Diskrete Optimierung
11. Übungsblatt

TECHNISCHE UNIVERSITÄT DARMSTADT

Fachbereich Mathematik
Prof. Dr. Stefan Ulbrich
Dipl.-Math. Madeline Lips

SoSe 2013
25. und 27. Juni 2013

Gruppenübung

Aufgabe G1 (Branch & Bound)
Benutzen Sie Ihren Übungsleiter als LP-Orakel, um das folgende Problem mit dem Branch & Bound-Algorithmus zu lösen:

$$\max\{c^T x \mid Ax \le b, x \in \mathbb{Z}^3\}$$

mit

$$A = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ -4 & -7 & 5 \\ -3 & 7 & -4 \\ 1 & 0 & -7 \\ -4 & 3 & -1 \end{pmatrix}, \qquad b = \begin{pmatrix} 5 \\ 5 \\ 0 \\ -1 \\ 5 \\ -3 \end{pmatrix} \qquad \text{und} \qquad c = \begin{pmatrix} 12 \\ -9 \\ 4 \end{pmatrix}.$$

Branchen Sie dabei immer auf der ersten nichtganzzahligen Variable.

Aufgabe G2 (Branch & Bound)
Betrachten Sie für $a_1, \dots, a_n, w_1, \dots, w_n > 0$ das 0/1-Knapsackproblem

$$\max\left\{\sum_{i=1}^n w_i x_i \mid \sum_{i=1}^n a_i x_i \le b, x_i \in \{0,1\}\right\}.$$

(a) Seien $\frac{w_1}{a_1} \ge \dots \ge \frac{w_n}{a_n}$ und $k \in \{1, \dots, n\}$ mit $a_1 + \dots + a_k \le b$ und $a_1 + \dots + a_{k+1} > b$. Zeigen Sie, dass $x \in [0,1]^n$ mit

$$x_i = \begin{cases} 1, & \text{falls } i \in \{1, \dots, k\}, \\ (b - a_1 - \dots - a_k)/a_{k+1}, & \text{falls } i = k+1, \\ 0, & \text{sonst.} \end{cases}$$

eine Optimallösung für das relaxierte 0/1-Knapsackproblem darstellt.

(b) Geben Sie mithilfe von ((a)) ein Branch & Bound-Verfahren für das 0/1-Knapsackproblem an.

(c) Berechnen Sie per Branch & Bound für $n = 7$, $b = 35$ und die in Tabelle 1 angegeben Werte für a_i und w_i eine Optimallösung für das 0/1-Knapsackproblem.

Objekt i	Gewicht a_i	Wert w_i
1	3	12
2	4	12
3	3	9
4	3	15
5	15	90
6	13	26
7	16	112

Tabelle 1: Die Werte a_i und w_i.

Aufgabe G3 (Coverungleichungen)
Definition:
Gegeben sei das Rucksackpolytop $P_{(N,a,b)} := \mathrm{conv}\{x \in \{0,1\}^N \mid \sum_{i\in N} a_i x_i \leq b\}$, $N = \{1,...,n\}$. *O.B.d.A. sei* $a_j \leq b$ *für alle* $j \in N$.

$C \subseteq N$ *heißt* **Cover**, *wenn* $\sum_{j\in C} a_j > b$. C *heißt* **minimal**, *falls* $C \backslash \{j\}$ *kein Cover ist für alle* $j \in C$. *Für alle Cover* $C \subseteq N$ *heißt* $\sum_{j\in C} x_j \leq |C| - 1$ **Coverungleichung**.

(a) Sei $R := \{x \in \{0,1\}^7 \mid 8x_1 + 9x_2 + 6x_3 + 6x_4 + 7x_5 + 5x_6 + x_7 \leq 22\}$. Geben Sie mindestens drei minimale Coverungleichungen für R an.

(b) Beweisen Sie folgenden Satz:
Sei $C \subseteq N$ *ein Cover für* $P_{(N,a,b)}$, *dann ist* $\sum_{j\in C} x_j \leq |C| - 1$ *gültig für* $P_{(N,a,b)}$ *und facettendefinierend für* $P_{(C,a,b)}$, *falls* C *minimal ist.*

Hausübung

Aufgabe H1 (Branch & Bound) (5 Punkte)
Lösen Sie folgendes Optimierungsproblem mittels Branch & Bound und skizzieren Sie den Branch & Bound-Baum. Zur Lösung der jeweils auftretenden LP-Relaxierungen darf ein LP-Solver benutzt werden.

$$
\begin{array}{lrcl}
\max & 4x_1 - x_2 & & \\
\text{s.t.} & 7x_1 - 2x_2 & \leq & 14 \\
 & x_2 & \leq & 3 \\
 & 2x_1 - 2x_2 & \leq & 3 \\
 & x_1, x_2 & \geq & 0 \\
 & x_1, x_2 & \in & \mathbb{Z}
\end{array}
$$

Aufgabe H2 (Branch & Bound) (5 Punkte)
Das binäre Programm

$$
\begin{array}{lrcll}
\min & \sum_{k=1}^{n} \delta_k & & & \\
\text{s.t.} & x_{ik} + x_{jk} & \leq & \delta_k & \text{für alle } \{i,j\} \in E \text{ und } k \in \{1,\dots,n\} \\
 & \sum_{k=1}^{n} x_{ik} & = & 1 & \text{für alle } i \in V \\
 & x & \in & \{0,1\}^{n\times n} & \\
 & \delta & \in & \{0,1\}^n &
\end{array}
$$

liefert die Färbungszahl des Graphen $G = (V,E)$. Eine *zulässige Färbung* eines Graphen ist eine Abbildung $f : V \to \mathbb{N}$ mit der Eigenschaft, dass je zwei Knoten, die durch eine Kante verbunden sind, unterschiedliche Farben zugeordnet werden, das heißt, dass für alle Kanten $\{i,j\} \in E$ die Bedingung $f(i) \neq f(j)$ erfüllt ist. Die *Färbungszahl* ist die kleinstmögliche Anzahl von Farben, für die es eine zulässige Färbung gibt.

Warum ist diese Formulierung ungünstig um mit dem Branch & Bound-Verfahren gelöst zu werden?
Hinweis: Welche Auswirkungen hat die Symmetrie des Problems auf den Branch & Bound-Baum?

Aufgabe H3 (Geliftete Coverungleichungen) (5 Punkte)
In Aufgabe G3 haben Sie gezeigt, dass minimale Coverungleichungen facettendefinierend für $P_{(C,a,b)}$ sind. Jedoch gilt diese Aussage nicht immer für $P_{(N,a,b)}$. Um diese Ungleichungen noch zu verschärfen gibt es die Methode des *Liftens* (siehe Algorithmus 1).

Gegeben sei das Rucksackproblem

$$
\begin{aligned}
\max\ & 3x_1 + 2x_2 + 5x_3 + 4x_4 + x_5 \\
\text{s.t. } & x_1 + 4x_2 + x_3 + 4x_4 + 2x_5 \leq 5 \\
& x \in \{0,1\}^5.
\end{aligned}
$$

Bestimmen Sie die gelifte Coverungleichung für das Cover $C = \{1,2,3\}$ mit Hilfe von Algorithmus 1.

Algorithm 1 Algorithmus zum Liften von Coverungleichungen

INPUT: Die Daten N, a, b für $X = \{x \in \{0,1\}^N \mid \sum_{j \in N} a_j x_j \le b\}$ und ein minimales Cover C.

OUTPUT: β_j für $j \in N \setminus C$, sodass $\sum_{j \in N \setminus C} \beta_j x_j + \sum_{j \in C} x_j \le |C| - 1$ gültig für X.

1: Seien $j_1, \dots, j_r$ die Indizes aus $N \setminus C$.

2: **for** $t \leftarrow 1, \dots, r$ **do**

3: Die gültige Ungleichung $\sum_{i=1}^{t-1} \beta_{j_i} x_{j_i} + \sum_{j \in C} x_j \le |C| - 1$ haben wir bereits.

4: Um die größten Werte β_{j_t} zu bestimmen, für die $\beta_{j_t} x_{j_t} + \sum_{i=1}^{t-1} \beta_{j_i} x_{j_i} + \sum_{j \in C} x_j \le |C| - 1$ gültig ist, lösen wir das folgende Rucksackproblem:

$$\begin{aligned} z_t = \max \; & \sum_{i=1}^{t-1} \beta_{j_i} x_{j_i} + \sum_{j \in C} x_j \\ \text{s.t.} \; & \sum_{i=1}^{t-1} a_{j_i} x_{j_i} + \sum_{j \in C} a_j x_j \le b - a_{j_t} \\ & x \in \{0,1\}^{|C|+t-1} \end{aligned}$$

5: Setze $\beta_{j_t} := |C| - 1 - z_t$.

6: **end for**

7: **return** β_j für $j \in N \setminus C$.