Otto-von-Guericke-Universität Magdeburg
Institut für Mathematische Optimierung
Prof. Dr. Friedrich Juhnke
Dr. Matthias Köppe, Gebäude 18, Raum 311

Wintersemester 2003/2004

5. Übungsblatt

Lineare Optimierung

`http://www.math.uni-magdeburg.de/~mkoeppe/lehre/opt1-2003`

Abgabe der Übungsaufgaben: bis Donnerstag, 20. November, zu Beginn der Übung

21. Aufgabe **(10 Punkte)**

Die Größe y ist eine Funktion einer anderen Größe x. Es wird vermutet, daß y eine lineare oder eine quadratische Funktion von x ist. Eine Reihe von Wertepaaren $\big(x, y(x)\big)$ wurden beobachtet und sind in der folgenden Wertetabelle aufgelistet.

x	0.0	0.5	1.0	1.5	1.9	2.5	3.0	3.5	4.0	4.5
y	1.0	0.9	0.7	1.5	2.0	2.4	3.2	2.0	2.7	3.5

x	5.0	5.5	6.0	6.6	7.0	7.6	8.5	9.0	10.0
y	1.0	4.0	3.6	2.7	5.7	4.6	6.0	6.8	7.3

Lösen Sie die folgenden Aufgaben mittels linearer Programmierung, d. h., geben Sie jeweils ein LP-Modell an, lösen Sie es mit GLPK und stellen Sie die Lösung graphisch dar.

(a) Finde die „beste“ Gerade $y = bx + a$ zu den gegebenen Daten, so daß die Summe der Abweichungen zwischen den beobachteten Werten für y und den durch die Geradengleichung „vorhergesagten“ Werten minimal ist.

(b) Finde die „beste“ Gerade, so daß die maximale Abweichung zwischen beobachteten und vorhergesagten Werten minimal ist.

Bemerkung: In der Statistik interessiert man sich im allgemeinen mehr für die Minimierung der Quadrate der Abweichungen. Es gibt aber Fälle, wo die Minimierung des Maximums der Abweichungen akzeptabel oder gar gewünscht ist. In jedem Fall ist es mit Hilfe eines linearen Modells möglich, auch sehr große Datenmengen zu bearbeiten.

22. Aufgabe **(10 Punkte)**

Seien M_1, M_2 nichtleere Teilmengen des $\mathbf{R}^n$. M_1 sei abgeschlossen, M_2 sei kompakt. Beweisen Sie: $M_1 - M_2$ (Minkowski-Addition) ist abgeschlossen.

23. Aufgabe **(10 Punkte)**

Zeigen Sie: Jede abgeschlossene konvexe Menge $M \subseteq \mathbf{R}^n$ ist gleich dem Durchschnitt aller M enthaltenden abgeschlossenen Halbräume.

24. Aufgabe **(10 Punkte)**

Lösen Sie die folgenden Optimierungsprobleme mit dem Simplex-Algorithmus auf Papier. (Zur Kontrolle können Sie GLPK verwenden.)

a)

$$\begin{aligned} \max \quad & 2x_1 - 3x_2 + x_3 \\ \text{s. t.} \quad & 3x_1 + 6x_2 + x_3 \leq 6 \\ & 4x_1 + 2x_2 + x_3 \leq 4 \\ & x_1 - x_2 + x_3 \leq 3 \\ & x_i \geq 0 \end{aligned}$$

b)

$$\begin{aligned} \min \quad & -15x_1 - 7x_2 + 28x_3 \\ \text{s. t.} \quad & x_1 + x_2 - 5x_3 \leq 12 \\ & 2x_1 + x_2 - 4x_3 \leq 10 \\ & 9x_1 + 4x_2 - 18x_3 \leq 44 \\ & x_i \geq 0 \end{aligned}$$

25. Aufgabe **(10 Punkte)**

Kann eine Spalte, die bei einem Pivotschritt des Simplexalgorithmus aus der Basis herausgekommen ist, beim nächsten Schritt wieder in die Basis kommen?

Programmieraufgabe, Teil II **(40 Punkte)**

Dieser zweite Teil der Programmieraufgabe ist bis zum **Freitag, 12. Dezember**, zu bearbeiten. Bitte fangen Sie rechtzeitig mit der Bearbeitung an.

Aufgabe: Implementieren Sie den in der Vorlesung eingeführten Simplex-Algorithmus. Beachten Sie dabei die folgenden Hinweise.

1. Zur **Speicherung der Tableaudaten** während des Ablaufs des Algorithmus können Sie eine Datenstruktur verwenden, die in den Dateien `/home/lop/src/lp.c` und `/home/lop/src/lp.h` zur Verfügung gestellt wird. (Lesen Sie die Erläuterungen zu den Strukturelementen und den Funktionen, die in letzterer Datei stehen.)

2. Zum **Einlesen des Problems aus einer MPS-Datei** können Sie entweder Ihr Programm aus Aufgabenteil I erweitern und dann verwenden, oder Sie benutzen den in den Dateien `/home/lop/src/mps.c` und `/home/lop/src/mps.h` zur Verfügung gestellten MPS-Parser. Diesen benutzen Sie so, wie es in der Datei `/home/lop/src/main.c` gezeigt wird.

3. Die folgenden **vereinfachenden Annahmen** dürfen Sie vorerst treffen:

   (a) Die Sections `BOUNDS` und `RANGES` kommen nicht vor. Dadurch sind alle Variablen nichtnegativ und haben keine obere Schranke.

   (Später werden wir durch Einsatz der *Upper Bound Technique* auch Probleme mit `BOUNDS` zulassen.)

   (b) Das Problem liegt bereits in der folgenden Basisdarstellungsform vor:

   $$\begin{aligned} &\min\ \mathbf{c}^\top \mathbf{x}_N \\ &\text{s. t. } \mathbf{A}\mathbf{x}_N + \mathbf{x}_B = \mathbf{b}, \\ &\qquad \mathbf{x}_N \geq \mathbf{0}, \quad \mathbf{x}_B \geq \mathbf{0} \end{aligned} \tag{$*$}$$

   Im MPS-Format werde diese Form wie folgt eingegeben. Setze $\bar{\mathbf{x}}_B = \mathbf{b} - \mathbf{x}_B$. Dann läßt sich $(*)$ schreiben als

   $$\mathbf{A}\mathbf{x}_N = \bar{\mathbf{x}}_B \leq \mathbf{b}.$$

   Die Basisvariablen treten also nicht direkt auf; dafür sind die $\bar{\mathbf{x}}_B$ die Hilfsvariablen für „$\leq$“-Zeilen. Im MPS-Format gebe es also nur „$\leq$“-Zeilen (also `L`-Zeilen) sowie eine `N`-Zeile für die Zielfunktion.

4. Es genügt, einfache Pivotregeln (für Auswahl von Pivotzeile und Pivotspalte) zu verwenden. Sie brauchen sich nicht um evtl. Ausartung und Kreiseln zu kümmern.

5. Zum Compilieren von C-Programmen, die aus mehreren Dateien bestehen, benutzen Sie z. B. den Befehl

```
cc -g -o PROGRAMM QUELLTEXT1.c QUELLTEXT2.c QUELLTEXT3.c
```