グラフの指定点集合に対する k 辺連結化問題

田岡智志 渡辺敏正

広島大学工学部第二類(電気系)
724 東広島市鏡山一丁目 4-1
(電話) 0824-24-7662 [渡辺]
(ファクシミリ) 0824-22-7195
(電子メール) watanabe@huis.hiroshima-u.ac.jp

アブストラクト
指定点に対する k 辺連結化問題 (kECA-SV と略記) とは，"グラフ $G=(V,E)$, コスト関数 $c: V\times V\rightarrow Z^+$ (非負整数) と部分集合 $\Gamma\subseteq V$ が与えられた時，$G'=(V,E\cup E')$ において，Γ 中のすべての 2 点間に少なくとも k 本の辺を共有しないパスが存在するような最小コスト辺集合 E' を求めよ，" と定義される．重みなしのこの問題を UW-kECA-SV で表す．
本稿では UW-$(\lambda+1)$ECA-SV に対する $O(\lambda^2|V|(|V|+|\Gamma|\log\lambda)+|E|)$ 時間のアルゴリズムを提案する．ただし，λ とは Γ の辺連結度 (Γ 中の 2 点を分離する最小カットの辺数) である．また，λ が G の辺連結度に等しい場合における $O(|V|\log|V|+|E|)$ 時間のアルゴリズムも提案する．

キーワード
辺連結化問題，辺連結度，指定点集合，多項式時間アルゴリズム

Smallest Augmentation to k-Edge-Connect All Specified Vertices in a Graph

Satoshi Taoka and Toshimasa Watanabe
Department of Circuits and Systems, Faculty of Engineering, Hiroshima University,
4-1 Kagamiyama, 1 Chome, Higashi-Hiroshima, 724 Japan
Phone: +81-824-24-7662 (Watanabe) Facsimile: +81-824-22-7195
E-mail: watanabe@huis.hiroshima-u.ac.jp

Abstract
The *k-edge-connectivity augmentation problem for a specified set of vertice* (kECA-SV for short) is defined by "Given a graph $G=(V,E)$, a cost function $c: V\times V\rightarrow Z^+$ (nonnegative integers) with $V\times V=\{\{u,v\}|u,v\in V, u\neq v\}$ and a subset $\Gamma\subseteq V$, find a minimum-cost set E' of edges, each connecting distinct vertices of V, such that $G'=(V,E\cup E')$ has at least k edge-disjoint paths between any pair of vertices in Γ." The ***unweighted version*** of the problem is denoted by UW-kECA-SV.
We propose an $O(\lambda^2|V|(|V|+|\Gamma|\log\lambda)+|E|)$ algorithm for UW-$(\lambda+1)$ECA-SV with $\Gamma\subset V$, where λ is edge-connectivity of Γ (the cardinality of a minimum cut separating two vertices of Γ). In a special case, we also propose an $O(|V|\log|V|+|E|)$ algorithm when λ is equal to the edge-connectivity of G.

Key word
connectivity augmentation problems, edge-connectivity, specified vertex set, polynomial-time algorithms,

1 Introduction

The *k-edge-connectivity augmentation problem for a specified set of vertices* (kECA-SV for short) is defined by "Given a graph $G = (V, E)$, a cost function $c : V \times V \to Z^+$ (nonnegative integers) with $V \times V = \{\{u, v\} | u, v \in V, u \neq v\}$ and a subset $\Gamma \subseteq V$, find a minimum-cost set E' of edges, each connecting distinct vertices of V, such that $G' = (V, E \cup E')$ has at least k edge-disjoint paths between any pair of vertices in Γ." Such an edge set E' is called a *minimum solution* to the problem, and we may assume $|\Gamma| \geq 2$. G' is also written as $G + E'$. Costs $c(\{u, v\})$ for $\{u, v\} \in V \times V$ is denoted as $c(u, v)$ for simplicity. The problem is called the *weighted version*, denoted by W-kECA-SV, if there may exist some distinct edge costs and the *unweighted one*, denoted by UW-kECA-SV, otherwise.

Let kECA-SV(*,**) denote kECA-SV with the following restriction (i) and (ii) on G and A', respectively: (i) * is set to S if G is required to be simple, and * means G may be a multiple graph; (ii) ** is set to MA if increase in edge multiplicity in constructing G' is allowed, and is set to SA otherwise.

If $\Gamma = V$ then the problem is called the k-edge-connectivity augmentation problem (denoted as kECA). UW-kECA(*,MA) have been mainly discussed in literature: see [1] for UW-2ECA(*,MA) and [26, 24] for UW-3ECA(*,MA) and [3, 4, 12, 17, 22] for UW-kECA(*,MA) with $k \geq 4$. Concerning UW-kECA, only UW-kECA(*,MA) has been discussed so far. The fastest algorithm for UW-kECA(*,MA) is the one proposed in [12], and its time complexity is $O(\delta^2|V||E| + |V|\phi(|V|, |E|))$ time, where δ is the increase of edge-connectivity of G and $\phi(|V|, |E|)$ is the time-complexity to find local edge-connectivity between some two vertices of V.

[20, 21] ([14, 23], respectively) show that there is an $O(|V| + |E|)$ algorithm for solving UW-kECA-SV(*,MA) with $k = 2$ ($k = 3$). These results show that UW-kECA-SV(*,MA) with $k = 2$ ($k = 3$) can be equivalently transformed into UW-2ECA(*,MA) (UW-3ECA(*,MA)) in $O(|V| + |E|)$ time. Since it is known that UW-2ECA(*,MA) has an $O(|V| + |E|)$ algorithm in [1] and UW-3ECA(*,MA) has an an $O(|V| + |E|)$ algorithm (by combining results [5, 9, 16] and [24, 26]; see also [25]), UW-kECA-SV(*,MA) with $k = 2$ ($k = 3$) can be solved in linear time. It should be mentioned that UW-kECA-SV(*,SA) with $k = 2$ ($k = 3$) can be solved similarly to the paper: the former is optimally solved since UW-kECA(*,SA) and UW-kECA(*,MA) with $k = 2$ ($k = 3$) have the same minimum solution if $|V| \geq 4$ (see [25]).

The subject of the paper is UW-$(\lambda + 1)$ECA-SV(*,MA) for a general nonnegative integer k, where $\lambda(\Gamma; G)$ is edge-connectivity of Γ, and $\lambda(\Gamma; G)$ is denoted as λ for simplicity, where $\lambda(\Gamma; G)$ is defined in Section 2.1. The paper shows that there is an $O(\lambda^2|V|(|V| + |\Gamma| \log \lambda) + |E|)$ ($O(|V| \log |V| + |E|)$, respectively) algorithm for solving UW-$(\lambda + 1)$ECA-SV(*,MA) if $\lambda(\Gamma; G) > \lambda(V; G)$ (if $\lambda(\Gamma; G) = \lambda(V; G)$). For UW-$k$ECA(*,MA), the algorithm proposed in [12] utilizes a structural graph, proposed in [6], which simply represents all minimum cuts of G, and an extreme set tree is used to find a minimum solution. For UW-kECA-SV(*,MA), we do not use a structural graph or an extreme set tree from the following reasons. A structural graph represents all minimum cuts of G, so, if $\Gamma \subset V$ and $\lambda(V; G) < \lambda(\Gamma; G)$ then a structural graph fails to represent some minimum cuts that have to be checked when we $(\lambda + 1)$-edge-connect Γ of G. The edge (n_1, n_2) of $F(G)$ in Fig. 2 represents a minimum cut $(\{1\}, V - \{1\})$ in G of Fig. 1. However it is other λ-cuts, say $(\{1, 2\}, V - \{1, 2\})$, that is required to be checked to $(\lambda + 1)$-edge-connected Γ. [12] introduced a k-extreme set such that $U \subset V$ is a k-extreme set if and only if $d(U) = k$ and $d(W) > k$ for any $W \subset U$. In Fig. 1, $\{1\}$ is a 1-extreme set and $\{2\}$ is a 3-extreme set. In UW-3ECA for G of Fig. 1, two new edges have to be incident upon a vertex 1, while a vertex 2 required no new edges to be incident upon it. On the other hand, in UW-3ECA-SV, we have only to add one edge whoes endvertex is either 1 or 2 in G of Fig. 1. Due to lack of such information, we do not use an extreme set for UW-$(\lambda + 1)$ECA-SV. Instead of using $F(G)$, we adopt an operation called edge-interchange proposed in [18, 19] a vertex set called k-palm is introduced: the definition will be given in Section 2.2. A structural graph, however, can be used for UW-kECA in case $\lambda(\Gamma; G) = \lambda(V; G)$. So, by using a structural graph, we propose in Section 5 an $O(|V| \log |V| + |E|)$ algorithm for UW-$(\lambda + 1)$ECA-SV(*,MA) with $\lambda(\Gamma; G) = \lambda(V; G)$.

The result of this papre is the first step for UW-kECA-SV(*,MA) with $k = \lambda + \delta$ and $\delta \leq 1$. In the rest of the paper UW-kECA-SV(*,MA) is simply denoted as UW-kECA-SV.

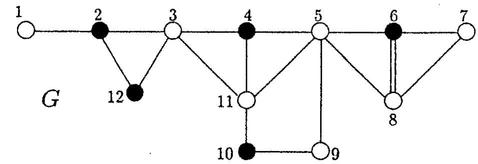

Figure 1: G is a given graph with $\lambda(V; G) = 1$ and $\lambda(\Gamma; G) = 2$. Each of black dots $\{2, 4, 6, 10, 12\}$ is a specified vertex. $\{(2, 6), (10, 12)\}$ is a solution of G for UW-3ECA-SV and $\{(1, 7), (10, 12)\}$ is also a solution.

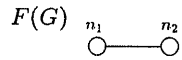

Figure 2: $F(G)$ is a structural graph of G. n_1 (n_2, respectively) corresponds a vertex set $\{1\}$ ($\{2, \ldots, 12\}$). A edge (n_1, n_2) of $F(G)$ represents a minimum cut $(\{1\}, V - \{1\})$ of G of Fig. 1.

2 Preliminaries

2.1 Basic Definitions

An *undirected graph* $G = (V(G), E(G))$ consists of a finite and nonempty set of vertices $V(G)$ and a finite set of undirected edges $E(G)$; an edge e incident upon two vertices u, v is denoted by (u, v); u and v are the *endvertices* of an edge e; e is called a loop if $u = v$. $V(G)$ and $E(G)$ are often denoted as V and E, respectively. If there are two edges both of which have the same pair of endvertices then G is called a *multigraph*. Such edges are called *multiple edges*; otherwise G is called a *simple graph*. In this paper, only graphs without loops are considered, and the term "a graph" means an undirected multigraph unless otherwise stated.

For a set E' of edges such that $E' \cap E(G) = \emptyset$, let $G + E'$ denote the graph $(V(G), E(G) \cup E')$. If $E' = \{e\}$ then we denote $G + e$.

A *path* between u and v, or a *(u, v)-path*, is an alternating sequence of vertices and edges $u = v_0, e_1, v_1, \ldots, v_{n-1}, e_n, v_n = v$ ($n \geq 0$) such that if $n \geq 1$ then $v_0, \ldots, v_n$ are all distinct and $e_i = (v_{i-1}, v_i)$ for each i, $1 \leq i \leq n$. The *length* of this path is n. Vertices $v_1, \ldots, v_{n-1}$ are called *inner vertices* of this path if $n \geq 2$. A cycle is a (v_0, v_n)-path together with an edge (v_0, v_n). The length of this cycle is $n + 1$. A pair of multiple edges are considered as a cycle of length two.

Two paths P, P' are said to be *edge-disjoint* (*internally disjoint*, respectively) if $E(P) \cap E(P') = \emptyset$ (P and P' have no inner vertex in common). Let $\lambda(u, v; G)$ ($\kappa(u, v; G)$, respectively) denote the maximum number of edge-disjoint (internally disjoint) (u, v)-paths of G. For a subset $\Gamma \subseteq V$, *edge-connectivity of* Γ (*vertex-connectivity*), denoted by $\lambda(\Gamma; G)$ ($\kappa(\Gamma; G)$), is the minimum number of $\lambda(v, w; G)$ ($\kappa(u, v; G)$) for $v, w \in \Gamma$. The *edge-connectivity* (*vertex-connectivity*, respectively) of a graph G, denoted by $\lambda(G)$ ($\kappa(G)$), is $\lambda(V; G)$ ($\kappa(V; G)$). A graph G is *k-edge-connected* (*k-vertex-connected*) if and only if $\lambda(G) \geq k$ ($\kappa(G) \geq k$). A *k-edge-connected component* (*vertex-connected component*, respectively) of G is a maximal subset of vertices such that $\lambda(u, v; G)$ ($\kappa(u, v; G)$) for any two vertices in the set. A k-edge-connected component is often denoted as a k-component in this paper unless any confusion arises. It is known that $\lambda(G) \geq k$ ($\kappa(G) \geq k$, respectively) if and only if $V(G)$ is a k-edge-connected component (a k-vertex-connected component). Note that distinct k-edge-connected component are disjoint sets. Each 1-edge-connected component is often called a *component*. A set $K \subseteq E(G)$ is called a *(u, v)-separator* if and only if u and v belong to distinct components of $G' = (V, E - K)$. A (u, v)-separator $K \subseteq E$ is called a *(u, v)-cut* if and only if $|K| = \lambda(u, v; G)$. For nonempty disjoint sets $S, S' \subset V(G)$, let $(S, S'; G) = \{(u, v) \in E(G) | u \in S \text{ and } v \in S'\}$, where it is often written as (S, S') if G is clear from the context. If $S' = V(G) - S$ then

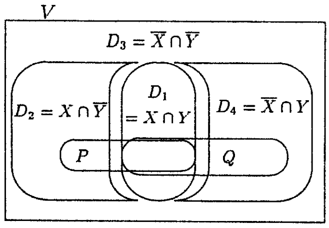

Figure 3: The four disjoint sets D_i $(1 \leq i \leq 4)$ in Lemma 2.2.

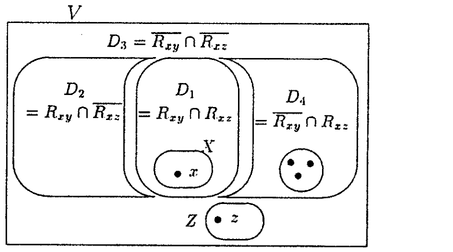

Figure 4: The four disjoint sets D_i $(1 \leq i \leq 4)$ in Proposition 2.2.

we denote $d(S,G) = |(S, V-S; G)|$. $d_G(S)$ is called the *degree* of a vertex set S in G. If $S = \{v\}$ then we denote $d_G(v)$ for simplicity and $d_G(v)$ is call the degree of a vertex v in G. For a cutpoint v of G, let $X_1, \ldots, X_k$ $(k \leq 2)$ be 1-components of $G - v$. Each subset $X_i \cup \{v\}$ is called a v-block of G.

2.2 λ-palms with respect to Γ

A class of k-components of G is denoted by $\mathrm{EC}(k; G)$ or $\mathrm{EC}(k)$, and a class of k-components containing some vertices of Γ is denoted by $\mathrm{EC}_\Gamma(k; G)$ or $\mathrm{EC}_\Gamma(k)$. Let $\lambda = \lambda(\Gamma; G)$ in the rest of the paper.

Definition 2.1 *$X \subseteq V$ is a λ-palm of G with respect to Γ if and only if the following (1) and (2) hold,*

(1) $|(X, \overline{X})| = \lambda$, $X \cap \Gamma \neq \emptyset$ and $\overline{X} \cap \Gamma \neq \emptyset$;

(2) any $Y \subset V$ with $Y \cap \Gamma \neq \emptyset$ and $Y \cap \Gamma \subset X \cap \Gamma$ has $|(Y, \overline{Y})| > \lambda$. □

In the following "with respect to Γ" is often omitted. A vertex set $X \cap \Gamma$ of a λ-palm X is called a *core* of this λ-palm.

Lemma 2.1 *[3] For distinct two sets $X, Y \subset V$, we have $d(X) + d(Y) \geq d(X-Y) + d(Y-X)$ and $d(X) + d(Y) \geq d(X \cap Y) + d(X \cup Y)$.*

Lemma 2.2 *Suppose that X and Y are distinct λ-palms with respect to Γ. If P or Q is a core of X or Y, respectively, then either $Q = P$ or $P \cap Q = \emptyset$ holds.*

(Proof) By supposing $P \cap Q \neq \emptyset$, we will show $P = Q$. We have $(P = Q)$ or $(P - Q \neq \emptyset$ and $Q - P \neq \emptyset)$ since if $P \subset Q$ $(Q \subset P$, respectively) then X (Y) is not a λ-palm. Assume that $P \neq Q$. Then we have two cases $X \cup Y = V$ and $X \cup Y \subset V$.

V can be partitioned into four sets $D_1 = X \cap Y$, $D_2 = X \cap \overline{Y}$, $D_3 = \overline{X} \cap \overline{Y}$ and $D_4 = \overline{X} \cap Y$, where $D_3 = \emptyset$ may hold (see Fig. 3). We have $D_i \cap \Gamma \neq \emptyset$, $i = 1, 2, 4$.

$d(D_2) > \lambda$ $(d(D_4) \geq \lambda$, respectively) since X (Y) is a palm so we have $d(D_2) + d(D_4) > 2\lambda$. From Lemma 2.1, we have $d(D_1 \cup D_2) + d(D_1 \cup D_4) \geq d(D_2) + d(D_4)$. This contradicts the fact that $d(D_2) + d(D_4) > 2\lambda$ and $d(D_1 \cup D_4) + d(D_2 \cup D_4) = 2\lambda$. □

From Lemma 2.2, there is a unique maximal class of subsets $C_i \subset \Gamma$ with $1 \leq i \leq q$ such that each C_i is a core of a λ-palm and $C_i \cap C_j = \emptyset$ if $i \neq j$. Let D_i be a maximal class of λ-palms containing C_i, $X_i \in D_i$ be a representative of D_i, and $\mathrm{PE}_\Gamma(G) = \{X_i | 1 \leq i \leq q\}$. For G of Fig. 1, $\mathrm{PE}_\Gamma(G) = \{\{1,2\}, \{6,7,8\}, \{10\}, \{12\}\}$.

We consider how to find $\mathrm{PE}_\Gamma(G)$ in the rest of this section. A network $N = (V, \vec{E}, c)$ of $G = (V, E)$ is defined by the following,

$$\vec{E} = \{\langle v, w\rangle, \langle w, v\rangle | (v, w) \in E\},$$

$$c : \vec{E} \to \{1\},$$

where $\langle v, w\rangle$ is a directed edge from v to w. For N, a flow $f : \vec{E} \to \{0, 1\}$ of $x, y \in V$ is defined by the following:

$$\sum_{\forall\langle v,w\rangle \in \vec{E}} f(\langle v,w\rangle) - \sum_{\forall\langle w,v\rangle \in \vec{E}} f(\langle w,v\rangle) \begin{cases} = 0 & \text{if } v \in V - \{x, y\}, \\ \geq 0 & \text{if } v = x, \\ \leq 0 & \text{if } v = y, \end{cases}$$

and

$$f(e) \leq c(e) \text{ for all } e \in \vec{E}.$$

A value of a flow f is defined by

$$\begin{aligned} |f| &= \sum_{\forall\langle x,w\rangle \in \vec{E}} f(\langle x,w\rangle) - \sum_{\forall\langle w,x\rangle \in \vec{E}} f(\langle w,x\rangle) \\ &= \sum_{\forall\langle y,w\rangle \in \vec{E}} f(\langle y,w\rangle) - \sum_{\forall\langle w,y\rangle \in \vec{E}} f(\langle w,y\rangle). \end{aligned}$$

A flow of x and y, whose value is maximum, is called a *maximum flow*. Let f be a maximum flow of x, y in the following. $\tilde{E}^f_{xy}$ is a set of directed edges each of which is either $\langle w, v\rangle \in \tilde{E}^f_{xy}$ if $f(\langle v, w\rangle) = 1$ for $\langle v, w\rangle \in \vec{E}$ or $\langle v, w\rangle \in \tilde{E}^f_{xy}$ if $f(\langle v, w\rangle) = 0$ for $\langle v, w\rangle \in \vec{E}$. Let $\tilde{G}^f_{xy} = (V, \tilde{E}^f_{xy})$. Let $R_{xy} \subseteq V$ be a maximal set of vertices each of which is reachable from x in $\tilde{G}^f_{xy}$. Then $(R_{xy}, \overline{R_{xy}})$ is a $\lambda(x, y; G)$-cut [13]. $(R_{xy}, \overline{R_{xy}})$ is called a *nearest $\lambda(x, y; G)$-cut of X* with respect to x, y, and denoted by NC_{xy}. Note that NC_{xy} is not uniquely. We obtain the following proposition by the definition of λ-palms and $(\lambda + 1)$-components.

Proposition 2.1 *A set $S \subset V$ is a λ-palm if and only if $d(S) = \lambda$, S is a union of some $(\lambda + 1)$-components and has exactly one $(\lambda + 1)$-component containing a vertex of Γ.* □

Proposition 2.2 *Suppose $x \in X \cap \Gamma$ and $y \in Y \cap \Gamma$ for distinct two $(\lambda + 1)$-components X, Y. Then $(R_{xy} - X) \cap \Gamma = \emptyset$ for G if and only if $(R_{xz} - X) \cap \Gamma = \emptyset$ for $\forall z \in \Gamma - X$.*

(Proof) We will only show necessity since sufficiency clearly holds. We will show a contradiction by assuming that $(R_{xy} - X) \cap \Gamma = \emptyset$, while there is $z \in \Gamma - (X \cup Y)$ such that $(R_{xz} - X) \cap \Gamma \neq \emptyset$. Let Z be a $(\lambda+1)$-component with $z \in Z$. First we will show that $D_i \neq \emptyset$ $(1 \leq i \leq 4)$, $x \in D_1$ and $y \in D_3$ if we set $D_1 = R_{xy} \cap R_{xz}$, $D_2 = R_{xy} \cap \overline{R_{xz}}$, $D_3 = \overline{R_{xy}} \cap \overline{R_{xz}}$ and $D_4 = \overline{R_{xy}} \cap R_{xz}$ (see Fig 4). Each of X, Y and Z cannot be partitioned into more two distinct sets D_i since X, Y and Z are $(\lambda + 1)$-components, and $(R_{xy}, \overline{R_{xy}})$ and $(R_{xz}, \overline{R_{xz}})$ are λ-cuts. Clearly $x \in D_1$, $z \in D_2 \cup D_3$ and $y \in D_3 \cup D_4$. $D_1 \cap \Gamma = X \cap \Gamma$ and $z \in D_3$, since $(R_{xz} - X) \cap \Gamma \neq \emptyset$. We have $D_4 \cap \Gamma \neq \emptyset$ from $(R_{xz} - X) \cap \Gamma \neq \emptyset$ and $D_1 \cap \Gamma = X \cap \Gamma$. If we assume $D_2 = \emptyset$ then we obtain a contradiction that $(R_{xy}, \overline{R_{xy}})$ is NC_{xz} with $R_{xy} \subset R_{xz}$. Hence $D_2 \neq \emptyset$.

We set $d_{12} = |(D_1, D_2)|$, $d_{13} = |(D_1, D_3)|$ $d_{14} = |(D_1, D_4)|$, $d_{23} = |(D_2, D_3)|$, $d_{24} = |(D_2, D_4)|$ and $d_{34} = |(D_3, D_4)|$ (see Fig. 4). Then, for $(R_{xy}, \overline{R_{xy}})$ and $(R_{xz}, \overline{R_{xz}})$, we obtain

$$d_{14} + d_{23} + d_{24} + d_{13} = \lambda, \quad d_{12} + d_{34} + d_{24} + d_{13} = \lambda, \tag{2.1}$$

and

$$d_{14} + d_{23} = d_{12} + d_{34}. \tag{2.2}$$

Since $x \in D_1$ and $(D_1, \overline{D_1})$ is neither NC_{xy} nor NC_{xz},

$$d_{12} + d_{14} + d_{13} > \lambda. \tag{2.3}$$

Since $\lambda = \lambda(\Gamma; G)$ and $D_3 \cap \Gamma \neq \emptyset$,

$$d_{23} + d_{34} + d_{13} \geq \lambda. \tag{2.4}$$

Hence

$$\begin{array}{rcl} d_{12}+d_{14}+d_{23}+d_{34}+2d_{13} & > & 2\lambda, \text{ (by (2.3) and (2.4))} \\ d_{14}+d_{23}+d_{13} & > & \lambda, \text{ (by (2.2))} \\ \lambda - d_{24} & > & \lambda, \text{ (by (2.1))} \\ 0 & > & d_{24}, \end{array}$$

a contradiction. □

Remark 2.1 *[10] introduced a sparse graph $G^{(i)} = (V, E_1 \cup E_2 \cdots E_i)$ for a given graph $G = (V, E)$ such that the following (1) through (3) hold for any $u, v \in V$:*

(1) $\lambda(u, v; G^{(i)}) = i$ if $\lambda(u, v; G) \geq i$;

(2) $\lambda(u, v; G^{(i)}) = \lambda(u, v; G)$ if $\lambda(u, v; G) < i$;

(3) $E_i \subseteq E$ and $|E_i| \leq |V|$,

where let (V, E_i) be recursively a maximal forest of $(V, E - E_1 \cup E_2 \cup \cdots \cup E_{i-1})$. [10] showed that E_i $(1 \leq i \leq |E|)$ can be obtained in $O(|V| + |E|)$ time. By utilizing the result in [2], checking whether $\lambda < \lambda(v, w; G)$ or not can be done in $O(\lambda^2|V|)$ time for each pair $v, w \in V$.

Proposition 2.3 *All λ-palms of G can be found in $O(\lambda^2|\Gamma||V|)$ time if λ and all $(\lambda+1)$-components are available.*

(Proof) For each $(\lambda+1)$-component X with $X \cap \Gamma \neq \emptyset$ and $x \in X \cap \Gamma$, we obtain a nearest λ-cut $(R_{xy}, \overline{R_{xy}})$ by a maximum flow of value λ from x to some $y \in \Gamma - X$. Then, by Proposition 2.2, if $(R_{xy} - X) \cap \Gamma = \emptyset$ then R_{xy} is a λ-palm, otherwise it is not. So we can find $\mathrm{PE}_\Gamma(G)$ in $O(\lambda^2|\Gamma||V|)$ time by Remark 2.1, since at most $|\Gamma|$ nearest λ-cuts may be found. □

3 Augmentation by Edge-Interchange

We explain an operation called edge-interchange which was originally introduced in [18, 19] for an efficient augmentation. It is also used in [15]. Let $\mathrm{PE}_\Gamma(G) = \{Y_1, \ldots, Y_q\}$ and choose $y_i \in Y_i \cap \Gamma$ as a representative of $Y_i \cap \Gamma$. Let

$$Y = \{y_i \in \Gamma | Y_i \in \mathrm{PE}_\Gamma(G)\}, \quad q \geq 2,$$

and let $r = \lceil q/2 \rceil$. We denote $V(e) = \{u, v\}$ for an edge $e = (u, v)$ and $V(F) = \bigcup_{e \in F} V(e)$ for an edge set F.

We can easily prove the next proposition.

Proposition 3.1 *If there is an attachment E' for G such that $V(E') = Y \subseteq S$ for some $(\lambda+1)$-component S in $G + E'$ then $\Gamma \subseteq S$.* □

3.1 Attachments

In G, we have $d_G(Y_i) \geq \lambda$ and $\lambda(y_i, y_j; G) = \lambda$ for $\forall i, j$ $(i \neq j)$. We call F an *attachment* (for G) if and only if the following (1) through (4) hold:

(1) $V(F) \subseteq Y$,

(2) $F \cap E(G) = \emptyset$,

(3) $V(e) \neq V(e')$ $(\forall e, e' \in F, e \neq e')$, and

(4) F has at most one pair f, f' such that $|V(f) \cap V(f')| = 1$.

Let F be any attachment for G. For each $e = (u, v) \in F$, $G + F$ has a new $(\lambda+1)$-component, denoted by $\alpha(e, G + F)$, containing $V(e)$.

We will show that we can find a minimum attachment $Z(\lambda + 1) = \{e_1, \ldots, e_r\}$ such that $\lambda(\Gamma; G + Z(\lambda + 1)) = \lambda + 1$. Although there are two cases: $r = 1$ and $r \geq 2$, we discuss only the latter case in the following. (Note that if $r = 1$ then we immediately obtain the desired attachment F.)

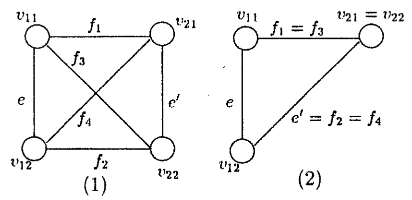

Figure 5: The edges e, e' and f_i, $1 \leq i \leq 4$. (1) $v_{21} \neq v_{22}$; (2) $v_{21} = v_{22}$.

3.2 Finding a minimum attachment

Suppose that there are an attachment F for G and vertices $v_{ij} \in Y - V(F)$, $1 \leq i, j \leq 2$, where v_{11}, v_{12}, v_{21} are distinct, and if v_{22} is equal to one of the other three then we assume that $v_{22} = v_{21}$ (see Fig. 5).

We use the following notations:

$$L = G + F, \quad e = (v_{11}, v_{12}), \quad e' = \begin{cases} (v_{21}, v_{22}) & \text{if } v_{21} \neq v_{22} \\ (v_{12}, v_{21}) & \text{if } v_{21} = v_{22} \end{cases}$$

$$\alpha(e) = \alpha(e, L + \{e, e'\}), \quad \alpha(e') = \alpha(e', L + \{e, e'\}),$$

$$f_1 = (v_{11}, v_{21}), \quad f_2 = (v_{12}, v_{22}), \quad f_3 = (v_{11}, v_{22}), \quad f_4 = (v_{12}, v_{21}),$$

where we set $f_1 = f_3$ and $e' = f_2 = f_4$ if $v_{21} = v_{22}$.

$$\alpha(f_i) = \begin{cases} \alpha(f_i, L + \{f_1, f_2\}) & \text{if } 1 \leq i \leq 2 \\ \alpha(f_i, L + \{f_3, f_4\}) & \text{if } 3 \leq i \leq 4 \end{cases}$$

(Note that $e, e', f_i \notin E(L), 1 \leq i \leq 4$.) We have two cases

Case I: $\alpha(e) \cap \alpha(e') = \emptyset$; Case II: $\alpha(e) \cap \alpha(e') \neq \emptyset$ (that is, $\alpha(e) = \alpha(e')$).

In Case I, we will show that there are two edges f, f' with $V(f) \cup V(f') = V(e) \cup V(e')$ such that

$$V(e) \cup V(e') \subseteq \alpha(f, L + \{f, f'\}) = \alpha(f', L + \{f, f'\}).$$

That is, we can add two edges so that the resulting $(\lambda+1)$-component contains $V(e) \cup V(e')$. Finding and adding such a pair of edges f, f' is called *edge-interchange* (with respect to $V(e_1) \cup V(e_2)$). On the other hand Case II considers $\alpha(e, L + e)$.

3.2.1 Case I: $\alpha(e) \cap \alpha(e') = \emptyset$.

Note that $v_{21} \neq v_{22}$ in this case. Let K be any fixed $(\alpha(e), \alpha(e'))$-cut of $L + \{e, e'\}$, and let B_i, $1 \leq i \leq 2$, denote the two sets of $L + \{e, e'\}$ such that $B_1 \cup B_2 = V$, $B_2 = V - B_1$, $K = (B_1; L + \{e, e'\})$, $\alpha(e) \subseteq B_1$ and $\alpha(e') \subseteq B_2$. $|K| = \lambda = \lambda(v_1, v_2; L'')$ for $\forall v_i \in B_i$, $1 \leq i \leq 2$, where L'' denotes L, $L + e$, $L + e'$ or $L + \{e, e'\}$. K is a (v_1, v_2)-cut of L. Suppose that f and f' satisfy either (i) or (ii):

(i) $f = f_1$, $f' = f_2$, or (ii) $f = f_3$, $f' = f_4$,

where $\{f, f'\} \cap E(L) = \emptyset$.

The next proposition shows a property of edge-interchange.

Proposition 3.2 *If $\alpha(e) \cap \alpha(e') = \alpha(f_1) \cap \alpha(f_2) = \emptyset$ then $\alpha(f_3) \cap \alpha(f_4) \neq \emptyset$, that is, $\alpha(f_3) = \alpha(f_4)$.*

(Proof) It is easy to see that $\lambda = |K| \geq 2$. Let K' be any fixed $(\alpha(f_1), \alpha(f_2))$-cut of $L + \{f_1, f_2\}$, where $|K'| = \lambda$ and $K' \neq K$. Let B'_i be the K'-block of $L + \{f_1, f_2\}$ such that $V(f_i) \subseteq B'_i$, $i = 1, 2$. Then $|K'| = \lambda = \lambda(v'_1, v'_2; L'')$ for $\forall v'_i \in B'_i$, $i = 1, 2$, where $L'' = L$, $L + f_1$ or $L + f_2$. K' is a (v'_1, v'_2)-cut of L. L has four disjoint sets $B_{ij} = B_i \cap B'_j$, $1 \leq i, j \leq 2$, such that $v_{ij} \in B_{ij}$ (see Fig. 6).

First we show that the proposition follows if we can prove the following (1) and (2):

(1) λ is even, $K \cap K' = \emptyset$ and there are partitions of K and K' such that

$$K = K_1 \cup K_2, K' = K'_1 \cup K'_2 \text{ with } |K_1| = |K_2| = |K'_1| = |K'_2|,$$

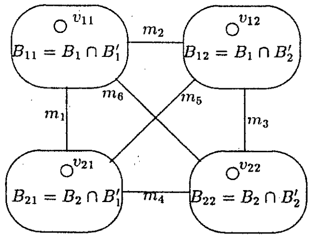

Figure 6: The four disjoint sets B_{ij}, $1 \le i, j \le 2$.

where

$$K_i = \{e_j \in K | V(e_j) \subseteq B'_i\},\ \ K'_i = \{e'_j \in K' | V(e'_j) \subseteq B_i\},\ \ i = 1, 2.$$

(2) L has λ edge-disjoint (v_{11}, v_{21})-paths P_i, $i = 1, \ldots, \lambda$ such that
  (i) $|E(P_i) \cap K_1| = 1$ if $1 \le i \le \lambda/2$;
  (ii) $|E(P_i) \cap K_2| = |E(P_i) \cap K'_j| = 1$ $(j = 1, 2)$ and $\{v_{12}, v_{22}\} \subset V(P_i)$ if $\lambda/2 + 1 \le i \le \lambda$.

Consider P_λ which consists of three subpaths: (v_{11}, v_{12})-subpath $P_{\lambda 1}$, (v_{12}, v_{22})-subpath $P_{\lambda 2}$ and (v_{22}, v_{21})-subpath $P_{\lambda 3}$. $L + \{f_3, f_4\}$ has two (v_{11}, v_{21})-paths P and P', where P (P', respectively) consists of $P_{\lambda 1}$ and f_4 (of f_3 and $P_{\lambda 3}$). It follows that $L + \{f_3, f_4\}$ has $(\lambda + 1)$ edge-disjoint (v_{11}, v_{21})-paths $P_1, \ldots, P_{\lambda-1}, P, P'$, and we have $\alpha(f_3) \cap \alpha(f_4) \neq \emptyset$.

Now we show that (1) and (2) hold. The similar idea is used in the proof of Lemma 3.2 of [22]. We partition K and K' as follows (see Fig. 6):

$$K = (B_{11}, B_{21}) \cup (B_{12}, B_{22}) \cup (B_{11}, B_{22}) \cup (B_{12}, B_{21}),$$

$$K' = (B_{11}, B_{12}) \cup (B_{21}, B_{22}) \cup (B_{11}, B_{22}) \cup (B_{12}, B_{21}).$$

Put

$$m_1 = |(B_{11}, B_{21})|,\ \ m_2 = |(B_{11}, B_{12})|,$$
$$m_3 = |(B_{12}, B_{22})|,\ \ m_4 = |(B_{21}, B_{22})|,$$
$$m_5 = |(B_{12}, B_{21})|,\ \ m_6 = |(B_{11}, B_{22})|.$$

Then

$$m_1 + m_3 + m_5 + m_6 = |K| = \lambda \text{ and } m_2 + m_4 + m_5 + m_6 = |K'| = \lambda.$$

Since $\lambda(u, v; L) = \lambda$ for any $u, v \in \{v_{11}, v_{12}, v_{21}, v_{22}\}$ $(u \neq v)$, we have

$$m_1 + m_2 + m_6 \ge \lambda,\ \ m_2 + m_3 + m_5 \ge \lambda,$$

$$m_3 + m_4 + m_6 \ge \lambda \ \text{ and } \ m_1 + m_4 + m_5 \ge \lambda.$$

It follows that

$$m_1 = m_2 = m_3 = m_4 = \lambda/2\ (\ge 1) \text{ and } m \text{ is even.}$$

Set

$$K_1 = (B_{11}, B_{21}),\ K_2 = (B_{12}, B_{22}),\ K'_1 = (B_{11}, B_{12}),\ K'_2 = (B_{21}, B_{22}),$$

and (1) follows. Let $P_1, \ldots, P_\lambda$ be λ edge-disjoint (v_{11}, v_{21})-paths of L, where we assume that

$$|E(P_i) \cap K_1| = 1 \ \text{ if } \ 1 \le i \le \lambda/2 \text{ , and}$$

$$|E(P_i) \cap K'_1| = |E(P_i) \cap K_2| = |E(P_i) \cap K'_2| = 1 \ \text{ if } \ 1 + \lambda/2 \le i \le \lambda.$$

L has λ edge-disjoint (v_{12}, v_{22})-paths P_i, $1 \le i \le \lambda$, and each of them has one (v_{12}, v)-subpath with $v \in (V(K_2) \cup V(K'_1)) \cap B_{12}$ and one (v_{22}, v')-subpath with $v' \in (V(K_2) \cup V(K'_2)) \cap B_{22}$. Hence L has λ edge-disjoint (v_{11}, v_{21})-paths P_i, $1 \le i \le \lambda$, as mentioned in (2). □

Corollary 3.1 *Let f_3, f_4 be the two edges of Proposition 3.2, $L' = L + \{f_3, f_4\}$ and f be either f_3 or f_4. Then $L' - f$ has no λ-cut separating $V(f_3)$ from $V(f_4)$. That is, if $L' - f$ has a λ-cut K separating a vertex of $V(f_3)$ from another one of $V(f_4)$ then K separates $\{u\}$ from $\{v\} \cup V(f')$ and $V(f')$ is not separated by K, where $V(f) = \{u, v\}$ and $\{f'\} = \{f_3, f_4\} - \{f\}$.* □

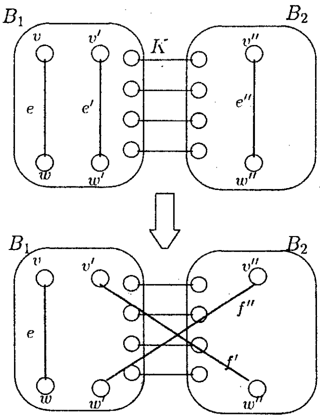

Figure 7: A situation for edges e, e', e'', f' and f'' in the case where $f' = (v', w'')$ and $f'' = (v'', w')$.

3.2.2 Case II: $\alpha(e) = \alpha(e')$.

Put

$$e = (v, w),\ e' = (v', w'),\ L' = L + e,$$

and suppose that there are distinct vertices $v'', w'' \in Y - (V(F) \cup V(e) \cup V(e'))$ such that

$$\alpha(e', L' + \{e', e''\}) \cap \alpha(e'', L' + \{e', e''\}) = \emptyset,$$

where $e'' = (v'', w'') \notin E(L' + e')$. By Proposition 3.2, there are edges f', f'' such that

$$\alpha(f', L' + \{f', f''\}) = \alpha(f'', L' + \{f', f''\}),$$

$$V(f') \cup V(f'') = V(e') \cup V(e'') \ \text{ and } \ V(f') \cap V(f'') = \emptyset.$$

We assume that $f' = (v', w'')$ and $f'' = (v'', w')$ (see Fig. 7). Then the next proposition follows from Corollary 3.1.

Proposition 3.3 $\alpha(e, L' + e') \subseteq \alpha(f', L' + \{f', f''\})$. □

Propositions 3.2 and 3.3 show that if $\lambda > 0$ then repeating edge-interchange finds a sequence of edges $e_1, \ldots, e_r$ $(r = \lceil q/2 \rceil \ge 1)$ such that

$$\alpha(e_i, H_i) \subseteq \alpha(e_{i+1}, H_{i+1}),\ \ 1 \le i \le r - 1,$$

$$V(e_{j-1}) \cap V(e_j) = \emptyset,\ \ 2 \le j \le r - 1, \text{ and}$$

$$V(e_{r-1}) \cap V(e_r) = \begin{cases} \emptyset & \text{if } q \text{ is even,} \\ \{y_q\} & \text{if } q \text{ is odd,} \end{cases}$$

where $H_0 = H$, and $H_{i+1} = H_i + e_{i+1}, 0 \le i \le r - 1$. Since $\alpha(e_r, H_r) = V(H)$ by Proposition 3.1, we obtain the following proposition.

Proposition 3.4 *$Z_H = \{e_1, \ldots, e_r\}$ is a minimum attachment such that $\lambda(H + Z_H) = \lambda + 1$.* □

Another important property of edge-interchange is given as follows.

Proposition 3.5 *For $q \neq 3$, $\alpha(e_i, H_i)$ is a leaf of H_i if and only if q is odd and $i = r - 1$.* □

Remark 3.1 *Even if $e_1, \ldots, e_r$ are selected so that H_i may be simple for each i, $i = 1, \ldots, r$, Proposition 3.5 also holds.* □

Remark 3.2 *Let f, f' be the two new edges such that*

$$V(f) \cap V(f') = \emptyset, \text{and } V(f) \cup V(f') = \{u_{11}, u_{12}, u_{21}, u_{22}\}$$

as in Proposition 3.2. Suppose that we are going to check whether $\alpha(f, G^{(i)} + \{f, f'\}) \cap \alpha(f', G^{(i)} + \{f, f'\}) = \emptyset$ or not. A maximum flow *algorithm can be used. Note that we have only to apply the algorithm to $G+\{f,f'\}$ (not to $G^{(i)}+\{f,f'\}$) or to $G+\{g,g'\}$, where*

$$V(g)\cap V(g')=\emptyset, \text{ and } V(g)\cup V(g')=\{u_{11},u_{12},u_{21},u_{22}\}$$

with $u_{ij}\in L(u_{ij})$, $i,j=1,2$. Thus this can be done in $O(\phi(n',m'+2))$ time, where n' and m' are the number of vertices and of edges of G and we assume that a maximum flow algorithm for G can be done in $\phi(n',m')$ time. [10] introduced a sparse graph $G^{(i)}=(N,E_1\cup E_2\cdots E_i)$ defined in Remark 2.1. By utilizing the results in Remark 2.1 and in [2], above checking operation can be done in $O(\lambda^2|N|)$ time. □

4 UW-$(\lambda(\Gamma;G)+1)$ECA-SV

Let $PE_\Gamma(G)$ be abbreviated as PE_Γ in the rest of the paper. Clearly we have Proposition 4.1.

Proposition 4.1 *Let SOL be a solution for G. We have*

$$\lceil\frac{|PE_\Gamma|}{2}\rceil\le|SOL|.$$

□

We consider how to compute $\lambda(\Gamma;G)$. If $\Gamma=V$ then [8] proposes an algorithm which computes $\lambda(V;G)$ in $O(\lambda|V|^2)$ time. However if $\Gamma\subset V$ then we cannot use this algorithm. Hence, for the case with $\Gamma\subset V$, we propose an algorithm which computes $\lambda(\Gamma;G)$ in $O(\lambda^2|V||\Gamma|\log\lambda)$ time by merging an algorithm proposed in [8] and the one proposed in Section 6.3 (page 131) of [2] for finding edge-connectivity of G. The algorithm is shown in the following.

Procedure Compute_λ

1. find E_i $(1\le i\le|E|)$ by an algorithm in [10];
2. $i\leftarrow 1$, $L\leftarrow 0$, choose $v\in\Gamma$ and repeat the following Steps 3 and 4;
3. find $\lambda(v,w;G^{(i)})$ for each $w\in\Gamma-v$ and set $L\leftarrow\min\{\lambda(v,w;G^{(i)})|w\in\Gamma-\{v\}\}$, where $G^{(i)}=(V,E_1\cup\cdots\cup E_i)$;
4. if $L<i$ then terminate the algorithm, otherwise set $i\leftarrow i\times 2$ and goto Step 3. □

Next proposition was shown in Section 6.3 (page 131) of [2].

Proposition 4.2 *[2] Let $v\in V$. Then we have $\lambda(V;G)=\min\{\lambda(v,v';G)|v'\in V-\{v\}\}$.* □

Corollary 4.1 *Let $S\subset V$ and $v\in S$. Then we have $\lambda(S;G)=\min\{\lambda(v,v';G)|v'\in S-\{v\}\}$.* □

Proposition 4.3 *We can compute λ in $O(\lambda^2|V||\Gamma|\log\lambda+|E|)$ time by Algorithm compute_λ, where $\lambda=\lambda(\Gamma;G)$.*

(Proof) A value L of Step 3 satisfies $L=\lambda(\Gamma;G^{(i)})$ from Corollary 4.1 by setting $S\leftarrow\Gamma$, and $L=\lambda(\Gamma;G^{(i)})=\lambda(\Gamma;G)$ hold from definition of $G^{(i)}$ and λ when the algorithm is terminated.

Step 1 is done in $O(|V|+|E|)$ time by [10]. Step 3 is done in $O(i^2|\Gamma||V|)$ time by Remark 2.1. Steps 3 and 4 are iterated $\lceil\log\lambda\rceil$ times until $\lambda<i=2^{\lceil\log\lambda\rceil}$ holds. Hence $\lambda(\Gamma;G)$ is computed in $O(\lambda(\Gamma;G)^2|V||\Gamma|\log\lambda(\Gamma;G)+|E|)$ time □

Next we propose Algorithm aug_sv-1 finding a solution for UW-$(\lambda(\Gamma;G)+1)$ECA-SV.

Algorithm aug_sv-1

1. find λ by Algorithm compute_λ; all $(\lambda+1)$-components of G and PE_Γ; set $P\leftarrow PE_\Gamma$;
2. set $E'\leftarrow\emptyset$; choose $V_1,V_2\in P$, set $P\leftarrow P-\{V_1,V_2\}$; iterate the following Steps 3-5;
3. if $P=\emptyset$ then $E'\leftarrow E'\cup\{(u_1,u_2)\}$, where $u_i\in V_i$, $i=1,2$; terminate the algorithm;
4. if $|P|\ge 2$ then choose distinct $V_3,V_4\in P$, otherwise set $V_3=V_4\in P$;
$P\leftarrow P-\{V_1,V_2\}$; find two edges e_1,e_2 by edge-interchange operations such that $G+\{e_1,e_2\}$ has $(\lambda+1)$-component containing $V_1\cup\cdots\cup V_4$ and $V(\{e_1,e_2\})=\{v_1,\ldots,v_4\}$, where $v_i\in V_i\cap\Gamma$ $(1\le i\le 4)$, and $v_3=v_4$ may hold if $|P|=1$;
5. $E'\leftarrow E'\cup\{e_1\}$; $V_1\leftarrow V_i$ and $V_2\leftarrow V_j$ for $V_i,V_j\in\{V_1,\ldots,V_4\}$ to V_1,V_2, where V_i (V_j, respectively) contains v_i (v_j) which is one of endvertices of e_2; goto Step 3; □

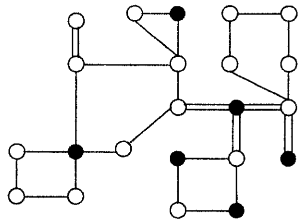

Figure 8: The cactus $G_1=(V_1,E_1)$ and $\Gamma_s\subseteq V_1$, where vertices of Γ_s are denoted as black spots.

We obtain next proposition from Section 3.

Proposition 4.4 *E' that is found by Algorithm aug_sv-1 satisfies that $G+E'$ has $(\lambda+1)$-component S with $\Gamma\subseteq S$ and $\lceil|PE_\Gamma|/2\rceil=|E'|$ holds.* □

Theorem 4.1 *Algorithm aug_sv-1 finds an edge set E' of minimum cardinarity such that $G+E'$ has a $(\lambda+1)$-component X with $\Gamma\subseteq X$ in $O(\lambda^2|V|(|V|+|\Gamma|\log\lambda)+|E|)$ time.*

(Proof) By Proposition 4.1 and 4.4, Algorithm aug_sv-1 correctly finds a minimum solution E' such that $G+E'$ has $(\lambda+1)$-component X with $\Gamma\subseteq X$.

We consider time complexity of Algorithm aug_sv-1. By Proposition 4.3, λ is computed in $O(\lambda^2|V||\Gamma|\log\lambda+|E|)$ time. All $(\lambda+1)$-components of G is found in $O(\lambda^2|V|^2+|E|)$ time by [11]. PE_Γ is found in $O(\lambda^2|V||\Gamma|+|E|)$ time by Proposition 2.3. Hence Step 1 is done in $O(\lambda^2|V|(|V|+|\Gamma|\log\lambda)+|E|)$ time. Executing of Step 4 once contains at most two edge-interchange operations, and by Remark 3.2, it is done in $O(\lambda^2|V|)$ time if $G^{(\lambda+1)}$ is obtained. The number of iterations of Steps 3–5 is bounded by $|\Gamma|$, and Steps 3–5 are done in $O(\lambda^2|\Gamma||V|)$ time. Thus the theorem follows. □

5 UW-$(\lambda(\Gamma;G)+1)$ECA-SV when $\lambda(\Gamma;G)=\lambda(V;G)$

We will propose an algorithm for this problem in the special case $\lambda(\Gamma;G)=\lambda(V;G)$. The algorithm utilizes a structural graph and a reduction which transforms UW-$(\lambda+1)$ECA-SV into UW-$(\lambda+1)$ECA. The algorithm runs in linear time. In the rest of this section let $\lambda=\lambda(\Gamma;G)=\lambda(G)$.

We consider checking whether $\lambda(\Gamma;G)=\lambda(G)$ or not. We obtain a structural graph of G in $O(|V|\log|V|+|E|)$ time by [4]. If Γ is partitioned into at least two $(\lambda+1)$-components corresponding to vertices of the structural graph then $\lambda(\Gamma;G)=\lambda(G)$, otherwise $\lambda(\Gamma;G)>\lambda(G)$. Hence the checking is done in $O(|V|\log|V|+|E|)$ time.

We obtain the algorithm by using the following results.

(1) [20, 21] ([14, 23], respectively) show that there is an $O(|V|+|E|)$ algorithm for solving UW-kECA-SV(*,MA) with $k=2$ $(k=3)$. These results show that UW-kECA-SV(*,MA) with $k=2$ $(k=3)$ can be equivalently transformed into UW-2ECA(*,MA) (UW-3ECA(*,MA)) in $O(|V|+|E|)$ time.

(2) A structural graph of G is a tree if λ is odd and is a cactus otherwise. A graph of G' obtained from G is a tree (a cactus, respectively) if $\lambda=2$ $(\lambda=3)$, where G' is a graph obtained

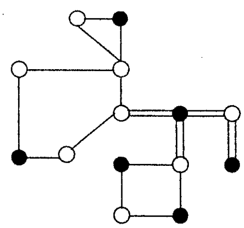

Figure 9: The cactus $G_2 = (V_2, E_2)$ and $\Gamma_s \subseteq V_2$.

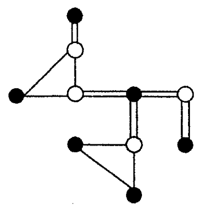

Figure 10: The condensation $G_3 = (V_3, E_3)$ of G_2 and $\Gamma_s \subseteq V_3$.

by shrinking each 2-component (3-component) S of G into an individual vertex v_S.

A tree is also a cactus, so we may only consider the case that a structural graph is a cactus.

We first explain reduction of UW-$(\lambda+1)$ECA-SV into UW-$(\lambda+1)$ECA such that a minimum solution to one of the two problems implies one to the other.

Let $G_1 = (V_1, E_1)$ denote a structural graph of G. Let Γ_s be a vertex set of V_1, each of which corresponds to a $(\lambda+1)$-component containing a vertex of Γ. G_1 is shown in Fig. 8, where vertices in Γ_s are denoted as black spots. G_1 is a cactus consisting of some cycles. A cycle of G_1 is called a *pendant cycle* if it contains at most one cutpoint. A pendant cycle is called a *core pendant* if it contains at least one vertex of Γ_s that is not a cutpoint. If G_1 has a cutpoint and there is any pendant cycle that is not a core one then delete all vertices except the cutpoint of this pendant, and repeat this procedure as much as possible. Let $G_2 = (V_2, E_2)$ denote the resulting cactus (see Fig. 9). Clearly any pendant of G_2 is a core one and $\Gamma_s \subseteq V_2$. The set $V_1 - V_2$ has a partition $V_1 - V_2 = W_1 \cup \cdots \cup W_k$ ($k \geq 1$; $W_i \cap W_j = \emptyset$ if $i \neq j$) such that, for each W_i, there is a cutpoint v_i for which $X_i = W_i \cup \{v_i\}$ induces a v_i-block of G_1. Each X_i and v_i are called an *outer component* of G_1 and the attachment of X_i, respectively. Clearly G_2 is obtained from G_1 by shrinking each X_i into v_i, $i = 1, \ldots, k$. If G_2 has a (v_0, v_n)-path of length $n \geq 2$ with inner vertices $v_i \notin \Gamma_s$ and $d_G(v_i) = 2$, $i = 1, \ldots, n-1$, then delete all inner vertices $v_1, \ldots, v_{n-1}$ and add an edge (v_0, v_n). Repeat this procedure as much as possible, and let $\chi(G_2)$ denote the resulting graph, which is called the *condensation* of G_2. Denote $\chi(G_2)$ as $G_3 = (V_3, E_3)$ (Fig. 10). Note that $\Gamma_s \subseteq V_3$ and any vertex v with $d_{G_3}(v) = 2$ belongs to Γ_s. Let $L_3 = \{v \in V_3 | d_{G_3}(v) = 2\}$ ($\subseteq \Gamma_s$).

Now we describe an algorithm aug_sv'-1.

Algorithm aug_sv'-1

1. construct a structural graph $G_1 = (V_1, E_1)$ from G; let $\Gamma_s \subseteq V_1$ be the set of vertices, each of which represents a $(\lambda+1)$-component containing at least one vertex of Γ;
2. find all outer components $X_1, \ldots, X_k$ ($k \geq 0$) of G_1;
3. if $k \geq 1$ then construct $G_2 = (V_2, E_2)$ by shrinking each outer component X_i into its attachment v_i, $i = 1, \ldots, k$;
4. construct the condensation $\chi(G_2)$ and denote $\chi(G_2)$ as $G_3 = (V_3, E_3)$;
5. solve UW-$(\lambda+1)$ECA for G_3 (that is, find a set E_3' of minimum cardinality such that $\lambda(G_3 + E_3') = 2$ ($= 3$, respectively) if λ is odd (even) by means of an $O(|V_3| + |E_3|)$ algorithm, denoted as ATEC, proposed in [24, 25, 26] or [12];
6. define E' by

$$E' = \{(u,v) | (u',v') \in E_3' \text{ and } u \ (v, \text{ respectively}) \text{ is a vertex contained in the } (\lambda+1)\text{-component represented by } u' \ (v')\};$$

□

The relation among G, G_2 and G_3 are shown by the following two lemmas. The point is that each vertex v with $d_{G_2}(v) = 2$ represents a λ-palm.

Lemma 5.1 *There is an edge set E' of minimum cardinarity such that $\lambda(u, v; G+E') \geq \lambda+1$ for $\forall u, v \in \Gamma$ if and only if there is an edge set E_2' of minimum cardinarity such that $\lambda(u', v'; G_2 + E_2') \geq 2$ (≥ 3, respectively) if λ is odd (even) for $\forall u', v' \in \Gamma_s$, where $|E'| = |E_2'|$.* □

Lemma 5.2 *There is an edge set E_2' of minimum cardinality such that $\lambda(u', v'; G_2 + E_2') \geq 3$ for $\forall u', v' \in \Gamma_s$ if and only if there is an edge set E_3' of minimum cardinality such that $\lambda(G_3 + E_3') \geq 3$, where $|E_2'| = |E_3'|$.* □

We obtain the next theorem, since a structural graph of G is obtained in $O(|V|\log|V| + |E|)$ by [4].

Theorem 5.1 *This problem can be solved optimally by the algorithm in $O(|V|\log|V| + |E|)$ time if $\lambda(G) = \lambda(\Gamma; G)$.* □

6 Concluding Remarks

We propose an $O(\lambda^2|V|(|V| + |\Gamma|\log\lambda) + |E|)$ ($O(|V|\log|V| + |E|)$, respectively) algorithm for UW-$(\lambda+1)$ECA-SV(*,MA) if $\lambda > \lambda(G)$ (if $\lambda = \lambda(G)$) and an $O(\lambda^2|V||\Gamma|\log\lambda + |E|)$ algorithm computing λ when $\Gamma \subset V$, where $\lambda = \lambda(\Gamma; G)$.

Feature researches are proposing the following (1)–(3):

(1) an algorithm for UW-kECA-SV(*,MA) with $k = \lambda(\Gamma; G) + \delta$ and $\delta \geq 2$;

(2) an algorithm for UW-kECA-SV(*,SA);

(3) an approximation algorithm for W-kECA-SV, since W-kECA-SV is known to be NP-complete [7].

References

[1] K. P. Eswaran and R. E. Tarjan. Augmentation problems. *SIAM J. Comput*, 5, pp. 653-655, 1976.

[2] S. Even. *Graph Algorithms.* Pitman, London, 1979.

[3] A. Frank. Augmenting graphs to meet edge connectivity requirements. *SIAM J. Discrete Mathematics*, 5(1), pp. 25–53, 1992.

[4] H. N. Gabow. Applications of a poset representation to edge connectivity and graph rigidity. In *Proc. 32nd IEEE Symposium on Foundations of Computer Science*, pp. 812–821, 1991.

[5] Z. Galil and G. F. Italiano. Reducing edge connectivity to vertex connectivity. *SIGACT NEWS*, 22, pp. 57-61, 1991.

[6] A. V. Karzanov and E. A. Timofeev. Efficient algorithm for finding all minimal edge cuts of a nonoriented graph. *Cybernetics*, 156–162, March-April 1986. Translated from Kibernetika, 2 (1986), pp.8–12.

[7] T. Mashima, S. Taoka, and T. Watanabe. *Approximation Algorithms for the k-Edge-Connectivity Augmentation Problem.* Tec. Rep. COMP92-24, IEICE of Japan. pp.11–20, 1992.

[8] H. Nagamochi and T. Ibaraki. Computing edge-connectivity in multigraphs and capacitated graphs. *SIAM J. Discrete Mathematics*, 5, pp. 54–66, 1992.

[9] H. Nagamochi and T. Ibaraki. A linear time algorithm for computing 3-edge-connected components in a multigraph. *Japan J. Industrial and Applied Math.*, 9(7), pp. 163–180, 1992.

[10] H. Nagamochi and T. Ibaraki. A linear time algorithm for finding a sparse k-connected spanning subgraph of a k-connected graph. *Algorithmica*, 7, pp. 583–596, 1992.

[11] H. Nagamochi and T. Watanabe. Computing k-edge-connected components of a multigraphs. *Trans. IEICE of Japan*, E76-A(4), pp. 513–517, 1993.

[12] D. Naor, D. Gusfield, and C. Martel. A fast algorithm for optimally increasing the edge connectivity. In *Proc. 31st Annual IEEE Symposium on Foundations of Computer Science*, pp. 698–707, 1990.

[13] J. C. Picard and M. Queyranne. On the structure of all minimum cuts in a network and applications. *Mathematical Programming Study*, 13, pp. 8–16, 1980.

[14] S. Taoka, T. Mashima, and T. Watanabe. *The 3-Edge-Connectivity Augmentation Problem for a Specified Set of Vertices.* Tec. Rep. COMP92-48, IEICE of Japan, 1992. pp. 1–10.

[15] S. Taoka, D. Takafuji, and T. Watanabe. Simplicity-preserving augmentation of the edge-connectivity of a graph. *Tech. Rep. of IEICE of Japan*, COMP93-73, pp. 49–56, 1994.

[16] S. Taoka, T. Watanabe, and K. Onaga. A linear time algorithm for computing all 3-edge-connected components of an multigraph. *Trans. IEICE*, E75-3, pp. 410–424, 1991.

[17] S. Ueno, Y. Kajitani, and H. Wada. Minimum augmentation of trees to k-edge-connected graph. *Networks*, 18, pp. 19–25, 1988.

[18] T. Watanabe. *An efficient way for edge-connectivity augmentation.* Tec. Rep. ACT-76-UILU-ENG-87-2221, Coordinated Science Lab., University of Illinois at Urbana, Urbana, IL 61801, April 1987. Also presented at Eighteenth Southeastern International Conference on Combinatorics, Graph Theory, Computing, No.15, Boca Raton, FL, U.S.A., February 1987.

[19] T. Watanabe. *A simple improvement on edge-connectivity augmentation.* Tech. Rep. CAS87-203, pp. 43–48, IEICE of Japan, 1987.

[20] T. Watanabe, Y. Higashi, and A. Nakamura. An approach to robust networks construction from graph augmentation problems. In *Proc. of 1990 IEEE Int. Symp. on Circuits and Systems*, pp. 2861–2864, May 1990.

[21] T. Watanabe, Y. Higashi, and A. Nakamura. Construction robust networks by means of graph augmentation problems. *Trans. IEICE of Japan*, 73-A(7), pp. 1242–1254, 1991. Also see Electronics and Communications in Japan, Part 3, Vol. 74, No. 2 (1991), pp. 79–96.

[22] T. Watanabe and A. Nakamura. Edge-connectivity augmentation problems. *J. Comput. System Sci.*, 35(1), pp. 96–144, 1987.

[23] T. Watanabe, S. Taoka, and T. Mashima. Minimum-cost augmentation to 3-edge-connect all specified vertices in a graph. In *Proc. 1993 IEEE International Symposium on Circuits and Systems*, pp. 2311–2314, May 1993.

[24] T. Watanabe, M. Yamakado, and K. Onaga. A linear-time augmenting algorithm for 3-edge-connectivity augmentation problems. In *Proc. 1991 IEEE International Symposium on Circuits and Systems*, pp. 1168–1171, 1991.

[25] T. Watanbe and M. Yamakado. A linear time algorithm for smallest augmentation to 3-edge-connect a graph. *Trans. IEICE of Japan*, E76-A(4), pp. 518–531, 1993.

[26] M. Yamakado and T. Watanabe. A linear-time augmenting algorithm for the 3-edge-connectivity augmentation ploblem. *Tech. Rep. Rep. of IEICE of Japan*, 1990.