Karlsruher Institut für Technologie
Institut für Theoretische Informatik

Prof. Dr. P. Sanders, Dr. S. Gog
L. Hübschle-Schneider, T. Maier

13. Übungsblatt zu Theoretische Grundlagen der Informatik im WS 2015/16

http://algo2.iti.kit.edu/TGI2015.php
{sanders,gog,huebschle,t.maier}@kit.edu

Musterlösungen

Aufgabe 1 (*ILP*, $2+2+3$ *Punkte*)

Gegeben seien die folgenden Probleme:

ILP: Gegeben einer Menge von ganzzahligen Variablen (manchmal Unbekannte genannt) $x_1, \ldots, x_n$, und einer Menge von Constraints $c_1, \ldots, c_m$ (der Form $c_i : t_{i1}x_1 + \cdots + t_{in}x_n \leq t_i$, alternativ $=$ oder $\geq$, $t_x \in \mathbb{Z}$), existiert eine ganzzahlige Belegung für $x_1, \ldots, x_n$ so dass alle Constraints erfüllt sind (Constraints lassen sich häufig verkürzt schreiben, wenn nicht alle Variablen verwendet werden)?

3SAT: Gegeben einer Menge von aussagenlogischen Variablen $(x_1, \ldots, x_n)$, und einer Menge von Klauseln $k_1, \ldots, k_m$ mit je drei Literalen, gibt es eine Belegung der Variablen die alle Klauseln erfüllt?

VERTEX COVER: Gegeben ein ungerichteter Graph $G = (V, E)$ und eine Zahl $k \in \mathbb{N}$, existiert eine Teilmenge $S \subseteq V$ mit $|S| \leq k$ sodass alle Kanten des Graphen inzident zu mindestens einem Knoten aus S sind $(\forall_{e=\{u,v\}\in E}\ u \in S \vee v \in S)$?

MAX CUT: Gegeben ein ungerichteter Graph $G = (V, E)$ und eine Zahl $k \in \mathbb{N}$, existiert eine Teilmenge $S \subseteq V$ sodass mindestens k Kanten im Mengenprodukt von S und $V \setminus S$ liegen?
Formell: $|\{\{u, v\} \in E \mid u \in S \wedge v \notin S\}| \geq k$?

Formulieren Sie für jedes der folgenden Probleme eine Umformung, die für eine gegebene Probleminstanz eine erfüllbarkeitsäquivalente ILP-Instanz erzeugt. Die Laufzeit Ihrer Konstruktion darf nur polynomiell von der Größe der Ursprungsinstanz abhängen.

a) 3SAT

b) VERTEX COVER

c) MAX CUT

Musterlösung:

a) Sei die 3SAT Instanz gegeben durch die Variabeln $V = x_1, \ldots, x_n$ und die Klauseln $K = k_1, \ldots, k_m$. Wir betrachten die Klauseln als Teilmengen der Variabelmenge $(k_i \subset V \cup V_\neg \ \wedge \ |k_i| \leq 3)$.

In der ILP Instanz modellieren wir jede aussagenlogische Variable x_i als eine unbekannte z_i (diese ist 1 genau dann, wenn x_i in einer erfüllenden Belegung `true` ist und sonst 0). Zunächst stellen wir sicher, das jedes z_i nur mit 0 oder mit 1 belegt werden kann:

$$\bigcup_{x_i \in V} z_i \leq 1 \qquad \bigcup_{x_i \in V} z_i \geq 0$$

Nun stellen wir sicher, dass zumindest ein Element jeder Klausel erfüllt ist:

$$\bigcup_{k_i=\{x_a,x_b,x_c\}\in K} z_a + z_b + z_c \geq 1 \quad \text{bzw.}$$

Negierte Variablen z.B. $\neg x_a$ werden durch $1 - z_a$ dargestellt.

$$\bigcup_{k_i=\{x_a,x_b,x_c\}\in K} (1 - z_a) + z_b + z_c \geq 1 \text{ entspricht } \quad - z_a + z_b + z_c \geq 0$$

z.Z. erfüllbarkeits-äquivalent (3SAT $\Leftrightarrow$ ILP)

- $\Rightarrow$ setzt man für jede mit `true` belegte Variable die entsprechende Unbekannte auf 1 (alle anderen auf 0), so sind alle Constraints erfüllt, da in jeder Klausel ein Literal wahr war.
- $\Leftarrow$ da in jedem Constraint mindestens eine Variable 1 sein muss, gibt es auch in der zugehörigen Klausel ein erfülltes Literal.

Die Konstruktion ist offensichtlich polynomiell (2 Constraints pro Variable + 1 Constaint pro Klausel + Alle Constraints haben konstante länge).

b) Sei die VERTEX COVER Instanz gegeben durch den ungerichteten Graph $G = (V, E)$. Für jeden Knoten v_i verwenden wir eine unbekannte x_i, welche 1 ist, genau dann wenn v_i zu dem berechneten Vertex Cover gehört (und Null sonst).

$$\bigcup_{v_i\in V} x_i \leq 1 \qquad \bigcup_{v_i\in V} x_i \geq 0$$

Nun stellen wir sicher, dass jede Kante einen inzidenten Knoten hat, der im Vertex Cover liegt.

$$\bigcup_{e_i=\{v_a,v_b\}\in E} x_a + x_b \geq 1$$

Zuletzt müssen wir die Größe des Vertex Covers beschränken.

$$\sum_{v_i\in V} x_i \leq k$$

z.Z. erfüllbarkeits-äquivalent (VERTEX COVER $\Leftrightarrow$ ILP)

- $\Rightarrow$ gegeben einem Vertex Cover lässt sich eine erfüllende Belegung finden, indem für jeden Knoten v_i die zugehörige Variable auf 1 gesetzt wird (die anderen auf 0). Dies erfüllt alle Constraints, da durch die Definition eines Vertex Covers erzwungen wird, dass am jeder Kante mindestens ein Knoten im Vertex Cover vorhanden ist.
- $\Leftarrow$ gegeben einer erfüllenden Belegung ist leicht ersichtlich, dass die Menge $S = \{v_i \in V \mid x_i = 1\}$ ein Vertex Cover ist. Auch die Größe des Vertex Covers kann k nicht überschreiten.

Die Konstruktion ist offensichtlich polynomiell (2 Constraints pro Knoten + 1 Constraint pro Kante + ein Constraint für die Grenze, alle maximal lineare Länge).

c) Sei eine MAX CUT Instanz gegeben durch den ungerichteten Graph $G = (V, E)$ (wobei V und E geordnete Mengen sind). Für jeden Knoten v_i verwenden wir eine unbekannte x_i, desweiteren konstruieren wir eine Unbekannte z_i für jede Kante e_i. Wir verwenden x_i um zu codieren in welcher Menge der Knoten v_i liegt ($S \Rightarrow 1$ oder $V \setminus S \Rightarrow 0$), und z_i um zu codieren ob die Kante e_i im Schnitt der beiden Mengen liegt.

Jeder Knoten liegt entweder in S oder in $V \setminus S$.

$$\bigcup_{v_i\in V} x_i \leq 1 \qquad \bigcup_{v_i\in V} x_i \geq 0$$

Eine Kante kann nur dann im Schnitt liegen, wenn genau einer ihrer inzidenten Knoten in S liegt.

$$\bigcup_{e_i=\{v_a,v_b\}\in E} x_a + x_b + z_i \leq 2 \qquad \bigcup_{e_i=\{v_a,v_b\}\in E} x_a + x_b - z_i \geq 0$$

Nun benötigt der entstandene Schnitt mindestens k Kanten.

$$\sum_{e_i \in E} z_i \geq k$$

z.Z. erfüllbarkeits-äquivalent (MAX CUT $\Leftrightarrow$ ILP)

- $\Rightarrow$ Für jeden Knoten v_i aus S wähle $x_i = 1$ (0 für alle Anderen). Nun können wir für jede Kante e_i aus dem Schnitt die Variable z_i auf wahr setzen. Da sich mindestens k Kanten im Schnitt befinden, erreicht die Summe mindestens den Wert k.
- $\Leftarrow$ Gegeben einer erfüllenden ILP-Belegung, lässt sich leicht die Zugehörige Schnittmenge Ablesen. Jedes z_i kann nur dann 1 sein, wenn genau eines der zugehörigen x_j bereits 1 ist. Dementsprechend steht jedes z_i welches 1 ist für eine Kante im Schnitt und die gesuchte Schnittgröße k wird erreicht.

Die Konstruktion ist offensichtlich polynomiell (2 Constraints pro Knoten + 2 Constraints pro Kante + ein Constraint für die Grenze, alle mit maximal linearer Länge)

Aufgabe 2 (*Entropie,* $2 + 2 + 2$ *Punkte*)

Berechnen Sie die Entropie, einen Shannon-Fano-Code und einen Huffman-Code für folgende Beispiele. Geben Sie die gewichtete durchschnittliche Codelänge Ihrer Codes an.

a)

a_i	P_i
A	0.4
B	0.3
C	0.2
D	0.1

b)

a_i	P_i
A	0.3
B	0.25
C	0.25
D	0.15
E	0.05

c)

a_i	P_i
A	0.36
B	0.18
C	0.18
D	0.12
E	0.09
F	0.07

Musterlösung:

a)

a_i	P_i	$-\log_2 P_i$	SF code	Huff code
A	0.4	1.322	0	0
B	0.3	1.737	10	10
C	0.2	2.322	110	110
D	0.1	3.322	111	111

Entropie: $\sum_i P_i \log_2 \frac{1}{P_i} \approx 1.846$ Bit/Zeichen
Durchschnittliche Codelänge: 1.9 (beide)

b)

a_i	P_i	$-\log_2 P_i$	SF code	Huff code
A	0.3	1.737	00	00
B	0.25	2	01	01
C	0.25	2	10	10
D	0.15	2.737	110	110
E	0.05	4.322	111	111

Entropie: $\sum_i P_i \log_2 \frac{1}{P_i} \approx 2.148$ Bit/Zeichen
Durchschnittliche Codelänge: 2.2 (beide)

c)

a_i	P_i	$-\log_2 P_i$	SF code	Huff code
A	0.36	1.474	00	0
B	0.18	2.474	01	100
C	0.18	2.474	10	101
D	0.12	3.059	110	110
E	0.09	3.474	1110	1110
F	0.07	3.837	1111	1111

Entropie: $\sum_i P_i \log_2 \frac{1}{P_i} \approx 2.370$ Bit/Zeichen
Durchschnittliche Codelänge: 2.44 (beide)

Anmerkung: In diesen Beispielen ist die durchschnittliche Codelänge des Huffman-Codes immer gleich der des Shannon-Fano-Codes. Dass dies im allgemeinen nicht gilt, zeigt die nächste Aufgabe.

Aufgabe 3 (*Shannon-Fano vs. Huffman,* 4 *Punkte*)

Geben Sie ein Alphabet und zugehörige Zeichenwahrscheinlichkeiten an, sodass die durchschnittliche gewichtete Codelänge des Shannon-Fano-Codes größer ist als die eines Huffmancodes.

Musterlösung:

a_i	P_i	SF code	|SF|	Huff code	|Huff|
A	0.4	00	2	0	1
B	$^6/_{35}$	01	2	100	3
C	$^1/_7$	10	2	101	3
D	$^1/_7$	110	3	110	3
E	$^1/_7$	111	3	111	3
Σ:	1	Schnitt:	≈ 2.26	Schnitt:	2.2

Aufgabe 4 (*Codelänge von Huffman-Codes,* 3 *Punkte*)

In der Literatur findet man teilweise folgende Behauptung:

> Man kann zeigen, dass die Länge der Huffmankodierung eines Zeichens mit Wahrscheinlichkeit P_i stets höchstens $\lceil -\log_2 P_i \rceil$ ist

Obwohl diese Behauptung in vielen Fällen stimmt, ist sie im Allgemeinen falsch. Geben Sie ein Beispiel an, das die Behauptung widerlegt.

Musterlösung:

P_i	Code	$-\log_2 P_i$	$\lceil -\log_2 P_i \rceil$
0.01	000	6.644	7
0.30	**001**	**1.737**	**2**
0.34	01	1.556	2
0.35	1	1.515	2