On the Edge-Connectivity Algorithm of Nagamochi and Ibaraki

András Frank*

Abstract

This note was written in 1994, while the author visited the J. Fourier University of Grenoble, but it has never been published. The present facsimile of the original note serves its more convenient accessibility.

*Egerváry Research Group of MTA-ELTE, Department of Operations Research, Eötvös University, Pázmány P. s. 1/c. Budapest, Hungary, H-1117. e-mail: frank@cs.elte.hu

ON THE EDGE-CONNECTIVITY ALGORITHM OF NAGAMOCHI AND IBARAKI

(A. Frank)

(March, 1994)

We are given an undirected (!) graph $G = (V, E)$ with n nodes and m edges in which parallel edges are allowed but loops not. Let $c(x, y) = c(x, y; G)$ denote the minimum cardinality of a cut separating x and y. The **edge-connectivity** $\lambda(G)$ of G is the minimum of the $c(x, y)$ values over all pairs of distinct nodes.

It is well-known that $\lambda(G)$ may be computed by $n - 1$ applications of the MFMC algorithm. H. Nagamochi and T. Ibaraki developed a revolutionary new algorithm to compute $\lambda(G)$ that does not use flows, paths, Menger's theorem. This note presents their algorithm along with a simplified proof of its validity.

For two disjoint subsets X, Y of nodes $d(X, Y)$ denotes the number of edges connecting X and Y. We use $d(X) := d(X, V - X)$.

For an ordering $v_1, \ldots, v_n$ of the nodes of G, V_i denotes the set of the first i elements. We say that an ordering is **legal** if

$$d(V_{i-1}, v_i) \geq d(V_{i-1}, v_j)$$

for every pair i, j $(2 \leq i < j \leq n)$. (With the help of an appropriate data structure, a legal ordering may be constructed in $O(m)$ time.)

One can easily observe that a legal ordering $v_1, \ldots, v_n$ stays legal if we delete the edges connecting v_n and v_{n-1}. Likewise, $v_1, \ldots, v_{n-1}$ is legal for $G' := G - v_n$ and $v_1, \ldots, v_{n-2}, v_n$ is legal for $G'' := G - v_{n-1}$. These facts will be used without any further reference.

LEMMA $c(v_n, v_{n-1}) = d(v_n, V_{n-1})$ $(= d(v_n))$.

Proof. Clearly, the left-hand side cannot be larger than the right-hand side. Hence we need only to prove the $\geq$ direction. Assume, indirectly, that the lemma is not true and G is a minimal counter-example. Then, obviously, $n \geq 3$.

We claim that $d(v_n) = d(v_n, V_{n-2})$ or, equivalently, that there is no edge connecting v_n and v_{n-1}. Indeed, by deleting such an edge we obtain a graph for which the lemma holds but the lemma then would hold for G, as well.

The lemma holds for $G - v_n$ and hence $c(v_{n-1}, v_{n-2}) \geq c(v_{n-1}, v_{n-2}; G - v_n) \geq d(v_{n-1}, V_{n-2}) \geq d(v_n, V_{n-2}) = d(v_n)$ where the last inequality follows from the legality of the given ordering.

The lemma holds for the graph $G - v_{n-1}$ and hence $c(v_n, v_{n-2}) \geq c(v_n, v_{n-2}; G - v_{n-1}) \geq d(v_n, V_{n-2}) = d(v_n)$.

Thus $c(v_n, v_{n-1}) \geq \min(c(v_n, v_{n-2}), c(v_{n-1}, v_{n-2})) \geq d(v_n)$ contradicting the assumption that G is a counter-example. ♠

(**REMARK.** The existence of two nodes x, y for which $c(x, y) = d(x)$ was proved by W. Mader as early as 1972.)

The algorithm of Nagamochi and Ibaraki is based on the following observation. Let x and y be two nodes for which $d(x) = c(x, y)$. If there is a minimum cut of G (i.e. a cut of $\lambda(G)$ elements) separating x and y, then $\lambda(G) = d(x)$ and the star of x is such a cut.

If no minimum cut of G separates x and y, then shrinking x and y into one node does not destroy any minimum cut. In this case it suffices to compute the edge-connectivity of the shrunken graph.

These considerations along with the lemma verifies the correctness of the algorithm.

Let $G_1 := G$ and iterate $n-1$ times the following procedure. Assume that graph G_i has already been constructed by successively shrinking $i-1$ pairs of nodes $(i = 2, \dots, n-1)$. Determine a legal ordering of the nodes of G_i and put in a list the last node x_i of this order along with the value $d(x_i)$. (Here the subscript refers to the graph G_i.) Let G_{i+1} be constructed from G_i by shrinking the last two nodes of the ordering.

When the $n-1$ iterations is completed, choose an element x_j of the list for which $d(x_j)$ is minimum. $\lambda(G)$ is equal to $d(x_j)$. The node x_j of G_j corresponds to a subset X_j of nodes of the original G and hence $d(X_j) = \lambda(G)$, that is, the cut $[X_j, V - X_j]$ is a minimum cut of G.

Since a legal ordering may be found in $O(m)$ time, the complexity of the whole algorithm is $O(mn)$.