Handout zur Session vom 14. November 2005

John Frederic Schommer

November 23, 2005

1 Allgemein

In der letzten Session vom 7. November wurde gezeigt, dass sich ein optimaler Steinerbaum effizienter berechnen läßt als mit dem Dreyfus-Wagner- Algorithmus. Die Voraussetzung für die Korrektheit des im Paper *A Faster Algorithm For The Steiner Tree Problem* vorgestellten Algorithmus, siehe auch Abschnitt Komplexität, ist dabei die q-Granularität der Knotenmenge Z mit $q \in \mathbb{N}$, für die der optimale Steinerbaum berechnet werden soll. Fortführend wurde in dieser Session nun erarbeitet, dass sich für eine Knotenmenge Z, welche für ein gewähltes q nicht granular ist, eine Knotenmenge Z' finden läßt mit $Z \subseteq Z' \subseteq V[T]$, Z' q-granular, wobei die Mächtigkeit der neuen Menge Z' nach oben beschränkt ist in Abhängigkeit von der Terminalmenge Y. Der Beweis wurde mit Hilfe eines passenden Algorithmus gegeben. Im weiteren Verlauf der Session ging es um die Laufzeit des vorgestellllten verbesserten Algorithmus zur Berechnung von Steinerbäumen.

2 Lemma 3

Die Granularität der Terminalmenge Y kann im *worst case* bei $q = |Y|$ liegen. Hier würde der Dreyfus-Wagner-Algorithmus den optimalen Steinerbaum für den Graph berechnen, was den vorgestellten Algorithmus nutzlos machen würde. Für beliebig gewählte q kann aber die q-Granularität für Z mit $Y \subseteq Z \subseteq V[T]$ durch Hinzufügen geeigneter Knoten $v \in V[T]$ in die Terminalmenge Z hergestellt werden. Folgendes Lemma zeigt, dass die Mächtigkeit der so erweiterten Terminalmenge Z nach oben abgeschätzt werden kann:

Sei T ein opt. Steinerbaum für $Y \subseteq V[T]$, $q \in \mathbb{N}$ mit $2 \leq q \leq |Y|$.
Dann ex. ein $Z \subseteq V[T]$, $Y \subseteq Z$, mit $|Z| \leq |Y| + \frac{|Y|}{q-1}$ und $|V[R] \cap Z| \leq q$

für alle $R \in \mathcal{R}(T, Z)$.

Existiert also der optimale Steinerbaum T in dem betrachteten Graph, dann gibt es eine q-granulare Knotenmenge Z, für die $|Z| \leq |Y| + \frac{|Y|}{q-1}$ gilt. Sowohl das Lemma als auch der folgende Beweis setzen die Bekanntheit des optimalen Steinerbaums T voraus, zeigen also nur die Existenz der q-granularen Menge Z und liefern keine passende Expansionsvorschrift.

Folgender Algorithmus beweist unter den Bedingungen des Lemmas die Behauptung:

1. $Z := Y$
2. Solange $T(T, Z)$-Region R mit $2 \leq |V[R] \cap Z| \leq q$ enthält die nur mit einem Knoten v mit anderen (T, Z)-Regionen verbunden ist, lösche R unter Beibehaltung von v
3. Ist T eine einzelne (T, Z)-Region und q-granular gebe Z aus und stoppe
4. Wähle Wurzel r in T, setze $v := r$
5. Solange es einen Teilbaum mit mehr als $q-1$ Knoten von Z gibt dessen Wurzel w ein Kind von v ist setze $v := w$
6. $Z := Z \cup \{v\}$, gehe zu 2

Der Algorithmus läßt sich wie folgt beschreiben: Z wird durch Y initialisiert. Im ersten Schritt der äußeren Iteration werden alle Regionen, welche höchstens q Terminale enthalten und bereits das Granularitätskriterium erfüllen, aus T entfernt, wobei man den diese Region und die weitere Konförderation verbindenden Knoten v in T belässt. Wenn nach ersten Iterationsschritt T nur noch aus einer q-granularen Region besteht, ist der Algorithmus fertig und endet.

Sonst ist nun T eine nicht q-granulare Region und es wird ein Knoten $v \in V[T]$ wie folgt in Z zugefügt: In der inneren Iteration wird eine Wurzel v in T gewählt, und solange ein Teilbaum eines der Nachfolger w von v mehr als $q-1$ Knoten aus Z enthält, setze v auf den Nachfolger w. Wenn keiner der Teilbäume des Nachfolgers w das Kriterium mehr erfüllen, füge v in Z ein und beginne wieder im ersten Iterationsschritt der äußeren Iteration. Dies führt dazu, dass immer mehr q-granulare Regionen in der äußeren Iteration aus T entfernt werden, welche in der inneren Iteration entstanden sind.

Durch die zwei Ebenen, nämlich der Wurzel v und ihrem Nachfolger w wird sichergestellt, dass aus T eine Konförderation entsteht, in der *mindestens* eine Region genau q Knoten aus Z enthält, nämlich genau der Teilbaum des Nachfolgers w von v, welcher im letzten Iterationsschritt der inneren Iteration die Schleifenbedingung nicht mehr erfüllte, zusammen mit v. Dies ergibt sich aus der Iterationsvorschrift der inneren Iteration, denn erfüllt kein Teilbaum von w mehr die Bedingung, dann wird v zu Z zugefügt, wäre nun v ein Terminal gewesen, hätte der Algorithmus diesen Bereich von T bereits gelöscht, wäre w ein Terminal gewesen, würde noch eine Iteration ausgeführt. Da v weitere Teilbäume besitzen kann, welche alle höchstens $q-1$ Knoten aus Z besitzen, können eventuell in der nächsten äußeren Iteration *zusätzliche* Knoten aus T entfernt werden.

Der Algorithmus liefert also auch direkt die obere Schranke für die Kontenmenge Z: Beim Aufruf des Algorithmus ist $|Z| = |Y|$. In jeder äußeren Iteration werden *mindestens* $q-1$ Knoten $\in Z$ aus T entfernt, d.h. im *worst case*, wenn also immer genau $q-1$ Knoten entfernt werden, wird die äußere Iteration $\frac{|Y|}{q-1}$ mal durchlaufen bis T leer ist. In jeder Iteration kommt genau ein gewähltes v zu Z hinzu, also liegt die Mächtigkeit dann im *worst case* bei $|Y| + \frac{|Y|}{q-1}$, was natürlich eine oberer Schranke ist.

Eine Veranschaulichung des Algorithmus wird in dem Paper auf Seite 8 gegeben.

3 Komplexität

Schließlich wurde gegen Ende der Session noch über die Komplexität diskutiert, hier noch einmal der Algorithmus:

for all $Z' \subseteq Z$ such that $|Z'| \leq q$ **do**
  compute $S(Z')$ *via Dreyfus-Wagner*;
for all $Z' \subseteq Z$ such that $|Z'| > q$ **do**
  $S(Z') := G$;
  for all $Z'' \subseteq Z'$ and $v \in Z''$ such that $2 \leq |Z''| \leq q$ **do**
    if $l(S(Z''))+l(S(Z'-Z''\cup\{v\})) < l(S(Z'))$ **then**
      $S(Z') := S(Z'') \cup S(Z'-Z''\cup\{v\})$
return $S(Z)$

Für die erste *for*-Schleife erhält man die von Dreyfus-Wagner bekannte Komplexität, diesmal aber nur für alle Teilbäume mit $|Z| \leq q$ laut der Konstruktion des Algorithmus, welche binominal viele sind:

$$\sum_{i=1}^{q} \binom{|Z|}{i} 3^i poly(n)$$

Diesen Term kann man analytisch nach oben abschätzen, da $n \in \mathbb{N}$ beliebig ist, die Abschätzung also eine sogenannte unendliche Summe, greift hier das Quotientenkriterium für natürliche Reihen:

$$\sum_{i=1}^{q} \binom{|Z|}{i} 3^i poly(n) \leq \binom{|Z|}{q} 3^q poly(n)$$

Im zweiten Teil des Algorithmus berechnet man mögliche Konföderationen in Abhängigkeit der aktuellen Größe von q

$$\sum_{i=q+1}^{|Z|} \binom{|Z|}{i} \sum_{j=2}^{q} \binom{i}{j} poly(n)$$

Auch diese Komplexität läßt sich nach oben abschätzen, mit Hilfe des Quotientenkriteriums und der Potenzmenge von $|Z|$, welche bei Bildung von Teilmenge von Z natürlich eine obere Schranke ist, erhält man:

$$\sum_{i=q+1}^{|Z|} \binom{|Z|}{i} \sum_{j=2}^{q} \binom{i}{j} poly(n) \leq 2^{|Z|} \binom{|Z|}{q} poly(n)$$

Diese Komplexitäten müssten nun zusammengeführt werden, was im Paper mit Hilfe einer Stirlingabschätzung gelingt, aber innerhalb der Session nicht mehr besprochen wurde. Die formale Abschätzung führt zu einer Komplexität von

$$O(2 + \delta)^k * poly(n))$$

für ein wählbares δ.

Mit der Besprechung der Komplexität des vorgestellten Algorithmus endete die letzte Session über Steinerbäume.