Fernau/Raible
Universität Trier

Wintersemester 2006/2007

Übungen zur Vorlesung

Näherungsalgorithmen

Aufgabenblatt 2

Aufgabe 1 (Δ-Hitting-Set)
Hitting Set ist das folgende Problem:

Eingabe: $\mathcal{U}= \{u_1, \ldots, u_n\}$, $\mathcal{S}= \{S_i, \ldots, S_m\}$ mit $S_i \subset \mathcal{U}$.
Gesucht: $\mathcal{L} \subseteq \mathcal{U}$ mit $S_i \cap \mathcal{L} \neq \emptyset$ und $w(\mathcal{L})$ minimal.

Für Δ-Hitting-Set gilt zusätzlich $|S_i| \leq \Delta$ für $1 \leq i \leq m$.
Adaptiere das Verfahren von Clarkson für Δ-Hitting-Set, so dass es eine Lösung mit Approximationsgüte Δ berechnet.

Aufgabe 2 ("Pseudo"-optimaler Vertex Cover-Algorithmus)
Betrachte folgenden Algorithmus PVC:

1. $VC \leftarrow \emptyset$
2. Solange $E \neq \emptyset$, tue:
   (a) Wähle $\{u, v\} \in E$ und dann beliebig $t = v$ oder $t = u$.
   (b) $VC \leftarrow VC \cup \{t\}$, $E \leftarrow E \setminus \{\{t, z\} \mid \{t, z\} \in E\}$

Betrachte nun folgenden "Optimalitätsbeweis":
Wir lassen den Algorithmus Rückwärts laufen und machen Induktion nach Anzahl der betrachteten Kanten: Sei $e_1 = \{u_1, v_1\}$ die letzte Kante, die PVC betrachtet. In $OptVC$ muss entweder u_1 oder v_1 sein. Da wir nach VC entweder u_1 oder v_1 nehmen, ist VC auf $G[e_1]$ nicht schlechter als $OptVC$.
Induktionsschritt: Sei nun VC auf $G[\cup_{i=1}^{k} e_j]$ (e_k ist k-letzte Kante die PVC nahm) nicht schlechter als $OptVC$ auf $G[\cup_{i=1}^{k} e_j]$: Betrachte nun $e_{k+1} = \{u_{k+1}, v_{k+1}\}$. PVC nimmt einen der beiden Knoten, das gleiche gilt aber auch für $OptVC$, also ist PVC auch auf $G[\cup_{i=1}^{k+1} e_j]$ nicht schlechter als $OptvC$.
Ist e_z die erste Kante die PVC betrachtete, folgt deshalb $G = G[\cup_{i=1}^{z} e_j]$ und damit ist PVC optimal.

Wo liegt der Fehler im vorigen Beweis? Es geht nicht darum ein Gegenbeispiel zu konstruieren, sondern den Beweis zu analysieren.

Aufgabe 3 (Matrix Row/Column Merging)

Matrix Row/Column Merging ist folgendes Problem:

Eingabe: Eine $n \times m$ Matrix M mit $\{0,1\}$-Einträgen.
Gesucht: Eine minimale Menge von Verschmelzungen von benachbarten Spalten bzw. Zeilen, so dass ein 0-Matrix resultiert.

Unter Verschmelzung von Spalten (resp. Zeilen) verstehen wir ein logisches **AND** der beiden Teile. Zu Spalte i (resp. Zeile i) sind die Spalten $i-1$ und $i+1$ benachbart.

1. Entwerfe eine Verfahren, das eine 4-Approximation garantiert.
   Tipp: Falls $M_{ij} = 1$, welche möglichen Verschmelzungen gibt es um die 1 zu löschen? Wie kann man 1er-Zeilen (Spalten) wegreduzieren.

2. Für 1en in einer Grenzlinie (Oberste bzw. unterste Zeile u. linkeste bzw. rechteste Spalte) reduziert sich die Anzahl der Möglichkeiten für Verschmelzungen, um diese 1 zu löschen, also könnten wir eine bessere Approximation erhalten. Dies setzt aber voraus, dass es keine 0-Zeilen (Spalten) gibt. Zeige, dass man in einem Vorverarbeitungsschritt diese 0-Zeilen (Spalten) löschen kann und die erhaltene Instanz M' in ebenso vielen Verschmelzungsschritten lösen kann, wie M. Zeige also, dass jede Verschmelzungssequenz $S = \{m_1, \ldots, m_k\}$, die o.B.d.A. die 0-Zeile benutzt, durch eine S‘ ersetzt werden kann, die nicht die 0-Zeile benutzt, also eine Lösung für M' ist.

Aufgabe 4 (2-Approximation für Clique Complement Cover**)**
Folgendes Problem ist Clique Complement Cover

Eingabe: $G(V, E)$, $w : E \to \mathbb{N}$
Gesucht: $C \subseteq E$, so dass $E \setminus C$ eine Clique ist und $w(C)$ minimal

Gebe für dieses Problem eine 2-Approximation an. Überlege, welche Kantenmengen zusammen nicht in einer Clique sein können. Wende Gewichtsreduktion an und argumentiere ähnlich wie bei den lokalen Verhältnissen.

Aufgabe 5 ($\frac{5}{3}$-Approximation für planares Vertex Cover**)**
Betrachte folgenden Algorithmus $PlanVC$:

1. $VC \leftarrow \emptyset$, $i = 0$

2. Solange G ein $U \subseteq V$ enthält, so dass $G[U]$ ein Dreieck ist: $VC \leftarrow VC \cup U$, $V \leftarrow (V \setminus U)$, $i \leftarrow i + 1$, $d_i \leftarrow 2$

3. Solange $V \neq \emptyset$, tue:

   (a) Nehme $v \in V$ mit minimalem Grad d, $i \leftarrow i + 1$, $d_i \leftarrow d$.
   (b) $X := N(v)$, $Y := N^2(v)$, $E' = \{\{x, y\} \mid x \in X, y \in Y\}$

(c) Finde in $G(X \cup Y, E')$ ein Matching M mit $|M| = |M \cap X| = d - 1$.

(d) $VC \leftarrow VC \cup X \cup (M \cap Y)$

Bemerkung:$N^2(v) = \{x \mid \exists \{v, u\}\{u, x\} \in E\}$, also in $N^2(v)$ sind Nachbarn von Nachbarn von v.

Fragen:

1. Warum ist der Graph in 3.(c) bipartit und wie lässt sich ein Matching M mit $|M| = |M \cap X| = d - 1$ leicht finden?

2. Sei e die Anzahl der Iterationen (der letzte i-Wert), dann $\tilde{d} := \frac{\sum_{i=1}^{e} d_i}{e}$.
   Zeige: $\dfrac{VC}{OptVC} \leq 2 - \frac{1}{\tilde{d}}$.
   Tipp: Überlege wie viele Knoten $OptVC$ für Dreiecke bzw. für $G[\{v\} \cup X \cup (M \cap Y)]$ mindestens braucht und wie viele $PlanVC$ nimmt.

3. Zeige: $\dfrac{VC}{OptVC} \leq \frac{5}{3}$. Gehe dabei folgend vor:

   (a) Für planare, dreiecksfreie Graphen gilt: $|E| < 2|V|$.
   Tipp: Schätze die Anzahl der Faces nach oben mit $|E|$ ab und verwende die Eulerformel.

   (b) Zeige, dass $d_i \leq 3$ für $i = 1, \ldots, e$ mit Hilfe von $\sum_{v \in V} deg(v) = 2|E|$.

   (c) Folgere nun $\frac{VC}{OptVC} \leq \frac{5}{3}$ mit 2.