Algorithmik

Sebastian Iwanowski
FH Wedel

8. Vorlesungswoche

Algorithmik 8

Nachlese: Algorithmus von Kruskal

Satz (Korrektheit):

Für jede Kantenmenge $\{k_1, k_2, ..., k_j\}$, die der Algorithmus von Kruskal sukzessive konstruiert, gibt es einen minimal spannenden Baum T_j von G, der diese Kantenmenge enthält

Beweis durch vollständige Induktion über j

Induktionsschluss:

Die Annahme gelte für $\{k_1, k_2, ..., k_j\}$. Sei k_{j+1} die nächste Kante. Wenn $k_{j+1} \in T_j$, setze $T_{j+1} = T_j$. Anderenfalls gibt es Kreis in T_j, der k_{j+1} erhält. Mindestens eine dieser Kreiskanten k kann nicht in $\{k_1, k_2, ..., k_j\}$ liegen (anderenfalls hätte Kruskal nicht k_{j+1} ausgewählt wegen der Kreisfreiheit). Ersetze diese Kante k durch k_{j+1} => spannender Baum T_{j+1}, der die ersten k_{j+1} Kanten von Kruskal enthält.
$c(k) \geq c(k_{j+1})$, denn anderenfalls hätte Kruskal k vor k_{j+1} ausgewählt.
Damit ist T_{j+1} genauso wie T_j minimal.

Referenzen zum Nacharbeiten und Vertiefen:

Skript Alt, Lemma 4.3.2 (S. 76): Beweisskizze eines verwandten Satzes
Turau, Kapitel 3.6.1: genauer Beweis des Satzes wie oben (inkl. Induktionsverankerung)
Lang: Skript Berechenbarkeit und Komplexität, Kap. 4.2.3 (Greedy-Algorithmen für Matroide)

Berechnung maximaler Flüsse in q/s-Netzwerken

Begriffswelt

Def.: q/s-Netzwerk bzw. s/t-Netzwerk:

Graph (V,E) mit positiven Kantenkapazitäten c(e) für alle Kanten e
und einer ausgezeichneten Quelle (q bzw. s) und Senke (s bzw. t) in V

Def.: Fluss f: Funktion $E \rightarrow \mathbb{N}$ mit

- $0 \leq f(e) \leq c(e)$ für alle Kanten e
- $f(u,v) = - f(v,u)$ (falls (u,v) eine Kante in G ist)
- Die Summe aller Flüsse zu den Nachbarn von Knoten ≠ q,s ist immer Null.

Def.: Wert |f| eines Flusses:

Nettoausfluss aus q bzw. Nettoeinfluss in s (sind immer gleich)

Referenzen zum Nacharbeiten und Vertiefen:

Cormen, Kap. 26.1
Alt, Kap. 4.5.1
Turau, Kap. 6.1 (siehe auch Ausarbeitung und Vortrag Seminararbeit Claudia Padberg)

Algorithmik 8

Berechnung maximaler Flüsse in q/s-Netzwerken

Begriffswelt

Def.: Erweiterungsweg:

Weg von q zu s, in dem für jede Kante e eine der beiden Bedingungen gilt:

- Kapazität von e noch nicht erschöpft.
- positiver Fluss in die entgegengesetze Richtung

Def.: Restegraph G_f:

Für jede Kante (u,v) mit Restkapazität in G wird eine Kante $(u,v) \in G_f$ gebildet, deren Kapazität gleich dieser Restkapazität ist.

Für jede Kante (v,u) mit positivem Fluss in G wird eine Kante $(u,v) \in G_f$ gebildet, deren Kapazität gleich f(v,u) ist.

Satz 1: Einem Erweiterungsweg in G entspricht genau ein Weg von q nach s in G_f

Satz 2: Ein Fluss f kann um den Restfluss, d.h. die minimale Kapazität der Kante eines Erweiterungsweges in G_f erhöht werden.

Referenzen zum Nacharbeiten und Vertiefen:

Cormen, Kap. 26.1
Alt, Kap. 4.5.2
Turau, Kap. 6.1, 6.3 (Restegraph) (siehe auch Ausarbeitung und Vortrag Seminararbeit C. Padberg)

Berechnung maximaler Flüsse in q/s-Netzwerken

Begriffswelt

Def.: q/s-Schnitt (X,X):

Partition der Ecken von G, sodass $q \in X$ und $s \in X$

Def.: Kapazität c(X,X) eines q/s-Schnitts:

Summe der Kapazitäten der Kanten (u,v) mit $u \in X$ und $v \in X$

Def.: Fluss f(X,X) eines q/s-Schnitts:

Summe der Flüsse der Kanten (u,v) mit $u \in X$ und $v \in X$

Satz 1: Für jeden q/s-Schnitt (X,X) gilt: $|f| = f(X,X) – f(X,X)$

Satz 2: $|f| \leq \min \{c(X,X);\ (X,X)$ ist q/s-Schnitt$\}$

Referenzen zum Nacharbeiten und Vertiefen:

Cormen, Kap. 26.1
Turau, Kap. 6.1 (siehe auch Ausarbeitung und Vortrag Seminararbeit Claudia Padberg)

Algorithmik 8

Berechnung maximaler Flüsse in q/s-Netzwerken

Satz von Ford-Fulkerson (maximaler Fluss = minimaler Schnitt)

Folgende Bedingungen sind äquivalent:

- f ist ein maximaler Fluss in G
- Es gibt keinen Erweiterungsweg zu f in G
- Es gibt einen q/s-Schnitt (X,X) mit |f| = c(X,X)

Beweisaufbau:

Ringschluss:

1) => 2) trivial
2) => 3) gezeigt in Vorlesung (nach Cormen)
3) => 1) folgt aus Satz 2 in vorheriger Folie

Referenzen zum Nacharbeiten und Vertiefen:

Cormen, Kap. 26.2
Turau, Kap. 6.2 (eigener Beweis)

Algorithmik 8

Berechnung maximaler Flüsse in q/s-Netzwerken

Algorithmus von Edmonds-Karp: (Notation nach Alt)

1) Initialisiere f mit 0.
Repeat
    2a) Berechne G_f
    2b) Finde Erweiterungsweg in G_f **mit Breitensuche**
    3) Erhöhe f um den Restfluss aus dem Erweiterungsweg (Satz 2, Folie 4)
until kein Erweiterungsweg mehr vorhanden

Korrektheit: folgt aus Satz von Ford-Fulkerson

Laufzeit: $O(nm^2)$

Beweisskizze für Laufzeit:

Die internen Schleifenoperationen 2a), 2b) und 3) kosten O(m) Zeit (einfach einzusehen)

Es gibt insgesamt O(nm) Schleifendurchläufe (Details beim nächsten Mal):
Jeder Erweiterungsweg hat eine kritische Kante. Jede Kante kann höchstens O(n) mal kritisch werden. Es gibt m Kanten.

Referenzen zum Nacharbeiten und Vertiefen:

Cormen, Kap. 26.2
Alt, Kap. 4.5.4
Turau, Kap. 6.3 (mit Programmierdetails) (siehe auch Seminararbeit Claudia Padberg)