29 Name of Lecture

- Global minimum cut

29.1 Global Minimum Cut

A **cut** in an undirected graph $G = (V, E)$ is a bipartition of the vertices ($V = A \cup B$, $A \cap B = \varnothing$).

We previously discussed s-t cuts in flow networks. What if we don't specify s and t, but only require that both parts of the cut are nonempty?

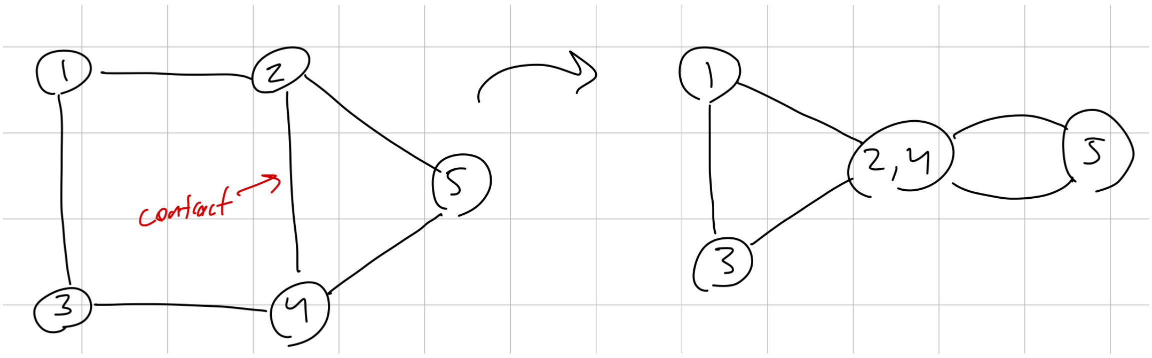
The **global minimum cut**, the nontrivial cut that has the smallest number of edges between A and B, measures the "robustness" of the graph.

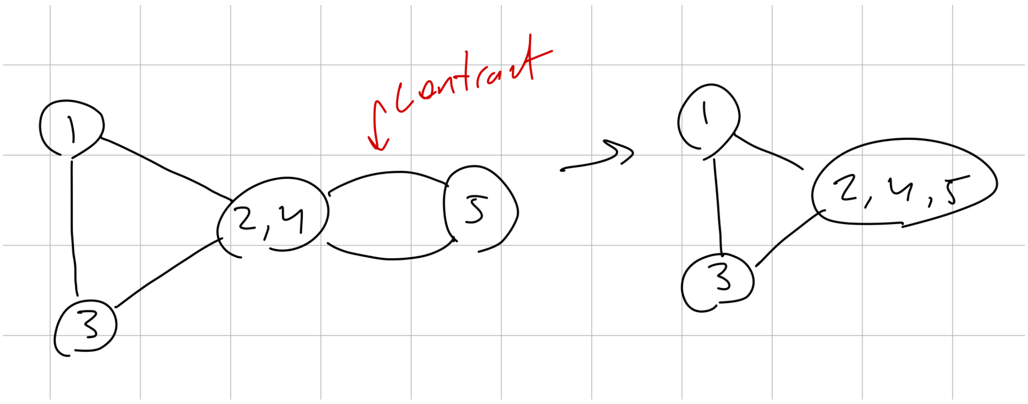
We can solve this using network flow. Fix some vertex s. In every cut, s must be on some side.
Since we don't know a t that must be in the other part of the global minimum cut, we run over all vertices $t \neq s$ and use Ford-Fulkerson algorithm to find the minimum s-t cut. The smallest of these is the global minimum cut.

This uses the Ford-Fulkerson algorithm $n - 1$ times. Ford-Fulkerson has cost $O(C(m + n)) = O(n^3)$, so the overall procedure is $O(n^4)$.

Alternative: Karger's contraction algorithm.

- Choose an edge uniformly at random
- Contract that edge, producing a multigraph (i.e., we allow multiple edges)
- Repeat until only 2 vertices remain
- Return the cut defined by the set of original vertices that led to the two final vertices





Why should this produce the minimum cut?

Lemma 29.1
The contraction algorithm returns a global minimum cut with probability at least $\frac{1}{\binom{n}{2}}$.

Proof. Suppose the minimum cut has size k. Then every vertex v has degree at least k, since otherwise $\{v\}, V \setminus \{v\}$ would be a smaller cut.
Thus $|E| \geq \frac{1}{2}kn$.
So the probability that a uniformly random edge belongs to the minimum cut is at most $\frac{k}{\frac{1}{2}kn} = \frac{2}{n}$.

Similarly, after j iterations, we have $n - j$ vertices.
Assuming we haven't contracted a minimum cut edge, te graph still has a minimum cut of size at least k, so it has at least $\frac{1}{2}k(n-j)$ edges, so a random edge blongs to the minimum cut with probabiliby at most $\frac{k}{\frac{1}{2}k(n-j)} = \frac{2}{n-j}$.
Let E_j be the even thtat an edge of the minimum cut is not contracted in the Jth step.

$$\begin{aligned}
\Pr(\text{success}) &= \Pr(E_1)\Pr(E_2|E_1)\Pr(E_3|E_1 \cap E_2)\cdots\Pr(E_{n-2}|E_1 \cap \cdots) \\
&\geq \left(1 - \frac{2}{n}\right)\left(1 - \frac{2}{n-1}\right)\left(1 - \frac{2}{n-2}\right)\cdots\left(1 - \frac{2}{3}\right) \\
&= \frac{n-2}{n} \cdot \frac{n-3}{n-1} \cdot \frac{n-4}{n-2} \cdots \frac{1}{3} \\
&= \frac{1}{\binom{n}{2}}
\end{aligned}$$

□

The success probability is small, but we can do better by repeating this many times and outputting the smallest cut we find.
Suppose the success probability in one run is ϵ. The probability we fail in all of k trials is

$$(1-\epsilon)^k \leq e^{-\epsilon k} \qquad \text{since } 1 - e \leq e^{-\epsilon}$$

We want failure probability $\leq \delta$, so $k = \frac{1}{\epsilon}\ln\frac{1}{\delta}$ suffices.
So we can find the minimum cut with probability arbitrarily close to 1 with $O(n^2)$ repetitions.

Each iteration takes time $O(m)$, so overall the algorithm has cost $O(m \cdot n^2)$ (Can improve to $O(n^2)$ with a bit more work).