COMPSCI 590.1: Graph Algorithms Feb 5, 2015

Lecture # 9

Lecturer: Debmalya Panigrahi *Scribe: Janardhan Kulkarni*

1 Overview

In previous lectures we saw a randomized contraction algorithm by Karger for computing (or finding) a global min-cut of a graph. In this lecture we will see a breakthrough result by Nagamochi and Ibaraki for computing the global min-cut. Nagamochi and Ibaraki's algorithm [2] is deterministic, and runs in $O(mn)$ time. We will start with a high level description of the algorithm and then see its full details.

2 Problem

Let $G = (V,E)$ be a graph. For a given pair of vertices $(s,t) \in V$, let $C(s,t)$ denote the minimum cut separating s and t. Min-cut of a graph is $\lambda(G) = \min_{s,t} C(s,t)$. Minimum cut of a graph (also called edge connectivity) can be calculated by running Max-Flow Min-Cut algorithm n times by taking each vertex as the source. In this lecture, we will see a much faster way to compute min-cut.

2.1 High Level Idea

Nagamochi-Ibaraki algorithm is based on the following beautiful observation. Suppose minimum cut for a vertex pair (u,v) of a graph is the *degree cut* of either vertex v or u. By degree-cut we mean the cut formed by taking the edges incident on vertex v. Then, a global min-cut can be found by using the following simple algorithm: Find a pair of vertices (u,v) such that minimum cut between $u-v$ is the degree cut of v. Add the degree of the vertex v, and the vertex to a list L. Now, contract the vertices (u,v) into a meta vertex uv and repeat the process. A global min cut for the graph then corresponds to the minimum degree cut in the list L. Note that elements of L may represent a subset of vertices because at every iteration we contract a pair of vertices. Correctness of the algorithm follows from the fact that either a minimum cut of a graph is equal to the minimum (u,v) cut, and if that is not the case, then the minimum cut of a graph remains unchanged by contracting the vertices (u,v).

The main idea of Nagamochi-Ibaraki algorithm is to find the pair of vertices (u,v) such that minimum cut separating (u,v) is simply a degree cut of either u or v as fast as possible. (Existence of such a pair of vertices was known earlier due to Mader in 1972.) Nagamochi-Ibaraki showed that this can be done in $O(m)$ time. There are several proofs of this result, and here we will use the proof as given in [1].

2.2 Legal Ordering

For a given subset of vertices X and Y, $X \subseteq V$, $Y \subseteq V$, and $X \cap Y = 0$, let $d(X,Y)$ denote the number of edges connecting X and Y. Let $d(u) := d(u, V \setminus u)$. For a given graph G, an ordering of vertices $v_1, v_2, \ldots, v_n$ is said to be *a legal ordering* if the following condition is satisfied:

$$d(v_i, \{v_1, v_2, \ldots v_{i-1}\}) \geq d(v_{i'}, \{v_1, v_2, \ldots v_{i-1}\}), \forall i' \in V \setminus \{v_1, v_2, \ldots v_{i-1}\} \quad (1)$$

In the other words, in a legal ordering the vertex at the position i has the most number of edges to the vertex set in the prefix $1, 2, \ldots i-1$. By using an appropriate data structure, one can show that legal ordering of a graph can be computed in time $O(m)$, where m is the number of vertices. We will use following simple observations about the legal ordering of a graph G.

Fact 1. *The legal ordering for the graph G remains unchanged if we delete the edge connecting v_n and v_{n-1}.*

Fact 2. *The legal ordering for the graph $G \setminus v_n$ is $v_1, v_2, \ldots v_{n-1}$.*

Fact 3. *The legal ordering for the graph $G \setminus v_{n-1}$ is $v_1, v_2, \ldots v_{n-2}, v_n$.*

Now we will show that given the legal ordering of graph G, then the minimum cut between (v_{n-1}, v_n) is equal to the degree cut of v_n.

Lemma 4. *For any graph G, and a legal odering of vertices $v_1, v_2, \ldots v_n$, $\;C(v_{n-1}, v_n) = d(v_n)$.*

Proof. First note that $C(v_{n-1}, v_n) \le d(v_n)$, since $d(v_n)$ is a valid cut separating v_n and v_{n-1}. Hence if we show that $C(v_{n-1}, v_n) \ge d(v_n)$, then the lemma is true. One way of proving the lemma is by contradiction. Assume that G is the minimal counter example graph. A graph is a minimal counter example graph, if there is no other graph G' either with less number of vertices or less number of edges but with same number of vertices as G. Clearly, number of vertices in such a counter example graph is at least 3. Furthermore, note that in such a graph G there cannot be an edge connecting v_{n-1} and v_n, since removing the edge we get a graph G' for which lemma is true. However, then the lemma would also be true for G.

Now consider the graphs G and the graph $G' := G \setminus v_n$ obtained by removing vertex v_n. The cardinality of minimum cut separating v_{n-1}, v_{n-2} is greater than the minimum cut separating v_{n-1}, v_{n-2} in G'. Therefore,

$$C(v_{n-1}, v_{n-2} : G) \ge C(v_{n-1}, v_{n-2} : G')$$

However, from the observation 2 and the assumption that lemma holds for G' we have,

$$C(v_{n-1}, v_{n-2} : G) \ge C(v_{n-1}, v_{n-2} : G') \ge d(v_{n-1}, V_{n-2})$$

Since there is no edge in between v_{n-1} and v_{n-1} and the legality of the ordering, we can extend the above inequalities as follows:

$$C(v_{n-1}, v_{n-2} : G) \ge C(v_{n-1}, v_{n-2} : G') \ge d(v_{n-1}, V_{n-2}) \ge d(v_n, V_{n-1}) = d(v_n) \tag{2}$$

Next, we consider the graphs G and $G'' := G \setminus v_n$. Since the lemma is also true for G'', we get the following inequalities using the same arguments as above.

$$C(v_n, v_{n-2} : G) \ge C(v_n, v_{n-2} : G'') \ge d(v_n, V_{n-2}) \ge d(v_n, V_{n-1}) = d(v_n) \tag{3}$$

Now, observe that

$$C(v_n, v_{n-1}) \ge \min\{C(v_n, v_{n-2}), C(v_{n-1}, v_{n-2})\}$$

The above inequality follows from the simple observation that if we consider the minimum cut seperating v_{n-1} and v_n, then the vertex v_{n-2} lies either in the component containing v_n or the component containing v_{n-2}. Therefore, from equations (2) and (3), we get

$$C(v_n, v_{n-1}) \ge \min\{C(v_n, v_{n-2}), C(v_{n-1}, v_{n-2})\} \ge d(v_n)$$

This contradicts our assumption that G is a minimal counter example graph for the lemma. This completes the proof. □

As discussed above, Nagamochi-Ibaraki algorithm uses the legal odering of a graph to find a min-cut of G as follows. Let G_i be the ith iteration of the algorithm. We compute a legal ordering of the vertices of graph G_i, and add the degree cut of vertex v_n^i to a list L. Here, v_n^i denotes the last vertex which appears in the legal ordering of graph G_i. Now, from the above lemma, if the minimum cut of G_i separates v_{n-1} and v_n, then degree of cut v_n^i is a minimum cut of G_i. Otherwise, minimum cut of G_i remains unchanged by contracting vertices v_{n-1}^i, v_n^i. Since global min cut of G should correspond to minimum cut of some G_i, by taking the minimum degree cut in the list L we obtain the global min cut of G.

2.3 Running time

Using an appropriate data structure one can find a legal ordering of G in $O(m)$ time. Since every iteration of Nagamochi-Ibaraki algorithm contracts a pair of vertices there can be at most n iterations in the algorithm. Therefore, the total running time of the algorithm is at most $O(mn)$.

References

[1] Andra's Frank. On the edge-connectivity algorithm of nagamochi and ibaraki.

[2] Hiroshi Nagamochi and Toshihide Ibaraki. A linear-time algorithm for finding a sparse k-connected spanning subgraph of a k-connected graph. *Algorithmica*, 7(5&6):583–596, 1992.