Institut für Theoretische Informatik
Lehrstuhl Prof. Dr. D. Wagner

Übungsblatt 5

Vorlesung Theoretische Grundlagen der Informatik im WS 18/19

Ausgabe 18. Dezember 2018
Abgabe 15. Januar 2019, 11:00 Uhr (im Kasten im UG von Gebäude 50.34)

Aufgabe 1

(3 + 2 + 2 = 7 Punkte)

(a) Beweisen Sie, dass es unter der Annahme $\mathcal{P} \neq \mathcal{NP}$ keinen polynomialen Approximationsalgorithmus mit relativer Gütegarantie für das TravelingSalesman-Problem gibt. Zeigen Sie dazu, dass für jedes $k \geq 1$ gilt: Wenn es einen polynomialen Approximationsalgorithmus mit relativer Gütegarantie k für das TSP gäbe, so ließe sich daraus ein Polynomialzeitalgorithmus für das $\mathcal{NP}$-vollständige HamiltonianCycle-Problem konstruieren.

(b) Wir betrachten nun das metrische TSP, bei dem die Gewichtsfunktion die Dreiecksungleichung erfüllt, d.h. für alle $u, v, w \in V$ gilt: $c(u, w) \leq c(u, v) + c(v, w)$. Zeigen Sie, dass das metrische TSP $\mathcal{NP}$-schwer ist, indem Sie von TSP reduzieren.

(c) In der Vorlesung wurde gezeigt, dass für das metrische TSP polynomiale Approximationsalgorithmen mit relativer Gütegarantie existieren. Warum lässt sich mit der Reduktion aus Aufgabenteil (b) kein polynomialer Approximationsalgorithmus mit relativer Gütegarantie für das TSP konstruieren?

Aufgabe 2

(1 + 1 = 2 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 3

(1 + 4 + 3 + 2 = 10 Punkte)

Das *k-Zentren-Problem* ist wie folgt definiert: Gegeben seien eine Menge von Punkten $P \subseteq \mathbb{R}^2$ im zweidimensionalen Raum, die euklidische Distanzmetrik $\operatorname{dist}(\cdot, \cdot)\colon P^2 \to \mathbb{R}$ und ein fester Parameter

k. Gesucht ist eine Teilmenge $C \subseteq P$ mit $|C| = k$ von *Zentren*, sodass die maximale Distanz eines Punktes $p \in P$ zu seinem nächsten Zentrum in C minimiert wird. Wir bezeichnen das nächste Zentrum eines Punktes p in C mit $C(p)$. Wir wollen also $\max\limits_{p \in P} \operatorname{dist}(p, C(p))$ minimieren.

Betrachten Sie folgenden Algorithmus für das k-Zentren-Problem:

Algorithmus 1: k-CenterApprox

Wähle beliebigen Punkt $p \in P$
$C_1 \leftarrow \{p\}$
Für $i = 2 \dots k$
  Für $p \in P \setminus C_{i-1}$
    $\operatorname{dist}_i[p] \leftarrow \min\limits_{c \in C_{i-1}} \operatorname{dist}(p, c)$
  Wähle $c_i \in P \setminus C_{i-1}$ mit $\operatorname{dist}_i[c_i]$ maximal
  $C_i \leftarrow C_{i-1} \cup \{c_i\}$
return C_k

Es wird also in jedem Schritt der Punkt als neues Zentrum ausgewählt, der die größste Distanz zu den bisher ausgewählten Zentren hat.

(a) Sei $C_\mathcal{A}$ die Menge der Zentren, die von k-CenterApprox berechnet wurden. Sei p der Punkt, für den $\operatorname{dist}(p, C_\mathcal{A}(p))$ maximal ist. Zeigen Sie: Wenn $\operatorname{dist}(p, C_\mathcal{A}(p)) > x$ gilt, dann haben alle Zentren in $C_\mathcal{A}$ paarweise untereinander Distanz $> x$.

(b) Zeigen Sie, dass k-CenterApprox eine 2-Approximation für das k-Zentren-Problem berechnet.

Wir betrachten nun als Variante das *k-Produzenten-Problem*: Hier wird die Menge P der Punkte unterteilt in eine Menge D von Konsumenten und eine Menge S von Produzenten. Als Zentren dürfen jetzt nur Produzenten gewählt werden, also $C \subseteq S$. Minimiert werden soll nun $\max\limits_{d \in D} \operatorname{dist}(d, C(d))$, d.h. uns interessiert nur noch die Distanz der Konsumenten zum nächsten Zentrum.

(c) Geben Sie einen Algorithmus an, der eine 3-Approximation für das k-Produzenten-Problem berechnet.

(d) Beweisen Sie, dass Ihr Algorithmus eine 3-Approximation berechnet.

Aufgabe 4 (3 Punkte)

Geben Sie einen pseudopolynomialen Algorithmus für das Partition-Problem an. Zeigen Sie, dass Ihr Algorithmus polynomielle Laufzeit in der Eingabegröße und der größten in der Eingabe vorkommenden Zahl hat.

Hinweis: Verwenden Sie dynamische Programmierung.

Aufgabe 5 (3 + 1 = 4 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.[1]

Aufgabe 6

(3 + 1 + 1 = 5 Punkte)

Für einen Punkt p in der Ebene und eine positive reelle Zahl $r > 0$ ist eine *Doppelkreisbeschriftung* durch zwei Punkte q_1, q_2 gegeben, die folgende Bedingungen erfüllen:

$$|pq_1| = |pq_2| = r \qquad \text{und} \qquad |q_1q_2| = 2r.$$

Anschaulich bedeutet dies, dass der Punkt p der Berührungspunkt zweier Kreise mit Radius r und mit den Mittelpunkten q_1 und q_2 ist.

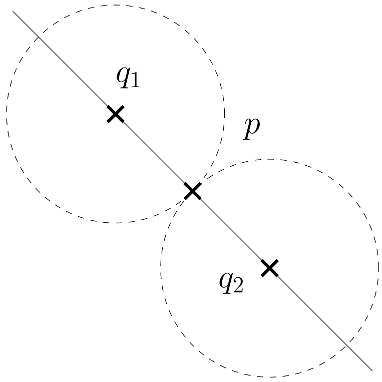

Abbildung 1: Beispiel einer Doppelkreisbeschriftung.

Bei dem Optimierungsproblem DOUBLECIRCLELABEL wird zu einer Menge von n Punkten $p_1, \ldots, p_n$ in der Ebene die größte (positive) reelle Zahl r gesucht, so dass jeder Punkt p_i eine r-Doppelkreisbeschriftung hat und die Doppelkreisbeschriftungen verschiedener Punkte schnittfrei sind.

(a) Geben Sie einen Approximationsalgorithmus mit relativer Gütegarantie 2 und polynomialer Laufzeit an und zeigen Sie ebendiese Eigenschaften.

Hinweis: Sie dürfen davon ausgehen, dass zwei mit Radius r schnittfrei doppelkreisbeschriftete Punkte p_1, p_2 mindestens Abstand $2r$ haben.

(b) Konstruieren Sie ein Beispiel, bei dem Ihr Approximationsalgorithmus nicht den optimalen Wert ausgibt. Erklären Sie!

[1] In der ursprünglichen Version des Übungsblatts hieß es fälschlicherweise: „Unter der Annahme $\mathcal{P} \neq \mathcal{NP}$ gibt es für ein stark $\mathcal{NP}$-vollständiges Problem kein FPTAS.“ Die Einschränkungen für Π aus dem vorigen Aufgabenteil müssen aber auch hier gelten.

(c) Das Problem DoubleCircleLabel ist $\mathcal{NP}$-vollständig. Geben Sie eine Intuition dafür, inwiefern die Entscheidungen, die Ihr Approximationsalgorithmus aus Aufgabenteil (a) trifft, „leichter“ sind als diejenigen, die ein optimaler Algorithmus für DoubleCircleLabel treffen muss.

Aufgabe 7 (3 + 2 + 1 = 6 Punkte)

Beim Entscheidungsproblem k-GraphPartition gilt es zu entscheiden, ob die Knoten eines Graphen $G = (V, E)$ so in k Mengen partitioniert werden können, dass alle Mengen der Partition exakt gleich groß sind und die Anzahl an Kanten, die nicht zwischen Knoten derselben Menge verlaufen, höchstens m ist (wobei m auch Teil der Eingabe ist).

Bei dem Problem 3-Partition erhält man eine Menge von $n = 3k$ Ganzzahlen $a_1, a_2, \ldots, a_n$ und einen Wert A mit

$$A/4 < a_i < A/2 \quad \text{und} \quad \sum_{i=1}^{n} a_i = kA.$$

Die Aufgabe ist es, zu entscheiden, ob die Zahlen so in Tripel partitioniert werden können, dass die Summe der Zahlen jedes Tripels gerade A ist. Dieses Problem ist stark $\mathcal{NP}$-vollständig. Also ist das Problem Restricted-3-Partition, bei dem alle a_i höchstens polynomielle Größe in n haben, immer noch $\mathcal{NP}$-vollständig.

(a) Geben Sie eine polynomiale Transformation von Restricted-3-Partition auf k-GraphPartition an. Beweisen Sie alle geforderten Eigenschaften.

(b) Beim Optimierungsproblem k-GraphPartition gilt es, den kleinsten Wert von m auszugeben, für den eine derartige Partition existiert. Zeigen Sie, dass es unter Annahme von $\mathcal{P} \neq \mathcal{NP}$ keinen Approximationsalgorithmus mit fester relativer Gütegarantie und polynomialer Laufzeit für das Optimierungsproblem k-GraphPartition gibt.

(c) Weshalb lässt sich ein analoger Beweis zu dem aus Teilaufgabe (b) nicht mit einer polynomiellen Transformation von Partition führen?