Journal of the Operations Research
Society of Japan
2004, Vol. 47, No. 4, 224-243



# NA-EDGE-CONNECTIVITY AUGMENTATION PROBLEMS BY ADDING EDGES

Hiroyoshi Miwa
*Kwansei Gakuin University*

Hiro Ito
*Kyoto University*

(Received September 30, 2003; Revised September 10, 2004)

*Abstract* The network reliability in multi-server environment is measured by the connectivity between a vertex and a vertex subset (NA-connectivity). The problem of augmenting a graph by adding the smallest number of new edges to meet NA-edge(vertex)-connectivity requirement is an important optimization problem that contributes to the network design problem to increase the reliability of a current network by adding the smallest number of links. This problem is a generalization of the well-known connectivity augmentation problems.

In this paper, we focus on the NA-edge-connectivity augmentation problem. First, we prove the NP-completeness of the problem which determines whether we can augment a graph to a 1-NA-edge-connected graph by adding a given number or less new edges. Next, we prove that the problem of augmenting a 1-NA-edge-connected graph or a 0-NA-edge-connected graph to be 2-NA-edge-connected graph by adding the smallest number of edges can be solved in polynomial time.

**Keywords**: Combinatorial optimization, graph theory, algorithm, connectivity augmentation problem

## 1. Introduction

The network reliability of a communication network has been measured by its connectivity, because a reliable network must ensure a route between any two nodes even if some links or nodes fail. In recent communication networks, many WWW (World Wide Web) sites offers their services by using some mirror servers which have the same contents as an original server. In such a multi-server environment, it is more practical and important for a reliable network to ensure some independent routes between a node and the set of nodes in which an original and its mirror servers are located. From this point of view, the NA-edge(resp., vertex)-connectivity between a vertex and a vertex subset was proposed in [7, 8], which is an extended measure of the edge(resp., vertex)-connectivity and it is equal to the number of edge(resp., vertex)-independent paths between the vertex and the vertex subset [7, 8]. When a graph and a family of its vertex subsets (areas) are given, if the minimum of the NA-edge(resp., vertex)-connectivity for all pairs of a vertex and an area is $k$, the graph is called $k$-NA-edge(resp., vertex)-connected. Some optimization problems based on the concept of the NA-connectivity have been extensively studied. For example, the problem of locating areas with NA-connectivity requirement (e.g. [9]) and the problem of determining a spanning subgraph of a given NA-connectivity with the smallest cost (e.g. [11]) are closely related to the practical network design problems.

In this paper, we investigate the problem of augmenting a graph by adding the smallest number of new edges with NA-edge-connectivity requirement. Our contribution is the following two results:

1. The problem of determining whether we can augment a graph to a 1-NA-edge-connected graph by adding a given number or less new edges is NP-complete.
2. The problem of augmenting a 1 or 0 -NA-edge-connected graph to be a 2-NA-edge-connected graph by adding the smallest number of edges can be solved in $O(pn + m)$ time, where $n$, $m$, and $p$ are the number of vertices, edges, and areas.

From the similar motivation, the augmentation problems to meet edge(resp., vertex)-connectivity requirement by adding the smallest number of new edges are closely related to the NA-edge(resp., vertex)-augmentation problem, and they have been extensively studied as an important subject, and many algorithms have been developed so far.

As to the edge-connectivity augmentation problem to determine whether a graph can be a $\lambda$-edge-connected graph by adding $b$ or less new edges, where $\lambda$ and $b$ are positive integers, it can be solved in polynomial time (e.g. [13–16]).

Our results are in contrast to these known results. Indeed, the problem of augmenting a graph to a 1-NA-edge-connected graph is NP-complete, although the edge-connectivity augmentation problem can be always solved in polynomial time.

As to the vertex-connectivity augmentation problem defined analogously, the problem augmenting a graph to a $\kappa$-vertex-connected graph can be solved in polynomial time for $\kappa = 2$ [1, 5], for $\kappa = 3$ [5, 17], and $\kappa = 4$ [4].

The NA-edge(vertex)-connectivity augmentation problem has been also investigated, since this problem was proposed and studied for the first time in [12]. In the report, for the problem of augmenting a graph to a $\lambda$-NA-edge-connected graph, we proved that the problem for $\lambda = 1$ is NP-complete and that the problem for $\lambda = 2$ can be solved in polynomial time. In [6], it was proved based on the edge-splitting operation [2, 14] that the problem for $\lambda \geq 3$ can be solved in polynomial time ($O(m + n(\lambda^3 + n^2)(p + \lambda n + n \log n) \log \lambda + p\lambda n^3 \log(n/\lambda))$ time). But the problems for $\lambda \leq 2$ have not been solved by the approach based on the edge-splitting operation so far.

The paper is organized as follows: In section 2, we define our problem after introducing some basic notations. In section 3, we prove the NP-completeness of the problem which determines whether we can augment a graph to a 1-NA-edge-connected graph by adding a given number or less new edges. In section 4 and section 5, we prove that the problem of augmenting a 1-NA-edge-connected graph and a 0-NA-edge-connected graph to be 2-NA-edge-connected graph by adding the smallest number of edges can be solved in polynomial time, respectively. In section 6, we give concluding remarks.

## 2. Preliminaries

Let $G$ be an undirected multigraph with a vertex set $V(G)$ and an edge set $E(G)$. We define an *area graph* $(G, \mathcal{W})$ as a pair of a graph $G$ and $\mathcal{W} = \{W_i \mid W_i \subseteq V(G), i = 1, 2, \ldots, p\}$. We refer to each $W_i$ as an *area*. If a vertex subset of $G$ has a non-empty intersection with every area of $\mathcal{W}$, we say that the subset is *area-complete*.

The edge set between $S$ and $T$ $(S, T \subseteq V(G), S \cap T = \emptyset)$ is defined as $E(S, T; G) = \{(v_i, v_j) \in E(G) \mid v_i \in S, v_j \in T\}$. Let $|E(S, T; G)|$ be denoted by $d(S, T; G)$. When $T = V(G) - S$, we refer to $d(S, T; G)$ as $d(S; G)$. $S$ is called a *cut*, when $S$ and $V(G) - S$ are non-empty, and $d(S; G)$ is called the *cut size* of $S$. A cut of size $k$ is called a *k-cut*.

For vertex subsets $P$ and $Q$ $(P \cap Q = \emptyset)$, when $S$ satisfies that $P \subseteq S$ and $Q \subseteq V(G) - S$, we say that cut $S$ separates $P$ and $Q$. When $d(S; G) \geq k$ for all cuts $S$ that separate $P$ and $Q$, $P$ and $Q$ is called *k-edge-connected*. Let $\lambda(P, Q; G)$ be defined as the maximum positive integer $k$ such that $P$ and $Q$ is $k$-edge-connected. For $P$ and $Q$ where $P \cap Q \neq \emptyset$, let $\lambda(P,Q;G)$ be defined as $\infty$. Let $\lambda(G)$ be defined as $\min_{P,Q\subseteq V(G)} \lambda(P,Q;G)$.

For $W \subset V(G)$ and $v \in V(G)$, $\lambda(v,W;G)$ is referred to as the *NA-edge-connectivity between $v$ and $W$ in $G$*, and we say that $v$ and $W$ is *$k$-NA-edge-connected* in $G$ for all $k \leq \lambda(v,W;G)$. We say that $(G,\mathcal{W})$ is $k$-NA-edge-connected, when $v$ and $W$ is $k$-NA-edge-connected for all pairs of $v \in V(G)$ and $W \in \mathcal{W}$. The maximum positive integer $k$ such that $(G,\mathcal{W})$ is $k$-NA-edge-connected is referred to as the *NA-edge-connectivity of* $(G,\mathcal{W})$.

There exists the partition $(D_1, D_2, \ldots, D_q)$ of $V(G)$ (i.e. $D_i \neq \emptyset$, $D_i \cap D_j = \emptyset$ for $1 \leq i, j (\neq i) \leq q$, and $V(G) = \bigcup_{k=1}^{q} D_k$ ) such that $\lambda(v_p, v_q; G) \geq k$ holds for any two vertices $v_p, v_q$ if and only if $v_p, v_q \in D_i$. We refer to each $D_i$ as a $k$-edge-connected component. For a 1-edge-connected component, each subgraph induced by the vertices in the resultant equivalence class is also called a 1-edge-connected component.

Now, we define the NA-edge-connectivity augmentation problem.

**NA-Edge-Connectivity Augmentation Problem ($(\lambda,\delta)$-NAECAP)**
**INSTANCE:** A $\lambda$-NA-edge-connected graph $(G,\mathcal{W})$, positive integers $\delta$ and $b$.
**QUESTION:** Is there an edge set $\hat{E}$ whose size is $b$ or less such that the area graph $(\hat{G},\mathcal{W})$ defined by $\hat{G} = (V(G), E(G) \cup \hat{E})$ is $(\lambda+\delta)$-NA-edge-connected ? ∎

In this paper, we show the following results:

**Theorem 2.1**
$(0,1)$-NAECAP is NP-complete. ∎

**Theorem 2.2**
$(1,1)$-NAECAP can be solved in $O(|\mathcal{W}||V(G)| + |E(G)|)$ time. ∎

**Theorem 2.3**
$(0,2)$-NAECAP can be solved in $O(|\mathcal{W}||V(G)| + |E(G)|)$ time. ∎

## 3. NP-completeness of $(0,1)$-NAECAP

In this section, we prove Theorem 2.1 by reducing the set splitting problem which is NP-complete.

**Set Splitting Problem (SSP)**[3]
**INSTANCE:** Collection $C$ of subsets of a finite set $S$.
**QUESTION:** Is there a partition of $S$ into two subsets $S_1$ and $S_2$ such that no subset in $C$ is entirely contained in either $S_1$ or $S_2$? ∎

**Proof of Theorem 2.1 :** $(0,1)$-NAECAP belongs to the class NP. Therefore, we show its NP-hardness.

We construct an instance of NAECAP from an instance $(S,C)$ of SSP as follows: Let the area graph $(G,\mathcal{W})$ be defined by $G = (S,\emptyset)$ and $\mathcal{W} = C$, and let the upper bound of the number of augmented edges be $|V(G)| - 2$.

We can construct a solution of SSP from a solution of NAECAP as follows: Partition arbitrarily a set $\{D_1, D_2, \ldots, D_p\}$ of all 1-edge-connected components contained in a 1-NA-edge-connected area graph of a solution of NAECAP into two non-empty subclasses $\{D_1, D_2, \ldots, D_i\}$ and $\{D_{i+1}, D_{i+2}, \ldots, D_p\}$. Let $S_1$ (resp., $S_2$) denote the vertex set composed of those vertices appearing in the first (resp., second) class. As every 1-edge-connected component has a common vertex with every area of $\mathcal{W}$, $S_1$ and $S_2$ have a common vertex with every set of $C(=\mathcal{W})$, respectively. Therefore, these sets are a solution of SSP.

Conversely, when a partition into $S_1$ and $S_2$ is a solution of SSP, we can make a solution of NAECAP as two trees which span $S_1$ and $S_2$. Note that the number of all the edges in these trees is $|V(G)|-2$, which is equal to the upper bound of NAECAP. Therefore, we can make a solution of NAECAP from a solution of SSP.

From the above examination, NAECAP has a solution if and only if SSP has a solution. Therefore, SSP is reducible to NAECAP in polynomial time. Consequently, NAECAP is NP-hard. ∎

## 4. $(1,1)$-NAECAP

In this section, we prove Theorem 2.2.

The set of all 1-cuts of a 1-edge-connected graph can be compactly represented by a tree $T$ by shrinking each 2-edge-connected component to a vertex. In addition, we can make the tree $T^*$ such that $T$ is a subdivision of $T^*$. Therefore, a 1-edge-connected graph has its tree representation $T^*$ such that there are no vertices whose degree is less than or equal to two except its leaves. As a 1-NA-edge-connected area graph is composed of some area-complete 1-edge-connected components, the area graph has its *forest representation* that is the set of the tree representations of the 1-edge-connected components. We can make the forest representation of an area graph in linear time. Therefore, in the rest of this paper, we assume that an area graph is given by its forest representation.

When a 2-edge-connected component which is not area-complete is shrinked to a leaf in the forest representation, we refer to the leaf as an *ill vertex*. We refer to a tree representation of a 1-edge-connected component as a *Type A tree* if it includes only one ill vertex and as a *Type B tree* if it includes two or more ill vertices. If there is a component $C$ which is neither a Type A tree nor a Type B tree, it is already 2-NA-edge-connected, since it has no ill vertex. In addition, $V(G)-C$ is area-complete, as a 1-NA-edge-connected area graph is composed of some area-complete 1-edge-connected components. Therefore, in this section, we assume that every 1-edge-connected component in an area graph is either a Type A tree or a Type B tree.

There are only nine cases in Table 1 for combination of the number of Type A trees and Type B trees in an area graph. We can determine the case to which an area graph belongs in $O(|\mathcal{W}||V(G)|)$ time.

Table 1: A classification of area graphs

| | Type B:0 | Type B:1 | Type B:two or more |
|---|---|---|---|
| Type A:0 | Case1 | Case4 | Case7 |
| Type A:1 | Case2 | Case5 | Case8 |
| Type A:two or more | Case3 | Case6 | Case9 |

Note: For example, 'Type A:1' means that the number of Type A trees is one.

Now, we use the following property to estimate the number of added edges.

**Theorem 4.1 (Ito and Yokoyama [10])**
An area graph $(G,\mathcal{W})$ is $\lambda$-NA-edge-connected, if and only if all $i$-edge-connected components $C$ for $1 \le i \le \lambda$ satisfy at least one of the following conditions:
(1) $C$ is area-complete,
(2) $d(C;G) \ge \lambda$. ∎

In $(1,1)$-NAECAP, it is necessary to add at least $\lceil |K|/2 \rceil$ edges where $K$ is the set of the ill vertices in an area graph, because each ill vertex must have at least two edges connected to other vertices by Theorem 4.1. We estimate the lower bound of the number of added edges in the following lemma.

**Lemma 4.1**
The lower bound of the number of added edges for $(1,1)$-NAECAP is

$$\begin{cases} |K|/2+1 & \text{(If a given area graph satisfies either the following Condition A or B)} \\ \lceil |K|/2 \rceil & \text{(otherwise)} \end{cases}$$

**Condition A:** The given area graph belongs to Case4, $|K|$ is even, $V(T)$ is not area-complete, and $d(V(T);G)=1$, where $T$ is the minimal tree spanning all the vertices of $K$.

**Condition B:** The given area graph belongs to Case5, $|K|$ is even, and there is no ill vertex $v$ such that $V(H)-\{v\}$ is area-complete, where $H$ is the Type B tree.

**Proof:** First, assume that a given area graph satisfies the Condition A. As the area graph of this case is a tree (note that a 1-NA-edge-connected area graph is given by a forest representation), $T$ is unique. We can make some 2-edge-connected-components by adding $|K|/2$ edges to the ill vertices (i.e. the leaves of $T$). In these components, there is at least one 2-edge-connected-component whose cut size is one. Indeed, if $V(T)$ becomes a 2-edge-connected-component, its cut size is one, from the condition that $d(V(T);G)=1$. If the 2-edge-connected-components are properly included in $V(T)$, there is a 2-edge-connected-component which corresponds to a leaf in the tree representation, and its cut size is one. In addition, no 2-edge-connected-components are area-complete, as $V(T)$ is not area-complete. Therefore, there is a 2-edge-connected-component such that its cut size is one and that it is not area-complete. It follows that we cannot satisfy the conditions of Theorem 4.1 by adding $|K|/2$ edges. Hence, at least $|K|/2+1$ edges are necessary.

Next, assume that a given area graph satisfies the Condition B. Let $P$ be the set of all the ill vertices in $H$. Note that $|P|(=|K|-1)$ is odd. The ill vertex in the Type A tree must be connected to an ill vertex $v$ in $P$. We show that there is at least one 2-edge-connected-component whose cut size is one after $(|P|-1)/2=|K|/2-1$ edges are added to all the ill vertices except $v$ in $P$. The 2-edge-connected-components are included in $V(T)-\{v\}$, where $T$ is the minimal tree spanning all the vertices of $P$. If $H$ has an area-complete leaf, $V(H)-\{v\}$ is area complete for every ill vertex $v$ in $P$; hence, it contradicts to the Condition B. Therefore, $H$ has no area-complete leaf, that is, $H=T$. It follows that all 2-edge-connected-components are included in $V(H)-\{v\}$. If $V(H)-\{v\}$ becomes a 2-edge-connected-component, its cut size is one. If the 2-edge-connected-components are properly included in $V(H)-\{v\}$, there is a 2-edge-connected-component which corresponds to a leaf in the tree representation, and its cut size is one. In addition, no 2-edge-connected-components are area-complete, as $V(H)-\{v\}$ is not area-complete. Therefore, there is a 2-edge-connected-component such that its cut size is one and it is not area-complete. It follows that we cannot satisfy the conditions of Theorem 4.1 by adding $1+(|P|-1)/2 = 1+|K|/2-1=|K|/2$ edges. Hence, at least $|K|/2+1$ edges are necessary. ∎

Based on the algorithm proposed in [15], we define procedure CEA (Cyclic Edge Augmentation) for a Type B tree $H$ as follows: Let the set of the ill vertices and the set of area-complete leaves in $H$ be $P$ and $Q$, respectively. Let $T$ be the minimal tree spanning all the vertices of $P$. As $H$ is a tree by the assumption that a 1-NA-edge-connected area graph is given by a forest representation, $T$ is unique and it can be found in linear time. Let $T(a)$ be the subgraph of $T$ and the minimal tree spanning all the ill vertices of $P-\{a\}$.

**Procedure CEA**

**Step.1** If $|P|$ is odd and $Q$ is empty, choose an ill vertex $v^*$. If $|P|$ is odd and $Q$ is not empty, choose an ill vertex $v^*$ such that $d(V(T(v^*)); H) \geq 2$. Let $\tilde{P}$ be $P - \{v^*\}$. If $|P|$ is even, let $\tilde{P}$ be $P$. Let $\tilde{T}$ be the minimal tree spanning all the vertices of $\tilde{P}$.

**Step.2** Scan $\tilde{T}$ in depth-first-search manner from a vertex in $V(\tilde{T})$, and number all ill vertices $v_i$ $(i = 1, 2, \ldots, |\tilde{P}|)$ in the order that they are first encountered. These vertices are ordered cyclically so that the last ill vertex $v_{|\tilde{P}|}$ is followed by the first ill vertex $v_1$.

**Step.3** Add an edge between $v_i$ and $v_{i+|\tilde{P}|/2}$ for all $i$ $(i = 1, 2, \ldots, |\tilde{P}|/2)$, then stop. ∎

If $|P|$ is odd and $Q$ is not empty, there is an ill vertex $v^*$ such that $d(V(T(v^*)); H) \geq 2$. Indeed, if $d(V(T(v)); H) = 1$ for an ill vertex $v$ in $P$, we can take an ill vertex in $V(T) - \{v\}$ as $v^*$. We can find it in linear time. Therefore, we can execute procedure CEA in $O(V(H) + E(H))$ time.

Figure 1 shows an example of procedure CEA.

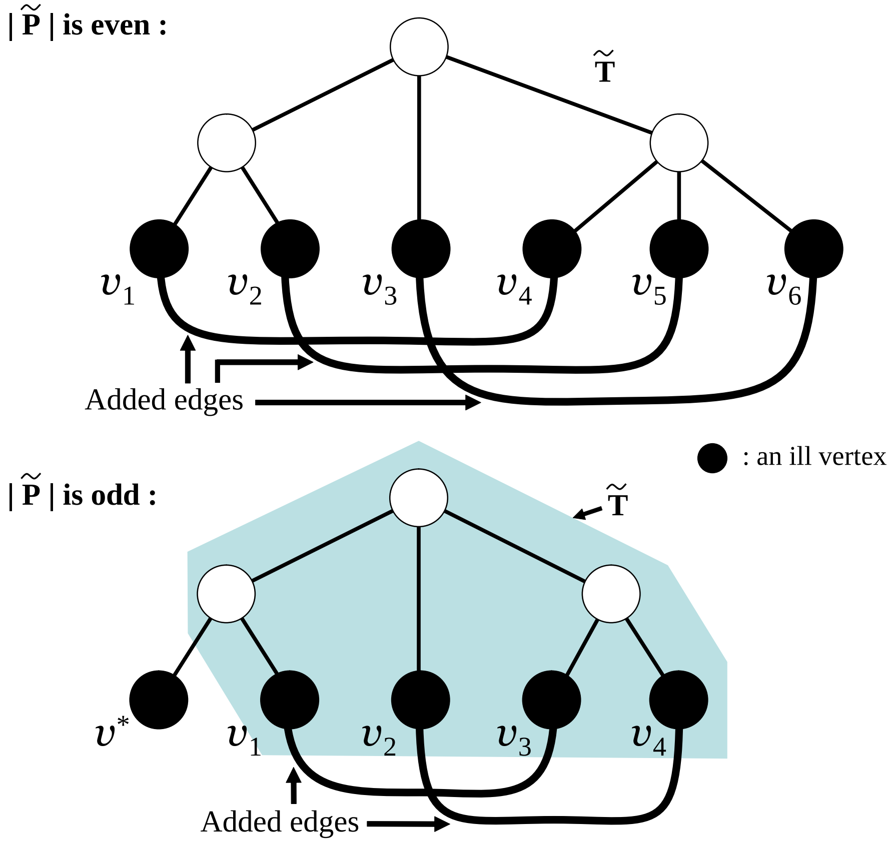


Figure 1: An example for CEA

**Lemma 4.2**
When procedure CEA is applied to Type B tree $H$, $V(\tilde{T})$ becomes a 2-edge-connected component.

**Proof:** Assume that the last ill vertex $v_{|\tilde{P}|}$ is followed by the first ill vertex $v_1$. For any 1-cut $C$ in $\tilde{T}$, all the ill vertices in $C$ have consecutive numbers, and $V(H) - C$ has also consecutive numbers. Without loss of generality, we can assume that $C$ contains less than or equal to $|\tilde{P}|/2$ ill vertices and that $v_i \in C$. It follows that $v_{i+|\tilde{P}|/2}$ must lie in $V(H) - C$. Therefore, the size of any 1-cut of $\tilde{T}$ is increased at least by one, as the edge $(v_i, v_{i+|\tilde{P}|/2})$ bridges $C$ and $V(H) - C$. In addition, as $H$ is a tree, there is no 2-edge-connected vertex subset which includes $\tilde{T}$ properly. Hence, $V(\tilde{T})$ becomes a 2-edge-connected component. ∎

**Lemma 4.3**
In a Type B tree $H$, if $|P| \geq 4$, there are two ill vertices $a$ and $b$ such that $d(V(T(a,b)); H) \geq 2$, where $T(a,b)$ is the minimal tree spanning all the ill vertices except $\{a,b\}$ in $H$.

**Proof:** Number all the ill vertices in a manner similar to procedure CEA. If $|P| \geq 4$, the ill vertices in $P - \{v_1, v_{\lceil |P|/2 \rceil + 1}\}$ does not have consecutive numbers. Therefore, $d(V(T(v_1, v_{\lceil |P|/2 \rceil + 1})); H) \geq 2$. ∎

The outline of the proposed algorithm is described as follows: For a 1-NA-edge-connected area graph $(G, \mathcal{W})$, we determine the case to which the area graph belongs. Then, we execute the procedure of its case.

We describe the procedure for each case in Table 1.

**Case1:**
**Procedure:** If $(G, \mathcal{W})$ belongs to Case1, then stop.

**Case2:**
**Procedure:** Add an edge between the ill vertex and another area-complete leaf. Then stop. (see Figure 2)

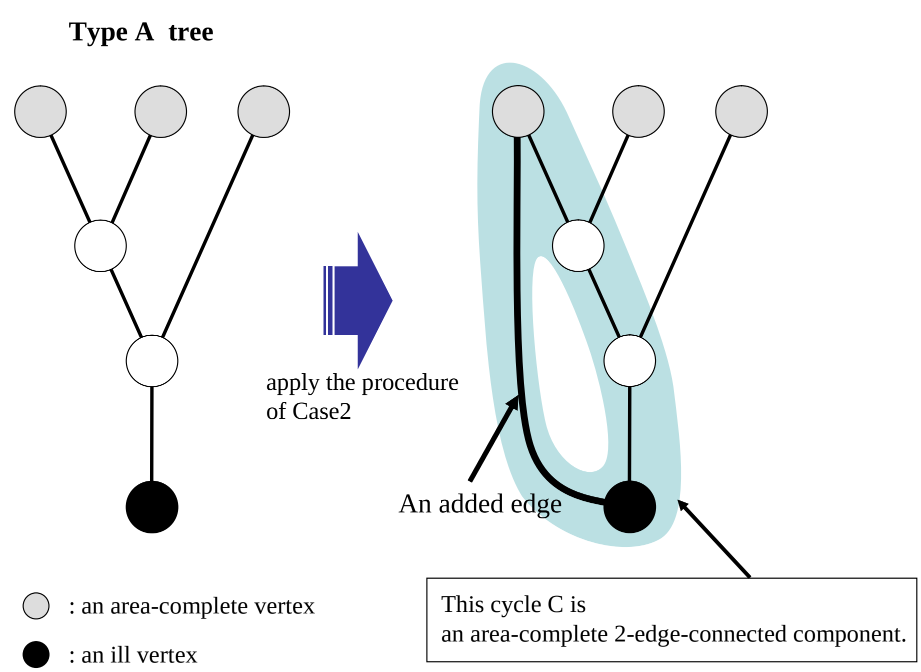


Figure 2: An example for case2

**Case3:**
**Procedure:** Make pairs of the Type A trees as many as possible, and then add an edge

between two ill vertices in each pair. When the number of the Type A trees is odd, add an edge to the remaining Type A tree by the procedure of Case2. Then stop. (see Figure 3)

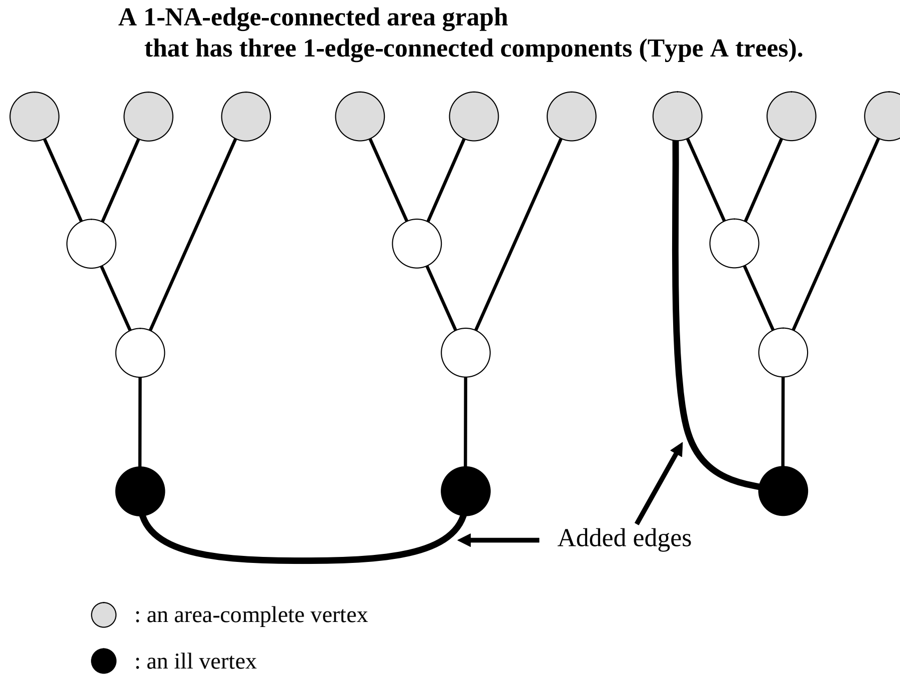


Figure 3: An example for case3

From Case4 to Case6, let $H$, $P$, and $Q$ be the Type B tree in $(G, \mathcal{W})$, the set of the ill vertices in $H$, and the set of the area-complete leaves in $H$, respectively. Let $T$ be the minimal tree spanning all the vertices in $P$. Let $T(a)$ (resp., $T(b,c)$) be the subgraph of $T$ and the minimal tree spanning all the ill vertices except $\{a\}$ (resp., $\{b,c\}$). Let $K$ be the set of all the ill vertices in $(G, \mathcal{W})$.

**Case4:**
In this case, $G = H$, and $K = P$.

**(4-1)** $|K|(= |P|)$ **is even.**
**Procedure:** Apply procedure CEA to $H$. If $(G, \mathcal{W})$ satisfies the Condition A, then add an edge between a vertex in $P$ and a vertex in $Q$ (we refer to this edge as the extra edge). Then stop. (see Figure 4 and Figure 5)

**(4-2)** $|K|(= |P|)$ **is odd.**
**Procedure:** Apply procedure CEA to $H$. If $Q$ is not empty, add an edge between $v^*$ and a vertex in $Q$; otherwise, add an edge between $v^*$ and an arbitrary vertex in $V(T) - \{v^*\}$. Then stop. (see Figure 6)

**Case5:**
**(5-1)** $|K|$ **is even**
**Procedure:** If $(G, \mathcal{W})$ satisfies the Condition B, apply procedure CEA to $H$, then add an edge between $v^*$ and the ill vertex in the Type A tree. Furthermore, add an edge between a vertex in $V(H) - \{v^*\}$ and a vertex in the Type A tree (we refer to this edge as the extra edge). Otherwise, if $Q$ is not empty, find an ill vertex $w$ such that $d(V(T(w)); H) \geq 2$;

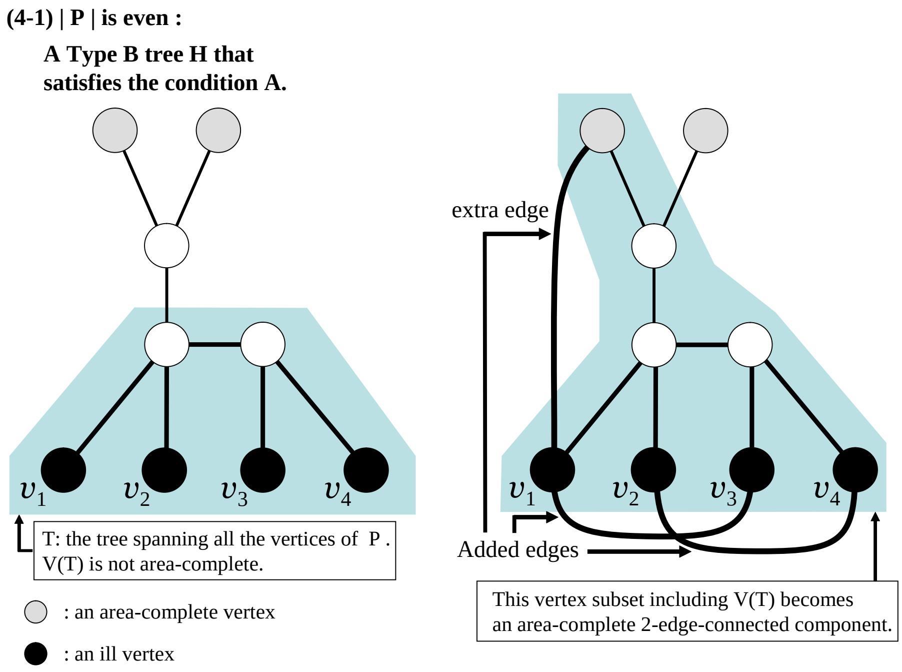


Figure 4: An example (1) for $|P|$ is even in case4

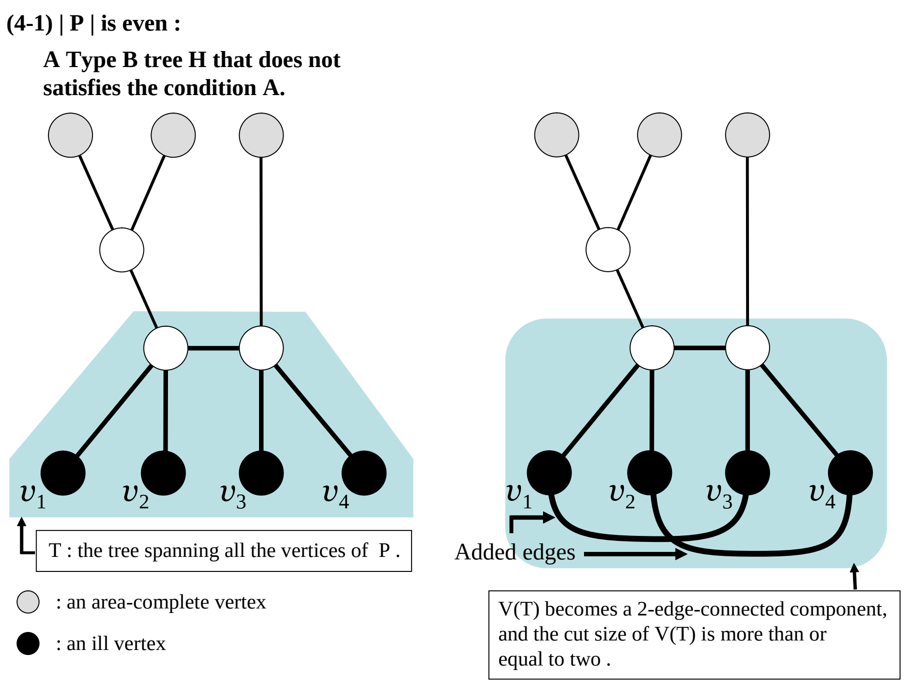


Figure 5: An example (2) for $|P|$ is even in case4

if $Q$ is empty, find an ill vertex $w$ such that $V(H) - \{w\}$ is area-complete. Then, apply procedure CEA to $H$ by using $w$ as $v^*$, and add an edge between $v^*(= w)$ and the ill vertex in the Type A tree. Then stop. (see Figure 7 and Figure 8)

**(5-2) $|K|$ is odd**
**Procedure:** Apply procedure CEA to $H$, then add an edge between a vertex in $V(T)$ and

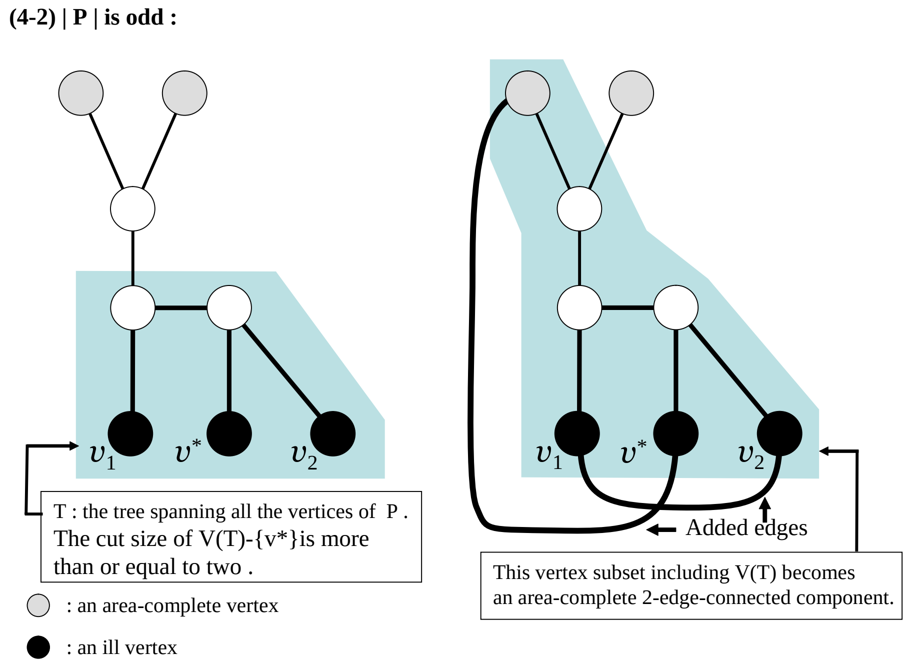


Figure 6: An example for $|P|$ is odd in case4

the ill vertex in the Type A tree. Then stop. (see Figure 9)

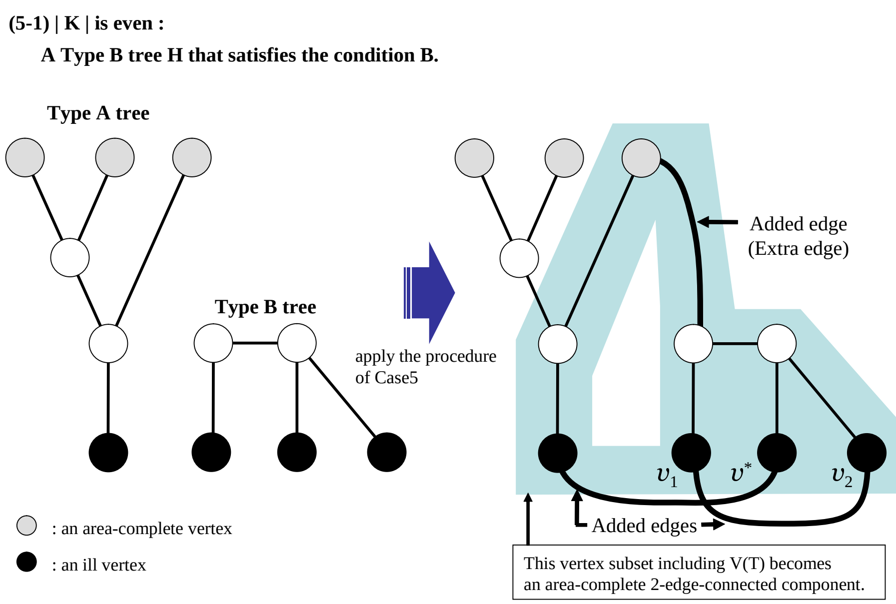


Figure 7: An example (1) for $|K|$ is even in case5

**Case6:**
**Procedure:** If $|P| \geq 4$, find two ill vertices $a$ and $b$ such that $d(V(T(a,b)); H) \geq 2$; otherwise, choose two ill vertices $a$ and $b$ in $P$. Add the edges $(t_1, a)$ and $(t_2, b)$ for the ill vertices $t_1$ and $t_2$ in different two Type A trees. If $|P| \geq 4$, apply procedure CEA to the resultant tree. Then, apply the procedure of Case1, Case2, or Case3 according to the number of the resultant Type A trees. Then stop. (see Figure 10)

**(5-1) | K | is even :**

**A Type B tree H that does not satisfies the condition B.**

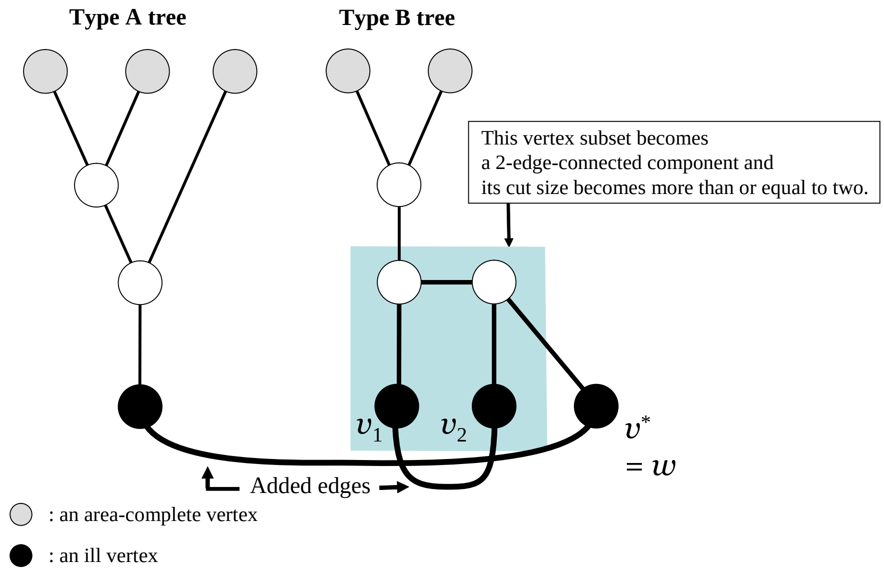


Figure 8: An example (2) for $|K|$ is even in case5

**(5-2) | K | is odd :**

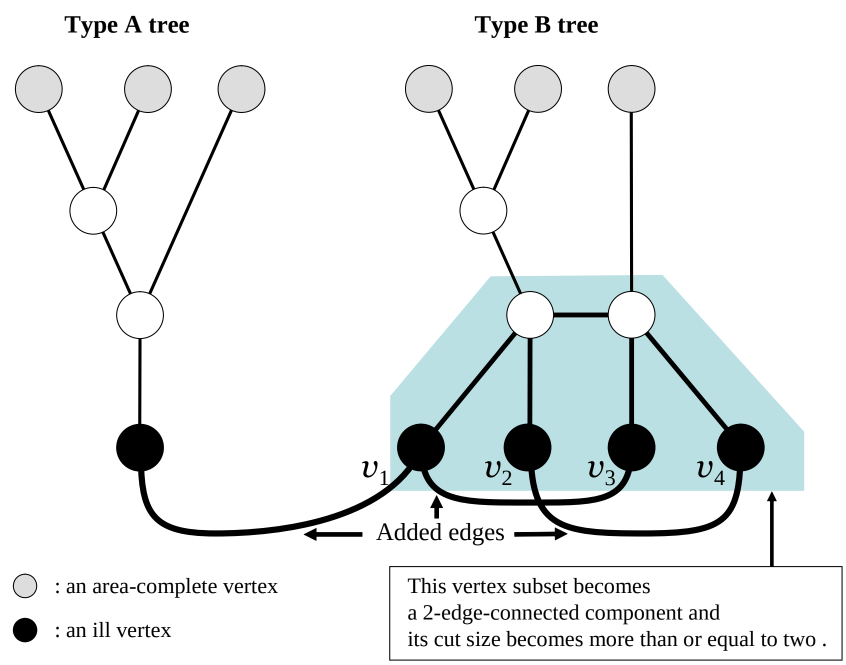


Figure 9: An example for $|K|$ is odd in case5

From Case7 to Case9, let $H_i$, $P_i$, and $Q_i$ $(1 \leq i \leq t)$ be a Type B tree in $(G, \mathcal{W})$, the set of the ill vertices in $H_i$, and the set of the area-complete leaves in $H_i$, respectively. Let $T_i$ be the minimal tree spanning all the vertices in $P_i$. If $|P_i| < 4$, let $a_i$ and $b_i$ be two ill vertices in $H_i$; otherwise, let $a_i$ and $b_i$ be two ill vertices in $H_i$ such that $d(V(T_i(a_i, b_i)); H_i) \geq 2$, where $T_i(a_i, b_i)$ be the subgraph of $T_i$ and the minimal tree spanning all the ill vertices except

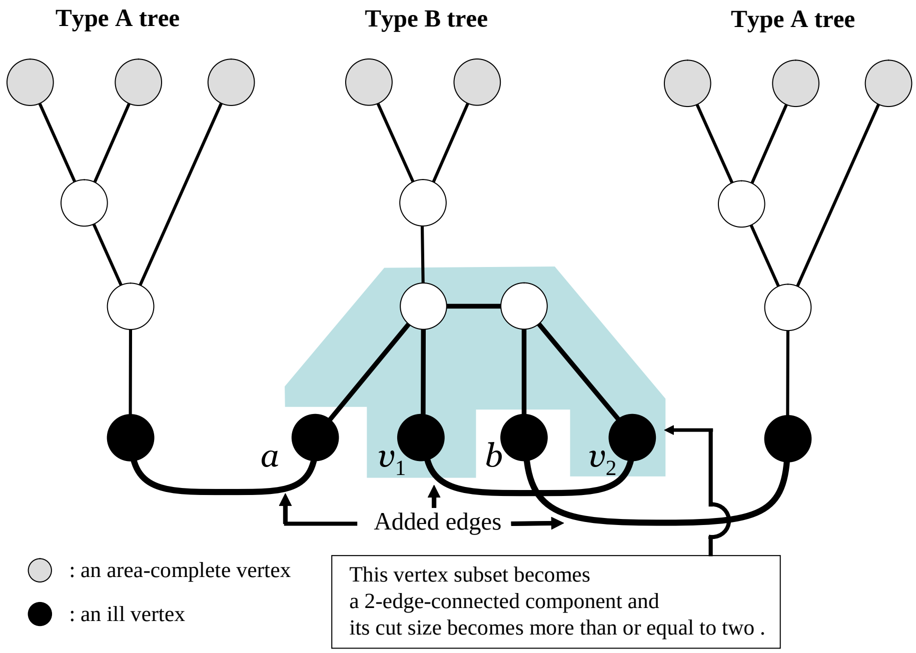


Figure 10: An example for case6

$\{a_i, b_i\}$. Let $K$ be the set of all the ill vertices in $(G, \mathcal{W})$.

**Case7:**
**Procedure:** Add edges $(b_1, a_2)$, $(b_2, a_3)$, $\ldots$, $(b_{t-1}, a_t)$, $(b_t, a_1)$. We refer to these edges as the cycle edges. If the resultant graph is 2-NA-edge-connected, then stop. If the tree representation of the resultant graph is a Type A (resp., Type B) tree, apply the procedure of Case2 (resp., Case4), then stop. (see Figure 11)

**Case8:**
**Procedure:** Add edges $(b_1, a_2)$, $(b_2, a_3)$, $\ldots$, $(b_{t-1}, a_t)$, $(b_t, a_1)$. If the resultant graph is 2-NA-edge-connected, apply the procedure of Case2 to the remaining Type A tree, then stop. If the tree representation of the resultant graph is a Type A (resp., Type B) tree, apply the procedure of Case3 (resp., Case5), then stop.

**Case9:**
**Procedure:** Add edges $(b_1, a_2)$, $(b_2, a_3)$, $\ldots$, $(b_{t-1}, a_t)$, $(b_t, a_1)$. If the resultant graph is 2-NA-edge-connected, apply the procedure of Case3 to the remaining Type A trees, then stop. If the tree representation of the resultant graph is a Type A (resp., Type B) tree, apply the procedure of Case3 (resp., Case6), then stop.

Now, we prove Theorem 2.2.

**Proof of Theorem 2.2 :** First, we prove that a given area graph $(G, \mathcal{W})$ becomes 2-NA-edge-connected with the smallest number of the added edges by the procedure of each case.

**Case1:**
$(G, \mathcal{W})$ is already 2-NA-edge-connected.

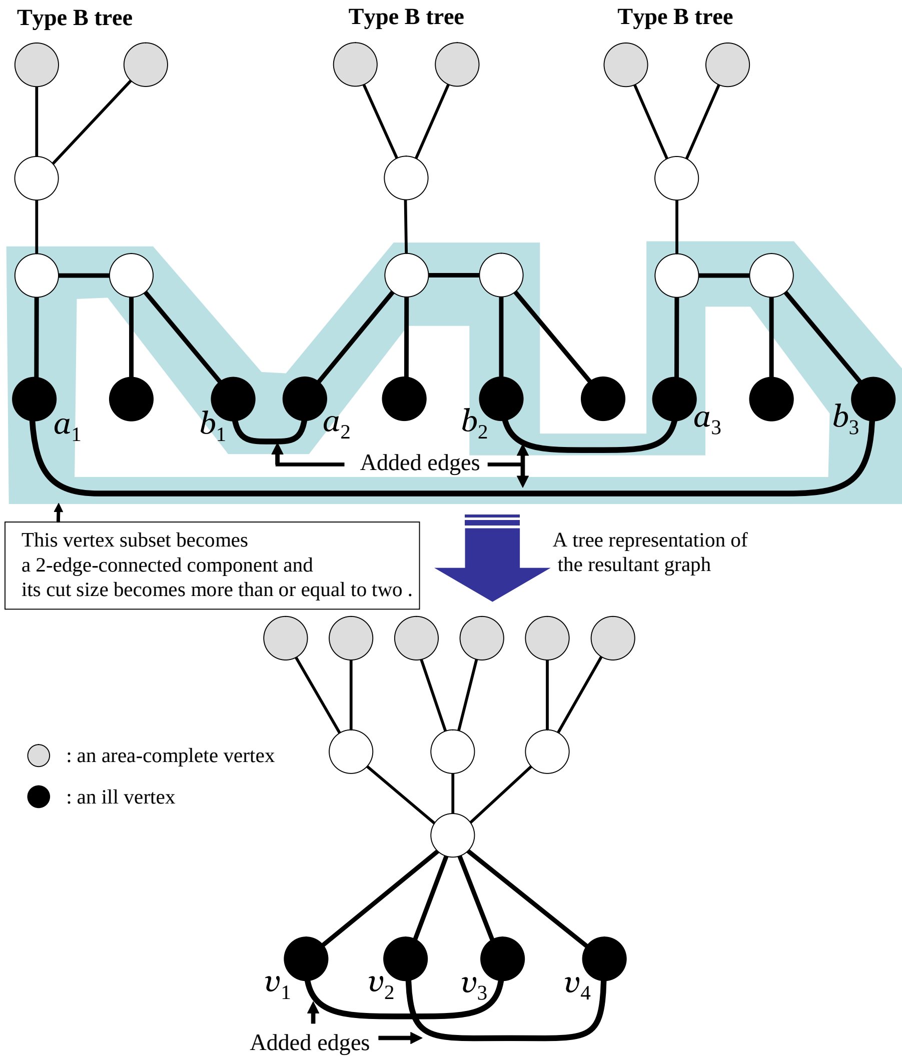


Figure 11: An example for case7

**Case2:**
The edge added by the procedure makes a cycle $C$ including the ill vertex. $V(C)$ is an area-complete 2-edge-connected component, because $C$ includes the area-complete leaf.

There exists at least one area-complete leaf in $G$. Assume that there is no area-complete leaf but only one ill vertex. It follows that $G$ is not area-complete, because $G$ is composed of only one ill vertex. This contradicts to the assumption that $G$ is 1-NA-edge-connected. Therefore, there is at least one area-complete leaf, and we can execute the procedure.

Consequently, the area graph to which an edge is added by the procedure satisfies the conditions of Theorem 4.1.

In addition, the number of the added edge by the procedure is the smallest, because $\lceil |P|/2 \rceil = 1$ is the lower bound by Lemma 4.1.

**Case3:**
The degree of every ill vertex becomes two. Therefore, the resultant area graph satisfies the conditions of Theorem 4.1.

The number of edges added by the procedure is $\lceil |P|/2 \rceil$. It is the smallest by Lemma 4.1.

**Case4:**

**(4-1) $|K|(=|P|)$ is even.**

When procedure CEA is applied to $H$, $V(T)$ becomes a 2-edge-connected component by Lemma 4.2.

If $V(T)$ is area-complete, the resultant area graph satisfies the conditions of Theorem 4.1.

Even if $V(T)$ is not area-complete, the resultant area graph also satisfies the conditions of Theorem 4.1. We show this fact as follows: (1) When $d(V(T);H) \geq 2$, the conditions of Theorem 4.1 are satisfied. (2) When $d(V(T);H) = 1$, the area graph satisfies the Condition A. In this case, the extra edge between $P$ and $Q$ is added by the procedure. It is possible to add the extra edge, because $Q \neq \emptyset$ from the assumption that $d(V(T);H) = 1$. The extra edge makes a new cycle. The union $U$ of this cycle and $V(T)$ becomes a 2-edge-connected component. In addition, $U$ is area-complete, as $U$ includes a vertex in $Q$. Hence, the resultant area graph also satisfies the conditions of Theorem 4.1. (3) When $d(V(T);H) = 0$, $T = H$. This implies that $V(T)$ is area-complete. But it contradicts that $V(T)$ is not area-complete. Therefore, this case is impossible.

If $(G, \mathcal{W})$ satisfies the Condition A, the number of edges added by the procedure is $|P|/2 + 1$; otherwise, it is $|P|/2$. It is the smallest by Lemma 4.1.

**(4-2) $|K|(=|P|)$ is odd.**

When procedure CEA is applied to $H$, $V(T) - \{v^*\}$ becomes a 2-edge-connected component by Lemma 4.2. Furthermore, the union $U$ of the new cycle including $v^*$ and $V(T) - \{v^*\}$ becomes a 2-edge-connected component. If $Q$ is not empty, $U$ is area-complete, as $U$ includes a vertex in $Q$. If $Q$ is empty, $U$ is also area-complete, as $U$ is equal to $V(T)$ which is area-complete. Consequently, the resultant area graph satisfies the conditions of Theorem 4.1.

The number of edges added by the procedure is $\lceil |P|/2 \rceil$. It is the smallest by Lemma 4.1.

**Case5:**

**(5-1) $|K|$ is even**

Note that $|P|$ is odd. If $(G, \mathcal{W})$ satisfies the Condition B, the resultant area graph is 2-NA-edge-connected. We show this fact as follows: Note that $H = T$, as shown in the proof in Lemma 4.1. When procedure CEA is applied to $H$, $V(T) - \{v^*\}(= V(H) - \{v^*\})$ becomes a 2-edge-connected component by Lemma 4.2. Furthermore, as the procedure makes the cycle including the extra edge and $v^*$, the union $U$ of this cycle and $V(H) - \{v^*\}$ becomes a 2-edge-connected component. $U$ is area-complete, as $U$ includes $V(H)$ which is area-complete. Consequently, the resultant area graph satisfies the conditions of Theorem 4.1.

If $(G, \mathcal{W})$ does not satisfy the Condition B, the resultant area graph is also 2-NA-edge-connected. We show this fact as follows: $V(T) - \{w\}$ becomes a 2-edge-connected component by procedure CEA. If $Q$ is empty, $V(T) - \{w\}(= V(H) - \{w\})$ is area-complete. Otherwise, $d(V(T(w)); H) \geq 2$. Consequently, in both cases, the resultant area graph satisfies the conditions of Theorem 4.1.

If the area graph satisfies the Condition B, the number of edges added by the procedure is $|K|/2 + 1(= |P - 1|/2 + 2)$ , otherwise, $|K|/2(= |P - 1|/2 + 1)$. It is the smallest by Lemma 4.1.

**(5-2) $|K|$ is odd**

Note that $|P|$ is even. When procedure CEA is applied to $H$, $V(T)$ becomes a 2-edge-connected component by Lemma 4.2. If $T = H$, $V(T)(= V(H))$ is area-complete. Otherwise, as there is an area-complete leaf, $d(V(T); \hat{G}) \geq 2$ where $\hat{G}$ is the resultant area graph.

Consequently, in both cases, $\hat{G}$ satisfies the conditions of Theorem 4.1.

The number of edges added by the procedure is $\lceil |K|/2 \rceil (= |P|/2+1)$. It is the smallest by Lemma 4.1.

**Case6:**
When two edges $(t_1, a)$ and $(t_2, b)$ are added, the set of the ill vertices in $H$ becomes $P - \{a, b\}$. If $|P - \{a, b\}|$ is even (note that $|P| \geq 4$), $V(T(a, b))$ becomes a 2-edge-connected component by Lemma 4.2, and $d(V(T(a, b)); H) \geq 2$ by Lemma 4.3. Therefore, the resultant tree becomes 2-NA-edge-connected. If $|P - \{a, b\}|$ is odd, the resultant tree becomes a Type A tree. If $|P - \{a, b\}| = 0$, the resultant tree is 2-NA-edge-connected. Consequently, the resultant area graph has only Type A trees. By applying the procedure of Case1, Case2, or Case3 according to the number of the resultant Type A trees, the area graph becomes 2-NA-edge-connected.

The number of edges added by the procedure is $\lceil |K|/2 \rceil$. It is the smallest by Lemma 4.1.

**Case7:**
As the tree representation of the resultant graph by adding the cycle edges is a Type A (resp., Type B) tree, it is a 2-NA-edge-connected component by applying the procedure of Case2 (resp., Case4).

If the tree representation of the resultant graph by adding the cycle edges is a Type A tree or a 2-NA-edge-connected graph, the number of edges added by the procedure is $\lceil |K|/2 \rceil$, which is the smallest by Lemma 4.1. If it is a Type B tree (let it be denoted by $\bar{H}$), it does not satisfies the Condition A. We show this fact as follows: Let $\bar{P}$ and $\bar{T}$ be the set of the ill vertices in $\bar{H}$ and the minimal tree spanning all the vertices in $\bar{P}$, respectively. Assume that $d(V(\bar{T}); \bar{H}) = 1$. This implies that (1) $V(\bar{T})$ was a 1-cut included in a Type B tree or that (2) $d(\cup V(T_i); G) = 1$. In the former case, $\bar{P}$ has consecutive numbers, but it contradicts to how to add the cycle edges. Hence, this case is impossible. In the latter case, $\cup V(T_i)$ is area-complete, because there exits an area-complete tree $T_p$ $(1 \leq p \leq t)$ such that $d(T_p; H_p) = 0$. Therefore, $\bar{H}$ does not satisfies the Condition A. Consequently, the number of added edges is $\lceil |K|/2 \rceil$, which is the smallest by Lemma 4.1.

**Case8:**
The resultant area graph is obviously 2-NA-edge-connected.

If the tree representation of the resultant graph by adding the cycle edges is a Type A tree or a 2-NA-edge-connected graph, the number of edges added by the procedure is $\lceil |K|/2 \rceil$, which is the smallest by Lemma 4.1. If it is a Type B tree (let it be denoted by $\bar{H}$), it does not satisfies the Condition B, because $V(\bar{H}) - \{v\}$ for an ill vertex $v$ in $\bar{H}$ includes an area-complete Type B tree. Consequently, the number of added edges is $\lceil |K|/2 \rceil$, which is the smallest by Lemma 4.1.

**Case9:**
The resultant area graph is obviously 2-NA-edge-connected. And the number of added edges is $\lceil |K|/2 \rceil$, which is the smallest by Lemma 4.1.

Next, we estimate the computational complexity. Let $n$, $m$, and $p$ be $|V(G)|$, $|E(G)|$, and $|\mathcal{W}|$, respectively. $O(n+m)$ is necessary to make a forest representation in depth-first-search manner. $O(pn)$ is necessary to determine whether every tree in a forest representation is a Type A or Type B tree by checking the areas including each vertex. Therefore, it takes $O(pn + m)$ time to determine the case to which the area graph belongs.

$O(n+m)$ time is necessary to execute the procedure of Case2 and Case3, respectively.

$O(n+m)$ time is necessary to execute the procedure of Case4, because it takes $O(n+m)$ time to check the Condition A and it takes $O(n+m)$ time to execute procedure CEA.

$O(pn+m)$ time is necessary to execute the procedure of Case5. Because it takes $O(pn)$ time to check the Condition B and to find a vertex $w$, in addition to $O(n+m)$ time to execute procedure CEA.

$O(n+m)$ time is necessary to execute the procedure of Case6, as it takes $O(n+m)$ time to find two ill vertices $a$ and $b$ such that $d(V(T)-\{a,b\};H)\geq 2$ and other procedures can be executed in $O(n+m)$ time.

$O(pn+m)$ time is necessary to execute the procedure of Case7, because it takes $O(pn)+O(n+m)$ time to check whether a tree representation of the resultant graph is a Type A tree, Type B tree, or a 2-NA-edge-connected area graph.

$O(pn+m)$ time is necessary to execute the procedure of Case8 (resp., Case9), because it takes $O(pn)+O(n+m)$ time to check whether a tree representation of the resultant graph is a Type A tree, Type B tree, or a 2-NA-edge-connected area graph and to execute the procedure of Case5 (resp., Case6).

Consequently, $O(pn+m)$ time is necessary in total.

∎

## 5. $(0,2)$-NAECAP

In this section, we prove Theorem 2.3.

We assume that every 1-edge-connected component in an area graph is either a Type A tree or a Type B tree by the following reason: Let $C$ be a component which is neither a Type A tree nor a Type B tree. $C$ is already 2-NA-edge-connected. If $V(G)-C$ is area-complete, we apply the algorithm for $(1,1)$-NAECAP or $(0,2)$-NAECAP to the area graph after removal of $C$; otherwise, we apply the algorithm for $(0,2)$-NAECAP to the area graph in which we regard $C$ as an ill vertex. The smallest number of edges added to the modified area graph is less than or equal to that to the original area graph, because the lower bound $B$ of the number of added edges does not change and we can augment an given area graph to a 2-NA-edge-connected graph by adding $B$ edges as we show in Theorem 2.2 and Theorem 2.3.

First, we estimate the lower bound of the number of added edges. In a forest representation of a 0-NA-edge-connected area graph, some trees which are not area-complete are included. Let the set of such trees be $F$. We assume that $F\neq\emptyset$. When a tree in $F$ is composed of an ill vertex $v$, we convert it to a Type B tree which is a complete graph on two ill vertices included by the same areas as $v$. The number of added edges does not change by this operation. Therefore, we can assume that all the trees included in $F$ are the Type B trees which are not area-complete. Let the set of the Type B trees not included in $F$ be $\hat{H}$. Let the minimal tree spanning all the ill vertices in a Type B tree $H$ in $\hat{H}$ be $T$ and the set of such trees be $\hat{T}$. Let the set of all the ill vertices in $(G,\mathcal{W})$ be $K$.

If $(G,\mathcal{W})$ satisfies the following conditions, the lower bound of the number of added edges is $|K|/2+1$.

**Condition C:** $(G,\mathcal{W})$ belongs to Case5 and $|K|$ is even.

**Condition D:** $(G,\mathcal{W})$ belongs to Case7, $V(\hat{T})\cup V(F)$ is not area-complete, $|K|$ is even, and $d(V(\hat{T})\cup V(F);G)=1$.

**Condition E:** $(G,\mathcal{W})$ belongs to Case8, $|K|$ is even, and there is no ill vertex $v$ included in a Type B tree with three or more ill vertices such that $V(\hat{H})\cup V(F)-\{v\}$ is area-complete.

We show that, if the above conditions are satisfied, the lower bound of the number of added edges is $|K|/2+1$.

Condition C:

In this case, $\hat{H} = \emptyset$ and $V(F)$ is not area-complete. Because, if $V(F)$ is area-complete, $F$ is composed of an area-complete Type B tree and it contradicts to the assumption. Therefore, two edges must be added between the Type B tree in $F$ and the Type A tree to increase the cut size of $V(F)$ to two. As at least $|K|/2$ edges are necessary to increase the degree of all the ill vertices to two, only one edge is added between the Type A tree and the Type B tree in $F$ by the addition of $|K|/2$ edges. Therefore, more one extra edge must be added between the Type B tree in $F$ and the Type A tree. Consequently, at least $|K|/2+1$ edges must be added.

Condition D:

We must add at least $|K|/2$ edges to increase the degree of all the ill vertices to two. But more one edge must be added between $V(\hat{T}) \cup V(F)$ and $V(G) - (V(\hat{T}) \cup V(F))$, because $V(\hat{T}) \cup V(F)$ is not area-complete and $d(V(\hat{T}) \cup V(F); G) = 1$. Therefore, at least $|K|/2+1$ edges must be added.

Condition E:

When $(G, \mathcal{W})$ satisfies the Condition E, there are two cases; (1) there is no ill vertex $v$ such that $V(\hat{H}) \cup V(F) - \{v\}$ is area-complete, or (2) there is an ill vertex $v$ included in a Type B tree with two ill vertices such that $V(\hat{H}) \cup V(F) - \{v\}$ is area-complete.

The former case (1): It is necessary to add an edge to increase the cut size of $V(\hat{H}) \cup V(F) - \{v\}$ from one to two for an ill vertex $v$. $|K|/2$ edges are necessary to increase the degree of all the ill vertices to two. In addition to the edge between $v$ and the ill vertex in the Type A tree, $|K|/2-1$ edges are necessary for $|K|-2$ ill vertices in $V(\hat{H}) \cup V(F) - \{v\}$. Therefore, we cannot increase the cut size of $V(\hat{H}) \cup V(F) - \{v\}$ to two by adding $|K|/2$ edges; hence, one extra edge must be added between $V(\hat{H}) \cup V(F) - \{v\}$ and $V(G) - (V(\hat{H}) \cup V(F) - \{v\})$. Consequently, at least $|K|/2+1$ edges must be added.

The latter case (2): An edge between $v$ and the ill vertex in the Type A tree must be added. There is a Type B tree with three or more ill vertices, because the number of the ill vertices in $V(\hat{H}) \cup V(F)$ is odd. Therefore, for any $|K|/2$ edges augmentation, there is a path $P_v$ from $v$ to an ill vertex $v'$ in a Type B tree with three or more ill vertices. $V(\hat{H}) \cup V(F) - V(P_v)$ is not area-complete, because, otherwise, there exists the ill vertex $v'$ included in a Type B tree with three or more ill vertices such that $V(\hat{H}) \cup V(F) - \{v'\}$ is area-complete, which contradicts to the Condition E. In addition, the cut size of $V(\hat{H}) \cup V(F) - V(P_v)$ is one. Therefore, there is no $|K|/2$ edges augmentation to meet the requirement, and $|K|/2+1$ edges must be added.

The algorithm is basically the same as that of $(1,1)$-NAECAP, but we modify the following procedures:

**Case5:** If the number of the ill vertices in the Type B tree $H'$ in $F$ is less than or equal to three, let $a$ and $b$ be two ill vertices in $H'$; otherwise, let $a$ and $b$ be two ill vertices in $H'$ such that $d(V(H'(a,b)); H') \geq 2$ where $H'(a,b)$ is the minimal tree spanning all the ill vertices except $\{a,b\}$ in $H'$. Add an edge between vertex $a$ (resp., vertex $b$) and an area-complete leaf (resp., the ill vertex) in the Type A tree. Apply procedure CEA to the resultant area graph. Then, if $|K|$ is even, add an edge between $v^*$ and a vertex in the Type A tree.

**Case7:** Apply the procedure of Case7. If $(G, \mathcal{W})$ satisfies the Condition D, add an extra edge between a vertex in $V(\hat{T}) \cup V(F)$ and a vertex in $V(G) - (V(\hat{T}) \cup V(F))$.

**Case8:** When $|K|$ is odd or when $|K|$ is even and the Condition E is satisfied: Let $a_0$ and $b_0$ be an area-complete leaf and the ill vertex in the Type A tree, respectively, and choose $a_i$ and $b_i$ in every Type B tree $H_i$ $(1 \leq i \leq t)$ in the similar manner to Case7. Add the cycle edges $(b_0, a_1)$, $(b_1, a_2)$, $\ldots$, $(b_{t-1}, a_t)$, $(b_t, a_0)$. Then, apply procedure CEA to the resultant graph.

When $|K|$ is even and the Condition E is not satisfied: If $d(V(\hat{T}) \cup V(F); G) \geq 1$, add the cycle edges to the trees and choose an ill vertex $v^*$; otherwise, let $v^*$ be an ill vertex included in a Type B tree with three or more ill vertices such that $V(\hat{H}) \cup V(F) - \{v^*\}$ is area-complete, and add the cycle edges to the trees induced by $V(\hat{H}) \cup V(F) - \{v^*\}$ from the Type B trees. Then, when the tree representation of the resultant graph is a Type A tree (resp., a Type B tree), apply the procedure of Case3 (resp., apply procedure CEA to the resultant graph and add an edge between $v^*$ and the ill vertex in the Type A tree).

**Case9:** Make pairs of the Type A trees as many as possible, and then add an edge between two ill vertices in each pair. If the number of the Type A trees is even (resp., odd), apply the procedure of Case7 (resp., Case8) to the remaining area graph.

**Proof of Theorem 2.3 :** As $F \neq \emptyset$, we can exclude Case1, Case2, and Case3.

**Case4:** There are no Type B trees except $F$. Therefore, $(G, \mathcal{W})$ is a 1-NA-edge-connected area graph. By the same proof as that in the section 4, the resultant area graph becomes a 2-NA-edge-connected area graph and the number of the added edges is the smallest.

**Case5:** The 2-edge-connected component including all the ill vertices becomes area-complete, as an area-complete leaf is included. Therefore, the resultant area graph becomes a 2-NA-edge-connected area graph. If $|K|$ is odd (resp., even), the number of the added edges is $2 + (|K| - 3)/2 = \lceil |K|/2 \rceil$ (resp., $2 + \lceil (|K| - 3)/2 \rceil = |K|/2 + 1$). Therefore, the number of the added edges is the smallest.

**Case6:** After an edge between the Type B tree in $F$ and each of two Type A trees is added, the Type B tree in $F$ becomes a 2-NA-edge-connected area graph, an area-complete Type A, or an area-complete Type B tree. Therefore, by the same proof as Case6, the resultant area graph becomes a 2-NA-edge-connected area graph. The number of the added edges is $\lceil |K|/2 \rceil$, which is the smallest.

**Case7:** If there is no Type B tree except $F$, $V(F)$ must be area-complete, because $(G, \mathcal{W})$ is composed of some Type B trees in $F$ and they have all areas in $\mathcal{W}$. Therefore, by the procedure, the resultant area graph becomes a 2-NA-edge-connected area graph. If there are some Type B trees except $F$ and $(G, \mathcal{W})$ does not satisfy the Condition D, $V(\hat{T}) \cup V(F)$ is area-complete, $|K|$ is odd, or $d(V(\hat{T}) \cup V(F); G) \geq 2$. Therefore, by procedure CEA, $V(\hat{T}) \cup V(F)$ satisfies the conditions of Theorem 4.1. If there are some Type B trees except $F$ and $(G, \mathcal{W})$ satisfies the Condition D, the cut size of $V(\hat{T}) \cup V(F)$ becomes two by the extra edge, and the conditions of Theorem 4.1 are satisfied. If the area graph satisfies the Condition D, the number of the added edges is $|K|/2 + 1$, which is the smallest; otherwise, $\lceil |K|/2 \rceil$ edges are added, which is also the smallest.

**Case8:** When $|K|$ is odd or when $|K|$ is even and the Condition E is satisfied: the 2-edge-connected component including all the ill vertices becomes area-complete, as an area-complete leaf is included. Therefore, the resultant area graph becomes a 2-NA-edge-connected area graph. When $|K|$ is even and the Condition E is satisfied, the number of the added edges is $2 + \lceil (|K| - 3)/2 \rceil = |K|/2 + 1$; when $|K|$ is odd, $2 + \lceil (|K| - 3)/2 \rceil = \lceil |K|/2 \rceil$.

Therefore, the number of the added edges is the smallest.
When $|K|$ is even and the Condition E is not satisfied: If $d(V(\hat{T}) \cup V(F); G) \geq 1$, the cut size of the 2-edge-connected component including all the ill vertices in $V(\hat{T}) \cup V(F) - \{v^*\}$ becomes two. If $d(V(\hat{T}) \cup V(F); G) = 0$, the 2-edge-connected component including all the ill vertices in $V(\hat{T}) \cup V(F) - \{v^*\}$ becomes area-complete. Therefore, the resultant area graph becomes a 2-NA-edge-connected area graph. The number of the added edges is $1 + (|K| - 2)/2 = |K|/2$, which is the smallest.
**Case9:** The resultant area graph obviously becomes a 2-NA-edge-connected area graph, and the number of the added edges is the smallest.

For the computational complexity, $O(pn + m)$ time is obviously necessary in total to execute the procedure.

∎

## 6. Conclusions

In this paper, we investigated the problem to augment a given area graph to an area graph with a given NA-edge-connectivity by adding the smallest number of new edges. As a result, we proved the NP-completeness of the problem which determines whether we can augment an area graph to a 1-NA-edge-connected area graph by adding a given number or less new edges. In addition, we proved that the problem of augmenting a 1-NA-edge-connected area graph or a 0-NA-edge-connected area graph to be a 2-NA-edge-connected area graph by adding the smallest number of edges can be solved in polynomial time.

We have no results for the NA-vertex-connectivity augmentation problem. These are remained for the future works.

## Acknowledgement

The authors wish to express our gratitude for the guidance and encouragement from Emeritus Professor Toshihide Ibaraki of Kyoto University. They also would like to thank anonymous reviewers for the valuable and helpful comments.

This study is supported by Grant-in-Aid for Young Scientists (B) (No.15700020).

## References

[1] K. P. Eswaran and R. E. Tarjan: Augmentation problems. *SIAM Journal of Computing*, **5** (1976), 653-665.

[2] A. Frank: Augmenting graphs to meet edge-connectivity requirements. *SIAM Journal of Discrete Mathematics*, **5** (1992), 25-53.

[3] M. Garey and D. Johnson: *Computers and Intractability* (W. H. Freeman and Company, San Francisco, 1978).

[4] T. Hsu: Undirected vertex-connectivity structure and smallest four-vertex-connectivity augmentation. *Lecture Notes in Computer Science*, **920** (1995), 274-283.

[5] T. Hsu and V. Ramachandran: Finding a smallest augmentation to biconnect a graph. *SIAM Journal of Computing*, **22** (1993), 889-912.

[6] T. Ishii, Y. Akiyama, and H. Nagamochi: Minimum augmentation of edge-connectivity between vertices and sets of vertices in undirected graphs. *Electronic Notes in Theoretical Computer Science*, **78** (2003), 1-24.

[7] H. Ito: Node-to-area connectivity of graphs. *Transactions of the Institute of Electrical Engineers of Japan*, **11C-4** (1994), 463-469.

[8] H. Ito: Node-to-area connectivity of graphs. M. Fushimi and K. Tone (eds.): *Proceedings of APORS94* (World Scientific Publishing, 1995), 89-96.

[9] H. Ito, M. Ito, Y. Itatsu, K. Nakai, H.Uehara, and M. Yokoyama: Source location problems considering vertex-connectivity and edge-connectivity simultaneously. *Networks*, **40** (2002), 63-70.

[10] H. Ito and M. Yokoyama: Edge connectivity between nodes and node-subsets. *Networks*, **31-3** (1998), 157-164.

[11] H. Miwa and H. Ito: Sparse spanning subgraph preserving connectivity and distance between vertices and vertex subsets. *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, **E81-A-5** (1998), 832-841.

[12] H. Miwa and H. Ito: Edge augmenting problems for increasing connectivity between vertices and vertex subsets. *Technical Report of IEICE*, **99-8** (1999), 17-24.

[13] H. Nagamochi and T. Ibaraki: A faster edge splitting algorithm in multigraphs and its application to the edge-connectivity augmentation problem. E. Balas and J. Clausen (eds.): *Lectures Notes in Computer Science*, **920** (1995), 401-413.

[14] H. Nagamochi and T. Ibaraki: Augmenting edge-connectivity over the entire range in $\tilde{O}(nm)$ time. *Proceedings of 8th Annual ACM-SIAM Symposium on Discrete Algorithms* (New Orleans, 1997), 649-658.

[15] D. Naor, D. Gusfield, and C. Martel: A fast algorithm for optimally increasing the edge-connectivity. *Proceedings of 31st Annual IEEE Symposium on Foundations of Computer Science* (St. Louis, 1990), 692-707.

[16] T. Watanabe and A. Nakamura: Edge-connectivity augmentation problems. *Journal of Computer and System Sciences*, **35** (1987), 96-144.

[17] T. Watanabe and A. Nakamura: A smallest augmentation to 3-connect a graph. *Discrete Applied Mathematics*, **28** (1990), 183-186.

Hiroyoshi Miwa
Kwansei Gakuin University
2-1, gakuen, Sanda-shi
Hyogo 669-1337, Japan
E-mail: `miwa@ksc.kwansei.ac.jp`