Mathematische Optimierung SS 2012

11. Übungsblatt

80. Betrachten Sie das folgende binäre lineare Optimierungsproblem mit $n+1$ Variablen, $n \in \mathbb{N}$.

$$\begin{array}{ll} \max & -x_{n+1} \\ \text{udNb.} & \\ & 2x_1 + 2x_2 + \ldots + 2x_n + x_{n+1} = n \\ & x \in \{0,1\}^{n+1} \end{array}$$

Zeigen Sie, dass für n ungerade jedes Branch-and-Bound Verfahren, das in jedem Knoten des Branch-and-Bound-Baumes den optimalen Wert der jeweiligen linearen Relaxation als obere Schranke verwendet, eine exponentielle Anzahl von Knoten untersuchen muss.

81. Formulieren Sie die folgende Optimierungsaufgabe als gemischt-ganzzahliges lineares Programm:

$$\min f_1(x_1) + f_2(x_2)$$

s.t.

(a) Entweder gilt $x_1 \geq 3$ oder $x_2 \geq 3$

(b) Mindestens eine der folgenden Restriktionen ist erfüllt:

$$\begin{array}{rcrcl} 2x_1 & + & x_2 & \geq & 7 \\ x_1 & + & x_2 & \geq & 5 \\ x_1 & + & 2x_2 & \geq & 7 \end{array}$$

(c) Es gilt $|x_1 - x_2| = 0$ oder $|x_1 - x_2| = 3$ oder $|x_1 - x_2| = 6$

(d) $x_1 \geq 0$, $x_2 \geq 0$

wobei

$$f_1(x_1) = \begin{cases} 7 + 5x_1 & \text{für } x_1 > 0 \\ 0 & \text{sonst} \end{cases}$$

und

$$f_2(x_2) = \begin{cases} 5 + 6x_2 & \text{für } x_2 > 0 \\ 0 & \text{sonst} \end{cases}$$

82. Bestimmen Sie die optimale Multiplizierreihenfolge zur Berechnung der Matrix $M_1M_2M_3M_4$, wobei M_1 eine 10×5 Matrix, M_2 eine 5×100 Matrix, M_3 eine 100×3 und M_4 eine 3×40 Matrix ist.

83. Formulieren Sie die folgende Aufgabe als ganzzahliges lineares Optimierungsproblem:

Es sind 5 Teile A, B, C, D und E auf einer Maschine herzustellen, die jeweils nur einen Teil auf einmal bearbeiten kann. Die Zeit zur Herstellung der Teile setzt sich aus zwei Komponenten zusammen, der Zeit die benötigt wird, um die Maschine auf die Bearbeitung dieses Teils vorzubereiten (Rüstzeit) und der Bearbeitungszeit des Teils. Teil A erfordert eine Bearbeitungszeit von 3 Stunden, Teil B von 2,5 Stunden, Teil C von 9 Stunden, Teil D von 3,7 Stunden und Teil E von 5 Stunden. Die Rüstzeiten hängen von der Bearbeitungsreihenfolge ab. Konkret hängt die Rüstzeit für Teil X vom Typ des davor auf der Maschine behandelten Teils ab. Die folgende Tabelle enthält die Matrix der Rüstzeiten (beachten Sie, daß diese Matrix nicht symmetrisch ist):

	Vorgängerteil				
Nachfolgerteil	A	B	C	D	E
A	0	2	5	3	7
B	3	0	2	4	4
C	2	1	0	4	5
D	1	3	3	0	5
E	5	5	2	2	0

Die 5 Teile werden zur Herstellung eines Gegenstandes benötigt, der sofort im Anschluß an die Fertigung der Teile durch Zusammenbau der Teile gefertigt werden muß. Aus diesem Grund erfolgt die Teilfertigung auf der Maschine zyklisch, d.h. es wird stets dieselbe Fertigungsreihenfolge der 5 Teile durchlaufen. (Dies ist insofern relevant als somit der letzte Teil dieser Reihenfolge den Vorgänger für den ersten Teil darstellt, wenn es um die Ermittlung der Rüstzeiten geht). Das Ziel ist es eine Produktionsreihenfolge zu finden, die die Gesamtzykluszeit (Summe aus Bearbeitungszeiten und Rüstzeiten) minimiert.

84. (Das Transportparadoxon - Sie können auch AMPL verwenden)

Betrachten Sie das Transportproblem mit der Kostenmatrix

$$C = \begin{pmatrix} 30 & 11 & 5 & 35 & 8 & 29 \\ 2 & 5 & 2 & 5 & 1 & 9 \\ 35 & 20 & 6 & 40 & 8 & 33 \\ 19 & 2 & 4 & 30 & 10 & 25 \end{pmatrix},$$

dem Angebotsvektor $a = (30,\ 10+\delta,\ 45,\ 30)$ und dem Bedarfsvektor $b = (25,\ 20+\delta,\ 6,\ 7,\ 22,\ 35)$. (Hierbei ist δ ein Parameter aus dem Intervall $[0, 22]$.)

(a) Lösen Sie das obige Transportproblem in Abhängigkeit von δ und zeigen Sie, daß der Zielfunktionswert mit wachsendem δ streng monoton abnimmt.

(b) Welche weiteren Eigenschaften besitzt der Zielfunktionswert, wenn er als Funktion des Parameters δ betrachtet wird, in diesem Beispiel?

(c) Inwiefern gelten die in (b) festgestellten Eigenschaften auf allgemeinerer Basis? Versuchen Sie diesbezüglich eine möglichst allgemeine Aussage aufzustellen und diese dann zu beweisen.

(d) Finden Sie ein einfacheres Beispiel, in dem ein ähnlich paradoxes Verhalten auftritt wie in (a). Versuchen Sie das zugrundeliegende Phänomen zu interpretieren.

85. Entwerfen Sie ein dynamisches Programmierverfahren für das untenstehende Problem:

$$\max \left\{ \sum_{j=1}^{n} a_j y_j : y_i + y_{i+1} \le b_i\,,\ \text{für } i = 1, 2, \ldots, n-1, y \in \mathbb{R}^n \right\}$$

wobei $n \in \mathbb{N}$, $a = (a_1, a_2, \ldots, a_n)^t \in \mathbb{Z}^n$ und $b = (b_1, b_2, \ldots, b_{n-1})^t \in \mathbb{Z}^{n-1}$.