Approximationsalgorithmen

Wintersemester 2018/2019

Dr. Annamaria Kovacs
Dipl-Math. Mahyar Behdju

GOETHE UNIVERSITÄT FRANKFURT AM MAIN
Institut für Informatik
AG Algorithm Engineering

Übung 2

Ausgabe: 24.10.2018
Abgabe: 31.10.2018

Aufgabe 2.1. (5 Punkte)

Zwei spezielle Varianten des LIST-Scheduling Algorithmus werden wie folgt definiert:

Longest Processing Time first (LPT):
Sortiere die Jobs absteigend nach Laufzeiten und führe LIST aus.

Shortest Processing Time first (SPT):
Sortiere die Jobs aufsteigend nach Laufzeiten und führe LIST aus.

Wir beschränken das min-SCHEDULING Problem auf drei Jobgrößen: 1, $\frac{1}{2}$ und $\frac{1}{3}$. Definiere je eine Worst-Case-Instanz (bezüglich des Approximationsfaktors) für LPT und für SPT aus (beliebig vielen) solchen Jobs. Ein Beweis, dass es keine schlechtere Eingabe gibt, ist nicht notwendig. Gib jeweils den Approximationsfaktor an.

Aufgabe 2.2. (2 + 3 Punkte)

Wir betrachten das min-EDGE-COLORING Problem, wobei die Kanten eines ungerichteten Eingabegraphen mit der minimalen Anzahl an verschiedenen Farben so gefärbt werden sollen, dass je zwei Kanten mit einem gemeinsamen Endknoten verschiedene Farben erhalten.

a) Bestimme eine einfache untere Schranke für die minimale Anzahl der Farben für beliebige Eingabegraphen $G = (V, E)$.

   Hinweis: Die Schranke soll keine Konstante sein, sondern von einem Parameter des Graphen abhängen.

b) Definiere einen 2-approximativen Greedy-Algorithmus für min-EDGE-COLORING auf beliebigen Graphen.

   Hinweis: Verwende hierfür die untere Schranke für OPT aus Teil a).

Bitte wenden!

Aufgabe 2.3. (4 Punkte)

Wir modellieren den d-dimensionalen Würfel als Graphen $W_d(V, E)$, wobei jede Ecke des Würfels einem Knoten in W_d entspricht. Jede Ecke wird dabei durch einen binären Vektor der Länge d dargestellt. Somit gilt $V = \{0, 1\}^d$. Zwei Knoten $x, y \in \{0, 1\}^d$ sind genau dann durch eine Kante verbunden, wenn sich x und y in exakt einem Eintrag unterscheiden.

Finde einen Hamiltonischen Kreis im 4-dimensionalen Würfel-Graphen W_4.

Hinweis: Löse das Problem zuerst auf dem 3-dimensionalen Würfel-Graphen W_3.

Aufgabe 2.4. *(Bonus)* (3 + 3 Bonuspunkte)

Gib einen neuen Beweis (mittels Induktion; bitte nicht den Beweis aus dem Skript als Lösung einreichen!) für die Optimalität des Algorithmus von Kruskal nach einem ähnlichen Schema wie beim Intervall-Scheduling:

a) Zeige, dass eine beliebige fixierte Kante e mit minimalem Gewicht in mindestens einem minimalen Spannbaum enthalten ist.

   Hinweis: Nimm einen optimalen Spannbaum, und tausche, falls nötig, eine andere Kante aus dem Spannbaum gegen e.

b) Zeige, dass man für die übrigen Kanten mit Kruskal weitermachen darf.

   Hinweis: Definiere einen (für den Induktionsschritt geeigneten) etwas kleineren Graphen.

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