned}
$$

Die neuen Nichtbasisindizes sind $N^+ := (N \setminus \{s\}) \cup \{j(r)\}$. Um nachzuweisen, dass y^+ dual zulässig ist, hat man $c_j - a_j^T y^+ \geq 0$ für alle $j \in N^+$ nachzuweisen. Nun ist

$$
c_{j(r)} - a_{j(r)}^T y^+ = \underbrace{c_{j(r)} - a_{j(r)}^T y}_{=0} - \gamma^* \underbrace{a_{j(r)}^T u}_{=1} = -\gamma^* \geq 0
$$

und

$$
c_j - a_j^T y = c_j - a_j^T y - \gamma^* a_j^T u = \bar{c}_j - \gamma^* v_j \geq 0, \qquad j \in N^+ \setminus \{j(r)\},
$$

nach Definition von γ^*. Also ist y^+ dual zulässig. Schließlich ist

$$
c_0^+ := c^T x^+ = b^T y^+ = b^T y + \gamma^* (A_B^{-1} b)_r \geq b^T y = c^T x =: c_0,
$$

der Satz ist bewiesen.

4. In einer Reifenfabrik[3] werden Sommer- und Winterreifen hergestellt. Die Fabrik hat Verträge, bis zu bestimmten Daten eine gewisse Zahl von Reifen mindestens herzustellen, nämlich

Zeitpunkt	Sommerreifen	Winterreifen
30. Juni	5 000	3 000
31. Juli	6 000	3 000
31. August	4 000	5 000

Zur Produktion stehen zwei Typen von Maschinen zur Verfügung. Die Anzahl der zur Verfügung stehenden Produktionsstunden für die beiden Maschinen während der Sommermonate sind:

Monat	Maschine A	Maschine B
Juni	700	1 500
Juli	300	400
August	1 000	300

Die Produktionsraten (Stunden pro Reifen) auf den beiden Typen von Maschinen sind

Typ	Maschine A	Maschine B
Sommerreifen	0.15	0.16
Winterreifen	0.12	0.14

Unabhängig von den benutzten Typen und den produzierten Reifen kostet eine Arbeitsstunde 100 DM. Das Material für einen Sommerreifen kostet 52.50 DM, das für einen Winterreifen 41.50 DM. Pro Reifen kommen noch 4 DM hinzu. Überschüssige Reifen

[3] Die Aufgabe ist

M. Asghar Bhatti (2000) *Practical Optimization Methods. With Mathematica Applications*. Springer-Verlag, New York-Berlin-Heidelberg

entnommen.

können in den nächsten Monat (also von Juni in den Juli und von Juli in den August) übernommen werden, die Lagerkosten sind 1.50 DM pro Reifen. Die produzierten Reifen werden für 200 DM (Sommerreifen) bzw. 150 DM (Winterreifen) verkauft. Wie[4] sollte die Produktion organisiert werden, um einerseits den Lieferbedingungen nachzukommen und andererseits den Gewinn der Fabrik zu maximieren?

Lösung: Als Variable führen wir ein:

x_1	Zahl der im Juni auf Machine A produzierten Sommerreifen
x_2	Zahl der im Juli auf Maschine A produzierten Sommerreifen
x_3	Zahl der im August auf Maschine A produzierten Sommerreifen
x_4	Zahl der im Juni auf Machine A produzierten Winterreifen
x_5	Zahl der im Juli auf Machine A produzierten Winterreifen
x_6	Zahl der im August auf Machine A produzierten Winterreifen
x_7	Zahl der im Juni auf Machine B produzierten Sommerreifen
x_8	Zahl der im Juli auf Maschine B produzierten Sommerreifen
x_9	Zahl der im August auf Maschine B produzierten Sommerreifen
x_{10}	Zahl der im Juni auf Machine B produzierten Winterreifen
x_{11}	Zahl der im Juli auf Machine B produzierten Winterreifen
x_{12}	Zahl der im August auf Machine B produzierten Winterreifen

Zunächst stellen wir die Zielfunktion auf. Durch den Verkauf erhält man

$$v = 200(x_1 + x_2 + x_3 + x_7 + x_8 + x_9) + 150(x_4 + x_5 + x_6 + x_{10} + x_{11} + x_{12})$$

DM. Die Materialkosten betragen

$$m = 52.50(x_1 + x_2 + x_3 + x_7 + x_8 + x_9) + 41.50(x_4 + x_5 + x_6 + x_{10} + x_{11} + x_{12})$$

DM. Arbeitskosten sind

$$a = 100[0.15(x_1+x_2+x_3)+0.16(x_7+x_8+x_9)+0.12(x_4+x_5+x_6)+0.14(x_{10}+x_{11}+x_{12})]$$

DM. Zusätzliche Kosten sind

$$z = 4(x_1 + x_2 + x_3 + x_4 + x_5 + x_6 + x_7 + x_8 + x_9 + x_{10} + x_{11} + x_{12})$$

DM, Lagerkosten

$$\begin{aligned} l \;=\; & 1.5[(x_1 + x_7 - 5000) + (x_4 + x_{10} - 3000) \\ & + (x_1 + x_2 + x_7 + x_8 - 11000) + (x_4 + x_5 + x_{10} + x_{11} - 6000)] \end{aligned}$$

DM. Der Gesamtgewinn ist $v - (m + a + z + l)$, dieser ist zu maximieren. Die Produktionsbeschränkungen führen auf die Restriktionen

$$0.15x_1 + 0.12x_4 \le 700, \quad 0.15x_2 + 0.12x_5 \le 300, \quad 0.15x_3 + 0.12x_6 \le 1000$$

und

$$0.16x_7 + 0.14x_{10} \le 1500, \quad 0.16x_8 + 0.14x_{11} \le 400, \quad 0.16x_9 + 0.14x_{12} \le 300.$$

[4]Eigentlich handelt es sich hier um ein ganzzahliges lineares Programm, wovon wir aber absehen wollen.

Wegen der eingegangenen Verträge kommen die weiteren Restriktionen

$$\begin{aligned} x_1 + x_7 &\geq 5000, \\ x_4 + x_{10} &\geq 3000, \\ x_1 + x_2 + x_7 + x_8 &\geq 11000, \\ x_4 + x_5 + x_{10} + x_{11} &\geq 6000, \\ x_1 + x_2 + x_3 + x_7 + x_8 + x_9 &\geq 15000, \\ x_4 + x_5 + x_6 + x_{10} + x_{11} + x_{12} &\geq 11000 \end{aligned}$$

hinzu, weiter sind natürlich alle Variablen nichtnegativ. Wir wollen zunächst eine Lösung mit Hilfe des mathematischen Anwenderprogramms *Mathematica* berechnen. Schreibt man die Aufgabe in der Form

$$\text{Minimiere} \quad c^T x \quad \text{unter der Nebenbedingung} \quad x \geq 0,\ Ax \geq b,$$

so sind die Daten durch

−125.5	−127	−128.5	−89.5	−91	−92.5	−124.5	−126	−127.5	−87.5	−89	−90.5	
−0.15	0	0	−0.12	0	0	0	0	0	0	0	0	−1500
0	−0.15	0	0	−0.12	0	0	0	0	0	0	0	−300
0	0	−0.15	0	0	−0.12	0	0	0	0	0	0	−1000
0	0	0	0	0	0	−0.16	0	0	−0.14	0	0	−1500
0	0	0	0	0	0	0	−0.16	0	0	−0.14	0	−400
0	0	0	0	0	0	0	0	−0.16	0	0	−0.14	−300
1	0	0	0	0	0	1	0	0	0	0	0	5000
0	0	0	1	0	0	0	0	0	1	0	0	3000
1	1	0	0	0	0	1	1	0	0	0	0	11000
0	0	0	1	1	0	0	0	0	1	1	0	6000
1	1	1	0	0	0	1	1	1	0	0	0	15000
0	0	0	1	1	1	0	0	0	1	1	1	11000

gegeben. Nach

```
c={-125.5,-127,-128.5,-89.5,-91,-92.5,-124.5,-126,
-127.5,-87.5,-89,-90.5};
a={{-0.15,0,0,-0.12,0,0,0,0,0,0,0,0},{0,-0.15,0,0,-0.12,0,0,0,0,0,0,0},
   {0,0,-0.15,0,0,-0.12,0,0,0,0,0,0},{0,0,0,0,0,0,-0.16,0,0,-0.14,0,0},
   {0,0,0,0,0,0,0,-0.16,0,0,-0.14,0},{0,0,0,0,0,0,0,0,-0.16,0,0,-0.14},
   {1,0,0,0,0,0,1,0,0,0,0,0},{0,0,0,1,0,0,0,0,0,1,0,0},
   {1,1,0,0,0,0,1,1,0,0,0,0},{0,0,0,1,1,0,0,0,0,1,1,0},
   {1,1,1,0,0,0,1,1,1,0,0,0},{0,0,0,1,1,1,0,0,0,1,1,1}};
b={-700,-300,-1000,-1500,-400,-300,5000,3000,11000,6000,15000,11000};
LinearProgramming[c,a,b]
```

erhalten wir

$$x^* = (1866.67, 0, 2666.67, 3500, 2500, 5000, 9375, 2500, 1875, 0, 0, 0)^T$$

als Lösung. Zur Kontrolle wollen wir auch noch unsere MATLAB-Funktion `simplex` anwenden. Bei der Anwendung der Phase I hat man ein Programm mit 12 Restriktionen und 30 Variablen. Man startet mit der Basisindexmenge

$$B = \{13, 14, 15, 16, 17, 18, 25, 26, 27, 28, 29, 30\},$$

hat zum Schluss der Phase I eine zulässige Basislösung zur Indexmenge

$$B = \{1, 5, 6, 16, 8, 18, 7, 4, 19, 11, 3, 9\}$$

berechnet. Hiermit kann die Phase II gestartet werden und man erhält als optimale Basisindexmenge

$$B^* = \{1, 5, 6, 21, 8, 23, 7, 4, 19, 20, 3, 9\}$$

und eine Lösung, die mit der durch *Mathematica* gefundenen übereinstimmt.

Nun wollen wir auch noch das Ergebnis für den Fall bestimmen, dass zum Schluss der Produktionszeit (also Ende August) kein Überschuss vorhanden ist. Die letzten beiden Restriktionen sind dann zu ersetzen durch

$$x_1 + x_2 + x_3 + x_7 + x_8 + x_9 = 15000, \qquad x_4 + x_5 + x_6 + x_{10} + x_{11} + x_{12} = 11000.$$

Als Lösung mit *Mathematica* erhalten wir

$$x^* = (1866.67, 0, 2666.67, 3500, 2500, 5000, 6333.33, 2500, 1333.33, 0, 0, 0)^T.$$

Dies stimmt genau mit dem Ergebnis bei M. Asghar Bhatti (2000, S. 417) überein. Auch mit Matlab erhält man dasselbe Ergebnis (gegenüber dem zuerst betrachteten Fall vermindert sich die Anzahl der Variablen um zwei, da man zwei Schlupfvariable weniger benötigt).

5. Man löse das lineare Programm[5] in Standardform

   (P) Minimiere $c^T x$ unter den Nebenbedingungen $x \geq 0,\ Ax = b,$

   wobei die Daten durch

$$\begin{array}{|c|c|}\hline c^T & \\ \hline A & b \\ \hline \end{array} := \begin{array}{|rrrrrrr|r|}\hline 5 & 3 & 3 & 6 & 0 & 0 & 0 & \\ \hline -6 & 1 & 2 & 4 & 1 & 0 & 0 & 14 \\ 3 & -2 & -1 & -5 & 0 & 1 & 0 & -25 \\ -2 & 1 & 0 & 2 & 0 & 0 & 1 & 14 \\ \hline \end{array}$$

   gegeben sind, mit dem (revidierten) dualen Simplexverfahren.

   Lösung: Nach

```
A=[-6 1 2 4 1 0 0;3 -2 -1 -5 0 1 0;-2 1 0 2 0 0 1];
b=[14;-25;14];c=[5;3;3;6;0;0;0];B=[5 6 7];
A_B=A(:,B);in=inv(A_B);x=in*b;y=in'*c(B);wert=c(B)'*x;
info=0;
while info==0
   [B,x,wert,in,y,info]=dualsimplex(A,b,c,B,x,wert,in,y);
end;
```

   erhält man als optimale Basisindexmenge B, optimalen Basisanteil x_B und Lösung des dualen Problems

$$B = \{2, 4, 7\}, \qquad x_B = \begin{pmatrix} 10 \\ 1 \\ 2 \end{pmatrix}, \qquad y = \begin{pmatrix} -1 \\ -2 \\ 0 \end{pmatrix}.$$

[5] Siehe V. Chvátal (1983, S. 155).

6. Man schreibe ein Programm, das zu vorgegebenen $A \in \mathbb{R}^{m\times n}$ und $b \in \mathbb{R}^m$ eine Lösung der diskreten, linearen Tschebyscheffschen Approximationsaufgabe, $f(x) := \|Ax-b\|_\infty$ auf dem $\mathbb{R}^n$ zu minimieren, berechnet. Der Einfachheit halber werde Rang $(A) = n$ vorausgesetzt. Anschließend teste man das Programm an den Daten $A = (a_{ij}) \in \mathbb{R}^{11\times 5}$, $b = (b_i) \in \mathbb{R}^{11}$, wobei die Einträge mit $t_i := (i-1)\frac{1}{10}$, $i = 1,\ldots,11$, durch

$$a_{ij} := t_i^{j-1}, \qquad b_i := \exp(t_i) \qquad (i = 1,\ldots,11,\ j = 1,\ldots,5)$$

gegeben sind (wobei $0^0 = 1$).

Lösung: Wir geben eine einfache MATLAB-Funktion an, welche `simplex` benutzt.

```
%Es wird das lineare, diskrete Tschebyscheffsche Approximations-
%problem

%Minimiere ||A*x-b||_00,    x aus R^n

%gel"ost. Hierbei wird die Funktion simplex benutzt.

%Aufruf:  [x,delta]=tscheby(A,b)
%Eingabe: Daten (A,b) des Problems
%Ausgabe: x ist L"osung, delta gibt den Minimalabstand an.
%=====================================================================
function [x,delta]=tscheby(A,b);
[m,n]=size(A);
%=====================================================================
%                                  Phase I, Aufbau der Matrix
    hatA=[A' -A' zeros(n,1) eye(n);ones(1,m) ones(1,m) 1 zeros(1,n)];
    hatb=[zeros(n,1);1];hatc=[zeros(m,1);zeros(m,1);0;ones(n,1)];
    B=2*m+1:2*m+1+n;hatA_B=hatA(:,B);in=inv(hatA_B);x=in*hatb;
    wert=hatc(B)'*x;info=0;
    while info==0
        [B,x,wert,in,y,info]=simplex(hatA,hatb,hatc,B,x,wert,in);
    end;
%=====================================================================
%          Phase I beendet, Start von Phase II
    hatA=hatA(:,1:2*m+1);hatc=[-b;b;0];info=0;
    while info==0
        [B,x,wert,in,y,info]=simplex(hatA,hatb,hatc,B,x,wert,in);
    end;
x=-y(1:n); delta=-y(n+1);
```

Im Testproblem erhalten wir (mit `format long` ausgegeben) als Lösung

$$x^* = \begin{pmatrix} 1.00002602631671 \\ 0.99871406400225 \\ 0.51007702177190 \\ 0.13971661696698 \\ 0.06972207308446 \end{pmatrix}, \qquad \delta^* = \|Ax^*-b\|_\infty = 2.602631671336231 \cdot 10^{-5}.$$

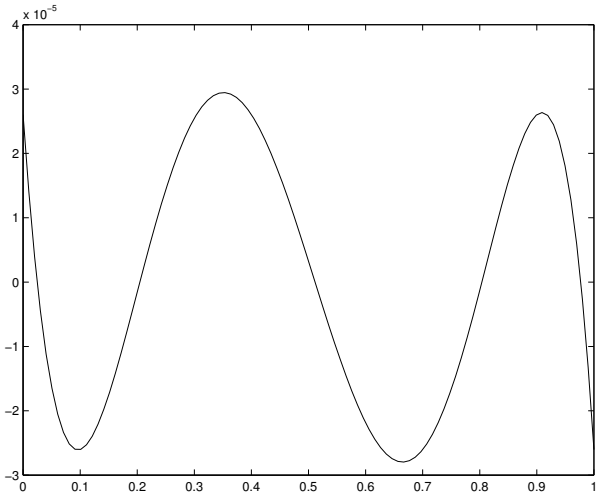


Abbildung 6.2: Fehler bei diskreter Tschebyscheff-Approximation

Dieses Ergebnis sowie einen Plot des Fehlers $p^*(t) - \exp(t)$ mit $p^*(t) := x_1 + x_2 t + \cdots + x_5 t^4$ (siehe Abbildung 6.2) haben wir durch

```
m=11;n=5;t=[0:1/(m-1):1]';
A(1:m,1)=1;A(1:m,2)=t;
for j=3:n
   A(1:m,j)=t.^(j-1);
end;b=exp(t);
[x,delta]=tscheby(A,b);
s=linspace(0,1);p=x(n:-1:1);p=p';
v=polyval(p,s)-exp(s);
plot(s,v);
```

erhalten.

7. Man benutze[6] das Simplexverfahren um nachzuweisen, dass für alle $t \in (-3, -\frac{1}{2})$ die eindeutige Lösung von

$$(\mathrm{P}_t) \quad \text{Minimiere} \quad tx_1 - x_2 \quad \text{unter den Nebenbedingungen} \quad \begin{matrix} x_1 + 2x_2 \le 4, \\ 6x_1 + 2x_2 \le 9, \end{matrix} \quad x \ge 0$$

in einem von t unabhängigen Punkt x^* angenommen wird. Man berechne $x^* = (x_1^*, x_2^*)$.

Lösung: Etwa für $t = -1$ erhält man die optimale Basismenge $B = \{2, 1\}$, den Basisanteil $x_B = (\frac{3}{2}, 1)^T$ einer Lösung und

$$A_B^{-1} = \begin{pmatrix} \frac{3}{5} & -\frac{1}{10} \\ -\frac{1}{5} & \frac{1}{5} \end{pmatrix}.$$

[6]Diese Aufgabe wurde

R. Fletcher (1987, S. 190) *Practical Methods of Optimization*. J. Wiley, Chichester

entnommen.

Mit den Nichtbasisindizes $N = \{3, 4\}$ sind bezüglich B die reduzierten Kosten der parametrisierten Zielfunktion durch

$$\begin{aligned}\overline{c}_N &= c_N - A_N^T A_B^{-T} \begin{pmatrix} -1 \\ t \end{pmatrix} \\ &= \begin{pmatrix} 0 \\ 0 \end{pmatrix} - \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} \frac{3}{5} & -\frac{1}{5} \\ -\frac{1}{10} & \frac{1}{5} \end{pmatrix} \begin{pmatrix} -1 \\ t \end{pmatrix} \\ &= \begin{pmatrix} \frac{3}{5} + \frac{1}{5}t \\ -\frac{1}{10} - \frac{1}{5}t \end{pmatrix}\end{aligned}$$

gegeben. Für $t \in (-3, 1)$ ist $\overline{c}_N > 0$, woraus man die Behauptungen abliest. Für diese t ist $x^* := (1, \frac{3}{2})$ die eindeutige Lösung von (P_t) und $y^*(t) := (-\frac{3}{5} - \frac{1}{5}t, \frac{1}{10} + \frac{1}{5}t)^T$ Lösung des hierzu dualen Programms (D_t).

8. Als Verallgemeinerung von Aufgabe 7 überlege man sich: Gegeben sei das lineare Programm

   $$(\mathrm{P}_t) \qquad \text{Minimiere} \quad (c + td)^T x \quad \text{auf} \quad M := \{x \in \mathbb{R}^n : x \geq 0,\ Ax = b\}.$$

   Mit Hilfe des Simplexverfahrens sei eine optimale Basisindexmenge B und die zugehörige optimale Basislösung x^* von (P_0) bestimmt. Es sei $\overline{c}_N := c_N - A_N^T A_B^{-T} c_B > 0$ mit $N := \{1, \ldots, n\} \setminus B$. Für welche t ist x^* auch Lösung von (P_t)?

   Lösung: Nach Voraussetzung ist $x_B^* = A_B^{-1} b$ der Basisanteil der Basislösung x^*. Wir definieren $y^*(t) := A_B^{-T}(c_B + td_B)$. Wegen des schwachen Dualitätssatzes ist x^* eine Lösung von (P_t) für alle t, für die

   $$\underbrace{c_N - A_N^T A_B^{-T} c_B}_{>0} - tA_N^T A_B^{-T} d_B \geq 0.$$

9. Gegeben sei das lineare Programm in Standardform

   $$(\mathrm{P}) \qquad \text{Minimiere} \quad c^T x \quad \text{auf} \quad M := \{x \in \mathbb{R}^n : x \geq 0,\ Ax = b\}.$$

   Sei Rang $(A) = m$ und $x^* \in M$ eine nichtentartete, optimale Ecke von M, d. h. x^* ist eine Lösung von (P), die Indexmenge $B := \{j \in \{1, \ldots, n\} : x_j^* > 0\}$ enthält genau m Elemente und die Matrix $A_B \in \mathbb{R}^{m \times m}$ ist nichtsingulär. Man zeige:

   (a) Es ist $y^* := A_B^{-T} c_B$ die eindeutige Lösung des zu (P) dualen linearen Programms

   $$(\mathrm{D}) \qquad \text{Maximiere} \quad b^T y \quad \text{auf} \quad N := \{y \in \mathbb{R}^n : A^T y \leq c\}.$$

   (b) Für beliebiges $d \in \mathbb{R}^m$ ist das gestörte Problem

   $$(\mathrm{P}_t) \qquad \text{Minimiere} \quad c^T x \quad \text{auf} \quad M_t := \{x \in \mathbb{R}^n : x \geq 0,\ Ax = b + td\}.$$

   für alle hinreichend kleinen $|t|$ lösbar und es ist

   $$\min(\mathrm{P}_t) = \min(\mathrm{P}) + td^T y^* \qquad \text{für alle hinreichend kleinen } |t|.$$

Lösung: Wir zeigen, dass $y^* := A_B^{-T} c_B$ dual zulässig ist. Wegen

$$c^T x^* = c_B^T x_B^* = c_B^T A_B^{-1} b = b^T A_B^{-T} c_B = b^T y^*$$

ist dann wegen des schwachen Dualitätssatzes auch die Optimalität von y^* nachgewiesen. Da (P) lösbar ist, ist es auch (D) und es tritt keine Dualitätslücke ein. D. h. es existiert ein $\hat{y} \in N$ mit $c^T x^* = b^T \hat{y}$ bzw. $(x^*)^T (c - A^T \hat{y}) = 0$. Wegen $x_B^* > 0$ und $x_N^* = 0$ ist $c_B - A_B^T \hat{y} = 0$ und folglich $y^* = \hat{y} \in N$. Damit ist der erste Teil der Aufgabe bewiesen.

Wegen $x_B^* = A_B^{-1} b > 0$ ist $A_B^{-1}(b + td) \geq 0$ für alle hinreichend kleinen $|t|$. Für diese t definiere man $x(t) \in \mathbb{R}^n$ durch $x(t)_B := A_B^{-1}(b + td)$ und $x(t)_j := 0$ für $j \in N$. Dann ist $x(t)$ zulässig für (P_t), ferner ist y^* zulässig für das zu (P_t) duale lineare Programm

$$(\mathrm{D}_t) \qquad \text{Maximiere} \quad (b + td)^T y \quad \text{auf} \quad N := \{y \in \mathbb{R}^m : A^T y \leq c\}.$$

Wegen

$$c^T x(t) = c_B^T x(t)_B = c_B^T A_B^{-1}(b + td) = (b + td)^T y^*$$

und des schwachen Dualitätssatzes ist $x(t) \in M_t$ eine Lösung von (P_t) (und y^* eine Lösung des hierzu dualen Programms (D_t)) für alle hinreichend kleinen $|t|$, ferner

$$\min(\mathrm{P}_t) = \min(P) + t d^T y^* \qquad \text{für alle hinreichend kleinen } |t|,$$

wie es behauptet wurde.

10. Man löse das lineare Programm

$$\text{Minimiere} \quad -x_1 - x_2 \quad \text{unter den Nebenbedingungen} \quad \begin{array}{rrrl} x_1 & & + 2x_3 & \leq 1 \\ & x_2 & - \; x_3 & \leq 1 \\ x_1 & + x_2 & + \; x_3 & = 2, \end{array} \qquad x \geq 0,$$

indem man es zunächst auf Standardform bringt und mit Hilfe der Methode der künstlichen Variablen eine Anfangsbasislösung bestimmt.

Lösung: In der Phase I wird das Simplexverfahren auf ein Problem in Standardform mit den Daten

$$\begin{array}{|c|c|}\hline \hat{c}^T & \\ \hline \hat{A} & b \\ \hline \end{array} := \begin{array}{|ccccccc|}\hline 0 & 0 & 0 & 0 & 0 & 1 & \\ \hline 1 & 0 & 2 & 1 & 0 & 0 & 1 \\ 0 & 1 & -1 & 0 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 & 0 & 1 & 2 \\ \hline \end{array}$$

angewandt. Der Wert dieses Programms ist 0, das Ausgangsproblem also zulässig. In der optimalen Basisindexmenge $B = \{1, 2, 6\}$ ist allerdings die künstliche Variable noch enthalten, ferner ist $x_B = (1, 1, 0)^T$. Es ist $\{1, \ldots, 5\} \setminus B = \{3, 4, 5\}$. Wegen

$$\hat{A}_B^{-1} \begin{pmatrix} 2 & 1 & 0 \\ -1 & 0 & 1 \\ 1 & 0 & 0 \end{pmatrix} = \begin{pmatrix} 2 & 1 & 0 \\ -1 & 0 & 1 \\ 0 & -1 & -1 \end{pmatrix}$$

kann der Basisindex 6 mit 4 oder 5 ausgetauscht werden. Wir wählen $s = 4$, haben die künstliche Variable aus der Basis vertrieben und können die Phase II anwenden. Hier erhält man sofort, dass $B = \{1, 2, 4\}$ eine optimale Basisindexmenge ist, $x_B = (1, 1, 0)^T$ ist der Basisanteil einer optimalen Lösung.

11. Mit Hilfe des Simplexverfahrens sei eine (optimale Basis-) Lösung x^* von

$$\text{(P)} \qquad \text{Minimiere} \quad c^T x \quad \text{auf} \quad M := \{x \in \mathbb{R}^n : x \geq 0,\ Ax = b\}$$

berechnet. Hierbei sei $A \in \mathbb{R}^{m\times n}$ und Rang $(A) = m$. Wie würden Sie das Simplexverfahren für die erweiterte Aufgabe

$$(\hat{\text{P}}) \quad \begin{cases} \text{Minimiere} \quad \begin{pmatrix} c \\ c_{n+1} \end{pmatrix}^T \begin{pmatrix} x \\ x_{n+1} \end{pmatrix} \quad \text{auf} \\ \hat{M} := \left\{ \begin{pmatrix} x \\ x_{n+1} \end{pmatrix} \in \mathbb{R}^{n+1} : \begin{pmatrix} x \\ x_{n+1} \end{pmatrix} \geq 0,\ \begin{pmatrix} A & a_{n+1} \end{pmatrix} \begin{pmatrix} x \\ x_{n+1} \end{pmatrix} = b \right\} \end{cases}$$

starten? Unter welcher Voraussetzung ist

$$\hat{x} := \begin{pmatrix} x^* \\ 0 \end{pmatrix}$$

eine Lösung von $(\hat{\text{P}})$?

Lösung: Sei B eine optimale Basisindexmenge für (P), also $x_B^* = A_B^{-1} b$ der Basisanteil der Lösung x^*. Mit dieser Basisindexmenge (bzw. der zulässigen Basislösung $\hat{x} = (x^*, 0)$ für das erweiterte System) kann offenbar sofort das Simplexverfahren zur Lösung von $(\hat{\text{P}})$ gestartet werden. Durch $y^* = A_B^{-T} c_B$ ist eine Lösung des zu (P) dualen Programms gegeben. Das Simplexverfahren sagt uns, dass $\hat{x}$ schon eine Lösung von $(\hat{\text{P}})$ ist, wenn

$$\begin{pmatrix} c_N \\ c_{n+1} \end{pmatrix} - \begin{pmatrix} A_N^T \\ a_{n+1}^T \end{pmatrix} y^* \geq \begin{pmatrix} 0 \\ 0 \end{pmatrix}.$$

Hierbei ist $N := \{1, \ldots, n\} \setminus B$ die Menge der Nichtbasisindizes. Die Ungleichung $c_N - A_N^T y^* \geq 0$ ist erfüllt (duale Zulässigkeit von y^* bzw. Optimalitätstest für x^*). Daher ist $\hat{x}$ eine Lösung von $(\hat{\text{P}})$, wenn $c_{n+1} - a_{n+1}^T y^* \geq 0$.

12. Gegeben sei die lineare Optimierungsaufgabe mit beschränkten Variablen

$$\text{(P)} \qquad \text{Minimiere} \quad \text{auf} \quad M := \{x \in \mathbb{R}^n : l \leq x \leq u,\ Ax = b\}.$$

Wie üblich sei hier $A \in \mathbb{R}^{m\times n}$, $b \in \mathbb{R}^m$ und $c \in \mathbb{R}^n$, ferner seien l, u zwei Vektoren des $\mathbb{R}^n$ mit $l < u$, wobei allerdings die Komponenten von l auch gleich $-\infty$ und die von u gleich $+\infty$ sein können. Eine Variable x_j (oder einfacher nur j) heißt *unbeschränkt* oder *frei*, wenn $l_j = -\infty$ und $u_j = +\infty$, andernfalls *beschränkt*. Sei $x \in M$ eine zulässige Basislösung von (P), es existiere also eine Indexmenge $B \subset \{1, \ldots, n\}$ mit $\#(B) = m$ und der Eigenschaft, dass A_B nichtsingulär ist und für alle beschränkten $j \in N := \{1, \ldots, n\} \setminus B$ entweder $x_j = l_j$ oder $x_j = u_j$ gilt. Man zeige, dass x eine Ecke von M ist, wenn alle Nichtbasisindizes beschränkt sind.

Lösung: Sei $x = (1-\lambda)x^{(1)} + \lambda x^{(2)}$ mit $x^{(1)}, x^{(2)} \in M$ und $\lambda \in (0,1)$. Nach Voraussetzung ist jedes $j \in N$ beschränkt. Dann ist $x_j = l_j$ oder $x_j = u_j$. In beiden Fällen ist $x_j = x_j^{(1)} = x_j^{(2)}$, also $x_N = x_N^{(1)} = x_N^{(2)}$. Wegen $x_B = A_B^{-1} b - A_B^{-1} A_N x_N$ folgt dann auch $x_B = x_B^{(1)} = x_B^{(2)}$, insgesamt also $x = x^{(1)} = x^{(2)}$. Damit ist bewiesen, dass x eine Ecke von M ist.

13. Gegeben sei die lineare Optimierungsaufgabe[7]

$$\text{Minimiere} \quad -4x_1 - x_2 \quad \text{unter den Nebenbedingungen} \quad \begin{aligned} 2x_1 + x_2 &\le 9 \\ -2x_1 + x_2 &\le 1 \\ x_1 &\le 3 \\ 1 \le x_2 &\le 5. \end{aligned}$$

Nach Einführung von Schlupfvariablen führt dies auf ein lineares Programm in Standardform (mit beschränkten Variablen)

$$\text{(P)} \qquad \text{Minimiere} \quad c^T x \quad \text{auf} \quad M := \{x \in \mathbb{R}^n : l \le x \le u,\ Ax = b\},$$

wobei die Daten durch

$$\begin{array}{|c|c|}\hline c^T & \\ \hline A & b \\ \hline l^T & \\ u^T & \\ \hline \end{array} := \begin{array}{|rrrr|c|}\hline -4 & -1 & 0 & 0 & \\ \hline 2 & 1 & 1 & 0 & 9 \\ -2 & 1 & 0 & 1 & 1 \\ \hline -\infty & 1 & 0 & 0 & \\ 3 & 5 & +\infty & +\infty & \\ \hline \end{array}$$

gegeben sind. Durch $x := (0, 1, 8, 0)^T$ ist eine zulässige Basislösung zur Basis $B := \{1, 3\}$ gegeben. Hiervon ausgehend bestimme man mit dem (für beschränkte Variable) modifizierten Simplexverfahren die Lösung.

Lösung: Durch $x = (0, 1, 8, 0)$ ist in der Tat eine zulässige Basislösung zur Basis $B = \{1, 3\}$ mit den Kosten -1 gegeben. Die (sämtlich beschränkten) Nichtbasisindizes sind $N = \{2, 4\}$.

(a) Die reduzierten Kosten berechnen sich zu

$$\overline{c}_N = c_N - A_N^T A_B^{-T} c_B = \begin{pmatrix} -3 \\ -2 \end{pmatrix} = \begin{pmatrix} \overline{c}_2 \\ \overline{c}_4 \end{pmatrix}.$$

Für $s = 2$ ist $\overline{c}_s = -3$ und $x_s = 1 < 5 = u_s$. Daher berechnen wir

$$w := A_B^{-1} a_s = \begin{pmatrix} -\frac{1}{2} \\ 2 \end{pmatrix}, \qquad x(t) := \begin{pmatrix} \frac{1}{2}t \\ 1 + t \\ 8 - 2t \\ 0 \end{pmatrix}.$$

Es ist $t^* = 4$, die Kosten der neuen Basislösung also $-1 - |c_s| \cdot t^* = -13$. Von den beiden Fällen zur Berechnung der neuen Basislösung tritt der zweite Fall ein. Es ist also $x^+ = (2, 5, 0, 0)^T$ neue Basislösung, nach wie vor ist $B^+ = B = \{1, 3\}$.

(b) Durch $x = (2, 5, 0, 0)^T$ haben wir eine zulässige Basislösung zu $B = \{1, 3\}$ mit den Kosten -13 erhalten, die Nichtbasisindizes sind $N = \{2, 4\}$. Die reduzierten Kosten haben sich nicht verändert. Diesmal kann $s = 4$ gewählt werden, da $\overline{c}_4 = -2 < 0$ und $x_4 = 0 < +\infty = u_4$. Dann ist

$$w := A_B^{-1} a_s = \begin{pmatrix} -\frac{1}{2} \\ 1 \end{pmatrix}, \qquad x(t) := \begin{pmatrix} 2 + \frac{1}{2}t \\ 5 \\ -t \\ t \end{pmatrix}.$$

Es ist $t^* = 0$, man hat $x^+ = (2, 5, 0, 0)^T$ mit $B^+ = \{1, 4\}$.

[7] Siehe K. Neumann, M. Morlock (1993, S. 95).

(c) Jetzt ist $x = (2, 5, 0, 0)^T$, $B = \{1, 4\}$ und $N = \{2, 3\}$. Die reduzierten Kosten sind jetzt

$$\overline{c}_N = c_N - A_N^T A_B^{-T} c_B = \begin{pmatrix} 1 \\ 2 \end{pmatrix} = \begin{pmatrix} \overline{c}_2 \\ \overline{c}_3 \end{pmatrix}.$$

Es kann $s = 2$ gewählt werden, da dann $\overline{c}_s = 1 > 0$ und $x_s = 5 > 1 = l_s$. Wir berechnen

$$w := A_B^{-1} a_s = \begin{pmatrix} \frac{1}{2} \\ 2 \end{pmatrix}, \qquad x(t) = \begin{pmatrix} 2 + \frac{1}{2}t \\ 5 - t \\ 0 \\ 2t \end{pmatrix}.$$

Jetzt ist $t^* = 2$ und es tritt wieder der erste Teil, also eine Veränderung der Basisindexmenge, ein. Es wird $x^+ = (3, 3, 0, 4)$ mit $B^+ = \{2, 4\}$, $N^+ = \{1, 3\}$ und neuen Kosten -15.

(d) Zu $B := \{2, 4\}$ und $N := \{1, 3\}$ sind die reduzierten Kosten

$$\overline{c}_N = c_N - A_N^T A_B^{-T} c_B = \begin{pmatrix} -2 \\ 1 \end{pmatrix} = \begin{pmatrix} \overline{c}_1 \\ \overline{c}_3 \end{pmatrix}.$$

Die Optimalitätsbedingung ist erfüllt und wir haben eine Lösung berechnet.

6.1.3 Aufgaben zu Abschnitt 2.3

1. Seien $x, z \in \mathbb{R}^n$ gegeben und $X := \operatorname{diag}(x)$, $\|\cdot\|$ bezeichne die euklidische Norm. Dann ist

$$\left\| Xz - \frac{x^T z}{n} e \right\| = \min_{\lambda \in \mathbb{R}} \|Xz - \lambda e\|,$$

d. h. $(x^T z/n)e$ ist die orthogonale Projektion von Xz auf $\operatorname{span}\{e\}$.

Lösung: Zu zeigen ist, dass

$$Xz - \frac{x^T z}{n} e \in \operatorname{span}\{e\}^{\perp}.$$

Wegen

$$\left(Xz - \frac{x^T z}{n} e \right)^T e = x^T z - x^T z = 0$$

ist dies trivial.

2. Gegeben sei ein lineares Programm in Karmarkar-Standardform, also

(P) Minimiere $c^T x$ auf $M := \left\{ x \in \mathbb{R}^n : x \geq 0, \begin{pmatrix} A \\ e^T \end{pmatrix} = \begin{pmatrix} 0 \\ n \end{pmatrix} \right\}$.

Hierbei seien $A \in \mathbb{R}^{m \times n}$ und $c \in \mathbb{R}^n$ mit $n \geq 2$ gegeben, weiter sei $e := (1, \ldots, 1)^T \in \mathbb{R}^n$. Es werde vorausgesetzt:

(V) Es ist $Ae = 0$ und $\operatorname{Rang}(A) = m$.

Man zeige, dass die Voraussetzungen (A1), (A2), (A3) aus Unterabschnitt 2.3.1 erfüllt sind.

Lösung: Wegen $e \in M_0$ ist (A1) trivialerweise erfüllt. Das zu (P) duale lineare Programm ist

$$\text{(D)} \qquad \text{Maximiere} \quad n\phi \quad \text{auf} \quad N := \{(y,\phi) \in \mathbb{R}^m \times \mathbb{R} : A^T y + \phi e \le c\}.$$

Offensichtlich ist das Innere N_0 von N nichtleer, so dass auch (A2) gilt. Schließlich ist

$$\text{Rang} \begin{pmatrix} A \\ e^T \end{pmatrix} = m+1.$$

Denn ist

$$0 = \begin{pmatrix} A \\ e^T \end{pmatrix}^T \begin{pmatrix} y \\ \phi \end{pmatrix} = A^T y + \phi e,$$

so folgt nach Multiplikation von links mit e^T, dass

$$0 = (\underbrace{Ae}_{=0})^T y + \phi n = \phi n$$

und dann $\phi = 0$, aus $A^T y = 0$ und Rang $(A) = m$ folgt auch $y = 0$.

3. Man betrachte[8] ein lineares Programm der Form

$$\text{Minimiere} \quad c^T x \quad \text{unter den Nebenbedingungen} \quad Ax \le b, \quad Cx = d.$$

Man führe dieses Programm in Standardform über. Hat das zu diesem Programm in Standardform duale Programm strikt zulässige Lösungen?

Lösung: In Standardform geschrieben lautet die Aufgabe

$$\text{(P)} \quad \begin{cases} \text{Minimiere} \quad \begin{pmatrix} c \\ -c \\ 0 \end{pmatrix}^T \begin{pmatrix} x_+ \\ x_- \\ y \end{pmatrix} \quad \text{unter den Nebenbedingungen} \\ \qquad \begin{pmatrix} x_+ \\ x_- \\ y \end{pmatrix} \ge 0, \quad \begin{pmatrix} A & -A & I \\ C & -C & 0 \end{pmatrix} \begin{pmatrix} x_+ \\ x_- \\ y \end{pmatrix} = \begin{pmatrix} b \\ d \end{pmatrix}. \end{cases}$$

Das zu (P) duale Programm ist

$$\text{(D)} \quad \begin{cases} \text{Maximiere} \quad \begin{pmatrix} b \\ d \end{pmatrix}^T \begin{pmatrix} u \\ v \end{pmatrix} \quad \text{unter den Nebenbedingungen} \\ \qquad \begin{pmatrix} A^T & C^T \\ -A^T & -C^T \\ I & 0 \end{pmatrix} \begin{pmatrix} u \\ v \end{pmatrix} \le \begin{pmatrix} c \\ -c \\ 0 \end{pmatrix}. \end{cases}$$

Ganz offensichtlich hat (D) keine strikt zulässige Lösung.

[8] Siehe S. J. Wright (1997, S. 124).

4. Gegeben sei die Nullstellenaufgabe $f(x) = 0$ mit einer in einer Umgebung U^* von $x^* \in \mathbb{R}$ vier mal stetig differenzierbaren reellwertigen Funktion f. Sei $f(x^*) = 0$ und $f'(x^*) \neq 0$. Man zeige, dass das *zusammengesetzte Newton-Verfahren*

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

lokal von mindestens dritter Ordnung konvergiert. An hand von $f(x) := x - \cos^2 x$ mit den Startwerten $x_0 := 1$, $x_0 := 2$, $x_0 := 5$ vergleiche man die durch das Newton- und das zusammengesetzte Newton-Verfahren erhaltenen Werte. Weshalb wird diese Aufgabe in dem Abschnitt über Innere-Punkt-Verfahren gestellt?

Hinweis: Man wende einen aus der Numerischen Mathematik her bekannten Satz über die Konvergenzgeschwindigkeit von Iterationsvererfahren bei nichtlinearen Gleichungen an, siehe z. B. J. Werner (1992, S. 98)[9]. Zum Nachweis der Voraussetzungen dieses Satzes kann man *Mathematica* benutzen.

Lösung: Die Iterationsfunktion des angegebenen Verfahrens ist

$$\phi(x) := x - \frac{f(x) + f(x - f(x)/f'(x))}{f'(x)}.$$

Wir haben zu zeigen, dass $\phi'(x^*) = 0$ und $\phi''(x^*) = 0$, eine Anwendung eines aus der Numerischen Mathematik I bekannten Satzes liefert die Behauptung. Das machen wir mit *Mathematica*. Als Output von

```
phi[x_]:=x-(f[x]+f[x-f[x]/D[f[x],x]])/D[f[x],x];
phi1[x_]:=D[phi[x],x];
phi1[x]/.{f[x]->0}
phi2[x_]:=D[phi[x],{x,2}];
phi2[x]/.{f[x]->0}/.{f[x]->0}
```

erhalten wir zweimal die 0. Nach

```
f[x_]:=x-Cos[x]^2;
iter1[x_]:=x-f[x]/D[f[x],x];
iter2[x_]:=x-(f[x]+f[x-f[x]/D[f[x],x]])/D[f[x],x];
x1=1.0;x2=1.0;
For[k=1,k<=4,k++,{x1=iter1[x]/.{x->x1};x2=iter2[x]/.{x->x2};
Print[N[x1,12],"    ",N[x2,12]]}]
```

erhalten wir (links Newton-Verfahren, rechts zusammengesetztes Newton-Verfahren) z. B.

0.629144517598	0.642017424966
0.641738615165	0.641714370874
0.641714370958	0.641714370873
0.641714370873	0.641714370873

[9] J. Werner (1992) *Numerische Mathematik 1*. Vieweg, Braunschweig-Wiesbaden.

Sehr viel anders sind die Ergebnisse, wenn wir mit $x_0 := 2$ bzw. $x_0 := 5$ starten. Die Ergebnisse sind links bzw. rechts angegeben.

−5.51167991113	19.2647704005
−2.49831289618	3.98082032263
−0.896923083151	1.2786990274
51.0281001916	0.672604371719
25.7623270168	0.641715303253
12.8980209772	0.641714370873
5.4684966547	0.641714370873

−5.7889551074	8.60632380143
−2.21222558728	$1.6980979928 \cdot 10^6$
−0.900097522337	283039.129569
48.205377482	57484.9965054
21.9804385109	2524.44410063
0.540677993367	681.475228766
0.643933285131	580.242000932
0.641714370873	$3.31998645878 \cdot 10^6$
0.641714370873	21349.3142248

Das normale Newton-Verfahren versagt also im ersten Fall, während das zusammengesetzte Newton-Verfahren nach anfänglichen Wirrungen offenbar konvergiert. Im zweiten Fall ist es genau umgekehrt. Das am Schluss des Abschnitts über primal-duale Innere-Punkt-Verfahren skizzierte Prädiktor-Korrektor-Verfahren erinnert an das zusammengesetzte Newton-Verfahren. Dieser Zusammenhang wird näher bei R. Tapia, Y. Zhang, M. Saltzman, A. Weiser (1996)[10] beschrieben.

5. Man programmiere einen Schritt des unzulässigen primal dualen Verfahrens für ein Problem in Standardform mit den Daten (A, b, c). Anschließend wende man das Verfahren auf ein Beispiel mit den Daten

$$\begin{array}{|c|c|}\hline c^T & \\ \hline A & b \\ \hline \end{array} := \begin{array}{|ccccccc|c|}\hline 5 & 3 & 3 & 6 & 0 & 0 & 0 & \\ \hline -6 & 1 & 2 & 4 & 1 & 0 & 0 & 14 \\ 3 & -2 & -1 & -5 & 0 & 1 & 0 & -25 \\ -2 & 1 & 0 & 2 & 0 & 0 & 1 & 14 \\ \hline \end{array}$$

an, wobei man mit $(x, y, z) := (e, 0, e)$ starte. Insbesondere beobachte man, wie sich der Defekt

$$d(x, y, z) := \frac{\|Ax - b\|}{\max(1, \|b\|)} + \frac{\|A^T y + z - c\|}{\max(1, \|c\|)} + \frac{|c^T x - b^T y|}{\max(1, |c^T x|, |b^T y|)}$$

verändert. Man sollte verschiedene Strategien bei der Wahl von σ ausprobieren, etwa $\sigma_k := 0.5$, $\sigma_k := 1/(k+1)$ und $\sigma_k := 1/(k+1)^2$.

Lösung: Wir haben in Matlab ein sehr einfaches Programm, in welchem ein Schritt des primal-dualen Innere-Punkt-Verfahrens durchgeführt wird, geschrieben. Hierbei benutzen wir im wesentlichen die selben Bezeichnungen wie in Unterabschnitt 2.3.2 und verzichten auf alle programmtechnischen Feinheiten.

```
%Es wird ein Schritt eines primal-dualen Innere-Punkt-Verfahrens
%durchgef"uhrt. Es werden die Bezeichnungen der Vorlesung benutzt.
%Der Aufruf geschieht durch [x_p,y_p,z_p]=primdual(A,b,c,x,y,z,sigma);
%Input:    Problem-Daten (A,b,c), ferner Tripel (x,y,z) mit x>0,z>0
%          und sigma>0.
```

[10] R. Tapia, Y. Zhang, M. Saltzman, A. Weiser (1996) "The Mehrotra predictor-corrector interior-point method as a perturbed composite Newton method". SIAM J. Optimization 6, 47–56.

```
%Output:   Neues N"aherungstripel (x_p,y_p,z_p).
%=====================================================================
function [x_p,y_p,z_p]=primdual(A,b,c,x,y,z,sigma);
[m,n]=size(A);tau_p=0.995;tau_d=0.995;mu=sigma*x'*z/n;
X=diag(x);Z=diag(z);D=diag(x./z);e=ones(size(x));
%Berechne Residuen:
r_p=A*x-b; r_d=A'*y+z-c; r_xz=X*z-mu*e;
%Berechne Newton-Richtungen:
q=-A*D*A'\(r_p+A*D*(r_d-r_xz./x));
r=-A'*q-r_d;p=-D*r-r_xz./z;
%Berechne primale Schrittweite
P=find(p<0);
alpha_pmax=min(-x(P)./p(P));alpha_p=min(1,tau_p*alpha_pmax);
%Berechne duale Schrittweite
R=find(r<0);
alpha_dmax=min(-z(R)./r(R));alpha_d=min(1,tau_d*alpha_dmax);
%Berechne neue N"aherung:
x_p=x+alpha_p*p;y_p=y+alpha_d*q;z_p=z+alpha_d*r;
%=====================================================================
```

In der folgenden Tabelle geben wir für 10 Iterationen bei Wahl verschiedener σ-Strategien den erhaltenen Defekt an:

$\sigma_k = 1/(k+1)^2$	$\sigma_k = 1/(k+1)$	$\sigma_k = 0.5$
1.2791	1.2372	1.2372
0.9517	0.8975	0.8772
0.3191	0.1969	0.0870
$2.7893e-04$	0.0012	0.0070
$7.7495e-06$	$2.0203e-04$	0.0035
$1.5815e-07$	$2.8864e-05$	0.0018
$2.4712e-09$	$3.6081e-06$	$8.7952e-04$
$3.0508e-11$	$4.0090e-07$	$4.3987e-04$
$3.0723e-13$	$4.0090e-08$	$2.1996e-04$
$3.3103e-15$	$3.6446e-09$	$1.0999e-04$

Man erkennt also sehr deutlich, dass die Folge $\{\sigma_k\}$ hinreichend schnell gegen Null konvergieren sollte. Es wäre sehr interessant, hierfür genauere theoretische Begründungen zu geben. Weiter müsste man sich über die Wahl einer geeigneten Startnäherung (mehr) Gedanken machen.

6. Man wende das primal-duale Innere-Punkt-Verfahren auf die linearen Programme in Standardform an, wie sie durch Aufgabe 4 in Unterabschnitt 2.2.7 gegeben sind.

   Lösung: Wir unterscheiden (wie schon früher) zwei Fälle. Zum einen darf Ende August ein Überschuss sein, im zweiten Fall nicht. Im ersten Fall hat die Koeffizientenmatrix 12 Zeilen und 24 Spalten (die letzten 12 durch Schlupfvariable verursacht). Wir begannen mit dem sehr schlechten und eigentlich durch nichts zu begründenden Startwert $(x, y, z) = (e, 0, e)$ (in der entsprechenden Literatur werden bessere Vorschläge für die Wahl eines Startwertes gemacht). Mit der σ-Strategie $\sigma_k := 1/(k+1)^2$ erhalten wir

für die Lösung (einschließlich der Schlupfvariablen) des primalen Problems die "Näherungslösung" x^*, die wir links angeben. Für den Fall, dass Ende August kein Überschuss vorhanden sein darf (man hat nach wie vor 12 Restriktionen aber nur 22 Variable, da man zwei Schlupfvariable weniger benötigt), geben wir die "Näherungslösung" x^{**} rechts an.

$$x^* = 10^3 \begin{pmatrix} 1.86666666666667 \\ 0.00000000000000 \\ 2.66666666666667 \\ 3.50000000000000 \\ 2.50000000000000 \\ 5.00000000000000 \\ 9.37499999999999 \\ 2.50000000000000 \\ 1.87500000000000 \\ 0.00000000000000 \\ 0.00000000000000 \\ 0.00000000000000 \\ 0.00000000000000 \\ 0.00000000000000 \\ 0.00000000000000 \\ 0.00000000000000 \\ 0.00000000000000 \\ 0.00000000000000 \\ 6.24166666666667 \\ 0.50000000000000 \\ 2.74166666666666 \\ 0.00000000000000 \\ 3.28333333333331 \\ 0.00000000000000 \end{pmatrix}, \qquad x^{**} = 10^3 \begin{pmatrix} 1.86666666666666 \\ 0.00000000000000 \\ 2.66666666666667 \\ 3.49999999999999 \\ 2.50000000000000 \\ 5.00000000000000 \\ 6.63333333333333 \\ 2.50000000000000 \\ 1.33333333333332 \\ 0.00000000000000 \\ 0.00000000000000 \\ 0.00000000000000 \\ 0.00000000000000 \\ 0.00000000000000 \\ 0.00000000000000 \\ 0.43866666666667 \\ 0.00000000000000 \\ 0.08666666666667 \\ 3.50000000000000 \\ 0.50000000000000 \\ 0.00000000000000 \\ 0.00000000000000 \end{pmatrix}$$

7. Gegeben seien das lineare Programm

   (P) $\qquad$ Minimiere $\quad c^T x \quad$ auf $\quad M := \{x \in \mathbb{R}^n : x \geq 0,\ Ax = b\}$

   und das hierzu duale lineare Programm

   (D) $\qquad$ Maximiere $\quad b^T y \quad$ auf $\quad N := \{y \in \mathbb{R}^m : A^T y \leq c\},$

   wobei $A = (\ a_1 \ \cdots \ a_n\ ) \in \mathbb{R}^{m \times n}$ eine Matrix mit Rang $(A) = m$ ist, $b \in \mathbb{R}^m$ und $c \in \mathbb{R}^n$. Sei $x^* \in M$ eine *nichtentartete, zulässige Basislösung* von $Ax = b$, die Indexmenge $B^* := \{j \in \{1, \ldots, n\} : x_j^* > 0\}$ besitze also genau m Elemente und die Spalten $\{a_j\}_{j \in B^*}$ sind linear unabhängig. Ferner existiere ein $y^* \in \mathbb{R}^m$ mit $(A^T y^*)_j = c_j$ für alle $j \in B^*$ und $(A^T y^*)_j < c_j$ für alle $j \notin B^*$. Man zeige:

   (a) Die linearen Programme (P) bzw. (D) besitzen $x^* \in M$ bzw. $y^* \in N$ als eindeutige Lösungen.

   (b) Es ist

   $$M_0 := \{x \in \mathbb{R}^n : x > 0,\ Ax = b\} \neq \emptyset, \qquad N_0 := \{y \in \mathbb{R}^m : A^T y < c\} \neq \emptyset.$$

(c) Für jedes $\mu > 0$ besitzen die Programme

$$(\mathrm{P}_\mu) \qquad \text{Minimiere} \quad f_\mu(x) := c^T x - \mu \sum_{j=1}^{n} \log x_j, \quad x \in M_0$$

und

$$(\mathrm{D}_\mu) \qquad \text{Maximiere} \quad g_\mu(y) := b^T y + \mu \sum_{j=1}^{n} \log(c - A^T y)_j, \quad y \in N_0$$

eindeutige Lösungen $x_\mu \in M_0$ bzw. $y_\mu \in N_0$. Ferner gilt

$$c - \mu\, X_\mu^{-1} e = A^T y_\mu \qquad \text{mit} \quad X_\mu := \operatorname{diag}(x_\mu)$$

und

$$\lim_{\mu \to 0+} x_\mu = x^*, \qquad \lim_{\mu \to 0+} y_\mu = y^*.$$

(d) Man definiere $z^* := c - A^T y^*$ und anschließend

$$X^* := \operatorname{diag}(x^*), \qquad Z^* := \operatorname{diag}(z^*).$$

Für $\mu > 0$ seien $x_\mu \in M_0$ bzw. $y_\mu \in N_0$ die eindeutigen Lösungen von (P_μ) bzw. (D_μ), ferner sei $z_\mu := c - A^T y_\mu$. Man zeige:

i. Die Matrix

$$\begin{pmatrix} Z^* & 0 & X^* \\ A & 0 & 0 \\ 0 & A^T & I \end{pmatrix}$$

ist nichtsingulär.

ii. Definiert man $(\dot{x}_0, \dot{y}_0, \dot{z}_0) \in \mathbb{R}^n \times \mathbb{R}^m \times \mathbb{R}^n$ durch

$$\begin{pmatrix} Z^* & 0 & X^* \\ A & 0 & 0 \\ 0 & A^T & I \end{pmatrix} \begin{pmatrix} \dot{x}_0 \\ \dot{y}_0 \\ \dot{z}_0 \end{pmatrix} = \begin{pmatrix} e \\ 0 \\ 0 \end{pmatrix}$$

(hierbei ist $e \in \mathbb{R}^n$ einmal wieder der Vektor, dessen Komponenten alle gleich Eins sind), so gilt

$$\lim_{\mu \to 0+} \frac{(x_\mu, y_\mu, z_\mu)) - (x^*, y^*, z^*)}{\mu} = (\dot{x}_0, \dot{y}_0, \dot{z}_0).$$

Lösung: Wegen des ersten Teiles von Satz 2.7 sind x^* und y^* Lösungen von (P) bzw. (D), hierbei x^* sogar eindeutige Lösung von (P). Wegen des entsprechenden Teils in Satz 2.9 ist y^* auch eindeutige Lösung von (D).

Wegen der eindeutigen Lösbarkeit von (P) und (D) sind die Lösungsmengen M_{opt} und N_{opt} insbesondere kompakt. Daher sind (Routineschluss)

$$c^T p \le 0, \quad p \ge 0, \quad Ap = 0$$

und

$$b^T q \ge 0, \quad A^T q \le 0$$

nur trivial lösbar. Das impliziert, dass

$$\begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix} - \begin{pmatrix} A \\ e^T \\ c^T \end{pmatrix} p \in \{0\} \times \{0\} \times \mathbb{R}_{\geq 0}, \quad p \geq 0$$

und

$$\begin{pmatrix} -A^T \\ b^T \end{pmatrix} q \geq \begin{pmatrix} 0 \\ 0 \end{pmatrix}, \quad (Ae)^T q < 0$$

nicht lösbar sind. Das Farkas-Lemma liefert die Existenz von $(u, \alpha, \beta) \in \mathbb{R}^m \times \mathbb{R} \times \mathbb{R}$ mit

$$A^T u + \alpha e + \beta c \geq 0, \quad \beta \geq 0, \quad \alpha < 0$$

und $(v, \gamma) \in \mathbb{R}^n \times \mathbb{R}$ mit

$$-Av + \gamma b = Ae, \quad v \geq 0, \quad \gamma \geq 0.$$

Ist $\beta > 0$, so ist $y_0 := -u/\beta \in N_0$. Für $\beta = 0$ ist $y_0 := y - u \in N_0$ mit einem beliebigen $y \in N$. Für $\gamma > 0$ ist $x_0 := (v+e)/\gamma \in N_0$. Ist dagegen $\gamma = 0$, so ist $x_0 := x+v+e \in M_0$ mit einem beliebigen $x \in M$. Insgesamt ist gezeigt, dass M_0 und N_0 nichtleer sind.

Die eindeutige Existenz von Lösungen $x_\mu \in M_0$ bzw. $y_\mu \in N_0$ von (P_μ) bzw. (D_μ) wird in Satz 3.2 bewiesen. Da x^* und y^* eindeutige Lösungen von (P) bzw. (D) sind, folgt $\lim_{\mu \to 0+} x_\mu = x^*$ und $\lim_{\mu \to 0+} y_\mu = y^*$ aus Satz 3.3. Die Beziehung $c - \mu X_\mu^{-1} = A^T y_\mu$ folgt z. B. aus Satz 3.4.

Die Nichtsingularität der Matrix

$$\begin{pmatrix} Z^* & 0 & X^* \\ A & 0 & 0 \\ 0 & A^T & I \end{pmatrix}$$

wurde in Lemma 3.5 bewiesen. Die restliche Aussage könnte mit dem Satz über implizite Funktionen bewiesen werden. Wir ziehen einen direkten Beweis vor. Mit $N^* := \{1, \ldots, n\} \setminus B^*$ werde die Menge aller Nichtbasisindizes bezeichnet. Ist x im Folgenden ein Vektor, dessen Komponenten alle ungleich Null sind, so bezeichne x^{-1} den Vektor, dessen Komponenten gerade die Reziproken entsprechender Komponenten von x sind. Aus

$$c - \mu x_\mu^{-1} = A^T y_\mu$$

erhält man

$$c_{B^*} - \mu (x_\mu)_{B^*}^{-1} = A_{B^*}^T y_\mu, \qquad c_{N^*} - \mu (x_\mu)_{N^*}^{-1} = A_{N^*}^T y_\mu.$$

Hieraus erhält man

$$y^* = A_{B^*}^{-T} c_{B^*} = A_{B^*}^{-T} [\mu (x_\mu)_{B^*}^{-1} + A_{B^*}^T y_\mu] = \mu A_{B^*}^{-T} (x_\mu)_{B^*}^{-1} + y_\mu$$

und folglich

$$\frac{y_\mu - y^*}{\mu} = -A_{B^*}^{-T} (x_\mu)_{B^*}^{-1} \to -A_{B^*}^{-T} (x_{B^*}^*)^{-1} =: \hat{y}.$$

Hieraus wiederum folgt unmittelbar

$$\frac{z_\mu - z^*}{\mu} = -A^T \left(\frac{y_\mu - y^*}{\mu} \right) \to A^T A_{B^*}^{-T} (x_{B^*}^*)^{-1} =: \hat{z}.$$

Ferner ist

$$\frac{(x_\mu)_{N^*} - x^*_{N^*}}{\mu} = \frac{(x_\mu)_{N^*}}{\mu} = (c_{N^*} - A^T_{N^*} y_\mu)^{-1} \to (c_{N^*} - A^T_{N^*} y^*)^{-1} =: \hat{x}_{N^*}.$$

Schließlich erhält man aus

$$0 = A\left(\frac{x_\mu - x^*}{\mu}\right) = A_{B^*}\left(\frac{(x_\mu)_{B^*} - x^*_{B^*}}{\mu}\right) + A_{N^*}(c_{N^*} - A^T_{N^*} y_\mu)^{-1},$$

dass

$$\frac{(x_\mu)_{B^*} - x^*_{B^*}}{\mu} = -A^{-1}_{B^*} A_{N^*} (c_{N^*} - A^T_{N^*} y_\mu)^{-1} \to -A^{-1}_{B^*}(c_{N^*} - A^T_{N^*} y^*)^{-1} = \hat{x}_{B^*}.$$

Definiert man nun

$$\hat{x} := \begin{pmatrix} \hat{x}_{B^*} \\ \hat{x}_{N^*} \end{pmatrix},$$

so weist man leicht

$$\begin{pmatrix} Z^* & 0 & X^* \\ A & 0 & 0 \\ 0 & A^T & I \end{pmatrix} \begin{pmatrix} \hat{x} \\ \hat{y} \\ \hat{z} \end{pmatrix} = \begin{pmatrix} e \\ 0 \\ 0 \end{pmatrix}$$

nach, womit alles bewiesen ist.

6.2 Aufgaben zu Kapitel 3

1. Man zeige, dass der Hyperwürfel Q_n genau $n2^{n-1}$ Kanten besitzt und bipartit ist.

   Lösung: Wir beweisen die erste Behauptung durch vollständige Induktion nach n. Für $n = 1$ ist die Aussage richtig. Denn Q_1 besitzt die beiden Ecken $\{0\}$ und $\{1\}$, die durch eine Kante verbunden sind, es ist also $|E(Q_1)| = 1$, wie behauptet. In der Induktionsannahme wird davon ausgegangen, dass $|E(Q_n)| = n2^{n-1}$ die Anzahl von Paaren von 0,1-Folgen der Länge n ist, die sich an genau einer Stelle unterscheiden. Seien nun $\tilde{x} = (x, x_{n+1}), \tilde{y} = (y, y_{n+1}) \in \{0,1\}^n \times \{0,1\}$ zwei 0,1-Folgen der Länge $n+1$, die sich an genau einer Stelle unterscheiden. Dies kann daran liegen, dass x, y zwei 0,1-Folgen der Länge n sind, die sich an genau einer Stelle unterscheiden, und $x_{n+1} = y_{n+1} = 0$ oder $x_{n+1} = y_{n+1} = 1$ gilt (das sind unter Berücksichtigung der Induktionsannahme $n2^{n-1} + n2^{n-1} = n2^n$ Möglichkeiten) oder dass $x = y$ (hier gibt es 2^n Möglichkeiten) und $x_{n+1} = 0$, $y_{n+1} = 1$ bzw. $x_{n+1} = 1$, $y_{n+1} = 0$. Insgesamt hat daher Q_{n+1} genau $n2^n + 2^n = (n+1)2^n$ Kanten), was zu zeigen war. Dass der Hyperwürfel $Q_n = (V_n, E_n)$ bipartit ist, ist einfach zu sehen. Sei nämlich U_n die Menge der Ecken $x \in V_n$, bei denen die Anzahl der Einsen gerade ist, entsprechend W_n die Menge der Ecken, bei denen die Anzahl der Einsen ungerade ist. Kanten können nur zwischen Ecken aus U_n und W_n bestehen. Denn zwei Ecken aus U_n bzw. W_n sind entweder gleich oder unterscheiden sich an mehr als einer Stelle. Daher ist Q_n bipartit.

2. Sei $G = (V, E)$ ein Graph mit $V = \{v_1, \ldots, v_n\}$ und $E = \{e_1, \ldots, e_q\}$. Sei $A \in \mathbb{R}^{n \times n}$ die Adjazenz- und $B \in \mathbb{R}^{n \times q}$ die Inzidenzmatrix von G, sei $D := \text{diag}\,(d(v_1), \ldots, d(v_n))$. Dann ist $BB^T = D + A$.

Lösung: Es ist

$$(BB^T)_{ij} = \sum_{k=1}^{n} b_{ik}b_{jk} = |\{k \in \{1,\ldots,n\} : v_i, v_j \in e_k\}| = \begin{cases} d(v_i) & \text{falls} \quad i = j, \\ a_{ij} & \text{falls} \quad i \neq j. \end{cases}$$

Da die Adjazenzmatrix A Nullen in der Diagonalen besitzt, folgt hieraus die Behauptung.

3. Ein Quickie: Man zeige, dass es keinen Graphen mit der Gradfolge $(5, 4, 3, 2, 2, 1)$ oder $(5, 5, 4, 4, 0)$ geben kann. Weiter konstruiere man einen Graphen mit der Gradfolge $(4, 3, 3, 3, 2, 2, 2, 1)$.

   Lösung: Im ersten Fall ist die Summe aller Grade ungerade, was wegen des Handshaking-Lemmas nicht sei kann. Im zweiten Fall gibt es mindestens eine Ecke mit 5 Nachbarn, wenn es einen Graphen mit dieser Gradfolge gäbe. Der Graph hätte also mindestens sechs Ecken. Da aber nur fünf Zahlen in der Gradfolge auftreten, hat man einen Widerspruch. Mit Hilfe von *Mathematica* hätte man das auch mit `GraphicQ` feststellen können. Z. B. erhält man auf `GraphicQ[{5,4,3,2,2,1}]` die Antwort `False`, während `True` die Antwort auf `GraphicQ[{4,3,3,3,2,2,2,1}]` ist. Ein entsprechender Graph ist in Abbildung 6.3 angegeben.

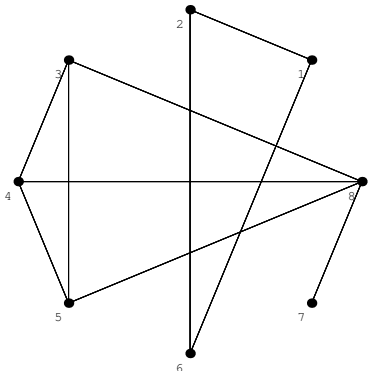


Abbildung 6.3: Ein Graph zur Gradfolge $(4, 3, 3, 3, 2, 2, 2, 1)$

4. Sei $G = (V, E)$ ein zusammenhängender Graph mit $n := |V|$, $m := |E|$. Dann ist $n \leq m + 1$.

   Lösung: Der zusammenhängende Graph $G = (V, E)$ enthält eine aufspannenden Baum $T = (V, E')$. Nach Satz 3.1 ist dann

$$|V| - 1 = |E'| \leq |E|,$$

   das ist die Behauptung.

5. Sei G ein Graph mit 10 Ecken, der nicht zusammenhängend ist. Man zeige, dass G höchstens 36 Kanten besitzt.

   Lösung: Da G nicht zusammenhängend ist, lässt sich die Eckenmenge V als disjunkte Vereinigung nichtleerer Eckenmengen V_1 und V_2 darstellen, wobei es keine Kante

zwischen Ecken aus V_1 und V_2 gibt. Sei $n_i := |V_i|$, $i = 1, 2$, also $n_i \in \{1, \ldots, 9\}$, und $n_1 + n_2 = 10$. Insgesamt kann es höchstens

$$\frac{n_1(n_1-1)}{2} + \frac{n_2(n_2-1)}{2} = \frac{n_1(n_1-1)}{2} + \frac{(10-n_1)(9-n_1)}{2} = n_1^2 - 10n_1 + 45 \leq 36$$

Kanten in G geben.

6. Ein Graph $G = (V, E)$ mit $|E| > \lfloor |V|^2/4 \rfloor$ enthält[11] ein Dreieck.

   Lösung: Der Graph $G = (V, E)$ enthalte kein Dreieck. Ist $xy \in E$, so haben x und y keinen gemeinsamen Nachbarn und es ist folglich $d(x) + d(y) \leq |V|$. Aufsummieren über alle Kanten $xy \in E$ liefert

   $$\sum_{z \in V} d(z)^2 = \sum_{xy \in E} [d(x) + d(y)] \leq |V|\,|E|.$$

   Hier bedarf die erste Gleichung noch eines Beweises. Wenn man berücksichtigt, dass zu jeder Kante $xy \in E$ die beiden Endecken x und y gehören, so erhalten wir

   $$\begin{aligned} \sum_{xy \in E} [d(x) + d(y)] &= \frac{1}{2}\Bigg(\sum_{x \in V} \sum_{y : xy \in E} d(x) + \sum_{y \in V} \sum_{x : xy \in E} d(y) \Bigg) \\ &= \frac{1}{2}\Bigg(\sum_{x \in V} d(x)^2 + \sum_{y \in V} d(y)^2 \Bigg) \\ &= \sum_{z \in V} d(z)^2. \end{aligned}$$

   Mit Hilfe des Handshaking-Lemmas und der Cauchy-Schwarzschen Ungleichung erhalten wir dann, dass

   $$\begin{aligned} 2\,|E| &= \sum_{z \in V} d(z) \\ &\leq |V|^{1/2} \Bigg(\sum_{z \in V} d(z)^2 \Bigg)^{1/2} \\ &\leq |V|^{1/2}\,|V|^{1/2}\,|E|^{1/2} \end{aligned}$$

   und damit $|E| \leq |V|^2/4$ bzw. $|E| \leq \lfloor |V|^2/4 \rfloor$. Die Aussage ist damit bewiesen.

7. Sei $G = (V, E)$ ein zusammenhängender Graph. Mit $d(x, y)$ bezeichnen wir den Abstand zweier Ecken $x, y \in V$, also die Länge eines kürzesten Weges von x nach y. Für $x \in V$ definieren wir $r(x) := \max_{y \in V \setminus \{x\}} d(x, y)$ als die *Exzentrizität* in x, also die Länge eines längsten von x ausgehenden kürzesten Weges. Dann heißt $r(G) := \min_{x \in V} r(x)$ der *Radius* von G und $Z(G) := \{x \in V : r(x) = r(G)\}$ das *Zentrum* von G.

   (a) Was ist der Radius und was das Zentrum zu dem in Abbildung 6.4 dargestellten Graphen G?

   (b) Man zeige, dass das Zentrum eines Baumes entweder aus einer Ecke oder zwei benachbarten Ecken besteht.

[11] Hierbei bedeutet $\lfloor x \rfloor$ für reelles x die größte ganze Zahl $\leq x$.

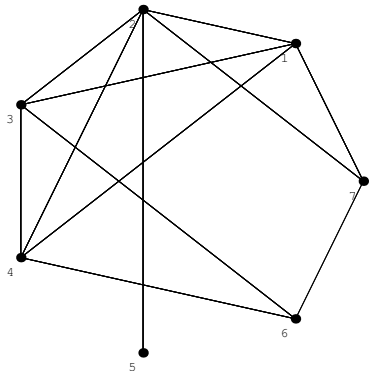


Abbildung 6.4: Was ist der Radius, was ist das Zentrum?

Lösung: Offenbar ist $r(G) = 2$ und $Z(G) = \{1, 2, 3, 4\}$. Im *Mathematica*-Zusatzpaket `DiscreteMath‘Combinatorica‘` gibt es die Befehle `Radius` und `GraphCenter`. Wendet man `Eccentricity` auf einen Graphen an, so erhält man den Vektor (Länge ist die Ordnung des Graphen) der Exzentrizitäten. In unserem Falle ist dies $\{2, 2, 2, 2, 3, 3, 3\}$, woraus man obige Aussage noch einmal ablesen kann.

Wir geben ein Verfahren zur Berechnung des Zentrums eines Baumes an. Jeder Baum hat Ecken vom Grad 1. Diese kommen als Zentrum nicht in Frage. Man streiche sie und die Kante auf der sie liegen. Der entstehende Graph ist wieder ein Baum. Man wiederhole das Verfahren und endet entweder mit einer Ecke oder zwei benachbarten Ecken. In Abbildung 6.5 haben wir zwei Bäume mit 15 Knoten dargestellt (sie wurden

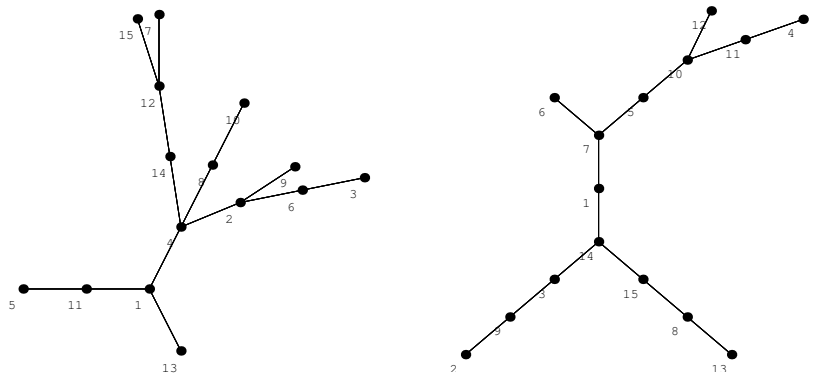


Abbildung 6.5: Zentrum eines Baumes

übrigens durch `RandomTree[15]` erzeugt). Der linke Baum hat $\{4\}$ als Zentrum, während der rechte Baum $\{1, 7\}$ als Zentrum besitzt.

8. Man zeige, dass die in Abbildung 6.6 dargestellten Graphen isomorph sind.

   Lösung: Wir nehmen im links angegebenen Graphen die in Abbildung 6.7 angegebene Umnummerierung der Ecken vor und erkennen die Isomorphie der beiden Graphen. *Mathematica* stellt die Befehle `IsomorphicQ` und `Isomorphism` zur Verfügung. Resultat von `IsomorphicQ[graph1,graph2]` ist `True`, wenn die beiden Graphen isomorph sind, andernfalls `False`. Durch `Isomorphism[graph1,graph2]` kann man eine zulässige

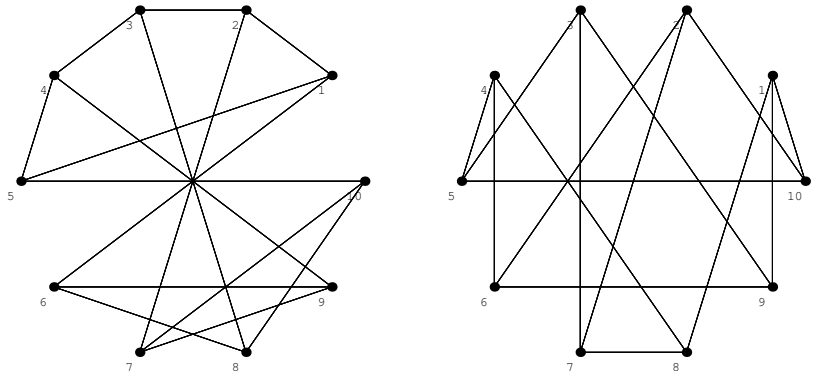


Abbildung 6.6: Zwei isomorphe Graphen

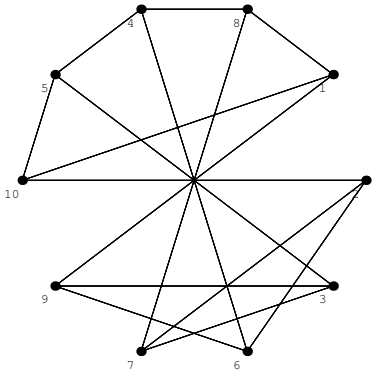


Abbildung 6.7: Eine Umnummerierung der Ecken

Umnummerierung der Ecken von `graph1` gewinnen (wenn eine solche existiert), durch `Isomorphism[graph1,graph2,All]` kann man sich alle entsprechenden Permutationen ausgeben lassen.

9. Man zeichne alle Bäume mit höchstens 5 Ecken.

   Lösung: In Abbildung 6.8 geben wir die acht verschiedenen (bis auf Isomorphie) Bäume mit bis zu 5 Ecken an.

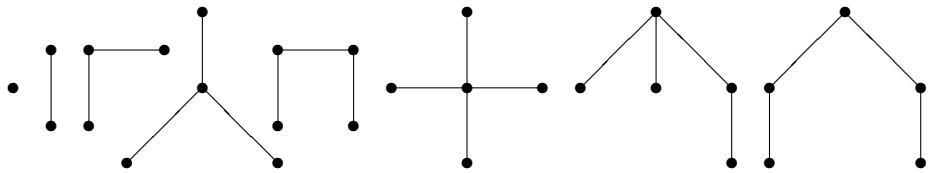


Abbildung 6.8: Bäume mit höchstens 5 Ecken

10. Ein Wurzelbaum (T, v_0) heißt ein *binärer Wurzelbaum*, wenn jeder innere Knoten höchstens zwei Kinder hat. Man zeige: Besitzt ein binärer Wurzelbaum t Blätter und hat er die Tiefe d, so ist $t \leq 2^d$.

    Lösung: Der Beweis wird durch vollständige Induktion nach der Tiefe d geführt. Ist $d = 0$, so hat der Baum nur einen Knoten, dieser Knoten ist ein Blatt. Es ist also $t = 1$,

die Aussage ist in diesem Falle richtig. Nun nehmen wir an, die Aussage sei für alle Wurzelbäume der Tiefe $d-1$ richtig. Ist die Wurzel v_0 ein Blatt, so hat v_0 genau ein Kind v_1 und der Wurzelbaum mit der Wurzel v_1 hat $t-1$ Blätter und die Tiefe $d-1$. Nach Induktionsannahme ist daher $t-1 \leq 2^{d-1}$ und folglich $t \leq 1+2^{d-1} \leq 2^d$, die Aussage in diesem Fall also richtig. Im andern Fall sei die Wurzel v_0 Vater der beiden Kinder v_1 und v_2. Man betrachte die bei v_1 und v_2 verwurzelten Unterbäume. Die Zahl ihrer Blätter sei t_1 bzw. t_2, ihre Tiefe d_1 bzw. d_2. Dann ist $t_1+t_2=t$ und $d_1, d_2 \leq d-1$. Nach Induktionsannahme ist $t_1 \leq 2^{d_1}$, $t_2 \leq 2^{d_2}$ und daher

$$t = t_1 + t_2 \leq 2^{d_1} + 2^{d_2} \leq 2 \cdot 2^{d-1} \leq 2^d,$$

insgesamt ist die Aussage bewiesen.

11. In der folgenden Tabelle sind die Entfernungen (in Hunderten von Meilen) von sechs Städten angegeben.

	Berlin	London	Moskau	Paris	Rom	Sevilla
Berlin	–	7	11	7	10	15
London	7	–	18	3	12	11
Moskau	11	18	–	18	20	27
Paris	7	3	18	–	9	8
Rom	10	12	20	9	–	13
Sevilla	15	11	27	8	13	–

In dem zugehörigen gewichteten vollständigen Graphen bestimme man einen minimalen aufspannenden Baum.

Lösung: Mit dem Kruskal-Verfahren erhält man offenbar das folgende "Movie" zur Bestimmung des minimalen aufspannenden Baumes. Die ersten drei Schritte sind in Abbildung 6.9 angegeben. Die Verbindung Berlin-Paris darf jetzt nicht gewählt werden,

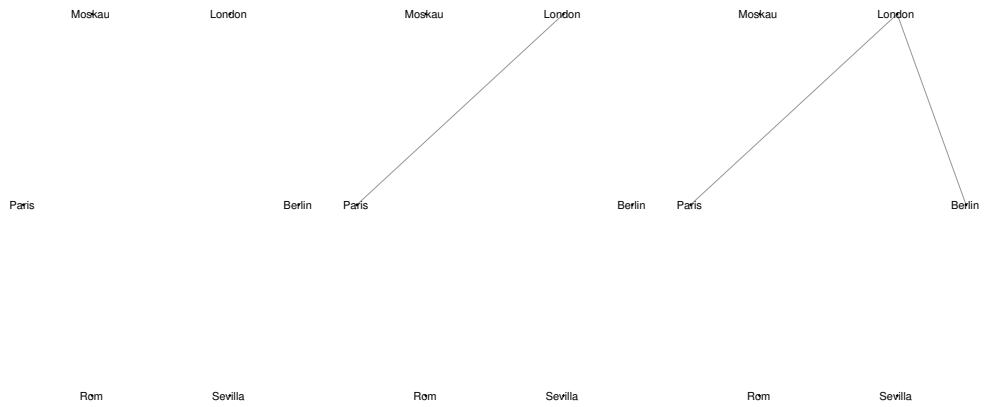


Abbildung 6.9: Die ersten drei Schritte im Kruskal-Verfahren

weil dadurch ein Kreis entstehen würde. Daher wird im nächsten Schritt Paris-Sevilla gewählt. Die restlichen Schritte sind in Abbildung 6.10 angegeben. Das Gesamtgewicht des aufspannenden Baumes ist 38. Man kann die entsprechende Aufgabe auch mit *Mathematica* (und dem Zusatzpaket `DiscreteMath‘Combinatorica‘`) lösen. Nach

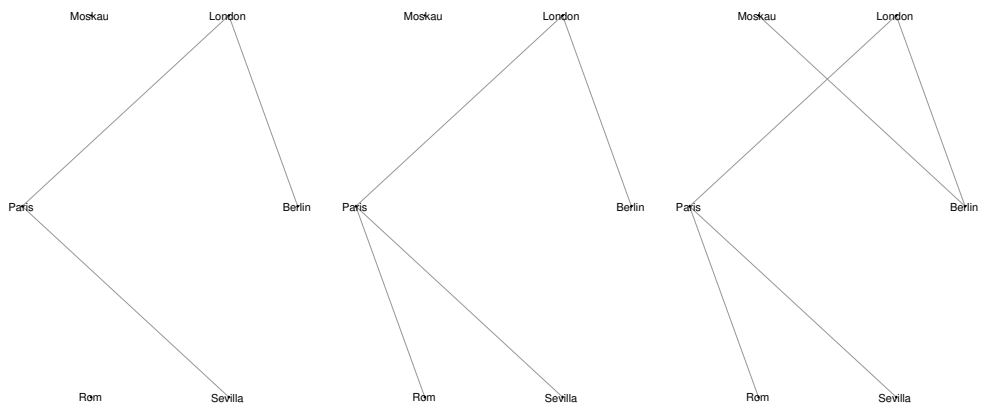


Abbildung 6.10: Die restlichen Schritte

```
g={{0,7,11,7,10,15},{7,0,18,3,12,11},{11,18,0,18,20,27},{7,3,18,0,9,8},
{10,12,20,9,0,13},{15,11,27,8,13,0}};
l=Vertices[CompleteGraph[6]];
gra=Graph[g,l];
ShowLabeledGraph[MinimumSpanningTree[gra],{"B","L","M","P","R","S"}]
```

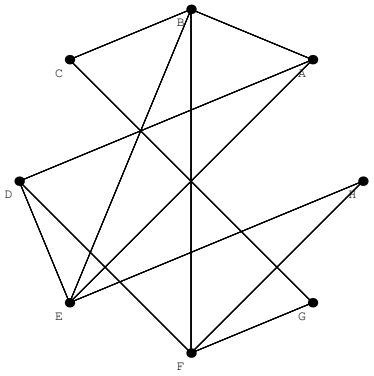
erhält man dasselbe Resultat.

12. Eine Firma hat sich zu überlegen, welches zusammenhängende Pipelinenetzwerk sie zwischen 7 Quellen $A, B, \ldots, G$ und einer Fabrik H bauen sollte. Die möglichen Pipelines und ihre Konstruktionskosten (in gewissen Geldeinheiten) sind gegeben durch

Pipeline	Kosten	Pipeline	Kosten
AB	23	CG	10
AE	17	DE	14
AD	19	DF	20
BC	15	EH	28
BE	30	FG	11
BF	27	FH	35

Welches Pipelinenetzwerk sollte gebaut werden und was sind seine Konstruktionskosten? Was ist die kostengünstigste Verbindung von der Quelle A zur Fabrik H?

Lösung: Die möglichen Pipelines zwischen den Quellen und der Fabrik mitsamt ihren Kosten ergeben einen gewichteten Graphen, der in Abbildung 6.11 mit seiner gewichteten Adjazenzmatrix dargestellt wird. Die Schritte zum Aufbau eines minimalen aufspannenden Baumes durch das Verfahren von Kruskal sind die folgenden: Zunächst werden die Kanten CG, FG, DE, BC und AE aufgenommen, da keine Kreise gebildet werden. Die Kante AD wird verworfen, da sonst ein Kreis ADE entstehen würde. Statt dessen wird DF aufgenommen. Auch die Kanten AB bzw. BF werden verworfen, weil dadurch die Kreise $ABCGFDE$ bzw. $BFGC$ entstehen würden. Schließlich wird EH als siebte und letzte Kante aufgenommen. Der minimale aufspannende Baum bzw. das kostengünstigste Pipelinenetzwerk ist in Abbildung 6.12 links angegeben. Die Kosten betragen 115 Geldeinheiten.



$$A := \begin{pmatrix} 0 & 23 & 0 & 19 & 17 & 0 & 0 & 0 \\ 23 & 0 & 15 & 0 & 30 & 27 & 0 & 0 \\ 0 & 15 & 0 & 0 & 0 & 0 & 10 & 0 \\ 19 & 0 & 0 & 0 & 14 & 20 & 0 & 0 \\ 17 & 30 & 0 & 14 & 0 & 0 & 0 & 28 \\ 0 & 27 & 0 & 20 & 0 & 0 & 11 & 35 \\ 0 & 0 & 10 & 0 & 0 & 11 & 0 & 0 \\ 0 & 0 & 0 & 0 & 28 & 35 & 0 & 0 \end{pmatrix}$$

Abbildung 6.11: Ein Pipelinenetzwerk und seine Gewichtung

Nun wenden wir das Verfahren von Dijkstra an. Wir erhalten der Reihe nach die Kanten AE, AD, AB, BC, DF, EH und CG und damit den in 6.12 rechts dargestellten aufspannenden Baum. An die Ecken haben wir die jeweiligen Kosten geschrieben. Hieraus liest man ab, dass AEH die günstigste Verbindung von A nach H ist.

13. Ein zusammenhängender Graph $G = (V, E)$ habe paarweise verschiedene Gewichte auf den Kanten. Man zeige, dass G einen *eindeutigen* minimalen aufspannenden Baum besitzt.

Lösung: Seien $T^* = (V, E^*)$ und $T' = (V, E')$ zwei verschiedene minimale aufspannende Bäume und $e = uv \in E^* \setminus E'$ eine zu T^*, nicht aber zu T' gehörende Kante. Auf dem eindeutigen Weg in T', welcher u mit v verbindet, liegt eine Kante e', welche die beiden Komponenten von $(V, E^* \setminus \{e\})$ miteinander verbindet. Dann sind $(V, (E^* \setminus \{e\}) \cup \{e'\})$ und $(V, (E' \setminus \{e'\}) \cup \{e\})$ jeweils aufspannende Bäume, von denen einer ein kleineres Gewicht als T^* bzw. T' hat (denn das Gewicht auf e ist verschieden von dem auf e').

14. Man zeige, dass ein bipartiter Graph mit einer ungeraden Zahl von Ecken nicht Hamiltonsch ist. Hiermit zeige man, dass der in Abbildung 6.13 angegebene Graph nicht Hamiltonsch ist.

Lösung: Die Eckenmenge V eines bipartiten Graphen $G = (V, E)$ kann so in zwei Mengen U und W zerlegt werden, dass jede Kante $e \in E$ in U startet und in W endet. Jeder Hamiltonsche Kreis muss zwischen diesen beiden Mengen alternieren und in der Menge enden, in der er startete. Daher müssen U und V dieselbe Anzahl von Elementen haben und daher $|V|$ gerade sein. Der in Abbildung 6.13 angegebene Graph ist bipartit (man nehme $U := \{1, 3, 5, 8, 10, 12\}$ und $V := \{2, 4, 6, 7, 9, 11, 13\}$) und hat eine ungerade Zahl von Ecken, ist also wegen der gerade eben bewiesenen Aussage nicht Hamiltonsch.

15. Man zeige, dass der Hyperwürfel Q_n, $n \geq 2$, ein Hamiltonscher Graph ist.

Lösung: Wir beweisen die Behauptung durch vollständige Induktion nach n. Für $n = 2$ ist die Behauptung richtig, denn $(0, 0)$, $(1, 0)$, $(1, 1)$, $(0, 1)$, $(0, 0)$ ist ein Hamilton-Kreis

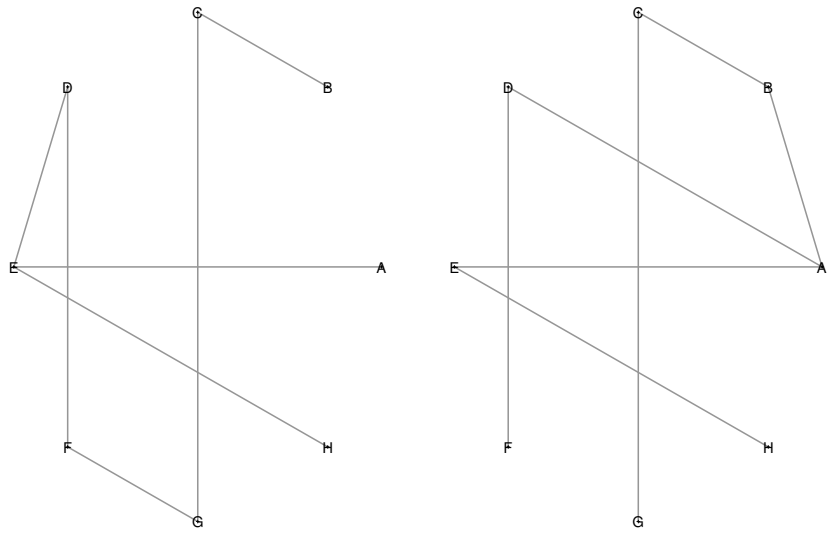


Abbildung 6.12: Billigstes Pipelinenetzwerk, Beste Verbindungen von A

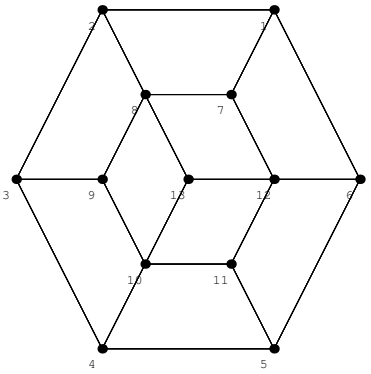


Abbildung 6.13: Bipartiter Graph mit ungerader Eckenzahl

in Q_2. Nun nehmen wir an, Q_n sei ein Hamiltonscher Graph. Die Anzahl der Ecken in Q_n ist 2^n. Sei $0, z_2, \ldots, z_{2^n-1}, z_{2^n}, 0$ ein Hamilton-Kreis in Q_n. Dann ist

$$(0,0),(z_2,0),\ldots,(z_{2^n-1},0),(z_{2^n},0),(z_{2^n},1),(z_{2^n-1},1),\ldots,(z_2,1),(0,1),(0,0)$$

ein Hamilton-Kreis in Q_{n+1}. Die Behauptung ist bewiesen.

16. Man zeige: Kann der Zusammenhang eines Graphen G durch die Entnahme einer einzigen Ecke und sämtlicher mit dieser Ecke inzidierender Kanten zerstört werden, so ist G kein Hamiltonscher Graph.

    Lösung: Sei $G = (V, E)$ ein zusammenhängender Graph und $v \in V$ eine Ecke mit der Eigenschaft, dass $(V \setminus \{v\}, E \setminus E(v))$ nicht zusammenhängend ist, wobei $E(v)$ die Menge der mit v inzidierenden Kanten in E ist. Wir nehmen an, es gäbe einen Kreis, der durch sämtliche Ecken von G führt, insbesondere durch v. Dieser Kreis verlässt und erreicht v jeweils über eine Kante aus $E(v)$, weitere Kanten aus $E(v)$ können im Kreis nicht vorkommen. Das wäre aber ein Widerspruch dazu, dass $(V \setminus \{v\}, E \setminus E(v))$ nicht zusammenhängend ist.

17. Sei G ein Graph mit n Ecken und der Eigenschaft, dass der Grad jeder Ecke mindestens $n/2$ ist. Man zeige, dass G Hamiltonsch ist.

    Lösung: Wir nehmen an, der Satz sei nicht wahr. Für ein gewisses n sei G also ein Graph mit n Ecken, deren Grad mindestens $n/2$ ist und der nicht Hamiltonsch ist. Wir können annehmen, dass G maximal viele Kanten besitzt, das Hinzufügen auch nur einer Kante zweier vorher nicht benachbarter Knoten (solche muss es geben, denn andernfalls wäre der Graph vollständig und damit Hamiltonsch) führe also auf einen Hamiltonschen Graphen. Seien y und z zwei nicht benachbarte Ecken. Da Hinzufügen von yz zur Existenz eines Hamiltonschen Kreises führt, gibt es einen Weg $y = x_1x_2\cdots x_n = z$ in G. Die Mengen

    $$\{i \in \{1,\ldots,n-1\} : y \text{ ist benachbart zu } x_{i+1}\}$$

    und

    $$\{i \in \{1,\ldots,n-1\} : z \text{ ist benachbart zu } x_i\}$$

    enthalten beide mindestens $n/2$ Elemente. Da beide in der $n-1$ Elemente enthaltenen Menge $\{1,\ldots,n-1\}$ enthalten sind, gibt es ein $i_0 \in \{1,\ldots,n-1\}$, welches in beiden Mengen enthalten sind. Dann wäre aber

    $$y = x_1x_2\cdots x_{i_0}z = x_nx_{n-1}\cdots x_{i_0+1}x_1 = y$$

    ein Kreis durch alle n Ecken von G, ein Widerspruch dazu, dass durch G ein Gegenbeispiel gegeben ist.

18. Gegeben sei der in Abbildung 6.14 dargestellte gewichtete Digraph mit der Quelle q und

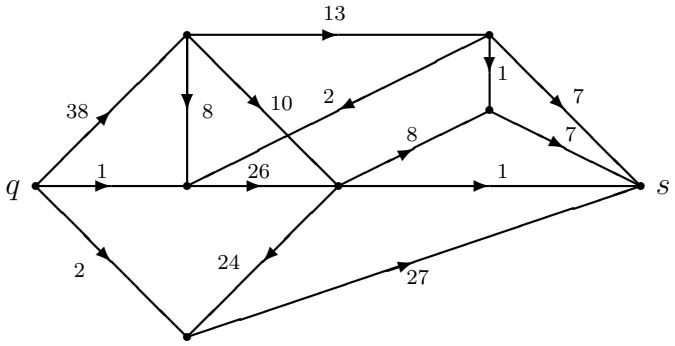


Abbildung 6.14: Ein gewichteter Digraph

der Senke s. Mit Hilfe von *Mathematica* bestimme man einen maximalen Fluss sowie einen kürzesten Weg von der Quelle $q = 1$ zu den übrigen Knoten.

Lösung: Wir nummerieren die 8 Knoten von links nach rechts und von oben nach unten. Wir definieren den Graphen `gra` in *Mathematica* durch

```
adj={{0,38,1,2,0,0,0,0},{0,0,8,0,10,13,0,0},{0,0,0,0,26,0,0,0},
{0,0,0,0,0,0,0,27},{0,0,0,24,0,0,8,1},{0,0,2,0,0,0,1,7},
{0,0,0,0,0,0,0,7},{0,0,0,0,0,0,0,0}};
gra=Graph[adj,Vertices[Cycle[8]]];
```

Anschließendes `NetworkFlow[gra,1,8]` ergibt den Wert des maximalen Flusses, nämlich 31. Nach `NetworkFlowEdges[gra,1,8]` erhält man als Output die Matrix

$$\begin{pmatrix} 0 & 28 & 1 & 2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 8 & 0 & 10 & 10 & 0 & 0 \\ 0 & 0 & 0 & 0 & 11 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 22 \\ 0 & 0 & 0 & 20 & 0 & 0 & 0 & 1 \\ 0 & 0 & 2 & 0 & 0 & 0 & 1 & 7 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix},$$

was dem maximalen Fluss in Abbildung 6.15 entspricht. Der Digraph `gra` sowie das

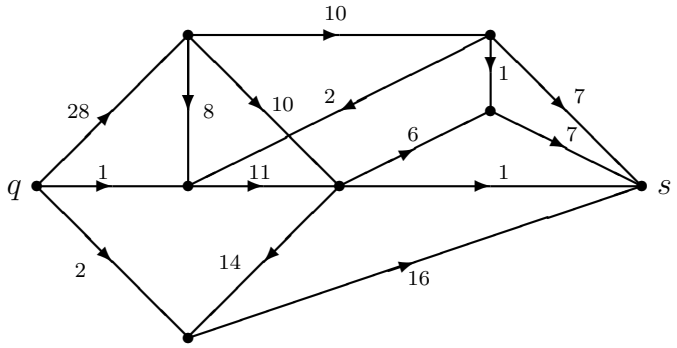


Abbildung 6.15: Ein maximaler Fluss im Digraphen

Ergebnis von `ShortestPathSpanningTree[gra,1]` sind in Abbildung 6.16 links bzw. rechts angegeben. Hieraus liest man ab, dass $\{1,2\}$, $\{1,3\}$, $\{1,4\}$, $\{1,3,5\}$, $\{1,2,6\}$,

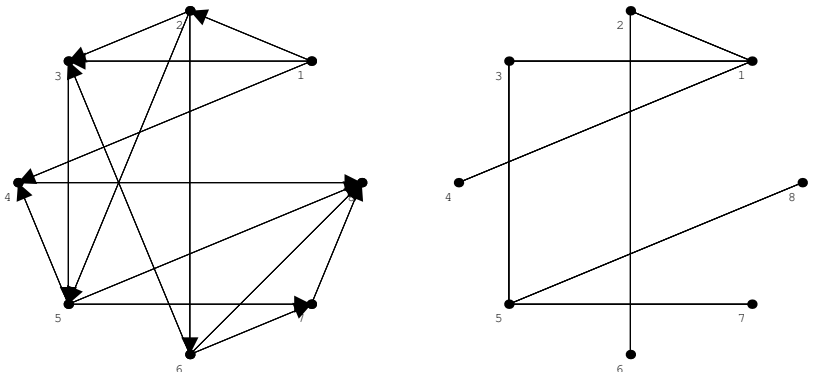


Abbildung 6.16: Ein Digraph und ein kürzester aufspannender Baum

$\{1.3,5,7\}$ und $\{1,3,5,8\}$ die kürzesten Wege zu den Knoten des Digraphen sind.

19. Zeige[12], dass ein bipartiter Graph $G = (U \cup W, E)$ mit $|U| = |W| = n$ und $|E| > (m-1)n$ ein Matching der Größe m enthält.

[12]Diese Aufgabe haben wir M. Aigner (1996, S. 147) entnommen.

Lösung: Wir zeigen, dass für jedes $T \subset U$ die Beziehung $|N(T)| \geq m - n + |T|$ gilt. Ist dies nämlich bewiesen, so ist $\max_{T \subset U}(|T| - |N(T)|) \leq n - m$, wegen Satz 8.3 folglich

$$m \leq n - \max_{T \subset U}(|T| - |N(T)|) = |U| - \max_{T \subset U}(|T| - |N(T)|) = m(G),$$

so dass es ein Matching der Mächtigkeit m gibt. Sei also $T \subset U$ beliebig. Wir setzen $r := |T|$ und $s := N(T)$. Dann ist

$$(m-1)n < |E| \leq rs + (n-r)n \leq n(s+n-r),$$

also $m - 1 < s + n - r$ bzw. $s \geq m - n + r$. Genau das war zu zeigen.

20. Sei $A = (a_{ij}) \in \mathbb{R}^{n \times n}$ eine Matrix mit $a_{ij} \in \{0, 1\}$, $1 \leq i, j \leq n$. Unter einer *Linie* in einer Matrix verstehe man eine Zeile oder eine Spalte. Man zeige, dass die Minimalzahl der alle Einsen enthaltenden Linien gleich der Maximahlzahl der Einsen ist, von denen nicht zwei auf einer Linie liegen.

Beispiel: In

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

sind alle Einsen in drei Linien (etwa erste beide Zeilen, zweite Spalte) enthalten, während es maximal drei Einsen gibt, von denen keine zwei auf einer Linie liegen, z. B. die Einsen in den Positionen $(1, 4)$, $(2, 1)$ und $(4, 2)$.

Hinweis: Man wende Satz 8.5 auf einen geeignet definierten bipartiten Graphen an.

Lösung: Wir definieren einen bipartiten Graphen $G = (U \cup W, E)$ durch die Eckenmengen $U := \{1, \ldots, n\}$ und $W := \{1, \ldots, n\}$ und die Kantenmange $E := \{(i, j) \in U \times W : a_{ij} = 1\}$. Hierbei wird U als Menge der Zeilen, W als Menge der Spalten in A interpretiert, so dass $U \cup W$ die Menge der Linien ist und Kanten die Positionen in der Matrix sind, in der Einsen stehen.

Für die obige Matrix erhalten wir z. B. den in Abbildung 6.17 angegebenen bipartiten Graphen. Gibt man diesen Graphen in *Mathematica* ein, etwa durch

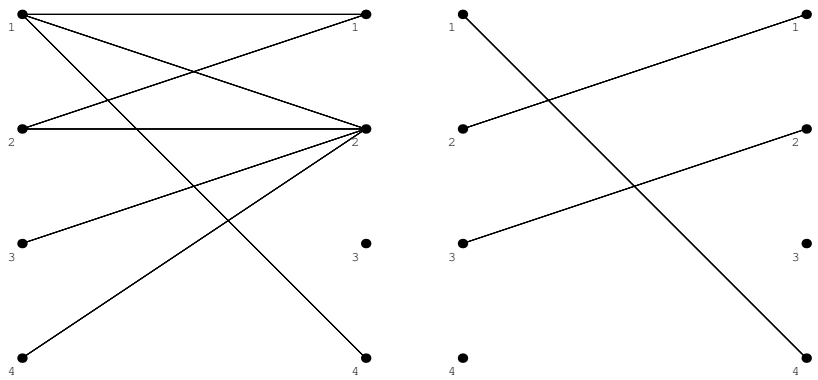


Abbildung 6.17: Ein zu $0, 1$-Matrix gehörender bipartiter Graph

```
g={{5,6,8},{5,6},{6},{6},{1,2},{1,2,3,4},{},{1}};
l={{0,3},{0,2},{0,1},{0,0},{3,3},{3,2},{3,1},{3,0}};
gra=FromAdjacencyLists[g,l];
```

und ruft man anschließend `BipartiteMatching[gra]` auf, so erhält man das in Abbildung 6.17 rechts dargestellte Maximum-Matching. `MinimumVertexCover[gra]` liefert die Spalten 1, 2 und 4. In der Tat werden durch diese drei Spalten alle Einsen überdeckt, während zu dem angegebenen Maximum-Matching die Positionen $(1,4)$, $(2,1)$ und $(3,2)$ gehören, an deren Stelle Einsen stehen, wobei nicht zwei zu einer Linie gehören. Man vermutet daher, dass die behauptete Aussage eng mit dem Satz 8.5 verwandt ist, dass nämlich in einem bipartiten Graphen die Maximalzahl der Kanten in einem Matching (also die Zahl der Kanten in einem Maximum-Matching) gleich ist der Minimalzahl der Elemente eines Trägers des bipartiten Graphen.

Nun ist der Beweis der Aussage einfach. Sei $D \subset U \cup W$ ein Träger von G. Nach Definition heißt dies, dass es zu jeder Kante (also jeder Eins in der Matrix) eine Ecke (also eine Linie) gibt, die auf dieser Kante liegt. Durch einen Träger sind also Linien gegeben, die alle Einsen überdecken. Ist dagegen $F \subset E$ ein Matching, so sind hierdurch Positionen in der Matrix gegeben, in denen Einsen stehen. Weil bei einem Matching keine zwei Kanten eine Ecke gemein haben, sind die durch ein Matching gegebenen Einsen so in der Matrix verteilt, dass nicht zwei auf einer Linie stehen. Damit ist die behauptete Aussage als Korollar zum Satz 8.5 enttarnt.

21. In einem Obstkorb liegen ein Apfel, eine Banane, ein Pfirsich und eine Mango. Vier Kindern, nämlich Erika, Frank, Gisela und Hubert, soll jeweils eine Frucht gegeben werden. Erika mag Pfirsiche und Mangos, Frank mag Äpfel und Pfirsiche, Gisela Bananen und Pfirsiche, Hubert mag Äpfel, Bananen und Mangos. Wie sollte das Obst verteilt werden?

Lösung: In naheliegender Weise definiert man einen bipartiten Graphen: Die Ecken sind die Kinder und die Früchte, eine Kante wird zwischen Kind und Frucht gezogen, wenn das Kind die Frucht mag. Wir geben ihn in Abbildung 6.18 links an. Ein Maximum-

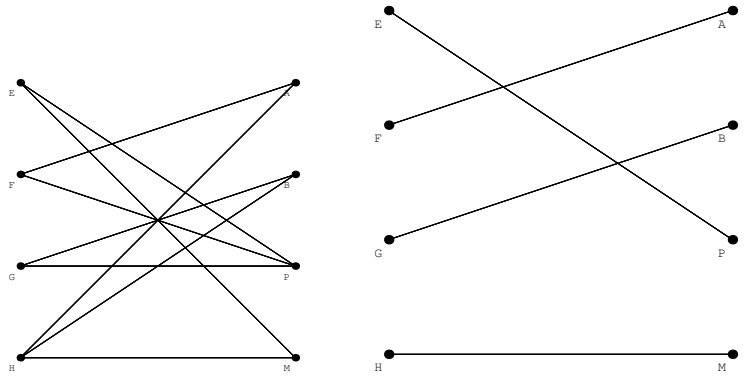


Abbildung 6.18: Ein Obst-Zuteilungsproblem

Matching ist in 6.18 rechts angegeben. Danach erhält Erika den Pfirsich, Frank den Apfel, Gisela die Banane und Hubert die Mango.

22. Man löse das Zuordnungsproblem mit der Kostenmatrix

$$C := \begin{pmatrix} 6 & 3 & 2 & 5 & 2 & 2 \\ 8 & 4 & 3 & 2 & 3 & 4 \\ 4 & 4 & 3 & 4 & 4 & 5 \\ 2 & 7 & 4 & 5 & 4 & 3 \\ 8 & 6 & 4 & 6 & 6 & 6 \\ 0 & 2 & 3 & 3 & 2 & 2 \end{pmatrix}.$$

Lösung: In der Initialierungsphase erhält man die Matrizen

$$C = \begin{pmatrix} 4 & 1 & 0 & 3 & 0 & 0 \\ 6 & 2 & 1 & 0 & 1 & 2 \\ 1 & 1 & 0 & 1 & 1 & 2 \\ 0 & 5 & 2 & 3 & 2 & 1 \\ 4 & 2 & 0 & 2 & 2 & 2 \\ 0 & 2 & 3 & 3 & 2 & 2 \end{pmatrix}, \qquad C = \begin{pmatrix} 4 & 0 & 0 & 3 & \mathbf{0} & 0 \\ 6 & 1 & 1 & \mathbf{0} & 1 & 2 \\ 1 & \mathbf{0} & 0 & 1 & 1 & 2 \\ \mathbf{0} & 4 & 2 & 3 & 2 & 1 \\ 4 & 1 & \mathbf{0} & 2 & 2 & 2 \\ 0 & 1 & 3 & 3 & 2 & 2 \end{pmatrix}.$$

Die bei dieser Reduktion anfallenden Kosten betragen 14. Durch $U^* \cup W^*$ mit $U^* := \{1, 2, 3, 5\}$ und $W^* := \{1\}$ ist eine minimale Bedeckung der Nullen gegeben, eine zugehörige 0-Diagonale der Länge 5 ist in der obigen Matrix durch fetten Druck gekennzeichnet. Der kleinste unbedeckte Eintrag ist $d = 1$, dieser ist von allen unbedeckten Einträgen zu subtrahieren und zu allen doppelt bedeckten Elementen zu addieren. Man erhält

$$C = \begin{pmatrix} 5 & 0 & 0 & 3 & \mathbf{0} & 0 \\ 7 & 1 & 1 & \mathbf{0} & 1 & 2 \\ 2 & \mathbf{0} & 0 & 1 & 1 & 2 \\ 0 & 3 & 1 & 2 & 1 & \mathbf{0} \\ 5 & 1 & \mathbf{0} & 2 & 2 & 2 \\ \mathbf{0} & 0 & 2 & 2 & 1 & 1 \end{pmatrix}.$$

Eine 0-Diagonale der Länge 6 ist durch fetten Druck verdeutlicht. Daher hat auch die minimale Bedeckung der Nullen eine Länge 6. Wir sind fertig, durch

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

ist eine Lösung des gestellten Zuordnungsproblem gegeben, die Kosten sind $14+1 = 15$.

23. Sei $G = (V, E)$ ein zusammenhängender, planarer Graph mit $n := |V| \geq 3$. Dann existiert eine Ecke mit einem Grad ≤ 5.

Hinweis: Mache einen Widerspruchsbeweis und zähle (wie im Handshaking-Lemma) die Anzahl der Elemente in $P := \{(x, e) \in V \times E : x \in e\}$.

Lösung: Mit $d(x)$ werde der Grad der Ecke $x \in V$ bezeichnet. Angenommen, es sei $d(x) \geq 6$ für alle $x \in V$. Man definiere (wie beim Beweis des Handshaking-Lemmas) $P := \{(x, e) \in V \times E : x \in e\}$. Nach Annahme ist einerseits $|P| \geq 6n$, andererseits ist $|P| = 2m$ mit $m := |E|$, insbesondere also $m \geq 3n$. Dies ist ein Widerspruch zum ersten Teil von Korollar 9.5.

24. Ist der in Abbildung 6.19 links angegebene Graph planar?

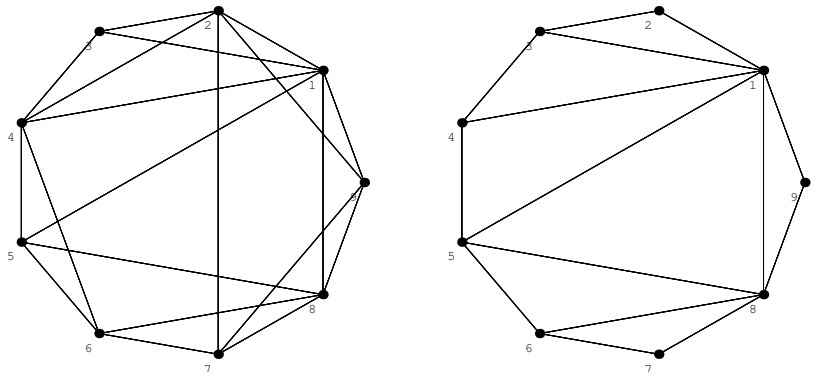


Abbildung 6.19: Ist der Graph links planar?

Lösung: Wir wollen zeigen, dass der angegebene Graph planar ist. Ein planarer Untergraph ist offensichtlich in Abbildung 6.19 rechts angegeben. Gegenüber dem gegebenen Graphen fehlen hier noch die Kanten 24, 27, 29, 46 und 79. Diese 5 Kanten können aber leicht "außen herum" überschneidungsfrei gelegt werden.

25. Sei $G = (V, E)$ ein zusammenhängender, planarer Graph mit $n := |V| \geq 5$ Ecken, $m := |E|$ Kanten, ferner habe jeder Kreis eine Länge ≥ 5. Man zeige:

    (a) Es ist $m \leq \frac{5}{3}(n-2)$.

    (b) Der in Abbildung 6.20 dargestellte Petersen-Graph ist nicht planar.

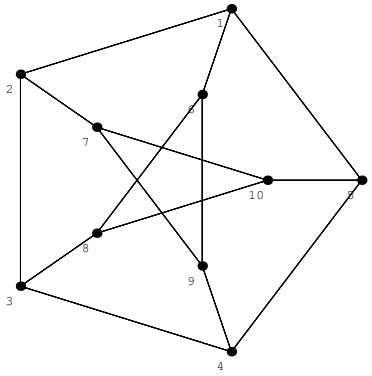


Abbildung 6.20: Der Petersen-Graph ist nicht planar

    (c) Man bestimme eine Eckenfärbung des Petersen-Graphen mit möglichst wenig Farben, wobei natürlich benachbarte Ecken unterschiedliche Farben haben sollen.

Lösung: Man betrachte einen zu G isomorphen ebenen Graphen. Jedes Land wird von mindestens fünf Kanten begrenzt, jede Kante gehört zum Rand von höchstens zwei Ländern. Bezeichnet f die Anzahl der Länder, so ist daher (siehe auch Korollar 9.5) $5f \leq 2m$. Wegen der Eulerschen Polyederformel ist $f = m - n + 2$, Einsetzen liefert die Behauptung.

Der Petersen-Graph besitzt offenbar keinen Kreis mit kleinerer Länge als 5. Er besitzt $n = 10$ Ecken und $m = 15$ Kanten. Es ist $15 > \frac{5}{3}(10 - 2)$, der erste Teil der Aufgabe liefert daher, dass der Petersen-Graph nicht planar ist.

Mit Hilfe von `VertexColoring` haben wir erhalten, dass drei Farben zum Färben der 10 Knoten des Petersen-Graphen genügen. Mögliche Gruppen gleich gefärbter Ecken sind z. B. $\{1, 3, 9, 10\}$, $\{2, 4, 6\}$ und $\{5, 7, 8\}$.

26. Sei $G = (V, E)$ ein Graph mit $d := \max_{x \in V} d(x)$, wobei $d(x)$ den Grad der Ecke $x \in V$ bedeutet. Dann ist die chromatische Zahl von G kleiner oder gleich $d + 1$. D. h. es ist eine Färbung der Ecken von G durch höchstens $d + 1$ Farben derart möglich, dass benachbarte Ecken verschieden gefärbt sind.

    Hinweis: Man mache einen Induktionsbeweis nach der Anzahl n der Ecken von G und schließe wie beim Beweis des Sechsfarbensatzes.

    Lösung: Genau wie beim Beweis des Sechs- und des Fünffarbensatzes ist der Induktionsanfang trivial. Wir nehmen an, die Aussage sei für Graphen mit weniger als n Ecken richtig. Für den Induktionsschluss sei G ein Graph mit n Ecken und v eine Ecke mit dem Grad d, also mit d Nachbarn. Man gewinne den Graphen H aus G, indem man die Ecke v und die Kanten zu den d Nachbarn weglässt. Da H weniger als n Kanten besitzt und der Grad jeder Ecke in H kleiner oder gleich d ist, ist H (bzw. die Ecken von H) nach Voraussetzung durch $d + 1$ Farben färbbar. Die Ecke v färbe man nun mit einer Farbe, die noch nicht von den d Nachbarn benutzt wurde. Dadurch ist die gewünschte Färbung von G mit höchstens $d + 1$ Farben erreicht.

27. Ein Zoo muss geeignete Umgebungen für acht Tierarten $A, B, \ldots, H$ bestimmen, wobei aus Sicherheitsgründen gewisse Tiere von anderen getrennt werden müssen. Es soll die minimale Anzahl der "Umgebungen" bestimmt werden, in denen die Tiere sicher leben können. In der folgenden Tabelle gibt ein $*$ eine notwendige Trennung an.

	A	B	C	D	E	F	G	H
A	$-$	$*$	$-$	$-$	$*$	$*$	$-$	$*$
B	$*$	$-$	$*$	$-$	$-$	$*$	$-$	$*$
C	$-$	$*$	$-$	$*$	$-$	$*$	$*$	$*$
D	$-$	$-$	$*$	$-$	$*$	$*$	$*$	$-$
E	$*$	$-$	$-$	$*$	$-$	$*$	$*$	$-$
F	$*$	$*$	$*$	$*$	$*$	$-$	$-$	$-$
G	$-$	$-$	$*$	$*$	$*$	$-$	$-$	$*$
H	$*$	$*$	$*$	$-$	$-$	$-$	$*$	$-$

    Man stelle den zugehörigen Graphen auf und bestimme dessen chromatische Zahl und eine geeignete Verteilung der Tiere. Hierbei kann *Mathematica* und das Zusatzpaket `DiscreteMath‘Combinatorica‘` benutzt werden.

    Lösung: Nach

```
adjdata={{2,5,6,8},{1,3,6,8},{2,4,6,7,8},{3,5,6,7},{1,4,6,7},
{1,2,3,4,5},{3,4,5,8},{1,2,3,7}};
gra=FromAdjacencyLists[adjdata];
ShowLabeledGraph[gra,{"A","B","C","D","E","F","G","H"}]
```

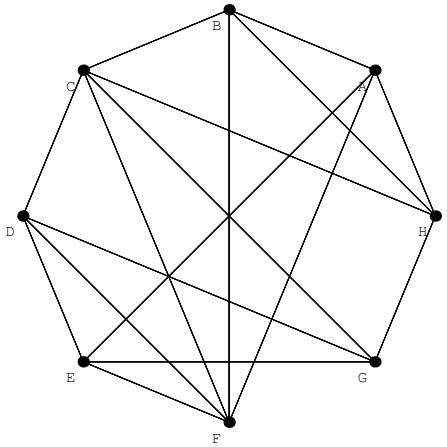


Abbildung 6.21: Ein Verträglichkeits-Graph

erhalten wir den in Abbildung 6.21 angegebenen Graphen. Dieser ist planar (Beweis?), nach `VertexColoring[gra]` erhält man allerdings nur eine Färbung der Ecken mit fünf Farben (und nicht etwa, wie der Vierfarbensatz aussagt, durch vier Farben), als Einteilung wird $\{A, C\}$, $\{B, D\}$, $\{E\}$, $\{F, H\}$, $\{G\}$ ausgegeben. Dass keine Lösung gefunden wurde, ist kein Widerspruch, denn `VertexColoring` benutzt eine Heuristik, die i. Allg. eine gute Näherung, aber nicht notwendig eine minimale Lösung bestimmt. Es ist leicht, eine Lösung mit vier Farben anzugeben, z. B. $\{A, C, D, G\}$, $\{B, E\}$, $\{F, G\}$, $\{H\}$. Die chromatische Zahl zu bestimmen ist zwar ein "schweres Problem", mit dem Befehl `ChromaticNumber[gra]` erhält man aber (in diesem Fall) sehr schnell die Antwort vier.

6.3 Aufgaben zu Kapitel 4

6.3.1 Aufgaben zu Abschnitt 4.1

1. Das Problem, einen kürzesten Weg von einem Knoten s zu einem anderen Knoten t in einem gewichteten Digraphen $(\mathcal{N}, \mathcal{A})$ zu bestimmen, kann als eine lineare Optimierungsaufgabe formuliert werden. Man stelle das dazu lineare Programm auf.

   Lösung: Diese Aufgabe ist ganz einfach. Mit der Knoten-Pfeil-Inzidenzmatrix $A \in \mathbb{R}^{|\mathcal{N}|\times|\mathcal{A}|}$, dem Vektor $b \in \mathbb{R}^{|\mathcal{N}|}$ mit

   $$b_k := \begin{cases} 1, & \text{falls} \quad k = s, \\ -1, & \text{falls} \quad k = t, \\ 0, & \text{sonst} \end{cases}$$

   und dem Kostenvektor $c \in \mathbb{R}^{|\mathcal{A}|}$ (nach entsprechender Nummerierung der Knoten und Pfeile) kann dem Problem des kürzesten Pfades (KP) in $(\mathcal{N}, \mathcal{A})$ zwischen s und t das lineare Programm

   $$\text{(P)} \qquad \text{Minimiere} \quad c^T x \quad \text{auf} \quad M := \{x \in \mathbb{R}^{|\mathcal{A}|} : x \geq 0,\ Ax = b\}$$

zugeordnet werden. Das duale Problem ist dann

$$\text{(D)} \qquad \begin{cases} \text{Maximiere} \quad u_s - u_t \quad \text{auf} \\ N := \{u \in \mathbb{R}^{|\mathcal{N}|} : u_i - u_j \le c_{ij},\ (i,j) \in \mathcal{A}\}. \end{cases}$$

Lässt man in (P) die Zeile zum Knoten t fort, damit die Koeffizientenmatrix (bei einem zusammenhängenden Digraphen) vollen Rang $|\mathcal{N}| - 1$ besitzt, so ist das duale Problem zu

$$\text{(}\tilde{\text{P}}\text{)} \qquad \text{Minimiere} \quad c^T x \quad \text{auf} \quad M := \{x \in \mathbb{R}^{|\mathcal{A}|} : x \ge 0,\ \tilde{A}x = \tilde{b}\}$$

(hierbei gehen $\tilde{A}$ und $\tilde{b}$ aus A bzw. b durch Streichen der Zeile bzw. Komponente zum Knoten t hervor) gerade

$$\text{(}\tilde{\text{D}}\text{)} \qquad \begin{cases} \text{Maximiere} \quad \tilde{u}_s \quad \text{auf} \\ \tilde{N} := \{\tilde{u} \in \mathbb{R}^{|\mathcal{N}|-1} : \tilde{u}_i - \tilde{u}_j \le c_{ij},\ (i,j) \in \mathcal{A},\ j \ne t,\ \tilde{u}_i \le c_{it},\ (i,t) \in \mathcal{A}\}. \end{cases}$$

2. Man betrachte den in Abbildung 6.22 dargestellten Digraphen $(\mathcal{N}, \mathcal{A})$. Nach einer geeig-

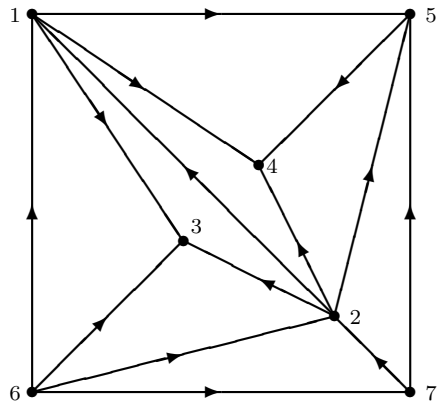


Abbildung 6.22: Ein gerichteter Graph

neten Nummerierung der Pfeile stelle man seine Knoten-Pfeil-Inzidenzmatrix auf. Man bestimme wenigstens zwei $(\mathcal{N}, \mathcal{A})$ (als ungerichteten Graphen betrachtet) aufspannende Bäume und deren Knoten-Pfeil-Inzidenzmatrix. Man zeige, dass diese nach Weglassen der letzten Zeile nichtsingulär sind.

Lösung: Wir nummerieren die Pfeile der Reihe nach von links nach rechts und von oben nach unten, also: $(6,1)$, $(1,5)$, $(1,4)$, $(2,1)$ usw. Als Knoten-Pfeil-Inzidenzmatrix erhalten wir dann

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

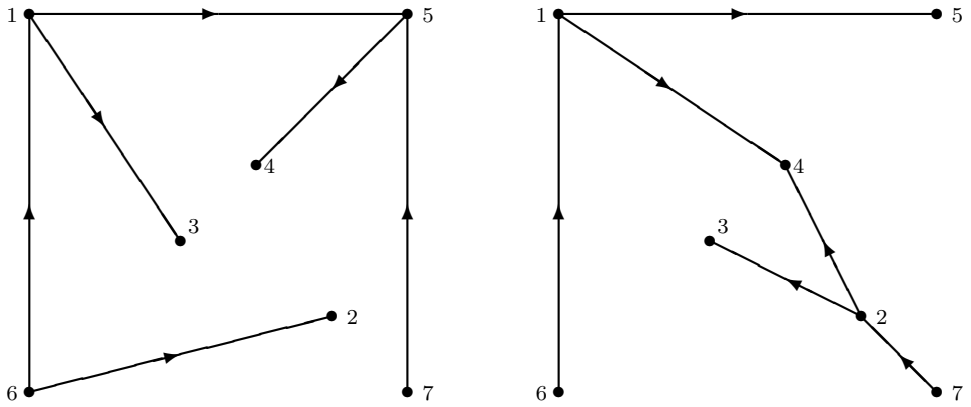


Abbildung 6.23: Zwei aufspannende Bäume

In Abbildung 6.23 geben wir zwei aufspannende Bäume an. Als Inzidenzmatrizen dieser (gerichteten) Bäume erhalten wir

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

Lässt man hier die letzte Zeile fort und bezeichnet die entsprechenden 6×6-Matrizen mit B, so erhält man als Inverse (wir haben die *Mathematica*-Funktion `Inverse` benutzt)

$$B^{-1} = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 0 & 0 & 1 \\ 0 & 0 & -1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 0 & 0 \\ -1 & -1 & -1 & -1 & -1 & -1 \end{pmatrix}, \quad B^{-1} = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & -1 & 0 \\ 1 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & -1 & 0 & 0 & 0 \\ -1 & 0 & 0 & -1 & -1 & -1 \\ -1 & -1 & -1 & -1 & -1 & -1 \end{pmatrix}.$$

Die Matrix B ist also nicht nur nichtsingulär. Ihre Inverse hat sogar nur Einträge aus $\{0, 1, -1\}$.

3. Sei $(\mathcal{N}, \mathcal{A})$ ein (schwach) zusammenhängender Digraph. Sei $A \in \mathbb{R}^{m\times n}$ mit $m := |\mathcal{N}|$ und $n := |\mathcal{A}|$ (nach geeigneter Nummerierung der Knoten und Pfeile) die zugehörige Knoten-Pfeil-Inzidenzmatrix. Man zeige, dass Rang $(A) = m - 1$. Genauer zeige man, dass die Spalten von A, die zu einem $(\mathcal{N}, \mathcal{A})$ aufspannenden Baum gehören, linear unabhängig sind.

   Lösung: Sei $\mathcal{A} = \{l_1, \ldots, l_n\}$ und $J \subset \{1, \ldots, n\}$ eine Indexmenge mit $|J| = m - 1$ und der Eigenschaft, dass $(\mathcal{N}, \mathcal{A}')$ mit $\mathcal{A}' := \{l_j\}_{j\in J}$ ein $(\mathcal{N}, \mathcal{A})$ aufspannender Baum ist. Wir wollen zeigen, dass A_J den Rang $m - 1$ besitzt. Man beachte, dass A_J die Knoten-Pfeil-Inzidenzmatrix zum Baum $(\mathcal{N}, \mathcal{A}')$ ist. Wir nehmen an, es sei $A_J\alpha_J = 0$.

Jeder Baum besitzt mindestens einen Knoten, der den Grad 1 hat bzw. mit nur einem Pfeil (als Anfangs- oder Endknoten) inzidiert. Das ist eine unmittelbare Folge des Handshaking-Lemmas. Daher gibt es in A_J mindestens eine Zeile mit genau einem von Null verschiedenen (1 oder -1) Eintrag. Ist dies die k-te Zeile bzw. der k-te Knoten und der Eintrag in der $j(k)$-ten Spalte, so ist $e_k^T A_J = \pm e_{j(k)}^T$ und daher die entsprechende Komponente von α_J gleich Null. Entfernt man diesen Knoten und den mit ihm inzidierenden Pfeil aus dem Baum, so erhält man wieder einen Baum, dessen Inzidenzmatrix man aus A_J dadurch erhält, dass man die k-te Zeile und die Spalte entfernt, in der der einzige von Null verschiedene Eintrag steht. In dieser Weise kann man fortfahren und erhält nach endlich vielen Schritten, dass $\alpha_J = 0$.

4. Man[13] betrachte das Zuordnungsproblem, bei dem die Kostenmatrix $C = (c_{ij})$ durch $c_{ij} = a_i b_j$ (also $C = ab^T$) gegeben ist, wobei

$$a_1 \le \cdots \le a_n, \qquad b_n \le \cdots \le b_1.$$

Man zeige, dass durch die Identität $X = I$ (bzw. $x_{ij} = \delta_{ij}$) eine optimale Zuordnung gegeben ist. Unter welcher zusätzlichen Bedingung ist diese eindeutig?

Lösung: Die Aufgabe besteht darin,

$$c(p) := \sum_{i=1}^{n} a_i b_{p(i)}$$

unter allen Permutationen $p = (p(1), \ldots, p(n))$ der Zahlen $\{1, \ldots, n\}$ zu minimieren. Wir zeigen, dass es eine optimale Permutation p^* mit $p^*(1) < \cdots < p^*(n)$ bzw. $p^*(i) = i$, $i = 1, \ldots, n$, gibt. Hierzu nehmen wir an, p sei eine optimale Permutation und für ein Paar (i, k) mit $1 \le i < k \le n$ sei $p(k) < p(i)$. Dann definieren wir die Permutation q durch

$$q = (p(1), \ldots, p(i-1), p(k), p(i+1), \ldots, p(k-1), p(i), p(k+1), \ldots, p(n)).$$

Dann ist

$$c(q) - c(p) = a_i b_{p(k)} + a_k b_{p(i)} - (a_i b_{p(i)} + a_k b_{p(k)}) = (\underbrace{a_k - a_i}_{\ge 0})(\underbrace{b_{p(i)} - b_{p(k)}}_{\le 0}) \le 0,$$

also auch q optimal. Nach endlich vielen Schritten erhält man die Behauptung. Außerdem erhält man die Eindeutigkeit, wenn

$$a_1 < \cdots < a_n, \qquad b_n < \cdots < b_1.$$

Damit ist die Aufgabe gelöst.

[13] Bei

D. P. Bertsekas (1998) *Network Optimization. Continuous and Discrete Models*. Athena Scientific, Belmont

wird die Aussage dieser Aufgabe Hardy zugeschrieben.

6.3.2 Aufgaben zu Abschnitt 4.2

1. Sei $A \in \mathbb{R}^{m\times n}$, $b \in \mathbb{R}^m$. Hiermit definiere man

$$M' := \{x \in \mathbb{R}^n : x \geq 0,\ Ax \leq b\}, \qquad M := \{z \in \mathbb{R}^{n+m} : z \geq 0,\ (\ A \quad I\ )z = b\}.$$

Man zeige, dass $x \in M'$ genau dann eine Ecke von M' ist, wenn

$$z := \begin{pmatrix} x \\ b - Ax \end{pmatrix}$$

eine Ecke von M ist.

Lösung: Sei zunächst $x \in M'$ eine Ecke von M'. Ist

$$\begin{pmatrix} x_1 \\ y_1 \end{pmatrix}, \begin{pmatrix} x_2 \\ y_2 \end{pmatrix} \in M \setminus \left\{ \begin{pmatrix} x \\ b - Ax \end{pmatrix} \right\},\ \lambda \in (0,1),$$

so ist

$$(1-\lambda)\begin{pmatrix} x_1 \\ y_1 \end{pmatrix} + \lambda \begin{pmatrix} x_2 \\ y_2 \end{pmatrix} = (1-\lambda)\begin{pmatrix} x_1 \\ b - Ax_1 \end{pmatrix} + \lambda \begin{pmatrix} x_2 \\ b - Ax_2 \end{pmatrix} \neq \begin{pmatrix} x \\ b - Ax \end{pmatrix},$$

denn andernfalls ist $x = (1-\lambda)x_1 + \lambda x_2$. Da $x_1, x_2 \in M'$, $\lambda \in (0,1)$ und x eine Ecke von M' ist, ist $x_1 = x_2 = x$ und folglich $y_1 = y_2 = b - Ax$, ein Widerspruch. Umgekehrt sei

$$z := \begin{pmatrix} x \\ b - Ax \end{pmatrix}$$

eine Ecke von M. Sind dann $x_1, x_2 \in M' \setminus \{x\}$, $\lambda \in (0,1)$, so ist $x \neq (1-\lambda)x_1 + \lambda x_2$, denn andernfalls wäre

$$z = (1-\lambda)\begin{pmatrix} x_1 \\ b - Ax_1 \end{pmatrix} + \lambda \begin{pmatrix} x_2 \\ b - Ax_2 \end{pmatrix},$$

ein Widerspruch.

2. Sei

$$A = \begin{pmatrix} a_1^T \\ \vdots \\ a_m^T \end{pmatrix} \in \mathbb{R}^{m\times n}, \qquad b = \begin{pmatrix} b_1 \\ \vdots \\ b_m \end{pmatrix} \in \mathbb{R}^m$$

und

$$M := \{x \in \mathbb{R}^n : Ax \leq b\}.$$

Man zeige, dass $x \in M$ genau dann eine Ecke von M ist, wenn mit

$$I(x) := \{i \in \{1, \ldots, m\} : a_i^T x = b_i\}$$

die Implikation

$$a_i^T p = 0 \quad (i \in I(x)) \Longrightarrow p = 0$$

gilt.

Lösung: Wir nehmen zunächst an, $x \in M$ sei eine Ecke von M. Angenommen, es existiert ein $p \neq 0$ mit $a_i^T p = 0$, $i \in I(x)$. Mit einem hinreichend kleinen $\delta > 0$ ist dann $x \pm \delta p \in M$ und folglich x keine Ecke von M. Daher gilt die Implikation

$$x \text{ Ecke von } M \Longrightarrow (a_i^T p = 0 \quad (i \in I(x)) \Longrightarrow p = 0).$$

Umgekehrt nehmen wir nun an, $x \in M$ sei keine Ecke von M. Dann existieren $x_1, x_2 \in M \setminus \{x\}$ und $\lambda \in (0,1)$ mit $x = (1-\lambda)x_1 + \lambda x_2$. Für $i \in I(x)$ ist

$$0 = b_i - a_i^T x = (1-\lambda)(\underbrace{b_i - a_i^T x_1}_{\geq 0}) + \lambda(\underbrace{b_i - a_i^T x_2}_{\geq 0}),$$

wegen $\lambda \in (0,1)$ ist $0 = b_i - a_i^T x_1 = b_i - a_i^T x_2$ und daher $a_i^T(x_1 - x_2) = 0$, $i \in I(x)$. Wegen $x_1 \neq x_2$ (andernfalls wäre $x = x_1 = x_2$) gilt auch die Implikation

$$(a_i^T p = 0 \quad (i \in I(x)) \Longrightarrow p = 0) \Longrightarrow x \text{ Ecke von } M.$$

3. Sei $a := (1,2,4)^T$ und $b := 4$. Man zeige, dass jede Ecke von

$$M := \{x \in \mathbb{R}^3 : x \geq 0,\ a^T x \leq b\}$$

ganzzahlig ist.

Lösung: Durch $(4,0,0)^T$, $(0,2,0)^T$ und $(0,0,1)^T$ sind die drei Ecken von M gegeben, diese sind ganzzahlig.

4. Mit $A \in \mathbb{Z}^{m \times n}$ sind auch A^T, $-A$ und $\begin{pmatrix} A & I \end{pmatrix}$ total unimodular.

Lösung: Die ersten beiden Aussagen sind trivial. Nun zeigen wir, dass $\begin{pmatrix} A & e_j \end{pmatrix}$ mit einem j-ten Einheitsvektor total unimodular ist. Denn man greife sich eine beliebige quadratische $k \times k$-Untermatrix heraus. Ist diese schon eine Untermatrix von A, so ist ihre Determinante 0, 1 oder -1, da A nach Voraussetzung total unimodular ist. Andernfalls ist ein Nullvektor oder ein Einheitsvektor im $\mathbb{R}^k$ in der Untermatrix enthalten. Im ersten Fall ist sie singulär, im andern Fall kann die Determinante nach der letzten Spalte entwickelt werden und man erkennt, dass sie den Wert 0 oder ± 1 hat. Auf diese Weise kann man sukzessive die Einheitsvektoren an die Matrix A anheften und erhält, dass $\begin{pmatrix} A & I \end{pmatrix}$ total unimodular ist.

6.3.3 Aufgaben zu Abschnitt 4.3

1. Man betrachte den bewerteten, gerichteten Graphen in Abbildung 6.24 und bestimme die kürzesten Entfernungen und die kürzesten Pfade vom Knoten 1 zu allen anderen Knoten. Hierbei wende man zunächst das Verfahren von Dijkstra (obwohl nicht alle Längen nichtnegativ sind), danach das Verfahren von Bellman-Ford an.

Lösung: In dem gerichteten Graphen existiert kein Zyklus negativer Länge, daher terminiert der Modellalgorithmus. Wir wenden zunächst das Verfahren von Dijkstra an.

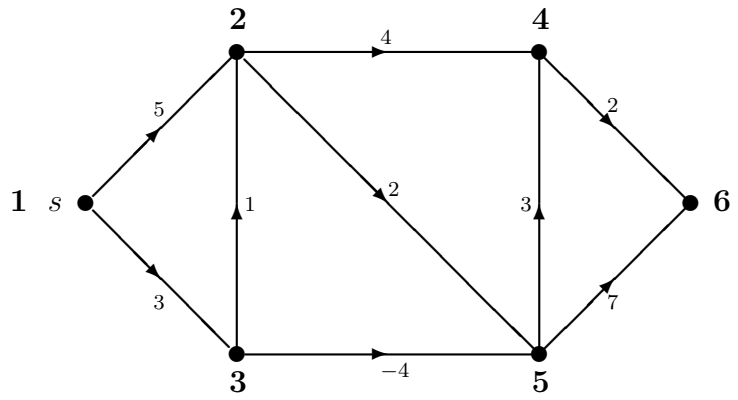


Abbildung 6.24: Ein gerichteter Graph mit Quelle $s = 1$

Mit unserer üblichen Notation erhalten wir

k	i	d	V
0		$(0, \infty, \infty, \infty, \infty, \infty)$	$\{1\}$
1	1	$(0, 5, 3, \infty, \infty, \infty)$	$\{2, 3\}$
2	3	$(0, 4, 3, \infty, -1, \infty)$	$\{2, 5\}$
3	5	$(0, 4, 3, 2, -1, 6)$	$\{2, 4, 6\}$
4	4	$(0, 4, 3, 2, -1, 4)$	$\{2, 6\}$
5	6	$(0, 4, 3, 2, -1, 4)$	$\{2\}$
6	2	$(0, 4, 3, 2, -1, 4)$	$\emptyset$

Damit sind die Abstände vom Knoten 1 zu den übrigen Knoten berechnet. Um die kürzesten Pfade selbst zu bestimmen, geben wir in Abbildung 6.25 den zugehörigen aufspannenden Baum an. Nun wenden wir das Verfahren von Bellman-Ford an. Jetzt

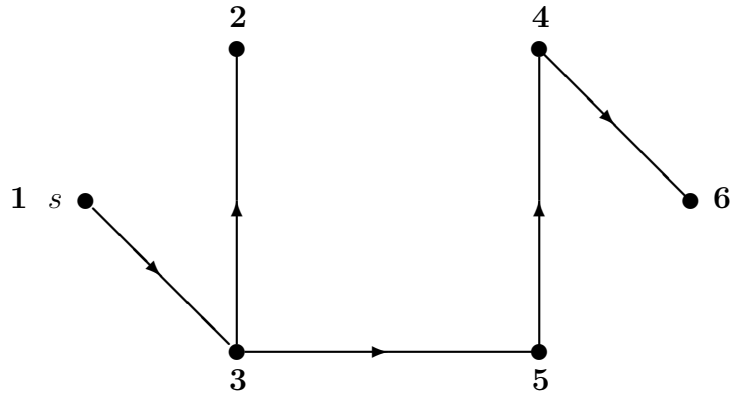


Abbildung 6.25: Kürzeste Pfade

werden durch k Zyklen gezählt.

k	i	d	V
0		$(0, \infty, \infty, \infty, \infty, \infty)$	$\{1\}$
1	1	$(0, 5, 3, \infty, \infty, \infty)$	$\{2, 3\}$
2	2	$(0, 5, 3, 9, 7, \infty)$	$\{3, 4, 5\}$
	3	$(0, 4, 3, 9, -1, \infty)$	$\{4, 5, 2\}$
3	4	$(0, 4, 3, 9, -1, 11)$	$\{5, 2, 6\}$
	5	$(0, 4, 3, 2, -1, 6)$	$\{2, 6, 4\}$
	2	$(0, 4, 3, 2, -1, 6)$	$\{6, 4\}$
4	6	$(0, 4, 3, 2, -1, 6)$	$\{4\}$
	4	$(0, 4, 3, 2, -1, 4)$	$\{6\}$
5	6	$(0, 4, 3, 2, -1, 4)$	$\emptyset$

2. Gegeben sei der Digraph in Abbildung 6.26. Die Kosten zwischen zwei Knoten sind

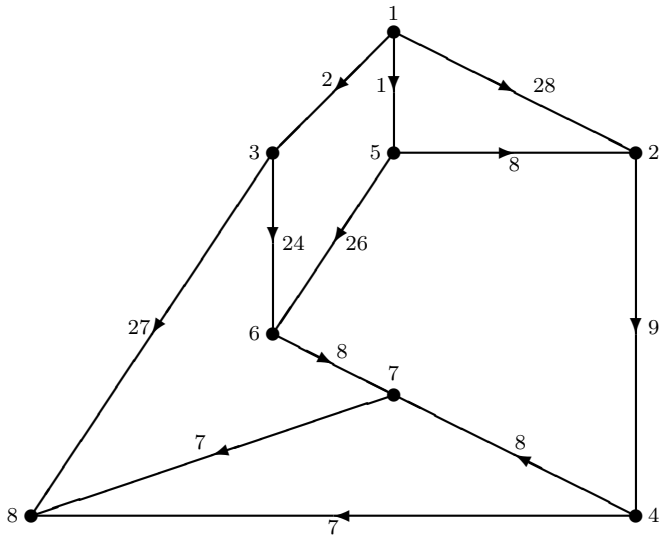


Abbildung 6.26: Was ist der kürzeste Weg von 1 zu den übrigen Knoten?

eingetragen. Mit Hilfe des Dijkstra-Verfahrens bestimme man einen kürzesten Weg zwischen 1 und den übrigen Knoten.

Lösung: Wir erhalten die folgende Tabelle:

k	i	d	V
0		$(0, \infty, \infty, \infty, \infty, \infty)$	$\{1\}$
1	1	$(0, 28, 2, \infty, 1, \infty, \infty, \infty)$	$\{2, 3, 5\}$
2	5	$(0, 9, 2, \infty, 1, 27, \infty, \infty)$	$\{2, 3, 6\}$
3	3	$(0, 9, 2, \infty, 1, 26, \infty, 29)$	$\{2, 6, 8\}$
4	2	$(0, 9, 2, 18, 1, 26, \infty, 29)$	$\{4, 6, 8\}$
5	4	$(0, 9, 2, 18, 1, 26, 25, 25)$	$\{6, 7, 8\}$
6	7	$(0, 9, 2, 18, 1, 26, 25, 25)$	$\{6, 8\}$
7	8	$(0, 9, 2, 18, 1, 26, 25, 25)$	$\{6\}$
8	6	$(0, 9, 2, 18, 1, 26, 25, 25)$	$\emptyset$

3. Gegeben sei der Digraph in Abbildung 6.26. Man bestimme einen kürzesten Pfad vom Knoten 1 zum Knoten 8 mit Hilfe des primal-dualen Algorithmus.

   Lösung: Wir erhalten die folgenden Iterationsschritte.

y	J	$\overline{u}$	t^*
$(0,0,0,0,0,0,0)^T$	$\emptyset$	$(1,1,1,1,1,1,1)^T$	7
$(7,7,7,7,7,7,7)^T$	$\{(7,8),(4,8)\}$	$(1,1,1,0,1,1,0)^T$	8
$(15,15,15,7,15,15,7)^T$	$\{(7,8),(4,8),(6,7)\}$	$(1,1,1,0,1,0,0)^T$	1
$(16,16,16,7,16,15,7)^T$	$\{(7,8),(4,8),(6,7),(2,4)\}$	$(1,0,1,0,1,0,0)^T$	8
$(24,16,24,7,24,15,7)^T$	$\{(7,8),(4,8),(6,7),(2,4),(5,2)\}$	$(1,0,1,0,0,0,0)^T$	1
$(25,16,25,7,24,15,7)^T$	$\{(7,8),(4,8),(6,7),(2,4),(5,2),(1,5)\}$	$(0,0,1,0,0,0,0)^T$	2
$(25,16,27,7,24,15,7)^T$	$\{(7,8),(4,8),(6,7),(2,4),(5,2),(1,5),(3,8)\}$	$(0,0,0,0,0,0,0)^T$	

   Schon in der vorletzten Zeile ist man fertig, denn es gibt einen Pfad von 1 nach 8, der nur Pfeile aus $\{(7,8),(4,8),(6,7),(2,4),(5,2),(1,5)\}$ benutzt. Dieser ist ein kürzester Pfad. Er verläuft von 1 über $5,2,4$ nach 8, die Kosten sind 25. Eine weitere Iteration liefert auch eine Lösung des dualen Problems, nämlich $y = (25,16,27,7,24,15,7)^T$.

6.3.4 Aufgaben zu Abschnitt 4.4

1. Eine Gruppe von 11 Personen trifft sich in San Francisco. Möglichst viele von ihnen sollen nach New York geschickt werden. Es gibt keine Direktflüge, sondern es muss umgestiegen werden, wobei der Anschluss jeweils gesichert ist. In der folgenden Tabelle sind diese Flüge und die jeweils noch vorhandenen freien Sitze aufgelistet.

Von	Nach	Zahl freier Sitze
San Francisco	Denver	5
San Francisco	Houston	6
Denver	Atlanta	4
Denver	Chicago	2
Houston	Atlanta	5
Atlanta	New York	7
Chicago	New York	4

   (a) Man formuliere die Aufgabe als Maximalflussproblem in einem geeigneten Digraphen.

   (b) Man stelle die Knoten-Pfeil-Inzidenzmatrix auf und begründe, dass die gestutzte (letzte Zeile weggelassen) 5×7-Knoten-Pfeil-Inzidenzmatrix maximalen Rang hat. Welche 5×5-Untermatrizen sind nichtsingulär? Wie viele gibt es? Man veranschauliche sich diese im Digraphen!

   (c) Man rate einen maximalen Fluss und beweise seine Optimalität mit dem Max-Flow Min-Cut Theorem.

   (d) Ausgehend vom trivalen Fluss bestimme man jeweils zu einem aktuellen zulässigen Fluss x einen x-vermehrenden Pfad, addiere diesen auf den Fluss, bis man eine Lösung bestimmt hat.

   Lösung: Durch die 6 Städte und die 7 Verbindungen ist ein Digraph gegeben, den wir in Abbildung 6.27 darstellen. Die Kapazitäten bzw. die Anzahl freier Sitze haben wir an die jeweiligen Pfeile geschrieben. Man hat also in natürlicher Weise ein Maximaler-Fluss-Problem mit San Francisco als Quelle und New York als Senke zu lösen.

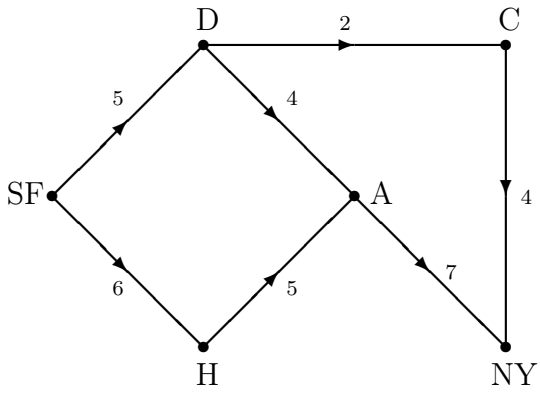


Abbildung 6.27: Flug von San Francisco nach New York

Um die Knoten-Pfeil-Inzidenzmatrix aufzustellen, müssen wir die Knoten und Pfeile nummerieren. Mit

1	San Francisco
2	Denver
3	Houston
4	Atlanta
5	Chicago
6	New York

und der Pfeilnummerierung gemäß

$$(1,2),\ (1,3),\ (2,4),\ (2,5),\ (3,4),\ (4,6),\ (5,6),$$

erhalten wir als Knoten-Pfeil-Inzidenzmatrix

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

Lässt man die letzte Zeile fort, so erhält man

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

als gestutzte Knoten-Pfeil-Inzidenzmatrix. Da der in Abbildung 6.27 angegebene Digraph zusammenhängend ist, folgt aus einer früheren Aufgabe, dass Rang $(\hat{A}) = 5$. Von den $\binom{7}{5} = 21$ möglichen 5×5-Untermatrizen von $\hat{A}$ sind genau diejenigen nichtsingulär, deren Spalten zu einem aufspannenden Baum des Digraphen gehören. Die Bedingung nach Zusammenhang und Kreisfreiheit lässt (wenn wir uns nicht verzählt haben) 15 den Digraphen in Abbildung 6.27 aufspannende Bäume übrig.

In Abbildung 6.28 geben wir einen zulässigen Fluss mit dem Wert $v = 9$ an, weiter einen

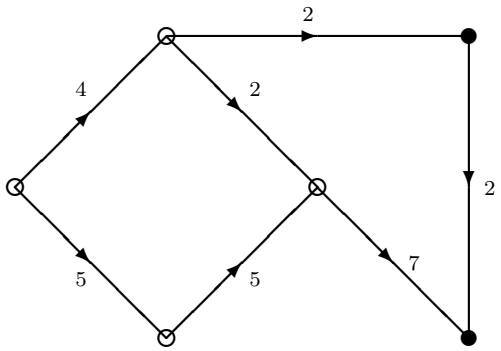


Abbildung 6.28: Optimaler Fluss von San Francisco nach New York

durch $\circ$ bzw. $\bullet$ gekennzeichneten Schnitt der Knoten, der die Kapazität 9 besitzt. Wegen des Max Flow-Min Cut Theorems hat man auch eine Lösung des Maximaler-Fluss- bzw. Minimaler-Schnitt-Problems gefunden.

Nun wenden wir, ausgehend vom trivialen Fluss, den Augmenting Path Algorithm von Ford-Fulkerson an. Wir erhalten z. B. die folgenden Iterationsschritte, wobei wir von $x = 0$ ausgehen:

- Sei $P = (SF, D, A, NY)$. Dann ist $x = (4, 0, 4, 0, 0, 4, 0)$.
- Sei $P = (SF, D, C, NY)$. Dann ist $x = (5, 0, 4, 1, 0, 4, 1)$.
- Sei $P = (SF, H, A, NY)$. Dann ist $x = (5, 3, 4, 1, 3, 7, 1)$.
- Sei $P = (SF, H, A, D, C, NY)$. Dann ist $x = (5, 4, 3, 2, 4, 7, 2)$. Diese Lösung, die von der oben angegebenen abweicht, veranschaulichen wir in Abbildung 6.29.

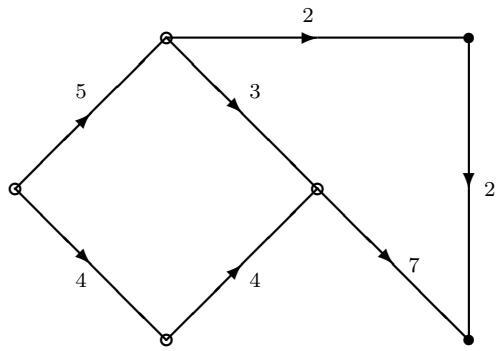


Abbildung 6.29: Optimaler Fluss von San Francisco nach New York

2. In einem Maximaler-Fluss-Problem seien alle (endlichen) Kapazitäten rationale Zahlen. Man zeige: Besitzt das Problem eine Lösung, so besitzt es auch eine rationale Lösung.

   Lösung: Man bestimme eine hinreichend große natürliche Zahl D derart, dass jede Komponente von Du ganzzahlig (oder gleich ∞) ist. Das skalierte Problem besitzt eine ganzzahlige Lösung x_D^*, durch $x^* = x_D^*/D$ ist eine (rationale) Lösung des Ausgangsproblems gegeben.

3. Gegeben sei ein Digraph $(\mathcal{N}, \mathcal{A})$ mit einer Quelle s und einer Senke t sowie Kapazitäten u_{ij} auf den Pfeilen $(i,j) \in \mathcal{A}$. Sei x ein zulässiger Fluss. Man bilde den abgeleiteten kapazitierten Digraphen $(\mathcal{N}, \mathcal{A}(x))$, wobei die Pfeile $\mathcal{A}(x)$ und Kapazitäten $u(x)$ gegeben sind durch:

   - Ist $(i,j) \in \mathcal{A}$ und $x_{ij} < u_{ij}$, so ist $(i,j) \in \mathcal{A}(x)$ und $u(x)_{ij} := u_{ij} - x_{ij}$.
   - Ist $(i,j) \in \mathcal{A}$ und $x_{ij} > 0$, so ist $(j,i) \in \mathcal{A}(x)$ und $u(x)_{ji} := x_{ij}$.

   Man zeige: Ist v^* der Wert eines maximalen Flusses in $(\mathcal{N}, \mathcal{A})$ und v der Wert des zulässigen Flusses, so ist $v^* - v$ der Wert eines maximalen Flusses in dem abgeleiteten kapazitierten Digraphen $(\mathcal{N}, \mathcal{A}(x))$.

   Lösung: Sei $(\mathcal{N}_1, \mathcal{N}_2)$ ein s-t-Schnitt in $(\mathcal{N}, \mathcal{A}(x))$, also eine Partition der Knotenmenge $\mathcal{N}$ mit $s \in \mathcal{N}_1$ und $t \in \mathcal{N}_2$. Wir berechnen die Kapazität $C(x)(\mathcal{N}_1, \mathcal{N}_2)$ des Schnittes $(\mathcal{N}_1, \mathcal{N}_2)$ bezüglich $(\mathcal{N}, \mathcal{A}(x))$. Nach Definition ist das die Summe aller Kapazitäten $u(x)_{ij}$, wobei (i,j) ein Pfeil aus $\mathcal{A}(x)$ mit $i \in \mathcal{N}_1$, $j \in \mathcal{N}_2$ ist. Also ist

$$
\begin{aligned}
C(x)(\mathcal{N}_1, \mathcal{N}_2) &= \sum_{\substack{(i,j) \in \mathcal{A}(x) \\ i \in \mathcal{N}_1,\, j \in \mathcal{N}_2}} u(x)_{ij} \\
&= \sum_{\substack{(i,j) \in \mathcal{A},\, x_{ij} < u_{ij} \\ i \in \mathcal{N}_1,\, j \in \mathcal{N}_2}} [u_{ij} - x_{ij}] + \sum_{\substack{(j,i) \in \mathcal{A},\, x_{ji} > 0 \\ i \in \mathcal{N}_1,\, j \in \mathcal{N}_2}} x_{ji} \\
&= \sum_{\substack{(i,j) \in \mathcal{A} \\ i \in \mathcal{N}_1,\, j \in \mathcal{N}_2}} [u_{ij} - x_{ij}] + \sum_{\substack{(j,i) \in \mathcal{A} \\ i \in \mathcal{N}_1,\, j \in \mathcal{N}_2}} x_{ji} \\
&= \sum_{\substack{(i,j) \in \mathcal{A} \\ i \in \mathcal{N}_1,\, j \in \mathcal{N}_2}} u_{ij} - \left[\sum_{\substack{(i,j) \in \mathcal{A} \\ i \in \mathcal{N}_1,\, j \in \mathcal{N}_2}} x_{ij} - \sum_{\substack{(j,i) \in \mathcal{A} \\ i \in \mathcal{N}_1,\, j \in \mathcal{N}_2}} x_{ji} \right] \\
&= C(\mathcal{N}_1, \mathcal{N}_2) - \sum_{i \in \mathcal{N}_1} \left(\sum_{\substack{j \in \mathcal{N}_2 \\ (i,j) \in \mathcal{A}}} x_{ij} - \sum_{\substack{j \in \mathcal{N}_2 \\ (j,i) \in \mathcal{A}}} x_{ji} \right) \\
&= C(\mathcal{N}_1, \mathcal{N}_2) - v \\
&\qquad \text{(siehe den Beweis des Max-Flow Min-Cut Theorems).}
\end{aligned}
$$

   Aus dieser Rechnung folgt offenbar schon die Behauptung, da der minimale Schnitt in $(\mathcal{N}, \mathcal{A})$ durch v^*, also der minimale Schnitt bzw. maximale Fluss in $(\mathcal{N}, \mathcal{A}(x))$ durch $v^* - v$ gegeben ist.

4. Man betrachte ein zulässiges Maximaler-Fluss-Problem im kapazitierten Digraphen $(\mathcal{N}, \mathcal{A})$. Man zeige: Werden die Kapazitätsschranken auf jedem Pfeil um $\alpha > 0$ vergrößert, so wird der maximale Fluss um höchstens $\alpha\,|\mathcal{A}|$ vergrößert.

   Lösung: Sei $(\mathcal{N}_1, \mathcal{N}_2)$ ein Schnitt in $(\mathcal{N}, \mathcal{A})$. Bezeichnet man mit $C_\alpha(\mathcal{N}_1, \mathcal{N}_2)$ die Kapazität zu den um α vergrößerten Kapazitäten, so ist

$$
C_\alpha(\mathcal{N}_1, \mathcal{N}_2) = \sum_{\substack{(i,j) \in \mathcal{A} \\ i \in \mathcal{N}_1,\, j \in \mathcal{N}_2}} (u_{ij} + \alpha) \leq C(\mathcal{N}_1, \mathcal{N}_2) + \alpha\,|\mathcal{A}|.
$$

Bezeichnet man mit v_α den Wert eines Flusses zu den um α vergrößerten Kapazitäten, so ist (schwacher Dualitätssatz)

$$v_\alpha \le C_\alpha(\mathcal{N}_1, \mathcal{N}_2) \le C(\mathcal{N}_1, \mathcal{N}_2) + \alpha\,|\mathcal{A}|.$$

Wählt man hier für $(\mathcal{N}_1, \mathcal{N}_2)$ einen minimalen Schnitt (zu den "Originalkapazitäten"), so liefert das Max Flow-Min Cut Theorem, das $v_\alpha \le v^* + \alpha\,|\mathcal{A}|$, wobei v^* der Wert des maximalen Flusses im ungestörten Problem ist. Folglich ist $v_\alpha^* \le v^* + \alpha\,|\mathcal{A}|$, das war zu zeigen.

6.3.5 Aufgaben zu Abschnitt 4.5

1. In Abbildung 6.30 ist ein gerichteter Graph dargestellt, ferner sind auf den Pfeilen Kosten, Kapazitäten und Flüsse angegeben. Die Zahlen an den Knoten sind die b_k, $k \in$

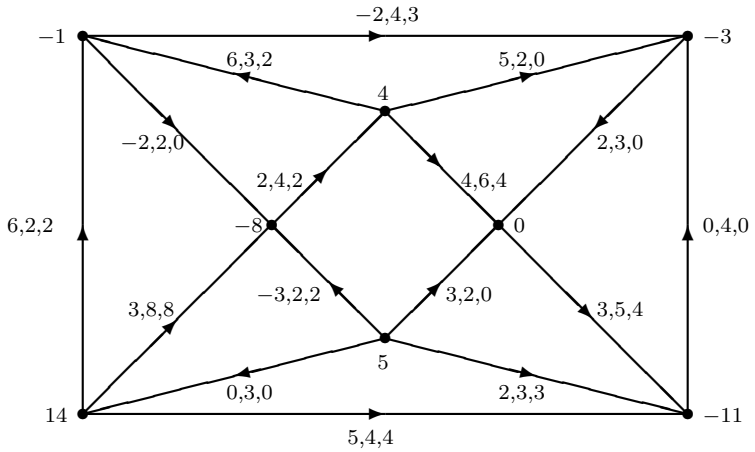


Abbildung 6.30: Ein Minimale-Kosten-Fluss-Problem

$\mathcal{N}$, die Zahlen an den Pfeilen sind in der Reihenfolge c_{ij}, u_{ij}, x_{ij} (Kosten, Kapazitäten, Flüsse) zu verstehen. Man zeige, dass der angegebene Fluss zulässig ist und eine Lösung des zugehörigen Minimale-Kosten-Fluss-Problems ist.

Lösung: Zunächst zeigen wir die Zulässigkeit des angegebenen Flusses. Da offensichtlich die Kapazitätsbeschränkungen eingehalten sind, sind nur die Flussbedingungen nachzuprüfen. Diese sind gegeben durch

$$\sum_{j:(k,j)\in\mathcal{A}} x_{kj} - \sum_{i:(i,k)\in\mathcal{A}} x_{ik} = b_k, \qquad k \in \mathcal{N}.$$

Die Differenz zwischen dem, was im Knoten k wegfliesst und dem, was ankommt, muss also b_k sein. Eine Inspektion zeigt, dass diese Bedingung erfüllt ist. Nun kommen wir zum Nachweis dafür, dass x eine Lösung ist. Wir nummerieren die Knoten von 1 bis 8: Zunächst die Äußeren von links nach rechts, danach die Inneren im Uhrzeigersinn (nicht ganz konsistent). Für die b_k, $k \in \mathcal{N}$, bedeutet diese Nummerierung, dass

$$b = (-1, 14, -3, -11, -8, 4, 0, 5)^T.$$

Wir vermuten, dass der angegebene Fluss eine optimale Basislösung ist. Mit A bezeichnen wir die in der letzten Zeile (zum Knoten 8) gestutzte Knoten-Pfeil-Inzidenzmatrix

zu $(\mathcal{N}, \mathcal{A})$. In die Menge $\mathcal{B} \subset \mathcal{A}$ der "Basispfeile" müssen auf alle Fälle die Pfeile aufgenommen werden, auf denen die Kapazitätsschranken weder nach unten noch nach oben ausgeschöpft sind. Daher müssen $(1,3)$, $(6,1)$, $(5,6)$, $(6,7)$, $(7,4)$. Wir ergänzen diese Pfeile durch zwei weitere (hier hat man einige Möglichkeiten) so zu einer Pfeilmenge $\mathcal{B}$, dass $(\mathcal{N}, \mathcal{B})$ ein $(\mathcal{N}, \mathcal{A})$ aufspannender Baum ist (jeweils als ungerichteter Graph betrachtet). Daher setzen wir z. B.

$$\mathcal{B} := \{(1,3),(6,1),(5,6),(6,7),(7,4),(2,5),(8,5)\}.$$

Mit

$$A_B := \begin{pmatrix} 1 & -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & -1 & -1 \\ 0 & 1 & -1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 1 & 0 & 0 \end{pmatrix}, \qquad c_B := \begin{pmatrix} -2 \\ 6 \\ 2 \\ 4 \\ 3 \\ 3 \\ -3 \end{pmatrix}$$

berechnen wir

$$y := A_B^{-T} c_B = \begin{pmatrix} -5 \\ 6 \\ -3 \\ -6 \\ 3 \\ 1 \\ -3 \end{pmatrix}.$$

Mit den "Nichtbasispfeilen"

$$\mathcal{A} \setminus \mathcal{B} = \{(1,5),(2,1),(2,4),(4,3),(3,7),(6,3),(8,2),(8,7),(8,4)\}$$

wird

$$A_N = \begin{pmatrix} 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & -1 & 1 & -1 & 0 & 0 & 0 \\ 0 & 0 & -1 & 1 & 0 & 0 & 0 & 0 & -1 \\ -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & 0 & 0 & -1 & 0 \end{pmatrix}, \qquad c_N = \begin{pmatrix} -2 \\ 2 \\ 5 \\ 0 \\ 2 \\ 5 \\ 0 \\ 3 \\ 2 \end{pmatrix}$$

sind die reduzierten Kosten gegeben durch

$$\overline{c}_N = c_N - A_N^T y = \begin{pmatrix} 6 \\ -9 \\ -7 \\ 3 \\ 2 \\ 1 \\ 6 \\ 0 \\ -4 \end{pmatrix}.$$

Der angebene Fluss ist optimal, wenn der Fluss in den Komponenten, in denen die reduzierten Kosten negativ sind, die oberen Kapazitätsschranken annimmt, und in denen, wo die reduzierten Kosten positiv sind, verschwindet. Da dies richtig ist, ist die Optimalität des angegebenen Flusses bewiesen.

2. Man zeige: Das gegebene Minimale-Kosten-Fluss-Problem sei zulässig. Dann besitzt das Problem genau dann eine Lösung, wenn es in $(\mathcal{N}, \mathcal{A})$ keinen gerichteten Zyklus mit negativen Kosten gibt derart, dass die Kapazitäten auf allen Pfeilen des Zyklus ∞ sind.

   Lösung: Angenommen, es gibt einen gerichteten Zyklus Z in $(\mathcal{N}, \mathcal{A})$ mit negativen Kosten $c(Z) = \sum_{(i,j)\in Z} c_{ij}$ und der Eigenschaft, dass $u_{ij} = \infty$ für alle $(i,j) \in Z$. Definiert man dann den Fluss $x(t)$ durch

$$x(t)_{ij} := \begin{cases} x_{ij} + t, & (i,j) \in Z, \\ x_{ij}, & (i,j) \notin Z, \end{cases}$$

   so ist $x(t)$ ein zulässiger Fluss und $c^T x(t) = c^T x + tc(Z) \to -\infty$. Das Minimale-Kosten-Fluss-Problem ist in diesem Falle also nicht lösbar. Nun nehmen wir an, dass es keinen solchen Zyklus gibt. Die Existenz einer optimalen Lösung des Minimale-Kosten-Fluss-Problems ist gesichert, wenn wir zeigen können, dass das duale Problem (D) eine zulässige Lösung besitzt. Wie wir uns früher schon überlegt haben, ist dies der Fall, wenn es ein $y \in \mathbb{R}^{|\mathcal{A}|}$ mit $\overline{c}_{ij} := c_{ij} - y_i + y_j \geq 0$ für alle $(i,j) \in \mathcal{A}$ mit $u_{ij} = \infty$ gibt. Nun definieren wir einen neuen gerichteten Graphen, in dem wir einen Knoten r zu $\mathcal{N}$ hinzufügen, aus $\mathcal{A}$ alle Pfeile mit endlicher Kapazität weglassen und Pfeile (r,j) für alle $j \in \mathcal{N}$ mit $c_{rj} := 0$ hinzufügen. In dem so definierten Digraphen gibt es keinen (gerichteten) Zyklus mit negativen Kosten. Das Kürzeste-Pfad-Problem, vom Knoten r einen kürzesten Pfad zu einem der Knoten aus $\mathcal{N}$ zu bestimmen, ist also lösbar. Also ist auch das hierzu duale Problem lösbar und dies liefert insbesondere das gesuchte y.

3. Gegeben sei ein Minimale-Kosten-Fluss-Problem mit den in Abbildung 6.31 angegebenen Daten. In der Abbildung links sind die Knoten nummeriert und zu den Pfeilen sind die Kosten angegeben. Rechts ist in den Knoten das Angebot bzw. der Bedarf und zu den Pfeilen die entsprechenden Kapazitäten angegeben. Mit Hilfe des Netzwerk-

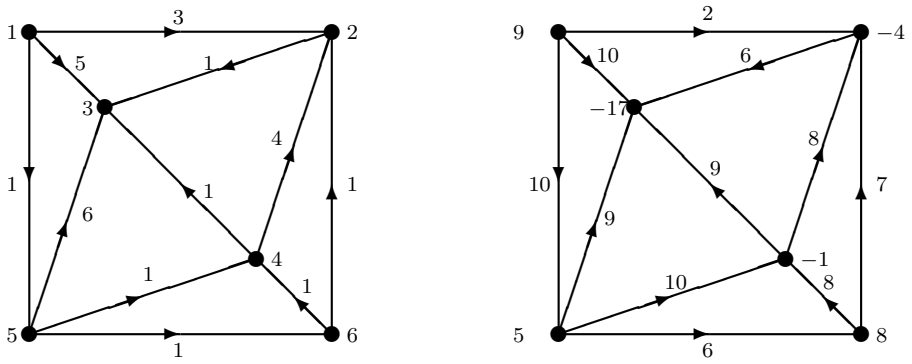


Abbildung 6.31: Ein Minimale-Kosten-Fluss-Problem

   Simplex-Verfahren löse man dieses Problem. Hierbei starte man mit der zulässigen Basislösung

$$x = (x_{12}, x_{13}, x_{15}, x_{23}, x_{42}, x_{43}, x_{53}, x_{54}, x_{56}, x_{62}, x_{64})^T = (0,0,9,6,4,3,8,0,6,6,8)^T$$

zur Basisindexmenge

$$\mathcal{B} = \{(1,5),(4,2),(4,3),(5,3),(6,2)\}.$$

Weshalb ist $(x, \mathcal{B})$ eine Basislösung?

Lösung: Bezeichnet man den in Abbildung 6.31 angegebenen Digraphen mit $(\mathcal{N}, \mathcal{A})$, so ist $(\mathcal{N}, \mathcal{B})$ offensichtlich ein $(\mathcal{N}, \mathcal{A})$ aufspannender Baum. Die Nichtbasispfeile sind $\mathcal{A} \setminus \mathcal{B} = \{(1,2),(1,3),(2,3),(5,4),(5,6),(6,4)\}$. Anhand der Tabelle

(i,j)	$(1,2)$	$(1,3)$	$(2,3)$	$(5,4)$	$(5,6)$	$(6,4)$
u_{ij}	2	10	6	10	6	8
x_{ij}	0	0	6	0	6	8
$\bar{c}_{ij}$	-7	-2	4	-4	-7	4

erkennen wir, dass für alle Nichtbasispfeile die Kapazitätsschranken angenommen werden. Also ist $(x, \mathcal{B})$ eine (zulässige) Basislösung. Anschließend berechnet man $y \in \mathbb{R}^5$ aus $y_i - y_j = c_{ij}$, $(i,j) \in \mathcal{B}$, mit $y_6 = 0$ bzw.

$$\begin{aligned} y_1 - y_5 &= 1, \\ y_4 - y_2 &= 4, \\ y_4 - y_3 &= 1, \\ y_5 - y_3 &= 6, \\ -y_2 &= 1, \end{aligned}$$

was auf $y = (9,-1,2,3,8)^T$ führt. Die zugehörigen reduzierten Kosten haben wir schon in obige Tabelle geschrieben. Dann ist

$$\begin{aligned} \mathcal{N}_1 &= \{(i,j) \in \mathcal{A} \setminus \mathcal{B} : x_{ij} = 0,\ \bar{c}_{ij} < 0\} \\ &= \{(1,2),(1,3),(5,4)\}, \\ \mathcal{N}_2 &= \{(i,j) \in \mathcal{A} \setminus \mathcal{B} : x_{ij} = u_{ij},\ \bar{c}_{ij} > 0\} \\ &= \{(2,3),(6,4)\}. \end{aligned}$$

Wir wählen $(i_s, j_s) = (1,3) \in \mathcal{N}_1$. Es wird $t^* = 8$ und $(i_r, j_r) = (5,3)$. Als neue Basislösung hat man

$$x = (x_{12}, x_{13}, x_{15}, x_{23}, x_{42}, x_{43}, x_{53}, x_{54}, x_{56}, x_{62}, x_{64})^T = (0,8,1,6,4,3,0,0,6,6,8)^T$$

zur Basisindexmenge

$$\mathcal{B} = \{(1,5),(4,2),(4,3),(1,3),(6,2)\}.$$

Diesmal ist $y = (7,-1,2,3,6)^T$ und

(i,j)	$(1,2)$	$(5,3)$	$(2,3)$	$(5,4)$	$(5,6)$	$(6,4)$
u_{ij}	2	9	6	10	6	8
x_{ij}	0	0	6	0	6	8
$\bar{c}_{ij}$	-5	2	4	-2	-5	4

Es wird $\mathcal{N}_1 = \{(1,2),(5,4)\}$, $\mathcal{N}_2 = \{(2,3),(6,4)\}$. Wir wählen $(i_s, j_s) = (5,4) \in \mathcal{N}_1$. Es wird $t^* = 6$ und $(i_r, j_r) = (4,3)$. Die neue Basislösung ist

$$x = (x_{12}, x_{13}, x_{15}, x_{23}, x_{42}, x_{43}, x_{53}, x_{54}, x_{56}, x_{62}, x_{64})^T = (0,2,7,6,4,9,0,6,6,8)^T$$

zur Basisindexmenge

$$\mathcal{B} = \{(1,5),(4,2),(5,4),(1,3),(6,2)\}.$$

Jetzt ist $y = (5,-1,0,3,4)^T$ und

(i,j)	$(1,2)$	$(5,3)$	$(2,3)$	$(4,3)$	$(5,6)$	$(6,4)$
u_{ij}	2	9	6	9	6	8
x_{ij}	0	0	6	9	6	8
$\overline{c}_{ij}$	-3	2	2	-2	-3	4

Es ist $\mathcal{N}_1 = \{(1,2)\}$, $\mathcal{N}_2 = \{(2,3),(6,4)\}$. Wir wählen $(i_s,j_s) = (1,2) \in \mathcal{N}_1$. Es wird $t^* = 2$, der hinzukommende Pfeil $(1,2)$ wird saturiert, bevor es die Pfeile auf dem Pfad von $j_s = 2$ nach $i_s = 1$ im aktuellen Baum $(\mathcal{N},\mathcal{B})$ werden, so dass $\mathcal{B}$ unverändert bleibt. Die zugehörige neue Basislösung ist

$$x = (x_{12},x_{13},x_{15},x_{23},x_{42},x_{43},x_{53},x_{54},x_{56},x_{62},x_{64})^T = (2,2,5,6,2,9,0,4,6,6,8)^T.$$

Die neue Tabelle verändert sich kaum:

(i,j)	$(1,2)$	$(5,3)$	$(2,3)$	$(4,3)$	$(5,6)$	$(6,4)$
u_{ij}	2	9	6	9	6	8
x_{ij}	2	0	6	9	6	8
$\overline{c}_{ij}$	-3	2	2	-2	-3	4

Daher ist $\mathcal{N}_2 = \{(2,3),(6,4)\}$. Wir wählen $(i_s,j_s) = (2,3) \in \mathcal{N}_2$. Es wird $t^* = 2$ und $(i_r,j_r) = (4,2)$, die neue Basislösung ist

$$x = (x_{12},x_{13},x_{15},x_{23},x_{42},x_{43},x_{53},x_{54},x_{56},x_{62},x_{64})^T = (2,4,3,4,0,9,0,2,6,6,8)^T$$

zur Basisindexmenge

$$\mathcal{B} = \{(1,5),(2,3),(5,4),(1,3),(6,2)\}.$$

Jetzt ist $y = (3,-1,-2,1,2)^T$ und

(i,j)	$(1,2)$	$(5,3)$	$(4,2)$	$(4,3)$	$(5,6)$	$(6,4)$
u_{ij}	2	9	8	9	6	8
x_{ij}	2	0	0	9	6	8
$\overline{c}_{ij}$	-1	2	2	-2	-1	2

Daher ist $\mathcal{N}_1 = \emptyset$, $\mathcal{N}_2 = \{(6,4)\}$. Daher ist $(i_s,j_s) = (6,4) \in \mathcal{N}_2$. Es wird $t^* = 1$ und $(i_r,j_r) = (6,2)$, die neue Basislösung ist

$$x = (x_{12},x_{13},x_{15},x_{23},x_{42},x_{43},x_{53},x_{54},x_{56},x_{62},x_{64})^T = (2,3,4,5,0,9,0,3,6,7,7)^T$$

zur Basisindexmenge

$$\mathcal{B} = \{(1,5),(2,3),(5,4),(1,3),(6,4)\}.$$

Dann ist $y = (1,-3,-4,-1,0)^T$ und man erhält die Tabelle

(i,j)	$(1,2)$	$(5,3)$	$(4,2)$	$(4,3)$	$(5,6)$	$(6,2)$
u_{ij}	2	9	8	9	6	7
x_{ij}	2	0	0	9	6	7
$\overline{c}_{ij}$	-1	2	2	-2	1	-2

Es ist $\mathcal{N}_1 = \emptyset$, $\mathcal{N}_2 = \{(5,6)\}$, daher $(i_s, j_s) = (5,6)$. Es ist $t^* = u_{i_s j_s} = 6$, daher bleibt die Basisindexmenge $\mathcal{B}$ unverändert, die neue Basislösung ist

$$x = (x_{12}, x_{13}, x_{15}, x_{23}, x_{42}, x_{43}, x_{53}, x_{54}, x_{56}, x_{62}, x_{64})^T = (2,3,4,5,0,9,0,9,0,7,1)^T.$$

Die neue Tabelle wird

(i,j)	$(1,2)$	$(5,3)$	$(4,2)$	$(4,3)$	$(5,6)$	$(6,2)$
u_{ij}	2	9	8	9	6	7
x_{ij}	2	0	0	9	0	7
$\bar{c}_{ij}$	-1	2	2	-2	1	-2

Daher ist $\mathcal{N}_1 = \mathcal{N}_2 = \emptyset$, folglich $(x, \mathcal{B})$ eine optimale Basislösung.

4. In einer Stadt braucht man zu einer gewissen Zeit Ersatzbusse. Und zwar werden an Orten O_1, O_2, O_3 und O_4 genau 3, 3, 4 bzw. 5 Busse benötigt, welche aus Garagen G_1, G_2 bzw. G_3 kommen müssen, in denen 2, 6 bzw. 7 Busse für diesen Notfall bereitstehen. Die Gesamtzeit, die die Busse von ihrer Garage zu ihrem Bestimmungsort fahren, soll minimiert werden. Die Zeiten liest man aus der folgenden Tabelle ab.

	O_1	O_2	O_3	O_4
G_1	13	11	15	20
G_2	17	14	12	13
G_3	18	18	15	12

Man bestimme einen optimalen Transportplan.

Lösung: Das Transportproblem ist durch die Daten

	k^T
l	c

$=$

	3	3	4	5
2	13	11	15	20
6	17	14	12	13
7	18	18	15	12

gegeben. Mit Hilfe der Matrixminimumregel erhalten wir die Ausgangsbasislösung

$$\begin{pmatrix} & 2 & & \\ 1 & 1 & 4 & \\ 2 & & & 5 \end{pmatrix}, \qquad \mathcal{B} = \{(1,2),(2,1),(2,2),(2,3),(3,1),(3,4)\}$$

mit den Kosten $c_0 = 197$. Den zugehörigen aufspannenden Baum im Transportgraphen geben wir in Abbildung 6.32 an. Danach berechnet man u_i, $i = 1, \ldots, 3$, und v_j, $j = 1, \ldots, 4$ mit $v_4 = 0$ aus

$$u_i - v_j = c_{ij}, \qquad (i,j) \in \mathcal{B}.$$

Dies ist sehr einfach, wenn man die c_{ij}, $(i,j) \in \mathcal{B}$, eingerahmt in ein $m \times n$-Feld einträgt und die Ränder u, v auffüllt. Anschließend besetzt man die Positionen, die zu $(i,j) \in \mathcal{A} \setminus \mathcal{B}$ gehören mit den reduzierten Kosten $\bar{c}_{ij} := c_{ij} - u_i + v_j$. Dies liefert:

c	u
v^T	

$=$

-1	11	6	12	8
17	14	12	2	11
18	3	2	12	12
-6	-3	-1	0	

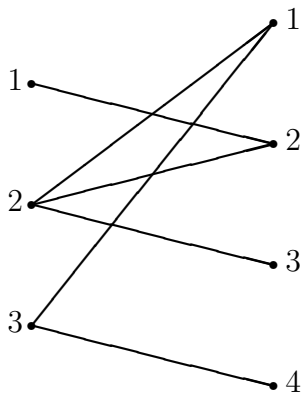


Abbildung 6.32: Ein aufspannender Baum zum Transportgraphen

In die Basis aufzunehmen ist daher $(1,1)$. Damit ist $x(t)$ gegeben durch

$$x(t) = \begin{pmatrix} t & 2-t & & \\ 1-t & 1+t & 4 & \\ 2 & & & 5 \end{pmatrix},$$

ferner ist $t^* = 1$. Die neue Basislösung ist

$$x = \begin{pmatrix} 1 & 1 & & \\ & 2 & 4 & \\ 2 & & & 5 \end{pmatrix}, \qquad \mathcal{B} = \{(1,2),(1,1),(2,2),(2,3),(3,1),(3,4)\},$$

die zugehörigen Kosten sind $c_0 = 196$. Im nächsten Schritt erhält man

c	u
v^T	

$=$

13	**11**	6	13	7
1	**14**	**12**	3	10
18	2	1	**12**	12
-6	-4	-2	0	

Alle reduzierten Kosten sind nichtnegativ, die erhaltene Basislösung daher optimal.