15 Global Mincut

Given an undirected, capacitated graph $G = (V, E, c)$ find a partition of $V$ into two non-empty sets $S, V \setminus S$ s.t. the capacity of edges between both sets is minimized.

We can solve this problem using standard maxflow/mincut.

- Construct a directed graph $G' = (V, E')$ that has edges $(u, v)$ and $(v, u)$ for every edge $\{u, v\} \in E$.
- Fix an arbitrary node $s \in V$ as source. Compute a minimum $s$-$t$ cut for all possible choices $t \in V, t \neq s$. (Time: $O(n^4)$)
- Let $(S, V \setminus S)$ be a minimum global mincut. The above algorithm will output a cut of capacity $\text{cap}(S, V \setminus S)$ whenever $|\{s, t\} \cap S| = 1$.

Edge Contractions

- Given a graph $G = (V, E)$ and an edge $e = \{u, v\}$.
- The graph $G/e$ is obtained by “identifying” $u$ and $v$ to form a new node.
- Resulting parallel edges are replaced by a single edge, whose capacity equals the sum of capacities of the parallel edges.

Example 1

- Edge-contractions do no decrease the size of the mincut.
Randomized MinCut Algorithm

Algorithm 23 KargerMincut(G = (V, E, c))
1: for i = 1 → n − 2 do
2: choose e ∈ E randomly with probability c(e)/c(E)
3: G ← G/e
4: return only cut in G

▶ Let G_1 denote the graph after the (n − t)-th iteration, when t nodes are left.
▶ Note that the final graph G_2 only contains a single edge.
▶ The cut in G_2 corresponds to a cut in the original graph G with the same capacity.
▶ What is the probability that this algorithm returns a mincut?

Example: Randomized MinCut Algorithm

Animation only available in the lecture version of the slides.

Analysis

What is the probability that a given mincut A is still possible after round i?
▶ It is still possible to obtain cut A in the end if so far no edge in (A, V \ A) has been contracted.

Analysis

What is the probability that we select an edge from A in iteration i?
▶ Let min = cap(A, V \ A) denote the capacity of a mincut.
▶ Let cap(v) be capacity of edges incident to vertex v ∈ V_{n−i+1}.
▶ Clearly, cap(v) ≥ min.
▶ Summing cap(v) over all edges gives

\[ 2c(E) = 2 \sum_{e \in E} c(e) = \sum_{v \in V} \text{cap}(v) \geq (n − i + 1) \cdot \text{min} \]

▶ Hence, the probability of choosing an edge from the cut is

\[ \frac{\text{min}}{c(E)} \leq \frac{2}{n − i + 1} \]

\[ n − i + 1 \text{ is the number of nodes in graph } G_{n−i+1} = (V_{n−i+1}, E_{n−i+1}), \text{ the graph at the start of iteration } i \]
Analysis

The probability that we do not choose an edge from the cut in iteration $i$ is

$$1 - \frac{2}{n-i+1} = \frac{n-i-1}{n-i+1}.$$  

The probability that the cut is alive after iteration $n-t$ (after which $t$ nodes are left) is

$$\prod_{i=1}^{n-t} \frac{n-i-1}{n-i+1} = \frac{t(t-1)}{n(n-1)}.$$  

Choosing $t = 2$ gives that with probability $1/\binom{n}{2}$ the algorithm computes a mincut.

Improved Algorithm

**Algorithm 24** RecursiveMincut$(G = (V, E, c))$

1. for $i = 1 \rightarrow n - n/\sqrt{2}$ do
2. choose $e \in E$ randomly with probability $c(e)/c(E)$
3. $G \leftarrow G/e$
4. if $|V| = 2$ return cut-value;
5. $cuta \leftarrow$ RecursiveMincut$(G)$;
6. $cutb \leftarrow$ RecursiveMincut$(G)$;
7. return $\min\{cuta, cutb\}$

Running time:

- $T(n) = 2T\left(\frac{n}{\sqrt{2}}\right) + O(n^2)$
- This gives $T(n) = O(n^2 \log n)$.

Probability of Success

Repeating the algorithm $c \ln n \binom{n}{2}$ times gives that the probability that we are never successful is

$$\left(1 - \frac{1}{\binom{n}{2}}\right)^{c \ln n} \leq \left(e^{-1/\binom{n}{2}}\right)^{c \ln n} \leq n^{-c},$$

where we used $1 - x \leq e^{-x}$.

**Theorem 2**

The randomized mincut algorithm computes an optimal cut with high probability. The total running time is $O(n^4 \log n)$.

Note that the above implementation only works for very special values of $n$. 

$\Box$
The probability of contracting an edge of the mincut during these iterations is only $\frac{1}{2}$.

We can estimate the success probability by using the following game on the recursion tree. Delete every edge with probability $\frac{1}{2}$. If in the end you have a path from the root to at least one leaf node you are successful.

**Lemma 3**
The probability that an edge $e$ is alive is at least $\frac{1}{h(e) + 1}$.

**Proof.**
- An edge $e$ with $h(e) = 1$ is alive if and only if it is not deleted. Hence, it is alive with probability at least $\frac{1}{2}$.
- Let $p_d$ be the probability that an edge $e$ with $h(e) = d$ is alive. For $d > 1$ this happens for edge $e = \{c, p\}$ if it is not deleted and if one of the child-edges connecting to $c$ is alive.
- This happens with probability
  
  $$p_d = \frac{1}{2} \left( 2p_{d-1} - p_{d-1}^2 \right)$$

  Note that $x - x^2/2$ is monotonically increasing for $x \in [0, 1]$.
  
  $$x - x^2/2 \geq \frac{1}{d - 2d^2} \geq \frac{1}{d} - \frac{1}{d(d+1)} = \frac{1}{d+1}.$$