

Der Linearzeit MST-Algorithmus

Der schnellste Algorithmus für das MST/ MSF Problem

Maximilian Springenberg

Proseminar Randomisierte Algorithmen

0 Konventionen

In dieser Ausarbeitung wird sich an die üblichen Konventionen zur Notation von Variablen in Graphen orientiert. So definieren wir einen Graphen als $G = (V, E)$, mit der Anzahl von Knoten $n = |V|$ und Kanten $m = |E|$. Zu G zugehörige Kanten und Knotenmengen können auch durch V_G, E_G als solche gekennzeichnet werden. Für jeden Knoten v nehmen wir an, dass eine Adjazenzliste $adj(v)$ von adjazenten Knoten für ihn vorhanden ist. Ferner betrachten wir ungerichtete gewichtete Graphen und bezeichnen die Gewichtungsfunktion als $w : E \rightarrow \mathbb{R}$. Wir werden auch Wege, bzw. Pfade betrachten. Dazu sei $P(\{v, u\})$, eine Funktion die in einem Baum den Pfad von v nach u als Knotenfolge angibt. Da es für uns handlicher sein wird gleich die Kanten angegeben zu haben definieren wir P_e als die Folge von Kanten auf dem Pfad.

1 Motivation

1.1 MST und MSF

Der minimale Spannbaum, oder auch MST, stellt einen azyklischen zusammenhängenden Teilgraph aus G , der alle Knoten verbindet und dessen Summe von Kantengewichte $\sum_{e \in E_{MST}} w(e)$ minimal ist dar.

Ist G selbst nicht zusammenhängend, so werden wir einen minimalen Spannwald als nächst beste Lösung betrachten. Ein minimaler Spannwald besteht aus einer Sammlung von minimalen Spannbäumen für die verbundenen Komponenten in G .

1.2 Vorteile eines nichtdeterministischen Ansatz

Wir werden in dieser Ausarbeitung zum Kapitel 10.3 aus dem Buch „Randomized Algorithms“ von Motwani, R., Raghavan, P. einen randomisierten Ansatz für einen Algorithmus, der einen minimalen Spannbaum in erwarteter Linearzeit approximiert betrachten.

Das MST Problem ist in P , und es sind bereits deterministische Algorithmen wie die von Prim, Kruskal, Borůvka bekannt, die mit einer Worst-Case Laufzeitschranke von $O(m * \log(n))$ das Problem lösen. Zudem existiert der Algorithmus von Bernard Chazelle, für den eine Worst-Case Laufzeitschranke von $O(m * \log \beta(m, n))$ bekannt ist, wobei mit $\beta(m, n) = \{i | \log^{(i)} n \leq m/n\}$ die inverse Ackermann Funktion verwendet wird. $\log^{(i)} n$ ist hierbei

die i -te Anwendung von \log auf n . Dementsprechend steigt die Funktion β so schwach, dass die aus ihr resultierenden Faktoren für die Worst-Case Laufzeit hinsichtlich der Größe von Graphen in der Praxis als nahezu konstant angesehen werden.

Wozu dient also ein nichtdeterministischer Algorithmus, der im Erwartungswert linear verläuft? - Die Antwort auf diese Frage kann sowohl durch die komplexe Implementierung des Algorithmus von Chazelle, als auch der Stabilität, bzw. Güte, des vorgestellten Algorithmus hinsichtlich seiner Laufzeit und nicht zuletzt durch das Betrachten des Algorithmus als akademisches Beispiel für kreative Laufzeitverbesserungen begründet werden.

2 F -schwere/-leichte Kanten

F -schwere und -leichte Kanten sind ein wesentlicher Bestandteil des MST-Algorithmus. Mittels der Identifizierung von Kanten in einem Graphen G als F -schwer hinsichtlich eines Waldes F in G kann bereits entschieden werden, dass diese Kante nicht im MST von G enthalten ist. Der umkehrschluss für F -leichte Kanten gilt jedoch nicht, wie wir sehen werden.

2.1 Definition

Wir betrachten neben der Gewichtungsfunktion w nun die Funktion w_F , mit

$$w_F(\{u, v\}) = \begin{cases} \infty, & P(\{u, v\}) = \emptyset \\ \max\{w(P_e(\{u, v\}))\}, & \text{sonst} \end{cases}$$

, wobei $w(P_e(\{u, v\}))$ bedeutet, dass w auf alle Kanten des Pfades angewandt wurde.

w_F , gibt also das Kantengewicht der Kante mit maximalen Gewicht auf dem Pfad von u nach v in F aus. Sollte dieser Pfad nicht existieren, so nehmen wir an, dass diese Kante unendlich schwer ist.

Ist das Gewicht einer Kante $w(e)$ echt größer als das maximale Gewicht auf dem Pfad $P_e(e)$ in F , bzw. $w(e) > w_F(e)$, so bezeichnen wir sie als F -schwer. Sonst ist sie F -leicht.

2.2 Verifikation durch F -schwere/-leichte Kanten

Es liegt nahe, dass F kein MSF in G ist, wenn eine Kante $\{u, v\}$ existiert dessen gewicht echt kleiner als das des maximalen Gewichts auf dem Pfad $P_e(\{u, v\})$ in F ist. Würde man annehmen, dass F ein MSF wäre, so könnte man E_F um die Kante $\{u, v\}$ erweitern und den dadurch entstandenen Zyklus mittels entfernen der Kante maximalen Gewichts auf dem Pfad $P_e(\{u, v\})$ lösen. Dadurch hätte man die Zusammenhängende Komponente als solche gewahrt und eine schwerere durch eine leichtere Kante substituiert. Folglich hätte man einen leichteren Wald und F wäre damit kein MSF.

Diese Erkenntnis reicht bereits aus um einen Verifikationsalgorithmus zum MST und MSF Problem zu konstruieren, der lediglich über alle Kanten iteriert und schaut ob die einzigen F -leichten Kanten die aus F selbst sind.

2.3 Informationsgewinn durch F -schwere Kanten

Wir haben bereits erwähnt, dass F -schwere Kanten nicht in einem MSF, bzw. MST enthalten sein können. Dies lässt sich durch folgenden Beweis verdeutlichen.

Wir können also sämtliche F -schwere Kanten für das erfassen des MST ignorieren, bzw. sogar aus G eliminieren.

Diese Erkenntnis nimmt sich auch der MST-Algorithmus zum Nutzen. In gewisser Hinsicht werden wir den umgekehrten Ansatz von Kruskal, welcher Kanten nach gewichten aufsteigend sortiert und immer nur F -leichte Kanten hinzunimmt, durch das erfassen von F -schweren Kanten über Stichproben von Wäldern in G verfolgen.

3 Reduzieren des Graphen

Wir werden einen rekursiven Algorithmus konstruieren, der genau dann aufhört, wenn nur noch ein Knoten in Graphen übrig bleibt. Folglich müssen wir bei jedem rekursiven Aufruf unseren Graphen reduzieren. Wir werden in diesem Teil betrachten, wie wir durch Verwendung von Borůvka-Phasen die Knoten des Graphen reduzieren können und anschließend durch randomisierte Stichproben zuzüglich eine Methode kennen lernen um die Anzahl von Kanten zu reduzieren. Letzteres ermöglicht uns erst eine erwartete lineare Laufzeit, da die Rekursionstiefe durch das alleinige reduzieren von Knoten für unseren Anspruch zu groß ist.

3.1 Borůvka-Phasen

Borůvka Phasen Beruhen auf der Erkenntnis, dass in einem beliebigen ungerichteten Graphen G für jeden Knoten $v \in V$ die inzidente Kante mit minimaler Gewichtung $e_{v,min} := \{v, u\}, u \in adj(v) : \nexists e' = \{v, u'\}, u' \in adj(v) : w(e') < w(e)$ im MST von G enthalten ist.

Ferner bedeutet das, dass wir durch die für die Kontraktion markierten Kanten E_{min} nach einer Borůvka-Phase einen Wald F in G , mit Kanten aus einem MST von G . Dies wird für uns insbesondere dann interessant, wenn wir rekursiv einen Wald aufbauen möchten.

Darauf aufbauend betrachten wir nun den Ablauf einer Borůvka-Phasen:

1. Markiere inzidente Kanten E_{min} mit minimaler Gewichtung
2. Bestimme die verbundenen Komponenten in $G' = (V, E_{min})$
3. Ersetze jede verbundene Komponente durch einen sie repräsentierenden Knoten in G' und erhalte den Graphen G''
4. Entferne alle Selbstschleifen in G''

Wir stellen fest, dass eine Borůvka-Phase die Menge an Knoten in G auf maximal die Hälfte reduziert.

3.2 Randomisierte Stichproben

3.3 Güte einer randomisierten Stichprobe

Im folgenden werden wir zeigen, dass Anzahl von F -leichten Kanten in G bezüglich des MSF F_p von $G(p)$ im Erwartungswert nach oben durch n/p beschränkt ist. Ferner zeigen wir also, dass die F -leichten Kanten in G der negativen Binomialverteilung mit Parametern n und p entsprechen.

Zunächst möchten wir festlegen, dass im folgenden alle Kanten aus G nach ihrer Gewichtung aufsteigend sortiert betrachtet werden. Dies wird für uns insbesondere dann handlich sein, wenn wir über die zu betrachtende Kante wissen möchten, ob sie F -leicht ist.

Nun da wir uns auf eine Iteration der Kanten festgelegt haben möchten wir $G(p = 0,5)$ wie üblich durch das

hinzunehmen der zu betrachtenden Kante mit der Wahrscheinlichkeit p konstruieren. Während wir $G(p)$ konstruieren können wir auch simultan, bzw. „online“ den MSF F_{MSF} zu $G(p)$ konstruieren. Dies ist beispielsweise durch einen Ansatz wie den von Kruskal möglich, bei dem wir genau dann eine Kante $\{u, v\}$ in F_{MSF} aufnehmen, wenn $\{u, v\}$ in verschiedenen verbundenen Komponenten in F_{MSF} sind, wobei wir F_{MSF} mit allen Knoten aus G und einer leeren Kantenmenge initialisieren. Die Korrektheit von F_{MSF} folgt aus der Sortierung der Kanten. Interessant ist, dass dies gleichbedeutend damit ist, dass Kante $\{u, v\}$ aus G genau dann F_{MSF} -leicht ist, wenn ihre u, v in verschiedenen verbundenen Komponenten in F_{MSF} liegen.

Wir können bereits folgende Eigenschaften der Konsistenz unseres Verfahren beobachten: (i) Ob die zu betrachtende Kante F_{MSF} -leicht oder -schwer ist ist alleinig von den vorhergehenden Zufallsexperimenten abhängig, (ii) es werden keine Kanten aus F_{MSF} entfernt, (iii) eine Kante ist genau dann nach dem i -ten Zufallsexperiment F_{MSF} -leicht, wenn sie auch vor dem i -ten Zufallsexperiment F_{MSF} -leicht war.

Definieren wir nun den Begriff von Phasen in unserem Zufallsexperiment. Die k -te Phase unseres Verfahrens beginne, sobald $|E_{F_{MSF}}| = k - 1$ Kanten vorhanden sind und ende bei $|E_{F_{MSF}}| = k$. Wir befassen uns in einer Phase also unter anderem mit der Anzahl von Zufallsexperimenten bis eine Kante hinzugenommen wird. Insbesondere ist die hinzugenommene Kante per definition unseres Verfahrens F_{MSF} leicht.

4 der MST-Algorithmus
