Randomness and Computation
or, “Randomized Algorithms”

Mary Cryan

School of Informatics
University of Edinburgh
Karger’s Min-Cut Algorithm

Given an undirected, unweighted graph \( G = (V, E) \) with \(|V| = n, |E| = m\), compute a “min cut”; that is, a partition of \( E \) into two non-empty sets \( S, V \setminus S \), such that the following quantity is minimized:

\[
\{ e = (u, v) : u \in S, v \in V \setminus S \}
\]

Deterministic algorithms for this undirected problem:

- “Net Flow”: replace each edge with two opposing arcs. Select fixed source vertex \( s \). Find \( \min (s, t) \)-cut, for each \( t \in V \setminus \{s\} \) via a fast Network Flow algorithm (best is \( \Theta(mn) \), via a merge of Orlin’s and King, Rao, Tarjan’s algorithms). \((n - 1)\) individual calls \( \Rightarrow \Theta(mn^2) \) total.

- Best deterministic “no flows” algorithm is Stoer and Wagner’s \( \Theta(mn + n^2 \log(n)) \) algorithm.
Karger’s (randomised) Min-Cut Algorithm

**Algorithm** \( \text{KARGERONETRIAL}(G = (V, E)) \)

1. \( n' \leftarrow |V| \)
2. **while** \( (n' > 2) \) do
3. Draw \( e = (u, v) \) uniformly at random from \( E \)
4. Let \( \hat{v} \) be the “vertex” containing \( v \) (may just be \( \{v\} \))
5. Let \( \hat{u} \) be the “vertex” containing \( u \) (may just be \( \{u\} \))
6. Merge the two “vertices” \( \hat{v}, \hat{u} \).
7. \( E \leftarrow E \setminus \{(w, z) : w \in \hat{u}, z \in \hat{v}\} \)
8. **return** \( E, |E| \)

In line 7. we will definitely delete \( (u, v) \), but might delete other parallel edges between the supervertices to be contracted.

Can be implemented in about \( \Theta(m \log(m)) \) time by using sampling without replacement (from \( E \)) and a fast Disjoint Sets Data Structure. Karger’s paper shows how to use a nice trick with connected component calculations to evaluate the result in \( \Theta(m) \) time.
Karger’s Min-Cut 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 $v \in V$.

Let $E_j$ be the event that the $j$th-random edge is not in $C_S$.

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 $\Pr[E_2 \mid E_1]$, probability that the 2nd edge avoids $C_S$, conditional that the first edge was outside $C_S$. . . .
Karger’s Min-Cut 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 must have degree \( \geq k \) (discuss); 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 compute, for any initial sequence of \( j \) edge-choices satisfying \( \cap_{i=1}^j E_i \), a lower bound on \( \Pr[E_{j+1} \mid E_1 \cap \ldots \cap E_j] \ldots \)
Karger’s Min-Cut 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 $E_1 \cap \ldots \cap E_j$).
- We have removed at least $j$ edges before selecting the $j + 1$-th (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).
- We have absorbed exactly $j$ vertices so far, graph now has $(n - j)$ “vertices”, and each must have degree $\geq k$ (discuss); hence the graph now has at least $k \cdot (n - j)/2$ edges overall.

Hence

$$\Pr[E_{j+1} \mid E_1 \cap \ldots \cap E_j] \geq 1 - \frac{2k}{(n-j)k} = 1 - \frac{2}{n-j}.$$
Karger’s Min-Cut 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 both are exactly the edges in the min-cut $C_S$.

The probability we arrive here is the probability that $E_1$ holds, and (conditioned on that) that $E_2$ also holds, and (conditioned on $E_1 \cap E_2$) \ldots that $E_3$ also holds, and \ldots

Formally,

$$
\Pr[\cap_{j=1}^{n-2} E_j] = \Pr[E_1] \cdot \Pr[E_2 | E_1] \cdot \ldots \cdot \Pr[E_{n-2} | \cap_{i=1}^{n-3} E_i]
$$

$$
= \prod_{j=1}^{n-2} \Pr[E_j | \cap_{i=1}^{j-1} E_i]
$$

$$
= \prod_{j=1}^{n-2} \left(1 - \frac{2}{n-(j-1)}\right) = \prod_{j=3}^{n} \left(1 - \frac{2}{j}\right)
$$
Karger’s Min-Cut Algorithm - Analysis

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

$$\prod_{j=3}^{n} \frac{j-2}{j}$$

$$= \left(\frac{1}{3}\right) \left(\frac{2}{4}\right) \left(\frac{3}{5}\right) \left(\frac{4}{6}\right) \cdots \left(\frac{n-4}{n-2}\right) \left(\frac{n-3}{n-1}\right) \left(\frac{n-2}{n}\right)$$

$$= \frac{2}{n(n-1)}$$

So the probability that a single “run” of KARGER ONE TRIAL 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 Min-Cut Algorithm - Repeated iterations

We can improve the quality of the returned cut by running \textsc{KargerOneTrial} many times, and returning the minimum of all the different cuts.

If we do \( k \) trials, the probability that \textit{none} is a min cut is \textit{at most}

\[
\left( 1 - \frac{2}{n(n-1)} \right)^k.
\]

We can relate this to \( e \) using \( (1 + \frac{1}{n})^n < e < (1 + \frac{1}{n})^{n+1} \).

\[
\Rightarrow \left( 1 - \frac{2}{n(n-1)} \right)^{\frac{n(n-1)}{2}+1} < e^{-1},
\]

and taking \( k = \frac{n(n+1)}{2} \cdot \ln(n) \), we get

\[
\left( 1 - \frac{2}{n(n-1)} \right)^k = \left( 1 - \frac{2}{n(n-1)} \right)^{\frac{n(n+1)}{2}} \ln(n) < (e^{-1})^\ln(n) = \frac{1}{n}.
\]
Also of interest is the Max Cut of a given graph:

Given an undirected, unweighted graph \( G = (V, E) \) with \( |V| = n, |E| = m \), compute a “max cut”; that is, a partition of \( E \) into two non-empty sets \( S, V \setminus S \), such that the following quantity is maximized:

\[
\{ e = (u, v) : u \in S, v \in V \setminus S \}
\]

Well-known as one of the classical NP-complete problems. So we believe there is no polynomial-time algorithm to compute this exactly (not in \( \Theta(n^2m) \), not in \( \Theta(m^5n^9) \) etc).

We will show that every graph \( G = (V, E) \) has a cut of size at least \( |E|/2 \).
Max Cuts in Graphs

Consider the following Algorithm:

**Algorithm** $\text{RANDOM CUT}(G = (V, E))$

1. $S \leftarrow \emptyset$
2. for every $v \in V$ in fixed order do
3. Draw a value $b$ uniformly from $\{0, 1\}$.
4. if $(b = 1)$ then
5. $S \leftarrow S \cup \{v\}$
6. return $S, V \setminus S$

We are going to analyse this algorithm and show that $C_S$ (the number of edges between $S$ and $V \setminus S$) has expected size at least $|E|/2$. 

*RC (2016/17) – Lecture 4 – slide 11*
Max Cuts in Graphs

Theorem (6.3)

*For any given graph* \( G = (V, E) \), *there is some cut* \( (S, V \setminus S) \) *such that* \( |C_S| \geq |E|/2 \).

**Proof.**

We show that the *expected* cardinality of \( C_S \), \( E[|C_S|] \) is at least \( |E|/2 \) when \( S \) is a random subset of \( V \).

We can write

\[
E[|C_S|] = \frac{1}{2^n} \sum_{S: S \subseteq V} \sum_{e = (u, v) \in E} \mathbb{I}_{|\{u,v\} \cap S| = 1}.
\]

Switching summations,

\[
E[|C_S|] = \sum_{e = (u, v) \in E} \frac{1}{2^n} \sum_{S: S \subseteq V} \mathbb{I}_{|\{u,v\} \cap S| = 1}.
\]

For every \( e \in E \), it has 4 options wrt a randomly generated \( S \):

- \( u, v \in S \), \( u \in S, v \notin S \), \( u \notin S, v \in S \), and \( u \notin S, v \notin S \).

Probability 1/4 each.
Max Cuts in Graphs

Proof cont.
Hence, for a fixed $e \in E$,

$$\frac{1}{2^n} \sum_{S:S \subseteq V} \mathbb{I}_{|\{u,v\} \cap S|=1} = \frac{2}{4} = \frac{1}{2}.$$ 

Hence, summing over all $e \in E$,

$$E[|C_S|] = \sum_{e=(u,v) \in E} \frac{1}{2} = \frac{|E|}{2},$$

as claimed.
Now switch back to thinking of this as the expectation over random $S$. If the expected size is $|E|/2$, then certainly there is at least one cut of at least that size. 

\[\Box\]
Probabilistic Method

- We did not analyse the probability that RANDOM CUT gives a good (high cardinality) cut, and are not going to do that yet.
- Can de-randomise the algorithm using conditional probabilities.
- The proof that every graph has a cut of cardinality $\geq |E|/2$ is a very very simple example of the probabilistic method.
- With the probabilistic method, we use randomness and the laws of expectation to prove that certain structures must exist.
- More later in the course.
Karger’s paper also presented results on parallelisation of the random trial, plus details of how to adapt for weighted graphs. Worth a read: “Global Min cuts in \( \mathbb{RNC} \) and Other Ramifications of a Simple Min-Cut Algorithm", by David R. Karger, 1993 (in SODA)
http://people.csail.mit.edu/karger/Papers/mincut.ps

TCS “cheat sheet” is always useful
http://www.tug.org/texshowcase/cheat.pdf

Please try to read as much of Chapter 2 of “Probability and Computing" as you can.