Approximationsalgorithmen

Wintersemester 2019/2020

Dr. Annamaria Kovacs
Dipl-Math. Mahyar Behdju

GOETHE UNIVERSITÄT FRANKFURT AM MAIN

Institut für Informatik
AG Algorithm Engineering

Übung 1

Ausgabe: 16.10.2019
Abgabe: 23.10.2019

Aufgabe 1.1. (1 + 1 + 1 + 1 + 2 + 2 Punkte)

Was können wir vom Approximationsfaktor der folgenden Algorithmen behaupten? Gib möglichst gute obere oder untere Schranken an, wo möglich.

a) Ein Algorithmus, der für jede Instanz I eines Maximierungsproblems jeweils mit einem Zielwert von mindestens $\frac{2}{3}\mathrm{OPT}_I$ eine Lösung findet.

b) Ein Algorithmus, der für jede Instanz I eines Maximierungsproblems jeweils mit einem Zielwert von mindestens $(1-\varepsilon)\cdot\mathrm{OPT}_I$ eine Lösung findet.

c) Ein Algorithmus, der für manche Instanzen I eines Maximierungsproblems jeweils mit einem Zielwert von genau $\frac{2}{3}\mathrm{OPT}_I$ eine Lösung findet.

d) Ein Algorithmus für ein Minimierungsproblem, der manche Instanzen genau optimiert.

e) Ein Algorithmus für Min-Scheduling auf einer bestimmten Klasse von Instanzen, der für jede Instanz I aus dieser Klasse einen Schedule mit Makespan höchstens $\frac{4}{3}\alpha_I$ ausgibt, wobei $\alpha_I = \max\{p_{\max}, \frac{1}{m}\sum_{i=1}^{n} p_i\}$.

f) Ein Algorithmus für Max-Clique, der

- ein Dreieck ausgibt, wenn der Graph (mindestens) ein Dreieck enthält,
- eine Kante ausgibt, wenn der Graph kein Dreieck, aber mindestens eine Kante enthält,
- sonst einen Knoten ausgibt.

Bitte wenden!

Aufgabe 1.2. (2 + 2 + 1 + 2 Punkte)

Wir betrachten den Scheduling-Algorithmus Shortest Processing Time first (SPT), der n Jobs auf m identische Maschinen verteilt: SPT sortiert zunächst die Jobs *aufsteigend* nach Ausführungszeit und wendet dann List-Scheduling an.

Wir wollen SPT auf eine Instanz mit $m = 35$ Maschinen und $n = 103$ anwenden, wobei 99 Jobs eine Ausführungszeit von 2, und 4 Jobs eine Laufzeit von 3 haben. Die zu minimierende Zielfunktion sei der Makespan.

a) Bestimme zunächst eine Reihenfolge der Jobs, mit der List optimal arbeitet. Beweise die Optimalität des Ergebnisses.

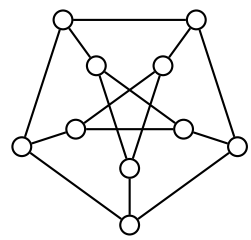
b) Bestimme den Approximationsfaktor von SPT auf der beschriebenen Eingabe.

c) Bestimme eine Instanz mit $m = 2$ Maschinen und größtmöglichem Approximationsfaktor für SPT und gib den Approximationsfaktor an. Ein Beweis, dass es keine schlechtere Eingabe gibt, ist nicht notwendig.

d) Nun betrachten wir den analogen Algorithmus Longest Processing Time first (LPT): LPT sortiert zunächst die Jobs *absteigend* nach Ausführungszeit und wendet dann List-Scheduling an.
Bestimme eine Instanz mit $m = 2$ Maschinen und größtmöglichem Approximationsfaktor für LPT und gib den Approximationsfaktor an. Ein Beweis, dass es keine schlechtere Eingabe gibt, ist nicht notwendig.

Aufgabe 1.3. (5 Punkte)

Gib ein **möglichst einfaches**, aber mathematisch korrektes Argument an, warum im Petersen-Graphen (siehe Abbildung) jede unabhängige Knotenmenge höchstens 4, während jede Knotenüberdeckung mindestens 6 Elemente enthält. Gib für beide Fälle eine entsprechende Lösung an.



Hinweis: Sei ein (ungerichteter) Graph $G = (V, E)$ gegeben. Eine Teilmenge $I \subseteq V$ heißt genau dann **unabhängige Knotenmenge** (independent set, kurz IS), wenn I keine benachbarte Knoten enthält, wenn also gilt: $\forall u, v \in I : \ \{u, v\} \notin E$.