Algorithmenentwurf HA 11

Lukas Brandt: 7011823, Clemens Damke: 7011488, Lukas Giesel: 7011495
7. Juli 2016

Aufgabe 23

a. ILP

$$\begin{aligned} \textbf{minimiere } & c \cdot x \text{ für gegebenes } c = (c_1 \ \cdots \ c_m) \\ \textbf{s. t. } & A \cdot x \geq \underbrace{(1 \ \cdots \ 1)}_{m\text{-mal}}^T, \ x \in \{0,1\}^m \\ A := & \begin{pmatrix} a_{11} & \cdots & a_{1m} \\ \vdots & \ddots & \vdots \\ a_{n1} & \cdots & a_{nm} \end{pmatrix}, \ a_{ij} := \begin{cases} 1 & \textbf{falls } e_i \in S_j \\ 0 & \textbf{sonst} \end{cases} \end{aligned}$$

Minimale Lösungen aus $\{0,1\}^m$ sind identisch zu minimalen Lösungen aus $\mathbb{N}_0^m$, da letztere an keiner Stelle x_i kleiner als erstere sein können und $\forall x_i > 1$ auch $x_i = 1$ eine valide Lösung ist, womit $x \notin \{0,1\}^m$ dann also nicht minimal wäre. $\implies$ obiges 0-1-LP $\equiv$ ILP.

b. Approximationsalgorithmus

1. Löse LP aus (a) reelwertig. Lösung $x' = (x'_1 \ \cdots \ x'_m)^T \in \mathbb{R}^m_{\geq 0}$
2. Berechne Lösung $x := (x_1 \ \cdots \ x_m)^T \in \{0,1\}^m, \ x_i := \begin{cases} 1 & \textbf{falls } x'_i \geq \frac{1}{f} \\ 0 & \textbf{sonst} \end{cases}$

Korrektheit:

$$\begin{aligned} & \forall i \in \{1,\dots,n\} : \sum_{j=1}^m a_{ij} \leq f \ \wedge \ \sum_{j=1}^m a_{ij} \cdot x'_j \geq 1 \\ \implies & \forall i \in \{1,\dots,n\} : \exists j \in \{1,\dots,m\} : a_{ij} = 1 \ \wedge \ x'_j \geq \frac{1}{f} \\ \implies & \forall i \in \{1,\dots,n\} : \exists j \in \{1,\dots,m\} : a_{ij} = 1 \ \wedge \ x_j = 1 \\ \implies & A \cdot x \geq \underbrace{(1 \ \cdots \ 1)}_{m\text{-mal}}^T \end{aligned}$$

Approximationsfaktor:

$$\begin{aligned} & c \cdot x_0 \geq c \cdot x', \ x_0 := \text{optimale ganzzahlige Lösung} \\ & \forall j \in \{1,\dots,m\} : x'_j \cdot f \geq x_j \\ \implies & c \cdot x_0 \cdot f \geq c \cdot x' \cdot f \geq c \cdot x \\ \implies & x \text{ ist max. } f\text{-mal schlechter als die optimale ganzzahlige Lösung.} \end{aligned}$$

Aufgabe 24

$$
\begin{aligned}
A &:= \text{online Algorithmus aus Aufgabenstellung}\\
O &:= \text{optimaler Algorithums}\\
s &:= \text{Eingabesequenz von Gegenständen} = (g_1, \ldots, g_n)\\
w(g) &:= \text{Gewicht des Gegenstands } g\\
a &:= \text{aufsteigende Zeitpunkte zu denen } A \text{ Koffer öffnet} = (a_1 = 1, \ldots, a_k) \in \{1, \ldots, n\}^k\\
p(i) &:= (g_\alpha, \ldots, g_\beta) \text{ mit } \alpha := a_{2i-1},\ \beta := a_{2i+1} - 1\\
&\implies \forall\, i \in \mathbb{N} : p(i) \text{ enthält für zwei sukzessive Koffer alle eingepackten Gegenstände.}\\
&\implies \forall\, p(i) = (g_\alpha, \ldots, g_\beta), i \in \mathbb{N} : \sum_{j=\alpha}^{\beta} w(g_j) > 1, \text{ da sonst kein zweiter Koffer geöffnet würde.}\\
&\implies \forall\, i \in \mathbb{N} : \text{Gegenstände in } p(i) \text{ verursachen in } O \text{ das Öffnen von min. einem Koffer.}\\
&\implies \text{Anzahl von Koffern in } A \text{ ist max. doppelt so groß, wie in } O.\\
&\implies A \text{ ist 2-competitive.}
\end{aligned}
$$