Minimum Cuts in Graphs

Andreas Klappenecker

Texas A&M University
Multigraphs

Let \( G = (V, E) \) be a connected, undirected, loopfree multigraph with \( n \) vertices. A multigraph may contain multiple edges between two vertices, as the following example shows.
Definition

A **cut** in the multigraph $G = (V, E)$ is a partition of the vertex set $V$ into two disjoint nonempty sets $V = V_1 \cup V_2$. An edge with one end in $V_1$ and the other in $V_2$ is said to **cross the cut**.

Remark

*The term cut is chosen because the removal of the edges in a cut partitions the multigraph.*
Example of a Cut

Example

If we partition \( V = \{A, B, C, D, E, F\} \) into the sets

\[
V_1 = \{A, C\} \quad \text{and} \quad V_2 = \{B, D, E, F\},
\]

then this cut has five crossing edges, and removing these edges yields a disconnected multigraph.
Definition

The **size** of the cut is given by the number of edges crossing the cut. The above example shows a cut of size 5.
Goal

Determine the minimum size of a cut in a given multigraph $G$. 
We describe a very simple randomized algorithm for this purpose. If $e$ is an edge of a loopfree multigraph $G$, then the multigraph $G/e$ is obtained from $G$ by **contracting** the edge $e = \{x, y\}$, that is, we identify the vertices $x$ and $y$ and remove all resulting loops.
Key Observation

Remark

Note that any cut of $G / e$ induces a cut of $G$.

\[
\begin{align*}
&{A, B} \cup {D, E, F} \quad \text{in} \quad G / \{C, D\} \quad \text{induces the cut} \quad {A, B} \cup \{C, D, E, F\} \quad \text{in} \quad G. \\
&\text{In general, the vertices that have been identified in} \quad G / e \quad \text{are in the same partition of} \quad G.
\end{align*}
\]
Remark

The size of the minimum cut of $G/e$ is at least the size of the minimum cut of $G$, because all edges are kept.
We can use successive contractions to estimate the size of the minimum cut of $G$.

We can select uniformly at random one of the remaining edges and contract it until two vertices remain.

The cut determined by this algorithm contains precisely the edges that have not been contracted.

Counting the edges between the remaining two vertices yields an estimate of the size of the minimum cut of $G$. 
Step 1

Contract by \{E, F\}. Partition \{\{A\}, \{B\}, \{C\}, \{D\}, \{E, F\}\}.
Step 2

Contract by \{D, F\}. Partition \{\{A\}, \{B\}, \{C\}, \{D, E, F\}\}. 
Step 3

Contract by \( \{C, D\} \). Partition \( \{\{A\}, \{B\}, \{C, D, E, F\}\} \).
Step 4

Contract by \{B, D\}. Partition \\{\{A\}, \{B, C, D, E, F\}\}. 
Karger’s Minimum Cut Algorithm

Contract\((G)\)

**Require:** A connected loopfree multigraph \(G = (V, E)\) with at least 2 vertices.

1: while \(|V| > 2\) do
2: \textbf{Select} \(e \in E\) uniformly at random;
3: \(G := G/e;\)
4: end while
5: \textbf{return} \(|E|\).

**Ensure:** An upper bound on the minimum cut of \(G\).
Conditional Probability

\[ \Pr[E \cap F] = \Pr[E|F] \Pr[F] \]
Exercise

Prove the following straightforward consequence of the previous formula

\[
Pr \left[ \bigcap_{\ell=1}^{n} E_\ell \right] = \left( \prod_{m=2}^{n} Pr \left[ E_m \left| \bigcap_{\ell=1}^{m-1} E_\ell \right. \right] \right) Pr[E_1].
\]

If you expand the formula then you will immediately see the pattern.
Solution

Idea:

\[
\Pr[E_n \cap E_{n-1} \cap \cdots \cap E_1] = \Pr[E_n \mid E_{n-1} \cap \cdots \cap E_1] \Pr[E_{n-1} \cap \cdots \cap E_1] \\
\Pr[E_{n-1} \cap E_{n-2} \cap \cdots \cap E_1] = \Pr[E_{n-1} \mid E_{n-2} \cap \cdots \cap E_1] \Pr[E_{n-2} \cap \cdots \cap E_1] \\
\Pr[E_{n-2} \cap E_{n-3} \cap \cdots \cap E_1] = \Pr[E_{n-2} \mid E_{n-3} \cap \cdots \cap E_1] \Pr[E_{n-1} \cap \cdots \cap E_1] \\
\vdots \\
\Pr[E_2 \cap E_1] = \Pr[E_2 \mid E_1] \Pr[E_1]
\]

Rigorous proof: Induction.
Idea:

\[
\begin{align*}
\Pr[E_n \cap E_{n-1} \cap \cdots \cap E_1] &= \Pr[E_n | E_{n-1} \cap \cdots \cap E_1] \Pr[E_{n-1} \cap \cdots \cap E_1] \\
\Pr[E_{n-1} \cap E_{n-2} \cap \cdots \cap E_1] &= \Pr[E_{n-1} | E_{n-2} \cap \cdots \cap E_1] \Pr[E_{n-2} \cap \cdots \cap E_1] \\
\Pr[E_{n-2} \cap E_{n-3} \cap \cdots \cap E_1] &= \Pr[E_{n-2} | E_{n-3} \cap \cdots \cap E_1] \Pr[E_{n-1} \cap \cdots \cap E_1] \\
&\vdots \\
\Pr[E_2 \cap E_1] &= \Pr[E_2 | E_1] \Pr[E_1]
\end{align*}
\]

Rigorous proof: Induction.
Idea:

\[
\Pr[E_n \cap E_{n-1} \cap \cdots \cap E_1] = \Pr[E_n \mid E_{n-1} \cap \cdots \cap E_1] \Pr[E_{n-1} \cap \cdots \cap E_1]
\]
\[
\Pr[E_{n-1} \cap E_{n-2} \cap \cdots \cap E_1] = \Pr[E_{n-1} \mid E_{n-2} \cap \cdots \cap E_1] \Pr[E_{n-2} \cap \cdots \cap E_1]
\]
\[
\Pr[E_{n-2} \cap E_{n-3} \cap \cdots \cap E_1] = \Pr[E_{n-2} \mid E_{n-3} \cap \cdots \cap E_1] \Pr[E_{n-1} \cap \cdots \cap E_1]
\]
\[
\vdots
\]
\[
\Pr[E_2 \cap E_1] = \Pr[E_2 \mid E_1] \Pr[E_1]
\]

Rigorous proof: Induction.
Suppose that the multigraph has a uniquely determined minimum cut. If the algorithm selects in this case any edge crossing this cut, then the algorithm will fail to produce the correct result. The analysis is largely guided by this observation.

Exercise

Give an example of a connected, loopfree multigraph with at least four vertices that has a uniquely determined minimum cut.
Remark

Let $G = (V, E)$ be a loopfree connected multigraph with $n = |V|$ vertices. Note that each contraction reduces the number of vertices by one, so the algorithm terminates after $n - 2$ steps.
Suppose that $C$ is a particular minimum cut of $G$. Let $E_i$ denote the event that the algorithm selects in the $i$th step an edge that does not cross the cut $C$. Therefore, the probability that no edge crossing the cut $C$ is ever picked during an execution of the algorithm is

$$\Pr \left[ \bigcap_{j=1}^{n-2} E_j \right].$$

This probability can be calculated by

$$\Pr \left[ \bigcap_{m=1}^{n-2} E_m \right] = \left( \prod_{m=2}^{n-2} \Pr \left[ E_m \bigg| \bigcap_{\ell=1}^{m-1} E_{\ell} \right] \right) \Pr[ E_1 ].$$
Suppose that the size of the minimum cut is $k$.

This means that the degree of each vertex is at least $k$, hence there exist at least $kn/2$ edges.

The probability to select an edge crossing the cut $C$ in the first step is at most $k/(kn/2) = 2/n$. Consequently,

$$\Pr[E_1] \geq 1 - \frac{2}{n} = \frac{n - 2}{n}.$$
Similarly, at the beginning of the $m$th step, with $m \geq 2$, there are $n - m + 1$ remaining vertices.

The minimum cut is still at least $k$, hence the multigraph has at this stage at least $k(n - m + 1)/2$ edges. Assuming that none of the edges crossing $C$ was selected in an earlier step, the probability to select an edge crossing the cut $C$ is $2/(n - m + 1)$.

It follows that

$$\Pr \left[ E_m \left| \bigcap_{j=1}^{m-1} E_j \right. \right] \geq 1 - \frac{2}{n - m + 1} = \frac{n - m - 1}{n - m + 1}.$$
Applying these lower bounds to the iterated conditional probabilities yields the result:

$$\Pr \left[ \bigcap_{j=1}^{n-2} E_j \right] \geq \prod_{m=1}^{n-2} \left( \frac{n - m - 1}{n - m + 1} \right)$$

In other words, we have

$$\Pr \left[ \bigcap_{j=1}^{n-2} E_j \right] \geq \left( \frac{n - 2}{n} \right) \left( \frac{n - 3}{n - 1} \right) \left( \frac{n - 4}{n - 2} \right) \cdots \left( \frac{3}{5} \right) \left( \frac{2}{4} \right) \left( \frac{1}{3} \right)$$

$$= \frac{2}{n(n - 1)} = \binom{n}{2}^{-1}$$
Applying these lower bounds to the iterated conditional probabilities yields the result:

\[
\Pr \left[ \bigcap_{j=1}^{n-2} E_j \right] \geq \prod_{m=1}^{n-2} \left( \frac{n - m - 1}{n - m + 1} \right)
\]

In other words, we have

\[
\Pr \left[ \bigcap_{j=1}^{n-2} E_j \right] \geq \frac{2}{n(n-1)} = \binom{n}{2}^{-1}
\]
In conclusion, we have shown that the contraction algorithm yields the correct answer with probability at least $\Omega(1/n^2)$. 
Number of Repetitions
Useful Inequality

\[ 1 + x \leq e^x. \]

Indeed, consider the function \( f(x) = e^x - 1 - x. \)

It has the derivative \( f'(x) = e^x - 1. \)

We have \( f'(x) = 0 \) if and only if \( x = 0. \)

The function \( f(x) \) has a (global) minimum at \( x = 0, \) since \( f''(0) = e^0 = 1 > 0. \) We can conclude that \( f(x) \geq f(0) = 0. \)
Consequence

Corollary

Consequently, for all positive integers $n$, we have

$$\left(1 + \frac{x}{n}\right)^n \leq \left(e^{x/n}\right)^n = e^x.$$
The probability that the algorithm fails to produce the correct result in one execution is

$$\Pr[\text{failure}] \leq (1 - 2/n^2).$$

Recall that for independent event $E$ and $F$, the probability is given by $\Pr[E \cap F] = \Pr[E] \Pr[F]$. Therefore, if we execute the algorithm $n^2/2$ times, then the probability that the repeated executions will never reveal the correct size of the minimum cut is at most

$$\left(1 - \frac{2}{n^2}\right)^{n^2/2} \leq e^{-1}.$$
Analysis

If we repeat the algorithm \( \frac{n^2 \ln n}{2} \) times, then the probability of obtaining an incorrect size of the minimum cut is at most

\[
\left(1 - \frac{2}{n^2}\right)^{\frac{n^2 \ln n}{2}} \leq e^{-\ln n} = \frac{1}{n}.
\]

We can conclude that repeating the contraction algorithm \( O(n^2 \log n) \) times yields the correct size of the minimum cut with high probability.
Assuming that the input multigraph has just a single minimum cut $C$, then Karger’s minimum cut algorithm fails in a single run if and only if it contracts an edge of the minimum cut $C$.

Selecting an edge crossing the cut $C$ is more likely towards a later stage of the algorithm rather than at the beginning.
We can subdivide the node contractions into different phases.

- Early phases need fewer repetitions (restarts), since they are less likely to err.
- Later phases need more repetitions (restarts), since this is where the errors are likely to happen.
- We use recursion.
**Contract**($G$, $t$)

**Require:** A connected loopfree multigraph $G = (V, E)$ with at least $t$ vertices.

1. 
   while $|V| > t$ do
2.   Select $e \in E$ uniformly at random;
3.   $G := G/e$;
4.   end while

**Idea**

Stop when $t$ nodes are reached.
As in the case $t = 2$, we have

$$\Pr \left[ \bigcap_{j=1}^{n-t} E_j \right] \geq \prod_{j=1}^{n-t} \left( \frac{n - m - 1}{n - m + 1} \right)$$

We get the lower bound

$$\Pr \left[ \bigcap_{j=1}^{n-t} E_j \right] \geq \left( \frac{n - 2}{n} \right) \left( \frac{n - 3}{n - 1} \right) \ldots \left( \frac{t}{t + 2} \right) \left( \frac{t - 1}{t + 1} \right) = \frac{\binom{t}{2}}{\binom{n}{2}}$$
Proposition

If \( t \geq \frac{n}{\sqrt{2}} + 1 \), then \( \binom{t}{2}/\binom{n}{2} \geq 1/2 \).

The function

\[
\binom{t}{2}/\binom{n}{2} = \frac{t(t - 1)}{n(n - 1)}
\]

is increasing in \( t \). Substituting \( t = \frac{n}{\sqrt{2}} + 1 \) yields

\[
\frac{t(t - 1)}{n(n - 1)} = \frac{\left(\frac{n}{\sqrt{2}} + 1\right)\frac{n}{\sqrt{2}}}{n(n - 1)} \geq \frac{n^2 - n}{2n(n - 1)} = \frac{1}{2}
\]
FastCut\((G)\)

**Require:** A connected loopfree multigraph \(G = (V, E)\).

1. \(n = |V|\).
2. \textbf{return} \(\text{mincut}(G)\) \textbf{if} \(n \leq 6\) \textbf{//} brute force computation
3. \(t = \lceil n/\sqrt{2} + 1 \rceil\).
4. \(G_1 = \text{Contract}(G, t)\).
5. \(G_2 = \text{Contract}(G, t)\).
6. \textbf{return} \(\min(\text{FastCut}(G_1), \text{FastCut}(G_2))\);
Complexity

Proposition

*FastCut runs in* $O(n^2 \log n)$ *time.*

Proof.

The algorithm Contract uses $O(n^2)$ time to reduce a multigraph with $n$ vertices down to 2 vertices. Thus, reducing it twice to $t$ vertices can certainly be done in $O(n^2)$ time. The time $T(n)$ of FastCut satisfies the recurrence

$$T(n) = 2T \left( \left\lfloor 1 + \frac{n}{\sqrt{2}} \right\rfloor \right) + O(n^2).$$

The solution to this recurrence satisfies $T(n) = O(n^2 \log n)$.
Proposition

*FastCut finds a minimum cut with probability $\Omega(1/\log n)$.*

Proof.

We already showed that minimum cut $C$ survives the contractions from $n$ to $n/\sqrt{2} + 1$ vertices with probability $1/2$ or more. Let $P(n)$ denote the probability that FastCut succeeds in finding a minimum cut in a multigraph with $n$ vertices. Then

$$P(n) \geq 1 - \left(1 - \frac{1}{2}P\left(\left\lfloor 1 + \frac{n}{\sqrt{2}} \right\rfloor\right)\right)^2.$$
Proof (Continued).

We will solve this recurrence by making a change of variables. The depth of the recursion is \( k = O(\log n) \). Let \( p(k) \) denote a lower bound on the success probability at level \( k \). Then

\[
p(0) = 1
\]

and (from the previous inequality)

\[
p(k + 1) = p(k) - \frac{p(k)^2}{4}.
\]
Success Probability

Proof (Continued).

We can solve this by setting $q(k) = 4/p(k) - 1$, which amounts to

$$p(k) = 4/(q(k) + 1).$$

Substituting into the previous equation yields

$$q(k + 1) = q(k) + 1 + \frac{1}{q(k)}.$$

By induction, we have

$$k < q(k) < k + H_{k-1} + 4,$$

where $H_{k-1} = 1 + 1/2 + \cdots + 1/(k - 1)$. It follows that

$$q(k) = k + \Theta(\log k).$$
**Proof (Continued).**

Since \( q(k) = k + \Theta(\log k) \) and by definition

\[
p(k) = \frac{4}{q(k) + 1} = \frac{4}{k + \Theta(\log k) + 1}.
\]

We have

\[
\lim_{k \to \infty} p(k)/(1/k) = \lim_{k \to \infty} \frac{4k}{k + \Theta(\log k) + 1} = 4.
\]

It follows that \( p(k) = \Theta(1/k) \). Since \( p(k) \) was the lower bound to \( P(n) \) with recursion depth \( k = \Theta(\log n) \), we can conclude that

\[
P(n) \geq p(\log n) = \Omega(1/\log n).
\]