Fernau/Raible
Universität Trier

Wintersemester 2008/2009

Übungen zur Vorlesung

Näherungsalgorithmen

Aufgabenblatt 6

Aufgabe 1(Local Search)

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

2. Definiere für folgende Probleme geeignete Nachbarschaften 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.

Aufgabe 2(Set Cover)
Sei Set Cover folgendes Problem:
Eingabe: $V = \{1,\ldots,n\}$, $\mathcal{S} = \{S_1,\ldots,S_m\}$ mit $S_i \subseteq V$ für $1 \leq i \leq n$.
Gesucht: $I \subseteq \{1,\ldots,m\}$ sodass $\bigcup_{j\in I} S_j = V$
Optimierung: Minimiere $|I|$.

Sei für $u \in V$ $deg(u) := |\{S_i \mid u \in S_i\}|$ und $\Delta := \max\{deg(u) \mid u \in V\}$

1. Formuliere Set Cover als ganzzahliges lineares Programm.

2. Betrachte folgenden Algorithmus:

```
Löse das relaxierte lineare Programm aus Teilaufgabe 1). (Seien x_1, ..., x_m
die korrespondierenden Variablen zu S_1, ..., S_m)
COV ← ∅.
for i := 1 to m do
   if x_i ≥ 1/Δ then
      COV ← COV ∪ {S_i}
   end if
end for
```

Zeige nun: $|COV| \leq \Delta \cdot Opt(\mathcal{S})$.
Tipp: Zeige als erstes $|COV| \leq \sum_{i=1}^{m} \Delta x_i$

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

Unrelated Minimum Makespan ist wie Minimum Makespan (siehe Aufgabe 2, Blatt3) mit der zusätzlichen Bedingung, daß die Maschinen $M = \{M_1, \ldots, M_m\}$ nicht identisch sind. Job J_i benötigt auf M_j eine Verarbeitungszeit von $p_{ij} \in \mathbb{Z}^+$.

1. Formuliere Unrelated Minimum Makespan als ganzzahliges lineares Programm.

2. Sei $S_T = \{(i,j) \mid p_{ij} \leq T\}$. 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 höchstens $n+m$ 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 Gleichungen der Form $x_{ij} \geq 0$ (x_{ij} Indikatorvariable für J_i auf M_j bearbeiten), die Null annehmen?

   (b) Sei x eine Basislösung und α die Anzahl der Jobs mit ganzzahligen Variablenbelegungen und β die der Jobs mit Nicht-ganzzahligen. 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. Überlege, welche Knoten Grad 1 haben können.

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 $LP(T*)$ (Dies macht der Simplex-Algorithmus).

   (d) Verteile alle ganzzahligen Jobs in x auf die entsprechenden Maschinen.

   (e) Konstruiere für die nichtganzzahligen Jobs in x den Graph H.

   (f)

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