Universität Dortmund **SS 2006**
Fachbereich Mathematik, Lehrstuhl V
Numerische Algorithmen / Numerische lineare Algebra (Lineare Optimierung)

Prof. Dr. Martin Skutella
Ronald Koch, Alexia Weber

6. Übungsblatt: Numerische Algorithmen / Numerische Lineare Algebra (Lineare Optimierung)

Abgabe: 17.05.2006, 10:00 Uhr (Briefkasten)

Aufgabe 18 (4 + 6 Punkte)
Betrachtet folgendes Zuordnungsproblem: Gegeben sind n verschiedene Flüssigkeiten und m verschiedene Behälter. Für jede Art Flüssigkeit $i \in \{1,\ldots,n\}$ ist eine Menge an Behältern $S_i \subseteq \{1,\ldots,m\}$ vorgegeben, die die Flüssigkeit aufnehmen können. Ein Behälter kann mehrere Flüssigkeiten gleichzeitig aufnehmen. Von jeder Flüssigkeit ist eine Volumeneinheit vorhanden und jeder Behälter besitzt eine Größe von einer Volumeneinheit. Ziel ist, die Flüssigkeiten so auf die Behälter zu verteilen, dass das Gesamtvolumen aller sich in den Behältern befindenen Flüssigkeiten maximiert wird. Natürlich wird es im Allgemeinen nicht möglich sein alle n verschiedenen Füssigkeiten in die m Behälter zu füllen.

a) Formuliert das Problem als lineares Programm.

b) Stellt das duale Problem auf und versucht dieses anschaulich zu interpretieren.

Aufgabe 19 (4 Punkte)
Wendet die Fourier–Motzkin-Elimination an, um zu entscheiden, ob das folgende Ungleichungssystem ein zulässige Lösung besitzt:

$$\begin{aligned} 2x_1 - x_2 &\leq 1 && (1)\\ x_1 + x_2 &\leq -2 && (2)\\ -3x_1 - 3x_2 &\leq 2 && (3)\\ -x_1 - 3x_2 &\leq -3 && (4)\\ x_2 &\leq 2 && (5)\\ & && (6) \end{aligned}$$

Führt dazu die Fourier–Motzkin-Elimination bis zum Schluss durch, indem Ihr erst x_1 und dann x_2 eliminiert.

Aufgabe 20 (2 + 4 Punkte)
Beweist folgende Alternativsätze:

a) $(\exists x : Ax = c) \quad \dot{\vee} \quad (\exists y : A^t y = 0, c^t y = 1)$

b) $(\exists x : Ax \leq 0, x \neq 0) \quad \dot{\vee} \quad (\forall c : \exists y : y^t A = c, y \leq 0)$

Aufgabe 21 (10 Punkte)

Sei $\epsilon \in (0, \frac{1}{2})$. Betrachtet das folgende lineare Programm in n Variablen und $2n$ Ungleichungen:

$$\begin{array}{lll} \max & x_n & \\ \text{s.t.} & 0 \leq x_1 \leq 1 & \\ & \epsilon x_{i-1} \leq x_i \leq 1 - \epsilon x_{i-1} & \forall i = 2, \ldots, n \\ & x \geq 0 & \end{array}$$

Zeigt, dass der Simplex–Algorithmus mit Blands Pivotregel $2^n - 1$ Iterationen benötigt. Blands Regel nimmt beim Basiswechsel immer diejenige Nichtbasisvariable (von den zulässigen) in die Basis auf, die den kleinsten Index hat und es verlässt diejenige Basisvariable (von den zulässigen) die Basis, die ebenfalls den kleinsten Index hat. (Hinweis: Benutzt vollständige Induktion über n und überlegt Euch eine gute Induktionsbehauptung, mit der Ihr zum Ziel kommt.)

Programmieraufgabe 1, Teil 4

Bindet folgende Pivotregeln in Euer Programm ein: Blands Regel, Größte–Reduzierte–Kosten und Größter–Fortschritt. Nähere Informationen dazu findet Ihr auf der Homepage dieser Vorlesung.