ETH zürich
Departement Informatik

Approximations- und Online-Algorithmen

Dr. Hans-Joachim Böckenhauer
Dr. Dennis Komm

`courses.ite.inf.ethz.ch/approx_online_alg_22`

Lösungsvorschläge – Blatt 1

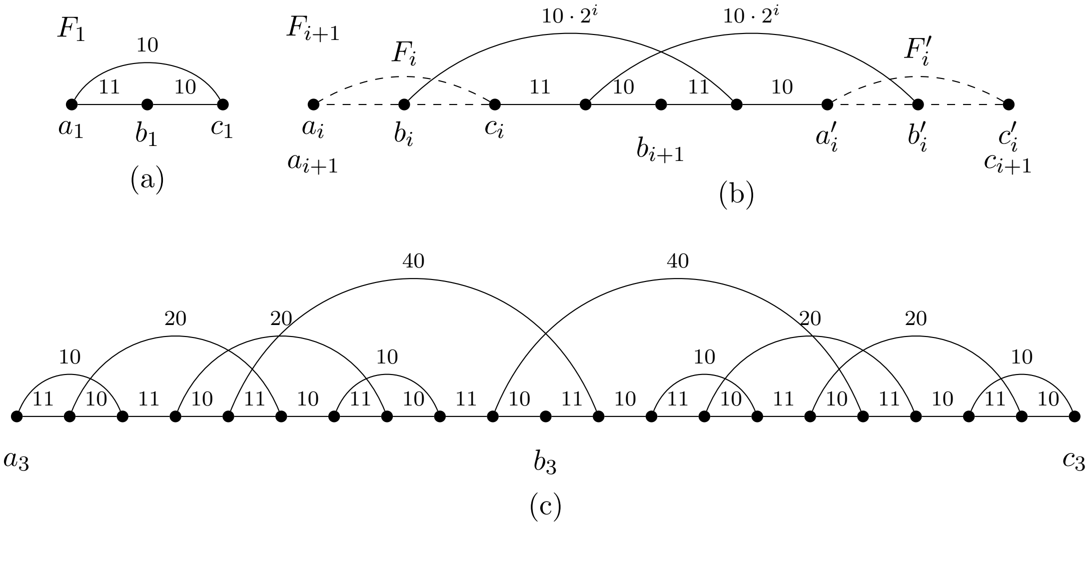
Zürich, 3. März 2022

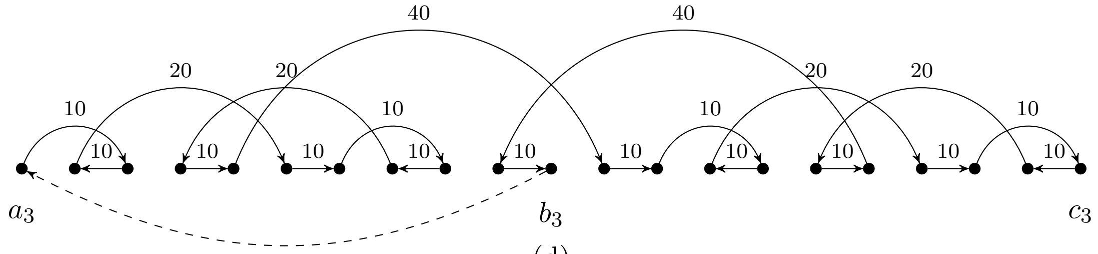
Lösung zu Aufgabe 1

(a) Die gesuchte schwere Eingabeinstanz kann analog zum schweren Beispiel für den Christofides-Algorithmus konstruiert werden, das in der Vorlesung vorgestellt wurde (im Buch in Abbildung 4.15 direkt nach Theorem 4.3.5.5). Ein möglicher Lauf des Greedy-Algorithmus kann in der in Abbildung 4.15 (a) dargestellten Instanz einen Hamiltonpfad von v_1 nach v_{2t} erzeugen, was ihn zwingt, mit der Kante $\{v_{2t}, v_1\}$ den Kreis zu schliessen (siehe Abbildung 4.15 (d)). Dementsprechend ergibt sich auch hier eine Approximationsgüte von $(3t-1)/(2t)$, was für grosse t gegen 1.5 geht.

(b) Wir zeigen, dass der Greedy-Algorithmus auf der Klasse von gewichteten vollständigen Graphen F_i mit $i \geq 1$ eine Hamilton-Tour berechnet, die um einen Faktor von $\geq \frac{10}{63} \cdot (i+1)$ länger ist als die optimale Tour. Weil der Graph F_i insgesamt $n_i = 3 \cdot (2^i - 1)$ Knoten hat (was sich durch eine einfache Induktion beweisen lässt), ergibt sich damit die behauptete logarithmische Approximationsgüte.

Im Folgenden bezeichne a_i den in der Abbildung am weitesten links stehenden Knoten von F_i, b_i den mittleren Knoten von F_i und c_i den am weitesten rechts stehenden Knoten von F_i. Die Abbildung (a) zeigt den Graphen F_1, die Abbildung (b) zeigt die rekursive Konstruktion von F_{i+1}. Alle nicht eingezeichneten Kanten haben den maximalen bzgl. der Dreiecksungleichung zulässigen Wert.





(d)

Die Abbildung (c) zeigt als Beispiel den Graphen F_3, und in Abbildung (d) ist die Hamilton-Tour gezeigt, die der Greedy-Algorithmus konstruiert, wenn er mit dem Knoten a_3 startet.

Die Kosten c_{opt} einer optimalen Lösung sind kleiner oder gleich den Kosten der Lösung, die von links nach rechts von a_i bis zu c_i verläuft und dann mit einer zusätzlichen Kante von c_i nach a_i zurückkehrt. Damit gilt

$$\begin{aligned} c_{\text{opt}} &\le 2 \cdot \left(\frac{n_i - 1}{2} \cdot 21 \right) \\ &= 21 \cdot (3 \cdot (2^i - 1) - 1) \\ &= 21 \cdot (3 \cdot 2^i - 4) \\ &= 63 \cdot 2^i - 84 \end{aligned}$$

Wir bestimmen jetzt die Kosten c_{greedy} der Greedy-Lösung. Die Greedy-Lösung verwendet auf ihrem Weg von a_i nach b_i alle eingezeichneten Kanten vom Gewicht $\neq 11$.

Jeder Graph F_i enthält genau zwei (eingezeichnete) Kanten vom Gewicht $10 \cdot 2^{i-1}$. Da F_i rekursiv aus 2^{i-j} Graphen F_j zusammengesetzt ist, enthält F_i somit genau $2 \cdot 2^{i-j}$ (eingezeichnete) Kanten vom Gewicht $10 \cdot 2^{j-1}$ für alle $2 \le j \le i$. Damit berechnet sich das Gesamtgewicht aller (eingezeichneten) Kanten mit einem Gewicht ≥ 20 in F_i als $10 \cdot (i-1) \cdot 2^i$. Somit bleibt das Gesamtgewicht der Kanten in F_i mit einem Gewicht von 10 zu bestimmen. Eine einfache Induktion zeigt, dass es in F_i genau $2^{i+1} - 2$ Kanten vom Gewicht 10 gibt.

Damit benötigt die Greedy-Lösung für ihren Weg von a_i nach b_i Kosten von $10 \cdot ((i-1) \cdot 2^i + 2^{i+1} - 2) = 10 \cdot ((i+1) \cdot 2^i - 2)$. Die Kante von b_i zurück nach a_i hat ein Gewicht von

$$\left(\frac{n_{i-1} - 1}{2} \cdot 21 \right) + 21 = 21 \cdot (3 \cdot 2^{i-2} - 1).$$

Somit folgt

$$\begin{aligned} c_{\text{greedy}} &= 10 \cdot ((i+1) \cdot 2^i - 2) + 21 \cdot (3 \cdot 2^{i-2} - 1) \\ &= 10 \cdot (i+1) \cdot 2^i - 20 + 63 \cdot 2^{i-2} - 21 \\ &= 10 \cdot (i+1) \cdot 2^i + 63 \cdot \left(2^{i-2} - \frac{41}{63} \right). \end{aligned}$$

Damit ergibt sich für alle $i \ge 2$:

$$\frac{c_{\text{greedy}}}{c_{\text{opt}}} \ge \frac{10 \cdot (i+1) \cdot 2^i + 63 \cdot (2^{i-2} - \frac{41}{63})}{63 \cdot 2^i - 84}$$

$$
\begin{aligned}
&\geq \frac{10 \cdot (i+1) \cdot 2^i}{63 \cdot 2^i - 84} \\
&\geq \frac{10 \cdot (i+1) \cdot 2^i}{63 \cdot 2^i} \\
&= \frac{10}{63} \cdot (i+1)
\end{aligned}
$$

Weil $n_i = 3(2^i - 1)$ gilt, folgt

$$
\frac{c_{\text{greedy}}}{c_{\text{opt}}} \geq \frac{10}{63} \cdot \left(\log_2\left(\frac{n_i}{3} + 1\right) + 1\right),
$$

was die behauptete logarithmische untere Schranke ergibt.