Übungsblatt 5

Vorlesung Theoretische Grundlagen der Informatik im WS 20/21

Ausgabe: 12. Januar 2021
Abgabe: 26. Januar 2021, 11:30 Uhr (digital im ILIAS)

Aufgabe 1

(1 + 2 = 3 Punkte)

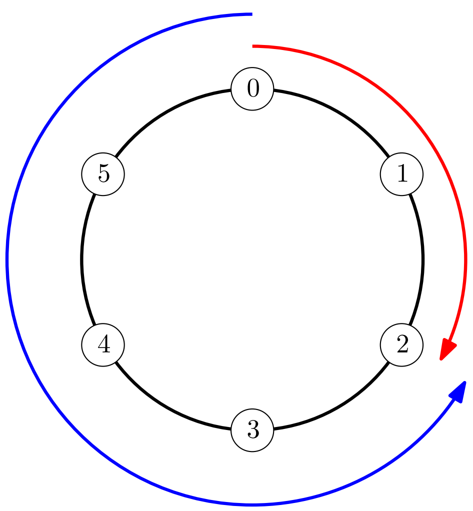
(a) Sei Π ein $\mathcal{NP}$-vollständiges Problem, zu dem ein pseudopolynomialer Algorithmus existiert. Warum impliziert dies nicht die Existenz eines pseudopolynomialen Algorithmus für jedes $\mathcal{NP}$-vollständige Problem?

(b) Zeigen Sie, dass ein stark $\mathcal{NP}$-vollständiges Problem genau dann von einem pseudopolynomialen Algorithmus entschieden wird, wenn $\mathcal{P} = \mathcal{NP}$ gilt.

Aufgabe 2

(2 + 1 + 1 + 1 + 1 + 2 = 8 Punkte)

Das Optimierungsproblem RingRouting tritt in Telekommunikationsnetzwerken auf. Gegeben sind n Knoten, die ringförmig durch ungerichtete Kanten verbunden sind. Formal lässt sich der Ring als Graph mit Knotenmenge $V = \{0, 1, \ldots, n-1\}$ und Kantenmenge $E = \{\{i, (i+1) \mod n\} \mid i \in V\}$ darstellen. Außerdem gegeben ist eine Menge C von m *Nachrichten* der Form (i, j), die von Startknoten i zu Zielknoten j geroutet werden sollen. Es gibt zwei Möglichkeiten, eine Nachricht zu routen: mit dem Uhrzeigersinn oder gegen den Uhrzeigersinn über den Ring. Folgendes Beispiel zeigt einen Ring mit 6 Knoten und die beiden Möglichkeiten, die Nachricht $(0, 2)$ zu routen:



Die *Last* $L(e)$ einer Kante e ist die Anzahl an Nachrichten, die über e geroutet werden. Die *Gesamtlast* $L = \max_{e \in E} L(e)$ ist die höchste Last unter allen Kanten im Ring. Ziel des Optimierungsproblems RingRouting ist es, die Nachrichten so routen, dass die Gesamtlast minimiert wird. Es kann gezeigt werden, dass die Entscheidungsvariante von RingRouting $\mathcal{NP}$-vollständig ist.

In der Vorlesung haben wir gesehen, dass sich jedes $\mathcal{NP}$-vollständige Problem als *ganzzahliges lineares Programm* (engl. *Integer Linear Program*, ILP) darstellen lässt. Um RingRouting als ILP darzustellen, führen wir für jede Nachricht c eine Variable $x_c \in \{0, 1\}$ ein. Dabei soll $x_c = 0$ bedeuten, dass die Nachricht gegen den Uhrzeigersinn geroutet wird, und $x_c = 1$, dass sie mit dem Uhrzeigersinn geroutet wird. Für eine Kante e bezeichnen wir mit $\overrightarrow{C}(e)$ die Menge der Nachrichten aus C, die über Kante e laufen, wenn man sie im Uhrzeigersinn routet. Analog ist $\overleftarrow{C}(e)$ die Menge der Nachrichten, die über e laufen, wenn man sie gegen den Uhrzeigersinn routet. Beachten Sie, dass $C = \overrightarrow{C}(e) \dot{\cup} \overleftarrow{C}(e)$ gilt.

Im Folgenden ist ein noch unvollständiges ILP für RingRouting gegeben, das wir mit RR-$\mathbb{N}$ bezeichnen:

Gegeben:
RingRouting-Instanz $I = (n, C)$

Variablen:
Für jede Nachricht $c \in C$ eine Variable x_c
Eine Variable L für die Gesamtlast

Nebenbedingungen:

(I) Für jede Nachricht $c \in C$: $x_c \in \{0, 1\}$

(II) $L \in \mathbb{N}_0$

(III) Für jede Kante $e \in E$: $L(e) \leq L$

Zielfunktion:
Minimiere L

(a) Vervollständigen Sie die ILP-Formulierung, indem sie eine Formel für die Last $L(e)$ in Abhängigkeit der Variablen x_c angeben.

(b) Zeigen Sie, dass für eine optimale Lösung von RR-$\mathbb{N}$ tatsächlich $L = \max_{e \in E} L(e)$ gilt.

Eine gängige Methode, ILPs zu approximieren, ist die *LP-Relaxierung.* Dabei wird die Beschränkung aufgehoben, dass die Variablen ganzzahlige Werte haben müssen. Das dadurch entstehende *lineare Programm* (LP) lässt sich in Polynomialzeit lösen. Im Fall von RingRouting erhalten wir das Problem RR-$\mathbb{R}$. Der einzige Unterschied gegenüber RR-$\mathbb{N}$ sind die Nebenbedingungen (I) und (II). Die Variablen x_c dürfen nun beliebige reelle Zahlen aus dem Intervall $[0, 1]$ sein dürfen. Intuitiv bedeutet das, dass die Nachrichten aufgeteilt werden dürfen – ein Teil der Nachricht wird im Uhrzeigersinn geroutet und der Rest gegen den Uhrzeigersinn. Dementsprechend sind auch die Last $L(e)$ der Kanten und die Gesamtlast L nun reellwertig.

Sei im Folgenden $\mathrm{OPT}_{\mathbb{R}}(I)$ der Wert von L für eine optimale Lösung von RR-$\mathbb{R}$ und $\mathrm{OPT}_{\mathbb{N}}(I)$ der Wert von L einer optimalen Lösung von RR-$\mathbb{N}$.

(c) Zeigen Sie, dass für jede RingRouting-Instanz $I = (n, C)$ gilt: $\mathrm{OPT}_{\mathbb{R}}(I) \leq \mathrm{OPT}_{\mathbb{N}}(I)$.

(d) Geben Sie eine RingRouting-Instanz $I = (n, C)$ an, für die $\mathrm{OPT}_{\mathbb{R}}(I) < \mathrm{OPT}_{\mathbb{N}}(I)$ gilt. Geben Sie jeweils eine optimale Lösung an.

Wir betrachten nun folgenden Algorithmus $\mathcal{A}$, der eine (nicht notwendigerweise optimale) Lösung für RR-$\mathbb{N}$ berechnet:

1. Berechne eine optimale Lösung $X = ((x_1, \ldots, x_m), L)$ mit $x_c \in [0, 1]$ und $L \in \mathbb{R}_0^+$ für RR-$\mathbb{R}$.
2. Generiere eine Lösung $X' = ((x'_1, \ldots, x'_m), L')$ mit $x'_c \in \{0, 1\}$ und $L' \in \mathbb{N}_0$ für RR-$\mathbb{N}$:
   (a) Wähle jedes x'_c wie folgt:
   $$x'_c = \begin{cases} 1 & \text{falls } x_c \geq \frac{1}{2} \\ 0 & \text{sonst.} \end{cases}$$
   (b) Wähle $L' =$ $\boxed{}$

(e) Wie muss L' gewählt werden, um eine gültige Lösung für RR-$\mathbb{N}$ um erzeugen?

(f) Zeigen Sie, dass $\mathcal{A}$ eine Approximationsalgorithmus für RR-$\mathbb{N}$ mit einer relativen Gütegarantie von 2 ist.

Aufgabe 3 (3 + 1 = 4 Punkte)

Das MIN-STEINERTREE-Problem ist wie folgt definiert: Gegeben sind ein zusammenhängender, ungerichteter, gewichteter Graph $G = (V, E)$ mit Kantengewichtsfunktion $d \colon E \to \mathbb{R}_{\geq 0}$ und eine Menge von *Terminalknoten* $R \subseteq V$. Ein *Steinerbaum* ist ein Baum $T = (V', E')$ mit $R \subseteq V' \subseteq V$ und $E' \subseteq E$, also ein Baum in G, der alle Terminale enthält. Gesucht ist ein Steinerbaum mit minimalem Gewicht, also mit
$$D(T) := \sum_{e \in E'} d(e)$$
minimal.

In dieser Aufgabe betrachten wir eine spezielle Variante, das MIN-METRIC-STEINERTREE-Problem: Wir gehen davon aus, dass G ein vollständiger Graph ist, also $E = \binom{V}{2}$. Außerdem fordern wir, dass die Kanten die Dreiecksungleichung erfüllen, d.h. für jedes Tripel aus Knoten $u, v, w \in V$ gilt $d(\{u, v\}) + d(\{v, w\}) \geq d(\{u, w\})$.

(a) Wir bezeichnen mit $G[R]$ den von R induzierten Subgraph von G, also den Subgraph von R, der genau R enthält sowie alle Kanten, die zwischen Knoten aus R verlaufen. Sei T ein Steinerbaum von G. Zeigen Sie: Es existiert ein Spannbaum T' von $G[R]$ mit $D(T') \leq 2 \cdot D(T)$.

(b) Betrachten Sie folgenden Algorithmus $\mathcal{A}$ für das MIN-METRIC-STEINERTREE-Problem: Gegeben eine MIN-METRIC-STEINERTREE-Instanz $I = (G, d, R)$, berechne einen minimalen Spannbaum von $G[R]$. Zeigen Sie, dass $\mathcal{A}$ ein Approximationsalgorithmus mit relativer Gütegarantie 2 ist.

Aufgabe 4 (5 + 2 + 1 = 8 Punkte)

Wir betrachten das $\mathcal{NP}$-schwere Maximierungsproblem LONGESTPATHWITHGIVENENDPOINTS:

Gegeben: Ungerichteter Graph $G = (V, E)$ mit zwei unterschiedlichen Knoten $v, w \in V$
Gesucht: Länge (= Anzahl der Kanten) des längsten Pfades von v nach w

Nehmen Sie an, für dieses Problem gibt es einen Approximationsalgorithmus mit relativer Gütegarantie k für ein $k > 1$.

(a) Zeigen Sie, dass es einen Approximationsalgorithmus für LongestPathWithGivenEndpoints mit relativer Gütegarantie $\sqrt{k}$ gibt.

Hinweis: Erzeugen Sie einen größeren Graphen, indem Sie Kanten durch Kopien des gegebenen Graphen ersetzen.

(b) Zeigen Sie, dass es ein PTAS für LongestPathWithGivenEndpoints gibt. Ist Ihr Algorithmus auch ein FPTAS? Begründen Sie.

(c) Gibt es ein FPTAS für LongestPathWithGivenEndpoints?

Aufgabe 5

(2 + 1 = 3 Punkte)

Sei p ein Polynom und Π ein $\mathcal{NP}$-schweres Minimierungsproblem, bei dem die Optimierungsfunktion f_Π des Problems ganzzahlig ist. Außerdem gelte für jede Instanz I, dass $\mathrm{OPT}_\Pi(I) < p(|I_u|)$, wobei I_u die unäre Kodierung von I bezeichnet.

Zeigen Sie:

(a) Falls es für Π ein FPTAS gibt, so gibt es auch einen pseudopolynomialen Algorithmus für Π.

(b) Falls Π stark $\mathcal{NP}$-vollständig ist und $\mathcal{P} \neq \mathcal{NP}$ gilt, gibt es für Π kein FPTAS.

Aufgabe 6

(2 + 1 + 1 + 1 + 2 + 1 = 8 Punkte)

Bei dem Optimierungsproblem Max-Sat ist eine Variablenmenge $V = \{v_1, v_2, \ldots, v_n\}$ und eine Klauselmenge $C = \{c_1, c_2, \ldots, c_m\}$ gegeben. Gesucht ist eine Wahrheitsbelegung der Variablen aus V so, dass möglichst viele Klauseln aus C erfüllt sind.

Betrachten Sie folgenden *randomisierten* Algorithmus für Max-Sat. Wähle eine zufällige Wahrheitsbelegung, indem jede Variable $v \in V$ unabhängig jeweils mit Wahrscheinlichkeit $1/2$ auf wahr bzw. falsch gesetzt wird. Bezeichne mit W die Anzahl an erfüllten Klauseln. Beachten Sie, dass W eine Zufallsvariable ist.

(a) Zeigen Sie $\mathbb{E}(W) = \sum_{c \in C} \left(1 - \left(\frac{1}{2}\right)^{|c|}\right)$, wobei $|c|$ die Anzahl an Literalen der Klausel c bezeichnet.

(b) Zeigen Sie $\mathbb{E}(W) \geq \frac{1}{2}\,\mathrm{OPT}$.

(c) Zeigen Sie $\mathbb{E}(W) = \frac{7}{8}m$, wenn jede Klausel genau drei (unterschiedliche) Literale enthält.

Sie haben gezeigt, dass der randomisierte MAX-SAT Algorithmus "erwartet" ein Approximationsalgorithmus ist. Wir *derandomisieren* den Algorithmus, um einen deterministischen, also "richtigen" Approximationsalgorithmus zu erhalten. Dazu benutzen wir bedingte Wahrscheinlichkeiten. Aufgrund der Konstruktion der zufälligen Wahrheitsbelegung gilt

$$\mathbb{E}(W) = \frac{1}{2}\mathbb{E}(W \mid v_1 \leftarrow \text{wahr}) + \frac{1}{2}\mathbb{E}(W \mid v_1 \leftarrow \text{falsch})\,.$$

Wir wählen nun $\varphi(v_1) \in \{\text{wahr}, \text{falsch}\}$ so, dass $\mathbb{E}(W \mid v_1 \leftarrow \varphi(v_1)) \geq \mathbb{E}(W \mid v_1 \leftarrow \neg\varphi(v_1))$ gilt.

(d) Zeigen Sie $\mathbb{E}(W \mid v_1 \leftarrow \varphi(v_1)) \geq \mathbb{E}(W)$.

(e) Erklären Sie, wie $\mathbb{E}(W \mid v_1 \leftarrow \text{wahr})$ und $\mathbb{E}(W \mid v_1 \leftarrow \text{falsch})$ in Polynomialzeit berechnet werden können.

Hinweis: Denken Sie an Teilaufgabe (a). Was gilt für eine Klausel, die durch die Wahrheitsbelegung $v_1 \leftarrow$ wahr erfüllt wird? Was gilt für eine Klausel, in der v_1 vorkommt, die aber durch die Wahrheitsbelegung $v_1 \leftarrow$ wahr nicht erfüllt wird? Was gilt für eine Klausel, in der v_1 nicht vorkommt?

(f) Definieren Sie $\varphi(v_i)$ für $2 \leq i \leq n$ so, dass $\mathbb{E}(W \mid v_i \leftarrow \varphi(v_i) \text{ für } 1 \leq i \leq n) \geq \mathbb{E}(W)$ gilt. Vergegenwärtigen Sie sich außerdem, dass diese Wahrheitsbelegung in Polynomialzeit berechnet werden kann.

Damit folgt, dass der derandomisierte Algorithmus ein 2-Approximationsalgorithmus für MAX-SAT ist. Hat jede Klausel genau drei (unterschiedliche) Literale, so ist der derandomisierte Algorithmus sogar ein 7/8-Approximationsalgorithmus. Johan Håstad hat gezeigt[1], dass für dieses Problem kein besserer Approximationsalgorithmus exisiert, es sei denn $\mathsf{P} = \mathsf{NP}$. In diesem Sinne ist der obige Approximationsalgorithmus optimal.

[1] https://doi.org/10.1145/502090.502098