Approximationsalgorithmen

1 Parallel machine scheduling

Um eine Menge an Aufgaben au Maschinen zu verteilen, kann approximativ folgendermassen vorgegangen werden: Man geht die Liste aller Aufgaben in einer beliebigen Reihenfolge durch, und weist jeden Job derjenigen Maschine zu, bei welcher er am schnellsten fertig ist. Dabei ist zu beachten, dass eine Maschine, die bereits Jobs zugewiesen bekommen hat, diese zuerst fertigstellen muss. Da die Jobs in beliebiger Reihenfolge betrachtet werden können, handelt es sich hierbei um einen Online-Algorithmus.

1.1 Approximation ratio

Um die Güte der Approximation abzuschätzen, sei $A(I)$ die Zeit, welche der Algorithmus für eine Probleminstanz I benötigt. Weiter sei $OPT(I)$ die optimale Lösung eines Offline-Algorithmus. Dann Suchen wir folgenden Quotienten:

$$\text{Approximation ratio} := \max_{\mathrm{I}} \frac{A(I)}{OPT(I)}$$

Es gibt einen Job der Länge l, welcher für die Zeit von unserem Approximationsalgorithmus verantwortlich ist. Zeitlich vor diesem Job ist keine Maschine fertig, Während der Zeit $k = A(I) - l$ sind also alle Maschinen beschäftigt. Da auch $OPT(I)$ den Job der Länge l unterbringen muss, ist l eine untere Schranke. Genauso mit k: Da während dieser Zeit alle Maschinen besetzt sind, kann OPT das nicht besser machen. Folglich ist die „approximation ratio" für diesen Algorithmus 2.

2 TSP

2.1 TSP mittels MST

Da ein MST effizient berechnet werden kann, wäre es interessant, diesen als Approximation für das TSP zu benutzen. Das funktioniert auch, sogar ganz einfach: Der MST kann zu einer Rundreise gemacht werden (die jedoch gewisse Städte mehrmals besucht), indem man den MST „umläuft". Nun kann irgendwo begonnen werden, und jede Stadt, die ich bereits besucht habe, „überspringe" ich einfach, indem ich auf der durch den MST gegebenen Rundreise „abkürze".

Da $MST \leq OPT$ gilt, und unsere schlussendliche Tour höchstens $2 \cdot MST$ lang ist, gilt also:

$$Lösung \leq 2 \cdot MST \leq 2 \cdot OPT$$

Wir haben also eine Approximation mit einer competitive ration von 2 gefunden.

2.2 Cristofides

Will man eine noch bessere Approximation mit Güte 1.5 haben möchte, so kann folgendermassen vorgegangen werden (G sei der vollständige Graph der TSP Instanz):

1. Finde einen minimalen Spannbaum (MST) T von G.

2. Sei O nun die Knotenmenge mit ungeradem Grad in T. Finde ein minimales „perfect matching“ M im kompletten Graph G für die Knoten O. Ein solches existiert immer, da die Anzahl Knoten mit ungeradem Grad stets gerade sein muss.
3. Kombiniere die Kanten von M und T zum Graphen H.
4. Bilde eine Eulerschleife in H. Eine solche existiert immer, da H zusammenhängend ist, und alle Knoten geraden Grad haben.
5. Transformiere diesen Weg, sodass er Hamitonisch wird, indem bereits besuchte Knoten einfach ausgelassen werden.

2.2.1 Kosten

Natürlich ist der minimale Spannbaum eine untere Schranke für die optimale Lösung, also $MST \leq OPT$. Weiter ist eine optimale Rundreise durch die Knoten in O (nennen wir sie $OOPT$) sicher eine untere Schranke für die optimale Lösung, also $OOPT \leq OPT$. Weiter ist das perfekte Matching höchstens halb-sogross wie $OOPT$, da mit $OOPT$ gleich zwei perfekte Matching für O gegeben sind: Entweder nimmt man die 1., 3., etc. Kante dieser Rundreise, oder jede 2., 4., etc. Und eine dieser beiden ist sicherlich höchstens $0.5 \cdot OOPT$. Also gilt für das perfekte Matching: $pM \leq 0.5 \cdot OOPT \leq 0.5 \cdot OPT$.

Damit lässt sich unsere Approximation abschätzen:

$$APPROX \leq MST + pM \leq OPT + 0.5 \cdot OPT = 1.5 \cdot OPT$$