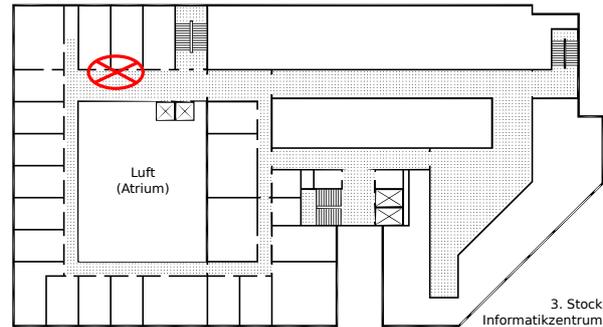


Prof. Dr. Sándor Fekete
Dominik Krupke

Mathematische Methoden der Algorithmik Übungsblatt 2 vom 13. 11. 2018

Die Abgabe der Lösungen zu Blatt 2 ist bis Dienstag, den 27. 11. 2018 um 13:15 Uhr im Hausaufgabenrückgabeschrank der Algorithmik möglich.

Bitte die Blätter vorne deutlich mit eigenem Namen sowie Matrikelnummer versehen und zusammenheften!



Aufgabe 1 (Formen von LPs): Betrachte das folgende lineare Optimierungsproblem:

$$(P) \left\{ \begin{array}{ll} \min & x_1 + x_2 \\ \text{s. t.} & x_1 + 3x_2 \geq 9 \\ & 2x_1 + x_2 \geq 8 \\ & 2x_1 + 5x_2 \leq 40 \\ & x_1 \leq 8 \\ & x_1 \geq 0 \\ & x_2 \geq 0 \end{array} \right.$$

Schreibe das lineare Programm P in den Formen

(a) $\min\{c^T x \mid Ax \leq b, x \geq 0\}$

(b) $\max\{c^T x \mid Ax = b, x \geq 0\}$

(c) $\min\{c^T x \mid Ax \leq b, x \text{ frei}\}$.

Die Matrixschreibweise ist nicht gefordert.

(10 Punkte)

Aufgabe 2 (Praktisches Lösen von Weighted Vertex Cover): Im *Weighted Vertex Cover* haben wir einen Graphen $G(V, E)$ und Knotengewichtung $w : V \rightarrow \mathbb{Z}^+$ gegeben und suchen eine Menge von Knoten C mit minimalen Gesamtgewichts bezüglich w sodass jede Kante e zu einem Knoten aus C inzident ist. Für den Graphen $V = \{v_0, v_1, v_2, v_3, v_4\}$, $E = \{v_0v_2, v_1v_2, v_2v_4, v_1v_3, v_3v_4, v_2v_3\}$, mit den Gewichten $w(v_0) = 2, w(v_1) = 3, w(v_2) = 10, w(v_3) = 11, w(v_4) = 1$ lässt sich das folgende Integer Program in CPLEX-Syntax (Siehe Hinweise auf Webseite) aufstellen:

Minimize

2 v0 + 3 v1 + 10 v2 + 11 v3 + 1 v4

Subject To

v0 + v2 >= 1

v1 + v2 >= 1

v1 + v3 >= 1

v2 + v4 >= 1

v3 + v4 >= 1

v2 + v3 >= 1

Bounds

0 <= v0 <= 1

0 <= v1 <= 1

0 <= v2 <= 1

0 <= v3 <= 1

0 <= v4 <= 1

Integers

v0

v1

v2

v3

v4

End

- Löse das Problem einmal fraktional (d.h. als polynomiell lösbares LP) und einmal integral. Hierfür kannst du den Javascript-Solver auf <http://hgourvest.github.io/glpk.js/> verwenden. Das Problem wird genau dann integral gelöst wenn MIP (für Mixed Integer Programming) aktiviert ist. (Hinweis: Der fraktionale Funktionswert sollte 13.5 sein und der integrale 14)

- Stelle auch die folgenden Graphen als IP dar und gebe die fraktionalen und integralen Lösungen an.

– $G_1(\{v_0, v_1, v_2\}, \{v_0v_1, v_1v_2, v_0v_2\}), w(v_0) = w(v_1) = w(v_2) = 1.$

– $G_2(\{v_0, v_1, v_2, v_3, v_4\}, \{v_0v_2, v_1v_2, v_2v_4, v_1v_3, v_3v_4\}), w(v_0) = 2, w(v_1) = 3, w(v_2) = 10, w(v_3) = 11, w(v_4) = 1$

- Probiere weitere Instanzen aus. Welche Werte nehmen die Variablen im fraktionalen an? Fällt dir eine Besonderheit bei bipartiten Graphen auf?

(3+7+10 Punkte)

Aufgabe 3 (LP-Formulierung eines Optimierungsproblems): Formuliere das folgende Optimierungsproblem als LP: Gegeben n Punkte (x_i, y_i) in der Ebene. Gesucht ist eine Gerade, die das Maximum (nicht die Summe!) der vertikalen Abstände zu den Punkten minimiert. Löse das Problem für $\{(1, 10), (3, 6), (4, 7), (6, 5), (8, 2), (9, 4)\}$ optimal etwa mit <http://hgourvest.github.io/glpk.js/>.

Hinweis: Die meisten Solver nehmen per default $x \geq 0$ an. Um dies zu unterbinden muss 'x free' zu den Bounds hinzugefügt werden. **(15 Punkte)**

Aufgabe 4 (Praktisches Problem): Du willst dir einen neuen Computer bauen. Er soll maximal 700 Euro kosten und die maximale Leistung liefern (Summe der Leistung aller Komponenten). Desweiteren kannst du dein altes Netzteil nur weiterverwenden, wenn der Computer weniger als 350W verbraucht. Ansonsten musst du für 100 Euro ein stärkeres Netzteil kaufen.

Schreibe das Problem als IP auf und löse es mit <http://hgourvest.github.io/glpk.js/>.

- Mainboard:

Konfiguration:	Preis:	Verbrauch:	Leistung:
MB_A	70	20	70
MB_B	50	80	100
MB_C	100	40	100

- CPU:

Konfiguration:	Preis:	Verbrauch:	Leistung:
C_1	100	50	100
C_2	200	75	350
C_3	100	55	90
C_4	150	80	175
C_5	200	60	250

- Kompatibilität:

CPU:	MB_A	MB_B	MB_C
C_1	x		
C_2	x		
C_3		x	x
C_4		x	x
C_5		x	x

- Arbeitsspeicher:

Konfiguration:	Preis:	Verbrauch:	Leistung:
16GB	100	20	75
2x16GB	200	40	150
4x8GB	125	55	150

- 4x8GB wird nur von Mainboard MB_C unterstützt
- Auf Mainboard MB_A mit CPU C_2 ist die Leistung für jeden Arbeitsspeicher um 20% höher.

- Grafikkarte:

Konfiguration:	Preis:	Verbrauch:	Leistung:
G_1	150	150	100
G_2	250	150	250
G_3	400	250	500

(15 Punkte)