Karger's Algorithm for Global Min-Cut

January 31, 2017

1 Problem Definition

The input is an undirected graph $G = (V, E)$. There are no edge capacities, but parallel edges are allowed. A cut in G is a partition (A, B) of V, such that $A, B \neq \emptyset$. The cut value is $|E(A, B)|$ - the total number of edges crossing the cut. The goal is to compute a cut of minimum value. Note: this can be solved by n^2 computations of minimum s-t cut, where every time we choose a different pair of vertices of G to serve as s and t.

2 The Algorithm

The algorithm performs $(n-2)$ iterations. In each iteration i, we have some graph G_i, obtained from G by contracting some subset of its edges. We start with $G_1 = G$. In order to execute the ith iteration, we choose an edge (u, v) of G_i uniformly at random and contract it, to obtain the graph G_{i+1}. In order to contract edge (u, v), we replace both vertices with a single vertex $w_{u,v}$. For each edge of G_i that is incident on either u or v, we add a corresponding edge to G_{i+1}, that is incident on the new vertex $w_{u,v}$. We may create parallel edges that we keep in the graph, but we delete self-loops.

Note that for each i, every vertex v of G_i corresponds to some subset S_v of vertices on G, that were contracted over the course of the algorithm into v (it is possible that $S_v = \{v\}$, if we did not contract any edges incident to v yet). Moreover, the subsets $\{S_v \mid v \in G_i\}$, define a partition of V.

The algorithm terminates after $(n-2)$ iterations. The corresponding graph G_{n-1} has exactly two vertices, u, v. We define the final cut to be (A, B), where $A = S_u$ (all vertices contracted into u), and $B = S_v$.

3 Analysis

Let C denote the value of the global minimum cut in G. We use the following two simple observations.

Observation 1 *The degree of every vertex in G is at least C.*

Proof: Assume otherwise, and let v be a vertex of degree less than C. Then the partition $(\{v\}, V \setminus \{v\})$ of V is a cut of value less than C, a contradiction. □

Observation 2 *For all i, the value of the minimum cut in G_i is at least C. In other words, cut values cannot decrease when we contract edges.*

Proof: The proof follows from the fact that for every partition (A', B') of vertices of G_i, we can find a partition (A, B) of vertices of G, such that the two cuts have exactly the same value. This is done by letting A be obtained from A', and B obtained from B', after we un-contract all contracted edges. $\square$

Corollary 3 *For all i, the degree of every vertex in G_i is at least C, and $|E(G_i)| \geq C(n-i+1)/2$.*

Proof: Consider some graph G_i. Since the value of the minimum cut in G_i is at least C, all vertices in G_i have degree at least C. The number of vertices in G_i is $n-i+1$: G_i was obtained after $i-1$ iterations, and in every iteration we decrease the number of vertices by 1. $\square$

Let us fix some minimum cut (A, B), and analyze the probability that this is the cut the algorithm outputs.

Theorem 4 *The probability that the algorithm outputs cut (A, B) is at least $1/\binom{n}{2}$.*

Proof: The algorithm outputs (A, B) if and only if no edge of $E(A, B)$ was contracted throughout the algorithm. For each $1 \leq i \leq n-2$, we say that cut (A, B) survives iteration i, iff no edge of $E(A, B)$ was contracted in the first i iterations of the algorithm. We let $\mathcal{E}_i$ be the event that cut (A, B) survived iteration i. We now analyze the probabilities of these events.

Event $\mathcal{E}_1$. Notice that G contains at least $nC/2$ edges, and only C of them belong to $E(A, B)$. The probability of choosing an edge of $E(A, B)$ is at most $\frac{C}{nC/2} = \frac{2}{n}$, and $\mathbf{Pr}\left[\mathcal{E}_1\right] \geq 1 - \frac{2}{n} = \frac{n-2}{n}$.

Event $\mathcal{E}_i$. We would now like to analyze $\mathbf{Pr}\left[\mathcal{E}_i \mid \mathcal{E}_{i-1}\right]$. Graph G_i contains at least $C(n-i+1)/2$ edges (see the corollary above), and $|E(A, B)| = C$. So the probability we choose an edge of $E(A, B)$ in iteration i is at most $\frac{C}{C(n-i+1)/2} = \frac{2}{n-i+1}$, and $\mathbf{Pr}\left[\mathcal{E}_i \mid \mathcal{E}_{i-1}\right] \geq 1 - \frac{2}{n-i+1} = \frac{n-i-1}{n-i+1}$.

Final accounting: The algorithm returns cut (A, B) iff $\mathcal{E}_{n-2}$ happens. We now bound $\mathcal{E}_{n-2}$. Using the chain rule in probability theory, and the fact that $\mathbf{Pr}\left[\mathcal{E}_i \mid \neg\mathcal{E}_{i-1}\right] = 0$ for all i, we get that:

$$\begin{aligned}
\mathbf{Pr}\left[\mathcal{E}_{n-2}\right] &= \mathbf{Pr}\left[\mathcal{E}_{n-2} \mid \mathcal{E}_{n-3}\right] \cdot \mathbf{Pr}\left[\mathcal{E}_{n-3} \mid \mathcal{E}_{n-4}\right] \cdots \mathbf{Pr}\left[\mathcal{E}_2 \mid \mathcal{E}_1\right] \cdot \mathbf{Pr}\left[\mathcal{E}_1\right] \\
&\geq \frac{n-2}{n} \cdot \frac{n-3}{n-1} \cdot \frac{n-4}{n-2} \cdots \frac{1}{3} \\
&= \frac{(n-2)!2!}{n!} = \frac{1}{\binom{n}{2}}
\end{aligned}$$

$\square$

Corollary 5 *The number of minimum cuts in an n-vertex graph G is at most $\binom{n}{2}$.*

Proof: Let $N = \binom{n}{2}$, and assume for contradiction that there are at least $N+1$ minimum cuts in G. Let $(A_1, B_1), \ldots, (A_{N+1}, B_{N+1})$ be any collection of $N+1$ different minimum cuts. For each $1 \leq j \leq N+1$, let $\tilde{\mathcal{E}}_j$ be the event that the algorithm returns cut (A_j, B_j). Then all events $\mathcal{E}_1, \ldots, \mathcal{E}_{N+1}$ are disjoint. Therefore, if we denote by p the probability that any minimum cut is returned, then we get: $p \geq \sum_{j=1}^{N+1} \mathbf{Pr}\left[\tilde{\mathcal{E}}_j\right] \geq \sum_{j=1}^{N+1} \frac{1}{\binom{n}{2}} > 1$, which is impossible. $\square$