Prof. Dr. Anne Driemel
Frederik Brüning, Andreas Hene
Institut für Informatik

Abgabe: 28.06.2023 bis 12:00 Uhr

Übungsblatt 5

Aufgabe 5.1: (2+3+1 Punkte)

Wir betrachten die Optimierungsvariante des Problems VertexCover. Eingabe hierfür ist ein ungerichteter Graph $G = (V, E)$. In der Optimierungsvariante soll eine Teilmenge $T \subseteq V$ minimaler Kardinalität berechnet werden, sodass für jede Kante von G mindestens einer der inzidenten Knoten in T liegt.

`APPROX-VERTEX-COVER`(G)
- $T = \emptyset$
- $E' = G.E$ (kopiere Kantenmenge E von G)
- **while** $E' \neq \emptyset$
  - Wähle beliebige Kante (u, v) aus E'
  - $T = T \cup \{u, v\}$
  - Entferne aus E' alle zu u oder v inzidenten Kanten
- **end while**
- **return** T

(a) Geben Sie je ein Beispiel für einen Graphen G, für welchen `APPROX-VERTEX-COVER`(G) stets eine optimale bzw. suboptimale Lösung berechnet.

(b) Zeigen Sie, dass `APPROX-VERTEX-COVER`(G) in polynomieller Zeit eine 2-Approximation eines optimalen VertexCover von G berechnet.

(c) Betrachten Sie folgenden Greedy-Algorithmus: Nehme wiederholt einen Knoten beliebigen Knoten, füge ihn zu T hinzu und lösche alle benachbarten Kanten.
Zeigen Sie, dass dieser Ansatz keine 2-Approximation eines optimalen VertexCover von G berechnet.

Aufgabe 5.2: (3+1 Punkte)

Wir betrachten die Optimierungsvariante des Problems MaxCut. Eingabe hierfür ist ein einfacher ungerichteter Graph $G = (V, E)$. In der Optimierungsvariante soll eine Teilmenge $S \subseteq V$ berechnet werden, sodass die Anzahl der Kanten im Schnitt $\{(v, w) \mid v \in S,\ w \in V \setminus S\}$ maximal ist.

`APPROX-MAX-CUT`(G)
- Wähle zwei beliebige Knoten $s, t \in V$
- $S = \{s\}$, $T = \{t\}$
- **while** $V \setminus (S \cup T) \neq \emptyset$
  - Wähle beliebigen Knoten v aus $V \setminus (S \cup T)$
  - **if** $|\{(v, w) \mid w \in S\}| < |\{(v, w) \mid w \in T\}|$
    - $S = S \cup \{v\}$
  - **else**
    - $T = T \cup \{v\}$
- **end while**
- **return** S

(a) Zeigen Sie, dass `APPROX-MAX-CUT`(G) stets in polynomieller Zeit ein S mit $|\{(v, w) \mid v \in S,\ w \in V \setminus S\}| \geq \frac{1}{2}|E|$ berechnet.

(b) Ist `APPROX-MAX-CUT`(G) ein Approximationsalgorithmus für MaxCut? Begründen Sie Ihre Antwort und geben Sie gegebenenfalls einen Approximationsfaktor an.

Aufgabe 5.3: (6 Punkte)

Wir betrachten das Problem der *Graphtraversierung*. Gegeben ist ein zusammenhängender Graph $G = (V, E)$ mit Kanten- und Knotengewichten, und ein Startknoten $v_s \in V$. Das Gewicht eines Knotens v ist w_v und das einer Kante e ist w_e, alle Gewichte sind ≥ 1. Alle Agenten starten auf dem Startknoten v_s, und diese können sich entlang der Kanten von G bewegen. Eine Kante e kann jeweils nur von mindestens w_e Agenten überquert werden, und ein Knoten v darf erstmals nur mit mindestens w_v vielen Agenten betreten werden. Wird v erstmals betreten so werden sofort w_v Agenten auf v platziert – diese dürfen v nicht mehr verlassen. Das Ziel ist mit möglichst wenigen Agenten auf jedem Knoten v mindestens w_v Agenten zu platzieren.

Wir betrachten folgenden Algorithmus zum Traversieren eines Graphen G; der Algorithmus TraversiereBaumOptimal(T, v_s) wird als Black-Box verwendet; sie berechnet eine Strategie mit der Baum T von Knoten v_s aus mit einer kleinstmöglichen Anzahl Agenten traversiert wird.

GraphtraversierungMST($G = (V, E), v_s$)
1. Konstruiere einen minimalen Spannbaum $T = (V, E_T)$ von G
2. TraversierungsStrategie $S :=$ TraversiereBaumOptimal(T, v_s)
3. **return** S

Sei $N := \sum_{v \in V} w_v$ und $w_{max} = \max_{e \in E_T} w_e$. Zeigen Sie

(a) S benötigt maximal $N + w_{max}$ Agenten zum Traversieren von G.

(b) Zum Traversieren von G benötigt man mindestens $w_{max} = \max_{e \in E_T} w_e$ Agenten, wobei E_T die Kantenmenge eines minimalen Spannbaums von G ist.

(c) S benötigt zum Traversieren von G maximal doppelt so viele Agenten wie mindestens notwendig.

Es darf angenommen werden, dass alle nicht platzierten Agenten sich immer auf einem Knoten befinden bzw. sich in einem Schritt alle über die gleiche Kante bewegen.

Aufgabe 5.4: (4 Punkte)

Wenden Sie den in der Vorlesung vorgestellten Algorithmus zur Berechnung einer 2-Approximation auf die unten abgebildete Instanz des metrischen Traveling Salesperson Problems an. Dabei sei das Kantengewicht einer nicht eingezeichneten Kante $\{u, v\}$ gerade die Länge eines kürzesten Weges von u nach v. Starten Sie die Suche nach einem Eulerkreis in Knoten A und besuchen Sie bei Wahlmöglichkeit Knoten mit im Alphabet vorangehenden Buchstaben zuerst. Zeichnen Sie die resultierende TSP-Tour und geben Sie die finalen Kantengewichte an!