## Einführung in die Optimierung, Übung 4, Lösungsvorschlag

### Gruppenübung

**G 11** Betrachten Sie das lineare Optimierungsproblem $\max\{c^T x : x \in \mathcal{P}\}$, wobei das Polyeder $\mathcal{P} = \{x \in \mathbb{R}^2 : Ax \leq b\}$ gegeben ist mit

$$A = \begin{pmatrix} -2 & 1 \\ -2 & -1 \\ -1 & -2 \\ 1 & -2 \end{pmatrix}, \qquad b = \begin{pmatrix} 3 \\ -3 \\ -3 \\ 3 \end{pmatrix}.$$

Für welche Vektoren $c \in \mathbb{R}^2 \setminus \{0\}$ hat das lineare Problem

(a) genau eine Optimallösung,
(b) unendlich viele Optimallösungen,
(c) keine Optimallösung?

Geben Sie eine Ungleichung $a^T x \leq \alpha$ (mit $a \in \mathbb{R}^n, \alpha \in \mathbb{R}$) an, so dass das lineare Problem

$$\max\{c^T x : Ax \leq b, a^T x \leq \alpha\}$$

für jedes $c \in \mathbb{R}^2 \setminus \{0\}$ mindestens eine Optimallösung hat.

*(a) Das lineare Optimierungsproblem hat dann genau eine Optimallösung, wenn diese in einer der drei Ecken angenommen wird.*

*Ecke $(3,0)$: Der Normalvektor der Zielfunktion $c_1x_1 + c_2x_2$ liegt in dem Inneren des Kegels, der von der Normalvektoren der Geraden $-2x_1 + x_2 = 3$ und $-2x_1 - x_2 = -3$, die vom Polyeder wegzeigen, aufgespannt wird:*

$$(c_1, c_2)^T = \alpha_1(-2,1)^T + \alpha_2(-2,-1)^T, \qquad \alpha_1, \alpha_2 > 0.$$

$$\begin{aligned} c_1 &= -2\alpha_1 - 2\alpha_2 \\ c_2 &= \alpha_1 - \alpha_2. \end{aligned}$$

*Bzw. der Normalvektor der Zielfunktion $c_1x_1 + c_2x_2$ ist das Vielfache einer echten Konvexkombination der Normalvektoren der Geraden $-2x_1 + x_2 = 3$ und $-2x_1 - x_2 = -3$, die vom Polyeder wegzeigen.*

$$(\frac{c_1}{\alpha}, \frac{c_1}{\alpha})^T = \lambda(-2,1)^T + (\lambda)(-2,-1)^T, \qquad \alpha > 0,\ 0 < \lambda < 1.$$

$$\begin{aligned} \frac{c_1}{\alpha} &= -2 \\ \frac{c_2}{\alpha} &= 2\lambda - 1 \Leftrightarrow -\alpha < c_2 < \alpha. \end{aligned}$$

*Ecken $(1,1)$ und $(0,3)$: Analog.*

*(b) Das lineare Optimierungsproblem hat unendlich viele Optimallösungen, wenn die Zielfunktion parallel zu einer der Kanten ist, d.h. wenn $c$ ein positives Vielfaches eines der folgenden Vektoren ist:*

$$c = (-2,1)^T, c = (-2,-1)^T, c = (-1,-2)^T, c = (1,-2)^T.$$

*Ausserdem hat das LP unendlich viele Optimallösungen, wenn $c = (0,0)^T$, da dann jeder zulässige Punkt Optimallösung ist.*

*(c) Das lineare Optimierungsproblem hat dann keine Optimallösung, wenn $c$ nicht eine der in (a) oder (b) beschriebenen Bedingungen erfüllt.*

*Um zu garantieren, dass das Problem immer mindestens eine Optimallösung hat, muss sichergestellt werden, dass die zulässige Menge kompakt, aber nicht leer ist. Dies kann zum Beispiel durch Hinzufügen der Ungleichung $x_1 + x_2 \leq 10$ geschehen.*

**G 12** (A) Beweisen oder widerlegen Sie die folgende Aussage: Seien $c \in \mathbb{R}^n, A \in \mathbb{R}^{m\times n}, b \in \mathbb{R}^m$. Wenn ein lineares Programm der Form $\max\{c^T x | Ax \leq b, x \geq 0\}$ unbeschränkt ist, dann gibt es einen Index $k = 1, \ldots, n$, so dass das Problem $\max\{c_k x_k | Ax \leq b, x \geq 0\}$ unbeschänkt ist.

Beweisen oder widerlegen Sie auch die Umkehrung dieser Aussage.

(B) Formulieren Sie das Problem $\mathsf{min}\{\mathsf{max}\{c^T x - c_0, d^T x - d_0\} | Ax \leq b, 0 \leq x \leq 1\}$ als lineares Programm. (Dabei sind $c_0$, $d_0 \in \mathbb{R}$, $b \in \mathbb{R}^m$, $c, d \in \mathbb{R}^n$, $A \in \mathbb{R}^{m\times n}$.)

(C) Formulieren Sie das Problem $\mathsf{max}\{\mathsf{max}\{c^T x - c_0, d^T x - d_0\} | Ax \leq b, 0 \leq x \leq 1\}$ als gemischt-ganzzahliges Programm. (Dabei sind $c_0$, $d_0 \in \mathbb{R}$, $b \in \mathbb{R}^m$, $c, d \in \mathbb{R}^n$, $A \in \mathbb{R}^{m\times n}$.)

*(A) (a) gilt. Beweis indirekt: Angenommen, für alle Indizes $k$ ist $\max\{c_k x_k : Ax \leq b, x \geq 0\}$ beschränkt. Dann gibt es $M_k \in \mathbb{R}$ mit*

$$c_k x_k \leq M_k \qquad \forall x \geq 0,\ Ax \leq b.$$

*Dann folgt:*

$$c^T x = \sum_k c_k x_k \leq \sum_k M_k < \infty \qquad \forall x \geq 0,\ Ax \leq b.$$

*Dann gilt aber auch*

$$\max\{c^T x : Ax \leq b, x \geq 0\} < \infty.$$

*(b) gilt nicht. Ein Gegenbeispiel ist:*

$$\max\{x_1 - x_2 : x_1 - x_2 = 0, x_1, x_2 \geq 0\}.$$

*(B) Das Problem kann als lineares Optimierungsproblem formuliert werden, indem man eine zusätzliche Variable $y \in \mathbb{R}$ einführt. Bezeichnet man $y = \max\{c^T x - c_0, d^T x - d_0\}$, so besteht das Optimierungsproblem darin, $y$ zu minimieren. Als Nebenbedingungen erhält man aus der Definition von $y$:*

$$c^T x - c_0 \leq y \qquad \text{und} \qquad d^T x - d_0 \leq y.$$

*Das LP lautet dann:*

$$\begin{array}{ll} \min & y \\ \text{s.t.} & c^T x - y \leq c_0 \\ & d^T x - y \leq d_0 \\ & Ax \leq b \\ & 0 \leq x \leq 1 \end{array}$$

*(C) Das Problem kann als lineares Optimierungsproblem formuliert werden, indem man eine zusätzliche Variable $y \in \mathbb{R}$ einführt. Bezeichnet man $y = \max\{c^T x - c_0, d^T x - d_0\}$, so besteht das Optimierungsproblem darin, $y$ zu maximieren. Wir führen eine binäre Variable $\delta$ ein, so dass:*

$$\begin{array}{lll} \delta = 0 & \Leftrightarrow & c^T x - c_0 \leq d^T x - d_0 \\ \delta = 1 & \Leftrightarrow & d^T x - d_0 \leq c^T x - c_0 \end{array}$$

*Das MIP lautet dann:*

$$\begin{array}{ll} \max & y \\ \text{s.t.} & y \leq c^T x - c_0 + (1-\delta)M \\ & y \leq d^T x - d_0 + \delta M \\ & c^T x - c_0 \leq d^T x - d_0 + \delta M \\ & d^T x - d_0 \leq c^T x - c_0 + (1-\delta)M \\ & Ax \leq b \\ & 0 \leq x \leq 1 \\ & \delta \in \{0, 1\}. \end{array}$$

*Wobei $M \in \mathbb{R}$ eine hinreichend große Zahl ist. Die Existenz von $M$ ist garantiert durch $0 \leq x \leq 1$.*

**G 13** (A) Sei $A \in \mathbb{R}^{m \times n}$ und $b \in \mathbb{R}^m$. Zeigen Sie: *Ein Polyeder* $\mathcal{P}(A,b) = \{x \in \mathbb{R}^n : Ax \leq b\}$ *hat nur endlich viele Ecken.*

Finden Sie eine obere Schranke $\mu(m,n)$ für die Anzahl der Ecken von $\mathcal{P}(A,b)$ und geben Sie ein Beispiel eines Polyeders an, das genau $\mu(m,n)$ Ecken hat.

(B) Sei $\mathcal{P}(A,b) \subset \mathbb{R}^n$ ein Polytop, und sei $x$ eine Ecke von $\mathcal{P}(A,b)$. $x$ heißt nicht degeneriert, falls $|eq(\{x\})| = n$, anderenfalls heißt $x$ degeneriert.

(a) Zeigen Sie: *Eine nicht degenerierte Ecke ist zu genau $n$ Ecken adjazent.*

(b) Skizzieren Sie ein Polytop, das nur nicht degenerierte Ecken hat, und ein Polytop, das nur degenerierte Ecken hat.

*(A) $x$ ist eine Ecke von $\mathcal{P}(A,b)$ $\iff$ $\operatorname{rang}(A_{\mathrm{eq}(\{x\})\cdot}) = n$. Das bedeutet, die Anzahl der Ecken ist kleiner oder gleich der Anzahl der Teilmatrizen von $A$, die $n$ Zeilen haben. Diese Anzahl ist gleich der Anzahl der Möglichkeiten, aus den $m$ Zeilen von $A$ $n$ Stück auszuwählen, also $\binom{m}{n}$. Damit ist*

$$\mu(m,n) = \binom{m}{n} < \infty$$

*eine obere Schranke für die Anzahl der Ecken von $\mathcal{P}(A,b)$, daher kann das Polyeder nur endlich viele Ecken haben.*

*Ein Polyeder mit genau $\mu(m,n)$ Ecken ist z.B. durch*

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

*gegeben.*

*(B) (a)*

*Sei $x$ eine nicht degenerierte Ecke, $I = \mathrm{eq}(\{x\})$. Dann ist $A_I$ regulär. Laut der Definition ist eine Kante $\mathcal{K} \subset \mathcal{P}$ eine Seitenfläche der Dimension eins. Nach dem Satz über die Seitenflächendimension gilt $\operatorname{rang}(A_{\mathrm{eq}(\mathcal{K})\cdot}) = n-1$. D.h. eine Kante $\mathcal{K} \subset \mathcal{P}$ ist durch mindestens $(n-1)$ Gleichungen bestimmt, bzw. es existiert eine $(n-1) \times n$–Teilmatrix $A_J$ von $A$ mit $\operatorname{rang} A_J = n-1$ so, dass $x \in \mathcal{K} \iff x \in \mathcal{P}, A_J x = b_J$.*

*Jede Kante von $\mathcal{P}$, auf der $x$ liegt, ist daher durch ein System von linearen Gleichungen charakterisiert, das aus $A_I x = b_I$ hervorgeht, indem man genau eine Gleichung weglässt. Es gibt $n$ Möglichkeiten, das zu tun, daher $n$ mögliche Kanten. Da das Polytop $\mathcal{P}$ beschränkt ist, muss auf jeder dieser Kanten eine weitere Ecke liegen, daher hat $x$ genau $n$ Nachbarn.*

*(b)*

*Das Einheitsquadrat ist ein Beispiel für ein Polytop, das nur nicht degenerierte Ecken hat. Oktaeder sind Polytope, die nur degenerierte Ecken besitzen.*

**Hausübung**

**H 10 Bonusaufgabe** (5 Punkte)

(A) Als Leiter der IT-Abteilung einer Versicherung ist es Ihre Aufgabe, den Betrieb und die Wartung des Großrechnerzentrums zu gewährleisten. Dazu stehen Ihnen sechs Systemadministratoren (kurz: Admins) zur Verfügung, die auf Stundenbasis beschäftigt werden. Sie sind für deren Einsatzplanung verantwortlich. Von Montag bis Freitag zwischen 8 Uhr morgens und 22 Uhr abends muss gewährleistet sein, dass stets ein Admin im Rechenzentrum arbeitet. Die Admins haben Ihnen folgende Liste gegeben, aus der hervorgeht, wieviele Stunden sie an welchem Wochentag höchstens arbeiten und wieviel sie pro Stunde verdienen.

| Name | Lohn/Stunde | Mo | Di | Mi | Do | Fr |
|---|---|---|---|---|---|---|
| K.C. | 300,- | 6 | 0 | 6 | 0 | 6 |
| D.H. | 210,- | 0 | 6 | 0 | 6 | 0 |
| H.B. | 195,- | 4 | 8 | 4 | 0 | 4 |
| S.C. | 250,- | 5 | 5 | 5 | 0 | 5 |
| K.S. | 220,- | 3 | 0 | 3 | 8 | 0 |
| N.K. | 270,- | 0 | 0 | 0 | 6 | 2 |

Ferner haben die sechs Admins mit Ihrem Vorgänger eine Mindestbeschäftigung ausgehandelt: K.C., D.H., H.B. und S.C. arbeiten mind. 8 Stunden pro Woche, K.S. und N.K. mind. 7 Stunden pro Woche. Da das Budget Ihrer Abteilung knapp bemessen ist, sind Sie an einer kostenminimalen Verteilung der Arbeit interessiert. Formulieren Sie das entsprechende Oprimierungsproblem.

(B) Formulieren Sie das Optimierungsproblem mittels der Modellierungssprache ZIMPL und berechnen Sie die Lösung mittels des MIP-Lösers SCIP.

Abgabe: Die ZIMPL-Datei bitte zu zweit bearbeiten und eine ausgedruckte Kopie der Datei sowie die Lösung abgeben.

*(A) (3 Punkte)*
*Sei* $W := \{$ *Mo, Di, Mi, Do, Fr* $\}$ *die Menge der Wochentage, und* $P := \{$ *K.C., D.H., H.B., S.C., K.S, N.K.* $\}$ *die Menge der einsetzbaren Personen. Bezeichne mit* $a_{i,j}$ *die maximale Arbeitszeit von Person* $i$ *am Tag* $j$*. Sei* $c_i$ *der Stundenlohn von Person* $i$*, und sei* $t_{i,j}$ *die Arbeitszeit von Person* $i$ *am Tag* $j$*.*

*Ziel ist es, die Admins so kostengünstig wie möglich einzusetzen:*

$$\min \sum_{(i,j)\in P\times W} c_i t_{i,j}.$$

*Dabei darf Admin* $i$ *am Tag* $j$ *nicht länger beschäftigt werden als vorgegeben:*

$$0 \leq t_{i,j} \leq a_{i,j} \quad \forall\, (i,j) \in P \times W.$$

*Einige Admins müssen mindestens 8 Stunden beschäftigt werden, andere mindestens 7 Stunden:*

$$\begin{aligned} \sum_{j\in W} t_{i,j} &\geq 8 \qquad \forall\, i \in \{ \text{K.C., D.H., H.B., S.C.} \}, \\ \sum_{j\in W} t_{i,j} &\geq 7 \qquad \forall\, i \in \{ \text{K.S., N.K.} \}. \end{aligned}$$

*Schließlich muss sichergestellt sein, dass der Rechnerraum täglich 14 Stunden besetzt ist:*

$$\sum_{i\in P} t_{i,j} = 14 \qquad \forall\, j \in W.$$

(B) *(2 Punkte)*

*ZIMPL-Modell:*

```
set W := { 'Mo','Di','Mi','Do','Fr' };
set P := { 'KC','DH','HB','SC','KS','NK' };
param a[P*W] :=
<'KC','Mo'> 6, <'KC','Di'> 0, <'KC','Mi'> 6, <'KC','Do'> 0, <'KC','Fr'> 6,
<'DH','Mo'> 0, <'DH','Di'> 6, <'DH','Mi'> 0, <'DH','Do'> 6, <'DH','Fr'> 0,
<'HB','Mo'> 4, <'HB','Di'> 8, <'HB','Mi'> 4, <'HB','Do'> 0, <'HB','Fr'> 4,
<'SC','Mo'> 5, <'SC','Di'> 5, <'SC','Mi'> 5, <'SC','Do'> 0, <'SC','Fr'> 5,
<'KS','Mo'> 3, <'KS','Di'> 0, <'KS','Mi'> 3, <'KS','Do'> 8, <'KS','Fr'> 0,
<'NK','Mo'> 0, <'NK','Di'> 0, <'NK','Mi'> 0, <'NK','Do'> 6, <'NK','Fr'> 2;
param c[P] := <'KC'> 300, <'DH'> 210, <'HB'> 195, <'SC'> 250, <'KS'> 220, <'NK'>
270;
var t[P*W] real >=0;
minimize cost: sum <p,w> in P*W do c[p]*t[p,w];
subto upperBound: forall <p,w> in P*W do t[p,w] <= a[p,w];
subto lowerBound8: forall <p> in { 'KC','DH','HB','SC' } do
sum <w> in W do t[p,w] >= 8;
subto lowerBound7: forall <p> in { 'KS','NK' } do
sum <w> in W do t[p,w] >= 7;
subto oneAdmin: forall <w> in W do
sum <p> in P do t[p,w] == 14; # 14 hours = 22 - 8
```

*Zielfunktionswert: 16190*

| *Name* | *Mo* | *Di* | *Mi* | *Do* | *Fr* |
|---|---|---|---|---|---|
| *K.C.* | *3* | *0* | *2* | *0* | *3* |
| *D.H.* | *0* | *6* | *0* | *6* | *0* |
| *H.B.* | *4* | *8* | *4* | *0* | *4* |
| *S.C.* | *4* | *0* | *5* | *0* | *3* |
| *K.S.* | *3* | *0* | *3* | *3* | *0* |
| *N.K.* | *0* | *0* | *0* | *5* | *2* |