The Fast-Cut Algorithm

1 Analysis of Basic Algorithm

The basic version of the randomized minimum cut algorithm has the following property: for any minimum cut $(S, \bar{S})$, it returns $(S, \bar{S})$ with probability at least $1/\binom{n}{2}$.

Lemma 1. *This implies there are at most $\binom{n}{2}$ distinct minimum cuts.*

Proof. Let $\mathcal{E}_S$ be the event that the algorithm returns $(S, \bar{S})$. The $\mathcal{E}_S$ are disjoint because we only return one cut at a time. Thus $\sum_S \Pr[\mathcal{E}_S] \leq 1$. For each minimum cut $(S, \bar{S})$, $\Pr[\mathcal{E}_S] \geq 1/\binom{n}{2}$. If there were more than $\binom{n}{2}$ minimum cuts, then $\sum_S \Pr[\mathcal{E}_S] > 1$, a contradiction. $\square$

We'd like to improve the success probability of the algorithm. To do so, we run the algorithm k times with independent coin flips and keep the minimum of the observed cuts. The failure probability for a single run is $1 - 1/\binom{n}{2}$. For k runs, we have $\left(1 - 1/\binom{n}{2}\right)^k$ due to independence. Then the success probability $\mathcal{S}$ over all runs is

$$\begin{aligned}
\mathcal{S} &\geq 1 - \left(1 - \frac{1}{\binom{n}{2}}\right)^k \\
&\geq 1 - \left(1 - \frac{2}{n^2}\right)^k \\
&\geq 1 - e^{-\frac{2k}{n^2}}.
\end{aligned}$$

Choose k to cancel the denominator and give an extra log term: $k = \alpha n^2 \log n$. Thus

$$\begin{aligned}
\mathcal{S} &\geq 1 - e^{\frac{-2\alpha n^2 \log n}{n^2}} \\
&= 1 - n^{-2\alpha}.
\end{aligned}$$

By making α large, we get any polynomially small failure probability we desire. Thus $k = \Theta(n^2 \log n)$ iterations is sufficient.

2 Improving the Algorithm

Recall that a failure in iteration i happens with probability at most $\frac{2}{n+1-i}$. This implies failures are much more likely in late iterations (from $\Theta(1/n)$ at the beginning to constant at the end). Intuitively, we can save time by redoing mostly the later iterations.

To more efficiently reuse early executions, use the following outline:

1. Create two copies H_1, H_2 of G.
2. Run contractions independently on both H_1, H_2 for a "while".
3. Recurse on H_1, H_2 independently.

Here, we choose a “while” to mean until right before the probability of the combined contractions hits 1/2 (for one run, say H_1). At what target size $t(n)$ does the (combined) failure probability hit 1/2?

To answer this question, we find a lower bound on the success probability of contractions going from size n to size t. Recall the success probability is

$$\begin{aligned}\prod_{i=1}^{n-1-t} \frac{n-1-i}{n+1-i} &= \frac{t(t-1)}{n(n-1)} \\ &\approx \frac{1}{2}.\end{aligned}$$

Solving for $t(n)$:

$$\begin{aligned}t^2 &= \frac{n^2}{2} \\ t &\approx \frac{n}{\sqrt{2}}.\end{aligned}$$

To be precise, $t(n) = \lceil 1 + n/\sqrt{2} \rceil$ guarantees success probability at least 1/2. To have $t(n) < n$, we need $n > 6$; for $n \leq 6$ we use exhaustive search.

Algorithm 1 Fast-Cut(G)

if $n \leq 6$ **then**
  Run exhaustive search to find a minimum cut.
else
  Let H_1, H_2 be two copies of G.
  Independently randomly contract H_1, H_2 down to size

$$t = \lceil 1 + \frac{n}{\sqrt{2}} \rceil.$$

  Return the smaller of Fast-Cut(H_1) and Fast-Cut(H_2).
end if

3 Runtime Analysis

We know exhaustive search is $\mathcal{O}(1)$ and contractions are $\mathcal{O}(n^2)$ for each of H_1, H_2. Thus we have the recurrence:

$$\begin{aligned}T(n) &= \mathcal{O}(n^2) + 2T(n/\sqrt{2}) \\ T(n) &= \Theta(1) \text{ for } n \leq 6\end{aligned}$$

We have $\log_{\sqrt{2}} n$ levels of recursion. On level i, we have 2^i instances of size

$$\frac{n}{\sqrt{2}^i}$$

each, doing work

$$\mathcal{O}\Big((\frac{n}{\sqrt{2}^i})^2 \Big) = \mathcal{O}(\frac{n^2}{2^i}).$$

Thus the total work on level i is

$$2^i \cdot \mathcal{O}(\frac{n^2}{2^i}) = \mathcal{O}(n^2).$$

This gives $T(n) = \mathcal{O}(n^2 \log n)$. We can also use the Master Theorem to see the same result.

4 Success Probability Analysis

Let $P(n)$ be a lower bound on the success probability of the algorithm for graphs of size n. Note that $1/2$ is the probability that the initial contractions were successful and $P(\lceil 1+n/\sqrt{2}\rceil)$ is the probability that the recursive call succeeds. We have two independent chances represented by H_1, H_2 and we only need one of them to succeed. So we obtain:

$$\begin{aligned}
P(n) &= 1 \text{ for } n \leq 6. \\
P(n) &\geq 1 - \big(1 - \frac{1}{2}P(\lceil 1 + \frac{n}{\sqrt{2}}\rceil)\big)^2 \\
&\approx 1 - \big(1 - \frac{1}{2}P(\frac{n}{\sqrt{2}})\big)^2 \\
&= P(\frac{n}{\sqrt{2}}) - \frac{1}{4}\big(P(\frac{n}{\sqrt{2}})\big)^2.
\end{aligned}$$

Let $n = \sqrt{2}^k$, then

$$P(\sqrt{2}^k) \geq P(\sqrt{2}^{k-1}) - \frac{1}{4}\big(P\sqrt{2}^{k-1})\big)^2$$

Let $p(k) = P(\sqrt{2}^k)$ to obtain

$$p(k) \geq p(k-1) - \frac{1}{4}(p(k-1))^2$$

Then, let $q(k) = 4/p(k) - 1$, so $p(k) = 4/(q(k)+1)$. Then,

$$\begin{aligned}
\frac{4}{q(k)+1} &\geq \frac{4}{q(k-1)+1} - \frac{1}{4}\big(\frac{4}{q(k-1)+1}\big)^2 \\
&= \frac{4}{q(k-1)+1} - \frac{4}{(q(k-1)+1)^2} \\
&= 4\frac{q(k-1)}{(q(k-1)+1)^2}.
\end{aligned}$$

We cancel the 4s and flip the numerator and denominator to obtain

$$\begin{aligned}
q(k)+1 &\leq q(k-1) + 2 + \frac{1}{q(k-1)} \\
q(k) &\leq q(k-1) + 1 + \frac{1}{q(k-1)}.
\end{aligned}$$

What do we know about q? For a base case, $q(0) = 3$ because $p(0) = 1$. Also, q is increasing, so $q(k) \geq 3$ $\forall k$. The right-hand-side is between $q(k-1)+1$ and $q(k-1)+2$, which implies $q(k) \leq 2k$. Thus

$$\begin{aligned}
p(k) &= \frac{4}{q(k)+1} \\
&= \Omega(1/k).
\end{aligned}$$

Because $k = \log_{\sqrt{2}} n$,

$$P(n) = \Omega(1/\log n).$$

Finally, we have that Fast-Cut succeeds with probability at least $\Omega(1/\log n)$. By running it $\Omega(\alpha \log^2 n)$ times with independent randomness, we can improve the success probability to $1 - n^{-\alpha}$. Each takes time $\mathcal{O}(n^2 \log n)$, so the total running time is $\mathcal{O}(n^2 \log^3 n)$.

Notice that we did not prove that there are only $\mathcal{O}(\log n)$ minimum cuts, because tie-breaking rules may make some cuts much less likely than others to be returned.