Approximationsalgorithmen

Wintersemester 2018/2019

Dr. Annamaria Kovacs
Dipl-Math. Mahyar Behdju

GOETHE UNIVERSITÄT FRANKFURT AM MAIN

Institut für Informatik
AG Algorithm Engineering

Übung 10

Ausgabe: 19.12.2018
Abgabe: 16.01.2019

Aufgabe 10.1. *Min-Max (Lineare Programmierung Dualität)* (3 + 2 Punkte)

a) Seien $S, T \subset \mathbb{R}$ beliebige *endliche* Mengen. Zeige, dass für alle $f\colon S \times T \to \mathbb{R}$ gilt:

$$\max_{y \in T} \left(\min_{x \in S} f(x, y) \right) \leqslant \min_{x \in S} \left(\max_{y \in T} f(x, y) \right)$$

Hinweis: Ein möglicher Beweis vergleicht beide Werte mit $f(x_0, y_0)$ für geeignetes $(x_0, y_0) \in S \times T$. Wenn nötig, betrachte zunächst den Fall $|S| = |T| = 2$.

b) Sei nun $|S| = |T| = 2$. Konstruiere eine Funktion f, sodass

$$\max_{y \in T} \left(\min_{x \in S} f(x, y) \right) < \min_{x \in S} \left(\max_{y \in T} f(x, y) \right).$$

Aufgabe 10.2. (4 + 5 Punkte)

Wir betrachten das Feedback Vertex Set Problem:

Eingabe: ein ungerichteter Graph $G = (V, E)$

Ausgabe: eine Knotenmenge $S \subseteq V$ mit minimalem Gesamtgewicht, sodass jeder (*einfache*) Kreis $C = (v_1, v_2, \ldots, v_{l-1}, v_l = v_1)$ in G mindestens einen Knoten aus S enthält.
(Ein Kreis heißt *einfach*, wenn keiner seiner Knoten mehrmals durchlaufen wird.)

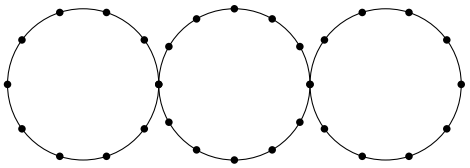
Das folgende LP ist eine LP-Relaxierung des Problems (für eine gegebene Instanz):

$$\begin{array}{lll} \text{Minimiere} & \sum_{v \in V} w_v x_v & \\ \text{(P) sodass} & \sum_{v \in C} x_v \geqslant 1 & \text{für alle } C \in \mathcal{K} \\ & x_v \geqslant 0 & \text{für alle } v \in V \end{array}$$

Hierbei bezeichnet $\mathcal{K}$ die Menge aller einfachen Kreise von G.

a) Bestimme das duale LP (D), und versuche dieses grob zu interpretieren.

b) Sei im folgenden Graphen jedes Knotengewicht $w_v = 1$. Bestimme drei verschiedene duale Lösungen y, sodass keine duale Variable (einzeln) höher gesetzt werden kann. Eine optimale Lösung soll auch dabei sein, mit kurzer Begründung der Optimalität.



Bitte wenden!

Aufgabe 10.3. (4 Punkte)

Bestimme das duale LP (D) für das folgende primale LP (P).

Maximiere $x_1 + 3x_3$, sodass

$$\begin{aligned} x_1 + x_2 + 2x_3 &\leqslant 8 \\ x_2 + 6x_3 &\leqslant 12 \\ x_1 &\leqslant 4 \\ x_2 &\leqslant 6 \\ x_1, x_2, x_3 &\geqslant 0 \end{aligned}$$

Aufgabe 10.4. *min*-Vertex Cover (3 Bonuspunkte)

Wir haben mehrere 2-approximative Algorithmen für min-Vertex Cover kennengelernt. Ein einfacher greedy Algorithmus für min-Vertex Cover kann wie folgt angegeben werden:

function Greedy$(G(V, E))$
  $C \leftarrow \emptyset$
  while $E \neq \emptyset$ **do**
    Sei $v \in V$ ein Knoten mit maximalem Grad
    $C \leftarrow C \cup \{v\}$
    $V \leftarrow V \setminus \{v\}$
    Entferne alle mit v inzidenten Kanten aus E und aktualisiere die Knotengrade
  return C

Was können wir über den Approximationsfaktor von Greedy anhand der folgenden Adjazenzmatrix A behaupten? (Die 0-Einträge wurden nicht aufgeführt.)

$$A = \left(\begin{array}{c|c|c|c|c|c|c|c|c|c|c|c|c} 1 & & & & & & 1 & & & 1 & & 1 & 1 \\ \hline & 1 & & & & & 1 & & & 1 & & 1 & 1 \\ \hline & & 1 & & & & & 1 & & 1 & & 1 & 1 \\ \hline & & & 1 & & & & 1 & & & 1 & 1 & 1 \\ \hline & & & & 1 & & & & 1 & & 1 & 1 & 1 \\ \hline & & & & & 1 & & & 1 & & 1 & 1 & 1 \\ \hline & & & & & & 1 & & 1 & & 1 & & 1 \end{array}\right)$$

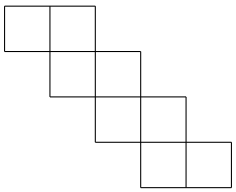
Hinweis: Wir definieren die Adjazenzmatrix A eines bipartiten Graphen $G(X, Y, E)$ wie folgt: Für $|X| = n_1$, $|Y| = n_2$ sei A eine $n_1 \times n_2$-Matrix mit

$$a_{ij} = \begin{cases} 1 & \text{falls } \{i, j\} \in E, \\ 0 & \text{sonst.} \end{cases}$$

Bitte wenden!

Aufgabe 10.5. (2 Bonuspunkte)

Schreibe die Zahlen 1 bis 8 in die unten dargestellten Kästchen, sodass keine aufeinander folgenden Zahlen in derselben Zeile, Spalte oder Diagonale vorkommen. Welches schwierige Problem aus der Graphentheorie steckt in diesem Rätsel?



Aufgabe 10.6. Facility Location (5 Bonuspunkte)

Wir betrachten das diskrete unendliche Gitter $\mathbb{Z} \times \mathbb{Z}$ als Instanz für Facility Location. Jeder Gitterpunkt entspricht hierbei einem Kunden und darf gleichzeitig als eine Service-Station s mit Betriebskosten $f_s = 2$ gewählt werden ($S = K = \{(a, b) \mid a, b \in \mathbb{Z}\}$). Die Distanz zweier Gitterpunkte ist die Distanz entlang vertikaler und horizontaler Gitterlinien, d. h. es gilt:

$$d\big((a_1, b_1), (a_2, b_2)\big) = |a_1 - a_2| + |b_1 - b_2|$$

Finde eine optimale Auswahl $X \subseteq S$ an Service-Stationen, d. h. ein X, welches die durchschnittlichen Kosten pro Gitterpunkt minimiert.

Hinweis: Beachte, dass in jeder Lösung X ein Punkt mit Distanz $\geqslant 2$ von X zur Service-Station deklariert werden kann, ohne zusätzliche Kosten zu verursachen. Bestimme die bestmögliche Häufigkeit an Stationen und finde hierfür eine geeignete Positionierung. Es ist ausreichend, die Optimalität der Lösung mit Hilfe dieser Überlegung zu begründen.

Aufgabe 10.7. *Der Handlungsreisende* (3 + 3 Bonuspunkte)

Ein Handlungsreisender bekommt den Auftrag durch n Städte zu reisen. Beim Betrachten der Karte stellt er fest, dass drei Rundreisen (nicht unbedingt minimaler Länge) existieren, die gemeinsam die gesamte Karte überdecken: Würde er allen drei Rundreisen folgen, würde er insgesamt jeden direkten Weg zwischen je zwei Städten (Kante) genau einmal durchfahren.

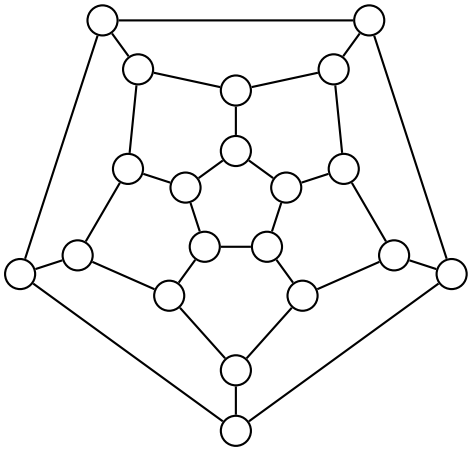
a) Bestimme die Anzahl der Städte n. Begründe deine Antwort.

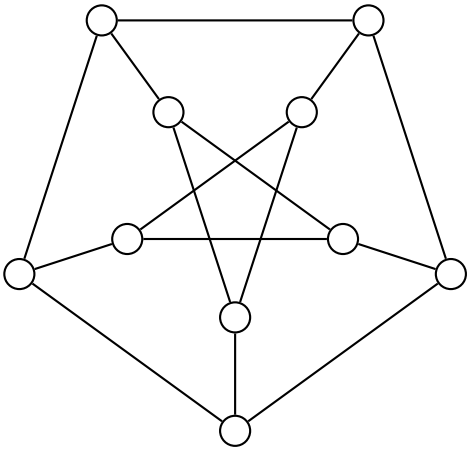
b) Beschreibe eine Möglichkeit für die drei Rundreisen.

Bitte wenden!

Aufgabe 10.8. (3 Bonuspunkte)

Welcher der folgenden beiden Graphen besitzt einen Hamilton-Kreis?





Aufgabe 10.9. (4 + 2 Bonuspunkte)

Bestimme den genauen Approximationsfaktor für die folgenden Greedy-Algorithmen für das Intervall-Scheduling-Problem. Analysiere hierfür jeweils die obere und die untere Schranke.

a) Ein Greedy-Algorithmus, der erst die Jobs aufsteigend nach ihrer Dauer sortiert und anschließend jeweils den nächsten Job in die Lösung aufnimmt, wenn dieser mit den bislang aufgenommenen Jobs nicht kollidiert.

b) Ein Greedy-Algorithmus, der erst die Jobs absteigend nach ihren Endpunkten sortiert und anschließend jeweils den nächsten Job in die Lösung aufnimmt, wenn dieser mit den bislang aufgenommenen Jobs nicht kollidiert.

E-Mail: `{panni,mahyar}@ae.cs.uni-frankfurt.de`