Algorithmen für verteilte Systeme **Universität Salzburg**
Sommersemester 2018 **FB Computerwissenschaften**

Vorlesung 5: 4. Juni 2018

Vortrag: Sebastian Krinninger *Mitschrift: Sara Seidl*

Bei den zuvor betrachteten Problemen (Broadcast, kürzeste Wege) handelt es sich um *globale* Problemen. Algorithmen zur Lösung dieser Art von Problemen sind vom Durchmesser D abhängig und benötigen $\Omega(D)$ Runden.

Im Gegensatz dazu stehen *lokale* Probleme, welche unagängig vom Durchmesser berechnet werden können. Für die benötigte Rundenanzahl gilt $o(n)$. Beispiele für lokale Probleme wären Maximal Independent Set, Spanner-Berechnung und Maximal Matching.

5.1 Definitionen

Definition 5.1 (Independent Set)**.** *In einem ungerichteten Graph $G = (V, E)$ ist ein* ***Independent Set*** *eine Teilmenge von Knoten $U \subseteq V$, in der keine zwei Knoten benachbart sind.*

Man spricht von einem ***Maximal*** *Independent Set*, wenn kein Knoten zu U hinzugefügt werden kann, sodass U ein Independent Set bleibt.
Ein ***Maximum*** *Independent Set* U hat größtmögliche Kardinalität unter allen Independent Sets

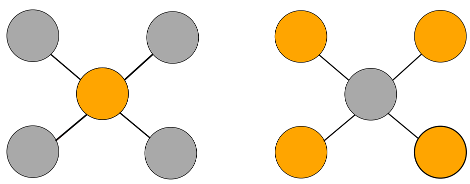


Abbildung 5.1: Die farbig markierten Knoten bilden sowohl links als auch rechts ein Maximal Independent Set. Nur das rechte Independent Set ist jedoch ein Maximum Independent Set.

5.2 Sequentieller Greedy-Algorithmus

Es werden alle Knoten in beliebiger Reihenfolge betrachtet. Ein betrachteter Knoten v wird nur zu U hinzugefügt, wenn v keinen Nachbarn $u \in U$ hat.

$U \leftarrow \emptyset$
foreach $v \in V$ **do**
  if $\forall u \in U : (u, v) \notin E$ **then**
    $U \leftarrow U \cup \{v\}$
  end
end

Laufzeit

Im **RAM-Model**, bei geeigneter Implementierung: $O(m+n)$
Im **CONGEST Modell**, im schlechtesten Fall: $\Omega(n)$

5.3 Verteilter Algorithmus nach [MRS+11]

Ziel ist, dass jeder Knoten weiß, ob er in U ist oder nicht. Der Ablauf pro Knoten folgt 3 Schritten:

1. Wähle **Zufallszahl:**
   Knoten v wählt uniforme reelle Zufallszahl $r(v) \in [0,1($ und sendet diese an alle Nachbarknoten u
2. Triff **Entscheidung:**
   Nachdem Knoten v die Nachrichten $r(u)$ von allen Nachbarknoten empfangen hat, kann v sich entscheiden Teil von U zu werden (wenn $r(v) < r(u)$ für jeden Nachbar u). Wenn $v \in U$ werden alle Nachbarknoten drüber informiert und v terminiert.
3. **Empfangsrunde:**
   Empfängt Knoten v eine Nachricht über einen zu U hinzugefügte Nachbar, dann terminiert v.

```
while v nicht terminiert do
    Wähle uniforme reelle Zufallszahl r(v) ∈ [0,1(
    Sende r(v) an alle Nachbarn
    Empfange r(u) von jedem Nachbar u
    if r(v) < r(u) für jeden Nachbar u then
        Füge v zur Menge U hinzu
        Benachrichtige Nachbarn, dass v ∈ U
        Terminiere v
    end
    Empfange Nachrichten über zu U hinzugefügte Nachbarn
    if mindestens ein Nachbar wurde zu U hinzugefügt then
        Terminiere v
    end
end
```

5.3.1 Korrektheit

Wir wollen zeigen, dass dieser Algorithmus ein Maximal Independent Set berechnet. Dazu beweisen wir zuerst die *Independent Set* Eigenschaft und anschließend die *Maximalität.*

z.z Independent Set

Beweis durch Induktion über Phasen.
Induktionshypothese: Knoten in U sind Independent Set und Menge der terminierten

Knoten besteht aus U und Nachbarn von U

IB: Am Anfang gilt die **IH** trivialerweise.

IS: Jeder Knoten, der zu U hinzugefügt wird,

- hat keine Nachbarn in U aus der aktuellen Phase:
  v wird nur Teil von U, wenn $r(v) < r(u)$ für jeden Nachbar u gilt. Dann folgt aber auch, dass die Gleichung für keinen Nachbarn u von v gilt und daher $u \notin U$ für alle Nachbarn u.
- und keine Nachbarn in U aus einer früheren Phase:
  Angenommen v hat Nachbarn $u \in U$ aus einer vorherigen Runde j. Dann wurde v in Runde j über das Hinzufügen von u informiert und ist terminiert (Widerspruch).

z.z Maximalität

Man betrachte die Terminierung eines Knoten. Für einen beliebegen terminierten Knoten v gilt:

- $v \in U$ oder
- $u \in U$ wobei u ein Nachbar von v ist

Sind also alle Knoten terminiert, gilt dies für jeden Knoten v. Daher gibt es keinen Knoten v, der noch zu U hinzugefügt werden könnte ohne die Independent Set Eigenschaft zu verletzen.

z.z Terminierung

Die Terminierung dieses Algorithmus ist auf Grund der Zufallskomponente nur mit *Wahrscheinlichkeit* 1 garantiert. (*Achtung:* Wahrscheinlichkeit 1 $\neq$ immer)

Die Wahrscheinlichkeit, dass zwei zufällig gewählte reelle Zahlen gleich sind $= 0$ und damit ist die Wahrscheinlichkeit, dass alle gewählten Zahlen unterschiedlich sind $= 1$. Es gibt daher in jeder Runde einen eindeutigen Knoten, der die kleinste Zufallszahl wählt. Dieser Knoten wird zu U hinzugefügt und terminiert. Nach endlich vielen Runden sind daher alle Knoten terminiert. Das Anwenden der Union Bound gibt uns Terminierung mit Wahrscheinlichkeit 1.

Reminder (Union Bound): Die Wahrscheinlichkeit, das eines von abzählbar vielen Ereignissen eintritt, ist maximal die Summe der Einzelwahrscheinlichkeit aller Ereignisse.

5.3.2 Laufzeit

Überlegung:
Betrachten wir den Fall, dass in jeder Phase genau 1 Knoten v Fortschritt gemacht wird. In dieser Phase terminiert v sowie alle Nachbarn von v. Somit ergibt sich für jede Runde eine Reduktion der Kanten im Subgraph der noch aktiven Knoten.

Wir treffen hier die *idealisierte Annahme*, dass das Senden von Zufallszahlen in $O(1)$ Runden passiert und somit jede Phase $O(1)$ Runden dauert.
Können wir also zeigen, dass sich # Kanten in jeder Phase um mindestens die Hälfte reduziert (mit hoher Wahrscheinlichkeit), dann

- werden $O(\log n)$ Phasen benötigt
- folgt aus obiger Annahme eine Laufzeit von $O(\log n)$

Implementierungsdetail: Für das CONGEST Modell reicht es aus, jeweils nur $O(\log n)$ Bits der Zufallszahlen zu übertragen (s.u.).

Analyse:
Betrachten wir also zu Beginn den *Erwartungswert* der entfernten Kanten in einer Phase. Wir bedienen uns folgender Analysetricks:

- Jede Kante wird als zwei gerichtete Kanten betrachtet
- Die Anzahl der entfernten Kanten wird systematisch unterschätzt

Zudem führen wir für die Analyse noch eine Dominanzbeziehung unter den Knoten ein:

Definition 5.2 (Dominanzbeziehung)**.** *u* ***dominiert*** *v ($u \twoheadrightarrow v$) falls:*

1. *$r(u) < r(v')$ für alle Nachbarn v' von u und*
2. *$r(u) < r(w)$ für alle Nachbarn $w \neq u$ von v*

Für die Wahrscheinlichkeit „u dominiert v“ gilt nun:

$$\Pr[u \twoheadrightarrow v] \geq \frac{1}{\deg(u) + \deg(v)}$$

Begründung:

- Anzahl Knoten in Nachbarschaften von u und v: $\leq \deg(u) + \deg(v)$ (Obere Schranke)
- Zufallszahlen $r(\cdot)$ induzieren uniform zufällige Permutation π von V
- Wahrscheinlichkeit erster in Permutation zu sein: 1/#Knoten

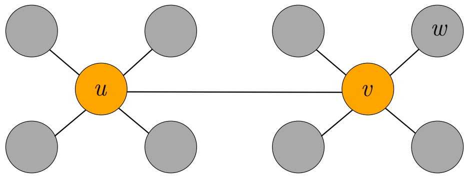
Wir können nun das entfernen von Kanten (u, v) als $\sum$ über Dominanzbeziehungen schreiben. Die # entfernter Kanten entspricht mindestens der # an Kanten in denen $u \twoheadrightarrow v$. Dazu definieren wir:

- E: Anzahl ungerichteter Kanten am Beginn der aktuellen Phase
- X: Zufallsvariable für Anzahl entfernter *gerichteter* Kanten

- $X_{(u \twoheadrightarrow v)}$: Anzahl Nachbarkanten von v, die wegen u entfernt werden

$$X_{(u \twoheadrightarrow v)} \begin{cases} \deg(v) & \text{für } u \twoheadrightarrow v \\ 0 & \text{sonst} \end{cases}$$

Anmerkung: Die zweite Eigenschaft der Dominanz Beziehung stellt sicher, dass v wegen u entfernt wird, und nicht etwa durch einen Nachbarn w von v. Somit wird keine entfernte Kante doppelt gezählt. (eindeutig einem $X_{(u \twoheadrightarrow v)}$ zuordbar)



$$\begin{aligned}
\mathbb{E}[X] &\geq \sum_{\{u,v\}\in E} \Big(\mathbb{E}[X_{(u \twoheadrightarrow v)}] + \mathbb{E}[X_{(v \twoheadrightarrow u)}]\Big) \\
&= \sum_{\{u,v\}\in E} \Big(\Pr[X_{(u \twoheadrightarrow v)}] \cdot \deg(v) + \Pr[X_{(v \twoheadrightarrow u)}] \cdot \deg(u)\Big) \\
&\geq \sum_{\{u,v\}\in E} \left(\frac{\deg(v)}{\deg(u)+\deg(v)} + \frac{\deg(u)}{\deg(v)+\deg(u)}\right) \\
&= \sum_{\{u,v\}\in E} 1 = |E|
\end{aligned}$$

In Erwartung ist die # der *gerichteten* Kanten die in einer Phase entfernt werden $|E|$. Da es für jede ungerichtete Kante zwei gerichtete Kanten gibt, entspricht das $\frac{|E|}{2}$ *ungerichteten* Kanten.

Lemma 5.3. *In jeder Phase wird in Erwartung wird die Hälfte der Kanten entfernt.*

Der Erwartungswert allein ist eine schwache Aussage. Wir wollen eine stärkere Aussage machen (Bound bezüglich Wahrscheinlichkeit an entfernten Kanten) und damit eine Bound für die Laufzeit bekommen. Dazu brauchen wir zunächst ein weiteres Standard Tool.

Theorem 5.4 (Markov Bound)**.** *Sei Y eine nicht-negative Zufallsvariable und $\alpha > 0$. Dann gilt*

$$\Pr[Y \geq \alpha\,\mathbb{E}[Y]] \leq \frac{1}{\alpha}\,.$$

Y: Zufallsvariable für Anzahl der nicht-entfernten Kanten

Wir wissen: $\mathbb{E}[Y] \leq \frac{|E|}{2}$

Wir wollen: Wahrscheinlichkeit enternter Kanten mit $\frac{1}{3} \cdot |E|$ unterschätzen

Somit:

$$\Pr\left[Y \geq \frac{2}{3} \cdot |E|\right] \leq \Pr\left[Y \geq \frac{4}{3} \cdot \mathbb{E}[Y]\right] \leq \frac{3}{4}$$

Lemma 5.5. *Mit Wahrscheinlichkeit* $\geq \frac{1}{4}$ *werden mindestens* $\frac{|E|}{3}$ *Kanten entfernt.*

Nun können wir mit Chernoff-Bound eine Aussage über die Laufzeit machen, die mit hoher Wahrscheinlichkeit gilt. Wir werden zeigen das folgendes Lemma gilt.

Lemma 5.6. *Mit hoher Wahrscheinlichkeit terminiert Algorithmus nach* $32\lceil c \log n \rceil$ *Phasen.*

Die Idee hier ist, Phasen als gute bzw. schlechte Phasen zu labeln. Wir führen dafür eine binäre Zufallsvariable ein:

$$Z_i = \begin{cases} 1 & \text{falls in Phase } i \text{ mindestens } 1/3 \text{ der Kanten entfernt werden} \\ 0 & \text{andernfalls} \end{cases}$$

Damit gibt $Z = \sum_{i=1}^{32\lceil c \log n \rceil} Z_i$ die #guter Phasen an.

Aus Markov Bound folgt:

$$\mathbb{E}[Z] \geq 32\lceil c \log n \rceil \cdot \frac{1}{4}$$

Nach k guten ($Z_i = 1$) Phasen:

$$\#\text{Kanten} \leq \frac{m}{3^k} \leq \frac{n^2}{3^k}$$

Für # Kanten ≤ 1:

$$\frac{n^2}{3^k} = 1 \implies 3^k = n^2 \implies k \cdot \log 3 = 2 \cdot \log n \implies k = \frac{2 \cdot \log n}{\log 3}$$

Nach $4 \log n \geq \frac{2 \log n}{\log 3} + 1$ guten Phasen sind daher alle Kanten entfernt.

$$\Pr[Z < 4 \log n] \leq \Pr\left[Z < \frac{1}{2} \cdot \mathbb{E}[Z]\right] \leq \frac{1}{e^{[\mathbb{E}[Z]/8]}} \leq \frac{1}{e^{[8c \log n/8]}} = \frac{1}{e^{c \log n}} = \frac{1}{n^c}$$

$\implies$ nach $O(\log n)$ Phasen sind alle Kanten und somit auch alle Knoten entfernt.

Markov Bound vs Chernoff Bound
Die Chernoff Bound ist stärker(Aussagen abhängig von Länge der Phasen, mit hoher Wahrscheinichkeit). Die Markov Bound gibt nur konstante Wahrscheinlichkeit an.

Bitkomplexität
Ziel: Für jede Kante $\{u, v\}$, entscheide, ob $r(u) < r(v)$ oder $r(u) > r(v)$
Beim Senden der Zufallszahlen darf nicht vergessen werden, dass im CONGEST-Model nur Nachrichtengröße $O(\log n)$ möglich ist.

Idee: Die Zahl als unendlichen String zufälliger Bits (Münzwurf unbiased) darzustellen.
$\implies$ Für eine Entscheidung reicht es, die Bits bis zur ersten unterschiedlichen Stelle zu senden. Wir wollen zeigen, dass es reicht, die ersten $(c + 3)\lceil \log n \rceil$ zu senden.

Wahrscheinlichkeit, dass die ersten $(c+3)\lceil \log n \rceil$ Bits gleich sind:

$$\left(\frac{1}{2}\right)^{(c+3)\lceil \log n \rceil} \leq \left(\frac{1}{2}\right)^{(c+3)\log n} = \frac{1}{n^{c+3}}$$

Anmerkung: $\frac{1}{2}$, da mögliche Kombinationen $\{\underline{00},01,10,\underline{11}\}$

Wahrscheinlichkeit, dass für irgendeines der $\leq n^2$ Paare in den $\leq n$ Runden die ersten $(c+3)\lceil \log n \rceil$ Bits gleich sind: $\leq n^3 \cdot \frac{1}{n^{c+3}} = \frac{1}{n^c}$ (Union Bound).

5.3.3 Anmerkungen:

Verbesserung des Algorithmus:

- Algorithmus kann so modifiziert werden, dass n nicht bekannt sein muss
- Schwierigkeit: Vergleich der Zufallszahlen kann für verschiedene Paare unterschiedlich lange dauern, erfordert Synchronisationstechniken

State of the Art: (Maximalgrad Δ)

- Obere Schranke randomisiert: $O(\log \Delta + 2^{O(\sqrt{\log \log n})})$ [Gha16]
- Obere Schranke deterministisch: $2^{O(\sqrt{\log n})}$ [PS96]
- Untere Schranke: $\Omega(\min\{\frac{\log \Delta}{\log \log \Delta}, \frac{\sqrt{\log n}}{\sqrt{\log \log n}}\})$ [KMW16]

Literatur

[Gha16] Mohsen Ghaffari. “An Improved Distributed Algorithm for Maximal Independent Set”. In: *Symposium on Discrete Algorithms (SODA)*. 2016, S. 270–277. DOI: `10.1137/1.9781611974331.ch20` (siehe S. 5-7).

[KMW16] Fabian Kuhn, Thomas Moscibroda und Roger Wattenhofer. “Local Computation: Lower and Upper Bounds”. In: *Journal of the ACM* 63.2 (2016), 17:1–17:44. DOI: `10.1145/2742012` (siehe S. 5-7).

[MRS+11] Yves Métivier, John Michael Robson, Nasser Saheb-Djahromi und Akka Zemmari. “An optimal bit complexity randomized distributed MIS algorithm”. In: *Distributed Computing* 23.5-6 (2011), S. 331–340. DOI: `10.1007/s00446-010-0121-5` (siehe S. 5-2).

[PS96] Alessandro Panconesi und Aravind Srinivasan. “On the Complexity of Distributed Network Decomposition”. In: *Journal of Algorithms* 20.2 (1996), S. 356–374. DOI: `10.1006/jagm.1996.0017` (siehe S. 5-7).