Einführung in die Optimierung
8. Übungsblatt

TECHNISCHE UNIVERSITÄT DARMSTADT

Fachbereich Mathematik **WS 2012/13**
Prof. Dr. Stefan Ulbrich **13./14.12.2012**
Dipl.-Math. Madeline Lips

Rechnerübung

Die Rechnerübung kann in Gruppen von bis zu 3 Personen bearbeitet werden. Die Abgabe erfolgt genau 7 Wochen später – 31.01.2013 bzw. 01.02.2013 – bis zu Beginn Ihrer Übung per email an Ihren jeweiligen Übungsleiter.

Übungsgruppe	Abgabe per email an
1	janine.barbehoen@gmx.de **und** lips@matheamtik.tu-darmstadt.de
2	janine.barbehoen@gmx.de **und** lips@matheamtik.tu-darmstadt.de
3	Jonas.Weyer@gmx.de **und** lips@matheamtik.tu-darmstadt.de
4	marc_fuechtenhans@live.de **und** lips@matheamtik.tu-darmstadt.de
5	arniberg@hotmail.com **und** lips@matheamtik.tu-darmstadt.de
6	martin.hameister@gmx.de **und** lips@matheamtik.tu-darmstadt.de

Außerdem müssen Sie mit dem/der jeweiligen Korrektor/-in einen Termin vereinbaren, an dem Sie ihm/ihr persönlich (Gruppe vollzählig) Ihre Aufgaben vorstellen.
Berücksichtigen Sie bei Ihrer Implementierung das mögliche Auftreten von Rundungsfehlern!

Aufgabe R1 (Simplex-Algorithmus) (15 Punkte)
Implementieren Sie mit Matlab den Simplex-Algorithmus aus der Vorlesung (Algorithmus 5.6). Definieren Sie dazu die Funktion

```
[xOpt, message] = simplex(A, b, c, B).
```

Die Variablen sollen die folgende Bedeutung haben:

- Der Vektor `xOpt` ist eine Optimallösung, falls eine solche existiert.
- Die Zeichenkette (*string*) `message` soll die Mitteilung „Das LP besitzt eine Optimallösung.“ oder „Das LP ist unbeschränkt.“(*unbounded*) beinhalten.
- Die Matrix `A` und die Vektoren `b` und `c` entsprechen denen aus der Standardform (siehe (5.1)). Dabei sollen für $A \in \mathbb{R}^{m\times n}$ die Vorraussetzungen (*assumptions*) $n \geq m$ und $\operatorname{rang}(A) = m$ gelten.
- Der Vektor `B` soll eine zulässige Startbasis (*initial solution*) repräsentieren. Die m Einträge des Vektors sollen die in der Basis enthaltenen Indizes sein.

Aufgabe R2 (PhaseI) (13 Punkte)
Implementieren Sie in Matlab die Phase I des Simplex-Algorithmus aus der Vorlesung (Algorithmus 5.17). Definieren Sie dazu die Funktion

```
[B, message] = phaseI(A, b).
```

Die Variablen sollen die folgende Bedeutung haben:

- Der Vektor `B` repräsentiert eine zulässige Basis, falls eine solche existiert.
- Die Zeichenkette `message` soll die Mitteilung „Das LP besitzt eine zulässige Basis.“, „Das LP ist unzulässig.“ oder „rang(A) < m“ beinhalten.
- Die Matrix `A` und der Vektor `b` entsprechen denen aus der Standardform (siehe (5.1)). Dabei soll $b \geq 0$ gelten.

Aufgabe R3 (LP-Löser) (2 Punkte)
Implementieren Sie in Matlab einen LP-Löser mit Hilfe der Funktionen `simplex` und `phaseI`. Definieren Sie dazu die Funktion

`[xOpt, message, BOpt] = lpSolve(A, b, c).`
Die Variablen sollen die folgende Bedeutung haben:

- Der Vektor `xOpt` ist eine Optimallösung, falls eine solche existiert.
- Die Zeichenkette `message` soll die Mitteilung „Das LP besitzt eine Optimallösung.“, „Das LP ist unbeschränkt.“, „rang(A) < m“ oder „Das LP ist unzulässig“ beinhalten.
- Der Vektor `BOpt` soll die Basis der Optimallösung enthalten – falls eine Lösung existiert.
- Die Matrix `A` und die Vektoren `b` und `c` entsprechen denen aus der Standardform (siehe (5.1)). Dabei soll $b \geq 0$ gelten.

Aufgabe R4 (Implementierungstests) (5 Punkte)

(a) Testen Sie Ihre Implementierung aus **Aufgabe R1** mit dem LP aus dem Beispiel 5.7 aus der Vorlesung und mit dem folgenden LP:

$$\begin{array}{lrcrcl} \max & x_1 & + & x_2 & & \\ \text{s.t.} & x_1 & - & x_2 & \leq & 2 \\ & -2x_1 & + & x_2 & \leq & 1 \\ & & & x_1, x_2 & \geq & 0 \end{array}$$

(b) Testen Sie Ihre Implementierung aus **Aufgabe R2** mit den folgenden Daten:

$$A_1 = \begin{pmatrix} 1 & -1 \\ -1 & -1 \end{pmatrix}, b_1 = \begin{pmatrix} 1 \\ 1 \end{pmatrix}; A_2 = \begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{pmatrix}, b_2 = \begin{pmatrix} 1 \\ 3 \\ 5 \end{pmatrix}; A_3 = \begin{pmatrix} 1 & 0 & 3 & -1 \\ -1 & -1 & 0 & -4 \\ -1 & -2 & 1 & -3 \end{pmatrix}, b_3 = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}.$$

(c) Geben Sie die Lösungen von angegebenen Optimierungsproblemen an! Verwenden Sie hierzu den von Ihnen entwickelten Code aus **Aufgabe R3**.

i. LPT1.mat
ii. LPT2.mat
iii. LPT3.mat

Aufgabe R5 (Kreiseln) (5 Punkte)

Gegeben sei das LP

$$\begin{array}{lrcrcrcrcl} \min & -2x_1 & - & 3x_2 & + & x_3 & + & 12x_4 & & \\ \text{s.t.} & -2x_1 & - & 9x_2 & + & x_3 & + & 9x_4 & \leq & 0 \\ & \frac{1}{3}x_1 & + & x_2 & - & \frac{1}{3}x_3 & - & 2x_4 & \leq & 0 \\ & & & & & & & x_1, \ldots, x_4 & \geq & 0 \end{array}.$$

Zeigen Sie, dass der Simplex-Algorithmus bei diesem Beispiel kreiseln (*cycle*) kann. Geben Sie dazu in jeder Iteration die aktuelle Basis an.
Überlegen Sie sich, wie das Kreiseln umgangen werden kann und verbessern Sie gegebenenfalls Ihre Implementierung dementsprechend. Geben Sie die Lösung des Optimierungsproblems an.

Aufgabe R6 (Modellierung) (10 Punkte)

(a) Gegeben sei ein Hängegerüst wie in Abbildung 1. Die Seile S_1 und S_2 können je 300kg Last, die Seile S_3 und S_4 je 100kg und die Seile S_5 und S_6 jeweils 50kg Last tragen. 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.

i. Formulieren Sie dieses Problem als lineares Programm.
ii. Stellen Sie das dazugehörige duale lineare Programm auf und diskutieren Sie die Bedeutung einer Optimallösung.

(b) Lösen Sie sowohl das primale als auch das duale LP mit Hilfe Ihrer Implementierung. Gilt für die Optimallösungen der ursprünglichen nicht in die Standardform gebrachten Probleme strikte Komplementarität? Lesen Sie aus der Optimallösung ab, um wieviel die Belastbarkeit der einzelnen Seile maximal verringert werden kann, sodass das Gerüst die Gewichte noch tragen kann.

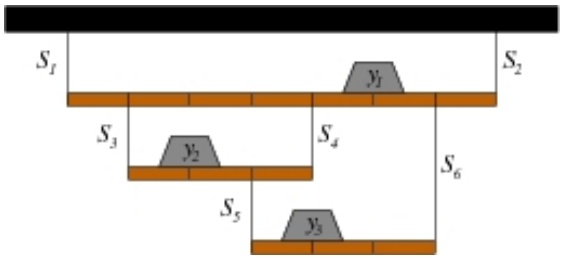

Abbildung 1: Gerüst zu Aufgabe (a)