technische universität dortmund

Christian Sohler, Melanie Schmidt
Lehrstuhl 2, Fakultät für Informatik
Wintersemester 2009/2010

AuD Übung – Blatt 1

Ausgabedatum: 14.10.2009 — Abgabedatum: 21.10.2009

Abgabe:

Die Abgabe der Bearbeitungen erfolgt in den Übungen oder im Anschluss an die Vorlesung. Alternativ kann auch per Mail abgegeben werden.

Scheine:

Leistungsnachweise können durch eine mündliche Prüfung erworben werden.

Aufgabe 1.1: Lineare Programme drehen

Es sei ein zweidimensionales lineares Programm

$$\begin{array}{rl} \max & c_1x_1 + c_2x_2 \\ \text{unter} & A^t x \leq b \end{array}$$

gegeben (A ist eine $n \times 2$ Matrix, $c \in R^n$ ein Spaltenvektor, gesucht ist ein Vektor $x = (x_1, x_2) \in R^2$). Beschreibe, wie man dieses lineare Programm verändern muss, damit die Zielfunktion "nach unten" zeigt, d.h. dass die Zielfunktion $-c_3x_2$ für ein $c_3 \in \mathbb{R}$ lautet. Berechne, wie das gedrehte lineare Programm zu dem folgenden Beispielprogramm

$$\begin{array}{rr} \max & x_1 + x_2 \\ \text{unter} & -x_2 \geq -4 \\ & x_1 \geq 1 \\ & -x_1 + x_2 \geq -2 \\ & x_1 + x_2 \geq 2 \end{array}$$

aussieht. Zeichne sowohl das ursprüngliche als auch das gedrehte LP einmal in ein Koordinatensystem.

Wie berechnet man aus einer Lösung des abgewandelten Programms eine Lösung für das ursprüngliche lineare Programm?

Aufgabe 1.2: Konvexität

Zeige:

1. Ein Halbraum $H = \{x \in \mathbb{R}^d \mid a^t x \leq d\}$ für gegebenes $a \in \mathbb{R}^d, d \in \mathbb{R}$ ist immer konvex.
2. Sind $M_1, M_2, \ldots, M_l$ konvexe Mengen im $\mathbb{R}^d$, so ist auch der Schnitt dieser Mengen konvex.
3. Der zulässige Bereich eines linearen Programms ist konvex.

Aufgabe 1.3: Iterativer Algorithmus für zweidimensionale LPs

1. Welchen Nutzen hat Randomisierung für den iterativen Algorithmus zur Lösung von linearen Programmen? Nenne ein weiteres Beispiel für einen randomisierten Algorithmus (aus DAP2), auf den dieses zutrifft.

2. Gib jeweils eine mögliche Eingabe an, für die ...

   - ... der deterministische Algorithmus zur Lösung von zweidimensionalen LPs eine Worst-Case-Laufzeit von $O(n^2)$ hat.
   - ... sowohl der randomisierte *als auch* der deterministische Algorithmus zur Lösung von zweidimensionalen LPs eine Worst-Case-Laufzeit von $O(n)$ erreicht.

   Wir gehen hier nicht davon aus, dass die Eingabe in allgemeiner Lage ist.

Aufgabe 1.4: Iterativer Algorithmus für zweidimensionale LPs

Zeige (mit Wissen aus der Vorlesung), dass $(x_1, x_2) = (1, 0)$ eine optimale Lösung des folgenden linearen Programms ist:

$$\begin{array}{rlll}
\text{maximiere} & x_1 & -10x_2 & \\
\text{unter} & 7x_1 & -x_2 & \leq 49 \\
 & x_1 & -x_2 & \leq 5 \\
 & x_1 & -2x_2 & \leq 1 \\
 & -2x_1 & -20x_2 & \leq 50 \\
 & -4x_1 & -x_2 & \leq 7 \\
 & x_1 & -10x_2 & \leq 20 \\
 & -x_1 & -3x_2 & \leq -1 \\
 & 2x_1 & -2x_2 & \leq 5 \\
 & -\frac{1}{8}x_1 & +x_2 & \leq 4 \\
 & x_1 & +x_2 & \leq 7
\end{array}$$

Aufgabe 1.5: Zufällige Permutationen

Entwirf einen Algorithmus, der eine zufällige Permutation der Zahlen 1 bis n ausgibt ($n > 0$). Geh dabei von einem Array mit n Einträgen aus, das zu Beginn die Zahlen in aufsteigender Reihenfolge enthält und nach Ablauf des Algorithmus eine zufällige Permutation enthalten soll. Dir steht eine Funktion *rand(n)* zur Verfügung, die Dir in konstanter Zeit eine zufällige natürliche Zahl z mit $1 \leq z \leq n$ liefert. Begründe, dass der Algorithmus korrekt arbeitet, d.h. dass jede Permutation der Zahlen mit *gleicher Wahrscheinlichkeit* auftritt.