



CHRISTIAN-ALBRECHTS-UNIVERSITÄT ZU KIEL

Institut für Informatik, Arbeitsgruppe Theoretische Informatik
Dr. Max A. Deppert, Janina Reuter, Annika Huch

29.06.2026

Präsenzaufgaben zur Vorlesung »Analyse von Algorithmen und Komplexität«

Blatt 12

Präsenzaufgabe 12.1 (Vertex Cover)

Ein Vertex Cover in einem Graphen $G = (V, E)$ ist eine Menge von Knoten $C \subseteq V$, so dass für jede Kante $\{v, w\} \in E$ gilt, dass $v \in C$ oder $w \in C$. Beim Optimierungsproblem Minimum Vertex Cover soll die kardinalitätskleinste Menge $C \subseteq V$ gefunden werden, die ein Vertex Cover von G ist.

Gegeben ist der folgende approximative Algorithmus mit Güte 2 für VertexCover:

Idee: Wähle eine beliebige Kante. Wenn einer der beiden Endknoten bereits in C ist, verwirf diese Kante. Sonst, füge beide Endknoten zur Menge C hinzu. Mache dies für jede Kante.

Algorithmus 2APPROXVC($G = (V, E)$)

```
1   $C = \{\}$ 
2  foreach  $\{u, v\} \in E$  do
3    if  $u \notin C$  and  $v \notin C$  then
4       $C = C \cup \{u, v\}$ 
5    fi
6  od
7  return  $C$ 
```

Beweisen Sie die Güte.

Lösung. Beweis: Sei C^* ein minimales Vertex Cover. Wir zeigen $|C| \leq 2|C^*|$.

Sei A die Menge der Kanten, für die beide Endknoten zu C hinzugefügt wurden. Es gilt $|C| = 2|A|$, da beide Knoten jeder ausgewählten Kante zu C hinzugefügt wurden und alle anderen Kanten nicht weiter betrachtet werden. Damit folgt auch, dass keine zwei Kanten aus A einen Knoten gemeinsam haben können. Sei nun C^* ein minimales Vertex Cover. C^* enthält nach Definition mindestens einen Knoten jeder Kante, also insbesondere einen Knoten jeder Kante aus A . Da keine zwei Kanten aus A vom gleichen Knoten aus C^* überdeckt werden können, gilt $|C^*| \geq |A|$. Es folgt $|C| = 2|A| \leq 2|C^*|$. Die Lösung C von **2ApproxVC** ist also maximal doppelt so groß wie die optimale Lösung C^* . Damit berechnet der Algorithmus eine 2-Approximation.

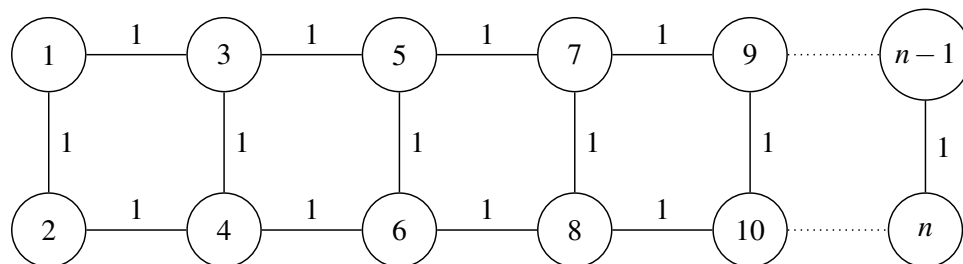
□

Präsenzaufgabe 12.2 (TSP)

In der Vorlesung haben Sie zwei approximative Algorithmen für das TSP-Problem kennen gelernt. Der erste Algorithmus ΔTSP_1 hat eine Approximationsrate von 2, während der zweite Algorithmus ΔTSP_2 von Christofides eine Approximationsrate von $3/2$ besitzt. In dieser Übung soll gezeigt werden, dass die zweite Rate scharf ist. D.h., dass es eine Folge von Instanzen gibt, auf denen der Algorithmus immer näher an diese Rate heran kommt.

Wir betrachten den vollständigen Graphen $K_n = (V, E)$ für ein $n \in \mathbb{Z}_{\geq 6}$ mit $n \bmod 4 = 2$ und die nachfolgende Gewichtsfunction $d_2: E \rightarrow \mathbb{Z}_{>0}$.

Wir definieren $E_2 := \{\{i, i+2\} | i \in \{1, \dots, n-2\}\} \cup \{\{2i+1, 2i+2\} | i \in \{0, \dots, n/2-1\}\}$ und d_2 , so dass alle Kanten aus E_2 auf 1 abgebildet werden und alle Kanten aus $E \setminus E_2$ auf den Wert des kürzesten Wegs zwischen diesen Knoten in dem Graphen $G_2 = (V, E_2)$ abgebildet werden. Der Graph G_2 ist im folgenden abgebildet.



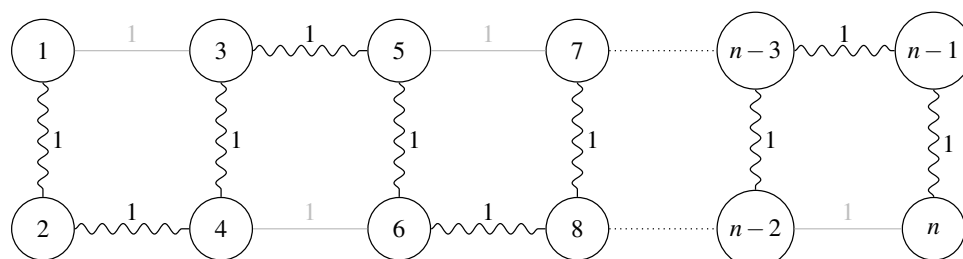
Zeigen Sie, dass

- (a) $\text{OPT}((K_n, d_2)) = n$ gilt und
- (b) eine mögliche Ausführung von ΔTSP_2 existiert, die auf Eingabe (K_n, d_2) eine Tour der Länge $(n-1) + n/2$ berechnet.

Lösung. Wichtig bei dieser Aufgabe ist der folgende Fakt: Bei beiden Algorithmen wird *irgendein* MST genommen und dann *irgendeine* Reihenfolge der Knoten. Typischerweise wird diese Reihenfolge durch bestimmte Knoten- oder Kanten-Labels festgelegt (wird zum Beispiel der Kruskal-Algorithmus zur Konstruktion des MST genutzt, so werden dort die Kanten nach Gewicht sortiert. Kanten gleichen Gewichts müssen also auch noch durch irgendeine Reihenfolge festgelegt werden.) Wir gehen also davon aus, dass wir diese Reihenfolge auch wählen können. Sollte der Algorithmus also einen “besseren” MST oder eine “bessere” Knotenreihenfolge nutzen, passen wir die Labels dementsprechend an.

Ein möglicher MST ist durch die unten angegebene Zick-Zack-Tour gegeben oder formal durch

$$E_T = \left[\bigcup_{\substack{1 \leq j \leq n-5 \\ j \bmod 4 = 1}} \{\{j, j+1\}, \{j+1, j+3\}, \{j+3, j+2\}, \{j+2, j+4\}\}\right] \cup \{\{n-1, n\}\}.$$



Es gilt $d(T) = n - 1$ für die Kosten $d(T)$ des MST T , da nur Kanten mit Kosten 1 gewählt werden und $n - 1$ Kanten gewählt werden müssen, um den Zusammenhang sicher zu stellen.

Da jeder Knoten im MST außer 1 und n geraden Grad hat, ist das Matching K sehr simpel, nämlich $K = \{\{1, n\}\}$. Die Kosten der Kante $\{1, n\}$ sind $n/2$ (der kürzeste Pfad von 1 zu n) und somit gilt $d(K) = n/2$ für die Kosten von K . Die Rundreise R entspricht einfach dem MST vereinigt mit der Matchingkante und hat somit Kosten: $d(R) = n - 1 + n/2$.

Die optimale Tour würde erst “oben” und dann “unten” entlang laufen, also von Knoten 1 zu Knoten 3 gehen, dann zu Knoten 5 und so weiter, bis Knoten $n - 1$ erreicht wurde. Danach folgen Knoten $n, n - 2, n - 4$ und so weiter. Diese Tour hat offensichtlich Kosten n , da wir n Kanten mit Gewicht 1 nutzen. Somit gilt $\text{OPT}((K_n, d_2)) = n$ und wir haben eine Rate von

$$\frac{\Delta TSP_2((K_n, d_2))}{\text{OPT}((K_n, d_2))} = \frac{n - 1 + n/2}{n} = 3/2 - \frac{1}{n} \leq 3/2.$$

Da $\lim_{n \rightarrow \infty} 3/2 - \frac{1}{n} = 3/2$, kommen wir der Rate von $3/2$ also beliebig nahe.

□