Fall 2012

CS 174 : Lecture Notes
Min-Cut Algorithm

Let $G = (V, E)$ be an undirected multigraph, *i.e.* there may be multiple edges between any two vertices $u, v \in V$. However, there are no self loops in G.

Recall from the textbook, the procedure Edge-Contract(e) for some edge, $e = (u, v)$, from G. The procedure produces a resulting graph, $G' = (V', E')$, where two end-points u and v of the edge are merged into a single vertex, say $\star$, and any edge (w, u) or (w, v) in the original graph is replaced by the edge $(w, \star)$. Thus, the only edges in G that are not present in G' are the edges between u and v (there may be multiple because G is a multi-graph).

Consider the following recursive algorithm for min-cut. The parameter $\sqrt{2}$ in the algorithm is somewhat optimized, but essentially could be replaced by any *constant* α with slightly worse guarantees. We will not explicitly use $\lfloor n/\sqrt{2} \rfloor$, but just use $n/\sqrt{2}$ as if it were an integer. This does not change the essence of the proof, but simplifies the notation considerably.

RecursiveMinCut($G = (V, E)$)

1. Let $G_1 = (V_1, E_1)$ be the graph obtained by $n - (n/\sqrt{2})$ edge-contract operations on G.
   Let $C_1 =$ RecursiveMinCut(G_1)
2. Let $G_2 = (V_2, E_2)$ be the graph obtained by $n - (n/\sqrt{2})$ edge-contract operations on G (independent of G_1).
   Let $C_2 =$ RecursiveMinCut(G_2)
3. Return the cut among C_1 and C_2 that is of a smaller size.

First, we show that following:

Lemma 1. *The depth of recursion for RecursiveMinCut, starting with a graph $G = (V, E)$ is $2\log(|V|)$.*

Proof. Every depth of recursion reduces the number of vertices by a factor $1/\sqrt{2}$. Therefore, when the depth is $2\log|V|$, the number of vertices remaining is at most $|V|(1/\sqrt{2})^{2\log|V|} = 1$. Thus, the depth of recursion can at most be $2\log|V|$. $\square$

Let $C \subseteq E$ be a specific min-cut in the graph $G = (V, E)$. We show next that the probability that the cut C is not destroyed before the recursive call is made is at least $1/2$.

Lemma 2. *The probability that a specific min-cut C is not destroyed after $n - n/\sqrt{2}$ edge contract operations is at least*

$$\frac{\frac{n}{\sqrt{2}} \cdot \left(\frac{n}{\sqrt{2}} - 1\right)}{n \cdot (n-1)}$$

Proof. This is along the lines of proof of Theorem 1.8 in Chapter 1 of the textbook. However, instead of stopping when the graph has 2 vertices (in the textbook), we stop when the graph has $n/\sqrt{2}$ vertices. The argument is identical. $\square$

For any graph, $G = (V, E)$, let d_G be the depth of recursion required for this graph for the algorithm RecursiveMinCut. We know that $d_G \leq 2 \log |V|$. Let p_d be the minimum (over all possible graphs) probability that the recursive min-cut algorithm succeeds in finding a min-cut for all graphs, G, with $d_G \leq d$. We will show by induction that $p_d \geq 1/(d+1)$.

Observe that $p_0 = 1$, *i.e.* in the base case the algorithm always succeeds. Suppose it is the case that $p_{d-1} \geq 1/d$. Now consider an arbitrary graph which requires recursion depth d.

In the algorithm, RecursiveMinCut, let p be the probability that C_1 is a min-cut. The probability that C_2 is a min-cut is also exactly p (since the process is identical). However, the event C_1 being a min-cut and C_2 being a min-cut are independent. Thus, the probability that at least one of them is a min-cut is $2p - p^2$. It is easy to show that $2p - p^2$ is an increasing function of p. Now, note that $p \geq (1/2)p_{d-1}$, thus we have

$$\begin{aligned}
p_d &= 2p - p^2 \\
&\geq p_{d-1} - \frac{1}{4}p_{d-1}^2 \\
&\geq \frac{1}{d} - \frac{1}{4d^2} \\
&\geq \frac{1}{d+1}
\end{aligned}$$

Running Time

Thus, the probability that RecursiveMinCut succeeds for a graph $G = (V, E)$ is at least $1/(2 \log |V| + 1)$. Now, in order to guarantee that the algorithm succeeds with probability say $1 - 1/|V|^2$, it is sufficient to run RecursiveMinCut on graph $G = (V, E)$ independently $(2 \log |V| + 1)^2$ times (Why?).

The running time of a single-run of RecursiveMinCut satisfies the following recursion.

$$T(|V|) = 2|V|(|V| - (|V|/\sqrt{2})) + 2T(|V|/\sqrt{2})$$

Using the master theorem, this gives us a running time $T(|V|) = O(|V|^2 \log |V|)$. Thus, when the total running time (after running this algorithm $(2 \log |V| + 1)^2$ times independently) is $O(|V|^2 \log^3 |V|)$, which is much better than the $O(|V|^4 \log |V|)$ bound if we run the algorithm in the book repreatedly to get the same error rate, and is even better than the best deterministic runtime of $O(|V|^3)$!