Karger's Randomized Global Minimum Cut - A Study

Surya Raghav B

June 2024

1 The Problem

cut : For an undirected graph $G = (V, E)$ is a partition of V into two non-empty sets A and B.
size of $cut(A, B)$: Number of edges with one end in A and the other in B.
Goal is to find smallest number of edges whose deletion disconnects the graph.

Theorem 1.1 *There is a polynomial-time algorithm to find a global min-cut in an undirected graph G.*

Proof. Given an undirected graph $G = (V, E)$, we transform it so that we replace every edge $e = (u, v) \in E$ with two oppositely oriented directed edges, $e' = (u, v)$ and $e'' = (v, u)$, each with capacity 1. Let G' be the transformed graph.

Fix any $s \in V$ as source and compute min-cut in G' for every other $t \in V - \{s\}$. The best among these n-1 directed min-cut computations will be a global min-cut of G.

Now we look at David Karger's Contraction Algorithm, a randomized method.

2 Designing the Contraction Algorithm

We have a multigraph $G = (V, E)$. The algorithm begins by uniform randomly choosing an edge $e = (u, v)$ of G and contracts it. In the new graph G' the nodes u, v are combined into a *supernode* $\{u, v\}$; all the edges between u,v are deleted; and the ends of others edges are updated accordingly.

This contraction process is done recursively until the graph just contains two supernodes v_1 and v_2. Each of these supernodes v_i has a subset $S(v_i) \subseteq V$. Then the cut $(S(v_1), S(v_2))$ is given as the output.

Algorithm 1 Contraction Algorithm

We have a multigraph $G = (V, E)$
Each node is represented by a set $S(v)$
Initially $S(v) = \{v\}$ for each v
if G has only two nodes **then**
  return the cut $(S(v_1), S(v_2))$
else
  choose an edge $e = (u, v)$ of G uniformly at random
  Contract by removing all the edges between u, v and replace the nodes u and v with new node Z_{uv}
  Define $S(z_{uv}) = S(u) \cup S(v)$
  Let G' be the contracted graph of G
  Apply the Contraction Algorithm recursively to G'
end if

3 Analyzing the Algorithm

Theorem 3.1 *The Contraction algorithm returns a global min-cut with probability at least* $\frac{1}{\binom{n}{2}}$

Proof. Assume that the global min-cut(A,B) of G has a size k and the set F of k edges that disconnect the graph into A and B. We look at the chances that an edge in F is contracted, which fails to return the cut (A, B).

We want upper bound on the probability that an edge in F is contracted. Note : If any node had degree less than k, then the cut $(\{v\}, V - \{v\})$ would have size less than, contradicting our assumption.

$$\sum_{v \in V} deg(v) = 2|E|$$

$$\sum_{v \in V} deg(v) \geq kn$$

$$|E| \geq \frac{1}{2}kn$$

The Probability that an edge in F is contracted is atmost

$$\frac{k}{\frac{1}{2}kn} = \frac{2}{n}$$

After j iterations, j edges are contracted, so n-j supernodes are in the graph G'. Since every cut of G' is a cut of G and the min-cut is k, the degree of every supernode in G' should be atleast k and the the number of edges should be atleast $\frac{1}{2}k(n-j)$. Then the probability that and edge of F is contracted in the next iteration j+1 is atmost

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

.

The cut (A, B) is found if not edge of F is contracted after n-2 iterations. Let ϵ_j be the event that an edge of F is not contracted in iteration j. We have $Pr[\epsilon_1 \geq 1 - 2/n]$ and $Pr[\epsilon_{j+1} | \epsilon_1 \cap \epsilon_2 \cap \dots \epsilon_j] \geq 1 - 2/(n-j)$. We need to find the lower bound of $Pr[\epsilon_1 \cap \epsilon_2 \cap \dots \epsilon_{n-2}]$, which we will unwind using conditional probability.

$$
\begin{aligned}
&Pr[\epsilon_1].Pr[\epsilon_2|\epsilon_1] \dots Pr[\epsilon_{j+1}|\epsilon_1 \cap \epsilon_2 \dots \cap \epsilon_j] \dots Pr[\epsilon_{n-2}|\epsilon_1 \cap \epsilon_2 \dots \cap \epsilon_{n-3}] \\
&\geq \left(1 - \frac{2}{n}\right)\left(1 - \frac{2}{n-1}\right) \dots \left(1 - \frac{2}{n-j}\right) \dots \left(1 - \frac{2}{3}\right) \\
&= \left(\frac{n-2}{n}\right)\left(\frac{n-3}{n-1}\right)\left(\frac{n-4}{n-2}\right) \dots \left(\frac{2}{4}\right)\left(\frac{1}{3}\right) \\
&= \frac{2}{n(n-1)} = \binom{n}{2}^{-1}.
\end{aligned}
$$

So the probability of the algorithm failing is atmost $(1 - 1/\binom{n}{2})$. If we run the algorithm $\binom{n}{2}$ times, then the probability of failing is atmost

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

4 Number of Global Minimum Cuts

For a directed flow network with nodes $s, t, v_1, v_2, \dots, v_n$, with each v_i connected to both s and t, The set of s with any subset of $\{v_1, v_2, \dots, v_n\}$ will be a min-cut. Thus for directed flow network the maximum number of min-cuts are 2^n.

For a undirected graph n-node cycles have the maximum number of global min-cuts of $\binom{n}{2}$. Following this we show that cycles are an extreme case.

Theorem 4.1 *An undirected graph $G = (V, E)$ on n nodes has at most $\binom{n}{2}$ global min-cuts.*

Proof. Let G be a graph, having r global min-cuts denoted by $C_1, \dots, C_r$. Let ϵ_i be the event that C_i is returned by the Contraction Algorithm, and let $\epsilon = \cup_{i=1}^{r} \epsilon_i$ denote the event that the algorithm returns any global min-cut.

From (3.1) we have $Pr[\epsilon_i \geq 1/\binom{n}{2}]$. Since each event pairs ϵ_i and ϵ_j are disjoint we have,

$$Pr[\epsilon] = Pr[\cup_{i=1}^{r} \epsilon_i] = \sum_{i=1}^{r} Pr[\epsilon_i] \geq r/\binom{n}{2}.$$

Since $Pr[\epsilon] < 1$, we must have $r < \binom{n}{2}$.