Technische Universität München
Zentrum Mathematik
M1 - Mathematische Optimierung
Prof. Dr. M. Ulbrich
Dipl.-Math. B. von Loesch

7. Übungen zu *Linear Optimization* (WS 2009/2010)

Hausaufgaben (Abgabe am 01./02.02. bzw. 08.02./09.02. in den Übungen)

Aufgabe 7.1 (Indexmengen und degenerierte Ecken): **(ca. 4 Punkte)**
Gegeben sei der zulässige Bereich

$$X_b := \left\{ x \in \mathbb{R}^3 \,\middle|\, Ax \leq b,\ A = \begin{pmatrix} 0 & 1 & 1 \\ 1 & 0 & 1 \\ -1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & -1 \end{pmatrix},\ b = \begin{pmatrix} 1 \\ 1 \\ 0 \\ 0 \\ 0 \end{pmatrix} \right\}$$

aus Aufgabe H3.3 b). Wir betrachten das Problem

$$\min_x -\frac{1}{2}x_1 - \frac{1}{2}x_2 - \frac{3}{4}x_3 \quad \text{u.d.N.} \quad x \in X_b,$$

sei $x^0 = (0, 0, 1)^T$.

Bestimmen Sie (ggf. mit einer Zeichnung) alle Indexmengen $\mathcal{A} \subset \mathcal{A}(x^0)$ mit $|\mathcal{A}| = \text{Rang}(A_{\mathcal{A}}) = 3$, so dass für mindestens ein $j \in \mathcal{A}$ die Lösung $s \in \mathbb{R}^n$ der Gleichung $A_{\mathcal{A}}s = -(\delta_{ij})_{i\in\mathcal{A}}$ eine Abstiegskante ist, also $A_{\mathcal{A}(x^0)}s \leq 0$ und $c^Ts < 0$ gilt.

Aufgabe 7.2 (Klee und Minty Beispiel): **(ca. 8 Punkte)**
Wir betrachten das folgende auf Klee und Minty zurückgehende LP

$$\begin{array}{lll} \min_{x\in\mathbb{R}^n} & -\sum_{i=1}^n 2^{n-i}x_i & \\ \text{u.d.N.} & x_1 & \leq 5 \\ & 4x_1 + x_2 & \leq 25 \\ & 8x_1 + 4x_2 + x_3 & \leq 125 \\ & \vdots & \\ & \sum_{i=1}^{n-2} 2^{n-i}x_i + x_{n-1} & \leq 5^{n-1} \\ & \sum_{i=1}^{n-1} 2^{n+1-i}x_i + x_n & \leq 5^n \\ & x & \geq 0 \end{array} \qquad \text{(KM)}$$

a) Zeigen Sie, dass der Punkt $x^* = (0, \ldots, 0, 5^n)^T$ eine optimale Lösung von (KM) ist.

b) Zeichnen Sie den zulässigen Bereich für $n = 2$ und $n = 3$.

c) Zeigen Sie, dass der zulässige Bereich 2^n Ecken hat für $n \geq 2$. **Tipp**: Induktion.

d) Schreiben Sie ein Matlab-Skript, welches bei Eingabe von n die Matrix `A` $\in \mathbb{R}^{2n\times n}$ und die Vektoren `b` $\in \mathbb{R}^{2n}$, `c` $\in \mathbb{R}^n$ zurückgibt, die das Problem (KM) in Normalform repräsentieren.

e) Verweden Sie den Matlab-Code `SimplexS` von der Webseite, um das Beispiel für verschiedene n jeweils mit Startwert $x^0 = 0$ zu lösen. Wie viele Iterationen in Abhängigkeit von n werden benötigt?

Aufgabe 7.3 (Regel von Bland): **(ca. 5 Punkte)**
Verwenden Sie Algorithmus 5.1.3 mit der Regel von Bland in Schritt 4, um das lineare Programm

$$\begin{aligned} \min_{x\in\mathbb{R}^4} \quad & -10x_1 + 57x_2 + 9x_3 + 24x_4 \\ \text{u.d.N.} \quad & x_1 - 11x_2 - 5x_3 + 18x_4 \leq 0 \\ & x_1 - 3x_2 - x_3 + 2x_4 \leq 0 \\ & x_1 \leq 1 \\ & x \geq 0. \end{aligned}$$

ausgehend von $x^0 = 0$ zu lösen. Wählen Sie als Arbeitsmenge $\mathcal{A}_0 = \{1, 2, 6, 7\}$. Für Zwischenrechnungen, wie das Lösen von linearen Gleichungssystemen, können Sie Matlab oder andere Mathematik-Programme verwenden.

Regel von Bland Wähle den kleinsten Index $j_k \in \mathcal{A}_k$ mit $\lambda^k_{j_k} < 0$, berechne s^k durch Lösen von

$$A_{\mathcal{A}_k} s^k = -(\delta_{ij_k})_{i\in\mathcal{A}_k}$$

und wähle $i_k \in \mathcal{A}(x^k)\backslash\mathcal{A}_k$ kleinstmöglich mit

$$a^T_{i_k} s^k > 0.$$

Setze $x^{k+1} = x^k$ und $\mathcal{A}_{k+1} := (\mathcal{A}_k\backslash\{j_k\}) \cup \{i_k\}$ und gehe in die nächste Iteration.

Aufgabe 7.4 (Programmieraufgabe): **(ca. 10 Punkte)**
Implementieren Sie den dualen Simplexalgorithmus unter Annahme von Ecken (Algorithmus 5.1.3 aus der Vorlesung) in Matlab. Verwenden Sie die gleiche Funktionsdeklaration wie der Code `SimplexS`, den Sie auf der Vorlesungs-Homepage finden:

```
function [x,k] = simplex(x, c, A, b)
```

Einige Hinweise zur Implementierung:

- Sie können Teile aus dem Code `SimplexS` verwenden;
- Zur Auswahl der aktiven Menge $\mathcal{A}_0$ im Fall von Degeneriertheit könnte der Befehl `rref` hilfreich sein;
- Verwenden Sie für Schritt 4 die Regel von Bland;
- Verwenden Sie den `find` Befehl, um die Indizes von wahren Einträgen eines Vektors zu bekommen. Beispiel: `find(s < 0)` gibt die Indizes zurück, für die der Vektor `s` negativ ist.

Testen Sie Ihren Code an den Problemen aus den Aufgaben H7.2 und H7.3 und mit dem Beispiel, welches Sie auf der Vorlesungshomepage finden.