Randomness and Computation

or, “Randomized Algorithms”

Heng Guo

(Based on slides by M. Cryan)

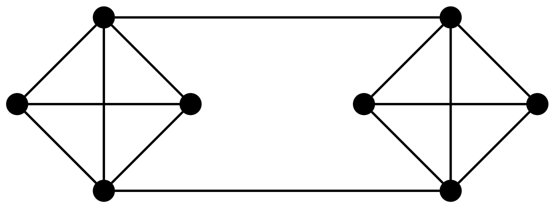
Karger’s contraction algorithm

“Global Min-cuts in RNC and Other Ramifications of a Simple Mincut Algorithm”, by David R. Karger, SODA 1993.

> *Repeatedly, choose an edge uniformly at random (from the not-yet contracted edges) and contract its endpoints.*
> *When there are just two “vertices” left, return that cut.*

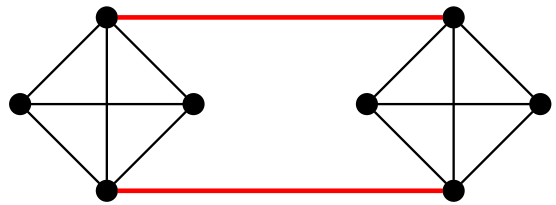
We will show that this algorithm finds the minimum cut with high probability in time $O(n^2 \log n)$.

Example



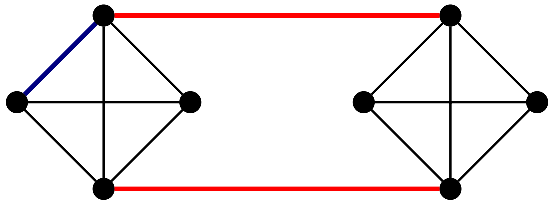
The min cut has size 2.

Example



The min cut has size 2.

Example

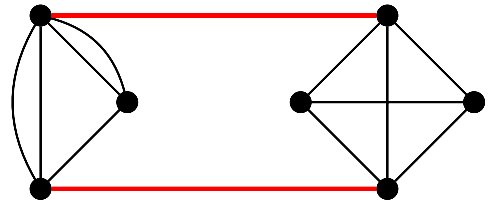


The algorithm randomly picks one edge out of 14.

We hope to avoid the min cut.

In this case the “bad” thing happens with probability $\frac{2}{14}$.

Example

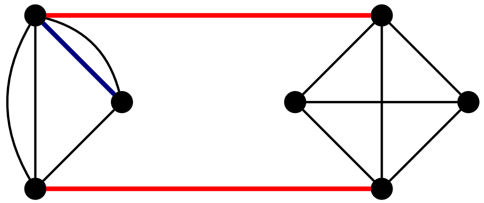


Contraction:

merge the endpoints of an edge into one.

Parallel edges are preserved, and self-loops removed.

Example

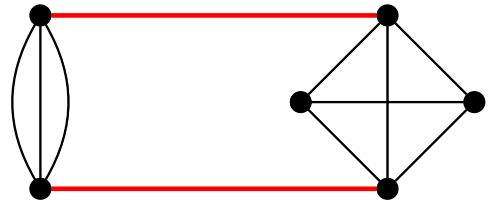


Contraction:

merge the endpoints of an edge into one.

Parallel edges are preserved, and self-loops removed.

Example

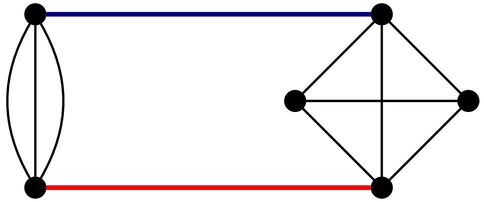


Contraction:

merge the endpoints of an edge into one.

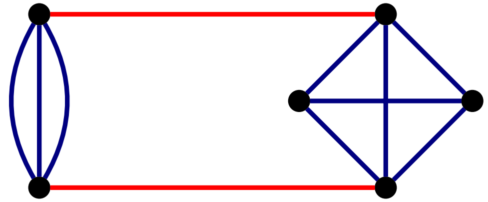
Parallel edges are preserved, and self-loops removed.

Example



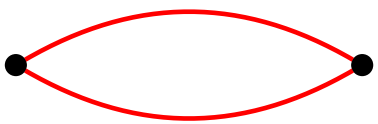
If we contract a cut edge, then we will not find that cut.

Example



Ideally, we should contract all edges except the min cut.

Example



Ideally, we should contract all edges except the min cut.

Implementation of the algorithm

Naive implementation of contractions would require complicated data structures to keep track of everything.

An equivalent way of looking at the algorithm is to pick a random permutation of all edges first, and then contracting edges from the first to the last.

Thus, what we need to find is the shortest prefix of the permutation such that they induce two connected components.

Finding connected components takes $O(m)$ time. Thus, by a binary search, this will take time

$$O(m) + O(m/2) + O(m/4) + \cdots = O(m).$$

Denote by KargerOneTrial one iteration of the above.

Karger's contraction algorithm - analysis

Let k be the size of a min cut of G, let $S \subset V$ be a *specific* partition where C_S, the set of edges between S and $V \setminus S$, is of cardinality k.

Karger's contraction algorithm - analysis

Let k be the size of a min cut of G, let $S \subset V$ be a *specific* partition where C_S, the set of edges between S and $V \setminus S$, is of cardinality k.

We must have $deg(v) \geq k$ for every $v \in V$. (Why?)

Karger's contraction algorithm - analysis

Let k be the size of a min cut of G, let $S \subset V$ be a *specific* partition where C_S, the set of edges between S and $V \setminus S$, is of cardinality k.

We must have $deg(v) \geq k$ for every $v \in V$. (Why?)

The algorithm chooses a sequence of random edges $e_1, e_2, \ldots$

Let E_j be the event that $e_j \notin C_S$. ("Good" event.)

Karger's contraction algorithm - analysis

Let k be the size of a min cut of G, let $S \subset V$ be a *specific* partition where C_S, the set of edges between S and $V \setminus S$, is of cardinality k.

We must have $deg(v) \geq k$ for every $v \in V$. (Why?)

The algorithm chooses a sequence of random edges $e_1, e_2, \ldots$

Let E_j be the event that $e_j \notin C_S$. ("Good" event.)

Calculating $\Pr[E_1]$, there are k "cut-edges" (from C_S), and at least $k \cdot n/2$ edges overall. Hence

$$\Pr[E_1] \geq 1 - \frac{2k}{kn} \geq 1 - \frac{2}{n}.$$

We next calculate $\Pr[E_2 \mid E_1]$, the probability that the *2nd* edge avoids C_S, *conditional that the first edge was outside C_S.*

Karger's contraction algorithm - analysis

$\Pr[E_2 \mid E_1]$:

- Still have all k C_S edges (since we assumed E_1).
- Graph now has $(n-1)$ "vertices", each having degree $\geq k$ (why?); hence the graph now has at least $k \cdot (n-1)/2$ edges overall.

Hence

$$\Pr[E_2 \mid E_1] \;\geq\; 1 - \frac{2k}{(n-1)k} \;=\; 1 - \frac{2}{n-1}.$$

Next we will generalise this bound, namely, for any initial sequence of j edge-choices satisfying $\cap_{i=1}^{j} E_i$, we give a lower bound on

$$\Pr[E_{j+1} \mid E_1 \cap \ldots \cap E_j].$$

Karger's contraction Algorithm - Analysis

For any $j = 1, \ldots, n-3$, we analyse the *conditional* probability $\Pr[E_{j+1} \mid E_1 \cap \ldots \cap E_j]$:

- All k C_S edges still remain (since we assume $E_1 \cap \ldots \cap E_j$).
- How many edges have been removed? At least j

  Not exactly j, as we might have contracted a "parallel edge" earlier on, which has the effect of removing more than one edge from the graph.
- How many vertices have been removed? Exactly j

  The graph now has $(n-j)$ "vertices", and each must have degree $\geq k$ (why?); hence the graph now has at least $k \cdot (n-j)/2$ edges overall.

Therefore

$$\Pr[E_{j+1} \mid E_1 \cap \ldots \cap E_j] \;\geq\; 1 - \frac{2k}{(n-j)k} \;=\; 1 - \frac{2}{n-j}.$$

Karger's contraction Algorithm - Analysis

We hope that our contraction of random edges will lead us to a scenario where we are left with two "vertices" without contracting any of the C_S edges (min-cut) on the way.

If we achieve this, then one "vertex" will contain all of S, the other "vertex" all of $V \setminus S$, and the parallel edges between them are exactly the edges in the min-cut C_S.

The probability we get to this nice scenario is the probability that E_1 holds, *and* (conditioned on that) that E_2 also holds, *and* (conditioned on $E_1 \cap E_2$) ...) that E_3 also holds, and ... Formally,

$$\begin{aligned}
\Pr[\cap_{j=1}^{n-2} E_j] &= \Pr[E_1] \cdot \Pr[E_2 \mid E_1] \cdot \ldots \cdot \Pr[E_{n-2} \mid \cap_{i=1}^{n-3} E_i] \\
&= \prod_{j=1}^{n-2} \Pr[E_j \mid \cap_{i=1}^{j-1} E_i] \\
&\geq \prod_{j=1}^{n-2} \left(1 - \frac{2}{n-(j-1)}\right) = \prod_{j=3}^{n} \left(1 - \frac{2}{j}\right)
\end{aligned}$$

Karger's contraction Algorithm - Analysis

Expanding $\prod_{j=3}^{n}\left(1 - \frac{2}{j}\right)$, we have

$$\begin{aligned}
&\prod_{j=3}^{n} \frac{j-2}{j} \\
&= \left(\frac{1}{3}\right)\left(\frac{2}{4}\right)\left(\frac{3}{5}\right)\left(\frac{4}{6}\right)\cdots\left(\frac{n-4}{n-2}\right)\left(\frac{n-3}{n-1}\right)\left(\frac{n-2}{n}\right) \\
&= \frac{2}{n(n-1)}
\end{aligned}$$

So the probability that a single "run" of KargerOneTrial generates a cut which is minimal for the original graph is *at least* $\frac{2}{n(n-1)}$.

Could be more in practice. *(WHY?)*

Karger's contraction Algorithm - Analysis

Expanding $\prod_{j=3}^{n}\left(1-\frac{2}{j}\right)$, we have

$$\begin{aligned}\prod_{j=3}^{n}\frac{j-2}{j} &= \left(\frac{1}{\cancel{3}}\right)\left(\frac{2}{\cancel{4}}\right)\left(\frac{\cancel{3}}{\cancel{5}}\right)\left(\frac{\cancel{4}}{\cancel{6}}\right)\cdots\left(\frac{\cancel{n-4}}{\cancel{n-2}}\right)\left(\frac{\cancel{n-3}}{n-1}\right)\left(\frac{\cancel{n-2}}{n}\right)\\ &= \frac{2}{n(n-1)}\end{aligned}$$

So the probability that a single "run" of KargerOneTrial generates a cut which is minimal for the original graph is *at least* $\frac{2}{n(n-1)}$.

Could be more in practice. *(WHY?)*

Karger's contraction Algorithm - Repeated iterations

We can improve our result by running KargerOneTrial many times, and returning the minimum of all the different cuts.

If we do k trials, the probability that *none* is a min cut is *at most* $\left(1-\frac{2}{n(n-1)}\right)^k$.

Karger's contraction Algorithm - Repeated iterations

We can improve our result by running KargerOneTrial many times, and returning the minimum of all the different cuts.

If we do k trials, the probability that *none* is a min cut is *at most* $\left(1-\frac{2}{n(n-1)}\right)^k$.
We can relate this to e using $(1+\frac{1}{n})^n < e < (1+\frac{1}{n})^{n+1}$:

$$\Rightarrow\ e^{-1} > \left(1+\tfrac{1}{n-1}\right)^{-n} = \left(1-\tfrac{1}{n}\right)^n.$$

Thus $\left(1-\frac{2}{n(n-1)}\right)^{\frac{n(n-1)}{2}} < e^{-1}$. Taking $k = c\cdot\frac{n(n-1)}{2}\cdot\ln(n)$, we get

$$\left(1-\frac{2}{n(n-1)}\right)^k = \left(\left(1-\frac{2}{n(n-1)}\right)^{\frac{n(n-1)}{2}}\right)^{c\ln(n)} < (e^{-1})^{c\ln(n)} = \frac{1}{n^c}.$$

Wrapping up

- Probability tools used in our analysis were simple: we have used conditional probability iteratively:

$$\begin{aligned}\Pr[\cap_{j=1}^{n-2}E_j] &= \Pr[\cap_{j=2}^{n-2}E_j \mid E_1]\cdot\Pr[E_1]\\ &= \Pr[\cap_{j=3}^{n-2}E_j \mid E_1\cap E_2]\cdot\Pr[E_1\cap E_2 \mid E_1]\cdot\Pr[E_1]\\ &= \ldots\end{aligned}$$

  (also used simple inequalities relating $(1+\frac{1}{x})^x$ and e)

- No approximation guarantee - analysis does not address the quality of C_S when it fails to be optimum.

#Min-Cut

We have shown that for a particular cut C, the probability of finding C is at least $\frac{2}{n(n-1)}$. This implies that $|\mathcal{C}| \leq \frac{n(n-1)}{2}$, where $\mathcal{C}$ is the set of all Min-Cut.

#Min-Cut

We have shown that for a particular cut C, the probability of finding C is at least $\frac{2}{n(n-1)}$. This implies that $|\mathcal{C}| \leq \frac{n(n-1)}{2}$, where $\mathcal{C}$ is the set of all Min-Cut.

Let F_C be the event of finding C. For $C \neq C'$, $\Pr[F_C \cap F_{C'}] = 0$.

Thus, $\Pr\left[\cup_{C\in\mathcal{C}} F_C\right] = \sum_{C\in\mathcal{C}} \Pr[F_C] \leq 1$.

#Min-Cut

We have shown that for a particular cut C, the probability of finding C is at least $\frac{2}{n(n-1)}$. This implies that $|\mathcal{C}| \leq \frac{n(n-1)}{2}$, where $\mathcal{C}$ is the set of all Min-Cut.

Let F_C be the event of finding C. For $C \neq C'$, $\Pr[F_C \cap F_{C'}] = 0$.

Thus, $\Pr\left[\cup_{C\in\mathcal{C}} F_C\right] = \sum_{C\in\mathcal{C}} \Pr[F_C] \leq 1$.

On the other hand, $\sum_{C\in\mathcal{C}} \Pr[F_C] \geq \sum_{C\in\mathcal{C}} \frac{2}{n(n-1)} = \frac{2|\mathcal{C}|}{n(n-1)}$.

#Min-Cut

We have shown that for a particular cut C, the probability of finding C is at least $\frac{2}{n(n-1)}$. This implies that $|\mathcal{C}| \leq \frac{n(n-1)}{2}$, where $\mathcal{C}$ is the set of all Min-Cut.

Let F_C be the event of finding C. For $C \neq C'$, $\Pr[F_C \cap F_{C'}] = 0$.

Thus, $\Pr\left[\cup_{C\in\mathcal{C}} F_C\right] = \sum_{C\in\mathcal{C}} \Pr[F_C] \leq 1$.

On the other hand, $\sum_{C\in\mathcal{C}} \Pr[F_C] \geq \sum_{C\in\mathcal{C}} \frac{2}{n(n-1)} = \frac{2|\mathcal{C}|}{n(n-1)}$.

Thus, $|\mathcal{C}| \leq \frac{n(n-1)}{2}$.

A u.a.r. cut is minimum with probability $\frac{|\mathcal{C}|}{2^n-1} \leq \frac{n(n-1)}{2(2^n-1)} = O\left(\frac{n^2}{2^n}\right)$. Hence Karger's algorithm succeeds with probability exponentially higher than a random cut.

#Min-Cut

We have shown that for a particular cut C, the probability of finding C is at least $\frac{2}{n(n-1)}$. This implies that $|\mathcal{C}| \leq \frac{n(n-1)}{2}$, where $\mathcal{C}$ is the set of all Min-Cut.

Let F_C be the event of finding C. For $C \neq C'$, $\Pr[F_C \cap F_{C'}] = 0$.

Thus, $\Pr\left[\cup_{C \in \mathcal{C}} F_C\right] = \sum_{C \in \mathcal{C}} \Pr[F_C] \leq 1$.

On the other hand, $\sum_{C \in \mathcal{C}} \Pr[F_C] \geq \sum_{C \in \mathcal{C}} \frac{2}{n(n-1)} = \frac{2|\mathcal{C}|}{n(n-1)}$.

Thus, $|\mathcal{C}| \leq \frac{n(n-1)}{2}$.

A u.a.r. cut is minimum with probability $\frac{|\mathcal{C}|}{2^n-1} \leq \frac{n(n-1)}{2(2^n-1)} = O\left(\frac{n^2}{2^n}\right)$. Hence Karger's algorithm succeeds with probability exponentially higher than a random cut.

This is tight! Consider a cycle of length n.

Other use of random contraction

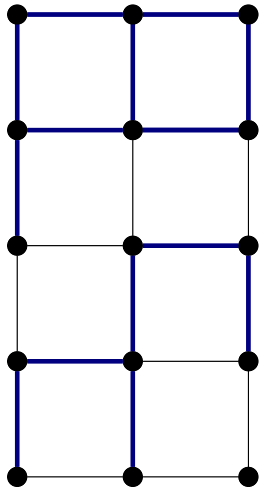
Each edge fails with prob. p independently. Can we compute or approximate

$$\Pr[G_p \text{ is connected}]?$$

Exact evaluation is **#P**-complete (Valiant, 1979), so a polynomial-time algorithm is unlikely.

Based on random contractions, Karger (1999) gave the first polynomial-time randomised approximation algorithm for Unreliability, namely $1 - \Pr[G_p \text{ is connected}]$.

The first efficient algorithm for Reliability was found by G. and Jerrum (2018). However it is based a variant of the constructive version of the Lovász Local Lemma.



Expectation vs. Whp

There are two typical kinds of guarantees we will work with.

- Expectation. This includes the expected running time, expected output, etc.
- With high probability (whp or w.h.p.). The meaning of this can vary. Sometimes it means probability $1-o(1)$, which for example would include $1 - \frac{1}{\log n}$. Sometimes it is stronger, namely $1 - n^{-c}$.

Reading

Our next topic will be the "Coupon Collector" problem.

- Some of you may have seen the "Coupon Collector" problem in lower level classes.
- We will re-visit it, but as well as deriving the expected value, we will also bound the variance (2nd moment), and look at the implications of that.
- You might want to read sections 2.3, 2.4 and 3.3 of [MU] in advance (if your probability is rusty, also read 2.1, 2.2, 3.1 and 3.2)