Vorlesungsmitschrift

Algorithmen und Berechenbarkeit

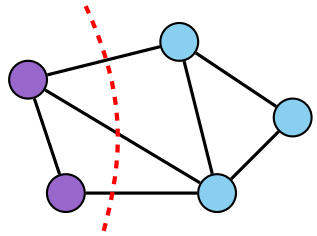
Vorlesung 02

Letztes Update: 2018/02/25 - 11:15 Uhr

Min-Cut-Problem

Cut

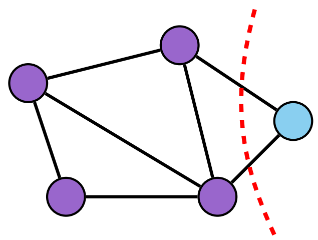
Ein Cut im Graphen unterteilt einen Graphen in zwei Hälften. Das heißt, es gibt Kanten, bei denen ein Knoten in einer Hälfte und bei denen der andere Knoten in der anderen Hälfte ist.

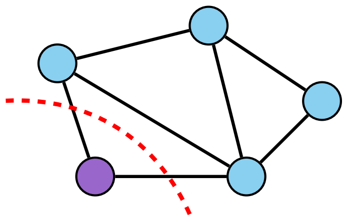


Ein Cut mit Wert 3

Min-Cut

Der Min-Cut beschreibt den Cut, bei dem am wenigsten Kanten „geschnitten" werden müssen, um den Graphen in zwei Hälften zu teilen.

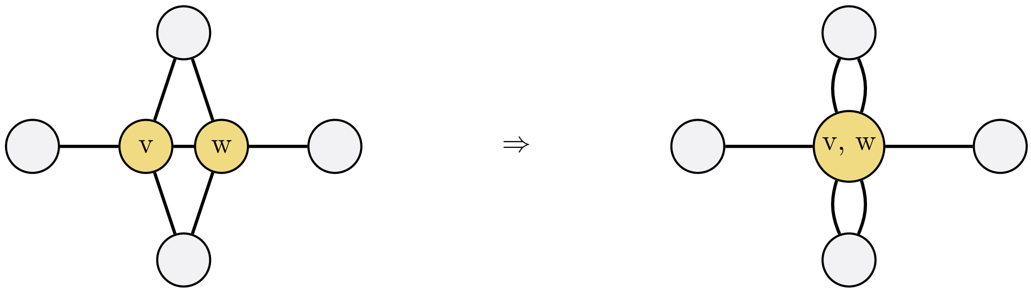




Min-Cut mit Wert 2 (hier nicht eindeutig)

Kargers Min-Cut-Algorithmus

Kargers-Min-Cut-Algorithmus beschreibt einen randomisierten Algorithmus, der mit einer Wahrscheinlichkeit von $\frac{1}{n^2}$ das richtige Ergebnis berechnet. Als zentrale Operation dient hier die Kantenkontraktion:



Kantenkontraktion der Knoten v, w

Der Algorithmus lässt sich wie folgt beschreiben:

```
for i = 1 to n − 2:
    Kontrahiere zufällige Kante
return Kantenmenge,
    die einem der beiden verbleibenden Knoten entspricht
```

Es bleiben zwei große Knoten mit einigen Kanten zwischen ihnen übrig. Der errechnete Min-Cut ist die Anzahl der Kanten zwischen diesen Knoten.

Analyse von Kargers Min-Cut-Algorithmus

Annahme: Min-Cut ist eindeutig.
Beobachtung: Algorithmus berechnet genau dann das richtige Ergebnis, wenn er nie eine der Min-Cut-Kanten kontrahiert.
Sei k die Größe des Min-Cuts, m die Anzahl der Kanten. Die Wahrscheinlichkeit im ersten Kontraktionsschritt einen Fehler zu machen, beträgt $\frac{k}{m}$, die Wahrscheinlichkeit keinen Fehler zu machen $1 - \frac{k}{m}$.

Satz: Betrachte einen Multigraphen (Mehrfachkanten zwischen Knoten erlaubt) $G(V, E)$ mit einem Min-Cut mit Wert k. Dann gilt: G hat mindestens $\frac{k \cdot n}{2}$ Kanten.

Beweis: Jeder Knoten hat Grad $\geq k$.

$$\#\text{Kanten} = \frac{\sum(\text{grad}(v))}{2} \geq \frac{k \cdot n}{2}$$

Sei E_i das Ereignis, dass im i-ten Kontraktionsschritt keine Min-Cut-Kante erwischt wurde. Wir wissen, dass die Wahrscheinlichkeit dafür $1 - \frac{k}{m} \geq 1 - \frac{2}{n}$ ($m \geq \frac{k \cdot n}{2}$) beträgt.
Falls E_1 eingetroffen ist, existieren vor dem zweiten Schritt mindestens $\frac{k \cdot (n-1)}{2}$ Kanten. Daraus folgt dass die Wahrscheinlichkeit, im zweiten Schritt keinen Fehler zu machen $P_r(E_2|E_1) \geq 1 - \frac{k}{m} \geq 1 - \frac{2}{n-1}$ beträgt, falls E_1 eingetreten ist.
...
Falls $E_1, E_2, ..., E_i$ eingetroffen sind, existieren vor dem $i + 1$-Schritt mindestens $\frac{k \cdot (n-i)}{2}$ Kanten: $P_r(E_{i+1}) \geq 1 - \frac{2}{n-i}$.

$$P_r(\bigcap_{i+1}^{n-2}) = P_r(E_1) \cdot P_r(E_2|E_1) \cdot P_r(E_3|E_1 \wedge E_2)...$$

$$\geq \prod_{i=1}^{n-2} 1 - \frac{2}{n-(i+1)} = \prod_{i=1}^{n-2} \frac{n-i+1-2}{n-i+1} = \prod_{i=1}^{n-2} \frac{n-i-1}{n-i+1}$$

$$= \frac{\cancel{n-2}}{n} \cdot \frac{\cancel{n-3}}{n-1} \cdot \frac{n-4}{\cancel{n-2}} \cdot \frac{n-5}{\cancel{n-3}} ... = \frac{2}{n \cdot (n-1)}$$

Die Wahrscheinlichkeit, dass der Algorithmus korrekt arbeitet, beträgt damit

$$\geq \frac{2}{n \cdot (n-1)}$$

Die Erfolgswahrscheinlichkeit kann beliebig erhöht werden. Dafür muss der Algorithmus so oft wie gewünscht wiederholt werden. Der kleinste Wert, der in Folge dieser Wiederholungen herauskommt, ist der Min-Cut. Die Wahrscheinlichkeit, in jedem der r Versuche **nicht** den Min-Cut zu finden, beträgt

$$\leq (1 - \frac{2}{n \cdot (n-1)})^r \leq e^{-\frac{2}{n \cdot (n-1)}} = e^{\frac{-2r}{n \cdot (n-1)}}$$

Wähle zum Beispiel $r = \frac{n \cdot (n-1)}{2}$. Die Wahrscheinlichkeit für falsch ist $\leq \frac{1}{e} \approx 0,36$.
Für $r \approx n^2$ bekommen wir eine **konstante** Erfolgswahrscheinlichkeit, für $r \approx n^2 \cdot n \cdot \log(n)$ ist das Ergebnis mit hoher Wahrscheinlichkeit korrekt, d.h. $\leq \frac{1}{n^2}$ für falsch.