Fernau/Raible
Universität Trier

Wintersemester 2006/2007

Übungen zur Vorlesung

Näherungsalgorithmen

Aufgabenblatt 4

Aufgabe 1 (Pseudopolynomieller Algorithmus für MINIMUM MAKESPAN**)**
MINIMUM MAKESPAN ist Scheduling auf identischen Maschinen (MSIM) aus der Vorlesung. Betrachte nun das Beispiel mit 3 Maschinen ($p = 3$). Zeige, daß es einen in $l_{\max}$ pseudopolynomiellen Algorithmus für MINIMUM MAKESPAN gibt, wobei $l_{\max} = \max\{l_1, \ldots, l_n\}$ ist. Gib seine Laufzeit in n und $l_{\max}$ an.
Hinweis: Benutze dynamisches Programmieren.

Aufgabe 2 MAXIMUM ACYCLIC SUBGRAPH
Gegeben sei ein gerichteter Graph $G(D, A)$ und gesucht ist eine größte Menge $A' \subseteq A$, so daß $G(D, A)'$ azyklisch ist.

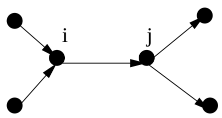
1. Betrachte folgenden Algorithmus:
   (a) Sei $D = \{d_1, \ldots, d_n\}$.
   (b) Arrangiere $d_1, \ldots, d_n$ in einer Linie an.
   (c) Zeichen die Kanten A ein.
   (d) Nehme nach L die Kanten, die von links nach rechts zeigen.
   (e) Nehme nach R die Kanten, die von rechts nach links zeigen.
   (f) **return** L falls $|L| > |R|$, sonst R.

   Zeige nun, daß es sich hierbei um eine gültige 2-Approximation handelt.

2. Wir betrachten nun dieses Problem auf Graphen mit maximalem Grad ≤ 3. Wir werden schrittweise eine $\frac{8}{9}$-Approximation herleiten.
   (a) Zeige, daß wir folgende Annahmen machen können:
      - Alle Knoten haben in-degree und out-degree mindestens 1 und gesamten Grad von 3.
      - G enthält weder gerichtete noch ungerichtete 2-er und 3-er Zykel.

Finde also dementsprechende Reduktionsregeln, welche die Ausnahmen der zwei Annahmen behandeln. Nach Anwendung der Regel, gibt es keine Ausnahme mehr, aber wir haben auch die optimal Lösungsgrösse nicht verändert.

(b) Eine α-Kante, ist eine Kante (i, j), so daß $d_{in}(i) = 2$ und $d_{out}(j) = 2$ ist, siehe Bild.



Zeige nun, daß in einem 3-regulären Graphen ohne α-Kanten jeder Kreis kantendisjunkt ist.

(c) Zeige, daß es für den Fall in (b) einen Polynomialzeit-Algorithmus für Maximum Acyclic Subgraph gibt.

(d) Betrachte folgenden Algorithmus (wobei $E(e)$ die offene Kantennachbarschaft der Kante e ist):

i. Solange es α-Kanten gibt, tue
   A. Wende die Reduktionsregeln aus 2) an.
   B. Nehme alle Kanten, die nicht in einem Kreis sind nach S
   C. Wähle eine α-Kante e.
   D. Falls die Komponente die e enthält genau 9 Kanten enthält, dann löse diese exakt.
   E. Entferne e. Nimm $E(e)$ zur Lösungsmenge S.
   F. Nehme auch Knoten v mit $d_{in} = 0$ oder $d_{out} = 0$ nach S.

ii. Löse den Rest polynomiell.

iii. **return** S

Wenn wir in Schritt ii durchgeführt haben, dann können wir jede Kante e' die durch Kontraktion eines Pfades von Grad-2-Knoten entstanden ist (siehe $2(a)$ wieder entrollen. War e' nicht in S so löschen wir eine beliebige Kante und der Rest kommt nach S. Für jedes Kontrahieren bekommen wir also mind. 1 Kante nach S.
Verwende dies um zu zeugen, daß der Algorithmus eine $\frac{8}{9}$-Approximation ist.

Aufgabe 3 (Unrelated Minimum Makespan **und Lineares Programmieren)**

Unrelated Minimum Makespan ist wie Minimum Makespan (siehe Aufgabe 1) mit der zusätzlichen Bedingung, daß die Maschinen $M = \{p_1, \ldots, p_m\}$ nicht identisch sind. Job J_i benötigt auf p_j eine Verarbeitungszeit von $l_{ij} \in \mathbb{Z}^+$.

1. Formuliere Unrelated Minimum Makespan als ganzzahliges lineares Programm.

2. Sei $S_T = \{(i,j) \mid l_{ij} \leq T\}$. T stellt eine obere Schranke für den Makespan dar. Betrachte nun das lineare Programm $LP(T)$, welches nur die Laufzeiten indiziert mit Paaren aus S_T und für das gelten soll das der Makespan $\leq T$ ist . Zeige nun:

   (a) Jede Basislösung von $LP(T)$ hat höchtstens $n+m$ von 0 verschiedene Variabelbelegungen ($n =$ Anzahl Jobs).
   Hinweis: Ist $r = |S_T|$, so zwingt eine Basislösung r linear unabhängige Ungleichungen zur Gleichheit. Was heißt dies für die Anzahl der Gleichugen der Form $x_{ij} \geq 0$ (x_{ij} Indikatorvariable für J_i auf p_j bearbeiten).

   (b) Sei x eine Basislösung und α die Anzahl der Jobs mit ganzzahligen Variablenbelegungen und β die der Jobs mit Nicht-ganzzahliegen. Klar $\alpha + \beta = n$. Zeige zuerst $\alpha + 2\beta \leq n + m$, damit $\alpha \geq n - m$.

3. Ein *Pseudo-Baum* ist ein zusammenhängender Graph mit $|V|$ Kanten, also ein Kreis mit Bäumen dran. Ein *Pseudo-Wald* besteht aus Komponenten die Pseudo-Bäume sind.
   Betrachte den bipartiten Graphen $G(J, M, E)$ mit $\{i, j\} \in E \iff x_{ij} \neq 0$. Zeige: G ist ein Pseudo-Wald.
   Hinweis: Zeige also, daß jede Komponente G_c von G ein Pseudo-Baum ist, also $E(G_c) \leq |V(G_c)|$. Beschränke $LP(T)$ und die Basislösung x auf die Jobs und Maschinen in G_c und wende $2(a)$ an.

4. Sei $F \subseteq J$ die Menge der Jobs die in x nicht-ganzzahlig sind. Sei $H = G(F \cup M)$ (H wird von $F \cup M$ auf G induziert). Zeige H hat ein perfektes Matching.
   Hinweis: Versuche konstruktiv zu beweisen und zeige, daß H ebenso ein Pseudo-Wald ist. Nutze dies.

5. Betrachte folgenden Algorithmus:

   (a) Finde ein Greedy-schedule mit Makespan γ.

   (b) Benutze Binär-Suche im Intervall $[\frac{\gamma}{m}, \gamma]$ um das kleinste $T* \in \mathbb{Z}^+$ zu finden, welches $LP(T*)$ erfüllt.

   (c) Finde eine Basislösung für $LP(T*)$.

   (d)

   (e)

   (f)

   Ergänze $(d), (e), (f)$, so daß der Algorithmus eine 2-Approximation wird.
   Hinweis: In welcher Beziehung stehen $T*$ und Opt?

Aufgabe 4 (Local Search)

1. Formuliere einen Meta-Local-Search-Algorithmus in dem der Nachbarschaftsbegriff zw. zwei Lösungen abstrakt bleibt.

2. Definiere für folgende Probleme geeignete Nachbatschaften für den Meta-Local-Search-Algorithmus. Halte den Nachbarschaftsbegriff so einfach wie möglich, betrachte immer nur höchstens zwei 'Elemente'.

   (a) Minimum Balanced Cut:
   Eingabe: $G(V, E)$ mit $|V| = 2n$ und $w : E \to \mathbb{Z}^+$.
   Lösung: Partition von V in gleich große Teile V_1 und V_2.
   Maß: $\sum_{\{u,v\} \in E, u \in V_1, v \in V_2} w(u, v)$

   (b) Das wohlbekannte Travelling Salesman.

   (c) Maximum Satisfiability: Maximiere die Anzahl der erfüllten Klauseln.