Der Linearzeit MST-Algorithmus
Abstract

Maximilian Springenberg
Proseminar Randomisierte Algorithmen

In der Ausarbeitung werden wir uns mit dem Kapitel 10.3 aus dem Lehrbuch [1] auseinander setzen, in dem ein randomisierter Algorithmus für das minimale Spannbaum/ Spannwald Problem (MST/MSF) vorgestellt wird.
Der Linearzeit MST-Algorithmus ist ein randomisierter Ansatz, der das MST und MSF Problem in erwarteter linearer Laufzeit löst. Es gibt bereits ein deterministisches Verfahren von Chazelle, dass einen MST in $O(m * log\beta(m,n)), \beta(m,n) = \{i|log^{(i)}n \leq m/n\}, n = |V|, m = |E|$, wobei β die inverse Ackermann Funktion verwendet, berechnet. Unser Algorithmus wird jedoch einfacher zu implementieren sein, bietet eine im Erwartungswert bessere Laufzeit und der randomisierte Anteil des Algorithmus erzeugt eine akzeptable Approximation des MST/MSF mit im Erwartungswert n/p F-leichten Kanten.
Eine Kante aus G ist F-leicht, wenn sie hinsichtlich eines Waldes F weniger oder gleich stark gewichtet ist, wie die Kante mit maximalem Gewicht auf dem Pfad in F, der die Knoten der Kante verbindet. Ist das nicht der Fall, so ist die Kante F-schwer. Mittels F-leichter Kanten können wir daher neben der Summe von Kantengewichten ein Gütemaß für einen Wald hinsichtlich G definieren. Ferner wird das erkennen von F-schweren Kanten nützlich sein, wenn wir Kanten entfernen möchten die nicht im MST/MSF vorkommen.
Der aus dem Algorithmus resultierende MST/MSF wird nicht rein randomisiert konstruiert. So werden wir uns auch ein deterministisches Verfahren, nämlich die Borůvka-Phase zur Hilfe nehmen. Dies ist notwendig, da wir unseren Algorithmus rekursiv aufbauen möchten und dazu insbesondere die Kanten- und Knotenmenge des betrachteten Graphen in jedem rekursiven Aufruf reduzieren müssen. Eine Borůvka-Phase nutzt aus, dass für jeden Knoten die minimal gewichtete inzidente Kante im MST/MSF vorkommen muss. Es werden also in einer Borůvka-Phase Kanten mit den minimalen Gewichten markiert und die daraus entstehenden verbundenen Komponenten auf je einen Knoten reduziert. Im Worst-Case erhalten wir ein perfektes Matching und reduzieren die Knotenmenge auf $n/2$.
Die randomisierte Konstruktion erfolgt über randomisierte Stichproben des Graphen. Dabei wird in unserem Fall jede Kante mit der Wahrscheinlichkeit $p = 0,5$ in die Stichprobe $G(p)$ inkludiert. Es ist leicht ersichtlich, dass hierbei die Kantenmenge im Erwartungswert halbiert wird. Wie wir im Teil zur Güte einer randomisierten Stichprobe sehen werden hat G dabei hinsichtlich des MSF von $G(p)$ im Erwartungswert n/p F-leichte Kanten. Für den Beweis werden wir eine Folge von Zufallsexperimenten definieren und anhand des Ablaufs derer zeigen, dass die negative Binomialverteilung mit dem Erwartungswert n/p die Anzahl von F-leichten Kanten in G stochastisch dominiert.
Der Algorithmus wird drei Borůvka-Phasen ausführen. Die Rekursion terminiert, wenn nach einer Borůvka Phase der Graph unverändert bleibt, oder G leer ist. Danach werden wir eine Stichprobe konstruieren und beginnen die erste Rekursion. Über das Resultat der Rekursion werden F-schwere Kanten ermittelt, diese aus G entfernt und ein letzter rekursiver Aufruf getätigt. Intuitiv is durch das Entfernen der F-schweren Kanten die Wahrscheinlichkeit, dass F-schwere Kanten nach dem zweiten Aufruf im Wald enthalten sind geringer.
Interessant ist hierbei, dass wir zwei sucessive rekursive Aufrufe verwenden, jedoch eine lineare Laufzeit erwarten. Dies ist darin begründet, dass wir den ersten rekursiven Aufruf mit $n/8$ Knoten, $m/2$ Kanten und den zweiten rekursiven Aufruf mit $n/8$ Knoten und im Erwartungswert $\frac{n/8}{p=0,5} = n/4$ Kanten ausführen. In unserer Laufzeitanalyse zum Algorithmus werden wir feststellen, dass wir die Rekursionsgleichung mit $T(n,m) \leq T(n/8, m/2) + T(n/8, n/4) + c(n+m) \leq 2 * c(n+m)$ abschätzen können.

[1] Motwani, R., Raghavan, P. : *Randomized Algorithms.* Cambridge : Cambridge University Press 1995, Kapitel 10.3.