Multigraph Augmentation under Biconnectivity and General Edge-Connectivity Requirements *

Toshimasa Ishii[†], Hiroshi Nagamochi[†] and Toshihide Ibaraki[††]

[†]Department of Information and Computer Sciences,
Toyohashi University of Technology,
Aichi 441-8580, Japan.
{ishii,naga}@ics.tut.ac.jp

[††]Department of Applied Mathematics and Physics,
Kyoto University
Kyoto 606-8501, Japan.
ibaraki@amp.i.kyoto-u.ac.jp

Abstract Given an undirected multigraph $G = (V, E)$ and a requirement function $r_\lambda : \binom{V}{2} \to Z^+$ (where $\binom{V}{2}$ is the set of all pairs of vertices and Z^+ is the set of nonnegative integers), we consider the problem of augmenting G by the smallest number of new edges so that the local edge-connectivity and vertex-connectivity between every pair $x, y \in V$ become at least $r_\lambda(x, y)$ and two, respectively. In this paper, we show that the problem can be solved in $O(n^3(m + n) \log(n^2/(m + n)))$ time, where n and m are the numbers of vertices and pairs of adjacent vertices in G, respectively. This time complexity can be improved to $O((nm + n^2 \log n) \log n)$, in the case of the uniform requirement $r_\lambda(x, y) = \ell$ for all $x, y \in V$. Furthermore, for the general r_λ, we show that the augmentation problem that preserves the simplicity of the resulting graph can be solved in polynomial time for any fixed $\ell^* = \max\{r_\lambda(x, y) \mid x, y \in V\}$.

Keywords: undirected multigraph, edge-connectivity, vertex-connectivity, graph augmentation, polynomial deterministic algorithm.

1 Introduction

Let $G = (V, E)$ stand for an undirected multigraph with a set V of *vertices* and a set E of *edges*, where we denote $|V|$ by n (or by $n(G)$) and the number of pairs of vertices which are adjacent in G by m (or by $m(G)$). An edge with end vertices u and v is denoted by (u, v). Throughout

*An extended abstract of this paper was presented at 8th International Symp. on Algorithms and Computation (ISAAC'97), Singapore, December 1997, under the title "Augmenting Edge and Vertex Connectivities Simultaneously".

the paper, an undirected multigraph is called a graph unless a confusion arises. The *local edge-connectivity* $\lambda_G(x,y)$ (resp., the *local vertex-connectivity* $\kappa_G(x,y)$) for two vertices $x, y \in V$ is defined to be the maximum number of edge-disjoint paths (resp., vertex-disjoint paths) between x and y. For a function $r_\lambda : \binom{V}{2} \to Z^+$ (resp., $r_\kappa : \binom{V}{2} \to Z^+$), where $\binom{V}{2}$ denotes the set of pairs of vertices and Z^+ denotes the set of nonnegative integers, we say that $G = (V, E)$ is r_λ*-edge-connected* (resp., r_κ*-vertex-connected*) if $\lambda_G(x,y) \geq r_\lambda(x,y)$ (resp., $\kappa_G(x,y) \geq r_\kappa(x,y)$) holds for every $x, y \in V$. In particular, for a nonnegative integer ℓ (resp., k), G is called ℓ*-edge-connected* (resp., k-vertex-connected), if $\lambda_G(x,y) \geq \ell$ (resp., $\kappa_G(x,y) \geq k$) holds for every $x, y \in V$. Then the r_λ*-edge-connectivity augmentation problem* (resp., the r_κ*-vertex-connectivity augmentation problem*) asks to augment G by a smallest number of new edges so that the resulting multigraph G' becomes r_λ-edge-connected (resp., r_κ-vertex-connected).

Multigraph augmentation problems to meet edge-connectivity or vertex-connectivity requirement have been extensively studied as important subjects in the network design problem, the data security problem [25] and the graph drawing problem [23, 24] and others.

Watanabe and Nakamura [31] first proved that the ℓ-edge-connectivity augmentation problem can be solved in polynomial time for any given integer ℓ. Their algorithm increases the edge-connectivity one by one on the basis of the structural information of G in order to an ℓ-edge-connected graph. Currently, an $O(e + \ell^2 n \log n)$ time algorithm due to Gabow [6], where $e = |E|$, is the fastest among existing algorithms of this type. Different from the approach by Watanabe and Nakamura, Cai and Sun [2] first pointed out that the ℓ-edge-connectivity augmentation problem can be directly solved by applying the edge-splitting theorem. Based on this, Frank [4] gave a refined $O(n^5)$ time augmentation algorithm by using Lovász edge-splitting theorem. Recently, an $O(n(m + n \log n) \log n)$ time augmentation algorithm is proposed by Nagamochi and Ibaraki [28]. For a general requirement function r_λ, Frank [4] showed that the edge-connectivity augmentation problem can be solved in polynomial time by using Mader's edge-splitting theorem [27]. The time complexity for this problem was recently improved by Gabow [7] to $O(n^3 m \log(n^2/m))$.

As to vertex-connectivity augmentation, several algorithms have been developed to add the minimum number of new edges to make a $(k-1)$-vertex-connected graph G k-vertex-connected. Eswaran and Tarjan [3] proved that the vertex-connectivity augmentation problem for $k = 2$ can be solved. Watanabe and Nakamura [32] stated the same result for $k = 3$. For a general k, Jordán presented an $O(n^5)$ time approximation algorithm for this problem [20, 21] such that the gap between the number of new edges added by his algorithm and the optimal value is at most $(k-2)/2$.

It is known that the uniform k-vertex-connectivity augmentation problem for $k \in \{2, 3, 4\}$ can be solved in polynomial time ([3, 14] for $k = 2$, [13, 32] for $k = 3$, and [11] for $k = 4$), where an input graph G may not be $(k-1)$-vertex-connected. However, whether there is a polynomial time algorithm for the vertex-connectivity augmentation problem for an arbitrary k is an open question (even if G is $(k-1)$-vertex-connected). For a general requirement function r_κ, the problem was shown to be NP-hard by Jordán [19].

In a communication network, both the edge-connectivity and vertex-connectivity are fundamental measures of reliability against link failures and node failures, respectively. In this paper, therefore, we consider the problem of augmenting G by the smallest number of new edges to satisfy both edge-connectivity and vertex-connectivity requirements. This problem has not been studied much except for the following work by Hsu and Kao [12]. Given a multigraph $G = (V, E)$ with two specified subsets X and Y of V, they present a linear time algorithm for augmenting G by the smallest number of edges so that the resulting multigraph G' satisfies $\lambda_{G'}(x, x') \geq 2$ for all $x, x' \in X$ and $\kappa_{G'}(y, y') \geq 2$ for all $y, y' \in Y$.

Now we define the *edge-and-vertex-connectivity augmentation problem*, denoted by EVAP(r_λ, r_κ), as the problem of augmenting G by the smallest number of new edges so that the resulting multigraph G' becomes r_λ-edge-connected and r_κ-vertex-connected (hereafter we call this (r_λ, r_κ)-connected). Without loss of generality, $r_\lambda(x, y) \geq r_\kappa(x, y)$ is assumed for all $x, y \in V$, since if a multigraph is r_κ-vertex-connected then it is r_κ-edge-connected. Clearly, EVAP(r_λ, r_κ) contains the edge-connectivity augmentation problem and the vertex-connectivity augmentation problem as its special cases. When the requirement function r_κ satisfies $r_\kappa(x, y) = \ell \in Z^+$ for all $x, y \in V$, this problem is also denoted as EVAP(r_λ, ℓ). In this paper, we present an algorithm to solve problem EVAP$(r_\lambda, 2)$. We first derive a lower bound on the number of edges in order to make a given multigraph G $(r_\lambda, 2)$-connected, and then show that this lower bound is always attainable by an optimal solution. The task of constructing such an optimal set of new edges can be performed in $O(n^3(m + n)\log(n^2/(m + n)))$ time.

In Section 2, after introducing basic definitions, we present a lower bound on the number of new edges necessary to make a multigraph G (r_λ, r_κ)-connected and introduce the concept of edge-splitting. In Section 3, we outline our algorithm for finding such an edge set. In Sections 4 – 7, we prove the correctness of our algorithm. In Section 8, we consider the problem of a simple graph while preserving the simplicity of the graph, and show that this can be solved in polynomial time for any fixed $k = \max\{r_\lambda(x, y) \mid x, y \in V\}$. In Section 9, we state a concluding remark.

2 Preliminaries

2.1 Definitions

In a graph $G = (V, E)$, its vertex set V and edge set E may be denoted by $V(G)$ and $E(G)$, respectively. A singleton set $\{x\}$ may be simply written as x, and " $\subset$ " implies proper inclusion while " $\subseteq$ " means " $\subset$ " or " $=$ ". A subset X *intersects* another subset Y if none of subsets $X \cap Y$, $X - Y$ and $Y - X$ is empty. We say that a subset X *crosses* another subset Y if they intersect each other and in addition $V - (X \cup Y) \neq \emptyset$ holds. A *partition* $X_1, \ldots, X_t$ of the 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 V' of V.

For a subset $V' \subseteq V$ (resp., $E' \subseteq E$) in G, $G[V']$ denotes the subgraph induced by V'. For $V' \subset V$ (resp., $E' \subseteq E$), we denote subgraph $G[V - V']$ (resp., $(V, E - E')$) by $G - V'$ (resp., $G - E'$). For an edge set E' with $E' \cap E = \emptyset$, we denote the augmented graph $G = (V, E \cup E')$ by

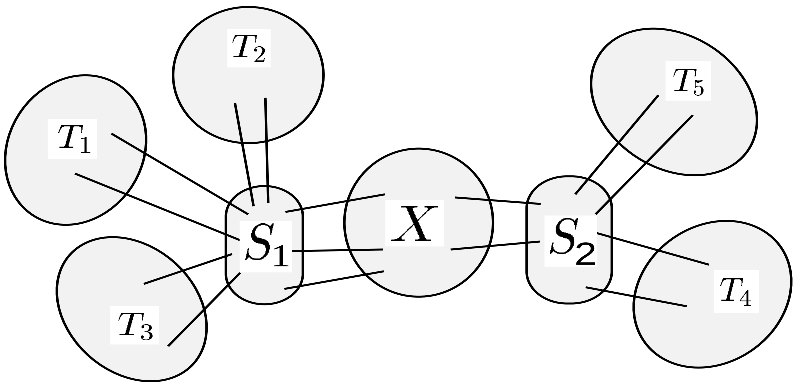

Figure 1: Illustrations of a multigraph which has exactly two minimum disconnecting set S_1 and S_2. Each of cuts T_1, T_2, T_3, and $X \cup S_2 \cup T_4 \cup T_5$ is tight, since its neighbor set is the minimum disconnecting set S_1. Similarly with respect to the minimum disconnecting set S_2, each of cuts T_4, T_5 and $X \cup S_1 \cup T_1 \cup T_2 \cup T_3$ is tight. In particular, cuts T_i for $i = 1, \ldots, 5$ are minimal tight sets since no $T' \subset T_i$ is tight.

$G + E'$. For two disjoint subsets of vertices $X,\ Y \subset V$, we denote by $E_G(X, Y)$ the set of edges $e = (x, y)$ such that $x \in X$ and $y \in Y$, and also denote $|E_G(X, Y)|$ by $c_G(X, Y)$. In particular, $E_G(u, v)$ is the set of edges with end vertices u and v.

A *cut* is defined as a subset X of V with $\emptyset \neq X \neq V$, and the *size* of a cut X is defined by $c_G(X, V - X)$, which may also be written as $c_G(X)$. A cut with the minimum size is called a (*global*) *minimum cut*, and its size, denoted by $\lambda(G)$, is called the *edge-connectivity* of G. We say that a cut X *separates* two disjoint subsets Y and Y' of V if $Y \subseteq X$ and $Y' \subseteq V - X$ (or $Y \subseteq V - X$ and $Y' \subseteq X$). In particular, a cut X separates vertices x and y if $x \in X$ and $y \in V - X$ (or $x \in V - X$ and $y \in X$) hold. The local edge-connectivity $\lambda_G(x, y)$ for two vertices $x, y \in V$ is also defined to be the minimum size of a cut in G that separates x and y by Menger's theorem. An edge e whose removal from G increases the number of components is called a *bridge* of G.

For a subset X of V, a vertex $v \in V - X$ is called a *neighbor* of X if it is adjacent to some vertex $u \in X$, and the set of all neighbors of X is denoted by $\Gamma_G(X)$. A maximal connected subgraph G' in a graph G is called a *component* of G (for notational convenience, a component H may be represented by its vertex set $X = V(H)$), and denote the number of components in G by $p(G)$.

A *disconnecting set* of G is defined as a subset S of V such that $p(G - S) > p(G)$ holds and no $S' \subset S$ has this property. We say that a set $S \subset V$ *disconnects* two disjoint subsets Y and Y' of $V - S$ if no two vertices $x \in Y$ and $y \in Y'$ are connected in $G - S$. In particular, S disconnects vertices x and y if x and y are contained in different components of $G - S$. Also by Menger's theorem, $\kappa_G(x, y)$ for nonadjacent vertices x and y is equal to the minimum size of a disconnecting set S that disconnects x and y. Let $\hat{G}$ denote the simple graph obtained from G by replacing multiple edges in $E_G(u, v)$ by a single edge (u, v) for all $u, v \in V$. A component

G_1 of G with $|V(G_1)| \geq 3$ always has a disconnecting set unless $\hat{G}_1$ is the complete graph. If G is connected and contains a disconnecting set, then a disconnecting set of the minimum size is called a (*global*) *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 $\hat{G}$ is a complete graph. A vertex v is called a *cut vertex* in $G = (V, E)$ if $S = \{v\}$ is a minimum disconnecting set in G. A subset $X \subset V$ is *biconnected* if $\kappa_G(x, y) \geq 2$ holds for all $x, y \in X$. A cut $T \subset V$ is called *tight* if $\Gamma_G(T)$ is a *minimum* disconnecting set in G (see Figure 1). Note that every tight set T satisfies $V - T - \Gamma_G(T) \neq \emptyset$. A tight set T is called *minimal* if no $T' \subset T$ is tight (hence, the induced subgraph $G[T]$ is connected). Let $t(G)$ be the maximum number of pairwise disjoint minimal tight sets in G. For a subset $S \subset V$, a component T in $G - S$ is called an S*-component* if $\Gamma_G(T) \cap S \neq \emptyset$ holds. If $S = \{x\}$, then such a component is called an x*-component*. Note that v is a cut vertex of G if and only if there is more than one v-component. It is not difficult to observe the following lemma about x-components for a vertex $x \in V$.

Lemma 2.1 [15] *Let $X \subset V$ be an x-component of a vertex $x \in V$ in a multigraph $G = (V, E)$. If X contains a cut vertex y in G, then there is a y-component $Y \subset X$.* □

2.2 Lower Bounds

In this section, we derive two lower bounds, $\lceil \hat{\alpha}(G)/2 \rceil$ and $\hat{\beta}(G)$, on the number of new edges that is necessary to make a given multigraph G $(r_\lambda, 2)$-connected. Let us first define

$$r_\lambda(X) = \max\{r_\lambda(u, v) \mid u \in X, v \in V - X\} \text{ for each cut } X.$$

To make G r_λ-edge-connected and 2-vertex-connected, it is necessary to add

(a) at least $\max\{r_\lambda(X) - c_G(X), 0\}$ edges between X and $V - X$, for each cut X (see Figure 2(a)),

(b) at least $\max\{2 - |\Gamma_G(X)|, 0\}$ edges between X and $V - X - \Gamma_G(X)$ for each cut X with $V - X - \Gamma_G(X) \neq \emptyset$ (see Figure 2(b)).

Therefore, given a subpartition $\mathcal{X} = \{X_1, \ldots, X_p, X_{p+1}, \ldots, X_q\}$ of V with $V - X_i - \Gamma_G(X_i) \neq \emptyset$ for $i = p+1, \ldots, q$, we can sum up "deficiency" $\max\{r_\lambda(X_i) - c_G(X_i), 0\}$, $i = 1, \ldots, p$, and $\max\{2 - |\Gamma_G(X_i)|, 0\}$, $i = p+1, \ldots, q$. Since adding one edge to G contributes to the deficiency of at most two cuts in $\mathcal{X}$, we need at least $\lceil \hat{\alpha}(G)/2 \rceil$ new edges to make G (r_λ, r_κ)-connected, where

$$\hat{\alpha}(G) = \max_{\text{all subpartitions } \mathcal{X}} \left\{ \sum_{i=1}^{p} (r_\lambda(X_i) - c_G(X_i)) + \sum_{i=p+1}^{q} (2 - |\Gamma_G(X_i)|) \right\}, \tag{2.1}$$

and the max is taken over all subpartitions $\mathcal{X} = \{X_1, \ldots, X_p, X_{p+1}, \ldots, X_q\}$ of V with $V - X_i - \Gamma_G(X_i) \neq \emptyset$, $i = p+1, \ldots, q$.

In a 2-vertex-connected graph, the deletion of any one vertex in V does not disconnect the graph. Hence in order to make G 2-vertex-connected, it is necessary to add

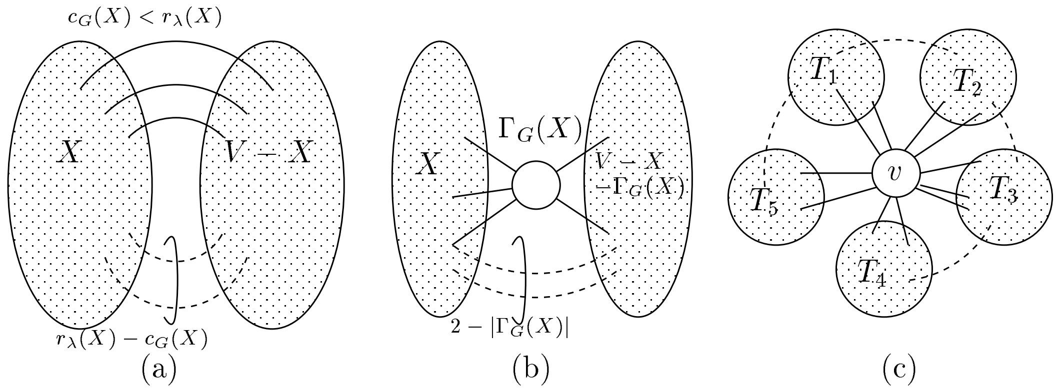

Figure 2: Illustrations of necessary edge augmentations.

(c) at least $p(G-v)-1$ edges to connect components of $G-v$ for a vertex v (see Figure 2(c)).

Let us define

$$\hat{\beta}(G) = \max\{p(G-v)-1 | v \in V\}. \tag{2.2}$$

Combining the above two lower bounds from (2.1) and (2.2), we establish the next lemma.

Lemma 2.2 (Lower Bound) *To make a given multigraph G $(r_\lambda, 2)$-connected, at least*

$$\gamma(G) = \max\{\lceil \hat{\alpha}(G)/2 \rceil, \hat{\beta}(G)\}$$

new edges must be added, where $\hat{\alpha}(G)$ and $\hat{\beta}(G)$ are given by (2.1) and (2.2), respectively. □

This says that a set of new edges is an optimal solution to EVAP(r_λ,2) if its size is equal to $\gamma(G)$ and the resulting multigraph is $(r_\lambda, 2)$-connected. We will show that this is always the case by presenting a polynomial time algorithm for constructing such a set of edges.

2.3 Edge-Splitting

In this subsection, we review the operation of *edge-splitting.* 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 multigraph $G' = (V, E')$ by deleting δ edges from both $E_G(s,u)$ and $E_G(s,v)$, and adding new δ edges to $E_G(u,v)$; $c_{G'}(s,u) := c_G(s,u) - \delta$, $c_{G'}(s,v) := c_G(s,v) - \delta$, $c_{G'}(u,v) := c_G(u,v) + \delta$, and $c_{G'}(x,y) := c_G(x,y)$ for all other pairs $x, y \in V$. In the case $u = v$, we interpret that $c_{G'}(s,u) := c_G(s,u) - 2\delta$, $c_{G'}(u,u) := c_G(u,u) + \delta$, and $c_{G'}(x,y) := c_G(x,y)$ for all other pairs $x, y \in V$, where an integer δ is chosen so as to satisfy $0 \leq \delta \leq \frac{1}{2}c_G(s,u)$. We say that G' is obtained from G by *splitting* δ pair of edges (s,u) and (s,v) (or by splitting (s,u) and (s,v) by size δ). A sequence of splittings is *complete* if the resulting multigraph G' does not have any neighbor of s. The following theorem is proven by Mader [27].

Theorem 2.1 [27] *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 $x,y \in V-s$. Then there is a pair of two edges $e_1, e_2 \in E_G(s)$ such that the multigraph G' obtained by splitting edges e_1 and e_2 satisfies $\lambda_{G'}(x,y) = \lambda_G(x,y)$ for all pairs $x,y \in V-s$.* $\Box$

Repeating this, we see that, if $c_G(s)$ is even, there always exists a complete splitting at s such that the resulting multigraph G' satisfies $\lambda_{G'-s}(x,y) = \lambda_G(x,y)$ for every pair of $x,y \in V-s$. Gabow [7] proved that such a complete splitting at s can be computed in $O(n^3 m \log(n^2/m))$ time and the number of new pairs of vertices which became adjacent by the created edges is $O(n)$.

3 A Polynomial Time Algorithm for EV-AUGMENT

In this section, a polynomial time algorithm for solving EVAP(r_λ,2), called EV-AUGMENT, is presented. For this, we introduce some definitions. Given a cut vertex v in G, an edge $e = (u,w)$ with $u,w \neq v$ is called *admissible* with respect to v, if $p((G-v)-e) = p(G-v)$. By definition, there is no admissible edge if G has no cut vertex. For a subset F of edges in G, we say that two edges $e_1 = (u_1,w_1)$ and $e_2 = (u_2,w_2)$ in F 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.

EV-AUGMENT consists of the following four major steps. In each step, we also describe some properties used to show its correctness. The proofs for these properties will be given in the subsequent sections. Figure 3 illustrates the process of these four steps for an example multigraph.

Algorithm EV-AUGMENT

Input: An undirected multigraph $G = (V,E)$ with $|V| \geq 3$, and a requirement function $r_\lambda : \binom{V}{2} \to Z^+$.

Output: A set F of the smallest number of new edges such that $G+F$ is $(r_\lambda, 2)$-connected.

Step I (Addition of vertex s and associated edges): Add a new vertex s together with a set F_1 of edges between s and V so that the resulting multigraph $G_1 = (V \cup \{s\}, E \cup F_1)$ satisfies

$$c_{G_1}(X) \geq r_\lambda(X) \qquad \text{for all cuts } X \subset V, \tag{3.1}$$

$$|\Gamma_G(X)| + |\Gamma_{G_1}(s) \cap X| \geq 2 \qquad \text{for all cuts } X \subset V \text{ such } V - X - \Gamma_G(X) \neq \emptyset \tag{3.2}$$

(except for $X = \{x\}$ which is an isolated vertex in G; i.e., $|X| = 1$ and $\Gamma_G(X) = \emptyset$)

and F_1 is minimal (i.e., any proper subset of F_1 violates (3.1) or (3.2)). By (3.1), G_1 satisfies r_λ-edge-connectivity: $\lambda_{G_1}(x, y) \geq r_\lambda(x, y)$ for all $x, y \in V$.

Property 3.1 *The subset F_1 obtained in Step I satisfies $|F_1| = \hat{\alpha}(G)$.* □

Step II (Edge-splitting): If $c_{G_1}(s)$ is odd, then add one edge $\hat{e} = (s, \hat{w})$ to F_1 by choosing an arbitrary vertex $\hat{w} \in V$ which is not a cut vertex in G.

Then find a complete edge-splitting at s in $G_1 = (V \cup \{s\}, E \cup F_1)$ which preserves the r_λ-edge-connectivity, i.e., $\lambda_{G_2}(x, y) \geq r_\lambda(x, y)$ for all pairs $x, y \in V$, where $G_2 = (V, E \cup F_2)$ denotes the resulting multigraph (ignoring the isolated vertex s). By Theorem 2.1, there always exists such a complete edge-splitting.

If $\kappa(G_2) \geq 2$, then we are done, because $|F_2| = |F_1|/2 = \lceil \hat{\alpha}(G)/2 \rceil$ attains the lower bound of Lemma 2.2. Otherwise, go to Step III.

Step III (Switching edges): The current multigraph G_2 is r_λ-edge-connected, but has cut vertices. G_2 satisfies

$$\begin{array}{l}\text{for any cut vertex } v \text{ and its } v\text{-component } T, G_2[T \cup \{v\}] \text{ contains} \\ \text{at least one edge in } F_2,\end{array} \tag{3.3}$$

since if this does not hold, it means by the property of edge splitting that T contains no end vertex of an edge in F_1 in G_1 in Step I; $|\Gamma_G(T)| \leq 1$ and $c_{G_1}(s, T) = 0$ in Step I, contradicting (3.2).

We switch some number of pairs of edges $e_1, e_2 \in F_2$ to recover 2-vertex-connectivity of G_2 while preserving the r_λ-edge-connectivity.

Property 3.2 *If G_2 has two cut vertices v_1 and v_2, then there are v_1-component T_1 and v_2-component T_2 such that $T_1 \cap T_2 = \emptyset$. For this T_1, an arbitrary edge $e_1 \in F_2$ in $G_2[T_1 \cup \{v_1\}]$ is admissible with respect to v_2.* □

Property 3.3 *Given a cut vertex v in G_2, assume that there is an edge $e_1 \in F_2$ in a v-component T_1 of G_2, admissible with respect to v. Let T_2 be another v-component such that $e_1 \notin E(G_2[T_2 \cup \{v\}])$. Then $G_2[T_2 \cup \{v\}]$ contains an edge $e_2 \in F_2$ (by (3.3)), and the multigraph G_2' resulting from switching e_1 and e_2 satisfies the followings.*

(i) *$\lambda_{G_2'}(x, y) \geq r_\lambda(x, y)$ for all $x, y \in V$ (i.e., the r_λ-edge-connectivity is preserved).*

(ii) *$p(G_2' - v) < p(G_2 - v)$ (i.e., the number of v-components in G_2 decreases at least by one).*

(iii) *$\kappa_{G_2'}(x, y) \geq 2$ holds for any pair $x, y \in V$ such that $\kappa_{G_2}(x, y) \geq 2$.*

(iv) *(3.3) holds in G_2'.* □

As long as Property 3.3 is applicable, we repeat switching pairs of edges $e_1, e_2 \in F_2$, by setting $G_2 := G_2'$ after each switching. Note that the number of v-components decreases and the number of v'-components with $v' \neq v$ does not increase since G_2' satisfies (ii) and (iii) in Property 3.3.

Let $G_3 = (V, E \cup F_3)$ be the multigraph obtained by such a sequence of switchings in F_2, where F_3 denotes the final F_2. Clearly, $|F_3| = |F_2| = \lceil \hat{\alpha}(G)/2 \rceil$ holds.

Note that if G_2 has at least two cut vertices, Property 3.2 ensures that the condition of Property 3.3 holds (consider v_2 as the cut vertex v). If G_3 has no cut vertex, then we are done, since $|F_3| = \lceil \hat{\alpha}(G)/2 \rceil$ implies that G_3 is optimally augmented. Now we assume that G_3 has exactly one cut vertex, and go to Step IV.

Step IV (Edge augmentation): The current G_3 has exactly one cut vertex v, and we can prove the following property.

Property 3.4 *For the multigraph G_3 and its cut vertex v, it holds $p(G_3 - v) = p(G - v) - \lceil \hat{\alpha}(G)/2 \rceil$.* □

Denote $G_3' = G_3 - v$, and consider all v-components $T_1, \ldots, T_{p(G_3')}$ in G_3. Choose a vertex x_i from each T_i, and add a set F_4 of $p(G_3') - 1$ edges (x_i, x_{i+1}), $i = 1, \ldots, p(G_3') - 1$ to G_3, by which G_3 becomes a 2-vertex-connected multigraph $G_4 = G_3 + F_4$. From Property 3.4 and $\hat{\beta}(G) \geq p(G - v) - 1$ (by (2.2)), we see that $p(G_3') - 1 = p(G - v) - \lceil \hat{\alpha}(G)/2 \rceil - 1 \leq \hat{\beta}(G) - \lceil \hat{\alpha}(G)/2 \rceil$. Therefore, $|F_3| + |F_4| = \lceil \hat{\alpha}(G)/2 \rceil + (p(G_3') - 1) \leq \hat{\beta}(G)$. By the lower bound $\hat{\beta}(G)$ of Lemma 2.2, this implies that G_4 is optimally augmented. □

This algorithm, together with the proofs and complexity analysis in the subsequent sections, establishes the next theorem.

Theorem 3.1 *Given a multigraph G with n vertices and m edges, and a requirement function $r_\lambda : \binom{V}{2} \to Z^+$, G can be augmented to a $(r_\lambda, 2)$-connected multigraph by adding $\gamma(G) = \max\{\lceil \hat{\alpha}(G)/2 \rceil, \hat{\beta}(G)\}$ new edges in $O(n^3(m+n)\log(n^2/(m+n)))$ time.* □

4 Step I

This section proves Property 3.1, which ensures that Step I correctly computes $\hat{\alpha}(G)$.

Proof of Property 3.1: Let G_1 be the multigraph obtained from G by Step I. By (3.1), G_1 satisfies $\lambda_{G_1}(x, y) \geq r_\lambda(x, y) \geq 2$ for all $x, y \in V$. First we see $|F_1| \geq \hat{\alpha}(G)$, since otherwise some cut $X \subset V$ violates (3.1) or (3.2).

In what follows, we prove the converse, $|F_1| \leq \hat{\alpha}(G)$. A cut $X \subset V$ is called *critical* in G_1 if $s \in \Gamma_{G_1}(X)$ holds and the removal of some edge $e \in E_{G_1}(s, X)$ violates (3.1) or (3.2). Clearly, a cut $X \subset V$ with $s \in \Gamma_{G_1}(X)$ is critical if and only if at least one of the following conditions holds:

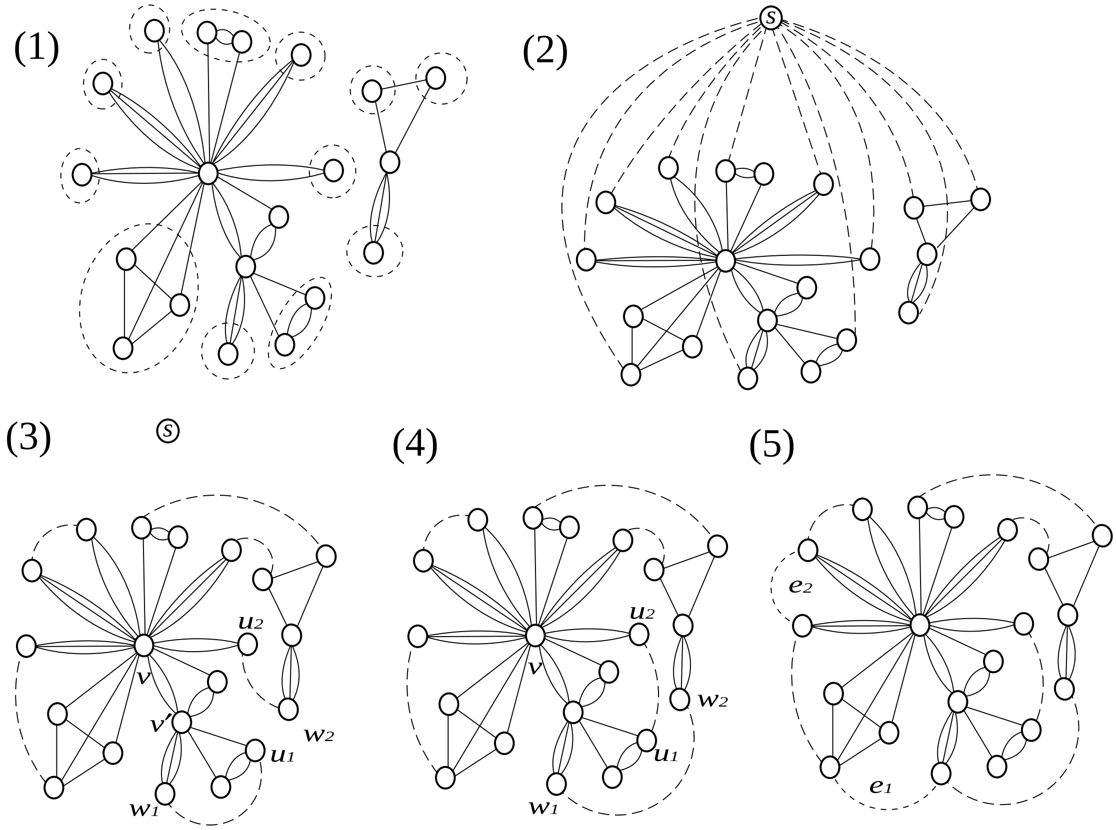

Figure 3: Computation of algorithm EV-AUGMENT for the requirement $r_\lambda(x,y) = 3$ and $r_\kappa(x,y) = 2$ for all $x, y \in V$. (1) An input multigraph $G = (V, E)$. The two lower bounds given in Section 2 are $\lceil \frac{\hat{\alpha}(G)}{2} \rceil = \frac{12}{2} = 6$ and $\hat{\beta}(G) = 9 - 1 = 8$, where the corresponding subpartition is illustrated by broken circles. (2) The multigraph $G_1 = (V \cup \{s\}, E \cup F_1)$ after Step I. Edges in F_1 are drawn as broken lines. Observe that $c_{G_1}(X) \geq 3$ for all cuts $X \subset V$, and $|\Gamma_{G_1}(X \cup s)| \geq 2$ for all cuts $X \subset V$ with $V - X - \Gamma_{G_1}(X) \neq \emptyset$, and $|F_1| = \hat{\alpha}(G)$. (3) The multigraph $G_2 = (V, E \cup F_2)$ obtained in Step II. G_2 satisfies $\lambda(G_2) \geq 3$ but has cut vertices v and v'. The edge $(u_1, w_1) \in F_2$ is admissible with respect to v in G_2, and edge $(u_2, w_2) \in F_2$ is contained in a v-component that does not contain (u_1, w_1). (4) The multigraph $G_2' = (V, E \cup F_2')$ obtained from G_2 by switching edges (u_1, w_1) and (u_2, w_2) into (u_1, u_2) and (w_1, w_2) during Step III. Observe that $\lambda(G_2') \geq 3$ holds and the number of v-components is decreased by one. Moreover, G_2' has no admissible edge in F_2' (that is, $G_3 = G_2'$ and $F_3 = F_2'$). (5) The multigraph $G_4 = (V, E \cup F_3 \cup F_4)$ obtained by adding an edge set $F_4 = \{e_1, e_2\}$ to $G_2' (= G_3)$ in Step IV, where $\hat{\beta}(G) - \lceil \frac{\hat{\alpha}(G)}{2} \rceil = 2$. This G_4 is $(3, 2)$-connected. $\square$

(1) $c_{G_1}(X) = r_\lambda(X)$.
(2) $|\Gamma_G(X)| = 1$, $c_{G_1}(s, X) = 1$ and $V - X - \Gamma_{G_1}(X) \neq \emptyset$.
(3) $|\Gamma_G(X)| = 0$, $|\Gamma_{G_1}(s) \cap X| = 2$, and $c_{G_1}(s, u) = 1$ for some $u \in \Gamma_{G_1}(s) \cap X$.

A cut X is called *critical of type* (1) (resp., (2) and (3)) if it satisfies (1) (resp., (2) and (3)).

We will now prove via several claims that G_1 has a set of critical cuts $X_1, \ldots, X_q$ only of types (1) and (2) such that

$$\Gamma_{G_1}(s) \subseteq X_1 \cup \cdots \cup X_q \text{ and } X_i \cap X_j = \emptyset,\ 1 \leq i < j \leq q. \tag{4.1}$$

This implies that $|F_1| = \left\{ \sum_{i=1}^{p} (r_\lambda(X_i) - c_G(X_i)) + \sum_{i=p+1}^{q} (2 - |\Gamma_G(X_i)|) \right\}$, where cuts X_i for $i = 1, \ldots, p$ are of type (1) and cuts X_i for $i = p+1, \ldots, q$ are of type (2). This and the definition of $\hat{\alpha}(G)$ imply $|F_1| \leq \hat{\alpha}(G)$.

Now it is not difficult to verify the following claims (see [15] for the proofs). A critical cut X is called *u-minimal* for $u \in \Gamma_{G_1}(s) \cap X$ if there is no critical cut X' with $\{u\} \subseteq X' \subset X$.

Claim 4.1 *Any critical cut X induces a connected subgraph $G_1[X]$ ($=G[X]$).* □

Claim 4.2 *Any critical cut X of type* (3) *is also critical of type* (1). □

Claim 4.3 [4] *Let X and Y be critical cuts of type* (1) *in G_1. Then at least one of the following statements holds.*

(i) *Both $X \cap Y$ and $X \cup Y$ are critical of type* (1).

(ii) *Both $X - Y$ and $Y - X$ are critical of type* (1), *and $c_{G_1}(X \cap Y, (V \cup \{s\}) - (X \cup Y)) = 0$.*

□

Claim 4.4 *Let X and Y be critical cuts of type* (2) *such that X and Y are respectively u-minimal and v-minimal for $u \in (X - Y) \cap \Gamma_{G_1}(s)$ and $v \in (Y - X) \cap \Gamma_{G_1}(s)$. Then $X \cup Y$ is a critical cut of type* (1) *or $X \cap Y = \emptyset$.* □

Claim 4.5 *Let X be a critical cut of type* (1), *and Y be a critical cut of type* (2) *such that $\Gamma_{G_1}(s) \cap (Y - X) \neq \emptyset$. If X and Y cross each other in G_1, then $c_{G_1}(X \cap Y, s) = 0$ holds and cut $Y - X$ is critical of type* (1). □

Let N_1 be the set of $u \in \Gamma_{G_1}(s)$ such that there is a critical cut X of type (1) with $u \in X$. Let us consider a set $\mathcal{X}_1$ of critical cuts of type (1) such that $N_1 \subseteq \cup_{X \in \mathcal{X}_1} X$. We choose $\mathcal{X}_1$ so that $\sum_{X \in \mathcal{X}_1} |X|$ is minimized. For $N_2 = \Gamma_{G_1}(s) - N_1$, we choose a u-minimal critical cut X_u of type (2) for each $u \in N_2$, and let $\mathcal{X}_2 = \{X_u \mid u \in N_2\}$ (note that Claim 4.2 implies that every $v \in N_2$ is contained in some critical cut of type (2)). Then the next claim proves the existence of a set of critical cuts of (4.1).

Claim 4.6 *The above family of cuts $\mathcal{X} = \mathcal{X}_1 \cup \mathcal{X}_2$ gives disjoint critical cuts of types* (1) *and* (2) *such that $\Gamma_{G_1}(s) \subseteq \cup_{X_i \in \mathcal{X}} X_i$.*

Proof: Let $\mathcal{X}_1 = \{X_1, \ldots, X_p\}$ and $\mathcal{X}_2 = \{X_{p+1}, \ldots, X_q\}$, where $\emptyset \neq X_i \subset V$ for all i. Clearly, $\Gamma_{G_1}(s) \subseteq \cup_{X_i \in \mathcal{X}} X_i$ holds from the construction of $\mathcal{X}$. Therefore we need to prove that the cuts are pairwise disjoint.

From the construction of $\mathcal{X}_1$ and Claim 4.3, it is shown in [4] that X_i and X_j are pairwise disjoint for any two cuts $X_i, X_j \in \mathcal{X}_1$. Claim 4.4 implies that X_i and X_j are disjoint for any two cuts $X_i, X_j \in \mathcal{X}_2$, because $X_i \cup X_j$ cannot be critical of type (1) (if so, it holds $\Gamma_{G_1}(s) \cap (X_i \cup X_j) \subseteq N_1$, a contradiction).

Finally, we show that X_i and X_j are disjoint for each $X_i \in \mathcal{X}_1$ and $X_j \in \mathcal{X}_2$. Note that $\Gamma_{G_1}(s) \cap (X_j - X_i) \neq \emptyset$ holds from definition of N_1. Then $X_j \subset X_i$ does not hold. Also note that $X_i \subset X_j$ does not hold, since otherwise $\Gamma_{G_1}(s) \cap X_i \neq \emptyset$ and $\Gamma_{G_1}(s) \cap (X_j - X_i) \neq \emptyset$ imply $c_{G_1}(X_j, s) \geq c_{G_1}(X_i, s) + 1 \geq 2$, contradicting that X_j is of type (2). Assume that X_i and X_j cross each other in G_1. Then, by $\Gamma_{G_1}(s) \cap (X_j - X_i) \neq \emptyset$, Claim 4.5 implies that $X_j - X_i$ is a critical cut of type (1). This implies that no vertex in X_j belongs to N_2, contradicting $X_j \in \mathcal{X}_2$. $\Box$

Before proceeding to Step II, let us consider the time complexity of Step I for constructing G_1 from G. Let G' be the multigraph obtained from G by adding a vertex s and $\max\{r_\lambda(x, y) \mid x, y \in V\}$ edges between s and every vertex $v \in V$. For each $v \in V$, we perform the following procedure sequentially.

1. Delete all edges in $E_{G'}(s, v)$ and denote the resulting multigraph again as G' (i.e., $G' := G' - E_{G'}(s, v)$). Compute $\lambda_{G'}(x, y)$ for all $x, y \in V$.

2. Add to G' $\max_{x,y \in V}[\{r_\lambda(x, y) - \lambda_{G'}(x, y)\}, 0]$ edges between s and v, and denote the resulting multigraph as G'.

3. If there is a cut vertex $w \neq s$ in the current G' (hence $c_{G'}(s, v) = 0$), then add one edge (s, v) to G'.

By Step 2, the current G' satisfies (3.1). In Step 3, whenever there is a cut vertex $w \neq s$, adding one edge (s, v) preserves (3.2), because this w becomes a cut vertex by $c_{G'}(s, v) = 0$. So the resulting G' satisfies (3.1) and (3.2), and it is easy to see that the removal of any edge in $E_{G'}(s, v)$ of G' violates (3.1) or (3.2). Step 1 for a vertex v can be carried out in $O(n^2 m' \log(n^2/m'))$ time by using a Gomory-Hu tree [9] which is constructed by $n-1$ maximum flow computations [8], where $m' = m(G') = O(m(G) + n(G))$ (by a Gomory-Hu tree, we can obtain the values $\lambda_{G'}(x, y)$ for all $x, y \in V$). Steps 2 and 3 can be executed in $O(m(G) + n(G))$ time. Therefore, the entire computation for all $v \in V$ can be done in $O(n^3(m + n) \log(n^2/(m + n)))$ time. $\Box$

5 Step II

Let $G_1 = (V \cup \{s\}, E \cup F_1)$ be the multigraph obtained from $G = (V, E)$ in Step I. Step II computes the multigraph G_2 by applying a complete splitting at s in G_1, which preserves the r_λ-edge-connectivity. As noted in Section 2.3, such a complete splitting can be computed in $O(n^3 m' \log(n^2/m'))$ time, where $m' = m(G_1) = O(m(G) + n(G))$, and the number of new pairs of adjacent vertices in G_2 is $O(n)$. Thus, the number $m(G_2)$ of pairs of adjacent vertices in G_2 is $O(m(G)+n(G))$.

Remark 5.1: If $c_{G_1}(s)$ is odd in the beginning of Step II, a "non cut vertex" $\hat{w}$ of G is chosen to add an extra edge $\hat{e}=(s,\hat{w})$ to G_1. The choice of this $\hat{w}$ will play an important role in proving the correctness of Step IV.

6 Step III

Let $G_2=(V,E\cup F_2)$ be the multigraph obtained in Step II. This G_2 is r_λ-edge-connected but has cut vertices. The correctness of Step III follows from Properties 3.2 and 3.3.

Proof of Property 3.2: This property follows since $T_1\cap T_2=\emptyset$ holds and $G_2[T_1\cup\{v_1\}]-e_1$ is connected by $\lambda(G_2)\geq 2$. See [15] for the details. □

To show Property 3.3, we use the next claim, which can be easily seen from the definition of the admissibility.

Claim 6.1 *Let $v\in V$ be a cut vertex in G_2. Assume that a v-component T contains an admissible edge $e=(u,u')$ with respect to v. Then $G_2[T]-e$ contains a path P between u and u'.* □

As defined in the statement of Property 3.3, given a cut vertex v, let $e_1=(u_1,w_1)\in F_2$ be an edge in a v-component T_1 of G_2, admissible with respect to v, T_2 be another v-component in G_2 such that $e_1\notin E(G_2[T_2\cup\{v\}])$, and $e_2=(u_2,w_2)\in F_2$ be an edge in $G_2[T_2\cup\{v\}]$. Then $G'_2=(V,E\cup F'_2)$ denotes the multigraph obtained from G_2 by switching e_1 and e_2, where $F'_2=F_2\cup\{(u_1,u_2),(w_1,w_2)\}-\{e_1,e_2\}$.

Proof of Property 3.3(i): We show that $\lambda_{G'_2}(x,y)\geq r_\lambda(x,y)$ holds for all $x,y\in V$. Assume otherwise; i.e., there is a cut X such that $c_{G'_2}(X)\leq r_\lambda(X)-1$ holds. Note that $c_{G'_2}(X)\geq c_{G_2}(X)$ holds if cut X does not separate $\{u_1,u_2\}$ and $\{w_1,w_2\}$ in G'_2. Therefore assume that X separates $\{u_1,u_2\}$ and $\{w_1,w_2\}$, and hence $c_{G'_2}(X)=c_{G_2}(X)-2$, i.e., $c_{G_2}(X)\leq r_\lambda(X)+1$. Since at least one of the cuts X and $V-X$ crosses each of the v-components T_1 and T_2, the subgraph $G_2[X]$ or $G_2[V-X]$ (the one not containing v) is not connected. Without loss of generality, assume that $G_2[X]$ is not connected, i.e., two cuts X_i, $i=1,2$, satisfy $X_1\cap X_2=\emptyset$, $X_1\cup X_2=X$, and $E_{G_2}(X_1,X_2)=\emptyset$. The r_λ-edge-connectivity of G_2 implies $c_{G_2}(X_i)\geq r_\lambda(X_i)\geq 2$ for $i=1,2$. Furthermore, at least one of $c_{G_2}(X_1)\geq r_\lambda(X)$ and $c_{G_2}(X_2)\geq r_\lambda(X)$ holds, because there is a vertex pair x and y such that $x\in X$, $y\in V-X$ and $r_\lambda(x,y)=r_\lambda(X)$. Therefore $c_{G_2}(X)=c_{G_2}(X_1)+c_{G_2}(X_2)\geq r_\lambda(X)+2$, contradicting the assumption $c_{G_2}(X)\leq r_\lambda(X)+1$. □

Proof of Property 3.3(ii): We show that $p(G'_2-v)<p(G_2-v)$ holds. It suffices to show that $G'_2[T_1\cup T_2]$ is connected. This follows from Claim 6.1, $\lambda(G_2)\geq 2$ and $E_{G'_2}(T_1,T_2)\neq\emptyset$ (see [15] for the details). □

Proof of Property 3.3(iii): We show that $\kappa_{G'_2}(x,y) \geq 2$ holds if $\kappa_{G_2}(x,y) \geq 2$. Assume that there are vertices $x, y \in V$ such that $\kappa_{G_2}(x,y) = 2$ but $\kappa_{G'_2}(x,y) = 1$. Let $v' \in V$ denote a cut vertex in G'_2 that disconnects x and y. Clearly, $v' \neq v$ (because v is a cut vertex in G_2 and $v = v'$ would imply $\kappa_{G_2}(x,y) = 1$). Let $W_1, W_2, \ldots, W_q$ ($q \geq 2$) be the v'-components of G'_2, where $x \in W_1$ and $y \in W_2$. In this case, $G_2[W_1 \cup W_2 \cup \{v'\}]$ contains all the end vertices u_1, w_1, u_2, and w_2 (otherwise, switching e_1 and e_2 would join some other two W_i, W_j, contradicting the definition of W_1, W_2). Since the cut vertex v' does not disconnect x and y in G_2, $e_1 \in E_{G_2}(W_1, W_2)$ or $e_2 \in E_{G_2}(W_1, W_2)$ holds. Also no edge other than e_1 and e_2 belongs to $E_{G_2}(W_1, W_2)$. We show that $u_i, w_i \in W_j$ cannot hold for any i, j with $1 \leq i, j \leq 2$. For this, assume $u_1, w_1 \in W_1$ without loss of generality. Then $e_2 = (u_2, w_2) \in E_{G_2}(W_1, W_2)$ holds, where we assume $u_2 \in W_1$ and $w_2 \in W_2$ without loss of generality. Hence $(w_1, w_2) \in E_{G'_2}(W_1, W_2)$ would hold, contradicting that W_1 and W_2 are v'-components in G'_2. Therefore, for each $i = 1, 2$, we see that $e_i = (u_i, w_i) \in E_{G_2}(W_1, W_2)$ or $v' \in \{u_i, w_i\}$.

We first consider the case $e_1 \in E_{G_2}(W_1, W_2)$. Thus $v' \in T_1$ holds since $G_2[T_1] - e_1$ is connected by Claim 6.1. Since $v' \in T_1$ implies $u_2 \neq v' \neq w_2$, we have $e_2 \in E_{G_2}(W_1, W_2)$; assume $u_1 \in T_1 \cap W_1$, $w_1 \in T_1 \cap W_2$ $u_2 \in T_2 \cap W_1$, $w_2 \in T_2 \cap W_2$ and $v \in W_1$ without loss of generality. Since the cut vertex $v \in W_1$ disconnects $w_1 \in T_1$ and $w_2 \in T_2$ in G_2, $c_{G_2}(T_2 \cap W_2, W_2 \cup \{v'\} - T_2) = 0$ holds. From this and $E_{G_2}(W_1, W_2) = \{e_1, e_2\}$, $c_{G_2}(T_2 \cap W_2) = |\{e_2\}| = 1$ holds, contradicting the r_λ-edge-connectivity of G_2.

We then consider the case $e_1 \notin E_{G_2}(W_1, W_2)$ (i.e., $v' = u_1 \in T_1$ or $v' = w_1 \in T_1$). In this case, we see that $e_2 \in E_{G_2}(W_1, W_2)$ and $v' \notin T_2$. This also leads to a contradiction, analogously to the above case. $\square$

Proof of Property 3.3(iv): We show that (3.3) holds in G'_2. If $G'_2[T \cup \{v\}]$ contains no edge in F'_2 for some cut vertex v and its v-component T, then T contains no end vertex of an edge in F'_2 in G'_2. This implies that $\Gamma_{G_1}(s) \cap T = \emptyset$ holds in Step I, i.e., $|\Gamma_G(T)| \leq 1$ but $c_{G_1}(s, T) = 0$, contradicting (3.2) in G_1. $\square$

Now we evaluate the time complexity of Step III. We can check whether G_2 has more than one cut vertex in linear time. If this is the case, for an admissible edge $e_1 \in F_2$ in $G_2[T_1 \cup \{v_1\}]$, an edge e_2 in Property 3.3 can be found in linear time by computing all biconnected components of G_2 [30]. Also, if G_2 has exactly one cut vertex v, then we can find a pair of edges e_1 and e_2 in Property 3.3 in linear time by computing all bridges in $G_2[V - v]$. By switching such a pair of edges in F_2, Property 3.3(ii) and (iii) tell that the number of v-components decreases at least by one, and for other cut vertices v' in G_2, the number of v'-components does not increase. Note that the total number of v-components over all cut vertices v in G_2 is at most $2n$. Therefore, the number of switching executed in Step III is $O(n)$. This also implies that the number of new pairs of adjacent vertices created by the switchings is $O(n)$. Thus, the number of pairs of adjacent vertices in G_3, $m(G_3)$, is $O(m + n)$. Therefore Step III can be performed in $O(n \cdot m(G_3)) = O(nm)$ time.

7 Step IV

Let $G_3 = (V, E \cup F_3)$ be the multigraph obtained after Step III, where $|F_3| = \lceil \hat{\alpha}(G)/2 \rceil$; G_3 is r_λ-edge-connected, has exactly one cut vertex v and satisfies (3.3). The correctness of Step IV follows from Property 3.4, which we now prove via two claims. Let $\mathcal{X}$ be the set of disjoint critical cuts of Claim 4.6 for G_1. Recall that one edge $\hat{e} = (s, \hat{w})$ is added to $|F_1|$ at the beginning of Step II if $c_{G_1}(s)$ is odd in Step II, where $\hat{w}$ is chosen to be a non-cut vertex in G. We call this edge $\hat{e} = (s, \hat{w})$ *additional.*

Claim 7.1 *For each critical cut $X \in \mathcal{X}$ of type (1) in G_1, the induced multigraph $G_3[X]$ contains no edge in F_3.*

Proof: It follows from $\lambda(G_3) \geq 2$ (see [15] for the details). □

Claim 7.2 *F_3 contains no edge incident to the cut vertex v in G_3.*

Proof: We assume that G_3 has an edge $e = (v, v') \in F_3$ incident to the cut vertex v. Note that Claim 4.6 implies that v is contained in a critical cut in G_1, except for the case that $v = \hat{w}$, (where $\hat{e} = (s, \hat{w})$ is an additional edge).

Case-1: v is contained in a critical cut X of type (1) in G_1. Then $v, v' \in \Gamma_{G_1}(s)$ holds by $e = (v, v') \in F_3$. We first show that both $G_3[X]$ and $G_3[V - X]$ are connected. Since $G[X]$ is connected by Claim 4.1, $G_3[X]$ is also connected. Assume that $G_3[V - X]$ has two disjoint vertex sets V_1 and V_2 such that $V_1 \cup V_2 = V - X$ and $c_{G_3}(V_1, V_2) = 0$. In this case, $c_{G_3}(V_i, X) \geq 2$ holds for $i = 1, 2$ since $\lambda(G_3) \geq 2$. There are vertices $x^* \in X$ and $y^* \in V - X$ with $r_\lambda(x^*, y^*) = r_\lambda(X) \geq c_{G_3}(X) - 1$ (note that $r_\lambda(X) + 1 = c_{G_3}(X)$ holds if $\hat{w} \in X$). Assume $y^* \in V_1$ without loss of generality. Then we have $c_{G_3}(V_1) = c_{G_3}(V - X) - c_{G_3}(V_2) \leq c_{G_3}(V - X) - 2 = c_{G_3}(X) - 2 \leq r_\lambda(x^*, y^*) - 1 \leq r_\lambda(V_1) - 1$, contradicting the r_λ-edge-connectivity of G_3. Therefore $G_3[V - X]$ is also connected.

Claim 7.1 and $v \in X$ imply $v' \in V - X$. Let T' be the v-component that contains v'. Since $G_3[X]$ and $G_3[V - X]$ are both connected, we see that $G_3[X]$ contains all the v-components except for T'. Now for each v-component $T \subset X$, there is another edge $(t, t') \in F_3$ with $t \in T \subset X$ by (3.3) (note that F_3 is the final F_2 in Step II and hence satisfies (3.3)). By Claim 7.1, the other end vertex t' is not in X, i.e., $t' \in V - X = T' - X$. Such edge (t, t') connects two v-components T and T' in G_3, contradicting that v is a cut vertex of G_3.

Case-2: v is contained in a critical cut X of type (2) in G_1. Let $\{x\} = \Gamma_G(X)$ and $V' \subseteq V$ be the component of G that contains v. Thus, X is an x-component of G (where x is not necessarily a cut vertex in G). Note that no other edge than e in F_3 is incident to v except for the case that $v = \hat{w}$.

(i) The case where e is the only edge in F_3 which is incident to v in G_3. We see that v is not a cut vertex in G (this can occur if e connects two components in G). If v is a cut vertex in G, then we have $v \neq \hat{w}$ and Lemma 2.1 implies that there is a v-component $X_v \subseteq X - v$ of G, which contradicts that the critical cut X of type (2) has no neighbor of s in G_1 other than v (see

[15] for the details). Therefore $G[V'-v]$ is connected. Thus $G_3[V'-v]$ is connected and hence $V'-v$ is contained in a v-component T_1 of G_3. Hence for any other v-component T_2 of G_3, T_2 is contained in a component of G other than V' and therefore $E_{G_3}(T_2) = E_{G_3}(T_2, v) \subseteq F_3$ holds. By $\lambda(G_3) \geq 2$, $E_{G_3}(T_2)$ must contain at least two edges in F_3, a contradiction.

(ii) The case where at least one more edge (say, $e' \neq e$) of F_3 is incident to v. In this case, at least two edges in F_3 is incident to v in G_3. However, since $c_{G_1}(s, X) = 1$ holds in G_1 (since X is of type (2)), it means that the additional edge $(s, \hat{w})$ with $v = \hat{w}$ has been chosen in Step II. Thus, $E_{G_3}(v) \cap F_3 = \{e, e'\}$ and $v = \hat{w}$. Since $\hat{w} = v$ is not a cut vertex in G, $G[V'-v]$ (and hence $G_3[V'-v]$) is connected, and $V'-v$ is contained in a v-component T_1 in G_3. From this, any other v-component T_2 satisfies $E_{G_3}(T_2) = E_{G_3}(T_2, v) \subseteq F_3$. By $\lambda(G_3) \geq 2$, $E_{G_3}(T_2)$ must contain at least two edges in F_3. Thus, by $E_{G_3}(v) \cap F_3 = \{e, e'\}$, we see that there are exactly two v-components T_1 and T_2 in G_3 and $E_{G_3}(v) \cap F_3 = \{e, e'\} = E_{G_3}(T_2)$. By (3.3), $G_3[T_1 \cup \{v\}]$ contains at least one edge $e^* = (u^*, w^*) \in F_3$. From $E_{G_3}(v) \cap F_3 = E_{G_3}(T_2)$, this e^* is not incident to v. Then we see that $e^* \in F_3$ is admissible with respect to v in G_3 since otherwise $\lambda(G[V-v']) > 0$ implies that there is another component $V''(\subseteq T_1 - V')$ of G with $E_{G_3}(V'', T_1 - V'') = \{e^*\}$, which contradicts $\lambda(G_3) \geq 2$ (see [15] for the details). This contradicts the assumption of G_3.

Case-3: The remaining case (i.e., v is contained in no critical cut $X \in \mathcal{X}$ and $v = \hat{w}$ holds). Let $V' \subseteq V$ be the component of G that contains v. Clearly, no other edge $e' \neq e$ in F_3 is incident to v, and $G[V'-v]$ (and hence $G_3[V'-v]$) is connected since $v = \hat{w}$ is not a cut vertex of G. Thus, $V'-v$ is contained in a v-component T_1, and $E_{G_3}(v, T_2) = \{e\}$ holds for another v-component T_2 in G_3. This T_2 satisfies $c_{G_3}(T_2) = 1$, contradicting $\lambda(G_3) \geq 2$. □

Proof of Property 3.4: Since $|F_3| = \lceil \hat{\alpha}(G)/2 \rceil$ holds from construction, it suffices to show $p(G-v) = p(G_3 - v) + |F_3|$. If $p(G-v) < p(G_3-v) + |F_3|$, then there is at least one edge $e \in F_3$ such that $p((G_3 - v) - e) = p(G_3 - v)$. Thus e is admissible with respect to v, since no edge in F_3 is incident to v by Claim 7.2. This contradicts the construction of G_3 (since this implies that Step III has not finished yet). Therefore $p((G_3 - v) - e) = p(G_3 - v) + 1$ for all edges $e \in F_3$. This leads to $p(G-v) = p(G_3 - v) + |F_3|$. □

Clearly, Step IV can be executed in linear time since computing all biconnected components of G_3 can be done in linear time [30].

As a result of proofs in Sections 4 – 7, the correctness of algorithm EV-AUGMENT has been proved. By summing up the running time of all steps, we conclude that the total time complexity of algorithm EV-AUGMENT is $O(n^3(m+n)\log(n^2/(m+n)))$. This proves Theorem 3.1.

Before concluding this section, we remark that in the special case of the uniform requirement $r_\lambda(x,y) = \ell$ for all $x, y \in V$, the complexity can be improved by a factor of n^2. By results of [28, 29], we observe the following theorem. See [15] for the proof.

Theorem 7.1 *Problem EVAP$(\ell, 2)$ can be solved by algorithm EV-AUGMENT in $O((nm + n^2 \log n)\log n)$ time.* □

8 Preserving Simplicity

In this section, we consider another variant of the augmentation problem: Given a *simple* graph $G = (V, E)$ and requirement functions r_λ and r_κ, find a smallest set F of new edges such that $G' = (V, E \cup F)$ remains simple and becomes (r_λ, r_κ)-connected. This problem is called the *simplicity preserving edge-and-vertex-connectivity augmentation problem*, and is denoted by SEVAP(r_λ, r_κ).

The problem SEVAP$(r_\lambda, 0)$ was first posed in [5] as an important open problem, and recently Jordán [22] proved that SEVAP$(\ell, 0)$ (i.e., $r_\lambda(x, y) = \ell$ for all $x, y \in V$) is NP-hard for a general ℓ even if the input simple graph G is assumed to be $(\ell - 1)$-edge-connected. On the other hand, Bang-Jensen and Jordán [1] showed that SEVAP$(r_\lambda, 0)$ can be solved in polynomial time if $\ell^* = \max\{r_\lambda(x, y) \mid x, y \in V\}$ is considered to be a fixed constant. They proved the next result, which plays a key role in their algorithm.

Lemma 8.1 [1] *Let $G' = (V \cup s, E')$ be a multigraph such that $G' - s$ is simple and r_λ-edge-connected, where $r_\lambda : \binom{V}{2} \to Z^+$ is a given function. Then there are polynomial functions $f(\ell^*)$ and $g(\ell^*)$ of $\ell^* = \max\{r_\lambda(x, y) \mid x, y \in V\}$ satisfying the following properties.*

(i) *If $c_{G'}(s) \geq f(\ell^*)$, then there is a complete splitting such that the resulting multigraph (ignoring s) is simple and r_λ-edge-connected. Moreover, such a complete splitting can be obtained in polynomial time.*

(ii) *If $c_{G'}(s) \leq f(\ell^*)$, then $G' - s$ can be augmented by at most $g(\ell^*)$ new edges so that the resulting multigraph becomes simple and r_λ-edge-connected.* □

For a uniform requirement $r_\lambda = \ell$, $f(\ell) = 3\ell^4$ and $g(\ell) = 3\ell^4/2 + 2\ell^2 + 1$ are shown in [1].

In this section, we show that algorithm EV-AUGMENT in Section 3 can be modified to exploit Lemma 8.1 so as to solve SEVAP$(r_\lambda, 2)$.

Theorem 8.1 *Given a simple graph $G = (V, E)$ and a function $r_\lambda : \binom{V}{2} \to Z^+$,* SEVAP$(r_\lambda, 2)$ *can be solved in polynomial time for a fixed $\ell^* = \max\{r_\lambda(x, y) \mid x, y \in V\}$.* □

To solve SEVAP$(r_\lambda, 2)$ we modify Steps II and III of algorithm EV-AUGMENT in order to maintain the simplicity of the graph.

Algorithm EV-AUGMENT$'$

Step I$'$: The same as Step I of EV-AUGMENT, which computes a multigraph G_1 for a given simple graph G.

Step II$'$: We distinguish the following two cases, where $\ell^* = \max\{r_\lambda(x, y) \mid x, y \in V\}$.

(1) $c_{G_1}(s) \geq f(\ell^*)$. We perform a complete splitting at s that preserves the simplicity of $G_1 - s$ as well as the r_λ-edge-connectivity of G_1. Such a splitting can be obtained in polynomial time for a fixed ℓ^* by Lemma 8.1(i). Then we proceed to Step III$'$.

(2) $c_{G_1}(s) < f(\ell^*)$. By Lemma 8.1(ii), we can make $G_1 - s$ (i.e., the input simple graph G) r_λ-edge-connected by adding a set F_1' of at most $g(\ell^*)$ new edges, while preserving simplicity. Let $G_2' = (G_1 - s) + F_1'$. Note that $t(G_2') \leq f(\ell^*)$ holds, because any minimal tight set in G_2' has at least one neighbor of s in G_1, by the construction of G_1. This tells that the input simple graph G_2' becomes 2-vertex-connected by adding a set F_2' of at most $f(\ell^*) - 1$ new edges while preserving simplicity. The resulting $(r_\lambda, 2)$-connected graph $G_2' + F_2'$ is obviously simple. Now, since the size $|F|$ of an optimal solution F is at most $g(\ell^*) + f(\ell^*) - 1$, such F can be found by inspecting all possible choices of subsets $F \subseteq V \times V - E$ with $|F| \leq g(\ell^*) + f(\ell^*) - 1$. This can be done in polynomial time for a fixed ℓ^*. Halt.

Step III$'$: As in Step III of EV-AUGMENT, we try to continue switching edges $e_1 = (u_1, w_1)$, $e_2 = (u_2, w_2) \in F_2$, which satisfy the assumption of Property 3.3, until there is no pair of such edges. However, to keep the multigraph resulting from a switching simple, the edges to be switched are carefully chosen.

(1) $u_1 \notin \Gamma_{G_2}(u_2)$ and $w_1 \notin \Gamma_{G_2}(w_2)$ holds, or $u_1 \notin \Gamma_{G_2}(w_2)$ and $w_1 \notin \Gamma_{G_2}(u_2)$. Then we switch from e_1, e_2 to $(u_1, u_2), (w_1, w_2)$ in the former case (resp., to $(u_1, w_2), (u_2, w_1)$ in the latter case). Notice that Property 3.3 says that the two switchings from e_1, e_2 to $(u_1, u_2), (w_1, w_2)$ and from e_1, e_2 to $(u_1, w_2), (u_2, w_1)$ both satisfy conditions (i) – (iv) in Property 3.3. Clearly, the resulting graph is also simple. Note that if G_2 has at least two cut vertices, then we can find such pairs $e_1, e_2 \in F_2$.

(2) Otherwise e_1 or e_2 (say, e_1) is incident to v. Let $e_1 = (u_1, v)$ and T_1 be a v-component containing u_1. We choose an arbitrary vertex w in a v-component T_2 different from T_1, replace the edge $e_1 = (u_1, v)$ with a new edge $e' = (u_1, w)$, and update F_2 by $(F_2 - \{e_1\}) \cup \{e'\}$. Clearly, the resulting multigraph G_2' remains simple. As will be shown as Property 8.1 below, G_2' still satisfies all conditions (i)–(iv) of Property 3.3.

We repeat these operations (1) or (2) as long as possible. If this leads to the graph G_2' without a cut vertex, we are done; output the resulting F_2 as an optimal solution and halt. Otherwise we proceed Step IV$'$.

Property 8.1 *Given a cut vertex v in $G_2 = (V, E \cup F_2)$, assume that there is an edge $e = (u, v) \in F_2$ incident to v. Let T_1 be the v-component containing u. Then for any vertex $w \in V - (T_1 \cup \{v\})$, the multigraph G_2' obtained by changing $e = (u, v)$ to $e' = (u, w)$ satisfies the conditions* (i)–(iv) *of Property 3.3.*

Proof: The proof is similar to that of Property 3.3. See [15] for the details. □

Step IV$'$: We see that the resulting graph G_3 has exactly one cut vertex $v \in V$, no edge $e \in F_2$ incident to v and no edge $e \in F_2$ admissible with respect to v. We add to G_3 another set F_4 of new edges which is computed in Step IV of EV-AUGMENT. Adding F_4 preserves simplicity of G_3, because for each edge $(x_i, x_{i+1}) \in F_4$, x_i and x_{i+1} belong to different

v-components of G_3. Thus, as in Step IV, we conclude that $F = F_3 \cup F_4$ is an optimal solution, which attains $|F_3 \cup F_4| = \hat{\beta}(G)$.

Clearly, all steps in the above algorithm EV-AUGMENT′ can be executed in polynomial time (for a fixed ℓ^*). Summarizing the argument given so far, Theorem 8.1 is now established. $\square$

9 Conclusion

We considered in this paper the problem of augmenting a multigraph $G = (V, E)$ with the smallest number of new edges so as to make G (r_λ,2)-connected for a general requirement function $r_\lambda : \binom{V}{2} \to Z^+$. To solve this, we introduced a lower bound on the number of new edges, and developed an edge-switching operation that preserves the edge-connectivity and vertex-biconnectivity. The resulting algorithm runs in $O(n^3(m+n)\log(n^2/(m+n)))$ time if r_λ is general, and in $O((nm + n^2 \log n)\log n)$ time if $r_\lambda(x, y) = \ell$ holds for all $x, y \in V$. It was further shown that the problem that augments a simple graph while preserving the simplicity of the graph can be solved in polynomial time for any fixed $\ell^* = \max\{r_\lambda(x, y) \mid x, y \in V\}$.

Recently, we generalized the above approach to the following problem: given an arbitrary multigraph $G = (V, E)$ and an integer $\ell \geq 3$, find the smallest number of new edges to make G $(\ell, 3)$-connected. The result in [16, 18] says that this problem can be solved in polynomial time for any fixed ℓ. Moreover, given a $(k-1)$-vertex-connected multigraph $G = (V, E)$ and two integers ℓ and k with $\ell \geq k \geq 4$, we showed that G can be made (ℓ, k)-connected by adding new edges whose size is $O(\ell)$ over the optimum [17].

Acknowledgments

This research was partially supported by the Scientific Grant-in-Aid from Ministry of Education, Science, Sports and Culture of Japan, and the subsidy from the Inamori Foundation.

References

[1] J. Bang-Jensen and T. Jordán, *Edge-connectivity augmentation preserving simplicity*, SIAM J. Discrete Math., Vol.11, 1998, pp. 603–623.

[2] G.-R. Cai and Y.-G. Sun, *The minimum augmentation of any graph to k-edge-connected graph*, Networks, Vol.19, 1989, pp. 151–172.

[3] K. P. Eswaran and R. E. Tarjan, *Augmentation problems*, SIAM J. Comput., Vol.5, 1976, pp. 653–665.

[4] A. Frank, *Augmenting graphs to meet edge-connectivity requirements*, SIAM J. Discrete Math., Vol.5, 1992, pp. 25–53.

[5] A. Frank, *Connectivity augmentation problems in network design*, in: Mathematical Programming: State of the Art 1994, J.R. Birge and K.G. Murty (Eds.), The University of Michigan, Ann Arbor, MI, 1994, pp. 34–63.

[6] H. N. Gabow, *Applications of a poset representation to edge connectivity and graph rigidity*, Proc. 32nd IEEE Symp. Found. Comp. Sci., 1991, pp. 812–821.

[7] H. N. Gabow, *Efficient splitting off algorithms for graphs*, Proc. 26th ACM Symp. Theory of Computing, 1994, pp. 696–705.

[8] A. V. Goldberg and R. E. Tarjan, *A new approach to the maximum flow problem*, J. Assoc. Comput. Mach., Vol.35, 1988, pp. 921–940.

[9] R. Gomory and T. Hu, *Multi-terminal network flows*, SIAM J. Applied Mathematics, Vol.9, 1961, pp. 551–570.

[10] T. Hsu, *On four-connecting a triconnected graph*, Proc. 33nd IEEE Symp. Found. Comp. Sci., 1992, pp. 70–79.

[11] T. Hsu, *Undirected vertex-connectivity structure and smallest four-vertex-connectivity augmentation*, Lecture Notes in Comput. Sci., 1004, Springer-Verlag, Algorithms and Computation (Proc. ISAAC '95), 1995, pp. 274–283.

[12] T. Hsu and M. -Y. Kao, *Optimal bi-level augmentation for selectively enhancing graph connectivity with applications*, Lecture Notes in Comput. Sci., 1090, Springer-Verlag, Computing and Combinatorics (Proc. COCOON '96), 1996, pp. 169–178.

[13] T. Hsu and V. Ramachandran, *A linear time algorithm for triconnectivity augmentation*, Proc. 32nd IEEE Symp. Found. Comp. Sci., 1991, pp. 548–559.

[14] T. Hsu and V. Ramachandran, *Finding a smallest augmentation to biconnect a graph*, SIAM J. Comput., Vol.22, 1993, pp. 889–912.

[15] T. Ishii, H. Nagamochi and T. Ibaraki, *Multigraph augmentation under biconnectivity and general edge-connectivity requirements*, Technical Report #98008, Kyoto University, Department of Applied Mathematics and Physics, 1998.

[16] T. Ishii, H. Nagamochi and T. Ibaraki, *k-edge and 3-vertex connectivity augmentation in an arbitrary multigraph*, Lecture Notes in Comput. Sci., 1533, Springer-Verlag, Algorithms and Computation (Proc. ISAAC '98), 1998, pp. 159–168.

[17] T. Ishii, H. Nagamochi and T. Ibaraki, *Augmenting a $(k-1)$-vertex-connected multigraph to an ℓ-edge-connected and k-vertex-connected multigraph*, Lecture Notes in Comput. Sci., 1643, Springer-Verlag, Algorithms (Proc. ESA '99), 1999, pp. 414–425.

[18] T. Ishii, H. Nagamochi and T. Ibaraki, *Optimal augmentation of a 2-vertex-connected multigraph to a k-edge-connected and 3-vertex-connected multigraph*, J. Combinatorial Optimization, Vol.4, 2000, pp. 35–78.

[19] T. Jordán, *Connectivity augmentation problem in graphs*, PhD thesis, Dept. of Computer Science, Eötvös University, Budapest, Hungary, 1994.

[20] T. Jordán, *On the optimal vertex-connectivity augmentation*, J. Combin. Theory Ser. B, Vol.63, 1995, pp. 8–20.

[21] T. Jordán, *A note on the vertex-connectivity augmentation problem*, J. Combin. Theory Ser. B, Vol.71, 1997, pp. 294–301.

[22] T. Jordán, *Two NP-complete augmentation problems*, Preprint no. 8, Department of Mathematics and Computer Science, Odense University, 1997.

[23] G. Kant, *Algorithms for drawing planar graphs*, PhD thesis, Dept. of Computer Science, Utrecht University, the Netherlands, 1993.

[24] G. Kant, *Augmenting outerplanar graphs*, J. Algorithms, Vol.21, 1996, pp. 1–25.

[25] M. -Y. Kao, *Data security equals graph connectivity*, SIAM J. Discrete Math., Vol.9, 1996, pp. 87–100.

[26] L. Lovász, Combinatorial Problems and Exercises, North-Holland, 1979.

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

[28] H. Nagamochi and T. Ibaraki, *Deterministic* $\tilde{O}(nm)$ *time edge-splitting in undirected graphs*, J. Combinatorial Optimization, Vol.1, 1997, pp. 5–46.

[29] H. Nagamochi, S. Nakamura and T. Ibaraki, *A simplified* $\tilde{O}(nm)$ *time edge-splitting algorithm in undirected graphs*, Algorithmica, Vol.26, 2000, pp. 50–67.

[30] R. E. Tarjan, *Depth-first search and linear graph algorithms*, SIAM J. Comput., Vol.1, 1972, pp. 146–160.

[31] T. Watanabe and A. Nakamura, *Edge-connectivity augmentation problems*, J. Comput. System Sci., Vol.35, 1987, pp. 96–144.

[32] T. Watanabe and A. Nakamura, *A minimum 3-connectivity augmentation of a graph*, J. Comput. System Sci., Vol.46, 1993, pp. 91–128.