TECHNISCHE UNIVERSITÄT DRESDEN

Fakultät Mathematik Institut für Numerik, Professur für Numerik der Optimalen Steuerung

OPTIMIERUNG UND NUMERIK

Dr. John Martinovic

Wintersemester 2019/20

Autor : Eric Kunze
E-Mail : eric.kunze@mailbox.tu-dresden.de

Inhaltsverzeichnis

1 Einführung **3**

1.1 Aufgabenstellung und Grundbegriffe . . . 3
1.2 Beispiele zur kontinuierlichen Optimierung . . . 4
1.2.1 Transportoptimierung . . . 4
1.2.2 Kürzeste euklidische Entfernung . . . 5
1.3 Beispiele zur diskreten Optimierung . . . 5
1.3.1 Das Rucksackproblem . . . 5
1.3.2 Das Bin-Packing-Problem . . . 6
1.3.3 Standortplanung . . . 7
1.3.4 Quadratisches Zuordnungsproblem . . . 8

2 Grundlagen **9**

2.1 Existenz von Lösungen . . . 9
2.2 Optimalitätsbedingungen . . . 12
2.3 Das Lemma von Farkas . . . 15

3 Lineare Optimierung **17**

3.1 Basislösungen und Ecken . . . 17
3.2 Das primale Simplex-Verfahren . . . 19
3.2.1 Phase 2 des Simplex-Verfahrens . . . 19
3.2.2 Phase 1 (Hilfsfunktionsmethode) . . . 24
3.2.3 Der Simplexalgorithmus . . . 24
3.3 Das duale Simplexverfahren . . . 25
3.4 Dualität . . . 27
3.5 Transportoptimierung . . . 30
3.5.1 Problemstellung . . . 30
3.5.2 Erzeugung eines ersten Transportplans . . . 32
3.5.3 Der Transportalgorithmus . . . 34

4 Diskrete Optimierung **37**

4.1 Spaltengenerierung . . . 37
4.2 Die Methode Branch & Bound . . . 38
4.2.1 Grundlagen . . . 38
4.2.2 Allgemeiner B&B-Algorithmus . . . 39
4.2.3 Beispiele für B&B-Verfahren . . . 40
4.3 Dynamische Optimierung . . . 46
4.3.1 Das Rucksackproblem in der dynamischen Optimierung . . . 46
4.3.2 2D-Guillotine-Zuschnitt . . . 47

4.4 Schnittebenenverfahren 48

5 Optimierung auf Graphen **53**

5.1 Grundbegriffe der Graphentheorie 53

5.2 Das Minimalgerüst-Problem / Minimum Spanning Tree Problem 53

5.2.1 Algotihmus von Kruskal 53

5.2.2 Algorithmus von Prim & Dijkstra 54

5.3 Optimale Wege 55

5.3.1 Das Kürzeste-Wege-Problem 55

5.3.2 Das Längste-Wege-Problem 57

5.3.3 Netzplantechnik 57

5.4 Maximale Flüsse in Graphen 58

5.4.1 Problemstellung 58

5.4.2 Algorithmus von Edmonds & Karp 60

Kapitel 1
EINFÜHRUNG

1.1 Aufgabenstellung und Grundbegriffe

Es seien $G \subseteq \mathbb{R}^n$ und $f\colon G \to \mathbb{R}$ gegeben. In dieser Vorlesung betrachten wir Optimierungsaufgaben (OA) der Form

$$f(x) \to \min \qquad \text{bei } x \in G \tag{1.1}$$

Man nennt

- f die **Zielfunktion**,
- G den **zulässigen Bereich** und
- ein $x \in G$ **zulässigen Punkt** (oder zulässige Lösung).

Ein zulässiger Punkt $x^* \in G$ heißt **optimal** (oder Lösung oder optimale Lösung), wenn für alle $x \in G$ die Ungleichung

$$f(x^*) \leq f(x) \tag{1.2}$$

gilt. Falls das Problem (1.1) lösbar ist, so wird mit $f^* = f(x^*)$ der **Optimalwert** bezeichnet. Das Problem (1.1) ist ein

- **unrestringiertes** (oder freies) Optimierungsproblem, wenn $G = \mathbb{R}^n$ gilt,
- andernfalls (d.h. für $G \neq \mathbb{R}^n$) ein **restringiertes** Problem

und außerdem eine

- **diskrete** (oder ganzzahlige) OA (engl. integer program), falls jede Variable einer diskreten Menge angehört
- **kontinuierliche** (oder stetige) OA, falls alle Variablen stetige Werte annehmen
- gemischt ganzzahlige OA, wenn sowohl stetige als auch diskrete Variablen vorkommen.

Gilt in (1.1) $f(x) = c^\top x$ für ein $c \in \mathbb{R}^n$ und ist G durch lineare Bedingungen beschreibbar, so heißt (1.1) **linear**. In diesem Fall lässt sich (1.1) schreiben als

$$c^\top x \to \min \qquad \text{bei } Ax = a, Bx \leq b \tag{1.3}$$

mit geeigneten Matrizen A und B sowie Vektoren a und b.

Gerade für (gemischt) ganzzahlige OA kann die Lösung der Originalaufgabe schwierig sein. Eine verwandte, jedoch im Allgemeinen leichter zu lösende Aufgabe kann in diesen Fällen wie folgt erhalten werden:

Definition 1.1 Wir betrachten die Optimierungsaufgaben

$$f(x) \to \min \quad \text{bei } x \in D \cap E \tag{P}$$
$$g(x) \to \min \quad \text{bei } x \in E \tag{Q}$$

(Q) heißt **Relaxation** zu (P) falls $g(x) \leq f(x)$ für alle $x \in D \cap E$ gilt. In vielen Fällen wird dabei $g = f$ gewählt.

Der Optimalwert der Relaxation kann als Näherung (bzw. untere Schranke) für den tatsächlichen Optimalwert von (P) genutzt werden. Meistens liefert die Lösung von (Q) jedoch keinen zulässigen Puntk für (P).

Satz 1.1 Ist $\overline{x}$ eine Lösung von (Q) und gilt $\overline{x} \in D$ sowie $f(\overline{x}) = g(\overline{x})$, dann löst $\overline{x}$ auch (P).

Beweis. siehe Übung □

Definition 1.2 Seien (Q1) und (Q2) Relaxationen zu (P). (Q1) heißt **stärker** (oder strenger) als (Q2), wenn die Schranke (d.h. der Optimalwert) von (Q1) größer oder gleich der Schranke (Optimalwert) von (Q2) für jede Instanz von (P) ist.

Anmerkung. *Zur Erklärung des Begriffes "Instanz" betrachte das folgende Beispiel.*

- *Problemklasse:* $c^\top x \to \min$
- *Instanz der Problemklasse:* $x_1 + 2x_2 - 3x_3 \to \min$

Eine Instanz ist also eine konkrete Belegung.

1.2 Beispiele zur kontinuierlichen Optimierung

1.2.1 Transportoptimierung

Die Transportoptimierung ist ein Beispiel einer linearen Optimierung.

Es gebe Erzeuger $i \in I = \{1, \ldots, n\}$ und Verbraucher $j \in J = \{1, \ldots, n\}$. Weiterhin seien die Kosten c_{ij} für den Transport einer Einheit von i nach j sowie der Vorrat $a_i > 0$ und der Bedarf $b_j > 0$ für alle i und j gegeben. Wie muss der Transport organisiert werden, damit die Gesamtkosten minimal sind?

Für jedes mathematische Modell einer OA braucht man

- geeignete Variablen ($\to x$)
- Zielfuntkion ($\to f$)
- Nebenbedingungen ($\to G$)

Variablen $x_{ij} \geq 0$ für alle $i \in I$ und $j \in J$ beschreibe die Einheiten, die von i nach j transportiert werden.

Zielfunktion $f(x) = \sum_{i \in I} \sum_{j \in J} c_{ij} x_{ij} \to \min$

Nebenbedingungen

- Kapazitätsbeschränkung der Erzeuger $i \in I$: $\sum_{j \in J} x_{ij} \leq a_i \quad (i \in I)$
- Bedarfserfüllung von Verbrauchern $j \in J$: $\sum_{i \in I} x_{ij} \geq b_j \quad (j \in J)$

Somit können wir als Modell formulieren:

$$f(x) = \sum_{i \in I} \sum_{j \in J} c_{ij} x_{ij} \to \min \quad \text{bei} \quad \begin{aligned} \sum_{j \in J} x_{ij} &\leq a_i \quad (i \in I), \\ \sum_{i \in I} x_{ij} &\geq b_j \quad (j \in J), \\ x_{ij} &\geq 0 \quad ((i,j) \in I \times J) \end{aligned}$$

1.2.2 Kürzeste euklidische Entfernung

Die Optimierung der kürzesten euklidischen Entfernung ist nichtlinear.

Gegeben seien ein Punkt $\widetilde{x} \in \mathbb{R}^n$ und eine Menge $G \subseteq \mathbb{R}^n$ mit $\widetilde{x} \notin G$. Wir betrachten die folgende OA:

$$f(x) = \|x - \widetilde{x}\|_2^2 \to \min \quad \text{bei } x \in G$$

Ist $G \neq \emptyset$ und abgeschlossen, so existiert eine Lösung. Ist G zusätzlich konvex, so ist die Lösung sogar eindeutig.

Weitere Beispiele und Theorie sind in der Vorlesung „Kontinuierliche Optimierung“ im Master Mathematik zu erfahren.

1.3 Beispiele zur diskreten Optimierung

1.3.1 Das Rucksackproblem

Gegeben seien ein Behälter („Rucksack“) mit Kapazität $b \in \mathbb{Z}_+ := \{0, 1, \ldots\}$ sowie m Teile, die jeweils durch ein Gewicht $a_i \in \mathbb{Z}_+$ und einen Nutzen $c_i \in \mathbb{Z}_+$ beschrieben werden $(i = 1, \ldots, m)$. Aus dieser Menge von Objekten ist eine *nutzenmaximale* Teilmenge auszuwählen.

Variablen

$$x_i := \begin{cases} 1 & \text{wenn Teil } i \text{ eingepackt wird} \\ 0 & \text{sonst} \end{cases} \qquad (i = 1, \ldots, m)$$

Zielfunktion $f(x) = \sum_{i=1}^{m} c_i x_i \to \max$

Nebenbedingungen Kapazitätsbedingung: $\sum_{i=1}^{m} a_i x_i \leq b$

Als Modell können wir somit formulieren:

$$f(x) = \sum_{i=1}^{m} c_i x_i \to \max \quad \text{bei} \quad \sum_{i=1}^{m} a_i x_i \leq b \quad \text{und} \quad x_i \in \{0,1\} \quad (i = 1, \ldots, m)$$

Aufgrund der binären Gestalt der Variablen wird das Problem auch als 0/1-Rucksackproblem bezeichnet. Im Gegensatz dazu ist beim klassischen Rucksackproblem jedes Teil mehrfach nutzbar. In diesem Fall ist $x_i \in \mathbb{Z}_+$ zu fordern.

1.3.2 Das Bin-Packing-Problem

Gegeben seien (sehr große) Anzahl an Behältern der Kapazität L sowie b_i Teile des Gewichts oder Volumens ℓ_i mit $i \in I = \{1, \ldots, m\}$. Man ermittle die minimale Anzahl an Behältern, die benötigt wird, um alle Objekte zu verstauen. Jede Packung (eines Behälters) kann als Vektor $a = (a_1, \ldots, a_m) \in \mathbb{Z}_+^m$ geschrieben werden, wobei a_i angibt, wie oft das Teil i benutzt wird. Ein solcher Vektor ist eine zulässige Packung, wenn

$$\sum_{i=1}^{m} \ell_i a_i \leq L$$

ist.

Modell nach Kantorovich Wir benötigen

- eine obere Schranke $u \in \mathbb{Z}_+$ für die maximal benötigte Anzahl an Behältern
- $y_k = \begin{cases} 1 & \text{wenn Rucksack } k \text{ benutzt wird} \\ 0 & \text{sonst} \end{cases} \qquad (k = 1, \ldots, u)$
- $x_{ik} \in \mathbb{Z}_+$, die angeben, wieviele Objekte vom Typ i in Behälter k gepackt werden $((i,k) \in \{1, \ldots, m\} \times \{1, \ldots, u\})$

Daraus ergibt sich nun folgendes Modell:

$$\begin{aligned} f^{\text{Kant}}(x,y) = \sum_{k=1}^{u} y_k \to \min \text{ bei } & \sum_{k=1}^{u} x_{ik} = b_i && (i = 1, \ldots, m) \\ & \sum_{i=1}^{m} x_{ik} \ell_i \leq L \cdot y_k && (k = 1, \ldots, u) \\ & y_k \in \{0,1\} && (k = 1, \ldots, u) \\ & x_{ik} \in \mathbb{Z}_+ && ((i,k) \in \{1, \ldots, m\} \times \{1, \ldots, u\}) \end{aligned}$$

Die erste Nebenbedingung sorgt dafür, dass alle Teile gepackt werden; die zweite Nebenbedingung liefert die Einhaltung der Kapazität unter Berücksichtigung, dass nur bepackte Behälter gezählt werden.

Es kann stets $u = \sum_{i=1}^{m} b_i$ gewählt werden. Das Auffinden besserer Schranken ist im Allgemeinen schwierig. Eine Relaxation kann z.B. durch $y_k \in [0,1]$ und $x_{ik} \in \mathbb{R}_+$ erhalten werden. Diese liefert jedoch keine guten Näherungen.

Modell von Gilmore & Gomory Es seien J eine Indexmenge aller zulässigen Packungen und $x_j \in \mathbb{Z}_+$ $(j \in J)$ die Häufigkeit, wie oft ein Behälter nach dem durch j angegebenen Schema $a^j = (a_1^j, \ldots, a_m^j)$ mit $\ell^\top a^j \leq L$ gefüllt wird. Daraus ergibt sich folgendes Modell:

$$f^{\mathrm{GG}}(x) = \sum_{j \in J} x_j \to \min \quad \text{bei} \quad \sum_{j \in J} a_i^j \cdot x_j = b_i \qquad (i = 1, \ldots, m)$$
$$x_j \in \mathbb{Z}_+ \qquad (j \in J)$$

Die Nebenbedingung sorgt dafür, dass alle Teile gepackt werden.

Es gibt im Allgemeinen exponentiell viele zulässige Packungen a^j $(j \in J)$, deren Koeffizienten allesamt in den Nebenbedingungen benötigt werden.

Eine Relaxation erhält man zum Beispiel durch $x_j \in \mathbb{R}_+$. Diese stetige Relaxation ist sehr gut; man vermutet, dass folgende Bedingung gilt:

$$f^{\mathrm{GG},*} - f^{\mathrm{GG},*}_{\mathrm{relax}} < 2$$

Erfreulicherweise gibt es zum Gilmore-Gomory-Modell äquivalente Formulierungen, die mit einer polynomiellen Zahl von Variablen arbeiten und eine ebenso gute stetige Relaxation besitzen (z.B. Flussmodelle).

1.3.3 Standortplanung

Ein Dienstleister möchte neue Filialen aufbauen, um seine Kunden $k \in K := \{1, \ldots, m\}$ zu versorgen. Dabei sind aus der Menge $S := \{1, \ldots, n\}$ mögliche Standorte, die neuen Standorte so auszuwählen, dass der Bedarf aller Kunden befriedigt wird und die Gesamtkosten minimial sind.

Wir benötigen

- $c_s > 0$ … Fixkosten für den Aufbau von Standort $s \in S$
- $d_{ks} > 0$ … Kosten, um den Kunden $k \in K$ (vollständig) von Standort $s \in S$ zu beliefern.

Variablen:

- $x_s = \begin{cases} 1 & \text{wenn Standort } s \in S \text{ gebaut wird} \\ 0 & \text{sonst} \end{cases}$
- $y_{ks} \geq 0$ … Anteil des Bedarfs des Kunden $k \in K$, der vom Standort $s \in S$ bedient wird[1]

Modell zur Standortplanung:

$$f(x, y) = \underbrace{\sum_{s \in S} x_s c_s}_{\text{Fixkosten}} + \underbrace{\sum_{s \in S} \sum_{k \in K} y_{ks} d_{ks}}_{\text{variable Kosten}} \to \min$$

[1] implizit: $y_{ks} \in [0, 1]$

bei

$$\begin{aligned} \sum_{s\in S} y_{ks} &= 1 && (k \in K) \\ y_{ks} &\le x_s && (s \in S, k \in K) \\ x_s &\in \{0,1\} && (s \in S) \\ y_{ks} &\ge 0 && (k \in K, s \in S) \end{aligned}$$

1.3.4 Quadratisches Zuordnungsproblem

Es sollen n Personen auf n Räume verteilt werden. Person i muss Person j genau c_{ij} mal am Tag treffen. Außerdem habe Büro k von Büro ℓ die Entfernung $d_{k\ell} > 0$. Wird Person i das Büro k zugewiesen und Person j das Büro ℓ, so ergibt sich eine Gesamtwegstrecke von $2c_{ij}d_{kl}$ (beachte Hin- und Rückweg). Gesucht ist die wegminimale Belegung der Büros.

Variablen $x_{ik} = \begin{cases} 1 & \text{wenn Person } i \text{ das Büro } k \text{ zugewiesen wird} \\ 0 & \text{sonst} \end{cases}$ $\quad (i,k) \in \{1,\dots,n\}\times\{1,\dots,n\}$

Zielfunktion

$$f(x) = \sum_{i=1}^{n}\sum_{k=1}^{n}\sum_{\substack{j=1\\ j\neq i}}^{n}\sum_{\substack{\ell=1\\ \ell\neq k}}^{n} x_{j\ell}x_{ik}\cdot 2c_{ij}d_{kl} \to \min$$

bei

$$\begin{aligned} \sum_{i=1}^{n} x_{ik} &= 1 && (k = 1,\dots,n) && \text{Büro } k \text{ bekommt genau einen Einwohner} \\ \sum_{k=1}^{n} x_{ik} &= 1 && (i = 1,\dots,n) && \text{Person } i \text{ bekommt genau ein Büro} \\ x_{ik} &\in \{0,1\} && (i,k) \in \{1,\dots,n\}\times\{1,\dots,n\} \end{aligned}$$

Weitere Beispiele sind in der Vorlesung „Diskrete Optimierung“ (Master Mathe) zu finden..

Kapitel 2
GRUNDLAGEN

2.1 Existenz von Lösungen

Wir betrachten die Optimierungsaufgabe

$$f(x) \to \min \qquad \text{bei } x \in G \tag{2.1}$$

wobei folgende Bedingungen erfüllt seinen:

- f ist stetig (zumindest auf G)
- $G \subseteq \mathbb{R}^n$ ist kompakt
- $G \neq \emptyset$

Satz 2.1 (Weierstrass) Unter diesen Voraussetzungen existiert ein $x^* \in G$ mit

$$f^* := f(x^*) \leq f(x) \quad \forall x \in G$$

Beweis. Sei $f^* := \inf_{x \in G} f(x)$. Wegen $G \neq \emptyset$, finden wir eine Folge $\{f_k\}_{k \in \mathbb{N}} \subseteq \mathbb{R}$ mit $f_k = f(x_k) \geq f^*$ und $x_k \in G$ für alle $k \in \mathbb{N}$ und $\lim_{k \to \infty} f_k = f^*$. Die daraus resultierende Folge $\{x_k\}_{k \in \mathbb{N}}$ besitzt wegen der Kompaktheit von G eine konvergente Teilfolge $\{\widetilde{x}_k\}_{k \in \mathbb{N}} \subseteq \{x_k\}_{k \in \mathbb{N}}$ mit $\lim_{k \to \infty} \widetilde{x}_k = x^* \in G$ (Abgeschlossenheit von G). Die Stetigkeit von f ergibt nun $\lim_{k \to \infty} f(\widetilde{x}_k) = f(x^*) = f^*$ (insbesondere $f^* \in \mathbb{R}$) □

Beispiel 2.1

(1) Satz 2.1 anwendbar (G kompakt, Minimum existiert):

$$f(x_1, x_2) = x_1 - x_2 \to \min \text{ bei } x_1^2 + 4x_2^2 \leq 1$$

Der zulässige Bereich ist eine Ellipse mit Rand.

(2) Satz 2.1 nicht anwendbar (G unbeschränkt, kein Minimum, $f^* = -\infty$):

$$f(x_1, x_2) = x_1 - x_2 \to \min \text{ bei } x_1^2 + 4x_2^2 \geq 1$$

(3) Satz 2.1 nicht anwendbar (G unbeschränkt, kein Minimum, $f^* = 0$)

$$f(x_1, x_2) = \frac{1}{x_1} \to \min \text{ bei } x_2 \leq \frac{1}{x_1}, x_1 \geq 1, x_2 \geq 0$$

(4) Satz 2.1 nicht anwendbar (G unbeschränkt, Minimum existiert, $f^* = -1$)

$$f(x_1, x_2) = -\frac{1}{x_1} \to \min \text{ bei } x_2 \leq \frac{1}{x_1}, x_1 \geq 1, x_2 \geq 0$$

Offensichtlich besitzen also nicht alle Optimierungsaufgaben eine (globale) Lösung, insbesondere deshalb, weil Bedingung (1.2) ziemlich stark ist. Stattdessen hat sich in der Literatur auch der folgende „schwächere“ Lösungsbegriff etabliert.

Definition 2.1 Ein zulässiger Punkt $\overline{x} \in G$ heißt lokale Lösung von (2.1), falls ein $\rho > 0$ existiert mit

$$f(\overline{x}) \leq f(x) \quad \forall x \in G \cap B_\rho(\overline{x})$$

wobei $B_\rho(\overline{x}) := \{x \in \mathbb{R}^n : \|x - \overline{x}\|_2 \leq \rho\}$ die offene Kugel vom Radius ρ um $\overline{x}$ ist.

Bemerkung 2.1 Jede globale Lösung ist auch lokale Lösung. Die Umkehrung ist im Allgemeinen nicht korrekt.

Sofern eine globale Lösung existiert, ist diese in der Menge der lokalen Lösungen enthalten. Die Betrachtung lokaler Lösungen ist damit im Allgemeinen ausreichend. Für eine spezielle Klasse von Optimierungsaufgaben sind beide Lösungskonzepte sogar äquivalent. Dazu betrachten wir die folgenden Definitionen:

Definition 2.2 (Konvexität)

(1) $G \subseteq \mathbb{R}^n$ ist **konvex**, falls für alle $x, y \in G$ gilt

$$[x, y] := \{x(\lambda) \in \mathbb{R}^n : x(\lambda) = (1 - \lambda)x + \lambda y,\ \lambda \in [0, 1]\} \subseteq G$$

(2) Sei G konvex. Die Funktion $f \colon G \to \mathbb{R}$ heißt **konvex**, wenn gilt

$$f(x + \lambda(y - x)) \leq f(x) + \lambda\,(f(y) - f(x))$$

für alle $x, y \in G$ und $\lambda \in [0, 1]$.

(3) Sei G konvex. Eine Funktion $f \colon G \to \mathbb{R}$ heißt **streng konvex**, wenn gilt

$$f(x + \lambda(y - x)) < f(x) + \lambda\,(f(y) - f(x))$$

für alle $x, y \in G$ und $\lambda \in [0, 1]$.

Ausgehend von diesen Begrifflichkeiten erhalten wir das folgende Resultat:

Satz 2.2 Sei $G \subseteq \mathbb{R}^n$ eine konvexe Menge und $f\colon G \to \mathbb{R}$ eine konvexe Funktion.

(a) Jede lokale Lösung ist gleichzeitig auch globale Lösung von (2.1).

(b) Falls f sogar streng konvex ist, dann existiert höchstens eine Lösung.

Beweis. (zu a) Sei $\widetilde{x} \in G$ eine lokale Lösung von (2.1). Wir nehmen an, dass dies jedoch keine globale Lösung ist, d.h. es existiert ein $\overline{x} \in G$ mit $f(\overline{x}) < f(\widetilde{x})$. Wegen der Konvexität von G gilt dann $x(\lambda) = \widetilde{x} + \lambda(\overline{x} - \widetilde{x}) \in G$ für alle $\lambda \in [0,1]$. Mit der Konvexität von f folgt letzlich

$$f(x(\lambda)) = f(\widetilde{x} + \lambda(\overline{x} - \widetilde{x})) \overset{f \text{ konvex}}{\leq} f(\widetilde{x}) + \underbrace{\lambda}_{>0} \underbrace{(f(\overline{x}) - f(\widetilde{x}))}_{<0} < f(\widetilde{x}) \qquad \forall \lambda \in (0,1]$$

Somit ist $\widetilde{x}$ keine lokale Lösung im Widerspruch zur Annahme.

(zu b) Seien x, y zwei voneinander verschiedene Lösungen, d.h. $f(x) = f(y) = f^*$ für $x \neq y$. Wir erhalten $x(\lambda) \in G$ für alle $\lambda \in [0,1]$ und

$$f(x(\lambda)) = f(x + \lambda(y - x)) \overset{f \text{ streng konvex}}{<} f(x) + \lambda \underbrace{(f(y) - f(x))}_{=0}$$

Somit ist x keine Lösung. □

Für konvexe Optimierungsaufgaben sind lokale und globale Lösungen also äquivalent. Als wichtigen Spezialfall konvexer Mengen halten wir die folgende Darstellung fest.

Aussage 2.3 Sei G gegeben durch

$$G := \{x \in \mathbb{R}^n : g_i(x) \leq 0 \ (i \in I), \ \ h_j(x) = 0 \ (j \in J)\}$$

Dann gilt: falls alle Funktionen g_i $(i \in I)$ konvex und alle Funktionen h_j $(j \in J)$ affin-linear sind, dann ist G konvex.

Beweis. Seien $x, y \in G$ und $\lambda \in (0,1)$. G ist genau dann konvex, wenn $x(\lambda) \in G$:

$$g_i(x(\lambda)) = g_i(x + \lambda(y - x)) \leq g_i(x) + \lambda\,(g_i(y) - g_i(x)) = \underbrace{1 - \lambda}_{>0}\underbrace{g_i(x)}_{\leq 0} + \underbrace{\lambda}_{>0}\underbrace{g_i(y)}_{\leq 0} \leq 0$$

$$\begin{aligned} h_j(x(\lambda)) &= h_j(x + \lambda(y - x)) = A_j(x + \lambda(y - x)) + b_j = (1 - \lambda)A_j x + \lambda A_j y + b_j \\ &= (1 - \lambda)\underbrace{[A_j x + b_j]}_{h_j(x)=0} + \lambda\underbrace{[A_j y + b_j]}_{h_j(y)=0} \\ &= 0 \end{aligned}$$

Somit ist $x(\lambda) \in G$ und G also konvex. □

Jeder zulässige Bereich einer linearen Optimierungsaufgabe (↗ Kapitel 3) hat diese Gestalt.

2.2 Optimalitätsbedingungen

Definition 2.3 Eine Menge $K \subseteq \mathbb{R}^n$ heißt **Kegel**, falls gilt:

$$x \in K \quad \Rightarrow \quad \lambda x \in K \quad \forall \lambda \geq 0$$

Ein Kegel K ist ein **konvexer Kegel**, falls K eine konvexe Menge bzw. falls gilt

$$x, y \in K \quad \Rightarrow \quad x + y \in K$$

für alle $x, y \in K$. Der **Kegel der zulässigen Richtungen** $Z(\widetilde{x})$ ist definiert durch

$$Z(\widetilde{x}) := \{d \in \mathbb{R}^n \mid \exists\, \overline{t} := \overline{t}(\widetilde{x}, d) > 0 \text{ sodass } \widetilde{x} + td \in G \ \forall t \in [0, \overline{t}]\}$$

Für Optimierungsaufgaben ist der Kegel der zulässigen Richtungen von großer Bedeutung.

Aussage 2.4 (notwendiges Optimalitätskriterium) Ist f auf G stetig differenzierbar und $\widetilde{x} \in G$ ein lokales Minimum. Dann gilt

$$\nabla f(\widetilde{x})^\top \cdot d \geq 0 \qquad \forall d \in Z(\widetilde{x}) \tag{2.2}$$

Ist G konvex, dann erhält man die Bedingung

$$\nabla f(\widetilde{x})^\top (x - \widetilde{x}) \geq 0 \qquad \forall x \in G \tag{2.3}$$

Beweis. Sei $\widetilde{x}$ ein lokales Minimum und $d \in Z(\widetilde{x})$ eine zulässige Richtung. Dann existiert gemäß Definition ein $\overline{t}$, sodass $\widetilde{x} + td \in G$ für alle $t \in [0, \overline{t}]$ gilt. Weil außerdem $\overline{x}$ eine lokale Lösung ist, gibt es $\rho > 0$ mit $\rho < \overline{t}$ sodass $f(\widetilde{x} + td) \geq f(\widetilde{x})$ für $t \in (0, \rho)$ gilt. Aus dieser Ungleichung folgt

$$\frac{f(\widetilde{x} - td) - f(\widetilde{x})}{t} \geq 0 \qquad \forall t \in (0, \rho)$$

Durch Grenzwertbildung $t \to 0$ auf beiden Seiten erhält man mithilfe der Definition der Richtungsableitung und der Stetigkeit von f die Behauptung (2.2). Für konvexe Mengen gilt stets $x - \widetilde{x} \in Z(\widetilde{x})$ für $x \in G$, also folgt (2.3). □

Dieses Kriterium sagt aus, dass im Punkt $\widetilde{x}$ alle Richtungsableitungen (bezüglich zulässiger Richtungen) nicht-negativ sind, d.h. es keine zulässige Abstiegsrichtung gibt.

Bemerkung 2.2 Ein Punkt, der die Bedingung (2.2) erfüllt, heißt **stationärer Punkt**.

Bemerkung 2.3 Bei der freien Minimierung (d.h. für $G = \mathbb{R}^n$) ergibt sich wegen $Z(\widetilde{x}) = \mathbb{R}^n$ für alle $\widetilde{x} \in G$ die notwendige Bedingung

$$\widetilde{x} \text{ ist lokales Minimum} \quad \Rightarrow \quad \nabla f(\widetilde{x}) = 0$$

Wähle dafür $d \in \{\pm e^i\}_{i=1}^n$.

Für konvexe Optimierungsaufgaben gilt auch die Umkehrung in Bemerkung 2.3

Aussage 2.5 (hinreichendes Optimalitätskriterium) Es seien $G \subseteq \mathbb{R}^n$ sowie $f\colon G \to \mathbb{R}$ konvex und stetig differenzierbar. Falls ein $\widetilde{x} \in G$ existiert, welches der Bedingung (2.3) genügt, dann ist $\widetilde{x}$ (globales) Minimum von (2.1).

Beweis. Wenn f konvex und stetig differenzierbar ist und gilt

$$f(x) \geq f(\widetilde{x}) + \nabla f(\widetilde{x})^\top (x - \widetilde{x}) \qquad \forall x \in G$$

Wegen (2.3) folgt unmittelbar die (globale) Optimalität. Ausführlicher: siehe Übung. □

Im Fall polyedrischer zulässiger Mengen $G \subseteq \mathbb{R}^n$ (wie z.B. in der linearen Optimierung) kann die Bedingung (2.2) präzisiert werden, da dann $Z(x)$ eine einfache Struktur besitzt.

Definition 2.4 $G \subseteq \mathbb{R}^n$ heißt **polyedrisch**, falls eine Darstellung $G = \{x \in \mathbb{R}^n \colon Ax \leq b\}$ für eine geeignete Matrix A und einen geeigneten Vektor b existiert. Hierbei gilt

$$Ax \leq b \;\; :\Leftrightarrow \;\; \forall i \in I = \{1, \ldots, n\} : a_i^\top x = \sum_{j=1}^{n} a_{ij} x_j \leq b_i$$

Bemerkung 2.4 Eine polyedrische Menge G ist konvex und abgeschlossen, aber im Allgemeinen nicht beschränkt. Implizit können in der Beschreibung von G aus Definition 2.4 auch Gleichungsrestriktionen enthalten sein.

Definition 2.5 Für $x \in G$ ist die **Indexmenge der aktiven Restriktionen** definiert durch

$$I_0(x) := \left\{ i \in I \colon a_i^\top x = b_i \right\}$$

Sei nun ein zulässiger Punkt $x \in G$ gegeben. Damit eine beliebige Richtung $d \in \mathbb{R}^n$ zulässig ist, nuss ein $\bar{t} > 0$ existieren, sodass $x + td \in G$ für alle $t \in [0, \bar{t}]$ gilt. Für einen polyedrischen Bereich G ist dies äquivalent zu

$$\forall i \in I \colon a_i^\top (x + td) \leq b_i \;\; \Leftrightarrow \;\; \forall i \in I \colon t a_i^\top d \leq b_i - a_i^\top x$$

für alle $t \in [0, \bar{t}]$.

- Für alle inaktiven Restriktionen (also solche $a_i^\top x < b_i$) wäre $t a_i^\top d \leq b_i - a_i^\top x$ zu erfüllen. Egal, welchen Wert $a_i^\top d$ annimmt, es kann stets eine hinreichend kleine Schrittweite (im Sinne der Definition einer zulässigen Richtung) gefunden werden. Somit schränken inaktive Restriktionen die möglichen Richtungen $d \in \mathbb{R}^n$ *nicht* ein.
- Für aktive Restriktionen (also $a_i^\top x = b_i$) erhält man $t a_i^\top d \leq 0$, also (wegen $t > 0$) $a_i^\top d \leq 0$.

Diese Bedingung lässt sich geometrisch interpretieren: das Skalarprodukt der zulässigen Richtungen und des Normalenvektors (nach außen gerichtet) a_i der begrenzenden Hyperebene muss

kleiner oder gleich Null sein, d.h. der Schnittwinkel beider Vektoren liegt im Bereich $[\frac{\pi}{2}, \pi]$. Folglich zeigt die zulässige Richtung $d \in \mathbb{R}^n$ tatsächlich in das Innere von G.

Für einen zulässigen Punkt $x \in G$ kann somit folgende Beobachtung angegeben werden:

$$d \in Z(x) \quad \Leftrightarrow \quad \forall i \in I_0(x) \colon a_i^\top d \leq 0 \tag{2.4}$$

Außerdem ist die Größe $\widetilde{t}$ (maximale Schrittweite) wohldefiniert.

$$\widetilde{t} := \min \left\{ \frac{b_i - a_i^\top x}{a_i^\top d} \colon i \in I(x,d) \right\} \tag{2.5}$$

wobei $I(x,d) := \left\{ i \in I \colon a_i^\top d > 0 \right\}$.

Bemerkung 2.5 Falls $I(x,d) = \emptyset$, setzen wir $\widetilde{t} := \infty$.

Beispiel 2.2 Wir betrachten $x := (1,1,1)^\top$ und die polyedrische Menge

$$G := \left\{ (x_1, x_2, x_3)^\top \in \mathbb{R}^3 \colon x_1 + 2x_2 + x_3 \leq 4, 3x_1 + x_2 + x_3 \leq 6, x_i \geq 0, i = 1,2,3 \right\}$$

Offenbar gilt $x \in G$. Wir betrachten die Richtungen

$$d^1 = (1,1,1)^\top \text{ und } d^2 = (-1,-2,-1)^\top$$

Als aktive Restriktionen erkennen wir $I_0(x) = \{1\}$ (da nur die erste Nebenbedingung von G mit Gleichheit erfüllt ist).

- Für $d = d^1$ gilt
$$a_i^\top d = \begin{pmatrix} 1 \\ 2 \\ 1 \end{pmatrix}^\top \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix} = 4 > 0$$
Somit ist d^1 keine zulässige Richtung wegen (2.4).
- Für $d = d^2$ gilt
$$a_i^\top d = \begin{pmatrix} 1 \\ 2 \\ 1 \end{pmatrix}^\top \begin{pmatrix} -1 \\ -2 \\ -1 \end{pmatrix} = -6 \leq 0$$
Somit ist d^2 eine zulässige Richtung wegen (2.4). Zur maximalen Schrittweite: Die Ungleichung $(3,1,1)^\top (x + td) \leq 6$ liefert die Bedingung $t \in [-\frac{1}{6}, \infty)$. Aus $x + td \geq 0$ folgt die Bedingung $t \leq \frac{1}{2}$. Insgesamt gilt $\widetilde{t} = \frac{1}{2}$.

Zusammengefasst erhalten wir das folgende Resultat:

Folgerung 2.6 Sei G polyedrisch, d.h. $G = \{x \in \mathbb{R}^n \colon Ax \leq b\}$ und $f \colon G \to \mathbb{R}$ stetig differenzierbar. Ist $\widetilde{x}$ eine lokale Lösung von (2.1), so gilt

$$\nabla f(\widetilde{x})^\top \cdot d \geq 0 \qquad \forall d \in \mathbb{R}^n \text{ mit } a_i^\top d \leq 0 \ \forall i \in I_0(\widetilde{x}) \tag{2.6}$$

Ist f zusätzlich konvex, dann gilt auch die Umkehrung.

2.3 Das Lemma von FARKAS

Das folgende Resultat besitzt vielfältige Anwendungen in der Optimierung (↗ Dualität).

Lemma 2.7 (Farkas) Es seien $A \in \mathbb{R}^{m\times n}$ und $a \in \mathbb{R}^m$. Von den Systemen

$$\begin{array}{lll} \text{(i)} & Az \le 0 & a^\top z > 0 \\ \text{(ii)} & A^\top u = a & u \ge 0 \end{array}$$

ist *genau* eines lösbar.

Beweis. ▪ *höchstens* eines der Systeme ist lösbar: Seien (i) und (ii) lösbar. Dann gilt

$$0 < a^\top z = \left(A^\top u\right)^\top z = \underbrace{u^\top}_{\ge 0}\underbrace{Az}_{\le 0} \le 0 \quad \lightning$$

▪ *mindestens* eines der Systeme ist lösbar — die Unlösbarkeit von (ii) impliziert die Lösbarkeit von (i): Sei (ii) nicht lösbar. Dann gilt $a \notin K := \left\{x = A^\top u \colon u \ge 0\right\}$, wobei K ein konvexer, abgeschlossener Kegel ist. Wir betrachten die Optimierungsaufgabe

$$f(x) = \frac{1}{2}\|a - x\|_2^2 = \frac{1}{2}(a-x)^\top (a-x) \to \min \text{ bei } x \in K$$

Dann existiert eine eindeutige (und globale) Lösung $\overline{x} \in K$ mit

$$(1) \quad \nabla f(\overline{x})^\top \overline{x} = 0 \qquad\qquad (2) \quad \nabla f(\overline{x})^\top x \ge 0 \ \forall x \in K$$

Zunächst folgt gemäß Aussage 2.4, dass $\nabla f(\overline{x})^\top (x - \overline{x}) \ge 0$ für alle $x \in K$. Durch Einsetzen von $x = \frac{1}{2}\overline{x} \in K$ und $x = 2 \cdot \overline{x} \in K$ (beachte: K ist Kegel) erhält man (1). Dies wiederum lässt sich zur notwendigen Bedingung dazu addieren und man erhält (2). Nun zeigen wir, dass $z := a - \overline{x} \ne 0$ (wegen $a \notin K$) das System (i) löst. Es gilt $\nabla f(\overline{x}) = -z$ und damit folgt

$$0 = \nabla f(\overline{x})^\top \overline{x} = -z^\top \cdot (\overline{x} - a + a) = z^\top (z - a) \ \Rightarrow \ a^\top z = z^\top z \overset{z\ne 0}{>} 0$$

Weiter gilt $x \in K$ genau dann, wenn ein $u \ge 0$ existiert mit $x = A^\top u$. Aus (2) folgt dann

$$\begin{array}{lrll} \nabla f(\overline{x})^\top x \ge 0 \ \forall x \in K & \Rightarrow & -z^\top A^\top u \ge 0 & \forall u \ge 0 \\ & \Rightarrow & (Az)^\top u \le 0 & \forall u \ge 0 \\ & \Rightarrow & Az \le 0 & (\text{wähle z.B. wieder } u = e^1, e^2, \dots) \end{array}$$

Damit löst z das System (1). □

Damit können die notwendigen Optimalitätsbedingungen (2.6) bzw. äquivalent dazu

$$\forall i \in I_0(x) \colon a_i^\top \cdot d \le 0 \ \Rightarrow \ \nabla f(\widetilde{x})^\top \cdot d \ge 0 \tag{2.7}$$

wie folgt umformuliert werden: Offenbar ist (2.7) gleichbedeutend mit der Unlösbarkeit von

$$\nabla f(\widetilde{x})^\top \cdot d < 0, \qquad a_i^\top \cdot d \leq 0 \ \ \forall i \in I_0(\widetilde{x})$$

Wählt man also im Lemma von Farkas $a = -\nabla f(\widetilde{x})$ und A bestehend aus den Zeilen $a_i^\top$, so folgt die Lösbarkeit des Systems

$$\nabla f(\widetilde{x}) + \sum_{i \in I_0(\widetilde{x})} u_i a_i = 0 \qquad (u \geq 0) \tag{2.8}$$

Für konvexe Optimierungsaufgaben ist die Lösbarkeit von (2.8) sogar äquivalent dazu, dass $\widetilde{x}$ Lösung der betrachteten Aufgabe ist.

Gerade im Hinblick auf die praktische Anwendbarkeit ist (2.8) in der jetzigen Form wenig hilfreich, da $\widetilde{x}$ und damit $I_0(\widetilde{x})$ unbekannt sind. Man betrachtet daher oftmals die folgende äquivalente Umformulierung:

Lemma 2.8 Sei $G := \left\{x \in \mathbb{R}^n \colon a_i^\top x \leq b_i, i \in I\right\}$ und $f \colon G \to \mathbb{R}$ stetig differenzierbar. Wenn $x \in \mathbb{R}^n$ Lösung von

$$f(x) \to \min \text{ bei } x \in G$$

ist, dann existiert ein Vektor u, sodass das Paar (x, u) das folgende System löst:

$$\begin{aligned} \nabla f(x) + \sum_{i \in I} u_i a_i = 0 \qquad & u_i \geq 0, \qquad a_i^\top - b_i \leq 0 \quad (i \in I) \\ u_i \left(a_i^\top x - b_i\right) = 0 \qquad & (i \in I) \end{aligned} \tag{2.9}$$

Dabei beschreibt $a_i^\top - b_i \leq 0$ die Zulässigkeit von $x \in G$ und $u_i \left(a_i^\top x - b_i\right) = 0$ gleicht die zu große Indexmenge der Summe wieder aus, d.h. für inaktive Restriktionen folgt $u_i = 0$. Ist f konvex, so gilt auch die Umkehrung. Man nennt (2.9) auch ein **KKT-System**.

Bemerkung 2.6 (1) KKT steht für Karush-Kuhn-Tucker.

(2) Die Variablen u heißen **Lagrange-Multiplikatoren**.

(3) Gibt es neben den Ungleichungen auch Gleichungsrestriktionen $a_i^\top x = b_i$ für $i = m+1, \ldots, \overline{m}$ und $\overline{m} > m$, dann erhält man das KKT-System

$$\begin{aligned} \nabla f(x) + \sum_{i=1}^{m} u_i a_i + \sum_{i=m+1}^{\overline{m}} u_i a_i = 0 & \\ u_i \geq 0, a_i^\top x - b_i \leq 0 \qquad & (i = 1, \ldots, m) \\ a_i^\top x - b_i = 0 \qquad & (i = m+1, \ldots, \overline{m}) \\ u_i \left(a_i^\top x - b_i\right) = 0 \qquad & (i = 1, \ldots, m) \end{aligned} \tag{2.10}$$

Kapitel 3
LINEARE OPTIMIERUNG

Wir betrachten die Optimierungsaufgabe

$$z = c^\top x \to \min \text{ bei } x \in G := \{x \in \mathbb{R}^n : Ax = b, x \geq 0\} \tag{3.1}$$

mit $A \in \mathbb{R}^{m \times n}$, $c \in \mathbb{R}^n$, $b \in \mathbb{R}^m$. Außerdem nehmen wir an, dass $\operatorname{rg}(A) = m$ gilt und dass $m \leq n$ erfüllt ist.

Bemerkung 3.1 (1) G ist eine polyedrische Menge.

(2) Alle endlich-dimensionalen linearen Optimierungsaufgaben lassen sich **Standardform** (3.1) überführen (↗ Übung).

3.1 Basislösungen und Ecken

Sei $I := \{1, \ldots, n\}$. Da $\operatorname{rg}(A) = m$, existiert eine Indexmenge $I_B \subseteq I$ mit $|I_B| = m$ derart, dass alle Spalten A^i $(i \in I_B)$ linear unabhängig sind. I_B wird **Basis-Indexmenge** genannt. Mit $I_N := I \setminus I_B$ (Nichtbasis) definieren wir

$$\begin{aligned} A_B &= (A^i)_{i \in I_B} & A_N &= (A^i)_{i \in I_N} \\ c_B &= (c_i)_{i \in I_B} & c_N &= (c_i)_{i \in I_N} \\ x_B &= (x_i)_{i \in I_B} & x_N &= (x_i)_{i \in I_N} \end{aligned}$$

Dann lässt sich (3.1) schreiben als

$$z = c_B^\top x_B + c_N^\top x_N \to \min \text{ bei } A_B x_B + A_N x_N = b, x_b \geq 0, x_N \geq 0 \tag{3.2}$$

bzw. durch Auflösen der Gleichung nach x_B (beachte: A_B hat Vollrang) als

$$z = \left(c_N^\top - c_B^\top A_B^{-1} A_N\right) x_N + c_B^\top A_B^{-1} b \to \min \text{ bei } x_B = -A_B^{-1} A_N x_N + A_B^{-1} b, x_B \geq 0, x_N \geq 0 \tag{3.3}$$

Definition 3.1 Der Punkt

$$x = \begin{pmatrix} x_B \\ x_N \end{pmatrix} = \begin{pmatrix} A_B^{-1} b \\ 0 \end{pmatrix}$$

heißt **Basislösung** zu I_B. Gilt zusätzlich $A_B^{-1} b \geq 0$, dann heißt $x = (x_B, x_N)$ **zulässige Basislösung**.

Definition 3.2 Der Punkt $x \in G$ heißt **Ecke** (von G), falls aus $x = \frac{1}{2}(x^1 + x^2)$ mit $x^1, x^2 \in G$ stets $x = x^1 = x^2$ folgt.

Ecken des zulässigen Bereichs können also nicht durch andere zulässige Punkte linear kombiniert werden.

Zur Wiederholung benennen wir im Folgenden (ohne Beweis) einige Eigenschaften von Ecken und zulässigen Basislösungen.

Satz 3.1 Sei $\operatorname{rg}(A) = m$. Dann ist jede zulässige Basislösung auch Ecke von G. Umgekehrt gibt es zu jeder Ecke mindestens eine zulässige Basislösung.

Häufig unterscheidet man zwischen

- **degenerierten** (oder entarteten) Ecken, die mehrere zulässige Basislösungen besitzen
- **nicht-degenerierten** (oder nicht-entarteten) Ecken, die genau eine zulässige Basislösung besitzen.

Dabei gilt: Eine Ecke $x \in G$ ist genau dann degeneriert, wenn ein $i \in I_B$ mit $x_i = 0$ existiert.

Beispiel 3.1 Sei

$$G := \{x \in \mathbb{R}^n : x_1 + x_2 + x_3 = 1,\ \ 2x_1 + x_2 + x_4 = 2,\ \ x_1, \ldots, x_4 \geq 0\}$$

Hierbei ist die Ecke $E_1 = (0, 1, 0, 1)^\top$ nicht degeneriert, da sie nur die Zerlegung $I_B = \{2, 4\}$ und $I_N = \{1, 3\}$ gestattet. Die Ecke $E_2 = (1, 0, 0, 0)^\top$ ist degeneriert, weil ein $i \in I_B$ zwangsläufig $x_i = 0$ erfüllen muss.

Satz 3.2 Seo $G \neq \emptyset$. Dann besitzt G

(1) mindestens eine Ecke
(2) höchstens endlich viele Ecken.

Beweis. siehe Übung □

Satz 3.3 Ist (3.1) lösbar, dann gibt es eine Ecke von G, die (3.1) löst.

Bei linearen Optimierungsaufgaben genügt es daher die Ecken von G zu betrachten. Ist die Aufgabe lösbar, so findet man durch systematisches Abschreiten der Ecken eine Lösung. Um dabei zu erkennen, ob Optimalität vorliegt, hilft folgendes Resultat:

Aussage 3.4 (Optimalitätskriterium) Gilt für die zuässige Basislösung $x = (x_B, x_N) = (A_B^{-1}b, 0)$ die Bedingung

$$c_N^\top - c_B^\top A_B^{-1} A_N \geq 0 \tag{3.4}$$

dann ist x Lösung von (3.1).

Beweis. Sei $x = (x_B, x_N)$ eine zulässige Basislösung. Wir zeigen zunächst:

$$Z(x) \subseteq \{d \in \mathbb{R}^n \colon Ad = 0, d_N \geq 0\}$$

Sei $d \in Z(x)$. Dann existiert $t > 0$ mit $A(x + td) \overset{!}{=} b$ (beachte die Definition von G mit Gleichheitsrestriktionen). Es gilt

$$Ax + tAd = b \;\Leftrightarrow\; b + tAd = b \overset{t>0}{\Leftrightarrow} Ad = 0$$

Wegen $x_N = 0$ ergibt sich aus $x + td \overset{!}{\geq} 0$ (nach Definition von G) sofort $d_N \geq 0$. Insbesondere gilt

$$Ad = 0 \;\Leftrightarrow\; A_B d_B + A_N d_N = 0 \;\Leftrightarrow\; d_B = -A_B^{-1} A_N d_N \qquad \forall d \in Z(x)$$

Damit folgt unter Berücksichtigung von (3.4)

$$\begin{aligned} \nabla f(x)^\top d &= c^\top d \\ &= c_B^\top d_B + c_N^\top d_N \\ &= -c_B^\top A_B^{-1} A_N d_N + c_N^\top d_N \\ &= \underbrace{\left(c_N^\top - c_B^\top A_B^{-1} A_N\right)}_{\geq 0} \underbrace{d_N}_{\geq 0} \geq 0 \qquad \forall d \in Z(x) \end{aligned}$$

d.h. x genügt der notwendigen Optimalitätsbedingung (2.2), die hier (im konvexen Fall) auch hinreichend ist. □

Eine entsprechende Systematik zum Abschreiten der Ecken wird im Folgenden Abschnitt behandelt.

3.2 Das primale Simplex-Verfahren

Das primale Simplexverfahren durchläuft zwei Phasen (falls nötig):

- Phase 1 besteht aus der Ermittlung einer ersten Ecke (zulässige Basislösung),
- Phase 2 aus der darauf aufbauenden Bestimmung einer optimalen Ecke.

3.2.1 Phase 2 des Simplex-Verfahrens

Wir betrachten die (erste) zulässige Basislösung (Ecke) $x = (x_B, x_N)$ und schreiben (3.3) als Simplex-Tableau:

T_0	x_N	1
$x_B =$	P	p
$z =$	$q^\top$	q_0

$$\begin{aligned} P &= -A_B^{-1}A_N & p &= A_B^{-1}b \\ q^\top &= c_N^\top - c_B^\top A_B^{-1}A_N & q_0 &= c_B^\top A_B^{-1}b \end{aligned} \tag{3.5}$$

Wir nehmen zunächst an, dass $x = (x_B, x_N)$ eine nicht-entartete Ecke mit $x_B = (x_1, \ldots, x_m)^\top$ und $x_N = (x_{m+1}, \ldots, x_n)^\top$ ist. Die hierzu gehörige Basislösung ist $x = (x_B, x_N) = (p, 0)$ und es gilt $p \geq 0$ (da zulässig). Folglich ist $x \in G$.

Frage: Wenn x nicht optimal ist – wie kann eine bessere zulässige Basislösung (Ecke) gefunden werden?

Antwort: Wahl einer zulässigen Richtung $d \in Z(x)$ mit maximaler Schrittweite, die eine Verkleinerung des Zielfunktionswerts ermöglicht.

Nach Aussage 3.4 ist x optimal, falls $q \geq 0$ gilt. Sei nun $q_\tau < 0$ für $\tau \in I_N$. Zur Konstruktion einer neuen Ecke setzen wir $x_\tau = t$ (bisher war $x_\tau = 0$). Dann folgt zunächst $x_N(t) = t \cdot e_\tau$ und wegen der Forderung $x_N(t) \geq 0$ auch $t \geq 0$. Ferner ergibt sich aus Tableau T_0 der Zusammenhang $x_i(t) = P_{i\tau} \cdot x_\tau + p_i = P_{i\tau} \cdot t + p_i$ für alle $i \in I_B$.

Insgesamt verfolgen wir ausgehend von $x = (p, 0)$ die zulässige Richtung $d \in \mathbb{R}^n$

$$d_i = \begin{cases} P_{i\tau} & i \in I_B \\ 1 & i = \tau \\ 0 & i \in I_N \setminus \{\tau\} \end{cases}$$

Die maximale Schrittweite $\bar{t}$ erhält man wie folgt: Für jedes $i \in I_B$ ist $x_i(t) \geq 0$ zu gewährleisten. Gilt $P_{i\tau} \geq 0$, so ergibt dies keine Einschränkung für die Schrittweite (weil $p_i \geq 0$, $t \geq 0$, $P_{i\tau} \geq 0$ $\Rightarrow$ $x_i(t) \geq 0$ für alle $t \geq 0$). Für $P_{i\tau} < 0$ muss hingegen $t \leq -\frac{p_i}{P_{i\tau}}$ (aus Tableauzusammenhang) gewählt werden. Die maximal mögliche Schrittweite ergibt sich folglich zu

$$t \leq \bar{t} = \bar{t}(x, d) := \min\left\{ -\frac{p_i}{P_i\tau} : P_{i\tau} < 0, i \in I_B \right\} \tag{3.6}$$

bzw. $\bar{t} = \infty$, falls $P_{i\tau} \geq 0$ für alle $i \in I_B$.

Aussage 3.5 Im Fall $\bar{t} = \infty$ besitzt (3.1) keine Lösung, da die Zielfunktion nach unten unbeschränkt ist.

Beweis. Wegen $\bar{t} = \infty$ gilt $x(t) \in G$ für alle $t \geq 0$. Dann liefert $q_\tau < 0$ sogleich $Z(t) = \overline{q} \cdot x_N(t) + q_0 \overset{x_N(t) = t \cdot e_\tau}{=} q_\tau \cdot t + q_0 \to -\infty$ für $t \to \infty$. □

Bemerkung 3.2 Die beiden Fälle

(1) $q_i \geq 0$ für alle $i \in I_N$ $\rightsquigarrow$ Optimalität

(2) es existiert ein $\tau \in I_N$ mit $q_\tau < 0$ und $P_{i\tau} \geq 0$ für alle $i \in I_B$ $\rightsquigarrow$ Unbeschränktheit

werden primal entscheidbar genannt.

Im sogenannten nicht-entscheidbaren Fall, d.h. falls

$$(\exists\, \tau \in I_N \colon a_\tau < 0) \wedge \left(\exists\, \sigma \in I_B \colon \bar{t} = -\frac{p_\sigma}{P_{\sigma\tau}} = \min\left\{-\frac{p_i}{P_{i\tau}} \colon P_{i\tau<0, i\in I_B}\right\} < \infty\right)$$

ergibt die (maximale) Schrittweite $\bar{t}$ den Punkt

$$\overline{x} = x + \bar{t}d \in G \text{ mit } f(\overline{x}) = f(x) + \bar{t}q_\tau = q_0 + \bar{t}q_\tau$$

Für entartete Ecken kann man die Schrittweite $\bar{t} = 0$ erhalten. In diesem Fall ändert sich der Punkt $\overline{x}$ nicht, aber die Menge I_N und I_B. Zum Verlassen einer (noch nicht optimalen) entarteten Ecke können mehrere Schritte nötig sein.

Satz 3.6 $\overline{x}$ ist eine Ecke von G mit Basis-Indexmenge

$$\overline{I_B} = \overline{I_B}(\overline{x}) := (I_B \setminus \{\sigma\}) \cup \{\tau\}$$
$$\overline{I_N} = \overline{I_N}(\overline{x}) := (I_N \setminus \{\tau\}) \cup \{\sigma\}$$

Um zu zeigen, dass die Matrix $\overline{A_B} := (A')_{i\in\overline{I_B}}$ regulär ist, nutzen wir das folgende Resultat.

Lemma 3.7 (Sherman / Morrison) Es seien $B \in \mathbb{R}^{m\times m}$ regulär und $u, v \in \mathbb{R}^m$. Die Matrix $\overline{B} := B + uv^\top$ ist genau dann regulär, wenn $1 + v^\top B^{-1}u \neq 0$ erfüllt ist und dann gilt

$$\overline{B}^{-1} = B^{-1} - \frac{B^{-1}uv^\top B^{-1}}{1 + v^\top B^{-1}u}$$

Beweis. Übung oder Selbststudium (aber wird nie gefragt werden) □

Beweis (Satz 3.6). Der "Austausch" der Spalten A^τ und A^σ kann durch ein dyadisches Produkt $uv^\top$ beschrieben werden:

$$\overline{A_B} = A_B + uv^\top \text{ mit } u = A^\tau - A^\sigma \text{ und } v = e^\sigma$$

Wegen

$$\begin{aligned} 1 + v^\top B^{-1}u &= 1 + (e^\sigma)^\top A_B^{-1}(A^\tau - A^\sigma) \\ &= 1 + (e^\sigma)^\top A_B^{-1}A^\tau - 1 && (\text{"}A^\sigma \in A_B\text{"}) \\ &= -P_{\sigma\tau} \neq 0 && (P = -A_B^{-1}A_N \text{ und "}A^\tau \in A_N\text{"}) \end{aligned}$$

folgt aus Lemma 3.7 die Regularität von $\overline{A_B}$. □

Beispiel 3.2 Wir betrachten $z = -x_1 - x_2 \to \min$ bei $x_1 + 2x_2 \leq 6$, $4x_1 + x_2 \leq 10$, $x_1, x_2 \geq 0$. Um aus den Ungleichungen Gleichungsnebenbedingungen zu machen, führen wir sogenannte Schlupfvariablen $x_3, x_4 \geq 0$ ein und erhalten

$$x_1+2x_2+x_3 = 6, \quad 4x_1+x_2+x_4 = 10, \quad x_1, x_2, x_3, x_4 \geq 0 \;\Rightarrow\; x_3 = 6-x_1-2x_2 \text{ und } x_4 = 10-4x_1+x_2$$

Damit liegt nun eine Optimierungsaufgabe in Standardform (3.1) vor. Notieren wir dies nun in Tableauform mit $I_N = \{1, 2\}$ und $I_B = \{3, 4\}$ (betrachte dazu $x_B = Px_N + p$):

T_0	x_1	x_2	1	
$x_3 =$	-1	-2	6	$\bar{t} = \frac{6}{1}$
$x_4 =$	-4	-1	10	$\bar{t} = \frac{10}{4}$
$z =$	-1	-1	0	

Wir können nun $\tau \in \{1, 2\}$ wählen, oBdA wählen wir hier $\tau = 1$. Für x_3 ergibt sich eine maximale Schittweite $\bar{t} = -\frac{p_i}{P_{i\tau}} = \frac{6}{1}$. Für x_4 ergibt sich $\bar{t} = \frac{10}{4}$. Damit wird $\sigma = 4$ gewählt.

Mit $\tau = 1$ und $\sigma = 4$ sowie $\bar{t} = \frac{5}{2}$ erhält man

$$\overline{x} = x + \bar{t}d = \begin{pmatrix} 0 \\ 0 \\ 6 \\ 10 \end{pmatrix} + \frac{5}{2} \begin{pmatrix} 1 \\ 0 \\ -1 \\ -4 \end{pmatrix} = \begin{pmatrix} \frac{5}{2} \\ 0 \\ \frac{7}{2} \\ 0 \end{pmatrix}$$

und

$$z(\overline{x}) = -\frac{5}{2}$$

Der Austausch von x_τ und x_σ im Simplextableau kann formal durch die sogenannten Austauschregeln erfolgen.

T_0	x_N	1
$x_B =$	P	p
$z =$	$q^\top$	q_0

$\Longrightarrow$

T_1	$x_{\widetilde{N}}$	1
$x_{\widetilde{B}} =$	$\widetilde{P}$	$\widetilde{p}$
$z =$	$\widetilde{q}^\top$	$\widetilde{q_0}$

$$\widetilde{I_B} = (I_B \cup \{\tau\}) \setminus \{\sigma\}$$
$$\widetilde{I_N} = (I_B \cup \{\sigma\}) \setminus \{\tau\}$$

Austauschregeln:

$$\widetilde{P}_{\sigma,\tau} := \frac{1}{P_{\sigma,\tau}} \qquad \text{(Pivotelement)}$$

$$\widetilde{P}_{\sigma,j} := -\frac{P_{\sigma,j}}{P_{\sigma,\tau}} \quad (j \in I_N \setminus \{\tau\}) \qquad \widetilde{p}_\sigma = -\frac{p_\sigma}{P_{\sigma,\tau}} \qquad \text{(Pivotzeile)}$$

$$\widetilde{P}_{i,\tau} := -\frac{P_{i,\tau}}{P_{\sigma,\tau}} \quad (i \in I_B \setminus \{\sigma\}) \qquad \widetilde{q}_\tau := \frac{q_\tau}{P_{\sigma,\tau}} \qquad \text{(Pivotspalte)}$$

$$\widetilde{P}_{i,j} := P_{i,j} - \frac{P_{\sigma,j}}{P_{\sigma,\tau}} P_{i,\tau} \quad (i \in I_B \setminus \{\sigma\}, j \in I_N \setminus \{\tau\}) \qquad \text{(sonstige Elemente)}$$

$$\widetilde{p}_i := p_i - \frac{p_\sigma}{P_{\sigma,\tau}} P_{i,\tau} \quad (i \in I_B \setminus \{\sigma\})$$

$$\widetilde{q}_j := q_j - \frac{P_{\sigma,j}}{P_{\sigma,\tau}} q_\tau \quad (j \in I_N \setminus \{\tau\})$$

Vergleiche dazu auch das Merkblatt zum Simplex-Verfahren unter
`https://www.math.tu-dresden.de/~martinovic/Zusammenfassung_Simplexverfahren.pdf`

Beispiel 3.3 Wir betrachten wie in Beispiel 3.2 die Optimierungsaufgabe $z = -x_1 - x_2 \to \min$ bei $x_1 + 2x_2 \le 6$, $4x_1 + x_2 \le 10$, $x_1, x_2 \ge 0$. mit Simplex-Starttableau:

T_0	x_1	x_2	1	
$x_3 =$	-1	$\boxed{-2}$	6	$\bar{t} = 3$
$x_4 =$	-4	-1	10	$\bar{t} = 10$
$z =$	-1	-1	0	
Kellerzeile	$-\frac{1}{2}$	$*$	3	= neue Pivotzeile

Nun wählen wir aber $\tau = 2$, woraus sich $\sigma = 3$ ergibt. Zur besseren Übersicht haben wir eine Kellerzeile eingeführt. Diese entspricht genau der neu berechneten Pivotzeile.

T_1	x_1	x_3	1	
$x_2 =$	$-1/2$	$-1/2$	3	(Division durch -1 * Pivot)
$x_4 =$	$-7/2$	$1/2$	7	
$z =$	$-1/2$	$1/2$	-3	
Kellerzeile	$-2/7$	$1/7$	2	

Nebenrechnung: z.B. $7 = 10 + 3 \cdot (-1)$

Im nächsten Schritt wählen wir nun $\tau = 1$ und $\sigma = 4$.

T_2	x_4	x_3	1
$x_2 =$	$1/7$	$-4/7$	2
$x_1 =$	$-2/7$	$1/7$	2
$z =$	$1/7$	$3/7$	-4

Da $\widetilde{p} = \binom{2}{2} \ge 0$ ist, ist die Lösung zulässig. Außerdem wissen wir wegen $\widetilde{q}^\top = (1/7, 3/7) \ge 0$, dass

die Lösung optimal ist. Somit ergibt sich

$$x^* = (x_1^*, x_2^*, x_3^*, x_4^*) = (2, 2, 0, 0) \text{ mit } z^* = -4$$

3.2.2 Phase 1 (Hilfsfunktionsmethode)

Wir betrachten das Problem

$$z = c^\top x \to \min \text{ bei } Ax = b, x \geq 0 \tag{3.7}$$

Ohne Einschränkung sei $b \geq 0$. Durch folgendes Hilfsproblem lässt sich eine Startecke ermitteln (sofern eine solche überhaupt existiert).

$$h = e^\top y \to \min \text{ bei } y + Ax = b, x \in \mathbb{R}^n_+, y \in \mathbb{R}^m_+ \tag{3.8}$$

mit $e = (1, \ldots, 1)^\top \in \mathbb{R}^m$. Eine erste Basislösung für (3.8) ist gegeben durch

T_0	x	1
$y =$	$-A$	b
$h =$	$-e^\top A$	$e^\top b$

(3.9)

Satz 3.8 Das Problem (3.7) besitzt genau dann eine zulässige Lösung, wenn $h_{\min} = 0$ den Optimalwert von (3.8) darstellt.

Beweis. Offenbar gilt $h_{\min} = 0 \;\Leftrightarrow\; y = 0$.

(⇒) Besitzt (3.7) eine zulässige Lösung $\widetilde{x}$, dann ist $\begin{pmatrix} \widetilde{x} \\ 0 \end{pmatrix}$ zulässig für (3.8). Wegen $0 \leq h = e^\top \widetilde{y} = 0$ folgt $h_{\min} = 0$.

(⇐) Hat man umgekehrt $h_{\min} = 0$, so gilt $\widetilde{y} = 0$ für jede optimale Lösung $\begin{pmatrix} \widetilde{x} \\ \widetilde{y} \end{pmatrix}$ von (3.8). Aus der Zulässigkeit von $\begin{pmatrix} \widetilde{x} \\ \widetilde{y} \end{pmatrix}$ für (3.8) folgt dann die Zulässigkeit von $\widetilde{x}$ für (3.7). □

3.2.3 Der Simplexalgorithmus

Mit den zuvor beschriebenen Vorgehensweisen lässt sich das Simplexverfahren zur Lösung der Optimierungsaufgabe (3.1) wie folgt algorithmisch formulieren:

- **Schritt 1** (*Initialisierung*): Ermittle eine erste zulässige Basislösung $x = \left(x_B^\top, x_N^\top\right)^\top = \left(p^\top, 0^\top\right)^\top$ mit $p = (A_B)^{-1} b \geq 0$, wobei I_B die Menge der Basisindizes ist und stelle ein erstes Simplextableau auf.
- **Schritt 2** (*Optimalitätstest*): Berechne entsprechend Aussage 3.4

$$\overline{q} := \min_{j \in I_N} q_j \quad \text{mit} \quad q_J := c_j d^\top A^j \;\; (j \in I_N) \tag{3.10}$$

wobei $d^\top := c_B^\top (A_B)^{-1}$ ist. Gilt $\overline{q} \geq 0$, dann ist x Lösung von (3.1). Andernfalls sei $q_\tau = \overline{q} < 0$.

- **Schritt 3** (*Test auf Unbeschränktheit*): Gilt $P_{i\tau} \geq 0$ für alle $I \in I_B$, so ist die Aufgabe nicht lösbar ($f^* = -\infty$).
- **Schritt 4** (*Austauschschritt*): Bestimme die Pivotzeile σ gemäß

$$-\frac{p_\sigma}{P_{\sigma\tau}} = \min\left\{-\frac{p_i}{P_{i\tau}} : \overline{P}_{i\tau} < 0, i \in I_B\right\}$$

und führe den Austauschschritt $\sigma \leftrightarrow \tau$ (Aktualisierung Simplextableau) durch. Gehe zu Schritt 2.

Bemerkung 3.3 (i) Der Simplexalgorithmus löst Problem (3.1) nach endlich vielen Schritten exakt oder stellt dessen Unlösbarkeit fest.

(ii) Pro Simplexschritt ist im Wesentlichen die Matrix P (der Dimension $m \times (n-m)$) zu transformieren. Für $n \gg m$ kann das recht aufwendig sein, sodass ggf. alternative Varianten des Simplexalgorithmus' (z.B. das revidierte Simplexverfahren oder die Technik der Spaltengenerierung) effizienter sind.

(iii) Der Test auf Unbeschränktheit der Zielfunktion kann auch für jede Spalte $j \in I_N$ mit $q_j < 0$ erfolgen, sofern dies nicht zu aufwendig ist.

3.3 Das duale Simplexverfahren

Nach Aussage 3.4 ist ein Tableau

T_0	x_N	1
$x_B =$	P	p
$z =$	$q^\top$	q_0

optimal, wenn $p \geq 0$ und $q \geq 0$ gelten. Nach Konstruktion gilt beim primalen Simplexverfahren stets $p \geq 0$. Sei nun ein Tableau T_0 gegeben mit $q \geq 0$, aber *nicht* $p \geq 0$, d.h. es gibt eine Zeile $\sigma \in I_B$ mit $p_\sigma < 0$. Die zu T_0 gehörige Basislösung ist dann *nicht* zulässig. Mithilfe des dualen Simplexverfahrens lässt sich jedoch (unter Beibehaltung von $q \geq 0$) eine zulässige Basislösung (d.h. mit "$p \geq 0$) erzeugen. Entsprechend der bekannten Austauschregeln ergeben sich folgende Bedingungen:

$$\widetilde{q}_j := q_j - \frac{P_{\sigma,j}}{P_{\sigma,\tau}} q_\tau \overset{!}{\geq} 0 \qquad \forall j \in I_N \setminus \{\tau\}$$

$$\widetilde{q}_\tau := \frac{q_\tau}{P_{\sigma,\tau}} \qquad \overset{!}{\geq} 0$$

$$\widetilde{p}_\sigma := -\frac{p_\sigma}{P_{\sigma,\tau}} \qquad \overset{!}{\geq} 0$$

Wegen $p_\sigma < 0$ und $q_\tau \geq 0$ ist somit ein Pivotelement mit $P_{\sigma,\tau} > 0$ zu wählen. Zur Sicherstellung von $\widetilde{q}_j \geq 0$ für alle $j \in I_N \setminus \{\tau\}$ muss ferner gelten

$$\frac{q_\tau}{P_{\sigma,\tau}} = \min\left\{ \frac{q_j}{P_{\sigma,j}} \colon P_{\sigma,j} > 0, j \in I_N \right\}$$

Die eigentlichen Austauschregeln sind analog zu denen des primalen Simplexverfahrens.

Bemerkung 3.4 Da dieses Verfahren mit einem *unzulässigen* Punkt startet, ist die Folge der Zielfunktionswerte (im Gegensatz zum primalen Simplexverfahren) nicht monoton fallend.

Bemerkung 3.5 Falls eine zulässige Basislösung gefunden wird, so ist diese zwangsläufig optimal.

Beispiel 3.4 Betrachten wir die Optimierungsaufgabe

$$\begin{aligned} z = 6x_1 + 5x_2 + 12x_3 + 8x_4 + 9x_5 \to \min \quad \text{bei} \quad & x_1 + x_3 + x_4 + x_5 \geq 300, \\ & x_2 + 2x_3 + x_4 \geq 400, \\ & x_i \geq 0 \qquad \forall i = 1, \ldots, 5 \end{aligned}$$

Um daraus Gleichungsrestriktionen zu machen, führen wir Schlupfvariablen $x_6, x_7 \geq 0$ ein, d.h.

$$\begin{aligned} x_1 + x_3 + x_4 + x_5 &= 300 + x_6, \\ x_2 + 2x_3 + x_4 &= 400 + x_7, \\ x_i &\geq 0 \qquad \forall i = 1, \ldots, 7 \end{aligned}$$

Daraus ergibt sich nun folgendes Tableau

T_0	x_1	x_2	x_3	x_4	x_5	1	
$x_6 =$	1	0	1	1	1	-300	
$x_7 =$	0	$\boxed{1}$	2	1	0	-400	
$z =$	6	5	12	8	9	0	$\leftarrow \sigma = 7$
Keller	0	$*$	-2	-1	0	400	

Zur Wahl von $\tau = 2$: $\frac{5}{1}$, $\frac{12}{6} = 6$, $\frac{8}{1} = 8$. Dabei ist 5 minimal, also $\tau = 2$. Fahren wir nun mit den weiteren Tableaus fort:

T_1	x_1	x_7	x_3	x_4	x_5	1	
$x_6 =$	1	0	$\boxed{1}$	1	1	-300	$\leftarrow \sigma = 6$
$x_2 =$	0	1	-2	-1	0	400	
$z =$	6	5	2	3	9	2000	
Keller	0	$*$	$*$	-1	0	400	

T_2	x_1	x_7	x_6	x_4	x_5	1	
$x_3 =$	1	0	1	-1	1	300	
$x_2 =$	2	1	-2	$\boxed{1}$	2	-200	$\leftarrow \sigma = 2$
$z =$	4	5	2	1	7	2600	
Keller	-2	-1	2	$\tau =$ 4	-2	200	

T_3	x_1	x_7	x_6	x_2	x_5	1
$x_3 =$	1	1	-1	-1	1	100
$x_4 =$	-2	-1	2	1	-2	200
$z =$	2	4	4	1	5	2800

Somit ergibt sich die Lösung

$$x^* = (0, 0, 100, 200, 0, 0, 0)^\top \quad \text{und} \quad z^* = 2800$$

3.4 Dualität

Wir betrachten nun die Optimierungsaufgabe

$$\begin{aligned} &c^\top x \to \min \text{ bei } Ax \leq b \text{ und } x \in \mathbb{R}^n_+ \\ &I := \{1, \ldots, m\} \text{ und } J := \{1, \ldots, n\} \end{aligned} \tag{P}$$

Satz 3.9 (Charakterisierungssatz) Ein Punkt $x \in \mathbb{R}^n$ ist genau dann Lösung von (P), wenn ein $\overline{x} \in \mathbb{R}^m$ existiert, sodass insgesamt das folgende System gelöst wird:

$$\begin{aligned} A\overline{x} - b \leq 0 \qquad & \overline{x} \geq 0 \qquad (1) \\ A^\top \overline{u} + c \geq 0 \qquad & \overline{u} \geq 0 \qquad (2) \\ \overline{u}^\top (A\overline{x} - b) = 0 \qquad & \overline{x}^\top \left(A^\top \overline{u} + c\right) = 0 \qquad (3) \end{aligned}$$

Beweis. Die vorliegende Optimierungsaufgabe ist äquivalent zu

$$f(x) = c^\top x \to \min \text{ bei } \underbrace{\begin{pmatrix} A \\ -\mathbb{1}_n \end{pmatrix}}_{=\widetilde{A} \in \mathbb{R}^{(m+n)\times n}} x \leq \underbrace{\begin{pmatrix} b \\ 0 \end{pmatrix}}_{=\widetilde{b} \in \mathbb{R}^{m+n}} \tag{P'}$$

Gemäß Lemma 2.8 ist x genau dann Lösung von (P') (und (P)), wenn ein Vektor $w = \left(\begin{smallmatrix} u \\ v \end{smallmatrix}\right) \in \mathbb{R}^{m+n}$ existiert mit

$$\begin{aligned} \nabla f(x) + \sum_{i \in I \cup J} w_i \widetilde{a}_i &= 0 \\ w_i &\geq 0 \qquad (i \in I \cup J) \\ \widetilde{a}_i^\top x - \widetilde{b}_i &\leq 0 \qquad (i \in I \cup J) \\ w_i \left(\widetilde{a}_i^\top x - b_i\right) &= 0 \qquad (i \in I \cup J) \end{aligned} \tag{KKT}$$

Trennung von I und J führt zu

$$c + \sum_{i \in I} u_i a_i + \sum_{j \in J} v_j (-e^j) = 0 \tag{KKT}$$

$$\begin{aligned} u_i &\geq 0 \quad (i \in I) & \qquad v_j &\geq 0 \quad (j \in J) \\ a_i^\top x - b_i &\leq 0 \quad (i \in I) & \qquad \left(-e^j\right)^\top x - 0 &\leq 0 \quad (j \in J) \\ u_i \left(a_i^\top x - b_i\right) &= 0 \quad (i \in I) & \qquad v_j \left(\left(-e^j\right)^\top x - 0\right) &= 0 \quad (j \in J) \end{aligned}$$

Überführt man dieses System in eine Matrix-Vektor-Schreibweise, so ergibt sich

$$\begin{aligned} c + A^\top u - v &= 0 \\ u, v, x &\geq 0 \\ Ax - b &\leq 0 \\ u^\top (Ax - b) &= 0 \\ x^\top v &= 0 \end{aligned}$$

Durch Umstellen der ersten Gleichung nach v lässt sich diese Variable im System "eliminieren" und wir erhalten die Behauptung. □

Definition 3.3 Das Problem

$$\text{(D)} \qquad z_D = -b^\top u \to \max \qquad \text{bei} \qquad A^\top u \geq -c, \;\; u \in \mathbb{R}^m_+ \tag{3.11}$$

heißt duale Optimierungsaufgabe zu (P).

Begründung: Die Anwendung von Satz 3.9 auf (D) ergibt das selbe KKT-System wie im Falle von (P). Dazu müssen wir (D) umformulieren als Minimierungsaufgabe

$$-z_D = b^\top u \to \min \qquad \text{bei} \qquad -A^\top u \leq c, \;\; u \in \mathbb{R}^m_+$$

(damit die selbe Form wie in Satz 3.9 vorliegt). Einsetzen in (1) - (3) ergibt

$$\begin{aligned} (1) \rightsquigarrow & \quad -A^\top u - c \leq 0, & & u \geq 0 \\ (2) \rightsquigarrow & \quad \left(-A^\top\right)^\top y + b \geq 0, & & y \geq 0 \\ (3) \rightsquigarrow & \quad y^\top \left(-A^\top u - c\right) = 0, & & u^\top \left(\left(-A^\top\right)^\top y + b\right) = 0 \end{aligned}$$

Umformulierung liefert

$$\begin{aligned} (1) \rightsquigarrow & \quad A^\top u + c \geq 0, & & u \geq 0 & & \text{entspricht (2) aus System für (P)} \\ (2) \rightsquigarrow & \quad Ay - b \leq 0, & & y \geq 0 & & \text{entspricht (1) aus System für (P) mit } y = x \\ (3) \rightsquigarrow & \quad y^\top \left(A^\top u + c\right) = 0, & & u^\top (Ay - b) = 0 & & \text{entspricht (3) aus System für (P)} \end{aligned}$$

(D) liefert also dasselbe System (1) - (3) wie (P).

Satz 3.10 (schwache Dualität) Sei x zulässig für (P) und u zulässig für (D). Dann gilt

$$-b^\top u \leq c^\top x$$

Beweis. Es gilt

$$\begin{aligned} -b^\top u &\leq (-Ax)^\top u = -x^\top A^\top u && (Ax \leq b, u \geq 0) \\ &\leq x^\top c = c^\top x && (A^\top u \geq -c, x \geq 0) \end{aligned}$$

□

Satz 3.11 (starke Dualität) Die Optimierungsaufgabe (P) ist genau dann lösbar, wenn (D) lösbar ist. Für die zugehörigen Lösungen $\overline{x}$ und $\overline{u}$ gilt dann

$$-b^\top \overline{u} = c^\top \overline{x}$$

also die Gleichheit der Optimalwerte.

Beweis. Der erste Teil der Aussage folgt direkt aus der Gleichheit der KKT-Systeme. Aus Eigenschaft (3) des KKT-Systems folgt dann die Gleichheit der Optimalwerte mittels

$$\overline{u}^\top (A\overline{x} - b) = 0 = \overline{x}^\top \left(A^\top \overline{u} + c\right) \quad \Rightarrow \quad -b^\top \overline{u} = \overline{u}^\top A\overline{x} = c^\top \overline{x}$$ □

Aus dem schwachen Dualitätssatz folgt insbesondere auch, dass die Existenz eines dual (primal) zulässigen Punktes eine endliche untere (obere) Schranke für den primalen (dualen) Optimalwert liefert.

Folgerung 3.12

$$\text{(P) lösbar} \quad \Leftrightarrow \quad \text{(D) lösbar} \quad \Leftrightarrow \quad \exists\, x \geq 0, u \geq 0 \colon Ax \leq b, A^\top u \geq -c$$

Die Bedingungen (3) im KKT-System werden **Komplementaritätsbedingungen** genannt.

zum Beispiel: $u^\top (Ax - b) = 0, x \geq 0, u \geq 0, Ax - b \leq 0$, d.h. $u_i(Ax - b)_i = 0$ für alle i.

Es ist möglich, primale und duale Aufgabe gleichzeitig innerhalb eines Tableaus zu lösen:

$$\begin{array}{llllll} (P) & z_P & = c^\top x \to \min & \text{bei} & Ax & \leq b, x \geq 0 \\ (D) & z_D & = -b^\top u \to \max & \text{bei} & -A^\top u & \leq c, u \geq 0 \end{array}$$

Durch Einführen von Schlupfvariablen $s \geq 0$, $v \geq 0$ erhält man

$$\begin{array}{llllll} (P) & z_P & = c^\top x \to \min & \text{bei} & s = b - Ax, \ x \geq 0 & s \geq 0 \\ (D) & -z_D & = b^\top u \to \min & \text{bei} & v = c + A^\top u & u \geq 0, v \geq 0 \end{array}$$

T_0	x	1
$s=$	$-A$	b
$z_P=$	$c^\top$	0

bzw.

T_0	u	1
$v=$	$A^\top$	c
$-z_D=$	$b^\top$	0

Beide Schemata sind (gewissermaßen) zueinander transponiert. Das duale Simplexverfahren für (P) kann als primales Simplexverfahrens für (D) interpretiert werden.

3.5 Transportoptimierung

3.5.1 Problemstellung

Zur Erinnerung: Es gebe Erzeuger $i \in I = \{0, \dots, r\}$ und Verbraucher $k \in K = \{1, \dots, s\}$. Weiterhin seien die Kosten c_{ik} für den Transport einer Einheit von i nach k sowie der Vorrat $a_i > 0$ und der Bedarf $b_k > 0$ für alle $i \in I$ und $k \in K$ bekannt. Wie ist der gesamte Transport kostenminimal zu gestalten.

Als Variablen verwenden wir die Transportmenge x_{ik} von i nach k.

$$z = \sum_{i\in I}\sum_{k\in K} c_{ik}x_{ik} \to \min \quad \text{bei} \quad \begin{aligned} \sum_{k\in K} x_{ik} &= a_i \quad (i \in I) \\ \sum_{i\in I} x_{ik} &= b_k \quad (k \in K) \\ x_{ik} &\geq 0 \quad (i,k) \in I \times K \end{aligned} \tag{3.12}$$

Mit

$$\begin{aligned} x &= (x_{11}, x_{12}, \dots, x_{1s}, x_{21}, \dots, x_{rs})^\top \\ c &= (c_{11}, c_{12}, \dots, c_{1s}, c_{21}, \dots, c_{rs})^\top \\ \overline{b} &= (a_1, \dots, a_r, b_1, \dots, b_s)^\top \end{aligned}$$

hat (3.12) die Form

$$z = c^\top x \to \min \text{ bei } Ax = \overline{b}, x \geq 0$$

Bemerkung 3.6 Das Transportproblem ist eine sehr spezielle Optimierungsaufgabe. Dei Koeffizientenmatrix

$$A = \left(\begin{array}{cccc|cccc|cccc} 1 & 1 & \cdots & 1 & & & & & & & & \\ & & & & 1 & 1 & \cdots & 1 & & & & \\ & & & & & & & & 1 & 1 & \cdots & 1 \\ \hline 1 & & & & 1 & & & & 1 & & & \\ & \ddots & & & & \ddots & & & & \ddots & & \\ & & & 1 & & & & 1 & & & & 1 \end{array}\right) \in \mathbb{R}^{(r+s)\times(r\cdot s)}$$

ist schwach besetzt. Insbesondere hat die Spalte von A, die zur Variablen x_{ik} gehört, die Gestalt $A^{ik} = \binom{e^i}{e^k} \in \mathbb{R}^{r+s}$.

Satz 3.13 Das Transportproblem ist genau dann lösbar, wenn die Sättigungsbedingung

$$\sum_{i\in I} a_i = \sum_{k\in K} b_k \tag{3.13}$$

gilt.

Beweis. Wir zeigen zuerst, dass (3.13) äquivalent zu $G \neq \emptyset$ ist.

- Einerseits folgt aus $x \in G \neq \emptyset$ durch Summation der Gleichungsnebenbedingungen

$$\sum_{i\in I} a_i = \sum_{i\in I}\sum_{k\in K} x_{ik} = \sum_{k\in K}\sum_{i\in I} x_{ik} = \sum_{k\in K} b_k$$

- Gilt hingegen (3.13), so ist mit $\sigma := \sum_{i\in I} a_i = \sum_{k\in K} b_k$ ein zulässiger Punkt $x = (x_{ik})$ wie folgt definiert:

$$x_{ik} := \frac{a_i b_k}{\sigma}$$

Der zulässige Bereich G ist polyedrisch (und damit abgeschlossen) und ferner wegen $0 \leq x_{ik} \leq \min\{a_i, b_k\}$ beschränkt und somit kompakt. Mit dem Satz von Weierstraß[1] folgt dann die Lösbarkeit des Transportproblems. □

Die Systemmatrix A besitzt für praxisrelevante Problemgrößen eine sehr große Anzahl an Einträgen, sodass die Anwendung des Simplexverfahrens im Allgemeinen nicht empfehlenswert ist; insbesondere deshalb, weil dieses die Struktur von A nicht mit einbezieht.

Zur Lösung des Transportproblems hat sich daher ein Verfahren etabiliert, das auch die duale Aufgabe

$$w := a^\top u + b^\top v \to \max \text{ bei } A^\top \binom{u}{v} \leq c, u \in \mathbb{R}^r, v \in \mathbb{R}^s \tag{3.14}$$

bzw.

$$w := \sum_{i\in I} a_i u_i + \sum_{k\in K} b_k v_k \to \max \text{ bei } u_i + v_k \leq c_{ik} \quad (u_i, v_k \in \mathbb{R}, i \in I, k \in K) \tag{3.15}$$

Satz 3.14 (Optimalitätskriterium) Sei $x \in G$, d.h. x ist zulässiger Transportplan, dann gilt

$$x \text{ optimal} \;\Leftrightarrow\; \exists u \in \mathbb{R}^r, v \in \mathbb{R}^s \text{ mit } u_i + v_k \leq c_{ik}, x_{ik}\cdot(c_{ik} - u_i - v_k) = 0 \quad \forall i \in I, k \in K$$

Beweis. Nach dem Charakterisierungssatz gilt: $x \in G$ ist genau dann optimal, wenn duale Variablen $u \in \mathbb{R}^r$ und $v \in \mathbb{R}^s$ existieren, sodass (u, v) dual zulässig ist und die Komplementaritätsbedingungen gelten. □

[1] Die Zielfunktion ist linear, d.h. stetig, auf einer komapkten Menge G.

Aussage 3.15 Der Rang von A ist $|I| + |K| - 1 = r + s - 1$.

Beweis. Einerseits sind die Spalten $A^{11}, \ldots, A^{1s}, A^{21}, A^{31}, \ldots, A^{r1}$ von A linear unabhängig, d.h. $\mathrm{rg}(A) \geq r + s - 1$. Andererseits ist die Summe der ersten r Zeilen identisch mit der Summe der letzten s Zeilen. Somit ist der Rang von oben beschränkt mit $\mathrm{rg}(A) \leq r + s - 1$. □

Folgerung 3.16 Jede Ecke des zulässigen Bereichs G hat höchstens $r + s - 1$ positive Komponenten.

Definition 3.4 Eine Folge von Zellen (Indexpaaren) $(i_1, k_1), (i_2, k_1), (i_2, k_2) \ldots (i_\ell, k_\ell), (i_1, k_\ell), (i_1, k_1)$ mit $i_\nu \neq i_\mu$, $k_\nu \neq k_\mu$ für $\nu \neq \mu$ heißt **Zyklus** (der Länge 2ℓ).

Beispiel 3.5 Im folgenden Schema ist ein Zyklus der Länge $2\ell = 8$ abgebildet:

i/k	1	2	3	4	5
1	*		*		
2		*			*
3	*				*
4		*	*		

Dieses Beispiel spielt eine wichtige Rolle bei der Feststellung, ob ein gegebener Transportplan eine Ecke von G ist.

Aussage 3.17
(i) Sei J eine Menge von Zellen. Gilt $|J| \geq r + s$, so enthält J mindestens einen Zyklus.
(ii) Sei $x = (x_{ik})$ ein zulässiger Transportplan. x ist genau dann eine Ecke von G, wenn $J_+ := \{(i, k) : x_{ik} > 0\}$ keinen Zyklus enthält.

Beweis. vielleicht in der Übung — oder auch nicht. □

3.5.2 Erzeugung eines ersten Transportplans

Dieser Teil entspricht der ersten Phase des Simplexverfahrens, d.h. also der Bestimmung einer Startecke. Gemäß der vorherigen Beobachtungen genügt es einen zyklenfreien zulässigen Transportplan zu finden. Hierfür können unterschiedliche Methoden genutzt werden.

Nordwest-Ecken-Regel: Die jeweilige noch nicht belegte Nordwest-Zelle wird mit maximaler Transportmenge belegt.

Regel der minimalen Kosten: In jedem Schritt wird eine noch nicht belegte Zelle, die minimale Kosten hat, mit maximaler Transportmenge belegt.

Methode von Vogel: Bestimme in jeder Zeile und Spalte die Differenz der zwei kleinsten Kostenkoeffizienten der noch freien Zellen. Wähle dann eine Zeile/Spalte mit maximaler Differenz und belege die Zelle mit kleinsten Kosten.

Darstellung der Inputdaten oder zulässigen Punkte in folgenden Schemata:

C	b_1	b_2	$\dots$	b_s
a_1	c_{11}	c_{12}	$\dots$	c_{1s}
$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$
a_r	c_{r1}	c_{r2}	$\dots$	c_{rs}

Tabelle 3.1: Inputdaten

X	b_1	b_2	$\dots$	b_s
a_1	x_{11}	x_{12}	$\dots$	x_{1s}
$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$
a_r	x_{r1}	x_{r2}	$\dots$	x_{rs}

Tabelle 3.2: Transportplan

Beispiel 3.6 Gegeben Sei das folgende Transportproblem:

C	12	5	6	7	7
4	12	6	10	9	5
19	10	16	17	3	7
14	4	11	5	8	10

Man erhält folgende Startecken:

X_{NW}	~~12~~ ~~8~~ 0	~~5~~ 0	~~6~~ 0	~~7~~ 0	~~7~~ 9
0	4	0	0	0	0
0 ~~6~~ ~~11~~ ~~19~~	8	5	6	0	0
0 ~~7~~ ~~14~~	0	0	0	7	7

X_{NW}	~~12~~ ~~8~~ 0	~~5~~ 0	~~6~~ 0	~~7~~ 0	~~7~~ 9
0	4	0	0	0	0
0 ~~6~~ ~~11~~ ~~19~~	8	5	6	0	0
0 ~~7~~ ~~14~~	0	0	0	7	7

Zielfunktionswert: $z(x_{\mathrm{NW}}) = 4 \cdot 12 + 8 \cdot 10 + 5 \cdot 16 + 6 \cdot 17 + 7 \cdot 8 + 7 \cdot 10 = 436$

X_{NW}	12	5	6	~~7~~ 0	~~7~~ 3
0 ~~4~~	0	0	0	0	$\boxed{4}^{3}$
0 ~~12~~ ~~19~~	0	$\boxed{5}$	$\boxed{4}$	$\boxed{7}^{1}$	$\boxed{3}$
0 ~~2~~ ~~14~~	$\boxed{12}^{2}$	0	$\boxed{2}^{4}$	0	0

Zielfunktionswert: $z(x_{\mathrm{MK}}) = 268$

X	0	5	6	7	7
4		4			
19		1	4	7	7
14	12		2		

Zielfunktionswert: $z(x_{\mathrm{V}}) = 236$

Je nach Qualität der Startlösung können unterschiedlich viele Iterationen des Transportalgorithmus vonnöten sein.

3.5.3 Der Transportalgorithmus

Ausgehend von einer Startlösung berechnet der Algorithmus zunächst ein Paar (u, v) dualer Variablen und prüft dann die Optimalität mit Satz 3.14. Liegt keine Optimalität vor, wird ein neuer Plan erzeugt.

Vorgehensweise:

(1) Bestimme einen zulässigen, zyklenfreien Transportplan X_0 mit genau $r+s-1$ markierten Basiszellen. (Diese bilden dann die Menge $J_B = J_B(X_0)$.)

(2) Bestimme für den aktuellen Plan X die zugehörigen dualen Variablen u_i und v_k aus dem Gleichungssystem

$$u_i + v_k = c_{ik} \qquad (i,k) \in J_B = J_B(X) \tag{3.16}$$

(3) Berechne für alle $(i,k) \notin J_B$ die Koeffizienten $w_{ik} = c_{ik} - u_i - v_k$. Falls $w_{ik} \geq 0$ für alle Zellen ist, dann ist X optimal. Andernfalls wähle man eine Zelle (p,q) mit $w_{pq} = \min\{w_{ik} : i \in I, k \in K\} < 0$.

(4) Markiere (p,q) im Schema von X, bestimme den (eindeutigen) Zyklus J_{pq} in $J_B \cup \{(p,q)\}$ und markiere abwechselnd die Zellen in J_{pq} mit "+" und "-". Sei J_{pq}^- die Menge der mit "-" gekennzeichneten Zellen.

(5) Ermittle $\delta = x_{gh} := \min\left\{x_{ik} : (i,k) \in J_{pq}^-\right\}$ und aktualisiere den Plan X gemäß

$$X^{\text{neu}} := (x_{ik}^{\text{neu}}) \quad \text{mit} \quad x_{ik}^{\text{neu}} := \begin{cases} x_{ik} + \delta & \text{falls } (i,k) \in (J_{pq} \cup \{(g,h)\}) \setminus J_{pq}^- \\ x_{ik} - \delta & \text{falls } (i,k) \in J_{pq}^- \\ x_{ik} & \text{sonst} \end{cases}$$

Aktualisere die Menge der Basiszellen

$$J_B^{\text{neu}} := (J_B \cup \{(p,q)\}) \setminus \{(g,h)\}$$

und gehe zu Schritt 2.

Bemerkung 3.7 Aufgrund von $|J_B| = r+s-1$ ist das in Schritt 2 zu lösende Gleichungssystem (zur Ermittlung von u und v) unterbestimmt. Eine der Variablen kann also beliebig festgelegt werden. Die in Schritt 3 bestimmten Werte w_{ik} sind jedoch unabhängig von dieser Wahl.

Im gesamten Algorithmus gilt stets $w_{ik} = 0$ für die aktuellen Basiszellen $(i,k) \in J_B$ (per Konstruktion in Schritt 2), diese beeinflussen den Optimalitätstest in Schritt 3 also nicht. Daher kann zur Darstellung das folgende komprimierte Schema genutzt werden:

T	v_1	v_2	$\dots$	v_s
u_1	$\boxed{x_{11}}$	w_{12}	$\dots$	w_{1s}
u_2	w_{21}	w_{22}	$\boxed{x_{23}}$	w_{2s}
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
u_r	w_{r1}	$\boxed{x_{r2}}$	$\dots$	w_{rs}

Beispiel 3.7 (Fortsetzung von Beispiel 3.6) Gegeben Sei das Problem

C	12	5	6	7	7
4	12	6	10	9	5
19	10	16	17	3	7
14	4	11	5	8	10

Mit der Minimale-Kosten-Regel haben wir bereits den Plan X_0 bestimmt:

X_0	12	5	6	7	7
4	0	0	0	0	$\boxed{4}$
19	0	$\boxed{5}$	$\boxed{4}$	$\boxed{7}$	$\boxed{3}$
14	$\boxed{12}$	0	$\boxed{2}$	0	0

Bestimmung der dualen Variablen (Potenziale) u_i und v_k

T_0	$v_1 = 16$	$v_2 = 16$	$v_3 = 17$	$v_4 = 3$	$v_5 = 7$
$u_1 = -2$	-2	-8	-5	8	$\boxed{4}$
$u_2 = 0$	-6	$\boxed{5}$	$\boxed{4}$	$\boxed{7}$	$\boxed{3}$
$u_3 =$ -12	$\boxed{12}$	7	$\boxed{2}$	17	15

$\boxed{5}$: $\underbrace{u_2}_{=0} + v_2 = c_{22} = 16 \;\Rightarrow\; v_2 = 16$

-12 : $u_3 + \underbrace{v_3}_{=17} = c_{33} = 5 \;\Rightarrow\; u_3 = c_{33} - 17 = -12$

Dieser Plan ist nicht optimal, da negative Einträge w_{ik} existieren. Wir wählen also den eindeutig bestimmten Zyklus $-8 \to 4 \to 3 \to 5 \to 8$ und geben alternierende "Vorzeichen", d.h. $-8^+ \to 4^- \to 3^+ \to 5^- \to 8^+$. Wir wählen das kleinste mit einem "$-$" markierte Zellenelement des Zyklus: $\delta = \min\left\{x_{ik} : (i,k) \in J_{pq}^-\right\} = 4$.

Ein neuer Plan ergibt sich nun mit

X_1					
		$\boxed{4}$			0
		$\boxed{1}$	$\boxed{4}$	$\boxed{7}$	$\boxed{7}$
	$\boxed{12}$		$\boxed{2}$		

Achtung: $x_{21} = \boxed{4}$ war vorher nicht in der Basis (also $x_{21} = 0$) und somit $x_{21}^{\text{neu}} = x_{21} + \delta = 4$.

Bestimmung der Potenziale u_i und v_k für X_1:

T_1	$v_1 = 16$	$v_2 = 16$	$v_3 = 17$	$v_4 = 3$	$v_5 = 7$
$u_1 = -10$	6	$\boxed{4}$	4	16	8
$u_2 = 0$	-6	$\boxed{1}$	$\boxed{4}$	$\boxed{7}$	$\boxed{7}$
$u_3 = -12$	$\boxed{12}$	7	$\boxed{2}$	17	15

Auch dieses Tableau ist noch nicht optimal. Wir erkennen den Zyklus $-6^+ \to 12^- \to 2^+ \to 4^-$.

Ein neuer Plan ergibt sich zu

X_2					
		$\boxed{4}$			
	$\boxed{4}$	$\boxed{1}$		$\boxed{7}$	$\boxed{7}$
	$\boxed{8}$		$\boxed{6}$		

mit den Potenzialen

T_2	$v_1 = 10$	$v_2 = 16$	$v_3 = 11$	$v_4 = 3$	$v_5 = 7$
$u_1 = -10$	12	$\boxed{4}$	9	16	8
$u_2 = 0$	$\boxed{4}$	$\boxed{1}$	6	$\boxed{7}$	$\boxed{7}$
$u_3 = -6$	$\boxed{8}$	1	$\boxed{6}$	11	9

Alle $w_{ik} \geq 0$, d.h. der das Tableau ist optimal und X_2 ist eine Lösung der gegebenen Optimierungsaufgabe. Es gilt $z(X_2) = 212$.

Kapitel 4
DISKRETE OPTIMIERUNG

In diesem Kapitel befassen wir uns mit Techniken zur Lösung ganzzahliger Optimierungsaufgaben. Dabei dürfen einige oder gar alle Variablen diskrete Werte annehmen. Somit entfällt eine Argumentation über Ableitung, zulässige Richtungen etc. Ganzzahlige Optimierungsaufgaben sind also "schwieriger" als die zugehörige stetige Relaxation. Dennoch kann, in einigen Fällen, das Lösen diskreter Aufgaben auch zur effizienten Lösung stetiger Aufgaben beitragen, wie folgendes Beispiel verdeutlicht.

4.1 Spaltengenerierung

Wir betrachten die stetige Relaxation des Bin-Packing-Problems (vgl. Abschnitt 1.3.2). Zur Erinnerung: Es sind b_i Teile der Länge ℓ_i $(i = 1, \dots, m)$ in möglichst wenige Behälter der Kapazität L zu packen.

- Packungsvarianten: $a^j = \left(a_1^j, \dots, a_m^j\right)^\top \in \mathbb{Z}_+^m$ mit $\ell^\top a^j \leq L$ $(j \in J)$
- Variablen: x_j beschreibt Häufigkeit, wie oft Variante a^j genutzt wird.

$$z = \sum_{j \in J} x_j \to \min \quad \text{bei} \quad \sum_{j \in J} a_i^j \cdot x_j = b_i \quad (i \in I) \quad \text{und} \quad x_j \geq 0 \quad (j \in J)$$

Grundsätzlich ist diese Aufgabe mit dem Simplexverfahren lösbar, jedoch gibt es im Allgemeinen exponentiell viele Variablen, sodass pro Austauschschritt ein großer Aufwand entstünde.

Wir können uns hierbei zu Nutze machen, dass alle Spalten der Systemmatrix A eine gemeinsame Struktur aufweisen:

$$a^j \text{ ist Spalte von } A \;\Leftrightarrow\; a^j \in \mathbb{Z}_+^m \text{ und } \ell^\top a^j \leq L$$

Offenbar gilt hie $c = e = (1, \dots, 1)^\top$, sodass für eine gewählte Basismatrix A_B in Schritt 2 des Simplexalgorithmus folgendes zu bestimmen wäre:

$$\overline{q} := \min_{j \in J_N} q_j \quad \text{mit} \quad q_j = c_j - d^\top a^j = 1 - d^\top a^j, \quad a^j \in \mathbb{Z}_+^m, \ell^\top a^j \leq L$$

wobei $d^\top := c_B^\top A_B^{-1}$. In Schritt 2 wäre folglich die Aufgabe

$$1 - d^\top a^j = q_j \to \min \text{ bei } \ell^\top a^j \leq L \text{ und } a^j \in \mathbb{Z}_+^m$$

bzw.

$$d^\top a^j \to \max \text{ bei } \ell^\top a^j \leq L, a^j \in \mathbb{Z}_+^m$$

zu lösen. Gilt $q_j^* < 0$, so liegt keine Optimalität vor und eine zugehörige Lösung $a^{j,*}$ wäre in die Basismatrix aufzunehmen. Gilt $g_j^* \geq 0$, so sind wir fertig.

4.2 Die Methode Branch & Bound

Branch & Bound (B&B) ist eine sehr flexible Technik, um exakte Lösungsverfahren für Probleme der diskreten Optimierung zu entwickeln. Anschaulich betrachtet wird dabei eine schwierige Optimierungsaufgabe sukzessiv in Teilprobleme zerlegt, die wiederum "leicht" (näherungsweise) gelöst werden können und somit zur Lösung des Gesamtproblems beitragen. Näherungslösungen erhält man dabei oftmals mithilfe geeigneter Relaxationen.

4.2.1 Grundlagen

Wir betrachten das Anfangsproblem

$$f(x) \to \min \text{ bei } x \in E \cap D \tag{P_0}$$

und eine zugehörige Relaxation

$$g(x) \to \min \text{ bei } x \in E \tag{Q}$$

wobei $g(x) \leq f(x)$ auf $D \cap E$ gilt.

Prinzip der B&B-Methode

Die Menge E wird durch Separation in Teilmengen E_i mit $i \in I$ zerlegt. Dadurch entstehen **Teilprobleme**

$$f(x) \to \min \text{ bei } x \in D \cap E_i \tag{P_i}$$

Jedem dieser Teilprobleme (P_i) soll nun eine Zahl $b(P_i)$, genannt **untere Schranke**, zugeordnet werden, sodass gilt

(a) $b(P_i) \leq \min\{f(x) : x \in D \cap E_i\}$
(b) $b(P_i) = f(\widehat{x})$ falls $D \cap E = \{\widehat{x}\}$
(c) $b(P_i) \leq b(P_j)$ falls $E_j \subset E_i$

Eine geeignete Möglichkeit besteht darin, z.B. die stetige Relaxation der Teilprobleme (P_i) zu betrachten, d.h.

$$b(P_i) := \begin{cases} \min\{g(x) : x \in E_i\} & \text{falls } |E_i \cap D| > 1 \\ f(\widehat{x}) & \text{falls } |E_i \cap D| = 1 \\ +\infty & \text{falls } E_i \cap D = \emptyset \end{cases}$$

4.2.2 Allgemeiner B&B-Algorithmus

Bezeichne mit R die Menge der noch zu bearbeitenden Teilprobleme ("Restmenge") und mit $\overline{z}$ den Zielfunktionswert der bisher besten gefundenen zulässigen Lösung $\overline{x} \in D \cap E$.

Schritt 0: **Initialisierung** — Bestimme $b(P_0)$.

(a) Falls $\overline{x} \in D \cap E$ bekannt ist mit $f(\overline{x}) = b(P_0)$, dann `STOP`.
(b) Setze $R := \{P_0\}$ und $\overline{z} := +\infty$ oder $\overline{z} = f(x)$, wenn ein $x \in D \cap E$ bekannt ist.

Schritt 1: **Abbruchtest** — Falls $R \neq \emptyset$, dann `STOP`. Falls $\overline{z} = +\infty$, dann ist (P_0) nicht lösbar (leerer zulässiger Bereich), andernfalls ist $\overline{x}$ Lösung von (P_0)

Schritt 2: **Strategie** — Wähle entsprechend einer Auswahlstrategie ein $P_i \in R$ und setze $R := R \setminus \{P_i\}$.

Schritt 3: **Zerlegung ("branch")** — Zerlege P_i durch Separation in endlich viele Teilprobleme $P_{i,1}, \ldots, P_{i,k_i}$. Setze $j := 1$.

Schritt 4: **Schranken- und Dominanztests ("bound")**

(a) Berechne $b(P_{i,j})$. Falls dabei ein $\widetilde{x} \in D \cap E$ gefunden wurde mit $f(\widetilde{x}) < \overline{z}$, setze $\overline{x} := \widetilde{x}$ und $\overline{z} := f(\widetilde{x})$.
(b) Falls $b(P_{i,j}) < \overline{z}$, dann setze $R := R \cup \{P_{i,j}\}$. Falls $j < k_i$, setze $j := j + 1$ und gehe zu (a).
(c) Setzte $R := R \setminus \{P_k\}$ für alle $P_k \in R$ mit $b(P_k) \geq \overline{z}$.

Gehe zu Schritt 1.

Bemerkung (1) Die Endlichkeit des Verfahrens ist zu sichern, z.B. durch $|E_{i,j} \cap D| \leq |E_i \cap D|$ für alle j (falls $E_i \cap D$ endlich ist) oder durch $b(P_{i,j}) > b(P_i) + \varepsilon$ mit $\varepsilon > 0$ für alle j und i.

(2) Das B&B-Verfahren kann mithilfe eines Verzweigungsbaumes veranschaulicht werden.

(3) In Schritt 2 können verschiedene Auswahlstrategien gewählt werden, z.B.

- **Minimalsuche** (best bound search): wähle $P_i \in R$ mit $b(P_i) \leq b(P_k)$ für alle $P_k \in R$.
- **Tiefensuche** (depth-first search, LIFO): wähle $P_i \in R$ mit kleinstem Schrankenwert unter allen Teilproblemen mit maximaler Verzweigungstiefe.
- **Breitensuche** (breadth-first-search, FIFO): wähle $P_i \in R$ mit kleinstem Schrankenwert unter allen Teilproblemen mit minimaler Verzweigungstiefe.

4.2.3 Beispiele für B&B-Verfahren

Das 0/1-Rucksackproblem

Zur Erinnerung: Gegeben seien ganze Zahlen $c_i > 0$, $0 < a_i < b$ für $i \in I := \{1, \ldots, n\}$

$$f(x) = c^\top x \to \max \text{ bei } a^\top x \leq b, \;\; x \in \{0,1\}^n \tag{P}$$
$$f(x) = c^\top x \to \max \text{ bei } a^\top x \leq b, \;\; c \in [0,1]^n \tag{Q}$$

Hier gilt also $E = \left\{x \in \mathbb{R}^n : a^\top x \leq b, 0 \leq x \leq 1, i \in I\right\}$ und $D = \mathbb{B}^n = \{0,1\}^n$.

Bemerkung 4.1 Die stetige Relaxation (Q) besitzt unter der Voraussetzung

$$\frac{c_i}{a_i} \geq \frac{c_{i+1}}{a_{i+1}} \quad \text{für } 1 \leq i \leq n$$

die Lösung

$$\widehat{x}_i = 1 \quad (i = 1, \ldots, k) \qquad \widehat{x}_{k+1} = \frac{b - \sum_{i=1}^{k} a_i}{a_{k+1}} \qquad \widehat{x}_i = 0 \;\; i = k+2, \ldots, n \tag{R}$$

mit $k := \max\left\{j \in I : \sum_{i=1}^{j} a_i \leq b\right\}$.

Bemerkung. Wir sortieren also abfallend nach Nutzen pro Volumen und packen dann soviel wie möglich in den Rucksack (k Elemente). Dann füllen wir den Restplatz noch mit (einem Anteil von) dem nächsten Element auf ($k+1$-tes Element), alles danach können wir nicht mehr mitnehmen.

Innerhalb des Verzweigungsbaums ergeben sich die folgenden Teilprobleme (für bereits fixierte Variablen $\overline{x}_i$, $i \in I_k \subseteq I$).

$$\sum_{i \in I_k} c_i \overline{x}_i + \sum_{i \notin I_k} c_i x_i \to \max \text{ bei } \sum_{i \notin I_k} a_i x_i \leq b - \sum_{i \in I_k} a_i \overline{x}_i, \quad i \notin I_k, x_i \in \mathbb{B} \tag{$P_k(\overline{x})$}$$
$$\sum_{i \in I_k} c_i \overline{x}_i + \sum_{i \notin I_k} c_i x_i \to \max \text{ bei } \sum_{i \notin I_k} a_i x_i \leq b - \sum_{i \in I_k} a_i \overline{x}_i, \quad i \notin I_k, x_i \in [0,1] \tag{$Q_k(\overline{x})$}$$

Beachte: Da es sich um eine Maximierungsaufgabe handelt, werden im B&B-Algorithmus *obere* Schranken $b(P_k(\overline{x}))$ benötigt. Diese gewinnen wir aus den Optimalwerten der stetigen Relaxation $Q_k(\overline{x})$.

Beispiel 4.1 Gegeben sei das binäre Rucksackproblem

$$\begin{aligned} z = c^\top x =& 8x_1 + 16x_2 + 20x_3 + 12x_4 + 6x_5 + 10x_6 + 4x_7 \to \max \\ \text{bei } a^\top x =& 3x_1 + 7x_2 + 9x_3 + 6x_4 + 3x_5 + 5x_6 + 2x_7 \leq 17 = b, \quad x_i \in \{0,1\}, i = 1, \ldots, 7 \end{aligned} \tag{P}$$

- Die korrekte Sortierung liegt bereits vor.

- Wurzelknoten (P)= (P_0): Die stetige Relaxation besitzt die Lösung $x_1 = x_2 = 1$ und $x_3 = \frac{7}{9}$ sowie $x_4 = \cdots = x_7 = 0$. Somit ist $b(P_0) = \lfloor 8 + 16 + 20 \cdot \frac{7}{9} \rfloor = 39$. Die Abrundung ist dabei erlaubt, da $c^\top x \in \mathbb{Z}$ für alle zulässigen Punkte von (P). Ein zulässiger Punkt für (P) ist gegeben durch $\overline{x}_1 = \overline{x}_2 = \overline{x}_4 = 1$, $\overline{x}_i = 0$ sonst. Dabei ist $z(\overline{x}) = 8 + 16 + 12 = 36 =: \overline{z}$. Wegen $36 < 39$ muss weiter verzweigt werden.

- Verzweigung: $x_3 = 0$ vs. $x_3 = 1$.

  ▷ Teilproblem: $(P_1) = (P_{0,1})$. Setzt man $x_3 = 0$, so hat die stetige Relaxation die Lösung $x_1 = x_2 = x_4 = 1$, $x_5 = \frac{1}{3}$ und $x_6 = x_7 = 0$. Somit ist $b(P_1) = \lfloor 8 + 16 + 12 + 6 \cdot \frac{1}{3} = 38$. Ein daraus ableitbarer Punkt ist gegeben durch $\overline{x}_1 = \overline{x}_2 = \overline{x}_4 = 1$ mit $z = 36$, d.h. $\overline{z}$ muss nicht aktualisiert werden.

  ▷ $(P_2) = (P_{0,2})$. Setzt man $x_3 = 1$, so erhält man $x_1 = 1$, $x_2 = \frac{5}{7}$ mit $b = \lfloor 8{+}20{+}16 \cdot \frac{5}{7} \rfloor = 39$ und einen zulässigen Punkt für (P) durch $\overline{x}_1 = \overline{x}_3 = \overline{x}_5 = \overline{x}_7 = 1$ mit $\overline{z} = 38$, d.h. $\overline{z}$ wurde verbessert.

  ▷ Damit kann (P_1) abgeschlossen werden, da maximal noch der Zielfunktionswert 38 möglich ist, der in (P_2) bereits erreicht worden ist.

- Verzweigung: $x_2 = 0$ vs. $x_2 = 1$.

  ▷ Teilproblem $(P_3) = (P_{2,1})$. Setzt man $x_2 = 0$ (und $x_3 = 1$ von oben), dann hat die stetige Relaxation die Lösung $x_1 = 1$, $x_4 = \frac{5}{6}$ mit $b(P_3) = 38 = \overline{z}$, also sind wir hier fertig.

  ▷ Teilproblem $(P_4) = (P_{2,2})$. Setzt man $x_2 = 1$ (und $x_3 = 1$) so folgt $x_1 = \frac{1}{3}$, also $b(P_4) = 38 = \overline{z}$, d.h. wir sind hier wieder fertig.

- Nun ist $R = \emptyset$ und wir haben eine Lösung gefunden:

$$x_1 = x_3 = x_5 = x_7 = 1 \quad \text{mit} \quad z = 38$$

Ganzzahlige lineare Optimierung nach Land/Doig/Dahin

Wir betrachten die ganzzahlige Optimierungsaufgabe

$$c^\top x \to \min \text{ bei } x \in D \cap E \tag{P}$$

mit $D = \mathbb{Z}^n$ und $E = E_0 = \{x \in \mathbb{R}^n : Ax = b, x \geq 0\}$, wobei alle Inputdaten (A, b, c) ganzzahlig sind. Die im Verlauf des Verfahrens zu betrachtenden Teilprobleme (P_i) haben die Form

$$c^\top x \to \min \text{ bei } x \in D \cap E_i \tag{P_i}$$

wobei E_i durch eine oder mehrere zusätzliche Ungleichungen aus E_0 entsteht. Sei x^{LP} eine Lösung der zu (P_i) gehörenden stetigen Relaxation

$$z(Q_i) = \min \left\{ c^\top x : x \in E_i \right\} \tag{Q_i}$$

mit $x_j^{\mathrm{LP}} \notin \mathbb{Z}$ für mindestens einen Index j. Dann kann der (gerundete) Optimalwert als Schranke $b(P_i)$ genutzt werden.

$$E_{i,1} = \left\{x \in E_i : x_j \leq \left\lfloor x^{\mathrm{LP}} \right\rfloor\right\} \tag{P$_{i,1}$}$$
$$E_{i,2} = \left\{x \in E_i : x_j \geq \left\lfloor x^{\mathrm{LP}} \right\rfloor + 1 = \left\lceil x^{\mathrm{LP}} \right\rceil\right\} \tag{P$_{i,2}$}$$

Bemerkung 4.2 Das Runden des Optimalwertes einer Relaxation zur Schrankenbestimmung ist nur für $c \in \mathbb{Z}^n$ zulässig.

Beispiel 4.2 Betrachte

$$z = -7x_1 - 2x_2 \to \min \text{ bei } -x_1+2x_2+x_4 = 4$$
$$5x_1+ x_2+x_4 = 20, \quad x_1, \ldots, x_n \in \mathbb{Z}_+$$

Die Relaxation ergibt sich durch $x_1, \ldots, x_n \geq 0$.

- Wurzelknoten:

T_1	x_1	x_2	1
$x_3 =$	1	-2	4
$x_4 =$	-5	-1	20
$z =$	-7	-2	0

$\xrightarrow{x_1 \leftrightarrow x_4}$ $\cdots$ $\xrightarrow{x_2 \leftrightarrow x_3}$

T_3	x_4	x_3	1
$x_2 =$	$-\frac{1}{11}$	$-\frac{5}{11}$	$\frac{40}{11}$
$x_1 =$	$\frac{-2}{11}$	$\frac{1}{11}$	$\frac{26}{11}$
$z =$	$\frac{16}{11}$	$\frac{3}{11}$	$-\frac{332}{11}$

- Verzweigung: nach x_2, da $\min\left\{4 - \frac{40}{11}, \frac{40}{11} - 3\right\} > \min\left\{4 - \frac{36}{11}, \frac{36}{11} - 3\right\}$.

  ▷ 1. Teilproblem: $x_2 \geq 4$ (führe $s_2 \geq 0$ ein)

$$\Rightarrow \quad s_2 = x_2 - 4 \overset{\text{aus T}_3}{=} \left(-\frac{1}{11}x_4 - \frac{5}{11}x_3 + \frac{40}{11}\right) - 4$$
$$= -\frac{1}{11}x_4 - \frac{5}{11}x_3 - \frac{4}{11} < 0$$

  Damit ist $x_2 \geq 4$ nicht möglich (leerer zulässiger Bereich) und dieser Fall muss nicht betrachtet werden.

  ▷ 2. Teilproblem: $x_2 \leq 3$ (führe $s_2 \geq 0$ ein)

$$s_2 = 3 - x_2 = 3 - (-\frac{1}{11}x_4 - \frac{5}{11}x_3 + \frac{40}{11})$$
$$= \frac{1}{11}x_4 + \frac{5}{11}x_3 - \frac{7}{11}$$

  Füge dies ist T_3 ein:

T_3'	x_4	x_3	1
$x_2 =$	$-\frac{1}{11}$	$-\frac{5}{11}$	$\frac{40}{11}$
$x_1 =$	$\frac{-2}{11}$	$\frac{1}{11}$	$\frac{26}{11}$
$s_2 =$	$\frac{1}{11}$	$\boxed{\frac{5}{11}}$	$-\frac{7}{11}$
$z =$	$\frac{16}{11}$	$\frac{3}{11}$	$-\frac{332}{11}$
Keller	$-\frac{1}{5}$	$*$	$\frac{7}{5}$

$\xrightarrow{s_2 \leftrightarrow x_3}$

T_4	x_4	s_2	1
$x_2 =$	0	-1	3
$x_1 =$	$-\frac{1}{5}$	$\frac{1}{5}$	$\frac{17}{5}$
$x_3 =$	$-\frac{1}{5}$	$\frac{11}{5}$	$\frac{7}{5}$
$z =$	$\frac{7}{5}$	$\frac{3}{5}$	$-\frac{149}{5}$

- Verzweigung: x_1 (x_3 ebenso möglich)

  ▷ 3. Teilproblem: $x_1 \geq 4 \;\Rightarrow\; s_1 = x_1 - 4 = -\frac{1}{5}x_4 + \frac{1}{5}s_2 - \frac{3}{5}$

T_4'	x_4	s_2	1
$x_2 =$	0	-1	3
$x_1 =$	$-\frac{1}{5}$	$\frac{1}{5}$	$\frac{17}{5}$
$x_3 =$	$-\frac{1}{5}$	$\frac{11}{5}$	$\frac{7}{5}$
$s_1 =$	$-\frac{1}{5}$	$\boxed{\frac{1}{5}}$	$\frac{3}{5}$
$z =$	$\frac{7}{5}$	$\frac{3}{5}$	$-\frac{149}{5}$

$\xrightarrow{s_1 \leftrightarrow s_2}$

T_5	x_4	s_1	1
$x_2 =$			0
$x_1 =$			4
$x_3 =$			3
$s_2 =$			8
$z =$	2	3	-28

  Damit haben wir zumindest *eine* ganzzahlige Lösung $z = -28$. Wir wissen jedoch noch nicht, ob es tatsächlich die optimale Lösung ist.

  ▷ 4. Teilproblem: $x_1 \leq 3 \;\Rightarrow\; s_1 = 3 - x_1 = \frac{1}{5}x_4 - \frac{1}{5}s_2 - \frac{2}{5}$. Dies fügt man zu T_4 hinzu. Ein dualer Simplexschritt führt dann zu

T_6	s_1	s_2	1
$x_2 =$			3
$x_1 =$			3
$x_3 =$			2
$x_3 =$			1
$z =$	2	3	-27

  Dabei haben wir also eine ganzzahlige Lösung mit $z = -27$.

Insgesamt wissen wir also $x^* = (4, 0, 3, 0)$ mit $z^* = -28$ aus dem dritten Teilproblem.

Das Rundreiseproblem (Traveling Salesman Problem, TSP)

Gegeben seien n Orte, eine Kostenmatrix $C = (c_{ik}) \in \mathbb{R}^{n \times n}$, wobei c_{ik} die Entfernung (Zeit, Kosten, ...) von i nach k beschreibt.
Annahme: $c_{ii} = +\infty$ für alle $i \in I = \{1, \ldots, n\}$. In einigen Anwendungen ist C auch symmetrisch.

Variablen: $x_{ik} \in \{0, 1\}$, $(i, k) \in I \times I$ mit $x_{ik} = 1 \;\Leftrightarrow\;$ man reist von i nach k

Optimierungsproblem:

$$z = \sum_{i \in I} \sum_{k \in I} c_{ik} x_{ik} \to \min \tag{4.1}$$

$$\text{bei} \sum_{i \in I} x_{ik} = 1 \quad (k \in I) \tag{4.2}$$

$$\sum_{k \in I} x_{ik} = 1 \quad (i \in I) \tag{4.3}$$

$$x_{ik} \in \{0, 1\} \quad (i, k) \in I \times I \tag{4.4}$$

$$\sum_{i,k} x_{ik} \in S \leq |S| - 1 \quad (S \subset I, 0 < |S|) \tag{4.5}$$

Die Bedingung (4.5) wird auch **Subtoureliminationsbedingung** (SEB) genannt. Die Bedingungen (4.1) bis (4.4) modellieren ein spezielles Transportproblem (das **Zuordnungsproblem**). Für das Problem (4.1) – (4.5) kann man auf folgende Weise Schranken erhalten:

- Lösung des (ganzzahligen) Zuordnungsproblems
- stetige Relaxation von (4.1) – (4.5): löse zunächst das Zuordnungsproblem. Falls dabei Subtouren entstehen, verbiete man diese mit geeigneten Bedingungen vom Typ (4.5) und löse danach das um diese Bedingung erweiterte Zuordnungsproblem.
- Zeilen- und Spaltenreduktion: Ermittlung einer zulässigen Lösung (u, v) des dualen Problems (der stetigen Relaxation des Zuordnungsproblems) mithilfe der folgenden Idee:

$$\begin{aligned} u_i &:= \min\{c_{ik} : k \in I\} && (i \in I) \\ v_i &:= \min\{c_{ik} - u_i : i \in I\} && (k \in I) \\ \Rightarrow\ w_{ik} &= c_{ik} - u_i - v_k \geq 0 && \forall (i,k) \in I \times I \text{ (duale Zulässigkeit)} \end{aligned}$$

Somit gilt

$$\begin{aligned} z = \sum_{i \in I} \sum_{k \in I} c_{ik} x_{ik} &\overset{(\star)}{=} \sum_{i \in I} \sum_{k \in I} \underbrace{w_{ik}}_{\geq 0} \underbrace{x_{ik}}_{\geq 0} + \sum_{i \in I} u_i + \sum_{k \in I} v_k \\ &\geq \sum_{i \in I} u_i + \sum_{k \in I} v_k \qquad \text{(untere Schranke für } z\text{)} \end{aligned}$$

Im Schritt $(\star)$ haben wir dabei $w_{ik} = c_{ik} + u_i + v_i$ und die Bedingungen (4.2) sowie (4.3) verwendet.

Die oben genannten Schrankenwerte unterscheiden sich (mitunter stark) hinsichtlich des numerischen Aufwands und der Güte der erhaltenen Näherungen. In dieser Vorlesung betrachten wir die dritte Variante.

Beispiel 4.3 Wir betrachten die Kostenmatrix

$C = (c_{ik})$						u_i
	∞	32	$\underline{22}$	30	24	22
	10	∞	$\underline{3}$	18	∞	3
	∞	$\underline{9}$	∞	14	12	9
	16	10	7	∞	$\underline{6}$	6
	15	19	15	$\underline{12}$	∞	12
v_k	3	0	0	0	0	$b = 55$

Damit ist $b = \sum_i u_i + \sum_k v_k = 55$ eine untere Schranke für den Optimalwert.

Bemerkung 4.3 Die Schranke, die aus Zeilen- und anschließender Spaltenreduktion erhalten wird, weicht im Allgemeinen von der Schranke ab, die aus Spalten- und anschließender Zeilenreduktion gewonnen wird.

Unter Einbeziehung der reduzierten Kostenmatrix $D = (w_{ik})$ mit $w_{ik} = c_{ik} - u_i - v_i$ kann eine Verzweigungs- und Auswahlstrategie formuliert werden: Dabei betrachten wir die Elemente mit

$w_{ik} = 0$ und verzweigen gemäß $x_{ik} = 0$ vs. $x_{ik} = 1$.

- Die Belegung $x_{ik} = 0$ ist gleichbedeutend mit der Änderung des aktuellen Kostenwertes auf $+\infty$.
- Eingedenk[1] der Bedingungen (4.2) und (4.3) impliziert die Wahl $x_{ik} = 1$, dass $x_{jk} = 0$ für alle $j \neq i$ und $x_{i\ell} = 0$ für alle $\ell \neq k$ gelten muss. Für diese Indexpaare kann man die Kosten auf $+\infty$ erhöhen. Weiterhin muss die Subtour $i \to k \to i$ verhindert werden, d.h. es gilt $x_{ki} = 0$ und damit $w_{ki} = +\infty$. (Analog verfahre man gegebenenfalls mit längeren Subtouren.) Infolge dieser Änderung der Kostenmatrix können im Anschluss weitere Zeilen- und Spaltenreduktionen ermöglicht werden, die zu verbesserte Schranken (in den entsprechenden Teilproblemen) führen können.

Um wenigstens in einem der beiden Fälle ($x_{ik} = 0$, $x_{ik} = 1$) einen möglichst guten Anstieg der unteren Schranke zu erreichen, ist folgende Auswahlregel empfehlenswert. Für jedes (i, k) mit $w_{ik} = 0$ überlege man im Vorfeld bereits, welcher Zuwachs der Schranke im Fall $x_{ik} = 0$ (d.h. wenn $w_{ik} = \infty$ gesetzt wird) zu erwarten ist. Dieses Gewicht $\widetilde{w}_{ik}$ bestimmt sich zu

$$\widetilde{w}_{ik} := \min_{q \neq k} w_{iq} + \min_{p \neq i} w_{pk}$$

und kann an jeder Nullzelle vorab notiert werden. Für die Verzweigung ist dann eine solche Zelle zu wählen, deren Wert $\widetilde{w}_{ik}$ am größten ist.

Beispiel 4.4 (Fortsetzung) Die reduzierte Matrix aus Beispiel 4.3 ergibt sich zu

w_{ik}					
	∞	10	0^{2}	8	2
	4	∞	0^{4}	15	∞
	∞	0^{7}	∞	5	3
	7	4	1	∞	0^{3}
	0^{4}	7	3	0^{5}	∞

Die Hilfsgröße $\widetilde{w}_{ik}$ ist für $(i, k) = (3, 2)$ maximal. Die nächste Verzweigung wäre also $x_{32} = 0$ vs. $x_{32} = 1$.

$x_{32} = 0$ liefert

w_{ik}						u_i
	∞	10	0^{2}	8	2	0
	4	∞	0^{4}	15	∞	0
	∞	∞	∞	5	3	3
	7	4	1	∞	0^{3}	0
	0^{4}	7	3	0^{5}	∞	0
v_k	0	4	0	0	0	$b = 55 + 7 = 62$

$\xrightarrow{\text{Reduktion}}$

w_{ik}					
	∞	6	0	8	2
	4	∞	0	15	∞
	∞	∞	∞	2	0
	7	4	1	∞	0
	0	3	3	0	∞

$x_{32} = 1$ liefert

[1] Alternativer Vorschlag: im Lichte der Bedingungen ...

w_{ik}						u_i
	∞	∞	0	8	2	0
	4	∞	∞	15	∞	4
	∞	0	∞	∞	∞	0
	7	∞	1	∞	0	0
	0	∞	3	0	∞	0
v_k	0	0	0	0	0	$b = 55 + 4 = 59$

Reduktion $\longrightarrow$

w_{ik}					
	∞	∞	0	8	2
	0	∞	∞	11	∞
	∞	0	∞	∞	∞
	7	∞	1	∞	0
	0	∞	3	0	∞

Fazit:

- $z \geq 59$ für jede Rundreise mit $x_{32} = 1$
- $z \geq 62$ für jede Rundreise mit $x_{32} = 0$

Ein vollständiges Beispiel befindet sich auf dem Übungsblatt.

4.3 Dynamische Optimierung

Die Grundidee der dynamischen Optimierung besteht in der Dekomposition des Ausgangsproblems in eine Folge ähnlich strukturierter Teilprobleme, deren Optimalwerte im Rahmen einer Rekursionsvorschrift zur Lösung der Originialaufgabe genutzt werden können. Eine solche Zerlegung erscheint besonders dann effektiv und zielführend, wenn der Einfluss einer Variable (oder Variablengruppe) auf das Gesamtproblem (also Zielfunktion und Nebenbedingung) „unabhängig“ von den anderen Variablen (Variablengruppen) beschrieben werden kann ($\rightarrow$ Separabilität). Hier erfolgt die Betrachtung nur anhand zweier Beispiele.

4.3.1 Das Rucksackproblem in der dynamischen Optimierung

Wir betrachten das klassische Rucksackproblem

$$f(x) = c^\top x \rightarrow \max \text{ bei } a^\top x \leq b \text{ und } x \in \mathbb{Z}_+^n \tag{P}$$

mit ganzzahligen Eingabedaten c, a und b.

Für $k \in I := \{1, \ldots, n\}$ und $y \in \{0, 1, \ldots, b\}$ definieren wir nun ein Teilproblem von (P) mittels

$$F(k, y) := \max \left\{ \sum_{i=1}^{k} c_i x_i : \sum_{i=1}^{k} a_i x_i \leq y, x_i \in \mathbb{Z}_+, 1 \leq i \leq k \right\}$$

Dabei hat der Rucksack durch y eine verminderte Kapazität und es liegen durch k beschränkt weniger Teiletypen vor. Um (P) muss also $F(n, b)$ bestimmt werden. Zunächst beobachten wir

$$F(1, y) = c_1 \cdot \left\lfloor \frac{y}{a_1} \right\rfloor \quad \forall y \in 0, \ldots, b$$

Außerdem legen wir fest, dass $F(k, y) = 0$ falls $y < 0$. Aufgrund der Linearität der Zielfunktion

und der Nebenbedingung gilt für $k > 1$ der Zusammenhang

$$F(k,y) = \max\left\{c_k x_k + F(k-1, y - a_k x_k) : x_k = 0, 1, \dots \left\lfloor \frac{y}{a_k} \right\rfloor\right\}$$

bzw. äquivalent dazu (für $y \geq a_n$)

$$F(k,y) = \max\Big\{\underbrace{F(k-1,y)}_{\widehat{=}x_k=0}, \underbrace{c_k + F(k, y-a_n)}_{\widehat{=}x_k \geq 1}\Big\} \tag{4.6}$$

Der Optimalwert des Zustandes (k, y) wird auf Optimalwerte „kleinerer" Zustände zurückgeführt. Diese müssen also zur Berechnung von $F(k, y)$ bereits bekannt sein.

Algorithmus (Gilmore & Gomory) Schritt 1: Setze $F(1, y) = c_1 \cdot \left\lfloor \frac{y}{a_1} \right\rfloor$ für $y = 0, \dots, b$

Schritt 2: Für $k = 2, \dots, n$:

- für $y = 0, \dots a_k - 1$ setze $F(k, y) = F(k-1, y)$
- für $y = a_k, \dots, b$ setze $F(k, y) = \max\{F(k-1, y), F(k, y-a_k) + c_k\}$

Beispiel 4.5 Betrachte das Rucksackproblem

$$5x_1 + 10x_2 + 12x_3 + 6x_4 \to \max \text{ bei } 4x_1 + 7x_2 + 9x_3 + 5x_4 \leq 15, x_i \in \mathbb{Z}_+, i = 1, \dots, 4$$

Dieses Rucksackproblem lösen wir mithilfe einer Tabelle:

$k\backslash y$	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	0	0	0	0	5	5	5	5	10	10	10	10	15	15	15	15
2	0	0	0	0	5	5	5	$\boxed{10}$	10	10	10	15	15	15	20	20
3	0	0	0	0	5	5	5	10	10	12	12	15	15	15	20	20
4	0	0	0	0	5	6	6	10	10	12	12	15	16	17	20	20

$\boxed{} = \max\{F(1,7), F(2, 7-7) + 10\} = \max\{5, 10\} = 10$

Somit ist der Zielfunktionswert $F(n, b) = 20$. Einen zulässigen Punkt mit diesem Zielfunktionswert findet man durch Rückwärtsauswertung der Rekursion, z.B. ergibt die Auswertung

$$(4,15) \overset{x_4=0}{\longleftarrow} (3,15) \overset{x_3=0}{\longleftarrow} (2,15) \overset{x_2\geq 1}{\longleftarrow} (2,8) \overset{x_2=1}{\longleftarrow} (1,8) \overset{x_1\geq 1}{\longleftarrow} (1,4) \overset{x_1=2}{\longleftarrow} (1,0)$$

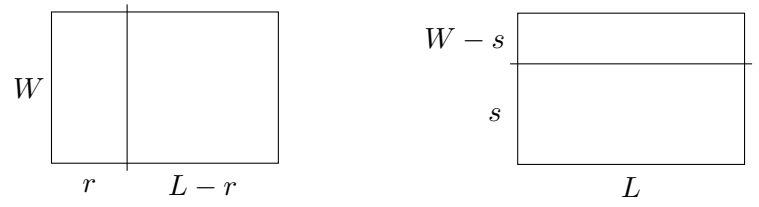
den Punkt $x^* = (2, 1, 0, 0)$. Diese Lösung ist nicht zwangsläufig eindeutig. Auch $x^* = (0, 2, 0, 0)$ ist hier eine Lösung.

4.3.2 2D-Guillotine-Zuschnitt

Dieses Thema ist informativ und wird nicht geprüft werden.

Aus einem Rechteck der Größe $L \times W$ sollen kleinere Rechtecke der Größe $\ell_i \times w_i$ mit $i \in$

$I = \{1, \ldots, m\}$ durch Guillotine-Schnitte so zugeschnitten werden, dass der Abfall minimal ist. Wir nehmen an, dass alle Inputdaten ganzzahlig sind und die gewünschten Teile auch mehrfach erhalten werden können.



Es bezeichnet $F(\ell, w)$ den Maximalwert der genutzten Fläche des (Teil-)Rechtecks $\ell \times w$, also

$$F(\ell, w) = \max \left\{ \sum_{i \in I} (\ell_i \cdot w_i) \cdot a_i : a = (a_1, \ldots, a_m)^\top \text{ ist eine zulässige } \begin{matrix} \text{Packungsvariante /} \\ \text{Zuschnittsvariante} \end{matrix} \right\}$$

Der Faktor $\ell_i \cdot w_i$ beschreibt dabei die Fläche von Objekt i. Offensichtlich ist $F(\ell, w) = 0$ falls $0 \leq \ell < \min\{\ell_i : i \in I\}$ oder $0 \leq w < \min\{w_i : i \in I\}$ gelten. Außerdem kann $F(\ell_i, w_i) = \ell_i \cdot w_i$ für $i \in I$ initialisiert werden. Darauf aufbauend lässt sich gemäß

$$F(\ell, w) := \max\{h(\ell, w), v(\ell, w)\}$$

eine Rekursion für alle noch verbleibenden Paare (ℓ, w) mit $0 \leq \ell \leq L$, $0 \leq w \leq W$ definieren. Dabei gilt

$$h(\ell, w) = \max_{r=1,\ldots\left\lfloor \frac{\ell}{2} \right\rfloor} \{F(r, w) + F(\ell - r, w)\}$$

$$v(\ell, w) = \max_{s=1,\ldots\left\lfloor \frac{w}{2} \right\rfloor} \{F(\ell, s) + F(\ell, w - s)\}$$

Der Optimalwert für ein großes Rechteck kann also aus den Optimalwerten kleinerer Rechtecke gewonnen werden.

4.4 Schnittebenenverfahren

Wir betrachten

$$c^\top x \to \min \text{ bei } Ax = b, x \in \mathbb{Z}_+^n \tag{P}$$

mit zumindest rationalen Daten A, b, c und die zugehörige stetige Relaxation

$$c^\top x \to \min \text{ bei } Ax = b, x \in \mathbb{R}_+^n \tag{Q}$$

Idee: Lösung der stetigen Aufgabe und anschließende Hinzunahme geeigneter Nebenbedingungen zur schrittweisen Konstruktion einer ganzzahligen Lösung

Nehmen wir also an, dass (Q) mit dem Simplex-Verfahren gelöst wurde. Dann existiert eine Partition $\{1, \ldots, n\} = B \cup N$, sodass (mit $P = -A_B^{-1} A_N$) das endgültige Tableau wie folgt

darstellbar ist:

$$\begin{aligned} x_B &= Px_N + p \quad (p \geq 0) \\ z &= q^\top x_n + q_0 \quad (q \geq 0) \end{aligned} \qquad \text{bzw.} \qquad \begin{aligned} x_B - Px_n &= p \\ z &= q^\top x_N + q_0 \end{aligned}$$

Die Lösung von (Q) lautet demnach $(x_B, x_N) = (p, 0)$. Falls $x_B = p$ bereits ganzzahlig ist, löst x auch (P). Andernfalls (für nicht ganzzahliges $x_B = p$) möchten wir eine (oder mehrere) Nebenbedingung(en) zu (Q) hinzufügen, sodass

- der Punkt x nicht mehr zulässig für (Q) ist
- jeder ganzzahlige Punkt des zulässigen Bereichs von (P) (bzw. (Q)) noch immer zulässig ist.

Dazu eignet sich folgender Ansatz: Es sei $p_i \notin \mathbb{Z}_+$ für $i \in I_B$. Wegen

$$[x_B]_i + \sum_{j \in N} (-P_{ij}) x_j = p_i \quad \forall x \in \mathbb{R}^n_+ \text{ mit } Ax = b$$

erhält man zunächst

$$\begin{aligned} & [x_B]_i + \sum_{j \in N} \lfloor -P_{ij} \rfloor \, x_j \leq p_i \quad \forall x \in \mathbb{R}^n_+ \text{ mit } Ax = b \\ \Rightarrow \quad & [x_B]_i + \sum_{j \in N} \lfloor -P_{ij} \rfloor \, x_j \leq \lfloor p_i \rfloor \quad \forall x \in \mathbb{Z}^n_+ \text{ mit } Ax = b \end{aligned}$$

Beide Beobachtungen zusammen ergeben

$$\underbrace{p_i - \lfloor p_i \rfloor}_{=:r_i} \leq \sum_{j \in N} (-P_{ij} - \lfloor -P_{ij} \rfloor) x_j = \sum_{j \in N} \underbrace{(\lceil P_{ij} \rceil - P_{ij})}_{=:r_{ij}} x_j \quad \forall x \in \mathbb{Z}^n_+ \text{ mit } Ax = b$$

Definition 4.1 Die Ungleichung $\sum_{j \in N} r_{ij} x_j \geq r_i$ heißt **Gomory-Schnitt** zur Zeile $i \in I_B$.

Aufgrund von $x_j = 0$ für alle $j \in N$ (für die aktuell betrachtete Lösung x von (Q)) und $p_i \in \mathbb{Z}_+$ erhält man

- $r_i = p_i - \lfloor p_i \rfloor > 0$
- $0 \overset{x_N = 0}{=} \sum_{j \in N} r_{ij} x_j \geq r_i > 0 \quad \Rightarrow \quad \lightning$

Das heißt also, dass die aktuelle (stetige) Lösung x vom zulässigen Bereich „abgetrennt“ wird.

Setzt man zulässig man zusätzlich $r_{ij} = 0$ für $j \in B$, so erhält man einen Koeffizientenvektor $r^{(1)} \in \mathbb{R}^n$ und es gilt

$$G_p := \{x \in \mathbb{Z}^n_+ : Ax = b\} = \left\{ x \in \mathbb{Z}^n_+ : Ax = b, \left(r^{(1)}\right)^\top x \geq r_i \right\}$$

Durch Gormory-Schnitte erhält man eine im Allgmeinen strengere Relaxation zu (P)

$$z = q^\top x_N + q_0 \to \min \text{ bei } x_B = Px_N + p, \quad s = \sum_{j \in N} r_{ij} x_j \underbrace{- r_i}_{<0}, \quad x_B, x_N, s \geq 0 \qquad (Q_1)$$

Die Nebenbedingung $s = \sum_{j \in N} r_{ij} x_j \underbrace{- r_i}_{<0}$ stellt dabei den Gomory-Schnitt mit Schlupfvariablen s dar.

Insgesamt ergibt sich damit das Gomory-Verfahren:

Schritt 0: Initialisierung: $k := 0$, $(Q_k) = (Q)$.

Schritt 1: Löse (Q_k), z.B. mit einem Simplex-Verfahren.

- Falls (Q_k) nicht zulässig ist (leerer zulässiger Bereich), dann ist auch (P) nicht zulässig. `STOP`
- Falls z für (Q_k) unbeschränkt ist, dann ist auch (P) nicht lösbar. `STOP`
- Falls Lösungen von (Q_k) ganzzahlig sind, dann sind diese auch Lösungen von (P). `STOP`

Schritt 2: Füge einen Gomory-Schnitt zu (Q_k) hinzu und erhalte (Q_{k+1}). Setze $k := k + 1$ und gehe zu Schritt 1.

Beispiel 4.6 Wir betrachten

$$x_1 \to \min \text{ bei } -4x_1 + 4x_2 \leq -1, 8x_1 + 16x_2 \leq 0, x_1, x_2 \in \mathbb{Z}_+ \qquad \text{(P)}$$

Die Einführung von Schlupfvariablen liefert

T_0	x_1	x_2	1
x_3	$\boxed{4}$	-4	-1
x_4	-8	-16	9
z	1	0	0
K	$*$	1	$\frac{1}{4}$

dual zulässig

T_1	x_3	x_2	1
x_1	$\frac{1}{4}$	1	$\frac{1}{4}$
x_4	-2	-24	7
z	$\frac{1}{4}$	1	$\frac{1}{4}$

optimales Tableau, aber nicht ganzzahlig

Es gilt $p_1 = \frac{1}{4} \notin \mathbb{Z}$, also erhält man den Gomory-Schnitt

$$\underbrace{\left(\left\lceil \frac{1}{3} \right\rceil - \frac{1}{4}\right)}_{=r_{13}} x_3 + \underbrace{(\lceil 1 \rceil - 1)}_{=r_{12}} x_2 \geq \underbrace{\frac{1}{4} - \left\lfloor \frac{1}{4} \right\rfloor}_{=r_1} \Leftrightarrow \frac{3}{4} x_3 \geq \frac{1}{4} \rightsquigarrow s_1 = \frac{3}{4} x_3 - \frac{1}{4} (s_1 \geq 0)$$

T_1	x_3	x_2	1
x_1	$\frac{1}{4}$	1	$\frac{1}{4}$
x_4	-2	-24	7
s_1	$\boxed{\frac{3}{4}}$	0	$-\frac{1}{4}$
z	$\frac{1}{4}$	1	$\frac{1}{4}$

T_2	s_1	x_2	1
x_1	$\frac{1}{3}$	1	$\frac{1}{3}$
x_4	$-\frac{8}{3}$	-24	$\frac{19}{3}$
x_3	$\frac{4}{3}$	0	$\frac{1}{3}$
z	$\frac{1}{3}$	1	$\frac{1}{3}$

Es gilt $p_1 = \frac{1}{3} \notin \mathbb{Z}$, also erhält man den Schnitt

$$\left(\left\lceil \frac{1}{3} \right\rceil - \frac{1}{3}\right) s_1 + (\lceil 1 \rceil - 1)\, x_2 \geq \frac{1}{3} - \left\lfloor \frac{1}{3} \right\rfloor \Leftrightarrow \frac{2}{3} s_1 \geq \frac{1}{3} \rightsquigarrow s_2 = \frac{2}{3} s_1 - \frac{1}{3} (s_2 \geq 0)$$

T_2	s_1	x_2	1
x_1	$\frac{1}{3}$	1	$\frac{1}{3}$
x_4	$-\frac{8}{3}$	-24	$\frac{19}{3}$
x_3	$\frac{4}{3}$	0	$\frac{1}{3}$
s_2	$\boxed{\frac{2}{3}}$	0	$-\frac{1}{3}$
z	$\frac{1}{3}$	1	$\frac{1}{3}$

$\longrightarrow$

T_3	s_2	x_2	1
x_1	$\frac{1}{2}$	1	$\frac{1}{2}$
x_4	-4	-24	5
x_3	2	0	1
s_1	$\frac{3}{2}$	0	$\frac{1}{2}$
z	$\frac{1}{2}$	1	$\frac{1}{2}$

Es gilt $p_1 = \frac{1}{2} \notin \mathbb{Z}$, also erhält man den Schnitt

$$\left(\left\lceil\frac{1}{2}\right\rceil - \frac{1}{2}\right) s_2 + (\lceil 1\rceil - 1)\, x_2 \geq \frac{1}{2} - \left\lfloor\frac{1}{2}\right\rfloor \;\Leftrightarrow\; \frac{1}{2}s_2 \geq \frac{1}{2} \rightsquigarrow s_3 = \frac{1}{2}s_2 - \frac{1}{2}(s_3 \geq 0)$$

T_3	s_2	x_2	1
x_1	$\frac{1}{2}$	1	$\frac{1}{2}$
x_4	-4	-24	5
x_3	2	0	1
s_1	$\frac{3}{2}$	0	$\frac{1}{2}$
s_3	$\boxed{\frac{1}{2}}$	0	$-\frac{1}{2}$
z	$\frac{1}{2}$	1	$\frac{1}{2}$

$\longrightarrow$

T_4	s_3	x_2	1
x_1			1
x_4			1
x_3			3
s_1			2
s_2			1
z	1	1	1

Die Lösung lautet also $x^* = (1, 0)$ mit $z^* = 1$.

Für eine grafische Visualisierung der Schnitte ist es (z.B. in Beispiel 4.6) notwendig, diese auf eine Form zu bringen, die nur die Originalvariablen (hier: x_1, x_2) enthält. Für den ersten Schnitt gilt

$$\frac{3}{4}x_3 \geq \frac{1}{4}$$

erhält man unter Verwendung von $x_3 = -1 + 4x_1 - 4x_2$ (aus dem Tableau T_0)

$$x_3 \geq \frac{1}{3} \;\Leftrightarrow\; -1 + 4x_1 - 4x_2 \geq \frac{1}{3} \;\Leftrightarrow\; x_1 - x_2 \geq \frac{1}{3}$$

Die Anzahl der Gomory-Schnitte ist im Allgemeinen exponentiell (zumindest bei willkürlicher Auswahl der Zeilen $i \in B$). Es gilt:

- Ist die Menge der zulässigen Punkte beschränkt, so liefert der Algorithmus nach endlich vielen Schritten eine ganzzahlige Lösung oder es existiert keine solche.
- Das lexikographische Gomory-Verfahren (d.h. immer Auswahl der Variable mit kleinstem Index) ist endlich.

Das Verfahren zur Lösung von (P) kann mitunter dadurch beschleunigt werden, dass

- „bessere“ (problemspezifische) Schnitte verwendet werden (z.B. Chvatal-Gomory-Schnitte, facettendefinierende Schnitte)
- Teilbarkeiten (bzw. Runden) geschickt genutzt werden

Beispiel 4.7 (Fortsetzung von 4.6) Wir betrachten erneut (P) aus Beispiel 4.6 und formen

dieses äquivalent um zu

$$x_1 \to \min \text{ bei } -x_1 + x_2 \leq -\frac{1}{4}, x_1 + 2x_2 \leq \frac{9}{8}, x_1, x_2 \in \mathbb{Z}_+$$

Da alle Terme auf der linken Seite der Nebenbedingung ganzzahlig sind, ist dies wiederum gleichwertig zu

$$x_1 \to \min \text{ bei } -x_1 + x_2 \leq -1, x_1 + 2x_2 \leq 1, x_1, x_2 \in \mathbb{Z}_+ \tag{P'}$$

Dann erhält man als (dual zulässiges) Starttableau

T_0	x_1	x_2	1
x_3	$\boxed{1}$	-1	-1
x_4	-1	-2	1
z	1	0	0
K	$*$	1	1

$\longrightarrow$

T_1	x_3	x_2	1
x_1	1	1	1
x_4	-1	-3	0
z	1	1	1

T_1 ist optimal und ganzzahlig, d.h. wir erhalten die gleiche Lösung $x^* = (1, 0)$ mit $z^* = 1$. Die Verbesserungen führen also hier dazu, dass kein einziger Gomory-Schnitt betrachtet werden muss.

Kapitel 5
OPTIMIERUNG AUF GRAPHEN

Die Bedeutung graphentheoretischer Modelle und Verfahren liegt in ihrer Anschaulichkeit und Praxisnähe. Zeitliche und logische Abläufe sowie Beziehungen zwischen Zuständen, Orten oder Systemkomponenten (allgemein **Knoten**) werden dabei durch **Bögen** oder **Kanten** direkt und zweckmäßig modelliert.

5.1 Grundbegriffe der Graphentheorie

siehe extra Blatt

5.2 Das Minimalgerüst-Problem / Minimum Spanning Tree Problem

Gegeben sei ein ungerichteter, zusammenhängender und gewichteter Graph $G = (V, E, c)$, wobei $c\colon E \to \mathbb{R}$ die Gewichtsfunktion ist.

Gesucht ist eine Teilmenge $T \subseteq E$ mit $|T| = |V| - 1$, sodass der induzierte Teilgraph $G_T = (V, T)$ kreisfrei ist.

Definition 5.1 Jede Teilmenge T mit diesen Eigenschaften wird **Gerüst** oder **Spannbaum** genannt.

Bemerkung 5.1 Der zu einem Spannbaum T gehörende Teilgraph G_T ist zusammenhängend.

Das Minimum-Spanning-Tree (MST) Problem besteht nun darin, einen Spannbaum zu finden, dessen Gesamtgewicht $\sum_{e \in T} c(e)$ kleinstmöglich ist. Es lässt sich mithilfe eines sogenannten **Greedy-Algorithmus** effizient lösen. Diese Algorithmen zeichnen sich vor allen dadurch aus, dass sie schrittweise denjenigen Folgezustand wählen, der „lokal“ (also zum Zeitpunkt der Auswahl) das beste Ergebnis verspricht. Derartige Algorithmen sind oft sehr schnell, lösen das Problem aber im Allgemeinen nicht optimal. Im Falle des MST-Problems führt jedoch das folgende Verfahren stets zur optimalen Lösung.

5.2.1 Algotihmus von KRUSKAL

Idee: Wähle in jeder Iteration eine Kante mit kleinstem Gewicht, wobei kein Kreis entstehen darf. Formal verfahren wir also wie folgt.

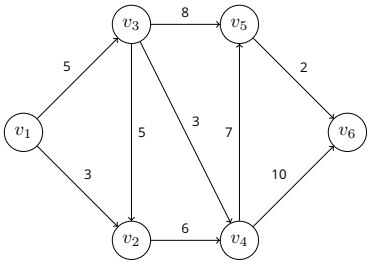
Algorithmus von Kruskal:

Schritt 0: Initialisierung: $T := \emptyset$

Schritt 1: Solange $|T| < n - 1$: ermittle eine Kante $e \in E$ mit kleinstem Gewicht $c(e)$, setze $E := E \setminus \{e\}$ und, falls $T \cup \{e\}$ kreisfrei ist, setze $T := T \cup \{e\}$.

Der Aufwand dieses Algorithmus wird maßgeblich von der Sortierung der Kantengewichte beeinflusst und lässt sich mit $\mathcal{O}(|E| \cdot \log |E|)$ abschätzen. Die Überprüfung der Kreisfreiheit ist bei geschickter Implementierung „schneller“ durchführbar.

Beispiel 5.1



Jeder minimale Spannbaum hat ein Gewicht von $c = 2 + 3 + 3 + 4 + 7 = 20$.

5.2.2 Algorithmus von Prim & Dijkstra

Eine alternative Vorgehensweise zur Gewinnung eines minimalen Spannbaums, bei der in jedem Schritt auf den Zusammenhang des Graphen G_T geachtet wird, kann wie folgt beschrieben werden.

Algorithmus von Prim & Dijkstra

Schritt 0: Initialisierung: $T := \emptyset$, $S := \{u\}$ für ein beliebiges $u \in V$. Für alle $v \in V \setminus \{u\}$ setze

- $h(v) := c(u, v)$, $p(v) := u$ falls $\{u, v\} \in E$
- $h(v) := +\infty$ sonst

Schritt 1: Solange $|T| < n - 1$: bestimme $v \in V \setminus S$ mit $h(v) = \min\{h(t) : t \in V \setminus S\}$ und setze $T := T \cup \{\{p(v), v\}\}$ sowie $S := S \cup \{v\}$. Für alle $t \in V \setminus S$ mit $\{v, t\} \in E$ und $c(v, t) < h(t)$ setze $h(t) := c(v, t)$, $p(t) := v$.

Dieser Algorithmus ist ebenfalls ein Greedy-Verfahren und besitzt (bei „gewöhnlicher“ Implementierung) eine Laufzeit von $\mathcal{O}(|V|^2)$, jedoch sind bei Verwendung spezieller Datenstrukturen auch bessere Abschätzungen möglich. Beide Algorithmen beginnen mit einer leeren Knotenmenge und fügen in jedem Schritt eine Kante mit minimalem „Gewicht“ (c bzw. h) hinzu. Sie unterscheiden sich vor allem darin, wie genau die Auswahl dieser Kante erfolgt und wie die Bildung eines Kreises vermieden wird.

5.3 Optimale Wege

5.3.1 Das Kürzeste-Wege-Problem

Gegeben sei ein gerichteter, gewichteter Graph $G = (V, E, c)$ mit $c(e) \geq 0$ für alle $(u, v) \in E$. Der Wert $c(e)$ kann zum Beispiel als Länge (oder allgemein „Kosten“) interpretiert werden. Es sind nun ausgehend von einem Startknoten v_1 ein kürzester Weg zu jedem anderen Knoten $v_k \in V$ zu finden.

Aussage 5.1 Es existiert eine Bogenmenge $E_w \subseteq E$ mit $|E_w| = |V| - 1$, die für jeden Knoten $v_k \neq v_1$ einen kürzesten Weg von v_1 zu v_k beinhaltet.

Beweis. Weniger als $|V| - 1$ Bögen würden einen Widerspruch zum Zusammenhang liefern. Angenommen es gäbe mehr als $|V| - 1$ Bögen. Dann existiert ein Knoten $v \neq v_1$ mit (Eingangs-) Grad $\delta^-(v) \geq 2$, d.h. es gibt zwei optimale Wege von v zu v_1. Nur einer dieser Bögen, die in v enden, ist erforderlich, um eine Bogenmenge gemäß Aufgabenstellung zu erhalten. □

Folgerung 5.2 E_w ist zusammenhängend und kreisfrei, also ein (gerichteter) Spannbaum.

Ein konkretes Verfahren zur Lösung dieses Problems liefert der Dijkstra-Algorithmus. Dabei werden, ausgehend von einem Startknoten $s = v_1$, bereits bekannte kürzeste Wege durch Hinzufügen weiterer Bögen/Kanten verlängert, um somit kürzeste Wege für bisher unerreichte Knoten zu finden.

Hierfür nutzen wir die Notation: Es sei

- M die Menge der Knoten, zu denen ein kürzester Weg bekannt ist
- $p(v_k) = p(k)$ der Vorgängerknoten von v_k auf dem kürzesten Weg zu v_k
- $d(v_k) = d(k)$ die Länge des (bisher) kürzesten Weges zu v_k

Schließlich erhalten wir den folgenden Algorithmus:

Algorithmus von Dijkstra:

Schritt 0: Initialisierung: Setze $M := \{s\}$, $d(s) = 0$ und initialisiere für $v \neq s$

$$p(v) := \begin{cases} s & (s,v) \in E \\ 0 & (s,v) \notin E \end{cases} \qquad d(v) := \begin{cases} c(s,v) & (s,v) \in E \\ +\infty & (s,v) \notin E \end{cases}$$

Schritt 1: Bestimme $u \notin M$ mit $d(u) = \min\{d(v) : v \notin M\}$. Falls $d(u) = +\infty$, dann `STOP`. Andernfalls setze $M := M \cup \{u\}$.

Schritt 2: Für alle $v \notin M$ mit $(u, v) \in E$: falls $d(v) > d(u) + c(u, v)$ (also ein kürzerer Weg ist gefunden), dann setze $d(v) = d(u) + c(u, v)$ und $p(v) = u$.

Schritt 3: Falls $M \neq V$, gehe zu Schritt 1. Sonst `STOP`.

Offenbar terminiert dieser Algorithmus nach endlich vielen Schritten, da in jeder Iteration entwe-

der ein Element zu M hinzugefügt wird oder der Algorithmus gänzlich abbricht (Schritt 1). Somit wird Schritt 1 höchstens $\mathcal{O}(n)$ mal ausgeführt und bewirkt dabei höchstens $\mathcal{O}(n)$ Durchläufe der in Schritt 2 benötigten Schleife. Bei naiver Implementierung besitzt dieser Algorithmus also eine Komplexität von $\mathcal{O}(|V|^2)$, es sind aber auch bessere Abschätzungen (z.B. $\mathcal{O}(|E| + |V| \cdot \log |V|)$) möglich.

Sei $d_G(s, v)$ sei die Länge eines kürzesten Weges von s nach v in $G = (V, E, c)$. Offenbar gilt $d(v) \geq d_G(s, v)$ für alle $v \in V$, da der Algorithmus zumindest irgendeinen Weg von s nach v findet oder feststellt, dass es keinen solchen gibt (also $d(v) = \infty$).

Aussage 5.3 Gilt $c(e) \geq 0$ für alle $e \in E$, so gibt $d(v)$ die Länge eines kürzesten Weges von s nach v an.

Beweis. Sobald ein Knoten $u \in V$ zu M hinzugefügt wird, ändert sich $d(u)$ nicht mehr. Es genügt daher den Zeitpunkt zu betrachten, an dem u zu M hinzugefügt wird.

(IA) Als erstes wird (in der Initialisierung) s in M aufgenommen und $d(s) = 0$ gesetzt. Wegen $c(e) \geq 0$ für alle $e \in E$ ist dies die Länge eines kürzesten Weges.

(IS) Sei nun $u \neq s$ der „zeitlich“ erste Knoten, für den bei Aufnahme in M die strikte Ungleichung $d(u) > d_G(s, u)$ gilt, d.h. bei u mach der Algorithmus den ersten Fehler. Sei γ ein tatsächlich kürzester Weg von s nach u. Dieser muss mindestens einen Knoten y enthalten, der (noch) nicht zur Menge M gehört. Wegen $s \in M$ gilt $y \neq s$. Wir zeigen nun, dass $d(y) = d_G(s, y)$ gilt. Offenbar gilt gemäß Induktionsvoraussetzung $d(x) = d_G(s, x)$ für alle Vorgänger $x \in M$ von y auf γ. Wenn ferner γ ein kürzester Weg von s nach u ist, dann enthält γ auch einen kürzesten Weg von s nach y. Sei $x \in M$ nun der direkte Vorgänger von y auf γ. Bei der Aufnahme von x in M wurde auch die Kante $(x, y) \in E$ geprüft und folglich $d(y)$ auf den korrekten Wert $d_G(s, y)$ gesetzt. Dann gilt

$$d(u) > d_G(s, u = d_G(s, y) + d_G(y, u) = d(y) + d_G(y, u) \quad \Rightarrow \quad d(u) > d(y)$$

Also hätte un aktuellen Schritt u nicht in M aufgenommen werde dürfen. Widerspruch. □

Beispiel 5.2 Graph

p =	0	v_1	v_1	v_1	v_4	v_4	v_4	~~v_5~~ v_7
	v_1	v_2	v_3	v_4	v_5	v_6	v_7	v_8
d =	$\boxed{0}$	3	$\underline{1}$	2	∞	∞	∞	∞
		3		$\underline{2}$				
		3			3	4	3	
					$\underline{3}$	4	3	
						4	$\underline{3}$	6
						$\underline{4}$		5

Ein minimaler Weg von v_1 nach v_8 lautet als

$$v_8 \leftarrow v_7 \leftarrow v_4 \leftarrow v_1$$

5.3.2 Das Längste-Wege-Problem

gegeben: gerichteter, gewichteter Graph $G = (V, E, c)$ mit beliebiger Funktion $c\colon E \to \mathbb{R}$

gesucht: längste Wege von $s \in V$ zu allen anderen Knoten

naive Idee: Dijkstra-Algorithmus für $G' = (V, E, -c)$ — funktioniert im Allgemeinen nicht, da $-c(e) \geq 0$ nicht erfüllt sein muss

Algorithmus von Ford/Moore:

Schritt 0: Wähle $s \in V$ als Startknoten und setze $d(s) = 0$, sowie

$$p(v) := \begin{cases} s & (s,v) \in E \\ 0 & \text{sonst} \end{cases} \qquad d(v) := \begin{cases} c(s,v) & (s,v) \in E \\ -\infty & \text{sonst} \end{cases}$$

für alle $s \neq v$. Definiere außerdem $A := \{s\} \cup \{v \in V : (s,v) \in E\}$, $B := \emptyset$ und $k := 1$.

Schritt 1: Falls $A = \emptyset$ oder $k = |V|$, dann `STOP`.

Schritt 2: Für alle $u \in A$ und alle $v \in V$ mit $(u,v) \in E$: falls $d(v) < d(u) + c(u,v)$, dann setze $d(v) = d(u) + c(u,v)$ und $p(v) = u$ sowie $B := B \cup \{v\}$.

Schritt 3: Setze $A = B$, $B = \emptyset$, $k := k + 1$ und gehe zu Schritt 1.

Nach Abschluss von Schritt 2 symbolisieren die Mengen A und B diejenigen Knoten, zu denen in der vorherigen bzw. aktuellen Iteration ein längerer Weg gefunden wurde. Das Verfahren von Ford/Moore besitzt eine Laufzeit von $\mathcal{O}(|V| \cdot |E|)$ und es kann gezeigt werden:

Aussage 5.4 Der Algorithmus von Ford/Moore ermittelt zu jedem Knoten, der von s erreichbar ist, einen längsten Weg, sofern er mit $A = \emptyset$ terminiert. Andernfalls enthält der Graph Kreise positiver Länge.

Da hierbei die Gewichte $c(e)$ auch negativ sein durften, kann eine modifizierte Variante des Algorithmus zum Auffinden kürzester Wege ich beliebigen Graphen genutzt werden.

5.3.3 Netzplantechnik

Längste Wege sind vor allem dann von Interesse, wenn die Wege eines Graphen eine logische Abfolge von Teilprojekten beschreiben, die erfüllt sein müssen, bevor ein neues Teilprojekt starten kann.

Formal:
- Teilprojekte TP_i mit Dauer D_i $(i = 1, \ldots, N)$
- Bedingungen an die Anfangszeiten AZ_i der Teilprojekte (**Koppelbedingungen**)

Zielstellung minimale Gesamtdauer des Projekts (und kritische Teilprojekte)

Wir betrachten zwei Fälle:

(1) $\mathrm{AZ}_j \geq \mathrm{AZ}_i + \tau_{ij}$ mit $\tau_{ij} \geq 0$: Teilprojekt TP_j darf frühestens τ_{ij} Zeiteinheiten nach Beginn von Teilprojekt TP_i starten

(2) $\mathrm{AZ}_j \leq \mathrm{AZ}_u + \tau_{ji}$ mit $\tau_{ji} \geq 0$: Teilprojekt TP_j muss spätestens τ_{ji} Zeiteinheiten nach Beginn von Teilprojekt TP_i starten

Modellierung als Graph $G = (V, E, c)$:

- $V = \{1, \ldots, N\}$ entsprechend Teilprojekten
- Füge Bögen (gerichtete Kanten) ein gemäß
  - ▷ $(i, j) \in E$ mit $c(i, j) = \tau_{ij}$ genau dann, wenn Fall (1) vorliegt
  - ▷ $(j, i) \in E$ mit $c(j, i) = \tau_{ji}$ genau dann, wenn Fall (2) vorliegt

Die Länge eines Weges von $a \in V$ nach $b \in V$ gibt nun an, wie viele Zeiteinheiten AZ_b mindestens nach AZ_a liegen muss. Die maximale Länge aller Wege in G von a nach b sei $\ell(a, b)$ und legt die frühestmögliche Startzeit von TP_b fest (relativ zur Startzeit von TP_a).

Ohne Einschränkung sei Knoten 1 der Projektbeginn mit $\mathrm{AZ}_1 = 0$. Ein Algorithmus, der für jedes TP die frühest- und spätestmögliche Anfangszeit FAZ_i bzw. SAZ_i bestimmt, kann wie folgt formuliert werden:

MPM-Algorithmus

Schritt 1: Ermittle FAZ_i für alle $i \in V$ gemäß $\mathrm{FAZ}_i = \ell(1, i)$. Definiere $T := \mathrm{FAZ}_N + D_N$ als minimale Projektdauer.

Schritt 2: Ermittle SAZ_i für alle $i \in V$ gemäß $\mathrm{SAZ}_i := \mathrm{FAZ}_N - \ell(i, N)$.

Die Bestimmung von $\ell(1, i)$ kann mit dem Ford-Moore-Algorithmus bewerkstelligt werden. Zur Bestimmung von $\ell(i, N)$ kann dieser Algorithmus ebenfalls genutzt werden, wobei dazu $\ell(N, i)$ in einem Graphen gesucht wird, der alle Bögen umorientiert (sodass N der feste Startknoten wird).

Definition 5.2 Die Zeit $\mathrm{PZ}_i := \mathrm{SAZ}_i - \mathrm{FAZ}_i$ heißt **Pufferzeit** von TP_i. Ein Teilprojekt i mit $\mathrm{PZ}_i = 0$ heißt **kritisch**.

Eine Verzögerung in einem kritischen Teilprojekt beeinflusst die gesamte Projektdauer.

Ein ausführliches Beispiel wird in der Übung besprochen.

5.4 Maximale Flüsse in Graphen

5.4.1 Problemstellung

Sei $G = (V, E, k)$ ein gerichteter Graph mit **Kapazitätsschranken** $k_e \geq 0$ für alle $e \in E$. Weiterhin sei $q \in V$ einzige **Quelle** und $s \in V$ einzige **Senke**. Ohne Einschränkung seien ale Inputdaten ganzzahlig. Für jedes $v \in V$ definieren wir

$$E^+(v) := \{e \in E : e = (v, p)\} \qquad E^-(v) := \{e \in E : e = (p, v)\}$$

der ausgehenden und eingehenden Bögen für v.

Definition 5.3 Eine Funktion $x\colon E \to \mathbb{R}$ heißt **Fluss**, wenn gilt

$$0 \leq x_e \leq k_e \quad (e \in E)$$
$$\sum_{e \in E^-(v)} x_e = \sum_{e \in E^+(v)} x_e \quad (v \in V \setminus \{q, s\})$$

Die zweite Bedingung (Flusserhaltung) ist auch als KIRCHHOFF'sches Gesetz bekannt.

Die Flusserhaltung impliziert

$$\sum_{e \in E^+(q)} x_e = \sum_{e \in E^-(1)} x_e =: f(x)$$

wobei $f(x)$ die Flussstärke angibt.

Beispiel 5.3 Dargestellt ist ein zulässiger Fluss der Stärke $f(x) = 5$. Die Bogenmarkierungen entsprechen dabei „x_e/k_e“. Der angegebene Fluss ist nicht optimal.

Ein geeignetes Hilfsmittel zur Prüfung der Optimalität ist der **Residualgraph** („Restgraph“). Ohne Einschränkung nehmen wir an, dass für alle $(u, v) \in E$ auch $(v, u) \in E$ gilt. Sofern dies nicht a priori erfüllt ist, ergänzen wir diese mit Kapazität $k_{(v,u)} = 0$.

Definition 5.4 Sei $x\colon E \to \mathbb{R}$ ein zulässiger Fluss, d.h. insbesondere ist $0 \leq x_e \leq k_e$ für alle $e \in E$. Für $e = (u, v) \in E$ definieren wir

$$r(e) = r(u, v) := k_e - x_e + x_{(v,u)}$$

das **Residuum** („Restkapazität“) von e. Weiterhin sei

$$E(x) := \{e \in E : r(e) > 0\}$$

die Menge aller Bögen mit positiven Residuum[a]. Der **Residualgraph** ist definiert als

$$G(x) := (V, E(x), r(e)_{e \in E(x)})$$

wobei $r(e)$ eine „aktualisierte Kapazität“ beschreibt.

[a] nur dort besteht die Möglichkeit der Flusserhöhung für den Gesamtgraphen

Beispiel 5.4 (Fortsetzung von 5.3) Aus dem oben angegebenen Fluss ergibt sich der folgende Residualgraph $G(x)$:

Offenbar gilt stets $r(e) \geq 0$ für alle $e \in E$. $G(x)$ enthält also all jene Bögen, die die Möglichkeit zur Flussvergrößerung modellieren, ergänzt um einige Rückwärtsbögen zur Verminderung von Flüssen auf bisher zu stark genutzten Verbindungen.

Eine mögliche Vorgehensweise zur Erzeugung eines maximalen Flusses ist der folgende Algorithmus:

Algorithmus von Ford-Fulkerson

Schritt 1: Initialisierung — Setze $x_e = 0$ für alle $e \in E$ (oder ein „besserer Startfluss“).

Schritt 2: Konstruiere den Residualgraphen $G(x)$. Wenn $G(x)$ keinen Weg von q nach s mit positiver Flussstärke enthält: `STOP` (Optimalität)

Schritt 3: Finde einen Fluss x' in $G(x)$, d.h. einen Weg von q nach s in $G(x)$ mit positiver Flussstärke f'. Aktualisiere x gemäßt $x_e := x_e + x'_e$ für alle $e \in E(x)$. Gehe zu Schritt 2.

Bei ganzzahligen Kapazitäten findet in jedem Durchlauf des Algorithmus eine Flussvergrößerung um mindestens eine Einheit statt (sofern er nicht in Schritt 2 abbricht). Damit terminiert das Verfahren nach endliche vielen Schritten. Diese Anzahl an Schritten kann jedoch proportional zum Optimalwert $f^* = f(x^*)$ sein. Darüber hinaus gibt es Beispiele mit irrationalen Eingabedaten, für die der obige Algorithmus nicht abbricht.

5.4.2 Algorithmus von Edmonds & Karp

Idee: grundsätzlich analog zu Ford/Fulkerson, aber im Schritt 3 wird jeweils ein Weg von s nach q mit kleinster Bogenzahl ermittelt

Es bezeichne $\delta_x(u, v)$ den Abstand zwischen $u \in V$ und $v \in V$ in $G(x)$, also die Anzahl der Bögen auf einem kürzesten Weg von u nach v.

Lemma 5.5 Sei x' ein Fluss, der aus dem Fluss x erhalten wird (durch einen flussvergrößernden Weg P in $G(x)$). Dann gilt

$$\delta_x(q, v) \leq \delta_{x'}(q, v) \quad \text{für alle } v \in V \setminus \{q\}$$

Beweis. Angenommen es gilt $\delta_x(q, v) > \delta_{x'}(q, v)$ für mindestens ein $v \neq q$. Ohne Einschränkung wählen wir dasjenige v mit minimalem Wert $\delta_{x'}(q, v)$. Also gilt

$$\delta_{x'}(q, u) < \delta_{x'}(q, v) \;\Rightarrow\; \delta_x(q, u) \leq \delta_{x'}(q, u) \tag{5.1}$$

weil v das „minimale Gegenbeispiel“ war. Sei P' ein kürzester Weg von q nach v in $G(x')$ und u der letzte Knoten vor v. Dann gilt $\delta_{x'}(q, u) = \delta_{x'}(q, v) - 1$ und folglich wegen (5.1)

$$\delta_x(q, u) \leq \delta_{x'}(q, u) \tag{5.2}$$

Betrachten wir $x(u, v)$ vor der Flussvergrößerung:

(a) $x(u,v) < k(u,v)$: Dann ist (u,v) in $G(x)$ vorhanden und es gilt

$$\delta_x(q,v) \leq \delta_x(q,u) + 1 \overset{(5.2)}{\leq} \delta_{x'}(q,u) + 1 = \delta_{x'}(q,v)$$

im Widerspruch zur Annahme.

(b) $x(u,v) = k(u,v)$: Dann ist (u,v) nicht in $G(x)$ enthalten, wohl aber (v,u). Da jedoch P' den Bogen (u,v) wieder nutzt, muss dieser in $G(x')$ vorkommen. Das geht nur dann, wenn der flussvergrößernde Weg P (der zum Fluss x gehörte) den Bogen (v,u) enthielt (um somit $x'(u,v) < k(u,v)$ und $(u,v) \in G'(x)$ zu bewirken). Man erhält

$$\delta_x(q,v) \leq \delta_x(q,u) - 1 \overset{(5.2)}{\leq} \delta_{x'}(q,u) < \delta_{x'}(q,v)$$

im Widerspruch zur Annahme. □

Der Algorithmus von Edmonds und Karp besitzt eine polynomielle Laufzeit für beliebige (nichtnegative) Kapazitäten.

Lemma 5.6 Der Algorithmus führt höchstens $\mathcal{O}(|V| \cdot |E|)$ Flussvergrößerungen durch.

Beweis. Jede Flussvergrößerung wird durch einen kritischen Bogen charakterisiert, d.h. auf jedem Weg P gibt es einen Bogen (u,v) mit $r_x(u,v) = \min\{r_x(e) : e \in P\}$, der die Erhöhung des Flusses am meisten beschränkt. Gemäß Konstruktion verschwindet ein kritischer Bogen im nächsten Schritt aus dem Restgraphen.

Frage: Wie oft kann ein Bogen (u,v) kritisch werden? Ein kritischer Bogen (u,v) kann in einem späteren Restgraphen wieder auftreten, wenn zuvor (v,u) auf einem flussvergrößernden Weg liegt. Sei x^0 ein Fluss, bei dem (u,v) kritisch war. Dann gilt $\delta_{x^0}(q,v) = \delta_{x^0}(q,u) + 1$. Ausgehend von x^0 betrachten wir nun die vom Algorithmus erzeugte Folge der Flüsse

$$x^0 \to x^2 \to \cdots \to x^t \to \cdots \to x^k \qquad (t \neq 0, t \neq k, k \geq 2)$$

wobei (u,v) in x^0 kritisch ist und das nächste Mal in x^k; außerdem wird (v,u) in x^t genutzt. Nun gilt

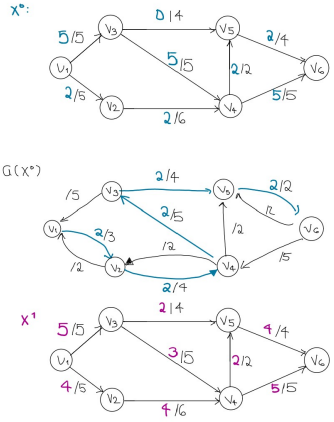
$$\delta_{x^k}(q,v) = \delta_{x^k}(q,u) + 1 \overset{\text{Lemma 5.5}}{\geq} \delta_{x^t}(q,u) + 1 = \delta_{x^t}(q,v) + 2 \overset{\text{Lemma 5.5}}{\geq} \delta_{x^0}(q,v) + 2$$

Also gilt: zwischen zwei Wegen, in denen (u,v) kritisch ist, wächst der Abstand zur Quelle um mindestens zwei Einheiten. Ein solcher Abstand kann höchsten $\mathcal{O}(|V|)$ Einheiten betragen. Demzufolge kann ein Bogen auch höchstens $\mathcal{O}(|V|)$ Mal kritisch werden. Insgesamt gibt es höchstens $\mathcal{O}(|V| \cdot |E|)$ Flussvergrößerungen. □

Dies impliziert:

Aussage 5.7 Der Algorithmus besitzt eine Komplexität von $\mathcal{O}(|V| \cdot |E|^2)$

Beispiel 5.5 Wir betrachten den folgenden Graphen



Flusstärke = 7