Algorithmen II
Vorlesung am 21.11.2013

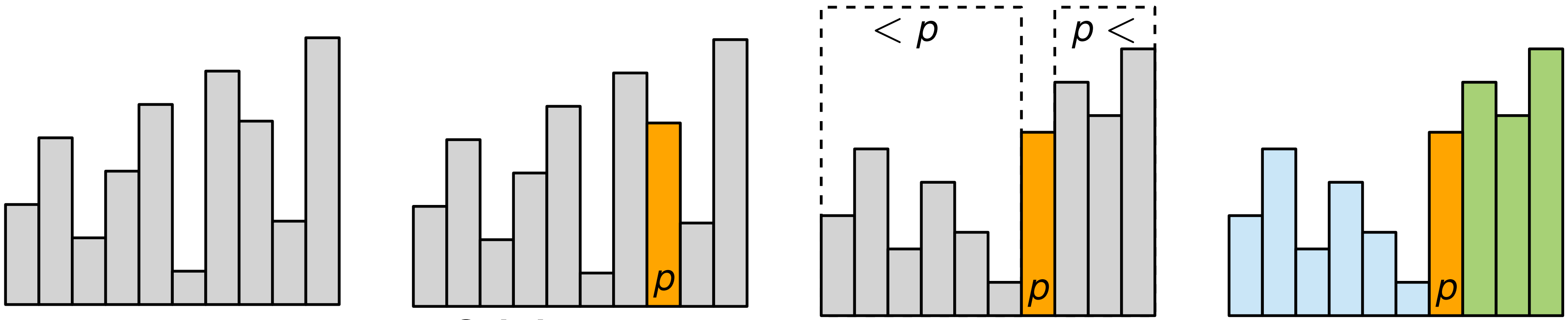
Randomisierte Algorithmen

INSTITUT FÜR THEORETISCHE INFORMATIK · PROF. DR. DOROTHEA WAGNER

Definition

Definition 8.1: Ein Algorithmus, der im Laufe seiner Ausführung zufällige Entscheidungen trifft, heißt *randomisierter Algorithmus*.

Bekanntes Beispiel: Quicksort



Eingabe: Unsortierte Liste

1. Schritt Wähle Element zufällig.

2. Schritt: Umsortieren

3. Schritt: Rekursives Vorgehen auf Teilmengen

Laufzeit:

- Schlimmster Fall: $\Theta(n^2)$.
- Zu erwartende Laufzeit: $\mathcal{O}(n \cdot \log n)$

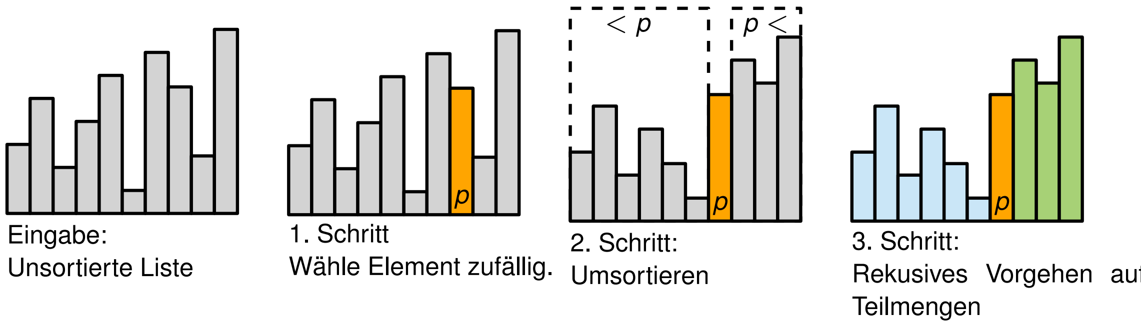
Bemerkung: Algorithmus liefert immer korrektes Ergebnis.

Institut für Theoretische Informatik
Prof. Dr. Dorothea Wagner

Arten von randomisierten Algorithmen

Las Vegas Algorithmus

- Liefert immer korrektes Ergebnis.
- Laufzeit variiert.
- Beispiel: Quicksort



Monte Carlo Algorithmus (Eselsbrücke: mostly correct)

- Kann auch falsches Ergebnis liefern.
- Betrachte Wahrscheinlichkeit für Fehler.
- Für Entscheidungsproblem, d.h. nur *JA*/*NEIN*-Antwort möglich, gibt es zwei Arten:
  - **beidseitig**: Für beide möglichen Antworten gibt es Wahrscheinlichkeit > 0, dass Antwort falsch ist.
  - **einseitig**: Für eine der beiden Antworten ist Wahrscheinlichkeit gleich Null, dass Antwort fehlerbehaftet ist.
    Beispiel: Die Antwort *JA* ist immer richtig, die Antwort *NEIN* kann auch falsch sein.

Wahrscheinlichkeitsklassen

Die Klasse $\mathcal{RP}$ (randomisiert polynomial) enthält alle Entscheidungsprobleme Π, für die es einen polynomialen, randomisierten Algorithmus A gibt, so dass für alle Instanzen I von Π gilt:

$$\begin{cases} I \in Y_\Pi \longrightarrow \Pr[A(I) \text{ ist „JA"}] \geq \frac{1}{2} \\ I \notin Y_\Pi \longrightarrow \Pr[A(I) \text{ ist „JA"}] = 0 \end{cases}$$

Die Klasse $\mathcal{PP}$ (probabilistic polynomial) enthält alle Entscheidungsprobleme Π, für die es einen polynomialen, randomisierten Algorithmus A gibt, so dass für alle Instanzen I gilt:

$$\begin{cases} I \in Y_\Pi \longrightarrow \Pr[A(I) \text{ ist „JA"}] > \frac{1}{2} \\ I \notin Y_\Pi \longrightarrow \Pr[A(I) \text{ ist „JA"}] < \frac{1}{2} \end{cases}$$

Die Klasse $\mathcal{BPP}$ (bounded error $\mathcal{PP}$) ist die Klasse der Entscheidungsprobleme Π, für die es einen polynomialen, randomisierten Algorithmus A gibt, so dass für alle Instanzen I gilt:

$$\begin{cases} I \in Y_\Pi \longrightarrow \Pr[A(I) \text{ ist „JA"}] \geq \frac{3}{4} \\ I \notin Y_\Pi \longrightarrow \Pr[A(I) \text{ ist „JA"}] \leq \frac{1}{4} \end{cases}$$

- Y_Π ist die Menge der sogenannten „JA-Beispiele" von Π.
- Dabei entspricht $\Pr[A(I) \text{ ist „JA"}]$ der Wahrscheinlichkeit, dass die Antwort, die A bei der Eingabe von I gibt, „JA" ist.

Wahrscheinlichkeitsklassen

Karlsruher Institut für Technologie

Die Klasse $\mathcal{RP}$ (randomisiert polynomial) enthält alle Entscheidungsprobleme Π, für die es einen polynomialen, randomisierten Algorithmus A gibt, so dass für alle Instanzen I von Π gilt:

$$\begin{cases} I \in Y_\Pi \longrightarrow \Pr[A(I) \text{ ist „JA"}] \geq \frac{1}{2} \\ I \notin Y_\Pi \longrightarrow \Pr[A(I) \text{ ist „JA"}] = 0 \end{cases}$$

einseitiger Monte Carlo Algorithmus

Die Klasse $\mathcal{PP}$ (probabilistic polynomial) enthält alle Entscheidungsprobleme Π, für die es einen polynomialen, randomisierten Algorithmus A gibt, so dass für alle Instanzen I gilt:

$$\begin{cases} I \in Y_\Pi \longrightarrow \Pr[A(I) \text{ ist „JA"}] > \frac{1}{2} \\ I \notin Y_\Pi \longrightarrow \Pr[A(I) \text{ ist „JA"}] < \frac{1}{2} \end{cases}$$

beidseitiger Monte Carlo Algorithmus

Die Klasse $\mathcal{BPP}$ (bounded error $\mathcal{PP}$) ist die Klasse der Entscheidungsprobleme Π, für die es einen polynomialen, randomisierten Algorithmus A gibt, so dass für alle Instanzen I gilt:

$$\begin{cases} I \in Y_\Pi \longrightarrow \Pr[A(I) \text{ ist „JA"}] \geq \frac{3}{4} \\ I \notin Y_\Pi \longrightarrow \Pr[A(I) \text{ ist „JA"}] \leq \frac{1}{4} \end{cases}$$

beidseitiger Monte Carlo Algorithmus

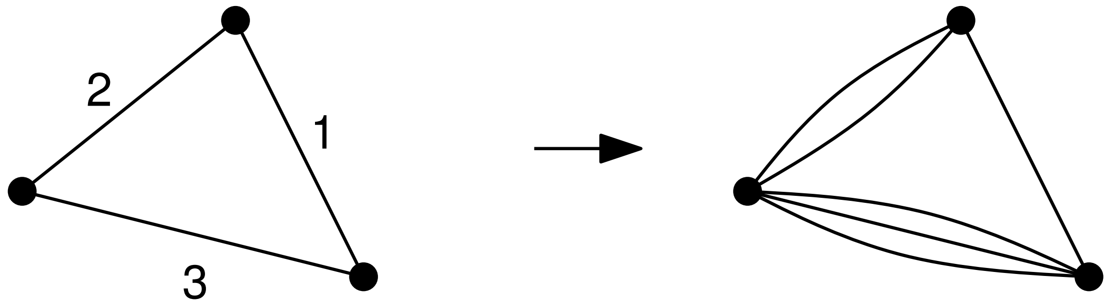
- Y_Π ist die Menge der sogenannten „JA-Beispiele" von Π.
- Dabei entspricht $\Pr[A(I)$ ist „JA"$]$ der Wahrscheinlichkeit, dass die Antwort, die A bei der Eingabe von I gibt, „JA" ist.

Monte Carlo Algorithmus für MinCut

Institut für Theoretische Informatik
Prof. Dr. Dorothea Wagner

Problemdefinition

Fasse $G = (V, E)$ mit Kantengewichtsfunktion $c : E \to \mathbb{N}$ als Multigraph auf, d.h. für $\{u, v\} \in E$ gibt es $c(\{u, v\})$ Kanten:



Problem MinCut: Sei $G = (V, E)$ mit $c : E \to \mathbb{N}$ ein solcher Multigraph. Gesucht ist eine Partition V_1 und V_2 von V, sodass

$$\text{cutsize}(V_1, V_2) := \left|\{\{u, v\} \in E \mid u \in V_1 \text{ und } v \in V_2\}\right|$$

minimal ist.

Algorithmus

RANDOM MINCUT

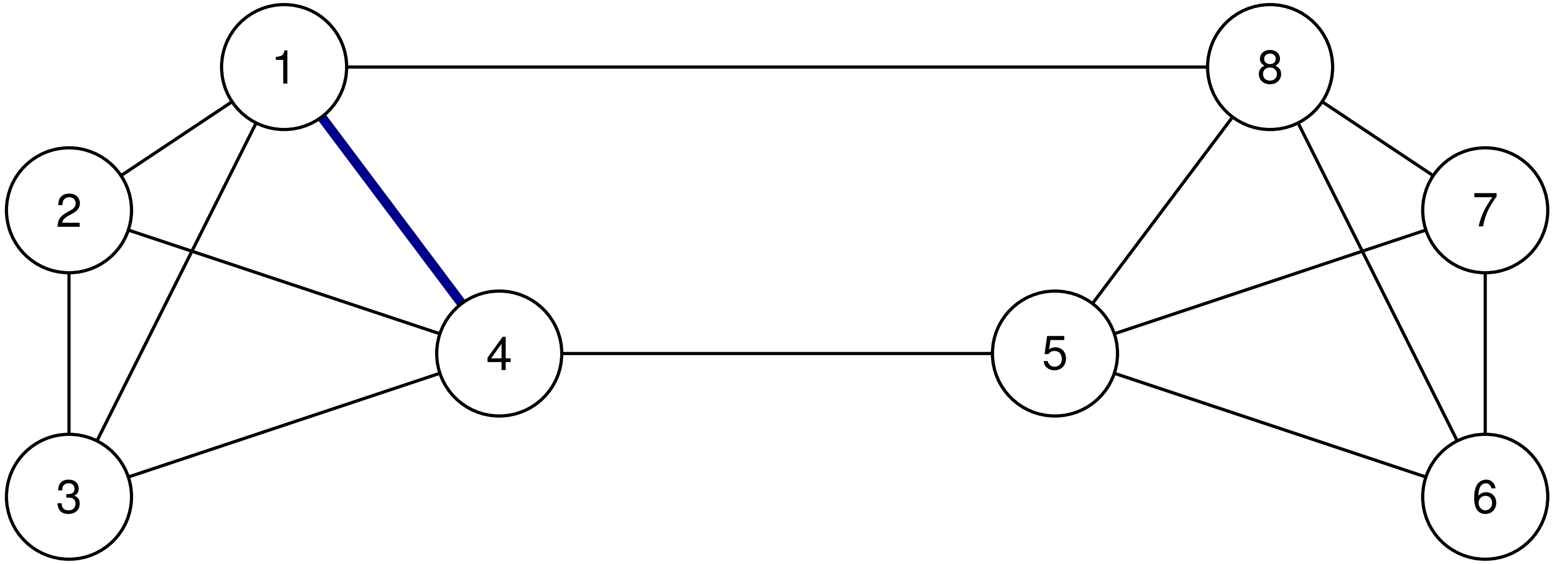
Eingabe: Multigraph $G = (V, E)$

Ausgabe: Schnitt in Form eines Graphen mit zwei Superknoten

solange $|V| > 2$ **tue**

- $e \leftarrow$ zufällige Kante in E
- Bilde neuen Graph $G = (V, E)$, der entsteht, wenn die Endknoten von e verschmolzen werden und alle Kanten zwischen Endknoten von e entfernt werden

Gebe V zurück.



Institut für Theoretische Informatik
Prof. Dr. Dorothea Wagner

Algorithmus

Karlsruher Institut für Technologie

RANDOM MINCUT

Eingabe: Multigraph $G = (V, E)$

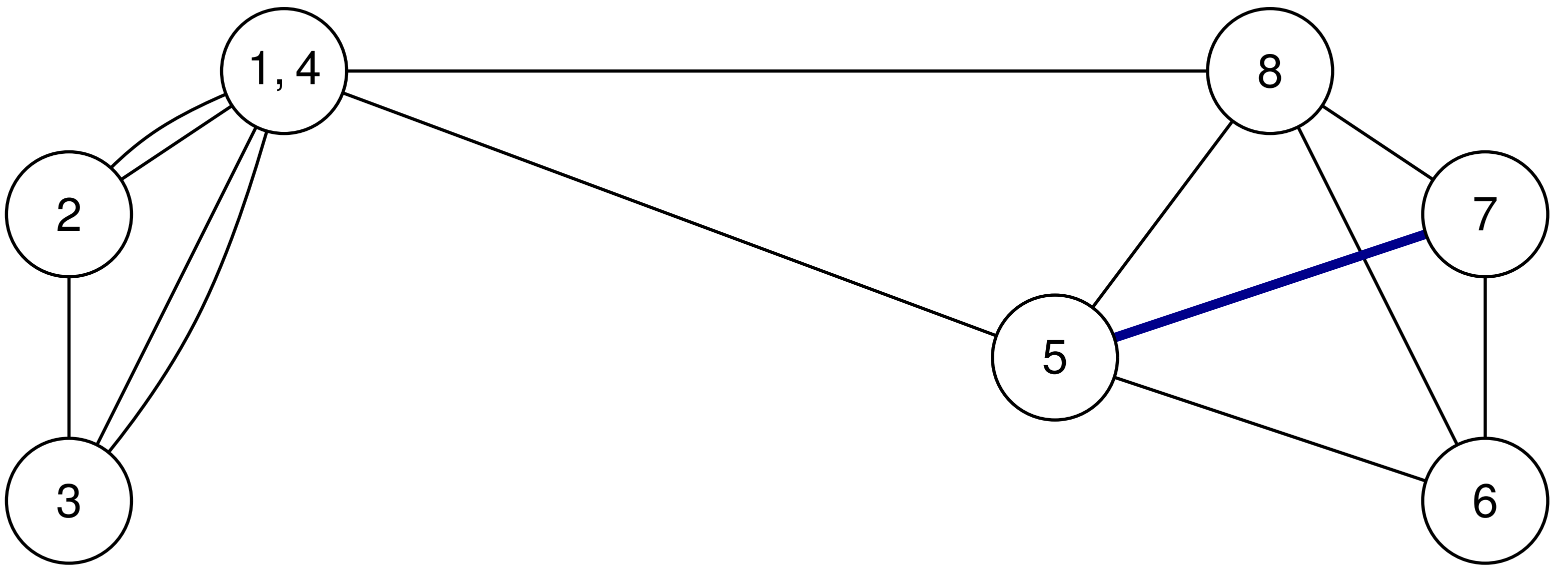
Ausgabe: Schnitt in Form eines Graphen mit zwei Superknoten

solange $|V| > 2$ **tue**

$e \leftarrow$ zufällige Kante in E

Bilde neuen Graph $G = (V, E)$, der entsteht, wenn die Endknoten von e verschmolzen werden und alle Kanten zwischen Endknoten von e entfernt werden

Gebe V zurück.



Algorithmus

RANDOM MINCUT

Eingabe: Multigraph $G = (V, E)$

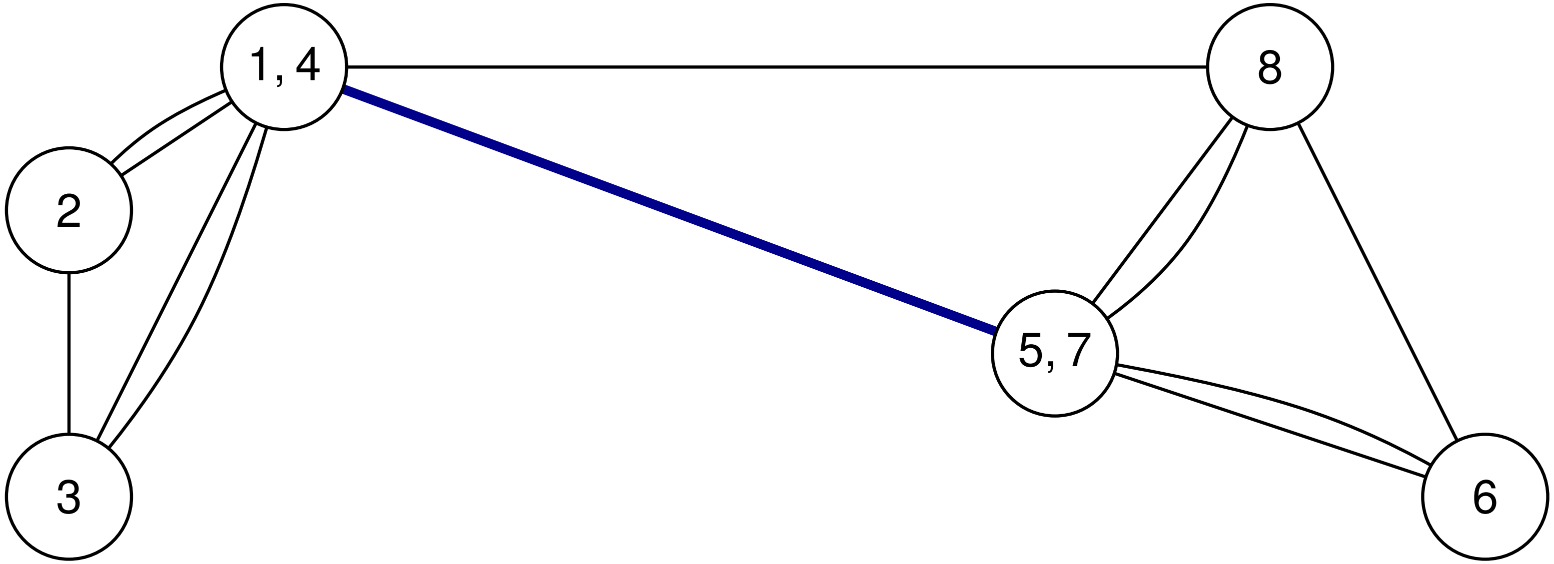
Ausgabe: Schnitt in Form eines Graphen mit zwei Superknoten

solange $|V| > 2$ **tue**

$e \leftarrow$ zufällige Kante in E

Bilde neuen Graph $G = (V, E)$, der entsteht, wenn die Endknoten von e verschmolzen werden und alle Kanten zwischen Endknoten von e entfernt werden

Gebe V zurück.



Algorithmus

RANDOM MINCUT

Eingabe: Multigraph $G = (V, E)$

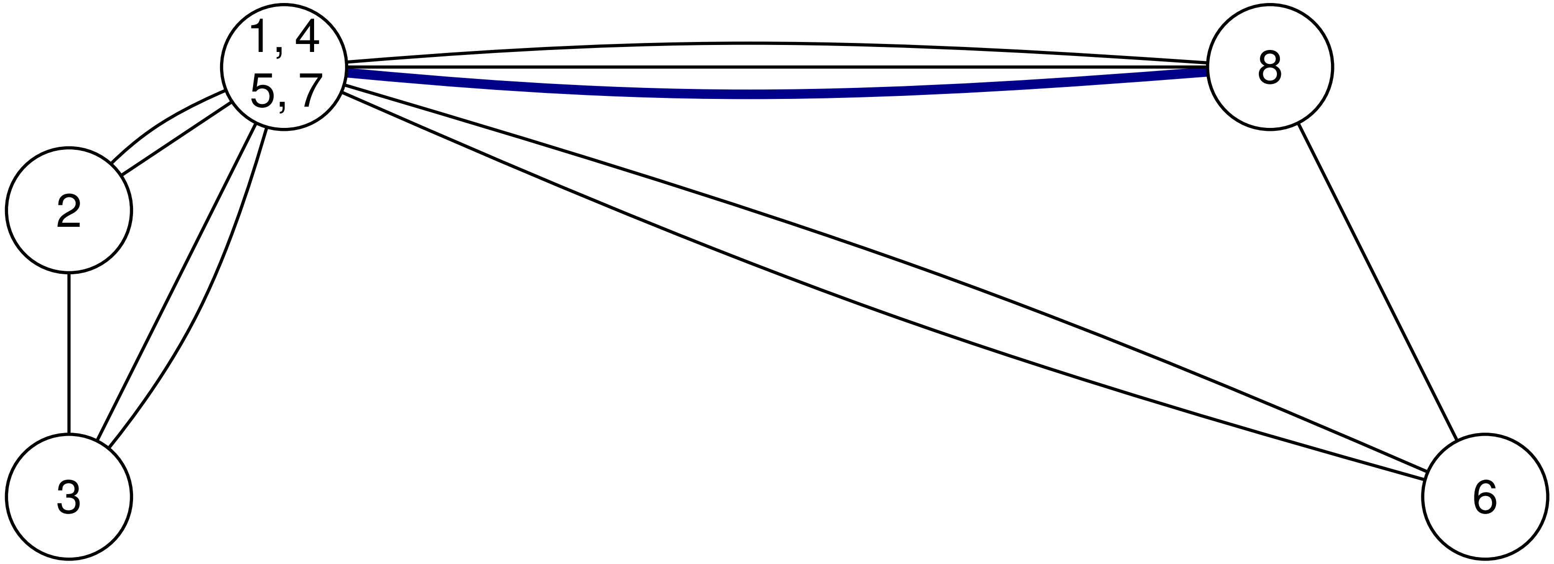
Ausgabe: Schnitt in Form eines Graphen mit zwei Superknoten

solange $|V| > 2$ **tue**

  $e \leftarrow$ zufällige Kante in E

  Bilde neuen Graph $G = (V, E)$, der entsteht, wenn die Endknoten von e verschmolzen werden und alle Kanten zwischen Endknoten von e entfernt werden

Gebe V zurück.



Algorithmus

Karlsruher Institut für Technologie

Random MinCut

Eingabe: Multigraph $G = (V, E)$

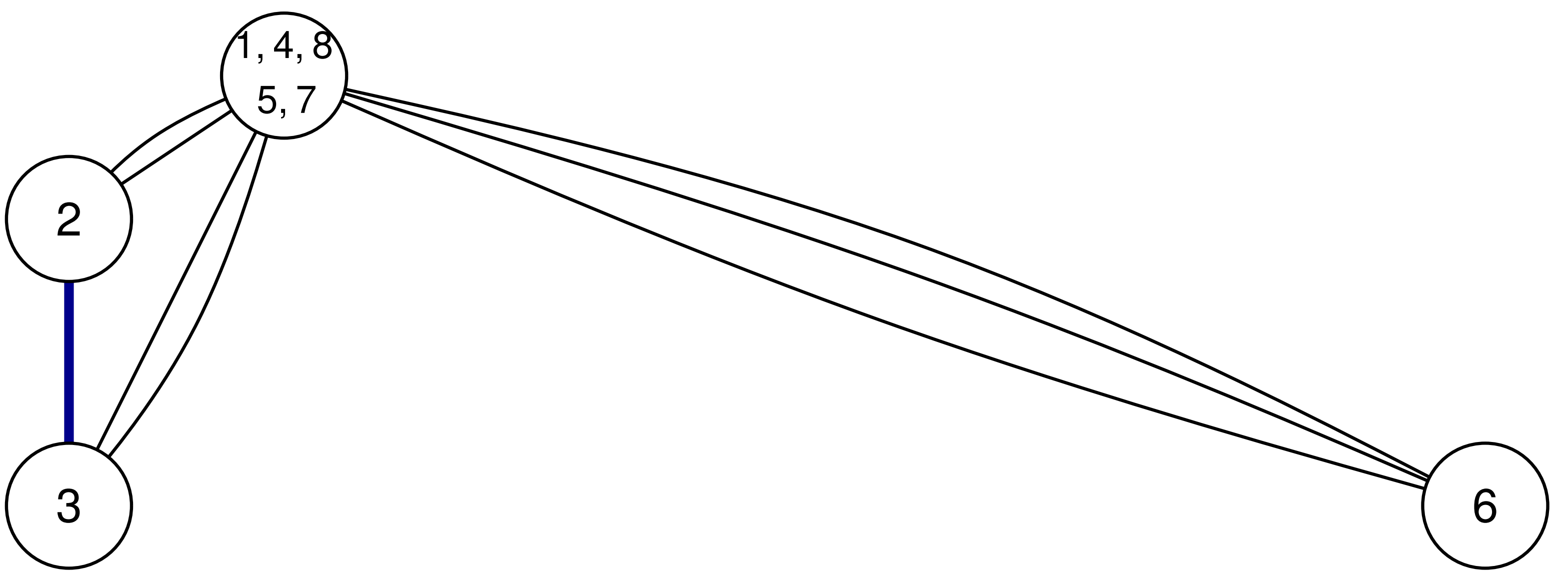
Ausgabe: Schnitt in Form eines Graphen mit zwei Superknoten

solange $|V| > 2$ **tue**

  $e \leftarrow$ zufällige Kante in E

  Bilde neuen Graph $G = (V, E)$, der entsteht, wenn die Endknoten von e verschmolzen werden und alle Kanten zwischen Endknoten von e entfernt werden

Gebe V zurück.



Algorithmus

RANDOM MINCUT

Eingabe: Multigraph $G = (V, E)$

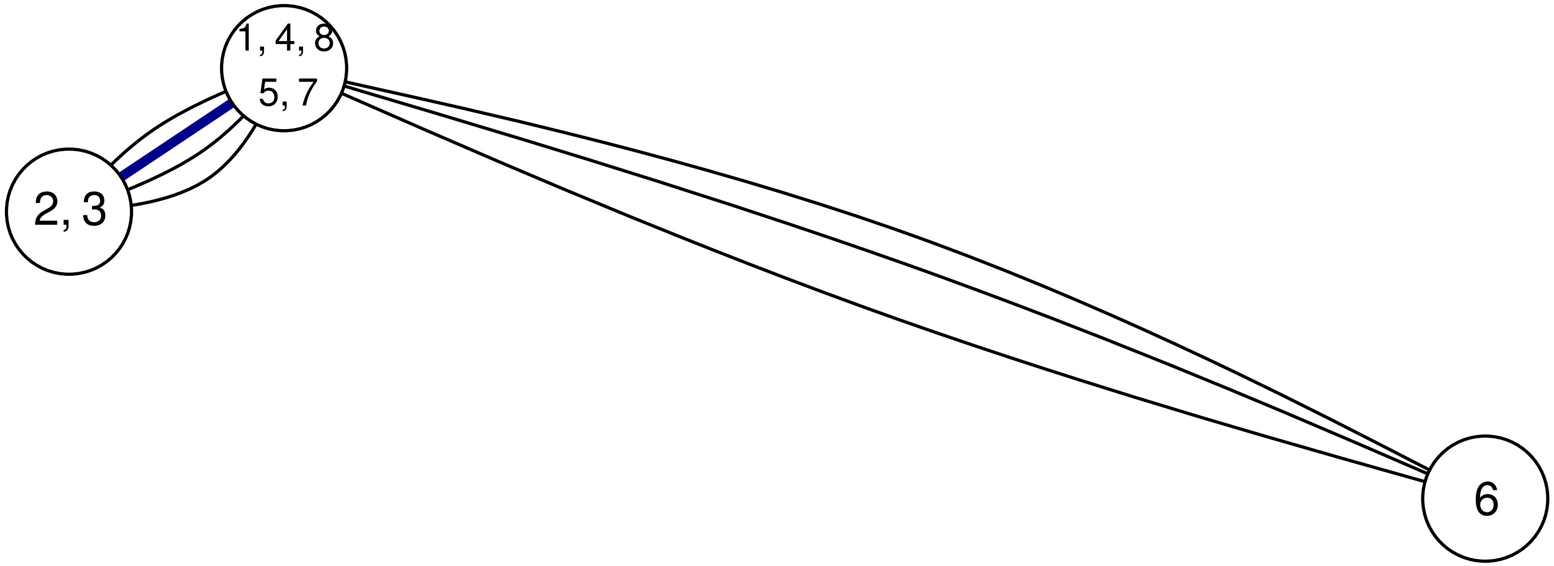
Ausgabe: Schnitt in Form eines Graphen mit zwei Superknoten

solange $|V| > 2$ **tue**

$e \leftarrow$ zufällige Kante in E

Bilde neuen Graph $G = (V, E)$, der entsteht, wenn die Endknoten von e verschmolzen werden und alle Kanten zwischen Endknoten von e entfernt werden

Gebe V zurück.



Algorithmus

RANDOM MINCUT

Eingabe: Multigraph $G = (V, E)$

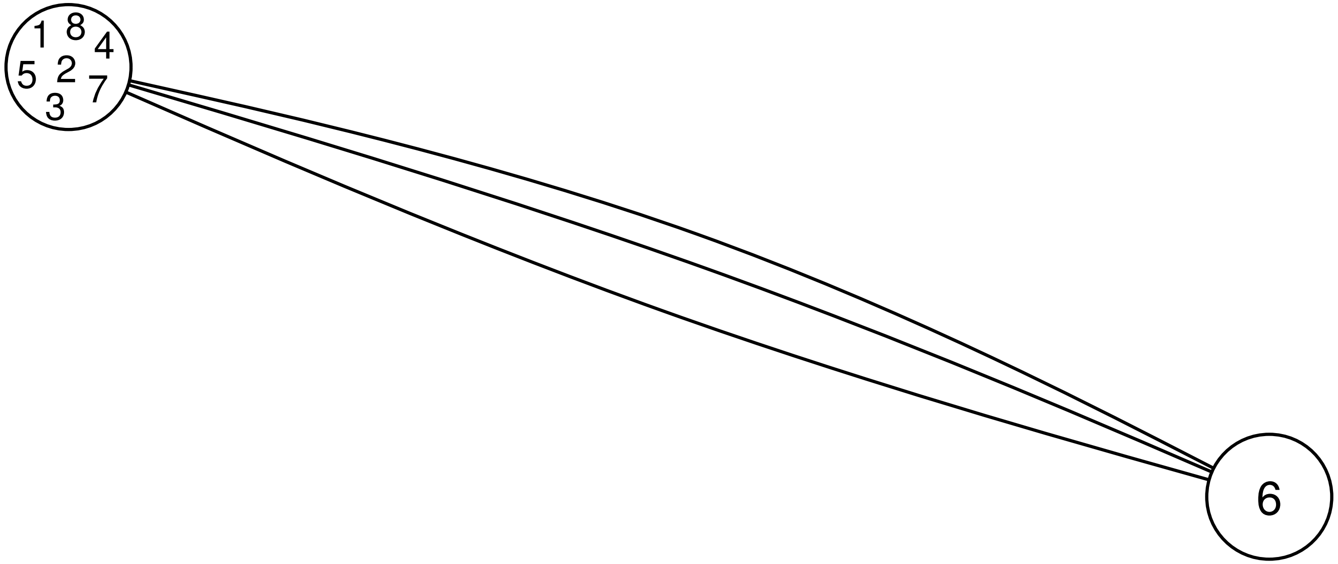
Ausgabe: Schnitt in Form eines Graphen mit zwei Superknoten

solange $|V| > 2$ **tue**

$e \leftarrow$ zufällige Kante in E

Bilde neuen Graph $G = (V, E)$, der entsteht, wenn die Endknoten von e verschmolzen werden und alle Kanten zwischen Endknoten von e entfernt werden

Gebe V zurück.



Algorithmus

RANDOM MINCUT

Eingabe: Multigraph $G = (V, E)$

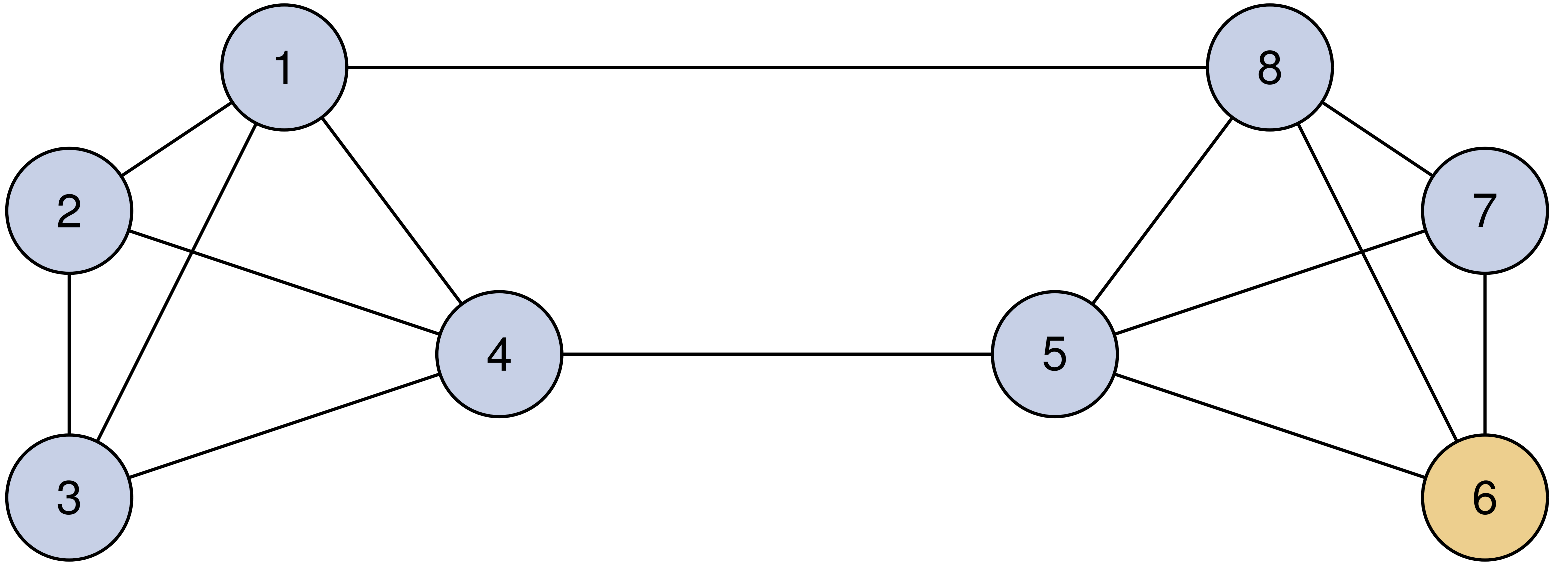
Ausgabe: Schnitt in Form eines Graphen mit zwei Superknoten

solange $|V| > 2$ **tue**

$e \leftarrow$ zufällige Kante in E

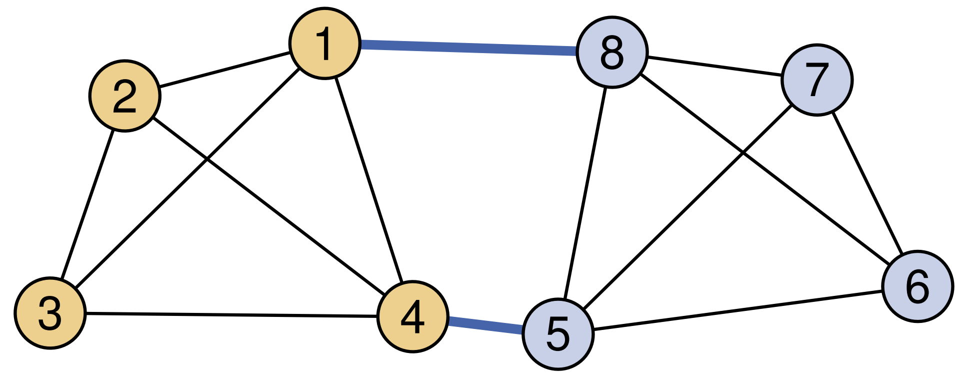
Bilde neuen Graph $G = (V, E)$, der entsteht, wenn die Endknoten von e verschmolzen werden und alle Kanten zwischen Endknoten von e entfernt werden

Gebe V zurück.



Qualität von RANDOM MINCUT

Satz 8.7. Wenn jede Kante mit gleicher Wahrscheinlichkeit gewählt wird, dann ist die Wahrscheinlichkeit das RANDOMMINCUT einen bestimmten minimalen Schnitt $(V_1, V_2 = V \setminus V_1)$ findet größer $\frac{2}{n^2}$, wobei $|V| = n$.



Qualität von RANDOM MINCUT

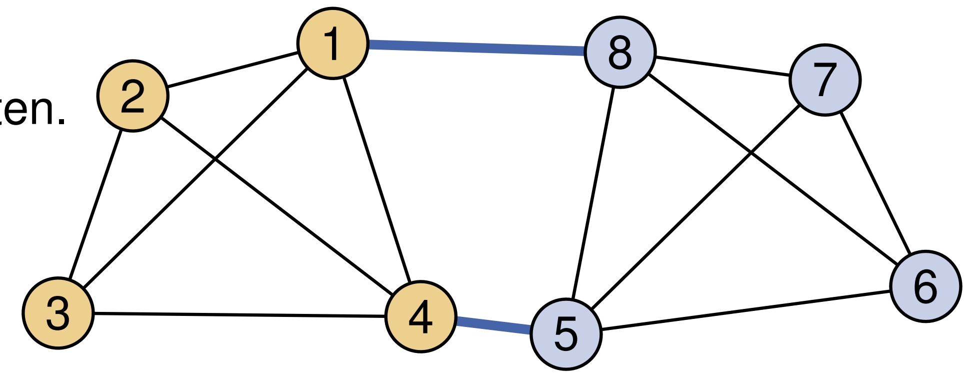
Satz 8.7. Wenn jede Kante mit gleicher Wahrscheinlichkeit gewählt wird, dann ist die Wahrscheinlichkeit das RANDOMMINCUT einen bestimmten minimalen Schnitt $(V_1, V_2 = V \setminus V_1)$ findet größer $\frac{2}{n^2}$, wobei $|V| = n$.

Beweis:

Sei (V_1, V_2) beliebiger minimaler Schnitt mit k Kanten.

⟶ Jeder Knoten bestizt mind. Grad k.

⟶ G hat mind. $\frac{k \cdot n}{2}$ Kanten.

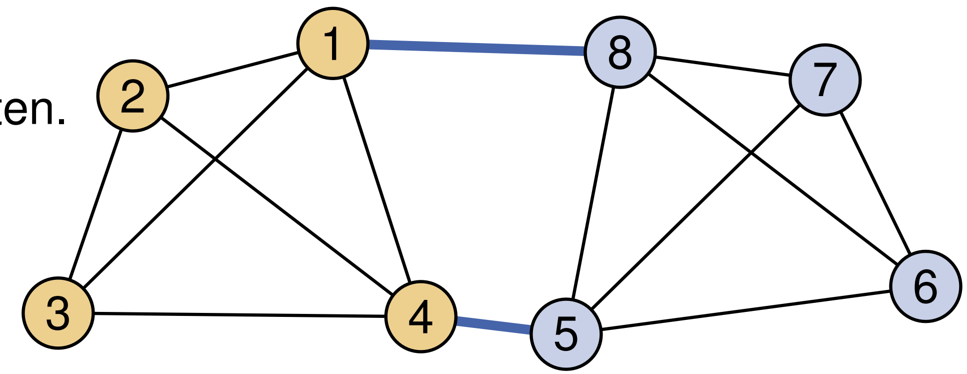


Institut für Theoretische Informatik
Prof. Dr. Dorothea Wagner

Satz 8.7. Wenn jede Kante mit gleicher Wahrscheinlichkeit gewählt wird, dann ist die Wahrscheinlichkeit das RANDOMMINCUT einen bestimmten minimalen Schnitt $(V_1, V_2 = V \setminus V_1)$ findet größer $\frac{2}{n^2}$, wobei $|V| = n$.

Beweis:

Sei (V_1, V_2) beliebiger minimaler Schnitt mit k Kanten.



→ Jeder Knoten bestizt mind. Grad k.

→ G hat mind. $\frac{k \cdot n}{2}$ Kanten.

Idee: Schätze Wahrscheinlichkeit ab, dass keine Kante aus (V_1, V_2) gewählt wird.

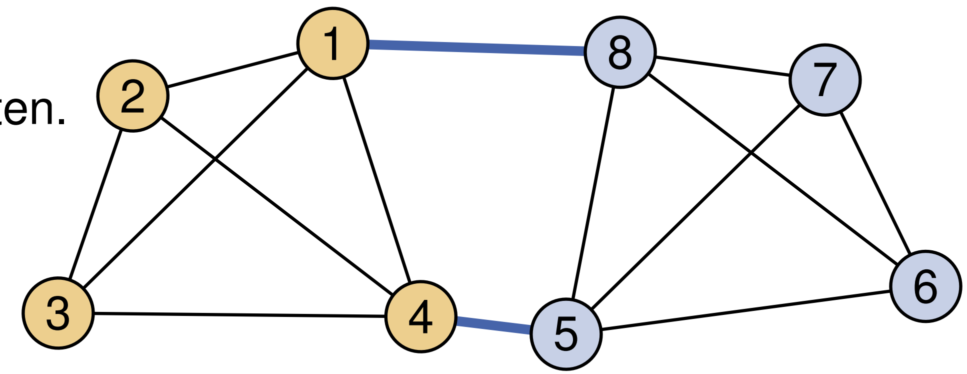
Ereignis: A_i = im i-ten Schritt wird keine Kante aus (V_1, V_2) gewählt.

Qualität von RANDOM MINCUT

Satz 8.7. Wenn jede Kante mit gleicher Wahrscheinlichkeit gewählt wird, dann ist die Wahrscheinlichkeit das RANDOMMINCUT einen bestimmten minimalen Schnitt $(V_1, V_2 = V \setminus V_1)$ findet größer $\frac{2}{n^2}$, wobei $|V| = n$.

Beweis:

Sei (V_1, V_2) beliebiger minimaler Schnitt mit k Kanten.



$\longrightarrow$ Jeder Knoten bestizt mind. Grad k.

$\longrightarrow$ G hat mind. $\frac{k \cdot n}{2}$ Kanten.

Idee: Schätze Wahrscheinlichkeit ab, dass keine Kante aus (V_1, V_2) gewählt wird.

Ereignis: A_i = im i-ten Schritt wird keine Kante aus (V_1, V_2) gewählt.

$\Pr[A_1] \quad \geq 1 - \frac{2}{n} = \frac{n-2}{n}$

Denn:

Wahrscheinlichkeit, dass im ersten Schritt Kante aus (V_1, V_2) gewählt wird: $\leq \frac{k}{\frac{k \cdot n}{2}}$

Qualität von RANDOM MINCUT

Karlsruher Institut für Technologie

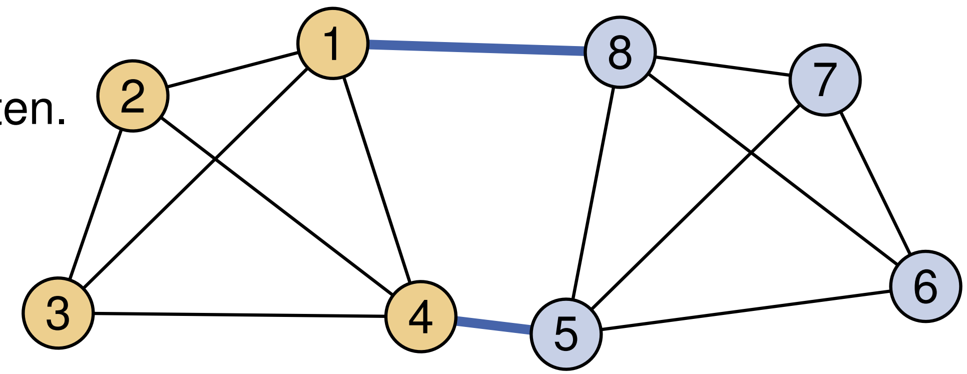
Satz 8.7. Wenn jede Kante mit gleicher Wahrscheinlichkeit gewählt wird, dann ist die Wahrscheinlichkeit das RANDOMMINCUT einen bestimmten minimalen Schnitt $(V_1, V_2 = V \setminus V_1)$ findet größer $\frac{2}{n^2}$, wobei $|V| = n$.

Beweis:

Sei (V_1, V_2) beliebiger minimaler Schnitt mit k Kanten.

→ Jeder Knoten bestizt mind. Grad k.

→ G hat mind. $\frac{k \cdot n}{2}$ Kanten.



Idee: Schätze Wahrscheinlichkeit ab, dass keine Kante aus (V_1, V_2) gewählt wird.

Ereignis: A_i = im i-ten Schritt wird keine Kante aus (V_1, V_2) gewählt.

$\Pr[A_1]$	$\geq 1 - \frac{2}{n} = \frac{n-2}{n}$
$\Pr[A_2 \mid A_1]$	$\geq 1 - \frac{2}{n-1} = \frac{n-3}{n-1}$

Es verbleiben mindestens $\frac{k \cdot (n-1)}{2}$ Kanten.

Pr[Im zweiten Schritt wird Kante aus (V_1, V_2) gewählt, nachdem A_1 eingetreten ist] $\leq \frac{k}{\frac{k \cdot (n-1)}{2}}$

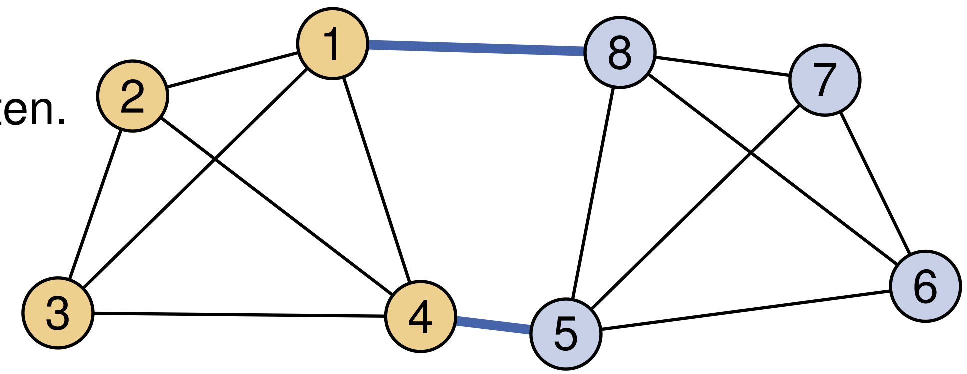
Institut für Theoretische Informatik
Prof. Dr. Dorothea Wagner

Qualität von RANDOM MINCUT

Satz 8.7. Wenn jede Kante mit gleicher Wahrscheinlichkeit gewählt wird, dann ist die Wahrscheinlichkeit das RANDOMMINCUT einen bestimmten minimalen Schnitt $(V_1, V_2 = V \setminus V_1)$ findet größer $\frac{2}{n^2}$, wobei $|V| = n$.

Beweis:

Sei (V_1, V_2) beliebiger minimaler Schnitt mit k Kanten.



→ Jeder Knoten bestizt mind. Grad k.

→ G hat mind. $\frac{k \cdot n}{2}$ Kanten.

Idee: Schätze Wahrscheinlichkeit ab, dass keine Kante aus (V_1, V_2) gewählt wird.

Ereignis: A_i = im i-ten Schritt wird keine Kante aus (V_1, V_2) gewählt.

$\Pr[A_1]$	$\geq 1 - \frac{2}{n} = \frac{n-2}{n}$
$\Pr[A_2 \mid A_1]$	$\geq 1 - \frac{2}{n-1} = \frac{n-3}{n-1}$
$\Pr[A_i \mid \bigcup_{j=1}^{i-1} A_j]$	$\geq 1 - \frac{2}{n-i+1} = \frac{n-i-1}{n-i+1}$

Es verbleiben mindestens $\frac{k \cdot (n-i+1)}{2}$ Kanten.

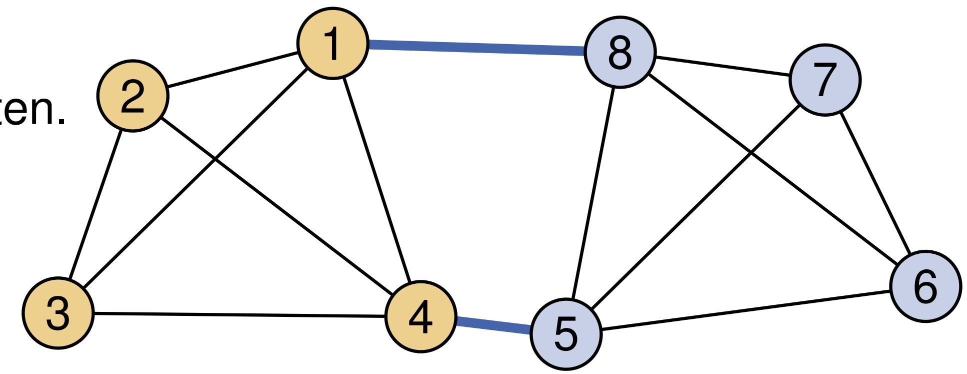
$\Pr[\, A_1, \ldots, A_{i-1}$ sind eingetreten und i-ter Schritt wählt Kante aus $(V_1, V_2)] \leq \frac{k}{\frac{k \cdot (n-i+1)}{2}}$

Qualität von RANDOM MINCUT

Satz 8.7. Wenn jede Kante mit gleicher Wahrscheinlichkeit gewählt wird, dann ist die Wahrscheinlichkeit das RANDOMMINCUT einen bestimmten minimalen Schnitt $(V_1, V_2 = V \setminus V_1)$ findet größer $\frac{2}{n^2}$, wobei $|V| = n$.

Beweis:

Sei (V_1, V_2) beliebiger minimaler Schnitt mit k Kanten.



→ Jeder Knoten bestizt mind. Grad k.

→ G hat mind. $\frac{k \cdot n}{2}$ Kanten.

Idee: Schätze Wahrscheinlichkeit ab, dass keine Kante aus (V_1, V_2) gewählt wird.

Ereignis: A_i = im i-ten Schritt wird keine Kante aus (V_1, V_2) gewählt.

$\Pr[A_1]$	$\geq 1 - \frac{2}{n} = \frac{n-2}{n}$
$\Pr[A_2 \mid A_1]$	$\geq 1 - \frac{2}{n-1} = \frac{n-3}{n-1}$
$\Pr[A_i \mid \bigcup_{j=1}^{i-1} A_j]$	$\geq 1 - \frac{2}{n-i+1} = \frac{n-i-1}{n-i+1}$

$$\Pr\left[\bigcap_{i=1}^{n-2} A_i\right] \geq \prod_{i=1}^{n-2} \left(\frac{n-i-1}{n-i+1}\right) = \frac{(n-2)\cdot(n-3)\cdot(n-4)\cdot(n-5)\cdot\ldots\cdot 2\cdot 1}{(n-0)\cdot(n-1)\cdot(n-2)\cdot(n-3)\cdot\ldots\cdot 4\cdot 3} = \frac{2}{n\cdot(n-1)}$$

Qualität von Random MinCut

Satz 8.7. Wenn jede Kante mit gleicher Wahrscheinlichkeit gewählt wird, dann ist die Wahrscheinlichkeit das RandomMinCut einen bestimmten minimalen Schnitt ($V_1, V_2 = V \setminus V_1$) findet größer $\frac{2}{n^2}$, wobei $|V| = n$.

Folgerung 8.8.
Wendet man Random MinCut nur $n - \ell$ Schritte lang an, d.h. man stoppt, wenn ℓ Knoten übrig sind, so ist die Wahrscheinlichkeit, dass bis dahin keine Kante eines bestimmten minimalen Schnitts (V_1, V_2) gewählt wurde, mindestens

$$\frac{\binom{\ell}{2}}{\binom{n}{2}}, \qquad \text{d.h. in } \Omega\left(\left(\frac{\ell}{n}\right)^2\right) .$$

$$\Pr\left[\bigcap_{i=1}^{n-\ell} A_i\right] \geq \prod_{i=1}^{n-\ell}\left(\frac{n-i-1}{n-i+1}\right) = \frac{(n-2)\cdot(n-3)\cdot(n-4)\cdot\ldots\cdot\ell\cdot(\ell-1)}{(n-0)\cdot(n-1)\cdot(n-2)\cdot\ldots\cdot(l+2)\cdot(l+1)} = \frac{\binom{\ell}{2}}{\binom{n}{2}}$$

Zusammenfassung

- Wenn Wahl einer zufälligen Kante in $\mathcal{O}(n)$ realisierbar, dann hat der Algorithmus eine Laufzeit von $\mathcal{O}(n^2)$.

$\Rightarrow$ Bessere Laufzeit als deterministische Variante ($\mathcal{O}(n^2 \log n + n \cdot m)$), siehe Skript.

- Wendet man RANDOM MINCUT $\frac{n^2}{2}$ mal unabhängig voneinander an, so ergibt sich:

$$Pr[\text{Bestimmter Schnitt nicht gefunden}] = \left(1 - \frac{2}{n^2}\right)^{\frac{n^2}{2}} < \frac{1}{e}$$

Eulersche Zahl

- Allerdings $\mathcal{O}(n^4)$-Algorithmus

$\Rightarrow$ Schlechter als deterministische Variante.

Institut für Theoretische Informatik
Prof. Dr. Dorothea Wagner

Ein effizienterer randomisierter MinCut-Algorithmus

Institut für Theoretische Informatik
Prof. Dr. Dorothea Wagner

Fast Random MinCut

Eingabe: Graph $G = (V, E)$ als Multigraph, $|V| = n$

Ausgabe: Schnitt

wenn $n \leq 6$ **dann**

  berechne direkt deterministisch einen MINCUT

sonst

  $\ell \leftarrow \left\lceil \frac{n}{\sqrt{2}} \right\rceil$

  $G_1 \leftarrow$ RANDOM MINCUT(bis ℓ Knoten übrig)

  $G_2 \leftarrow$ RANDOM MINCUT(bis ℓ Knoten übrig)

  $C_1 \leftarrow$ FAST RANDOM MINCUT(G_1) (rekursiv)

  $C_2 \leftarrow$ FAST RANDOM MINCUT(G_2) (rekursiv)

  Gib den kleineren der beiden Schnitte C_1 und C_2 aus.

Fast Random MinCut

Eingabe: Graph $G = (V, E)$ als Multigraph, $|V| = n$

Ausgabe: Schnitt

wenn $n \leq 6$ **dann**

  berechne direkt deterministisch einen MinCut

sonst

  $\ell \leftarrow \left\lceil \frac{n}{\sqrt{2}} \right\rceil$

  $G_1 \leftarrow$ Random MinCut(bis ℓ Knoten übrig)
  $G_2 \leftarrow$ Random MinCut(bis ℓ Knoten übrig) A

  $C_1 \leftarrow$ Fast Random MinCut(G_1) (rekursiv)
  $C_2 \leftarrow$ Fast Random MinCut(G_2) (rekursiv) B

  Gib den kleineren der beiden Schnitte C_1 und C_2 aus.

Satz 8.9: Fast Random MinCut hat eine Laufzeit $\mathcal{O}(n^2 \log n)$.

Beweis: Laufzeit $T(n)$ ergibt sich aus folgender Rekursionsabschätzung:

$$T(n) = \underbrace{2 \cdot T\left(\left\lceil \frac{n}{\sqrt{2}} \right\rceil\right)}_{\text{B}} + \underbrace{c \cdot n^2}_{\text{A}}$$

wobei c eine Konstante ist.

Kann mithilfe des Master-Theorems gelöst werden.

Fast Random MinCut (FRMC)

Satz 8.10: Wahrscheinlichkeit, dass FRMC einen minimalen Schnitt findet, ist in $\Omega\left(\frac{1}{\log n}\right)$.

wenn $n \leq 6$ **dann**
  berechne direkt deterministisch einen MinCut
sonst
  $\ell \leftarrow \left\lceil \frac{n}{\sqrt{2}} \right\rceil$
  $G_1 \leftarrow$ Random MinCut(bis ℓ Knoten übrig)
  $G_2 \leftarrow$ Random MinCut(bis ℓ Knoten übrig)
  $C_1 \leftarrow$ Fast Random MinCut(G_1) (rekursiv)
  $C_2 \leftarrow$ Fast Random MinCut(G_2) (rekursiv)
  Gib den kleineren der beiden Schnitte C_1 und C_2 aus.

Fast Random MinCut (FRMC)

Karlsruher Institut für Technologie

Satz 8.10: Wahrscheinlichkeit, dass FRMC einen minimalen Schnitt findet, ist in $\Omega\left(\frac{1}{\log n}\right)$.

Beweisskizze: Sei k Größe eines minimalen Schnitts in G.

Annahme: Es gibt Graph G', der ℓ Knoten besitzt, aus G durch Verschmelzung von Knoten hervorgegangen ist und einen Schnitt mit k Kanten besitzt.

wenn $n \leq 6$ **dann**
  berechne direkt deterministisch einen MinCut
sonst
  $\ell \leftarrow \left\lceil \frac{n}{\sqrt{2}} \right\rceil$
  $G_1 \leftarrow$ Random MinCut(bis ℓ Knoten übrig)
  $G_2 \leftarrow$ Random MinCut(bis ℓ Knoten übrig)
  $C_1 \leftarrow$ Fast Random MinCut(G_1) (rekursiv)
  $C_2 \leftarrow$ Fast Random MinCut(G_2) (rekursiv)
  Gib den kleineren der beiden Schnitte C_1 und C_2 aus.

Fast Random MinCut (FRMC)

Satz 8.10: Wahrscheinlichkeit, dass FRMC einen minimalen Schnitt findet, ist in $\Omega\left(\frac{1}{\log n}\right)$.

Beweisskizze: Sei k Größe eines minimalen Schnitts in G.

Annahme: Es gibt Graph G', der ℓ Knoten besitzt, aus G durch Verschmelzung von Knoten hervorgegangen ist und einen Schnitt mit k Kanten besitzt.

FRMC liefert minimalen Schnitt für G' $\Leftrightarrow$ Rekursion liefert Schnitt der Größe k für G_1 oder G_2

↳ Ebenfalls minimaler Schnitt für G.

wenn $n \leq 6$ **dann**
  berechne direkt deterministisch einen MinCut
sonst
  $\ell \leftarrow \left\lceil \frac{n}{\sqrt{2}} \right\rceil$
  $G_1 \leftarrow$ Random MinCut(bis ℓ Knoten übrig)
  $G_2 \leftarrow$ Random MinCut(bis ℓ Knoten übrig)
  $C_1 \leftarrow$ Fast Random MinCut(G_1) (rekursiv)
  $C_2 \leftarrow$ Fast Random MinCut(G_2) (rekursiv)
  Gib den kleineren der beiden Schnitte C_1 und C_2 aus.

Fast Random MinCut (FRMC)

Satz 8.10: Wahrscheinlichkeit, dass FRMC einen minimalen Schnitt findet, ist in $\Omega\left(\frac{1}{\log n}\right)$.

Beweisskizze: Sei k Größe eines minimalen Schnitts in G.

Annahme: Es gibt Graph G', der ℓ Knoten besitzt, aus G durch Verschmelzung von Knoten hervorgegangen ist und einen Schnitt mit k Kanten besitzt.

Nach Folgerung 8.8:

Pr[Berechnung von G' wählt 0 Kanten eines bestimmten Schnitts] $\geq \left\lceil \frac{\ell}{\sqrt{2}} \right\rceil \cdot \frac{\left\lceil \frac{\ell}{\sqrt{2}} \right\rceil - 1}{\ell \cdot (\ell - 1)} \geq \frac{1}{2}$

wenn $n \leq 6$ **dann**
  berechne direkt deterministisch einen MinCut
sonst
  $\ell \leftarrow \left\lceil \frac{n}{\sqrt{2}} \right\rceil$
  $G_1 \leftarrow$ Random MinCut(bis ℓ Knoten übrig)
  $G_2 \leftarrow$ Random MinCut(bis ℓ Knoten übrig)
  $C_1 \leftarrow$ Fast Random MinCut(G_1) (rekursiv)
  $C_2 \leftarrow$ Fast Random MinCut(G_2) (rekursiv)
  Gib den kleineren der beiden Schnitte C_1 und C_2 aus.

Fast Random MinCut (FRMC)

Satz 8.10: Wahrscheinlichkeit, dass FRMC einen minimalen Schnitt findet, ist in $\Omega\left(\frac{1}{\log n}\right)$.

Beweisskizze: Sei k Größe eines minimalen Schnitts in G.

Annahme: Es gibt Graph G', der ℓ Knoten besitzt, aus G durch Verschmelzung von Knoten hervorgegangen ist und einen Schnitt mit k Kanten besitzt.

Nach Folgerung 8.8:

Pr[Berechnung von G' wählt 0 Kanten eines bestimmten Schnitts] $\geq \left\lceil \frac{\ell}{\sqrt{2}} \right\rceil \cdot \frac{\left\lceil \frac{\ell}{\sqrt{2}} \right\rceil - 1}{\ell \cdot (\ell - 1)} \geq \frac{1}{2}$

$P(\ell)$:= Pr[FRMC findet minimalen Schnitt in Graphen mit ℓ Knoten]

$$P(\ell) \geq 1 - \left(1 - \frac{1}{2} \cdot P\left(\left\lceil \frac{\ell}{\sqrt{2}} \right\rceil\right)\right) \cdot \left(1 - \frac{1}{2} \cdot P\left(\left\lceil \frac{\ell}{\sqrt{2}} \right\rceil\right)\right)$$

für $\ell \geq 7$

(für $\ell \leq 6$ gilt $P(\ell)$) = 1)

wenn $n \leq 6$ **dann**
- berechne direkt deterministisch einen MinCut

sonst
- $\ell \leftarrow \left\lceil \frac{n}{\sqrt{2}} \right\rceil$
- $G_1 \leftarrow$ Random MinCut(bis ℓ Knoten übrig)
- $G_2 \leftarrow$ Random MinCut(bis ℓ Knoten übrig)
- $C_1 \leftarrow$ Fast Random MinCut(G_1) (rekursiv)
- $C_2 \leftarrow$ Fast Random MinCut(G_2) (rekursiv)
- Gib den kleineren der beiden Schnitte C_1 und C_2 aus.

Fast Random MinCut (FRMC)

Satz 8.10: Wahrscheinlichkeit, dass FRMC einen minimalen Schnitt findet, ist in $\Omega\left(\frac{1}{\log n}\right)$.

Beweisskizze: Sei k Größe eines minimalen Schnitts in G.

Annahme: Es gibt Graph G', der ℓ Knoten besitzt, aus G durch Verschmelzung von Knoten hervorgegangen ist und einen Schnitt mit k Kanten besitzt.

Nach Folgerung 8.8:

Pr[Berechnung von G' wählt 0 Kanten eines bestimmten Schnitts] $\geq \left\lceil \frac{\ell}{\sqrt{2}} \right\rceil \cdot \frac{\left\lceil \frac{\ell}{\sqrt{2}} \right\rceil - 1}{\ell \cdot (\ell - 1)} \geq \frac{1}{2}$

$P(\ell)$:= Pr[FRMC findet minimalen Schnitt in Graphen mit ℓ Knoten]

$$P(\ell) \geq 1 - \left(1 - \frac{1}{2} \cdot P\left(\left\lceil \frac{\ell}{\sqrt{2}} \right\rceil\right)\right) \cdot \left(1 - \frac{1}{2} \cdot P\left(\left\lceil \frac{\ell}{\sqrt{2}} \right\rceil\right)\right) \text{ für } \ell \geq 7$$

(für $\ell \leq 6$ gilt $P(\ell)) = 1$)

Pr[G' enthält alle Kanten von min. Schnitt in G]

Pr[FRMC findet min. Schnitt C_1 in G_1]

wenn $n \leq 6$ **dann**
  berechne direkt deterministisch einen MinCut
sonst
  $\ell \leftarrow \left\lceil \frac{n}{\sqrt{2}} \right\rceil$
  $G_1 \leftarrow$ Random MinCut(bis ℓ Knoten übrig)
  $G_2 \leftarrow$ Random MinCut(bis ℓ Knoten übrig)
  $C_1 \leftarrow$ Fast Random MinCut(G_1) (rekursiv)
  $C_2 \leftarrow$ Fast Random MinCut(G_2) (rekursiv)
  Gib den kleineren der beiden Schnitte C_1 und C_2 aus.

Fast Random MinCut (FRMC)

Satz 8.10: Wahrscheinlichkeit, dass FRMC einen minimalen Schnitt findet, ist in $\Omega\left(\frac{1}{\log n}\right)$.

Beweisskizze: Sei k Größe eines minimalen Schnitts in G.

Annahme: Es gibt Graph G', der ℓ Knoten besitzt, aus G durch Verschmelzung von Knoten hervorgegangen ist und einen Schnitt mit k Kanten besitzt.

Nach Folgerung 8.8:

Pr[Berechnung von G' wählt 0 Kanten eines bestimmten Schnitts] $\geq \left\lceil \frac{\ell}{\sqrt{2}} \right\rceil \cdot \frac{\left\lceil \frac{\ell}{\sqrt{2}} \right\rceil - 1}{\ell \cdot (\ell - 1)} \geq \frac{1}{2}$

$P(\ell)$:= Pr[FRMC findet minimalen Schnitt in Graphen mit ℓ Knoten]

$$P(\ell) \geq 1 - \left(1 - \frac{1}{2} \cdot P\left(\left\lceil \frac{\ell}{\sqrt{2}} \right\rceil\right)\right) \cdot \left(1 - \frac{1}{2} \cdot P\left(\left\lceil \frac{\ell}{\sqrt{2}} \right\rceil\right)\right)$$ für $\ell \geq 7$

(für $\ell \leq 6$ gilt $P(\ell)) = 1$)

Pr[G' enthält alle Kanten von min. Schnitt in G]

Pr[FRMC findet min. Schnitt C_1 in G_1]

Pr[C_1 ist min. Schnitt in G]

wenn $n \leq 6$ **dann**
  berechne direkt deterministisch einen MinCut
sonst
  $\ell \leftarrow \left\lceil \frac{n}{\sqrt{2}} \right\rceil$
  $G_1 \leftarrow$ Random MinCut(bis ℓ Knoten übrig)
  $G_2 \leftarrow$ Random MinCut(bis ℓ Knoten übrig)
  $C_1 \leftarrow$ Fast Random MinCut(G_1) (rekursiv)
  $C_2 \leftarrow$ Fast Random MinCut(G_2) (rekursiv)
  Gib den kleineren der beiden Schnitte C_1 und C_2 aus.

Fast Random MinCut (FRMC)

Satz 8.10: Wahrscheinlichkeit, dass FRMC einen minimalen Schnitt findet, ist in $\Omega\left(\frac{1}{\log n}\right)$.

Beweisskizze: Sei k Größe eines minimalen Schnitts in G.

Annahme: Es gibt Graph G', der ℓ Knoten besitzt, aus G durch Verschmelzung von Knoten hervorgegangen ist und einen Schnitt mit k Kanten besitzt.

Nach Folgerung 8.8:

Pr[Berechnung von G' wählt 0 Kanten eines bestimmten Schnitts] $\geq \left\lceil \frac{\ell}{\sqrt{2}} \right\rceil \cdot \frac{\left\lceil \frac{\ell}{\sqrt{2}} \right\rceil - 1}{\ell \cdot (\ell - 1)} \geq \frac{1}{2}$

$P(\ell)$:= Pr[FRMC findet minimalen Schnitt in Graphen mit ℓ Knoten]

$$P(\ell) \geq 1 - \left(1 - \frac{1}{2} \cdot P\left(\left\lceil \frac{\ell}{\sqrt{2}} \right\rceil\right)\right) \cdot \left(1 - \frac{1}{2} \cdot P\left(\left\lceil \frac{\ell}{\sqrt{2}} \right\rceil\right)\right)$$

für $\ell \geq 7$

(für $\ell \leq 6$ gilt $P(\ell)) = 1$)

Pr[G' enthält alle Kanten von min. Schnitt in G]

Pr[FRMC findet min. Schnitt C_1 in G_1]

Pr[C_1 ist min. Schnitt in G]

Pr[C_1 ist nicht min. Schnitt in G]

Pr[C_2 ist nicht min. Schnitt in G] (Analog wie C_1)

wenn $n \leq 6$ **dann**
  berechne direkt deterministisch einen MinCut
sonst
  $\ell \leftarrow \left\lceil \frac{n}{\sqrt{2}} \right\rceil$
  $G_1 \leftarrow$ Random MinCut(bis ℓ Knoten übrig)
  $G_2 \leftarrow$ Random MinCut(bis ℓ Knoten übrig)
  $C_1 \leftarrow$ Fast Random MinCut(G_1) (rekursiv)
  $C_2 \leftarrow$ Fast Random MinCut(G_2) (rekursiv)
  Gib den kleineren der beiden Schnitte C_1 und C_2 aus.

Institut für Theoretische Informatik
Prof. Dr. Dorothea Wagner

Fast Random MinCut (FRMC)

Karlsruher Institut für Technologie

Satz 8.10: Wahrscheinlichkeit, dass FRMC einen minimalen Schnitt findet, ist in $\Omega\left(\frac{1}{\log n}\right)$.

Beweisskizze: Sei k Größe eines minimalen Schnitts in G.

Annahme: Es gibt Graph G', der ℓ Knoten besitzt, aus G durch Verschmelzung von Knoten hervorgegangen ist und einen Schnitt mit k Kanten besitzt.

Nach Folgerung 8.8:

Pr[Berechnung von G' wählt 0 Kanten eines bestimmten Schnitts] $\geq \left\lceil \frac{\ell}{\sqrt{2}} \right\rceil \cdot \frac{\left\lceil \frac{\ell}{\sqrt{2}} \right\rceil - 1}{\ell \cdot (\ell - 1)} \geq \frac{1}{2}$

$P(\ell)$:= Pr[FRMC findet minimalen Schnitt in Graphen mit ℓ Knoten]

$$P(\ell) \geq 1 - \left(1 - \frac{1}{2} \cdot P\left(\left\lceil \frac{\ell}{\sqrt{2}} \right\rceil\right)\right) \cdot \left(1 - \frac{1}{2} \cdot P\left(\left\lceil \frac{\ell}{\sqrt{2}} \right\rceil\right)\right)$$

für $\ell \geq 7$
(für $\ell \leq 6$ gilt $P(\ell)) = 1$)

Pr[G' enthält alle Kanten von min. Schnitt in G]

Pr[FRMC findet min. Schnitt C_1 in G_1]

Pr[C_1 ist min. Schnitt in G]

Pr[C_1 ist nicht min. Schnitt in G]

Pr[C_2 ist nicht min. Schnitt in G] (Analog wie C_1)

Pr[C_1 oder C_2 ist min. Schnitt in G]

wenn $n \leq 6$ **dann**
  berechne direkt deterministisch einen MinCut
sonst
  $\ell \leftarrow \left\lceil \frac{n}{\sqrt{2}} \right\rceil$
  $G_1 \leftarrow$ Random MinCut(bis ℓ Knoten übrig)
  $G_2 \leftarrow$ Random MinCut(bis ℓ Knoten übrig)
  $C_1 \leftarrow$ Fast Random MinCut(G_1) (rekursiv)
  $C_2 \leftarrow$ Fast Random MinCut(G_2) (rekursiv)
  Gib den kleineren der beiden Schnitte C_1 und C_2 aus.

Fast Random MinCut (FRMC)

Satz 8.10: Wahrscheinlichkeit, dass FRMC einen minimalen Schnitt findet, ist in $\Omega\left(\frac{1}{\log n}\right)$.

Beweisskizze: Sei k Größe eines minimalen Schnitts in G.

Annahme: Es gibt Graph G', der ℓ Knoten besitzt, aus G durch Verschmelzung von Knoten hervorgegangen ist und einen Schnitt mit k Kanten besitzt.

Nach Folgerung 8.8:

Pr[Berechnung von G' wählt 0 Kanten eines bestimmten Schnitts] $\geq \left\lceil \frac{\ell}{\sqrt{2}} \right\rceil \cdot \frac{\left\lceil \frac{\ell}{\sqrt{2}} \right\rceil - 1}{\ell \cdot (\ell - 1)} \geq \frac{1}{2}$

$P(\ell)$:= Pr[FRMC findet minimalen Schnitt in Graphen mit ℓ Knoten]

$$P(\ell) \geq 1 - \left(1 - \frac{1}{2} \cdot P\left(\left\lceil \frac{\ell}{\sqrt{2}} \right\rceil\right)\right)^2 = P\left(\left\lceil \frac{\ell}{\sqrt{2}} \right\rceil\right) - \frac{1}{4} \cdot P\left(\left\lceil \frac{\ell}{\sqrt{2}} \right\rceil\right)^2$$

wenn $n \leq 6$ **dann**
- berechne direkt deterministisch einen MinCut

sonst
- $\ell \leftarrow \left\lceil \frac{n}{\sqrt{2}} \right\rceil$
- $G_1 \leftarrow$ Random MinCut(bis ℓ Knoten übrig)
- $G_2 \leftarrow$ Random MinCut(bis ℓ Knoten übrig)
- $C_1 \leftarrow$ Fast Random MinCut(G_1) (rekursiv)
- $C_2 \leftarrow$ Fast Random MinCut(G_2) (rekursiv)
- Gib den kleineren der beiden Schnitte C_1 und C_2 aus.

Fast Random MinCut (FRMC)

Satz 8.10: Wahrscheinlichkeit, dass FRMC einen minimalen Schnitt findet, ist in $\Omega\left(\frac{1}{\log n}\right)$.

Beweisskizze: Sei k Größe eines minimalen Schnitts in G.

Annahme: Es gibt Graph G', der ℓ Knoten besitzt, aus G durch Verschmelzung von Knoten hervorgegangen ist und einen Schnitt mit k Kanten besitzt.

Nach Folgerung 8.8:

Pr[Berechnung von G' wählt 0 Kanten eines bestimmten Schnitts] $\geq \left\lceil \frac{\ell}{\sqrt{2}} \right\rceil \cdot \frac{\left\lceil \frac{\ell}{\sqrt{2}} \right\rceil - 1}{\ell \cdot (\ell - 1)} \geq \frac{1}{2}$

$P(\ell)$:= Pr[FRMC findet minimalen Schnitt in Graphen mit ℓ Knoten]

$$P(\ell) \geq 1 - \left(1 - \frac{1}{2} \cdot P\left(\left\lceil \frac{\ell}{\sqrt{2}} \right\rceil\right)\right)^2 = P\left(\left\lceil \frac{\ell}{\sqrt{2}} \right\rceil\right) - \frac{1}{4} \cdot P\left(\left\lceil \frac{\ell}{\sqrt{2}} \right\rceil\right)^2$$

Für $\ell = \sqrt{2^{k+1}}$ folgt

$$P\left(\sqrt{2^{k+1}}\right) \geq P\left(\left(\sqrt{2}\right)^k\right) - \frac{1}{4} \cdot P\left(\left(\sqrt{2}\right)^k\right)^2$$

wenn $n \leq 6$ **dann**
  berechne direkt deterministisch einen MINCUT
sonst
  $\ell \leftarrow \left\lceil \frac{n}{\sqrt{2}} \right\rceil$
  $G_1 \leftarrow$ RANDOM MINCUT(bis ℓ Knoten übrig)
  $G_2 \leftarrow$ RANDOM MINCUT(bis ℓ Knoten übrig)
  $C_1 \leftarrow$ FAST RANDOM MINCUT(G_1) (rekursiv)
  $C_2 \leftarrow$ FAST RANDOM MINCUT(G_2) (rekursiv)
  Gib den kleineren der beiden Schnitte C_1 und C_2 aus.

Fast Random MinCut (FRMC)

Satz 8.10: Wahrscheinlichkeit, dass FRMC einen minimalen Schnitt findet, ist in $\Omega\left(\frac{1}{\log n}\right)$.

Beweisskizze: Sei k Größe eines minimalen Schnitts in G.

Annahme: Es gibt Graph G', der ℓ Knoten besitzt, aus G durch Verschmelzung von Knoten hervorgegangen ist und einen Schnitt mit k Kanten besitzt.

Nach Folgerung 8.8:

Pr[Berechnung von G' wählt 0 Kanten eines bestimmten Schnitts] $\geq \left\lceil \frac{\ell}{\sqrt{2}} \right\rceil \cdot \frac{\left\lceil \frac{\ell}{\sqrt{2}} \right\rceil - 1}{\ell \cdot (\ell - 1)} \geq \frac{1}{2}$

$P(\ell) :=$ Pr[FRMC findet minimalen Schnitt in Graphen mit ℓ Knoten]

$$P(\ell) \geq 1 - \left(1 - \frac{1}{2} \cdot P\left(\left\lceil \frac{\ell}{\sqrt{2}} \right\rceil\right)\right)^2 = P\left(\left\lceil \frac{\ell}{\sqrt{2}} \right\rceil\right) - \frac{1}{4} \cdot P\left(\left\lceil \frac{\ell}{\sqrt{2}} \right\rceil\right)^2$$

Für $\ell = \sqrt{2^{k+1}}$ folgt

$$P\left(\sqrt{2^{k+1}}\right) \geq P\left(\left(\sqrt{2}\right)^k\right) - \frac{1}{4} \cdot P\left(\left(\sqrt{2}\right)^k\right)^2$$

Man kann zeigen, dass

$P\left(\left(\sqrt{2}\right)^k\right) \in \Omega\left(\frac{1}{k}\right)$ und damit $P(\ell) \in \Omega\left(\frac{1}{\log \ell}\right)$

Setze nun $\ell = \sqrt{2^{k+1}}$, dann folgt

$$P\left(\sqrt{2^{k+1}}\right) \geq P\left(\left(\sqrt{2}\right)^k\right) - \frac{1}{4} \cdot P\left(\left(\sqrt{2}\right)^k\right)^2,$$

also wenn

$$s(k) := P\left(\left(\sqrt{2}\right)^k\right), \quad \text{so ist} \quad s(k+1) \geq s(k) - \frac{1}{4} \cdot s(k)^2 .$$

Wenn man nun

$$q(k) := \frac{4}{s(k) - 1} \quad \text{setzt, d.h.}$$

$$s(k) = \frac{4}{q(k) + 1} \quad \text{, dann folgt:}$$

$$\begin{aligned}
s(k+1) = \frac{4}{q(k+1)+1} &\geq \frac{4}{q(k)+1} - \frac{4}{(q(k)+1)^2} \\
\Longrightarrow \quad q(k)+1 &\geq q(k+1)+1 - \frac{q(k+1)+1}{q(k)+1} \\
&= (q(k+1)+1) \cdot \left(1 - \frac{1}{q(k)+1}\right) \\
\Longrightarrow \quad q(k+1)+1 &\leq (q(k)+1) \cdot \left(\frac{1}{1 - \frac{1}{q(k)+1}}\right) \\
q(k+1) &\leq q(k) + 1 - 1 + \frac{\frac{1}{q(k)+1}}{1 - \frac{1}{q(k)+1}} \\
&= \frac{q(k) + \frac{1}{q(k)+1}}{1 - \frac{1}{q(k)+1}} \\
&= q(k) + 1 + \frac{1}{q(k)} .
\end{aligned}$$

Induktiv lässt sich nun zeigen, dass

$$q(k) < k + \sum_{i=1}^{k-1} \frac{1}{i} + 3 \quad \in \quad \Theta(k + \log k)$$

Institut für Theoretische Informatik
Prof. Dr. Dorothea Wagner

Maximum Satisfiability Problem

Institut für Theoretische Informatik
Prof. Dr. Dorothea Wagner

Problemdefinition

Problem MAXIMUM SATISFIABILITY (MAXSAT):
Gegeben: Menge von m Klauseln über n Variablen.
Gesucht: Wahrheitsbelegung, die eine maximale Anzahl von Klauseln erfüllt.

Bereits $\mathcal{NP}$-schwer, wenn Anzahl der Literale auf zwei pro Klausel beschränkt.

→ MAX-2-SAT

Beispiel:

1. Klausel: $X_1 \vee \overline{X_2}$ 2. Klausel: $\overline{X_1} \vee \overline{X_2}$

3. Klausel: $X_1 \vee X_2$ 4. Klausel: $\overline{X_1} \vee X_3$

5. Klausel: $X_2 \vee \overline{X_3}$

Nicht alle Klauseln sind gleichzeitig erfüllbar:

- Für $X_1 = \mathit{falsch}$ kann 1. Klausel nicht mit 3. Klausel gleichzeitig erfüllt sein.
- Für $X_1 = \mathit{wahr}$ kann 5. Klausel nicht mit 2. und 4. Klausel gleichzeitig erfüllt sein.

Optimale Belegung: $X_1 = \mathit{wahr}$, $X_2 = \mathit{falsch}$, $X_3 = \mathit{wahr}$

Algorithmus Random Sat

Karlsruher Institut für Technologie

Vorgehen: Für jede Variable $x \in V$ setze $x :=$ *wahr* mit der Wahrscheinlichkeit $\frac{1}{2}$.

Satz 8.16.
Für eine Instanz I von MAX SAT mit m Klauseln, in der jede Klausel mindestens k Literale enthält, erfüllt der erwartete Wert der Lösung von RANDOM SAT:

$$E[X_{RS}(I)] \geq \left(1 - \frac{1}{2^k}\right) \cdot m,$$

wobei $X_{RS}(I)$ die Zufallsvariable bezeichnet, die den Wert der Lösung von RANDOM SAT bei der Eingabe von I angibt.

Beweis:

- Wahrscheinlichkeit, dass Klausel mit k Literalen nicht erfüllt wird, ist $\frac{1}{2^k}$.
- Entsprechend ist Wahrscheinlichkeit, dass Klausel mit mindestens k Literalen erfüllt wird mindestens $1 - \frac{1}{2^k}$.
- Damit ist der erwartete Beitrag einer Klausel zu $E[X_{RS}(I)]$ mindestens $1 - \frac{1}{2^k}$.
- Es folgt die Behauptung.

Algorithmus Random Sat

Vorgehen: Für jede Variable $x \in V$ setze $x := wahr$ mit der Wahrscheinlichkeit $\frac{1}{2}$.

Satz 8.16.
Für eine Instanz I von MAX SAT mit m Klauseln, in der jede Klausel mindestens k Literale enthält, erfüllt der erwartete Wert der Lösung von RANDOM SAT:

$$E[X_{RS}(I)] \geq \left(1 - \frac{1}{2^k}\right) \cdot m,$$

wobei $X_{RS}(I)$ die Zufallsvariable bezeichnet, die den Wert der Lösung von RANDOM SAT bei der Eingabe von I angibt.

Korollar 8.17: RANDOM SAT ist 2-approximativ, d.h.

$$\frac{OPT(I)}{E[X_{\mathrm{RS}(I)}]} \leq 2$$