Technische Universität Chemnitz
Prof. Dr. C. Helmberg, A. Lau

Chemnitz, 26.10.2009

Optimierung für Nichtmathematiker
Übung 3

1. Einführung in die Modellierungssprache AMPL, vgl. Übung 2
   Verwendung des NEOS-Solvers: `http://neos.mcs.anl.gov/neos/solvers/index.html`
   Transportproblem: Gegeben sind m Lager, wobei jedes k_i Waren enthält, und n Supermärkte, die s_j Waren benötigen. Durch die Transporte zwischen Lager und Supermarkt entstehen Kosten pro Einheit tranportiertes Gut, welche auch von Start und Ziel abhängen. Gesucht sind die optimalen Transporte, so dass alle Bedarfe gedeckt werden und kein Lager mehr liefert, als es Kapazität hat.

   Modell `transport.mod`:

```
set Lager;
set Supermarkt;

param Kosten{ i in Lager, j in Supermarkt };
param Lager_kap{ i in Lager };
param Supermarktbedarf{ j in Supermarkt };

var x { i in Lager, j in Supermarkt } integer;

minimize Gesamtkosten: sum { i in Lager, j in Supermarkt } Kosten[i,j]*x[i,j];

subject to Lagerkapazitaet { i in Lager }:
        sum{ j in Supermarkt } x[i,j] <= Lager_kap[i];
subject to Bedarf_decken { j in Supermarkt }:
        sum{ i in Lager } x[i,j] = Supermarktbedarf[j];

subject to Schranken { i in Lager, j in Supermarkt }: 0 <= x[i,j];
```

   Beispieldaten `transport.dat`:

```
set Lager := L1 L2 L3;
param Lager_kap:= L1 300 L2 300 L3 200;
set Supermarkt := S1 S2 S3 S4;
param Supermarktbedarf := S1 100 S2 200 S3 250 S4 150;
param Kosten: S1 S2 S3 S4 :=
L1 100 150 50 200
L2 80 200 130 160
L3 120 160 70 100;
```

   Neu:

   - Es kann auch mehrdimensionale Indexmengen geben, z.B. `var x { i in X, j in Y }`, Zugriff auf Element $x_{i,j}$: `x[i,j]`

   Lösen Sie das Transportproblem mithilfe des NEOS-Solvers – verwenden Sie MINTO. Nutzen Sie dazu die folgende Kommando-Datei `transport.cmd`:

```
solve;
display x, Gesamtkosten;
```

Was passiert mit der Lösung, wenn man die Ganzzahligkeit der Variablen nicht fordert? Stimmt das auch in Aufgabe 2 aus Übung 2 (Rucksackproblem)?

2. **Wiederholung Simplex**
Implementieren Sie den (primalen) Simplexalgorithmus aus der Vorlesung. Nutzen Sie dazu die Matlab-Datei `simplex.m` und ergänzen Sie die Funktion
`function [x,basis] = simplex(A,b,c,basis,pricing)`. Dabei sind `A,b,c` die Daten des linearen Programms

$$\begin{array}{ll} \min & c^T x \\ \text{s.t.} & Ax = b \\ & x \geq 0 \end{array}$$

und `basis` ist eine zulässige Startbasis. Mit `pricing` können Sie die eine Auswahl zwischen den Strategien *negativste reduzierte Kosten* und *Regel von Bland* treffen.

Lösen Sie damit das Mozartproblem aus der Vorlesung und testen Sie die folgenden Eingangsdaten

$$A = \begin{pmatrix} 1/4 & -8 & -1 & 9 & 1 & 0 & 0 \\ 1/2 & -12 & -1/2 & 3 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 \end{pmatrix}, \quad b = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix},$$

$c = (-3/4 \quad 20 \quad -1/2 \quad 6 \quad 0 \quad 0 \quad 0)^T$ und Startbasis $\{5, 6, 7\}$. Untersuchen Sie dabei das Verhalten der zwei unterschiedlichen Auswahlstrategien.

3. **Lösung des Rucksackproblem mit dynamischer Programmierung**
Implementieren Sie mit Matlab einen Löser für das Rucksackproblem aus der zweiten Vorlesung. Nutzen Sie dabei das Konzept der dynamischen Programmierung. Testen Sie Ihr Programm am Beispiel aus der Vorlesung und am Beispiel

$$b_0 = 35,\ b = (5\ \ 4\ \ 7\ \ 8\ \ 5\ \ 3\ \ 4\ \ 4\ \ 6\ \ 8),\ \hat{y} = (10\ \ 9\ \ 8\ \ 7\ \ 6\ \ 5\ \ 4\ \ 3\ \ 2\ \ 1).$$

Wiederholung:

geg.: m Objekte mit Gewichten b_i und Wert $\hat{y}_i \geq 0$ $(i = 1, \ldots, m)$ (b und $\hat{y}$ in Vektorform) sowie ein Maximalgewicht b_0 (Beachte: Objekte dürfen mehrfach gewählt werden.)

ges.: Auswahl der Objekte, sodass Gesamtwert maximal wird und Maximalgewicht nicht überschritten wird

$$\max\ \hat{y}^T s \ \text{ s.t. } b^T s \leq b_0,\ s \in \mathbb{N}_0^m$$

Rekursion: Erstelle Matrix F mit den Zeilen $\bar{b} = 0, \ldots, b_0$ und Spalten $\bar{k} = 0, \ldots, m$.
Anfangswert: $F(0, 0) := 0$ und $F(\bar{b}, \bar{k}) := -\infty$ für $(-\bar{b}) \in \mathbb{N}, \bar{k} = 0, \ldots, m$
Rekursion:

$$F(\bar{b}, \bar{k}) = \max\{F(\bar{b}, \bar{k} - 1), F(\bar{b} - b_{\bar{k}}, \bar{k}) + \hat{y}_{\bar{k}}\}, \bar{b} = 1, \ldots, b_0,\ \bar{k} = 1, \ldots, m$$

Ausgabe: $F(b_0, m)$