ETH zürich
Departement Informatik

Approximations- und Online-Algorithmen

Dr. Hans-Joachim Böckenhauer
Dr. Dennis Komm
Dr. Manuel Wettstein

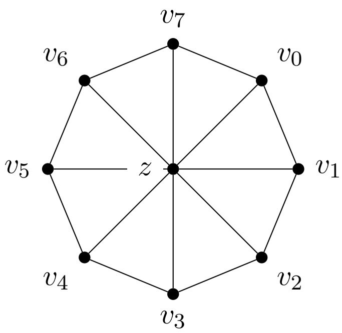
courses.csedu.inf.ethz.ch/approx_online_alg_24

Lösungsvorschläge – Blatt 1

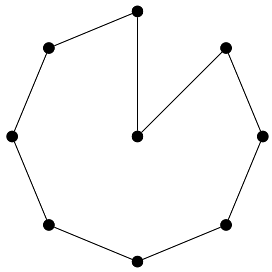
Zürich, 29. Februar 2024

Lösung zu Aufgabe 1

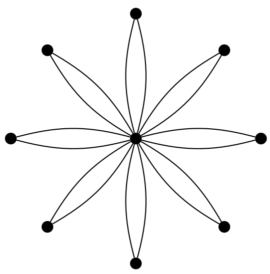
(a) Für jedes ungerade n konstruieren wir den folgenden Graphen $G = (V, E)$ mit n Knoten. Zunächst erstellen wir einen Stern mit $n - 1$ Strahlen, deren Kanten jeweils Kosten von 1 besitzen. Wir bezeichnen den mittleren Knoten mit z und die Blätter mit v_0 bis v_{n-2}, wobei die Nummerierung im Uhrzeigersinn erfolgt. Nun fügen wir alle Kanten $\{v_i, v_{i+1 \bmod (n-1)}\}$ für $0 \leq i \leq n - 2$ ein und geben ihnen ebenfalls Kosten 1 (Abbildung (a)). Um G abschliessend zu vervollständigen, fügen wir alle fehlenden Kanten hinzu und setzen ihre Kosten auf 2. Offensichtlich erfüllt ein Graph, in dem alle Kanten entweder Kosten 1 oder 2 haben, die Dreiecksungleichung.



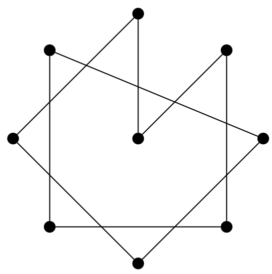
(a)



(b)



(c)



(d)

Eine optimale Lösung in G mit Kosten n ist gegeben durch den Hamiltonkreis $z, v_0, v_1, \ldots, v_{n-2}, z$ (Abbildung (b)). Es ist klar, dass keine Lösung mit geringeren Kosten existieren kann, da hier nur Kanten mit Kosten von jeweils 1 verwendet werden und es in G keine Kanten mit einem geringeren Gewicht gibt.

Betrachten wir nun eine mögliche Lösung, die der vorgestellte Spannbaum-Algorithmus berechnen kann. Zunächst konstruiert er einen minimalen Spannbaum in G. Wir sehen sofort, dass es keinen eindeutigen gibt und jeder solche Baum Kosten von $n-1$ besitzt. Ein billigster Spannbaum T ist beispielsweise durch den Stern gegeben, den wir zu Beginn konstruiert haben, und wir dürfen annehmen, dass dieser berechnet wird. Anschliessend werden die Kanten von T verdoppelt, wodurch eine Eulertour auf T entsteht (Abbildung (c)). Diese wird nun zu einem Hamiltonkreis verkürzt, indem, von einem beliebigen Knoten aus startend, eine Tiefensuche in T durchgeführt wird und bereits besuchte Knoten übersprungen werden. Die wichtige Beobachtung ist nun, dass es einen möglichen Verlauf dieser Tiefensuche gibt, sodass alle verwendeten Kanten bis auf zwei Kosten von 2 besitzen.

Nehmen wir an, der Algorithmus für die Tiefensuche startet beim Knoten z (den er als Wurzel interpretiert), und besucht anschliessend den Knoten v_0, der in T ein Blatt ist. Somit geht er zurück zu z und fährt mit seiner Tiefensuche fort. Angenommen, der nächste besuchte Knoten ist v_2. Da v_2 ebenfalls ein Blatt ist, geht er wieder zurück zu z. Der Algorithmus fährt mit der Tiefensuche fort, bis er alle Knoten mindestens einmal besucht hat. Nehmen wir an, die restlichen Knoten werden in der Reihenfolge $v_4, v_6, v_1, v_3, v_5, v_7$ gefunden. Die Eulertour auf T, die sich dadurch ergibt, ist also $z, v_0, z, v_2, z, v_4, z, \ldots$. Um den Hamiltonkreis zu berechnen, werden alle schon einmal besuchten Knoten ausgelassen, also ergibt sich: $z, v_0, v_2, v_4, v_6, v_1, v_3, v_5, v_7, z$ (Abbildung (d)). Die Gesamtkosten dieser Lösung belaufen sich also auf $2(n-2)+2$, und somit finden wir für jede Konstante $\varepsilon > 0$ ein n_0, sodass für alle $n > n_0$ gilt, dass $2(n-1)/n > 2 - \varepsilon$.

(b) Wir können für jede konkrete Strategie, die der Spannbaum-Algorithmus (dieser ist deterministisch) verfolgt, die Knoten $v_0, \ldots v_{n-2}$ in Teilaufgabe (a) so permutieren, dass garantiert eine Lösung mit Approximationsgüte $2-\varepsilon$ gefunden wird. Dies geschieht allerdings nur unter der Voraussetzung, dass der Algorithmus mit dem Stern als Spannbaum startet. Der Graph besitzt andere minimale Spannbäume, die zu einer besseren Güte führen. Man kann aber sogar erzwingen, dass der einzige minimale Spannbaum in G der Stern ist, indem man die Kantengewichte etwas anders wählt als zuvor: Sei $\varepsilon' = \frac{1}{n-1}$. Allen Kanten, die zum Stern gehören, geben wir das Gewicht $1-\varepsilon'$. Die Kanten $\{v_i, v_{i+1 \bmod (n-1)}\}$ bekommen das Gewicht 1. Um die Dreiecksungleichung zu erhalten, müssen wir nun allen restlichen Kanten die Kosten $2(1-\varepsilon')$ geben. Somit ist der minimale Spannbaum eindeutig: Der Stern ist der einzige Spannbaum mit Kosten $(1-\varepsilon')(n-1) = n-2$, alle anderen Spannbäume von G haben grössere Kosten. Der Algorithmus kann nun wie oben schon beschrieben den Hamiltonkreis $z, v_0, v_2, v_4, v_6, v_1, v_3, v_5, v_7, z$ berechnen. Mit den neuen Kantengewichten hat dieser Kosten von $2(1-\varepsilon') + (n-2) \cdot 2(1-\varepsilon') = 2(1-\varepsilon')(n-1)$, während die optimale Tour Kosten $2 \cdot (1-\varepsilon') + n - 2 = n - 2\varepsilon'$ hat. Wenn wir wie oben gewählt $\varepsilon' = \frac{1}{n-1}$ einsetzen, ergibt sich als Approximationsgüte:

$$\frac{2(1-\varepsilon')(n-1)}{n-2\varepsilon'} > \frac{2(n-2)}{n} = 2 - \frac{4}{n}.$$

Wenn wir nun $n_0 = \frac{4}{\varepsilon}$ setzen, gilt also für jede Konstante $\varepsilon > 0$ und alle $n > n_0$, dass die Approximationsgüte des Spannbaum-Algorithmus mindestens $2 - \frac{4}{n} > 2 - \frac{4}{4/\varepsilon} = 2 - \varepsilon$ ist.