1998年度日本オペレーションズ・リサーチ学会
秋季研究発表会

k-Edge and 3-Vertex Connectivity Augmentation in an Arbitrary Multigraph

02004044 Kyoto University * ISHII Toshimasa
01403794 Kyoto University NAGAMOCHI Hiroshi
01001374 Kyoto University IBARAKI Toshihide

1 Introduction

Let $G = (V, E)$ stand for an undirected multigraph with a set V of *vertices* and a set E of *edges*. The connectivity augmentation problem has been extensively studied as an important problem in the network design problem.

The local edge-connectivity $\lambda_G(x, y)$ for two vertices $x, y \in V$ is defined to be the minimum size of a cut in G that separates x and y (i.e., x and y belong to different sides of X and $V - X$), or equivalently the maximum number of edge-disjoint path between x and y by Menger's theorem [1]. The local vertex-connectivity $\kappa_G(x, y)$ for two vertices $x, y \in V$ is defined to be the number of internally-disjoint paths between x and y in G. For a given integer k, we call G *k-edge-connected* (resp., *k-vertex-connected*) if $\lambda_G(x, y) \geq k$ (resp., $\kappa_G(x, y) \geq k$) holds for every $x, y \in V$. Given a graph $G = (V, E)$ and an integer k, the *edge-connectivity augmentation problem*, (resp., the *vertex-connectivity augmentation problem*) asks to augment G by adding the smallest number of new edges so that the resulting graph G' becomes k-edge-connected (resp., k-vertex-connected). Recently, many efficient algorithms are developed for solving the edge-connectivity augmentation problem and the vertex-connectivity augmentation problem.

In this paper, we consider the problem of augmenting the edge-connectivity and the vertex-connectivity of a given graph G simultaneously by adding the smallest number of new edges. For two given integers k and ℓ, we say that G is *(k, ℓ)-connected* if G is k-edge-connected and ℓ-vertex-connected. Given a multigraph $G = (V, E)$, and two integers k, ℓ, the *edge- and vertex-connectivity augmentation problem*, denoted by EVAP(k, ℓ), asks to augment G by adding the smallest number of new edges to G so that the resulting graph G' becomes (k, ℓ)-connected. Recently, the authors presented algorithms EV-AUGMENT [2] and EV-AUGMENT3 [3]. The first algorithm solves EVAP$(k, 2)$ in $((nm + n^2 \log n) \log n)$ time, and the second solves EVAP$(k, 3)$ in $O(n^4)$ time under the assumption that k is a fixed constant and the input multigraph is 2-vertex-connected. In this paper, we show that EVAP$(k, 3)$ can be solved in polynomial time for any fixed k and an arbitrary input multigraph.

2 Definitions

For a subset $V' \subseteq V$ in G, $G - V'$ denotes the subgraph induced by $V - V'$. For an edge set F with $F \cap E = \emptyset$, we denote $G = (V, E \cup F)$ by $G + F$. An edge with end vertices u and v is denoted by (u, v). A *partition* $X_1, \cdots, X_t$ of vertex set V means a family of nonempty disjoint subsets of V whose union is V, and a *subpartition* of V means a partition of a subset of V. For two disjoint subsets of vertices X, $Y \subset V$, we denote by $E_G(X, Y)$ the set of edges, one of whose end vertices is in X and the other is in Y, and also denote $c_G(X, Y) = |E_G(X, Y)|$. A *cut* is defined as a subset X of V with $\emptyset \neq X \neq V$, and the *size* of a cut X is denoted by $c_G(X, V - X)$, which may also be written as $c_G(X)$. A cut with the minimum size is called a *minimum cut*, and its size, denoted by $\lambda(G)$, is called the *edge-connectivity* of G. For a subset X of V, $\{v \in V - X \mid (u, v) \in E$ for some $u \in X\}$ is called the *neighbor set* of X, denoted by $\Gamma_G(X)$. Let $p(G)$ denote the number of components in G. A *disconnecting set* of G is defined as a cut S of V such that $p(G - S) > p(G)$ holds and no $S' \subset S$ has this property. If G is connected and does not contain K_n, then a disconnecting set of the minimum size is called a *minimum disconnecting set*, and its size, denoted by $\kappa(G)$, is called the *vertex-connectivity* of G. On the other hand, we define $\kappa(G) = 0$ if G is not connected, and $\kappa(G) = n - 1$ if G is connected and contains the complete graph K_n. If $\kappa(G) = 2$, then we call a minimum disconnecting set S a *disconnecting pair* in G.

2.1 Edge-Splitting

We introduce an operation of transforming a graph, called *edge-splitting*, which is helpful to solve the edge-connectivity augmentation problem.

Given a multigraph $G = (V, E)$, a designated vertex $s \in V$, vertices $u, v \in \Gamma_G(s)$ (possibly $u = v$) and a nonnegative integer $\delta \leq \min\{c_G(s, u), c_G(s, v)\}$, we construct graph $G' = (V, E')$ from G by deleting δ edges from $E_G(s, u)$ and $E_G(s, v)$, respectively, and adding new δ edges to $E_G(u, v)$: $c_{G'}(s, u) := c_G(s, u) - \delta$, $c_{G'}(s, v) := c_G(s, v) - \delta$, $c_{G'}(u, v) := c_G(u, v) + \delta$, $c_{G'}(x, y) := c_G(x, y)$ for all other pairs $x, y \in V$. We say that G' is obtained from G by *splitting* (s, u) and (s, v) by size δ, and denote the resulting graph G' by $G/(u,v;\delta)$. A sequence of splittings is *complete* if the resulting graph G' does not have any neighbor of s.

The following theorem is proven by Mader [4].

Theorem 2.1 [4] *Let $G=(V,E)$ be a multigraph with a designated vertex $s \in V$ with $c_G(s) \neq 1,3$ and $\lambda_G(x,y) \geq 2$ for all pairs $x,y \in V$. Then for any edge $(s,u) \in E$ there is an edge $(s,v) \in E$ such that $\lambda_{G/(u,v;1)}(x,y) = \lambda_G(x,y)$ holds for all pairs $x,y \in V-s$.* □

This says that if $c_G(s)$ is even, there always exists a complete splitting at s such that the resulting graph G' satisfies $\lambda_{G'-s}(x,y) = \lambda_G(x,y)$ for every pair of $x,y \in V-s$.

3 EVAP(k,3) for an Arbitrary Graph

We now present a polynomial time algorithm for EVAP(k,3) for an arbitrary input multigraph.

Let $\beta(G) \equiv \max\{p(G-S)-1+\max[0, \max\{k-c_G(v_1), k-c_G(v_2)\}\} \mid S=\{v_1,v_2\}$ is a pair of two vertices in $G]$. To make a graph G $(k,3)$-connected, it is necessary to add at least $k-c_G(X)$ edges to $E_G(X,V-X)$ for each cut X, to add at least $3-|\Gamma_G(X)|$ edges to $E_G(X,V-X)$ for each cut X with $V-X-\Gamma_G(X)\neq\emptyset$, and to add at least $p(G-S)-1+\max[0, \max\{k-c_G(v_1), k-c_G(v_2)\}\}]$ edges to connect components of $G-S$ for each two vertex pair $S=\{v_1,v_2\}$ in G.

Lower Bound: $\gamma(G)\equiv\max\{\lceil\alpha(G)/2\rceil, \beta(G)\}$, where

$$\alpha(G)=\max\left\{\sum_{i=1}^{p}(k-c_G(X_i))+\sum_{i=p+1}^{q}(3-|\Gamma_G(X_i)|)\right\}$$

and the max is taken over all subpartitions $\{X_1,\cdots,X_p,X_{p+1},\cdots,X_q\}$ of V such that $q\geq p\geq 0$ and $V-X_i-\Gamma_G(X_i)\neq\emptyset$, $i=p+1,\cdots,q$. □

For a subset F of edges in a multigraph G, we say that two edge $e_1=(u_1,w_1)$ and $e_2=(u_2,w_2)$ are *switched* in F if we delete e_1 and e_2 from F, and add edges (u_1,u_2) and (w_1,w_2) to F, and that an edge $e_1=(u_1,w_1)$ is *shifted* in F, if we delete e_1 from F and add an edge (u_1,w_2) $(w_1\neq w_2)$ to F. The sketch of our algorithm for solving the EVAP(k,3) for an arbitrary multigraph, denoted by Algorithm EV-AUGMENT3*, is given as follows. This algorithm is based on EV-AUGMENT [2] and EV-AUGMENT3 [3].

Algorithm EV-AUGMENT3*

Input: An undirected multigraph $G=(V,E)$ and $k\geq 4$.

Output: An undirected multigraph $G^*=G+F$ with $\lambda(G^*)\geq k$ and $\kappa(G^*)\geq 3$ where the size of new edge set F is the minimum.

Step I. (Adding vertex s and associated edges): After adding a new vertex s, we can add a set F_1 of new edges between s and V so that $|F_1|=\alpha(G)$ and the resulting graph $G_1=(V\cup\{s\},E\cup F_1)$ satisfies $c_{G_1}(X)\geq k$ for all cuts $X\subset V$, $|\Gamma_G(X)|+|\Gamma_{G_1}(s)\cap X|\geq 3$ for all cuts $X\subsetneq V$ with $V-X-\Gamma_{G_1}(X)\neq\emptyset$ and $|\Gamma_G(X)|+|X|\geq 3$, and $|\Gamma_G(X)|+c_{G_1}(s,x)\geq 3$ for all cuts $X=\{x\}$ with $x\in V$.

Step II. (Edge-splitting): We obtain a $(k,2)$-connected multigraph $G_2=(V,E\cup F_2)$ (ignoring the isolated vertex s), by combining algorithm EV-AUGMENT [2] and a complete edge-splitting at s in G_1 preserving the k-edge-connectivity (according to Theorem 2.1).

If $\kappa(G_2)\geq 3$, we are done because $|F_2|=|F_1|/2=\lceil\alpha(G)/2\rceil$ implies that G_2 is optimally augmented by lower bound $\lceil\alpha(G)/2\rceil$. Otherwise, go to Step III.

Step III. (Switching and Shifting edges): Now G_2 has disconnecting pairs. We repeat switching or shifting edges $e_1,e_2\in F_2$ with $\kappa(G_2-\{e_1,e_2\})\geq 2$ so that the resulting graph G_2' satisfies the k edge-connectivity, , $\kappa_{G_2'}(x,y)\geq 3$ for all $x,y\in V$ with $\kappa_{G_2}(x,y)\geq 3$, and $p(G_2'-S)<p(G_2-S)$ for some disconnecting pairs S.

Let $G_3=(V,E\cup F_3)$ be the resulting graph, where F_3 denotes the final F_2. If G_3 has no disconnecting pair, then we are done, since $|F_3|=\lceil\alpha(G)/2\rceil$ implies that G_3 is optimally augmented. Otherwise, go to Step IV.

Step IV. (Edge augmentation): Now one of the following (i) or (ii) satisfy:

(i) We can make G_3 3-vertex-connected by adding a set F_4 of $\beta(G)-\lceil\alpha(G)/2\rceil$ new edges, i.e., we are done since $|F_3|+|F_4|=\beta(G)$ implies that the resulting graph is optimally augmented by lower bound $\beta(G)$.

(ii) The input graph G can be made $(k,3)$-connected by adding $O(k)$ edges.

Theorem 3.1 *For an arbitrary multigraph G, G can be made $(k,3)$-connected by adding $\gamma(G)=\max\{\lceil\alpha(G)/2\rceil,\beta(G)\}$ new edges or $O(k)$ new edges in polynomial time.* □

References

[1] L. R. Ford and D. R. Fulkerson, Flows in Networks, Princeton University Press, Princeton, N. J., 1962.

[2] T. Ishii, H. Nagamochi, and T. Ibaraki, *Augmenting edge-connectivity and vertex-connectivity simultaneously*, LNCS, Springer-Verlag, 8th ISAAC 1997, pp. 102-111.

[3] T. Ishii, H. Nagamochi, and T. Ibaraki, *Optimal augmentation to make a graph k-edge-connected and triconnected*, Proc. of 9th Annual ACM-SIAM Symposium on Discrete Algorithms, 1998, pp. 280-289.

[4] W. Mader, *A reduction method for edge-connectivity in graphs*, Ann. Discrete Math., Vol.3, 1978, pp. 145-164.