

Übung zur Vorlesung Effiziente Algorithmen

Tutoraufgabe 28

Approximationsalgorithmen werden gewöhnlich für NP-schwere Probleme entworfen, denn Probleme in P können ja sowieso „leicht“ gelöst werden. Dennoch betrachten wir hier das Problem *Matching maximaler Kardinalität*, welches wir zwar in polynomieller Zeit lösen können, aber dennoch nicht sehr schnell, wenn man den Grad des Laufzeitpolynoms ernst nimmt.

- Entwerfen Sie einen Approximationsalgorithmus für *Matching maximaler Kardinalität* mit einem Approximationsfaktor 2.
- Wie schnell ist Ihr Algorithmus?
- Wie können Sie Ihren Algorithmus verwenden, um *Vertex Cover* zu approximieren? Wie schnell geht das und welchen Approximationsfaktor erhalten Sie?

Tutoraufgabe 29

Entwerfen Sie Approximationsalgorithmen für folgende Probleme. Welche Approximationsfaktoren erzielen Sie dabei?

- MaxSAT: Eingabe ist eine aussagenlogische Formel in CNF mit m Klauseln; Ausgabe ist eine Variablenbelegung, die möglichst viele Klauseln erfüllt.
- Metrisches TSP: Eingabe ist ein Graph mit Kantengewichten. Dabei werde die Dreiecksungleichung für Wegelängen zwischen Knoten erfüllt (wie z.B. in einem Straßennetz). Ausgabe: Eine möglichst kurze TSP-Tour.

Hausaufgabe 19 (10 Punkte)

Beweisen Sie folgende Behauptung aus der Vorlesung: Falls ein optimaler Plan für I jedem Prozessor höchstens zwei Aufgaben zuordnet, dann ist ein LPT-Schedule für I ebenfalls optimal.

