

Übung zu
Optimierung II
SS 2006
Blatt 3

Aufgabe 3: (Programmieraufgabe)

- a) Schreibe ein Generatorprogramm, das bei Eingabe von $n, m \in \mathbb{N}$ ein 2-Personen Matrixspiel mit zufällig erzeugten Matrixeinträgen $A_{i,j} \in \mathbb{N}$, $B_{i,j} \in \mathbb{N}$ generiert.
- b) Betrachte das folgende Spiel $\text{HANDEL}(k)$:

2 Spieler spielen ein Hinspiel und ein Rückspiel, in denen jeweils k Euro aufgeteilt werden. Im Hinspiel bietet Spieler 1 dem Spieler 2 einen Anteil von x_1 Euro, $x_1 \in \{0, \dots, k\}$. Spieler 2 vergleicht die angebotenen x_1 Euro mit seinem gewählten Mindestanteil $y_2 \in \{0, \dots, k\}$. Ist $x_1 \geq y_2$, so akzeptiert er: Spieler 1 erhält dann $k - x_1$ Euro, Spieler 2 erhält x_1 Euro. Ist $x_1 < y_2$ so akzeptiert Spieler 2 nicht und die k Euro verfallen. Im Rückspiel tauschen Spieler 1 und Spieler 2 die Rollen, d.h. Spieler 2 bietet Spieler 1 jetzt x_2 Euro an und Spieler 1 vergleicht mit seiner Akzeptanzrate y_1 .

Beide Spieler müssen ihre Raten x_i, y_i $i = 1, 2$ vor Beginn des Hinspiels verschlossen hinterlegen. Beide Spieler versuchen ihren Gewinn zu maximieren.

Modelliere HANDEL als 2-Personen Matrixspiel und erweitere den Generator aus a) so, daß er bei Eingabe $k \in \mathbb{N}$ das entsprechende 2-Personen Matrixspiel erzeugt.

- c) Implementiere den Lemke-Howson Algorithmus (Tableauversion) für 2-Personen Matrixspiele, so daß er die in a) erzeugten Spiele als Eingabe lesen kann.

Der Algorithmus soll für seinen gewählten Suchpfad die Bedingung für degenerierte Spiele überprüfen, und im Falle eines degenerierten Spieles eine Warnung ausgeben.

Die Ausgabe des Lemke-Howson Algorithmus soll ein (gemischtes) Nash Equilibrium $(x, y) \in [0, 1]^m \times [0, 1]^n$, die Nutzen $u_1(x, y)$ und $u_2(x, y)$, sowie die Anzahl der Pivot-schritte zur Berechnung von (x, y) sein.

- d) Teste deinen Algorithmus auf allen 2-Personen Matrixspielen der Vorlesung, sowie auf dem Spiel $\text{HANDEL}(k)$.

Freiwillig: Der Lemke-Howson Algorithmus gilt als numerisch sehr empfindlich. Daher ist die Implementierung mit einer rationalen Arithmetik zu empfehlen, z.B. mit der gmp-Library (<http://www.swox.com/gmp/>), die auf den Rechnern der Uni implementiert ist.