



Wintersemester 2017/2018

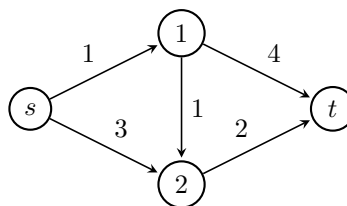
Lineare und kombinatorische Optimierung

Übungsblatt 11

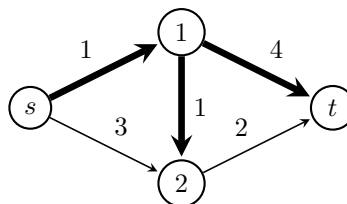
Gruppenübungen

Aufgabe 1. (Kürzeste Wege und der Simplex-Algorithmus)

(0 Punkte)



1. Formulieren Sie das Problem der Bestimmung eines kürzesten Weges von s nach t als lineares Optimierungsproblem in Standardform.
2. Lösen Sie das Problem mithilfe des Simplex-Algorithmus. Benutzen Sie die folgende Basislösung mit Zielfunktionswert 5 als Startlösung.



3. Lösen Sie das Problem mithilfe des Dijkstra-Algorithmus.
4. Vergleichen Sie die beiden letzten Aufgabenteile. Welche Optimalitätskriterien kommen jeweils zum Einsatz und wo sehen Sie Parallelen?

Lösung:

1. Um das Problem des kürzesten Weges als lineares Optimierungsproblem zu formulieren, müssen positive Flüsse und die Flusserhaltung sichergestellt werden. Man stelle sich ein Max-Flow-Problem vor, bei dem nur eine Einheit hindurchgeschickt wird. Diese soll minimale Kosten bezüglich der Kantengewichte generieren:

$$\begin{aligned} \min c^\top x \\ \sum_{e \in \delta^-(v)} x_e - \sum_{e \in \delta^+(v)} x_e &= \begin{cases} -1, & \text{falls } v = s \\ 1, & \text{falls } v = t \\ 0, & \text{sonst} \end{cases} \quad \forall v \in V \\ x_e &\geq 0 \quad \forall e \in E. \end{aligned}$$

Daraus ergibt sich für das obige Beispiel das folgende Problem:

$$\begin{aligned} \min \underbrace{(1 \quad 3 \quad 1 \quad 4 \quad 2)}_{=c} \cdot \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \end{pmatrix} \\ \underbrace{\begin{pmatrix} -1 & -1 & 0 & 0 & 0 \\ 1 & 0 & -1 & -1 & 0 \\ 0 & 1 & 1 & 0 & -1 \\ 0 & 0 & 0 & 1 & 1 \end{pmatrix}}_{=A} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \end{pmatrix} &= \underbrace{\begin{pmatrix} -1 \\ 0 \\ 0 \\ 1 \end{pmatrix}}_{=b} \\ x_1, x_2, x_3, x_4, x_5 &\geq 0. \end{aligned}$$

Dabei sind die Zeilen mit Knoten indiziert $(s, 1, 2, t)$ und die Spalten mit Kanten indiziert $((s, 1), (s, 2), (1, 2), (1, t), (2, t))$. Eigentlich müsste noch die Forderung $x_e \in \{0, 1\} \forall e \in E$ hinzugefügt werden, um gebrochene Lösungen zu verhindern. Dies ist allerdings überflüssig, da A total unimodular ist. Genauer zu ganzzahliger Optimierung finden Sie in den Vorlesungen Diskrete Optimierung I+II.

Auch ist darauf hinzuweisen, dass $-A$ der *Inzidenzmatrix* des gerichteten Graphen entspricht, weshalb auch ein Ansatz mit $\min\{c^\top x : -Ax = -b, x \geq 0\}$ möglich wäre. Dadurch hätte man in der ersten Iteration des Simplex-Algorithmus allerdings $\bar{y} = (-1, -2, -5)^\top$ erhalten und die nachfolgend aufgezeigten Parallelen zum Dijkstra-Algorithmus wären nicht so schön ersichtlich.

- Um ein Optimierungsproblem in Standardform zu erhalten, welches mit dem Simplex Algorithmus gelöst werden kann, muss eine zulässige Basis gefunden werden. Da in diesem Beispiel alle 4×4 -Matrizen linear abhängig sind, wird eine redundante Ungleichung gestrichen. In diesem Fall entscheiden wir uns für die erste Nebenbedingung. Sie sichert die Flusserhaltung im Knoten s . Durch die Flusserhaltung der anderen Knoten, ist dies aber sichergestellt:

$$\begin{aligned} \min (1 \quad 3 \quad 1 \quad 4 \quad 2) \cdot \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \end{pmatrix} \\ \begin{pmatrix} 1 & 0 & -1 & -1 & 0 \\ 0 & 1 & 1 & 0 & -1 \\ 0 & 0 & 0 & 1 & 1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \end{pmatrix} &= \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} \\ x_1, x_2, x_3, x_4, x_5 &\geq 0. \end{aligned}$$

It.1: Wir beginnen wie gefordert mit $x = (1, 0, 0, 1, 0)$, $B = (1, 3, 4)$ und $N = (2, 5)$.

BTRAN:

$$\bar{y}^\top A_B = c_B^\top \Rightarrow \bar{y}^\top \begin{pmatrix} 1 & -1 & -1 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} = (1 \quad 1 \quad 4) \Rightarrow \bar{y} = \begin{pmatrix} 1 \\ 2 \\ 5 \end{pmatrix}.$$

Pricing:

$$z_N = c_N - A_N^\top \bar{y} = \begin{pmatrix} 3 \\ 2 \end{pmatrix} - \begin{pmatrix} 0 & 1 & 0 \\ 0 & -1 & 1 \end{pmatrix} \begin{pmatrix} 1 \\ 2 \\ 5 \end{pmatrix} = \begin{pmatrix} 1 \\ -1 \end{pmatrix}.$$

Da $z_2 < 0$, wählen wir $N_2 = j = 5$ in die Basis.

FTRAN:

$$A_B w = A_{\cdot j} \Rightarrow \begin{pmatrix} 1 & -1 & -1 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} w = \begin{pmatrix} 0 \\ -1 \\ 1 \end{pmatrix} \Rightarrow w = \begin{pmatrix} 0 \\ -1 \\ 1 \end{pmatrix}.$$

Ratio-Test:

$$\gamma = x_{B_3}/w_3 = \min\{1/1\} = 1.$$

Also verlässt $B_3 = 4$ die Basis.

Update:

$$x_B = x_B - \gamma w = \begin{pmatrix} 1 \\ 0 \\ 1 \end{pmatrix} - 1 \begin{pmatrix} 0 \\ -1 \\ 1 \end{pmatrix} = \begin{pmatrix} 1 \\ 1 \\ 0 \end{pmatrix},$$

$$N = (2, 4),$$

$$B = (1, 3, 5),$$

$$x_5 = 1.$$

It.2: Neue Basislösung ist also $x_B = \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix}$,

BTRAN:

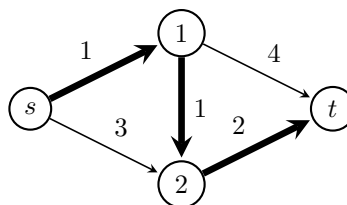
$$\bar{y}^\top A_B = c_B^\top \Rightarrow \bar{y}^\top \begin{pmatrix} 1 & -1 & 0 \\ 0 & 1 & -1 \\ 0 & 0 & 1 \end{pmatrix} = (1 \quad 1 \quad 2) \Rightarrow \bar{y} = \begin{pmatrix} 1 \\ 2 \\ 4 \end{pmatrix}.$$

Pricing:

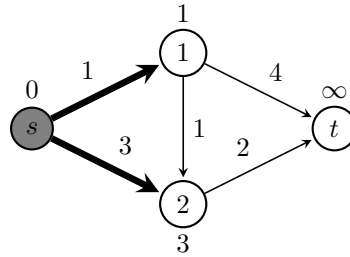
$$z_N = c_N - A_N^\top \bar{y} = \begin{pmatrix} 3 \\ 4 \end{pmatrix} - \begin{pmatrix} 0 & 1 & 0 \\ -1 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 \\ 2 \\ 4 \end{pmatrix} = \begin{pmatrix} 1 \\ 1 \end{pmatrix}.$$

Da $z_N \geq 0$: Lösung optimal.

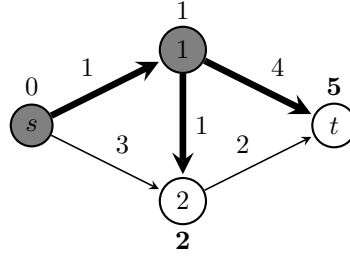
Damit ist $x^* = \begin{pmatrix} 1 \\ 0 \\ 1 \\ 0 \\ 1 \end{pmatrix}$ die Optimallösung mit ZF-Wert 4. Zeichnerisch ergibt sich:



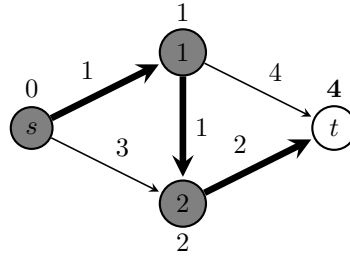
3. Dijkstra löst das Problem in vier Schritten. Grau hinterlegte Knoten sind bereits markiert. Abgeänderte Label sind fett geschrieben. Fett gezeichnete Kanten veranschaulichen die Vorgängerstruktur:



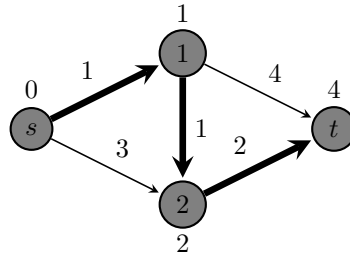
1. Schritt:



2. Schritt:



3. Schritt:



4. Schritt:

4. Für den Dijkstra-Algorithmus haben wir das Optimalitätskriterium aus Satz 5.6, also die Nichtnegativität von c_{ij}^d . Für den Simplex-Algorithmus muss die Nichtnegativität des Vektors der reduzierten Kosten z_N gelten. Man erkennt leicht die Gleichheit der beiden Kriterien. Beispielsweise erkennt der Dijkstra-Algorithmus vom 2. auf den 3. Schritt, dass die Kante $(2, t)$ aktualisiert werden muss, da

$$\begin{aligned} c_{2,t}^d &= c_{2,t} + d(2) - d(t) \\ &= 2 + 2 - 5 = -1 < 0. \end{aligned}$$

Der Simplex-Algorithmus rechnet im Pricing-Schritt analog

$$z_N = c_N - A_N^\top \bar{y} = \begin{pmatrix} 3 \\ 2 \end{pmatrix} - \begin{pmatrix} 0 & 1 & 0 \\ 0 & -1 & 1 \end{pmatrix} \begin{pmatrix} 1 \\ 2 \\ 5 \end{pmatrix} = \begin{pmatrix} 1 \\ -1 \end{pmatrix},$$

da die Labels d dem Vektor y aus **BTRAN** entsprechen:

$$\begin{aligned} d(1) &= \bar{y}_1 = 1, \\ d(2) &= \bar{y}_2 = 2, \\ d(t) &= \bar{y}_3 = 5. \end{aligned}$$

Die Bezeichnung *reduzierte Bogengewichte* für c_{ij}^d macht daher Sinn.

Aufgabe 2. (Modellierung)

(0 Punkte)

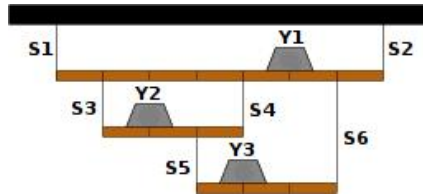


Abbildung 1: Hängegerüst

Gegeben sei das Hängegerüst wie in Abbildung 1. Die Seile S_1 und S_2 können je 300 kg Last, die Seile S_3 und S_4 je 100 kg und die Seile S_5 und S_6 jeweils 50 kg Last tragen. Das Gewicht der Lasten $y_i, i = 1, 2, 3$, verteilt sich jeweils linear auf die beiden angrenzenden Seile im Verhältnis zum Abstand der Last zum Seil. Die Lasten sollen nicht bewegt werden. Unter Vernachlässigung des Gewichts der Seile und der Bohlen soll das maximal zulässige Gesamtgewicht $y_1 + y_2 + y_3$ für die Lasten gefunden werden. Formulieren Sie dieses Problem als lineares Optimierungsproblem. (Das LP muss nicht gelöst werden.)

Lösung:

Die Last verteilt sich jeweils linear auf die beiden Seile im Verhältnis zum Abstand der Last zum Seil. Von unten beginnend ergeben sich die folgenden Belastungen und Beschränkungen für die Seile:

$$\text{Seil } S_6: \quad \frac{1}{3}y_3 \leq 50$$

$$\text{Seil } S_5: \quad \frac{2}{3}y_3 \leq 50$$

$$\text{Seil } S_4: \quad \frac{1}{3}y_2 + \frac{2}{3}\left(\frac{2}{3}y_3\right) \leq 100$$

$$\text{Seil } S_3: \quad \frac{2}{3}y_2 + \frac{1}{3}\left(\frac{2}{3}y_3\right) \leq 100$$

$$\text{Seil } S_2: \quad \frac{1}{7}\left(\frac{2}{3}y_2 + \frac{1}{3}\left(\frac{2}{3}y_3\right)\right) + \frac{4}{7}\left(\frac{1}{3}y_2 + \frac{2}{3}\left(\frac{2}{3}y_3\right)\right) + \frac{5}{7}y_1 + \frac{6}{7}\left(\frac{1}{3}y_3\right) \leq 300$$

$$\text{Seil } S_1: \quad \frac{6}{7}\left(\frac{2}{3}y_2 + \frac{1}{3}\left(\frac{2}{3}y_3\right)\right) + \frac{3}{7}\left(\frac{1}{3}y_2 + \frac{2}{3}\left(\frac{2}{3}y_3\right)\right) + \frac{2}{7}y_1 + \frac{1}{7}\left(\frac{1}{3}y_3\right) \leq 300$$

Damit erhalten wir folgendes LP:

$$\begin{array}{llllll} \max & y_1 & + & y_2 & + & y_3 \\ \text{s.t.} & \frac{2}{7}y_1 & + & \frac{5}{7}y_2 & + & \frac{3}{7}y_3 & \leq & 300 \\ & \frac{5}{7}y_1 & + & \frac{2}{7}y_2 & + & \frac{4}{7}y_3 & \leq & 300 \\ & & & \frac{2}{3}y_2 & + & \frac{2}{9}y_3 & \leq & 100 \\ & & & \frac{1}{3}y_2 & + & \frac{4}{9}y_3 & \leq & 100 \\ & & & & & \frac{2}{3}y_3 & \leq & 50 \\ & & & & & \frac{1}{3}y_3 & \leq & 50 \\ & & & & & y_1, y_2, y_3 & \geq & 0. \end{array}$$

Aufgabe 3. (Das Minimale-Spannbaum-Problem als LP)

(0 Punkte)

Gegeben sei ein Graph $G = (V, E)$ mit Kantengewichten $c : E \rightarrow \mathbb{R}$. Gesucht ist ein minimaler Spannbaum für G . Entwickeln Sie für das Problem zwei verschiedene lineare Optimierungsprobleme (mit Ganzzahligkeitsbedingungen) mithilfe jeweils einer der folgenden Definitionen für Bäume:

1. $|E| = |V| - 1$ und kreisfrei,
2. $|E| = |V| - 1$ und zusammenhängend.

Was lässt sich über die Anzahl der Nebenbedingungen aussagen? Ist es sinnvoll, das Spannbaum-Problem mithilfe dieser linearen Optimierungsprobleme zu lösen?

Bemerkung: Führe für jede Kante $e \in E$ eine Variable $x_e \in \{0, 1\}$ ein, die anzeigt, ob e im Spannbaum enthalten ist oder nicht.

Jack Edmonds¹ zeigte 1970, dass wenn man die Ganzzahligkeitsbedingungen $x_e \in \{0, 1\}$ in Modell 1. durch $0 \leq x_e \leq 1$ ersetzt, trotzdem alle Optimallösungen ganzzahlig sind. Man kann also das Spannbaum-

¹https://en.wikipedia.org/wiki/Jack_Edmonds

Problem tatsächlich als lineares Problem (ohne Ganzzahligkeitsbedingungen) modellieren. Bei Modell 2. ist dies nicht der Fall, da es auch gebrochene Optimallösungen haben kann. Hier ist die Bedingung $x_e \in \{0, 1\}$ also notwendig.

Mehr Information dazu gibt es in den Vorlesungen Diskrete Optimierung 1 & 2.

Lösung:

Wir führen für jede Kante $e \in E$ eine Variable $x_e \in \{0, 1\}$ ein, die aussagt, ob e im Spannbaum liegt ($x_e = 1$) oder nicht ($x_e = 0$). Dann lautet das Lineare Programm:

1.

$$\begin{array}{ll} \min & \sum_{e \in E} c_e x_e \\ \text{s.t.} & \sum_{e \in E} x_e = |V| - 1 \\ & \sum_{e \in E(S)} x_e \leq |S| - 1 \quad \text{for } \emptyset \neq S \subset V \\ & x \in \{0, 1\}^{|E|} \end{array}$$

2.

$$\begin{array}{ll} \min & \sum_{e \in E} c_e x_e \\ \text{s.t.} & \sum_{e \in E} x_e = |V| - 1 \\ & \sum_{e \in \delta(S)} x_e \geq 1 \quad \text{for } \emptyset \neq S \subset V \\ & x \in \{0, 1\}^{|E|} \end{array}$$

Die Anzahl der Nebenbedingungen ist jeweils exponentiell groß. Daher ist die Laufzeit eines LP-Solvers in diesem Fall auch exponentiell. Kruskal oder Prim sind dagegen schnelle polynomiale Algorithmen für das Spannbaum-Problem.

Hausübungen

Aufgabe 4. (Modellierung)

(4 Punkte)

Ein Whisky-Importeur unterhält zwar einen unbegrenzten Markt für seine Ware, aber durch die Produktionskapazitäten der drei Hersteller werden seine monatlichen Einkaufsmengen folgendermaßen begrenzt:

Hersteller A:	höchstens 2000 L zu 35 € je Liter,
Hersteller B:	höchstens 2500 L zu 25 € je Liter,
Hersteller C:	höchstens 1200 L zu 10 € je Liter.

Daraus stellt er drei Verschnitte *Sir Roses*, *Highland Wind* und *Old Regent* her, die er zu 34 €, 28,50 € bzw. 22,50 € pro Liter verkauft. Die Zusammensetzung der Verschnitte ist:

<i>Sir Roses</i>	wenigstens 60 % von A, höchstens 20 % von C,
<i>Highland Wind</i>	wenigstens 15 % von A, höchstens 60 % von C,
<i>Old Regent</i>	höchstens 50 % von C.

Wie sollten die Mischungen aussehen und wie viel sollte von jeder Mischung hergestellt werden, um einen maximalen Gewinn zu erzielen? Modellieren Sie dieses Problem als LP. (Das LP muss nicht gelöst werden.)

Lösung:

Wir verwenden die Variablen $x_{ij} \geq 0$, um die Menge Whisky (in Litern) von Hersteller $i \in \{A, B, C\}$ im Verschnitt $j \in \{S, H, O\}$ zu bezeichnen. Der Gewinn, d.h. die Differenz von Verkaufspreis minus Einkaufskosten, ist zu maximieren:

$$\begin{aligned} & 34(x_{AS} + x_{BS} + x_{CS}) + 28.5(x_{AH} + x_{BH} + x_{CH}) + 22.5(x_{AO} + x_{BO} + x_{CO}) \\ & - 35(x_{AS} + x_{AH} + x_{AO}) - 25(x_{BS} + x_{BH} + x_{BO}) - 10(x_{CS} + x_{CH} + x_{CO}) \\ = & -x_{AS} + 9x_{BS} + 24x_{CS} - 6.5x_{AH} + 3.5x_{BH} + 18.5x_{CH} - 12.5x_{AO} - 2.5x_{BO} + 12.5x_{CO} \end{aligned}$$

Die Verschnitte sollen Mindestanteile der Whiskys der Hersteller A, B, C enthalten:

$$\begin{aligned}x_{AS} &\geq 0.6(x_{AS} + x_{BS} + x_{CS}) \\x_{CS} &\leq 0.2(x_{AS} + x_{BS} + x_{CS}) \\x_{AH} &\geq 0.15(x_{AH} + x_{BH} + x_{CH}) \\x_{CH} &\leq 0.6(x_{AH} + x_{BH} + x_{CH}) \\x_{CO} &\leq 0.5(x_{AO} + x_{BO} + x_{CO})\end{aligned}$$

Die Liefermengen sind durch die Hersteller begrenzt:

$$\begin{aligned}x_{AS} + x_{AH} + x_{AO} &\leq 2000 \\x_{BS} + x_{BH} + x_{BO} &\leq 2500 \\x_{CS} + x_{CH} + x_{CO} &\leq 1200\end{aligned}$$

Wir haben insgesamt:

max	$-x_{AS}$	$+9x_{BS}$	$+24x_{CS}$	$-6.5x_{AH}$	$+3.5x_{BH}$	$+18.5x_{CH}$	$-12.5x_{AO}$	$-2.5x_{BO}$	$+12.5x_{CO}$	
s. t.	$-0.4x_{AS}$	$+0.6x_{BS}$	$+0.6x_{CS}$							≤ 0
	$-0.2x_{AS}$	$-0.2x_{BS}$	$+0.8x_{CS}$							≤ 0
				$-0.85x_{AH}$	$+0.15x_{BH}$	$+0.15x_{CH}$				≤ 0
				$-0.6x_{AH}$	$-0.6x_{BH}$	$+0.4x_{CH}$				≤ 0
							$-0.5x_{AO}$	$-0.5x_{BO}$	$+0.5x_{CO}$	≤ 0
	x_{AS}			$+x_{AH}$			$+x_{AO}$			≤ 2000
		x_{BS}			$+x_{BH}$			$+x_{BO}$		≤ 2500
			x_{CS}			$+x_{CH}$			$+x_{CO}$	≤ 1200
									x_{ij}	$\geq 0 \quad \forall i, j$

Aufgabe 5. (Konvexe Mengen)

(2+2 Punkte)

Eine Funktion $f : \mathbb{R}^n \rightarrow \mathbb{R}$ heißt *konvex*, falls für $x, y \in \mathbb{R}^n$ und $\lambda \in [0, 1]$

$$f(\lambda x + (1 - \lambda)y) \leq \lambda f(x) + (1 - \lambda)f(y)$$

gilt.

1. Beweisen oder widerlegen Sie: Sei $f : \mathbb{R}^n \rightarrow \mathbb{R}$ eine konvexe Funktion und \mathcal{C} eine konvexe Menge. Dann ist $\arg \min(f, \mathcal{C})$, d. h. die Menge der Punkte, bei denen f ihr Minimum über \mathcal{C} annimmt, konvex.
2. Eine Menge $\mathcal{K} \subseteq \mathbb{R}^n$ heißt *Kegel*, wenn mit $x \in \mathcal{K}$ auch $\alpha x \in \mathcal{K}$ für jede Zahl $\alpha \geq 0$. Beweisen oder widerlegen:
 - (a) Ein Kegel \mathcal{K} ist konvex.
 - (b) Sei \mathcal{K} ein Kegel. Es gilt $x + y \in \mathcal{K}$ für alle $x, y \in \mathcal{K}$ genau dann, wenn \mathcal{K} konvex ist.

Lösung:

1. Die Aussage ist wahr: Seien $x, y \in \arg \min(f, \mathcal{C})$. Dann gilt $f(x) = f(y) \leq f(z)$ für alle $z \in \mathcal{C}$. Da f konvex ist, gilt für $\lambda \in [0, 1]$, dass

$$f(\lambda x + (1 - \lambda)y) \leq \lambda f(x) + (1 - \lambda)f(y) \leq \lambda f(z) + (1 - \lambda)f(z) = f(z).$$

Also ist $\lambda x + (1 - \lambda)y \in \arg \min(f, \mathcal{C})$, also ist $\arg \min(f, \mathcal{C})$ konvex.

2. (i) Die Aussage ist falsch, z.B. ist

$$\mathcal{K} = \{x \in \mathbb{R}^2 : x = \alpha(1, 0)^T, \alpha \geq 0\} \cup \{x \in \mathbb{R}^2 : x = \alpha(0, 1)^T, \alpha \geq 0\}$$

ein Kegel, aber nicht konvex.

- (ii) Die Aussage ist wahr: Es gelte $x + y \in \mathcal{K}$ für alle $x, y \in \mathcal{K}$. Seien $x, y \in \mathcal{K}$ und $\lambda \in [0, 1]$.
 „ \Rightarrow “: Da \mathcal{K} ein Kegel ist, folgt $\lambda x \in \mathcal{K}$ und $(1 - \lambda)y \in \mathcal{K}$. Da nach Voraussetzung damit auch $\lambda x + (1 - \lambda)y \in \mathcal{K}$ gilt, ist \mathcal{K} konvex.
 „ \Leftarrow “: Ist \mathcal{K} konvex, folgt $x + y \in \mathcal{K}$ für alle $x, y \in \mathcal{K}$ sofort aus der Definition der Konvexität.

Aufgabe 6. (Polyeder)

(1+1+1+2 Punkte)

Welche der folgenden Mengen sind Polyeder? Beweisen oder widerlegen Sie!

1. $\mathcal{M}_1 := \{X \in \mathbb{R}^{n \times n} : a_1^\top X a_1 \leq a_2^\top X a_2\}$ mit $a_1, a_2 \in \mathbb{R}^n$.
2. $\mathcal{M}_2 := \{x \in \mathbb{R}^n : x \geq 0, e^\top x = 1, \sum_{i=1}^n x_i a_i = b_1, \sum_{i=1}^n x_i a_i^2 = b_2\}$ mit $a_1, \dots, a_n \in \mathbb{R}$ und $b_1, b_2 \in \mathbb{R}$. Dabei sei e der Vektor in \mathbb{R}^n , dessen Komponenten alle gleich 1 sind.
3. $\mathcal{M}_3 := \{x \in \mathbb{R}^n : \|x - x_0\|_2 \leq \alpha\}$ für einen gegebenen Punkt $x_0 \in \mathbb{R}^n$ und $\alpha \in \mathbb{R}$ mit $\alpha > 0$.
4. $\mathcal{M}_4 := \{x \in \mathbb{R}^n : x \geq 0, x^\top y \leq 1 \text{ für alle } y \text{ mit } \|y\|_2 = 1\}$.

Lösung:

1. Die Menge \mathcal{M}_1 ist ein Polyeder, da es durch endlich viele lineare Ungleichungen beschrieben wird.
2. Die Menge \mathcal{M}_2 ist ein Polyeder, da es durch endlich viele lineare Gleichungen und Ungleichungen beschrieben wird. (Jede Gleichung kann durch zwei Ungleichungen ausgedrückt werden).
3. \mathcal{M}_3 beschreibt eine abgeschlossene Kugel um den Punkt x_0 mit Radius α . Die Extrempunkte der Kugel sind die Randpunkte. Da von diesen unendlich viele existieren, kann \mathcal{M}_3 kein Polyeder sein.
4. Es gilt:

$$x^\top y \leq 1 \quad \forall y \text{ mit } \|y\|_2 = 1 \quad \Leftrightarrow \quad \|x\|_2 \leq 1.$$

Beweis: Aus $\|x\|_2 \leq 1$ und $\|y\|_2 = 1$ folgt wegen der Ungleichung von Cauchy-Schwarz $1 \geq \|x\|_2 \|y\|_2 \geq x^\top y$.

Sei nun $x^\top y \leq 1$ für alle y mit $\|y\|_2 = 1$. Angenommen, es existiert x mit $\|x\|_2 > 1$ und $x^\top y \leq 1$ für alle y mit $\|y\|_2 = 1$. Setze $\hat{y} := (1/\|x\|_2)x$. Dann gilt $\|\hat{y}\|_2 = 1$ und

$$x^\top \hat{y} = \frac{1}{\|x\|_2} x^\top x = \|x\|_2 > 1,$$

im Widerspruch zur Voraussetzung.

Daher ist \mathcal{M}_4 der Durchschnitt von Einheitskugel $\mathcal{B} = \{x \in \mathbb{R}^n \mid \|x\|_2 \leq 1\}$ und \mathbb{R}_+^n . Die Extrempunkte der Einheitskugel sind die Randpunkte. Da von diesen unendlich viele in \mathcal{M}_4 enthalten sind, kann \mathcal{M}_4 kein Polyeder sein.

Aufgabe 7. (Programmieraufgabe: Ernährungsproblem als LP)

(4 Punkte)

Laden Sie die Datei `ProgAufgabe11.py` aus Studon herunter und fügen Sie sie in ihr PyCharm ein.

In dieser Aufgabe implementieren Sie eine Methode, die ein simples Ernährungsoptimierungsproblem löst. Implementieren Sie hierfür die Methode

```
diet(categories, minNutrition, maxNutrition, foods, cost, nutritionValues).
```

- `categories` ist eine Liste von Nährstoffen (Kalorien, Protein, etc...).
- `minNutrition` und `maxNutrition` sind Dictionaries, die zu jedem Nährstoff den minimalen und maximalen Nährwert angeben (mindestens so-und-so-viel Protein, maximal so-und-so-viel Kalorien, usw...).
- `foods` ist eine Liste aller verfügbaren Speisen.
- `cost` ist ein Dictionary, das zu jeder Speise den Preis pro Einheit angibt.
- `nutritionValues` ist ein Dictionary, das zu jeder Kombination `[food, category]` aus Speise und Nährstoff angibt, welchen Nährwert zum angegebenen Nährstoff die jeweilige Speise besitzt (pro Einheit). D.h. Pasta hat einen Nährwert von `nutritionValues['pasta', 'protein']` für Protein.

Das Optimierungsproblem ist nun, eine Kombination aus Speisen zu finden, die die gegebenen Nährwertschranken erfüllt und minimalen Preis besitzt (die Speisemengen müssen nicht ganzzahlig sein). Natürlich dürfen Speisen nicht negativ sein.

Die Methode `diet` soll dieses Problem mittels `gurobipy` als Gurobi-Modell darstellen und lösen.

Überlegen Sie sich auch eine sinnvolle (!) Ausgabe, die folgendes anzeigt: Welche Speisen wurden überhaupt gekauft, wie viel von jeder Speise und welche Nährwerte ergeben sich insgesamt.

TIPP: Die Werte der Gurobi-Variablen der nichtgekauften Speisen sind numerisch bedingt meist nicht exakt 0, sondern nur nahe 0. Bedenken Sie dies bei der Feststellung, welche Speisen gekauft wurden.

Lösung:

Die Musterlösung der Programmieraufgabe finden Sie im StudOn in der Datei `ProgAufgabe11_Loesung.py`.

Gruppe 1: Abgabe der Hausaufgaben am 23.01.2018 in der Übung.
Gruppe 2+3: Abgabe der Hausaufgaben am 24.01.2018 in der jeweiligen Übung.