Journal of the Operations Research
Society of Japan
Vol. 39, No. 2, June 1996

CONSTRUCTING CACTUS REPRESENTATION FOR ALL MINIMUM CUTS IN AN UNDIRECTED NETWORK

Hiroshi Nagamochi *Kyoto University*
Tiko Kameda *Simon Fraser University*

(Received May 16, 1994)

Abstract The construction of a *cactus representation* for all minimum cuts in an edge-weighted, undirected network is studied in this paper. Given such a network with a set V of vertices and a set E of edges, the previously known algorithms for constructing a cactus representation require solving $|V|-1$ maximum flow problems, each with a prescribed pair of source and sink, in order to find all minimum cuts that separate them. We first show that computing a maximum flow between a pair of vertices, s and t, reveals many other minimum cuts that do not separate s and t. Based on this fact, we propose a new algorithm that can choose an arbitrary pair of source and sink for each maximum flow problem. For a network with unit edge weights, our algorithm runs in $O(|E|+\lambda|V|^2)$ time, where λ is the weight of a minimum cut.

1. Introduction

Let $N=(V,E,c)$ denote a network with a set V of vertices and a set E of edges weighted by a function c. The minimum cut in a network is one of the most fundamental notions in graph theory and is a rich source of interesting combinatorial problems (e.g., [7, 8]). Many algorithms for finding at least one minimum cut in a network have been designed so far [4, 7, 10, 11, 12, 14]. Currently, the asymptotically fastest algorithms can find a minimum cut in a network $N=(V,E,c)$ in $O(|E||V|+|V|^2\log|V|)$ [14] or $O(|E||V|\log(|V|^2/|E|))$ time [10]. It is well-known that one can find all minimum cuts separating two specified vertices, s and t, by computing a maximum flow between s and t [19].

We investigate in this paper the problem of constructing a "compact" representation for all minimum cuts in a network. A network is commonly called a **cactus** if any pair of cycles in it, if any, has at most one vertex in common. It is known [3, 8] that all minimum cuts in a network $N=(V,E,c)$ can be embedded in a cactus of size $O(|V|)$, from which any minimum cut can be easily obtained as a minimum cut of the cactus. This representation has been relatively unknown in the West since it was reported in [3]. However, its great usefulness has gradually been recognized. For example, it can considerably speed up the algorithms for the edge-connectivity augmentation problem [8, 17].

For the special class of networks whose edges have unit weights, some efficient algorithms are known for constructing a cactus representation. For example, Gabow's algorithm [8] runs in $O(|E|+\lambda^2|V|\log|V|)$ time, where λ is the size of a minimum cut. If such a network has no multiple edges, a cactus representation can be constructed in $O(\lambda|V|^2)$ [11] or $O(|E|+\lambda^2|V|\log(|V|/\lambda))$ time [8].

Most of the known cactus construction algorithms [11, 18] utilize a maximum flow algorithm. These algorithms are based on the fact that the set of all minimum cuts can be divided into $|V|-1$ disjoint sets, each of which is the set of minimum cuts separating two prescribed vertices and, hence, can be obtained by solving a single maximum flow problem in which both the source and sink are specified [19].

Journal of the Operations Research
Society of Japan
Vol. 39, No. 2, June 1996

CONSTRUCTING CACTUS REPRESENTATION FOR ALL MINIMUM CUTS IN AN UNDIRECTED NETWORK

Hiroshi Nagamochi
Kyoto University

Tiko Kameda
Simon Fraser University

(Received May 16, 1994)

Abstract The construction of a *cactus representation* for all minimum cuts in an edge-weighted, undirected network is studied in this paper. Given such a network with a set V of vertices and a set E of edges, the previously known algorithms for constructing a cactus representation require solving $|V|-1$ maximum flow problems, each with a prescribed pair of source and sink, in order to find all minimum cuts that separate them. We first show that computing a maximum flow between a pair of vertices, s and t, reveals many other minimum cuts that do not separate s and t. Based on this fact, we propose a new algorithm that can choose an arbitrary pair of source and sink for each maximum flow problem. For a network with unit edge weights, our algorithm runs in $O(|E|+\lambda|V|^2)$ time, where λ is the weight of a minimum cut.

1. Introduction

Let $N=(V,E,c)$ denote a network with a set V of vertices and a set E of edges weighted by a function c. The minimum cut in a network is one of the most fundamental notions in graph theory and is a rich source of interesting combinatorial problems (e.g., [7, 8]). Many algorithms for finding at least one minimum cut in a network have been designed so far [4, 7, 10, 11, 12, 14]. Currently, the asymptotically fastest algorithms can find a minimum cut in a network $N=(V,E,c)$ in $O(|E||V|+|V|^2\log|V|)$ [14] or $O(|E||V|\log(|V|^2/|E|))$ time [10]. It is well-known that one can find all minimum cuts separating two specified vertices, s and t, by computing a maximum flow between s and t [19].

We investigate in this paper the problem of constructing a "compact" representation for all minimum cuts in a network. A network is commonly called a **cactus** if any pair of cycles in it, if any, has at most one vertex in common. It is known [3, 8] that all minimum cuts in a network $N=(V,E,c)$ can be embedded in a cactus of size $O(|V|)$, from which any minimum cut can be easily obtained as a minimum cut of the cactus. This representation has been relatively unknown in the West since it was reported in [3]. However, its great usefulness has gradually been recognized. For example, it can considerably speed up the algorithms for the edge-connectivity augmentation problem [8, 17].

For the special class of networks whose edges have unit weights, some efficient algorithms are known for constructing a cactus representation. For example, Gabow's algorithm [8] runs in $O(|E|+\lambda^2|V|\log|V|)$ time, where λ is the size of a minimum cut. If such a network has no multiple edges, a cactus representation can be constructed in $O(\lambda|V|^2)$ [11] or $O(|E|+\lambda^2|V|\log(|V|/\lambda))$ time [8].

Most of the known cactus construction algorithms [11, 18] utilize a maximum flow algorithm. These algorithms are based on the fact that the set of all minimum cuts can be divided into $|V|$-1 disjoint sets, each of which is the set of minimum cuts separating two prescribed vertices and, hence, can be obtained by solving a single maximum flow problem in which both the source and sink are specified [19].

A partition $\{X, \overline{X}\}$ of V, is called a **cut** of N. We say that a cut $\{X, \overline{X}\}$ **separates** two subsets $\emptyset \subset V', V'' \subset V$ if $V' \subseteq X$ and $V'' \subseteq \overline{X}$. A cut $\{X, \overline{X}\}$ separating two vertices $u, v \in V$ is called a ***uv*-cut**. For two vertices $u, v \in V$, define

$$\lambda(u, v; N) \equiv \min\{c(X, \overline{X}) \mid \{X, \overline{X}\} \text{ is a } uv\text{-cut of } N\}.$$

For example, for the network N_1 in Fig. 1, we have $\lambda(N_1) = \lambda(v_1, v_{19}; N_1) = 4$ and $\lambda(v_9, v_{12}; N_1) = 5$. We define

$$\lambda(N) \equiv \min\{c(X, \overline{X}; N) \mid \{X, \overline{X}\} \text{ is a cut of } N\},$$

which is equivalent to $\min\{\lambda(u, v; N) \mid \text{ distinct } u, v \in V\}$. Since we consider only connected N, we have $\lambda(N) > 0$. We call a cut $\{X, \overline{X}\}$ satisfying $c(X, \overline{X}; N) = \lambda(N)$ a **minimum cut** (or **min-cut**, for short) of N, and denote the set of all min-cuts of N by $\mathcal{C}(N)$.

For a set of vertices, $U \subset V$, the network **induced by** U in N is the network N' whose vertex set is U and whose edge set consists of all the edges $e = (u, v) \in E$ such that $u, v \in U$. In other words,

$$N' = (U, [U \times U] \cap E, c'),$$

where c' is the same as c, except that it is restricted to the edge set of N'. For $E' \subseteq E$, $N{-}E'$ denotes the network obtained from N by deleting the edges in E', keeping the vertex set V unchanged. For $V' \subset V$, $N{-}V'$ denotes the network obtained from N by deleting the edges incident to vertices in V' as well as the vertices in V'.

Throughout the paper, we may write a singleton set $\{x\}$ simply as x.

2.2. Cactus Representation for Min-Cuts

We call a network consisting of a single vertex a **trivial** cactus. A network with more than one vertex is called a **cactus** if any edge in it belongs to a cycle[†] and any pair of cycles, if any, has at most one vertex in common. A cactus $N = (V, E, c)$ is said to be **uniform**,[‡] if $c(e) = 1$ for $\forall e \in E$. Since the weight of any edge of a uniform cactus is always 1, we simply write $N = (V, E)$ to refer to a uniform cactus $N = (V, E, c)$. Obviously, for any non-trivial, uniform cactus N, $\lambda(N) = 2$ holds and $E(X, \overline{X}; N)$ consists of two edges on the same cycle for any min-cut $\{X, \overline{X}\}$.

This subsection shows how to embed a set of min-cuts of a network N in a "small" uniform cactus. For a given network $N = (V, E, c)$, we introduce a uniform cactus $\mathcal{R} = (W, F)$ and a mapping $\varphi : V \to W$. Throughout the remainder of the paper, we shall use the term "vertex" to denote an element in V, and the term "node" to denote an element in W. In what follows, we normally use u and v with or without subscript/superscript to denote vertices, and x, y, and z with or without subscript/superscript to denote nodes. W may contain a node x such that V contains no vertex v with $\varphi(v) = x$. Such a node x is called an **empty node**. As before, $\mathcal{C}(N)$ and $\mathcal{C}(\mathcal{R})$ denote the set of all min-cuts of N and $\mathcal{R}$, respectively.

Definition 2.1 *For a given subset $\mathcal{C}' \subseteq \mathcal{C}(N)$ of min-cuts, $(\mathcal{R}, \varphi)$ is a* **representation** *for $\mathcal{C}'$ if it satisfies* (i) *and* (ii)*:*

(i) *For an arbitrary min-cut $\{S, W{-}S\} \in \mathcal{C}(\mathcal{R})$, $\{X, \overline{X}\} \in \mathcal{C}'$ holds, where $X = \{u \mid \varphi(u) \in S\}$ and $\overline{X} = \{v \mid \varphi(v) \in W{-}S\}$.*

[†] This condition is normally not imposed on a cactus, so that a cactus may have a "bridge." However, a bridge can be replaced by two edges in order to satisfy this condition. Our definition makes the subsequent discussion of cacti slightly easier.

[‡] This definition of "uniform" is more restrictive than that in [16].

(ii) *Conversely, for any min-cut $\{X, \overline{X}\} \in \mathcal{C}'$, there exists a min-cut $\{S, W-S\} \in \mathcal{C}(\mathcal{R})$ such that $X = \{u \mid \varphi(u) \in S\}$ and $\overline{X} = \{v \mid \varphi(v) \in W-S\}$.*

A representation $(\mathcal{R}, \varphi)$ for $\mathcal{C}'$ is called a **cactus representation** *for $\mathcal{C}'$ if $\mathcal{R}$ is a uniform cactus.* □

For convenience, we use the following notation:

$$\begin{aligned} \varphi(X) &\equiv \{\varphi(v) \mid v \in X\} \quad \text{for } X \subseteq V, \\ \varphi^{-1}(S) &\equiv \{v \in V \mid \varphi(v) \in S\} \quad \text{for } S \subseteq W, \text{ and} \\ \varphi^{-1}(\mathcal{C}') &\equiv \{\{\varphi^{-1}(S), \varphi^{-1}(W-S)\} \mid \{S, W-S\} \in \mathcal{C}'\} \quad \text{for } \mathcal{C}' \subseteq \mathcal{C}(\mathcal{R}). \end{aligned}$$

When $(\mathcal{R}, \varphi)$ is a representation for $\mathcal{C}' \subseteq \mathcal{C}(N)$, we say that cut $\{X, \overline{X}\} \in \mathcal{C}'$ and cut $\{S, W-S\} \in \mathcal{C}(\mathcal{R})$ **correspond** to each other if $\varphi(X) \subseteq S$ and $\varphi(\overline{X}) \subseteq W-S$. The conditions (i) and (ii) in Definition 2.1 can be expressed simply as

$$\varphi^{-1}(\mathcal{C}(\mathcal{R})) = \mathcal{C}'.$$

Note that any min-cut $\{S, W-S\} \in \mathcal{C}(\mathcal{R})$ always corresponds to exactly one min-cut in $\mathcal{C}'$. However, a min-cut in $\mathcal{C}'$ may correspond to more than one min-cut in $\mathcal{C}(\mathcal{R})$ if $\mathcal{R}$ has an empty node. For the special case $\mathcal{C}' = \emptyset$, $(\mathcal{R}, \varphi)$ with a trivial cactus $\mathcal{R} = (x, \emptyset)$ and $\varphi(V) = x$ is a cactus representation for it. It is shown in [3] that, for any network N, there exists a cactus representation $(\mathcal{R}, \varphi)$ for $\mathcal{C}(N)$. For example, Fig. 2 shows a cactus representation for $\mathcal{C}(N_1)$ for network N_1 in Fig. 1, where z_2, z_8, z_{15} are empty nodes.

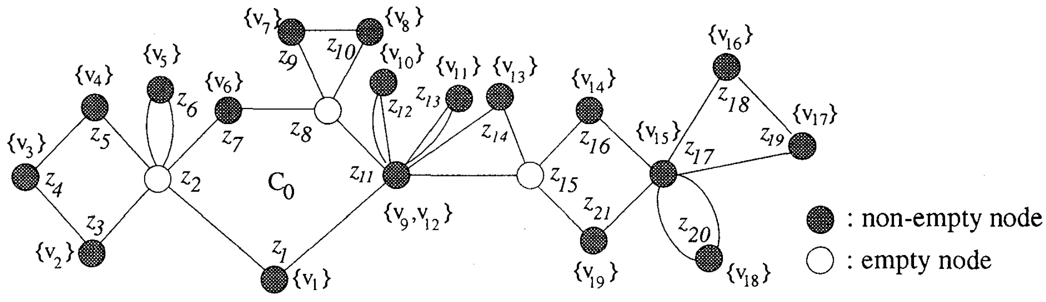


Figure 2: A cycle-type normal cactus representation for $\mathcal{C}(N_1)$ ($\{\cdot\}$ beside node z_i shows $\varphi^{-1}(z_i)$).

Unfortunately, a cactus representation for a given $\mathcal{C}'$, if one exists, may not be uniquely determined without imposing further structural restrictions, as shown below. In a cactus representation, we call a cycle on n nodes an **n-cycle**, and a node v belonging to exactly j cycles a **j-junction node**. Note that, by definition, neither S nor $W-S$ contains only empty nodes, if $\{S, W-S\} \in \mathcal{C}(\mathcal{R})$. In particular, there is no empty 1-junction node in a cactus representation. We call a cactus representation **normal**, if it has no empty 2-junction node belonging to a 2-cycle, and we call a normal cactus representation without empty 3-junction nodes a **cycle-type** normal cactus representation (see [16]). For example, Figs. 3(b), (c) and (d) show three cactus representations for the same network in Fig. 3(a). The cactus representation in Fig. 3(b) is not normal since it has an empty 2-junction node, z_1, belonging to the 2-cycle $\{z_1, z_2\}$. The cactus representation in Fig. 3(c) is not of cycle-type since it has an empty 3-junction node, z_3, while that in Fig. 3(d) is of cycle-type. The cactus representation in Fig. 2 is also a cycle-type normal cactus representation.

In the rest of this paper, we shall restrict ourselves to the cycle-type normal cactus representations, which is justified by the following lemma.

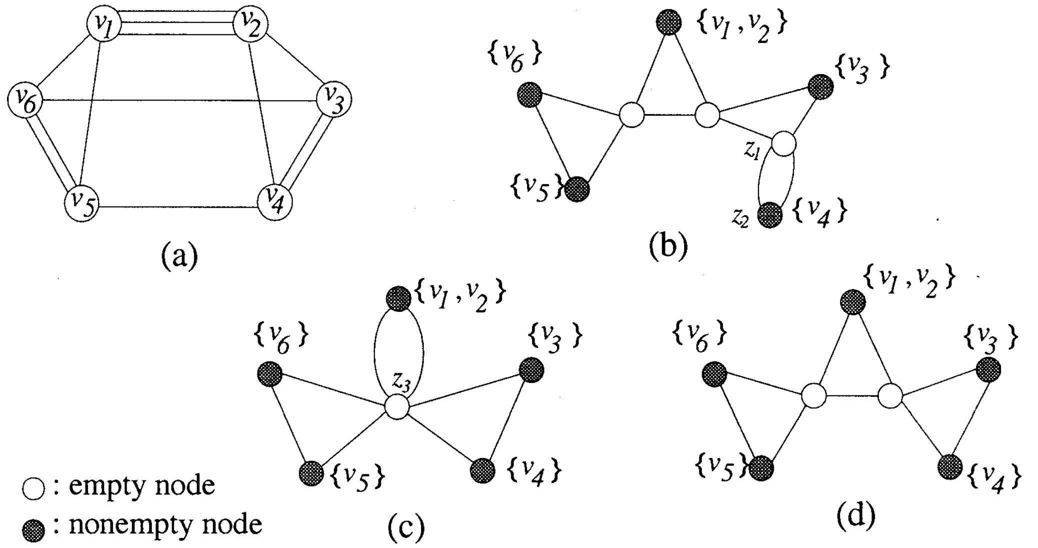


Figure 3: Three examples of cactus representations.

Lemma 2.1 [16] *Given a network* $N = (V, E, c)$, (i),(ii) *and* (iii) *below hold for any subset* $\mathcal{C}' \subseteq \mathcal{C}(N)$.

(i) *If* $\mathcal{C}'$ *has a cactus representation, then* $\mathcal{C}'$ *has a cycle-type normal cactus representation.* (Lemma 3.3(ii) in [16].)
(ii) *If* $\mathcal{C}'$ *has a cycle-type normal cactus representation, then such a representation is unique.* (Theorem 4.1 in [16].)
(iii) *Any cycle-type normal cactus representation* $(\mathcal{R}, \varphi)$ *for* $\mathcal{C}'$, *if one exists, has at most* $|V|$ *empty nodes.* (Lemma 4.4 in [16].) □

3. Representing Min-Cuts by a Partition

As a preparation for cactus representation of all min-cuts in $N = (V, E, c)$, we introduce and discuss in this section the concept of an "ordered partition" of V, by which we represent some min-cuts of N.

3.1. Minimum Cut Partition

Let $N = (V, E, c)$ be a network. A cut $\{X, \overline{X}\}$ of N **crosses** another cut $\{Y, \overline{Y}\}$ of N if

$$X \cap Y \neq \emptyset,\ \ \overline{X} \cap Y = Y - X \neq \emptyset,\ \ X \cap \overline{Y} = X - Y \neq \emptyset, \text{ and } \overline{X} \cap \overline{Y} = \overline{X \cup Y} \neq \emptyset.$$

The following elegant lemma lays the groundwork for subsequent discussions.

Lemma 3.1 [2, 3, 18] *Let* $\{X, \overline{X}\}$ *and* $\{Y, \overline{Y}\}$ *be any two min-cuts of* $N = (V, E, c)$. *If these cuts cross each other, then*

(i) $c(V_1, V_2) = c(V_2, V_3) = c(V_3, V_4) = c(V_4, V_1) = \lambda(N)/2$, *and*
(ii) $c(V_1, V_3) = c(V_2, V_4) = 0$

hold, where $V_1 = X \cap Y$, $V_2 = \overline{X} \cap Y$, $V_3 = \overline{X} \cap \overline{Y}$, *and* $V_4 = X \cap \overline{Y}$. □

We call an ordered set $(V_1, V_2 \ldots, V_r)$, $r \geq 2$, an **ordered partition** (or **o-partition**, for short) of V if $\{V_1, \ldots, V_r\}$ is a partition of V. Given an o-partition $(V_1, \ldots, V_r)$ and two indices h and k $(1 \leq h \leq k \leq r)$, we define

$$V_{(h,k)} \equiv V_h \cup V_{h+1} \cup \cdots \cup V_k.$$

For $\mathcal{C}' \subseteq \mathcal{C}(N)$, an o-partition $\pi_1 = (V_1, V_2, \ldots, V_r)$ of V is called a **min-cut o-partition** (or **MC-partition**, for short) over $\mathcal{C}'$, if

$$\mathcal{C}_1 = \{\{V_{(1,k)}, \overline{V_{(1,k)}}\} \mid 1 \leq k \leq r-1\} \subseteq \mathcal{C}'.$$

An o-partition $\pi_2 = (V_1', V_2', \ldots, V_{r'}')$ of V is called a **circular** MC-partition over $\mathcal{C}'$, if

$$\mathcal{C}_2 = \{\{V'_{(h,k)}, \overline{V'_{(h,k)}}\} \mid 1 \leq h \leq k \leq r'-1\} \subseteq \mathcal{C}'.$$

We say that a cut in the above $\mathcal{C}_1$ (resp. $\mathcal{C}_2$) **belongs** to the MC-partition π_1 (resp. circular MC-partition π_2), and that π_1 (resp. π_2) **represents** $\mathcal{C}_1$ (resp. $\mathcal{C}_2$). Clearly, any circular MC-partition over $\mathcal{C}'$ is an MC-partition over $\mathcal{C}'$, but not vice versa. With these notions, Lemma 3.1 can be restated as follows.

Lemma 3.2 *Let $\{X, \overline{X}\}$ and $\{Y, \overline{Y}\}$ be any two min-cuts of a network $N = (V, E, c)$. If these cuts cross each other, then (V_1, V_2, V_3, V_4) is a circular MC-partition over $\mathcal{C}(N)$, where $V_1 = X \cap Y$, $V_2 = \overline{X} \cap Y$, $V_3 = \overline{X} \cap \overline{Y}$, and $V_4 = X \cap \overline{Y}$.* □

We now introduce two basic types of cactus representations, which represent the set of all min-cuts belonging to an MC-partition and a circular MC-partition, respectively. We call a uniform cactus a **chain** if it consists only of 2-cycles and each node is either a 1- or a 2-junction node. A representation $(\mathcal{R}, \varphi)$ in which $\mathcal{R}$ is a chain is called a **chain representation**. Let π_1, $\mathcal{C}_1$, π_2, and $\mathcal{C}_2$ be as defined above. Clearly, $\mathcal{C}_1$ has a chain representation, which we denote by $(\ddot{\mathcal{R}}_{\pi_1}, \psi_{\pi_1})$, where

$$\ddot{\mathcal{R}}_{\pi_1} = (\{x_i \mid 1 \leq i \leq r\}, \{e_1, e_1', e_2, e_2', \ldots, e_{r-1}, e_{r-1}'\}), \text{ and } \psi_{\pi_1}^{-1}(x_i) = V_i \quad (1 \leq i \leq r).$$

Here $\{e_i, e_i'\} = E(x_i, x_{i+1})$, $1 \leq i \leq r-1$. Similarly, $\mathcal{C}_2$ has a **cycle representation**, which we denote by $(\mathring{\mathcal{R}}_{\pi_2}, \phi_{\pi_2})$, where

$$\mathring{\mathcal{R}}_{\pi_2} = (\{x_i \mid 1 \leq i \leq r'\}, \{e_1, e_2, \ldots, e_{r'}\}). \text{ and } \phi_{\pi_2}^{-1}(x_i) = V_i' \quad (1 \leq i \leq r').$$

Here $e_i = (x_i, x_{i+1})$, $1 \leq i \leq r'-1$, and $e_{r'} = (x_{r'}, x_1)$.

3.2. Maximal Circular MC-Partitions and Subcactus

Let $\mathcal{C}' \subseteq \mathcal{C}(N)$ for a network $N = (V, E, c)$. A circular MC-partition $(V_1, V_2, \ldots, V_r)$ over $\mathcal{C}'$ is said to be **maximal** if for any $X \subset V_i$ with $1 \leq i \leq r$, $(V_1, \ldots, V_{i-1}, X, V_i - X, V_{i+1}, \ldots, V_r)$ is no longer a circular MC-partition over $\mathcal{C}'$. Suppose that $\mathcal{C}'$ has the cycle-type normal cactus representation $(\mathcal{R}, \varphi)$. For any r-cycle C $(r \geq 2)$ in $\mathcal{R}$, let $W(C)$ (resp. $F(C)$) denote the set of nodes (resp. edges) on C, and let (W_i, F_i), $i = 1, 2, \ldots, r$, denote the connected components of $\mathcal{R} - F(C)$, where each (W_i, F_i) appears on C in the order $i = 1, 2, \ldots, r$. Clearly, each (W_i, F_i) is a uniform cactus, which we call a **subcactus** in $\mathcal{R} - F(C)$. Since any two cycles in $\mathcal{R}$ has at most one node in common, each subcactus (W_i, F_i), $i = 1, \ldots, r$, has exactly one node (say, y_i) from $W(C)$; these nodes appear in the order, $y_1, y_2, \ldots, y_r$ on C. Let $\pi_C = (V_1, V_2, \ldots, V_r)$ be the o-partition of V such that $V_i = \varphi^{-1}(W_i)$, $i = 1, 2, \ldots, r$. Clearly, π_C is a circular MC-partition over $\mathcal{C}'$. We say that C **yields** π_C from $\mathcal{R}$. Interestingly, it is known [16] that there is a one-to-one correspondence between the maximal circular MC-partitions and the cycles in a cycle-type normal cactus representation. More formally, we have the following lemma.

Lemma 3.3 [16] *Given $N = (V, E, c)$ and $\mathcal{C}' \subseteq \mathcal{C}(N)$, let $(\mathcal{R}, \varphi)$ be the cycle-type normal cactus representation for $\mathcal{C}'$. Then, an o-partition $\pi = (V_1, V_2, \ldots, V_r)$ of V is a maximal circular MC-partition over $\mathcal{C}'$ if and only if $\mathcal{R}$ has an r-cycle which yields π from $\mathcal{R}$.* □

For example, the 5-cycle $C_0 = \{z_1, z_2, z_7, z_8, z_{11}\}$ in Fig. 2 yields a circular MC-partition,

$$\pi_{C_0} = (\{v_1\}, \{v_2, \ldots, v_5\}, \{v_6\}, \{v_7, v_8\}, \{v_9, \ldots, v_{19}\}),$$

over $\mathcal{C}(N_1)$ in the network N_1 of Fig. 1. Since the representation in Fig. 2 is the cycle-type normal cactus representation for $\mathcal{C}(N_1)$, π_{C_0} is a maximal circular MC-partition over $\mathcal{C}(N_1)$ by Lemma 3.3. Conversely, Lemma 3.3 implies that a maximal circular MC-partition over $\mathcal{C}(N)$ uniquely determines a cycle in the cycle-type normal cactus representation for $\mathcal{C}(N)$.

3.3. (s, t)-MC-Partitions

We say that an edge $e = (s, t)$ is **critical** if $\lambda(s, t; N) = \lambda(N)$. Thus, if e is critical, then we have $\lambda(N{-}e) < \lambda(N)$. This subsection first shows, for a given a critical edge $e = (s, t)$, how to derive an MC-partition of V which represents all min-cuts separating s and t. Based on Lemma 3.1, Dinits, et al.[3], and Karzanov and Timofeev[11] proved the following lemma, which plays an important role in enumerating all min-cuts effectively.

Lemma 3.4 [11, 18] *For a network $N = (V, E, c)$, let $(s, t) \in E$ be critical. Then, all min-cuts in $\mathcal{C}(N)$ that separate s and t belong to one MC-partition over $\mathcal{C}(N)$.*

Proof: We first show that any two min-cuts $\{X, \overline{X}\}$ and $\{Y, \overline{Y}\}$ in $\mathcal{C}(N)$ that separate s and t never cross each other. Assume $s \in X$ and $s \in Y$, i.e., $s \in X \cap Y = V_1$, without loss of generality. If these cuts cross each other, edge $e = (s, t)$ is contained in $E(V_1, V_3)$ and hence $c(V_1, V_3) \geq c(e) > 0$, where $V_3 = \overline{X} \cap \overline{Y}$. However, by Lemma 3.1(ii), $c(V_1, V_3) = 0$ must hold, a contradiction. Therefore, $\{X, \overline{X}\}$ and $\{Y, \overline{Y}\}$ do not cross each other. This implies that all min-cuts $\{X_i, \overline{X}_i\}$ ($s \in X_i$, $i = 1, \ldots, q$) in $\mathcal{C}(N)$ that separate s and t can be arranged so that

$$\{s\} \subseteq X_1 \subset X_2 \subset \cdots \subset X_q \subseteq V{-}\{t\}$$

holds. Thus, $(X_1, X_2{-}X_1, \ldots, X_q{-}X_{q-1}, \overline{X}_q)$ is the MC-partition over $\mathcal{C}(N)$ that represents all min-cuts $\{X_i, \overline{X}_i\}$ $(i = 1, \ldots, q)$. □

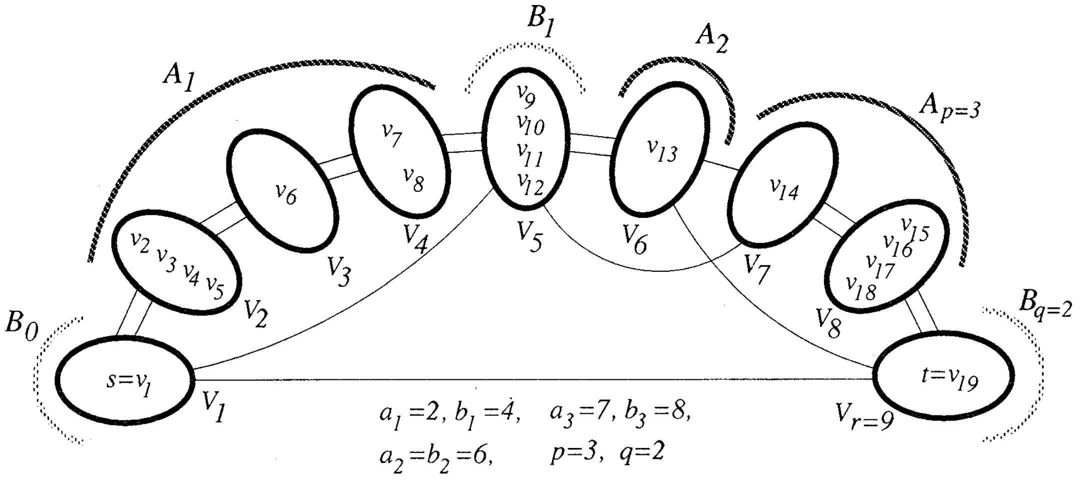


Figure 4: The (v_1, v_{19})-MC partition $\pi_{(v_1, v_{19})}$ over $\mathcal{C}(N_1)$ and its partition blocks $\{A_i\}$ and $\{B_i\}$.

The fact that $(s,t) \in E$ (i.e., $c(s,t;N) > 0$) is important in this lemma. For example, let N be a 4-cycle on the vertex set $\{v_0, v_1, v_2, v_3\}$, with edge set $\{(v_i, v_{i+1 (\mathrm{mod}\ 4)}) \mid 0 \le i \le 3\}$ such that each edge has a unit weight. Then, $\pi = (\{v_0\}, \{v_1\}, \{v_3\}, \{v_2\})$ is an MC-partition over $\mathcal{C}(N)$. However, the min-cut $\{\{v_0, v_3\}, \{v_1, v_2\}\}$, separating v_0 and v_2, does not belong to π. Note in this case that $(v_0, v_2) \notin E$.

The MC-partition in Lemma 3.4 for a critical edge (s,t) is unique. We call it the (s,t)-**MC-partition** over $\mathcal{C}(N)$, and denote it by $\pi_{(s,t)}$. For example, the (v_1, v_{19})-MC-partition, $\pi_{(v_1,v_{19})}$, of the network in Fig. 1 is $\pi_{(v_1,v_{19})} = (\{v_1\}, \{v_2, v_3, v_4, v_5\}, \{v_6\}, \{v_7, v_8\}, \{v_9, v_{10}, v_{11}, v_{12}\}, \{v_{13}\}, \{v_{14}\}, \{v_{15}, v_{16}, v_{17}, v_{18}\}, \{v_{19}\})$ as shown in Fig. 4.

It turns out that $\pi_{(s,t)}$ can be computed efficiently by the following lemma.

Lemma 3.5 [11, 18] *Let $(s,t) \in E$ be a critical edge in a network $N = (V, E, c)$. The (s,t)-MC-partition, $\pi_{(s,t)}$, over $\mathcal{C}(N)$ can be obtained by finding a maximum flow between source s and sink t in N.*

Proof sketch: Picard and Queyranne[19] introduced a directed acyclic graph which represents all st-cuts with weight $\lambda(s,t;N)$, and Ball and Provan[1] describe an algorithm that computes such a directed graph from a maximum flow[5] between s and t. Karzanov and Timofeev[11] and also Naor and Vazirani[18] show how to construct $\pi_{(s,t)}$ from the directed graph. □

It can be shown that, after finding a maximum flow between such s and t, $\pi_{(s,t)}$ can be constructed in $O(|V|+|E|)$ time [1, 11, 18].

We now consider the other min-cuts that do not separate s and t for any critical edge (s,t), i.e., the min-cuts not belonging to any (s,t)-MC partition. We say that a cut $\{X, \overline{X}\}$ **crosses** a partition $\{V_1, \ldots, V_r\}$ (or an o-partition $(V_1, \ldots, V_r)$) of V if $\{X, \overline{X}\}$ crosses some cut of the form $\{V_i, V - V_i\}$. A cut $\{X, \overline{X}\}$ is **compatible** with a partition $\{V_1, \ldots, V_r\}$ (or an o-partition $(V_1, V_2, \ldots, V_r)$) of V, if

$$V_i \subseteq X \quad \text{or} \quad V_i \subseteq \overline{X} \quad \text{for all } i = 1, 2, \ldots, r.$$

We prove below the first set of our new results which implies that, by solving a maximum flow between two vertices s and t, not only minimum cuts that separate s and t, but also many other minimum cuts which do not separate s and t can be obtained.

Lemma 3.6 *Let (s,t) be any critical edge in a network $N = (V, E, c)$. Then no min-cut $\{X, \overline{X}\} \in \mathcal{C}(N)$ crosses the (s,t)-MC-partition, $\pi_{(s,t)}$, over $\mathcal{C}(N)$.*

Proof: See Appendix 1. □

Given an o-partition, π, of V for a network $N = (V, E, c)$, let $\mathcal{C}_{comp}(\pi)$ denote the set of all min-cuts in $\mathcal{C}(N)$ that are compatible with π.

Lemma 3.7 *For a critical edge (s,t) in a network $N = (V, E, c)$, let $\pi_{(s,t)} = (V_1, \ldots, V_r)$ be the (s,t)-MC-partition over $\mathcal{C}(N)$, and let $\{X, \overline{X}\} \in \mathcal{C}_{comp}(\pi_{(s,t)})$. Then, we have the following:*

(i) *If $\{X, \overline{X}\}$ separates s and t, then either $X = V_{(i,r)}$ or $\overline{X} = V_{(i,r)}$ for some i $(1 < i \le r)$.*

(ii) *If $\{X, \overline{X}\}$ does not separate s and t, then $X = V_{(i,j)}$ or $\overline{X} = V_{(i,j)}$ for some i, j $(1 < i \le j < r)$ such that $\{V_k, \overline{V_k}\} \in \mathcal{C}(N)$ for each k $(i \le k \le j)$.*

Proof: (i) Immediate from Lemma 3.4. (ii) See Appendix 2. □

Clearly, by this lemma, one can compute $\mathcal{C}_{comp}(\pi_{(s,t)})$ in polynomial time from a given (s,t)-MC-partition, $\pi_{(s,t)}$. In Section 6, we will see that $\mathcal{C}_{comp}(\pi_{(s,t)})$ can be computed from $\pi_{(s,t)}$ in linear time, and furthermore it has a cactus representation.

4. Algorithm Outline

This section gives an overview of our algorithm (to be presented in Section 7) for constructing a cactus representation for all min-cuts of a network N. To this end, we first review a previously known algorithm for enumerating all min-cuts of N.

4.1. Enumerating All Min-Cuts

Suppose that we have the (s,t)-MC-partition, $\pi_{(s,t)}$, of V for a critical edge (s,t) in $N=(V,E,c)$. We thus know all min-cuts in $\mathcal{C}(N)$ that separate s and t. In order to find other min-cuts in $\mathcal{C}(N)$, we need not consider any cut separating s and t. We can therefore identify t with s, changing all edges of the form $e=(t,w)$ to $e=(s,w)$ and deleting t and all the resulting self-loops, if any. Let $N_{t=s}$ denote the resulting network, which inherits its weight function from N. Let $\{X,\overline{X}\}$ be any cut of N such that $\{s,t\}\subseteq X$. Then, clearly, $\{X{-}t,\overline{X}\}$ is a cut of $N_{t=s}$ satisfying $E(X{-}t,V{-}X;N_{t=s})=E(X,\overline{X};N)$. From this, it follows that $\lambda(N_{t=s})\geq\lambda(N)$. If $\lambda(N_{t=s})>\lambda(N)$, then all min-cuts separate s and t, and we know all of them. If $\lambda(N_{t=s})=\lambda(N)$, on the other hand, the rest of the min-cuts of N can be found as the min-cuts of $N_{t=s}$. Based on this observation, one can enumerate all cuts in $\mathcal{C}(N)$ by repeating edge contraction until the network has only two vertices:

```
Algorithm CONTRACT {Input: a network $N=(V,E,c)$. Output: $\mathcal{C}(N)=\bigcup_{i=2}^{|V|}\mathcal{C}^{(i)}$.}
begin
  Compute $\lambda(N)$; $\lambda:=\lambda(N)$; $N^{(|V|)}:=N$;
  for $i=|V|,|V|-1,\ldots,2$ do
    begin
      Choose an arbitrary pair of adjacent vertices $s^{(i)},t^{(i)}$ in $N^{(i)}$
        and test if $\lambda(s^{(i)},t^{(i)};N^{(i)})=\lambda$;
      If so, compute the set $\mathcal{C}^{(i)}$ of all min-cuts separating $s^{(i)}$ and $t^{(i)}$ in $N^{(i)}$
        and store it as the $(s^{(i)},t^{(i)})$-MC-partition
      else, let $\mathcal{C}^{(i)}:=\emptyset$;
      $N^{(i-1)}:=N^{(i)}_{t^{(i)}=s^{(i)}}$
    end
end.
```

CONTRACT divides the set $\mathcal{C}(N)$ into disjoint subsets $\mathcal{C}^{(i)}$, $i=|V|,|V|-1,\ldots,2$, and outputs $\mathcal{C}(N)=\bigcup_{i=2}^{|V|}\mathcal{C}^{(i)}$. By Lemma 3.5, $\mathcal{C}^{(i)}$ can be computed by finding a maximum flow between $s^{(i)}$ and $t^{(i)}$ in $N^{(i)}$. Since the $(s^{(i)},t^{(i)})$-MC-partition for $\mathcal{C}^{(i)}$ represents at most $O(|V|)$ min-cuts, $\mathcal{C}(N)=\bigcup_{i=2}^{|V|}\mathcal{C}^{(i)}$ contains at most $O(|V|^2)$ min-cuts. Clearly, the amount of space required to store $(s^{(i)},t^{(i)})$-MC-partition for all i is also $O(|V|^2)$.

4.2. Constructing Cactus Representation

In CONTRACT, after finding all $(s^{(i)},t^{(i)})$-MC-partitions, $\pi_{(s^{(i)},t^{(i)})}$, it may be possible to merge the obtained MC-partitions compactly into a single structure, that is, a uniform cactus. The previously known algorithms [11, 18] for finding a cactus representation for $\mathcal{C}(N)$ are based on this idea, and can be described as follows.

```
Algorithm CACTUS {Input: a network $N=(V,E,c)$. Output: a cactus
  representation $(\mathcal{R},\varphi)$ for $\mathcal{C}(N)$.}
1 begin
2 Compute $\lambda(N)$; $N^{(|V|)}:=N$; let $s^{(|V|+1)}$ be a vertex in $V$;
```

```
3  for $i = |V|, |V|-1, \ldots, 2$ do
4    begin
5      $s^{(i)} := s^{(i+1)}$; choose as $t^{(i)}$ a vertex adjacent to $s^{(i)}$ in $N^{(i)}$;
6      $N^{(i-1)} := N^{(i)}_{t^{(i)}=s^{(i)}}$
7    end
8  $\mathcal{R}^{(1)} := (x, \emptyset)$;  $\varphi^{(1)}(V) := x$;  $\{(\mathcal{R}^{(1)}, \varphi^{(1)})$ is a cactus representation for
   $\mathcal{C} = \emptyset\}$
9  for $i = 2, 3, \ldots, |V|$ do
10   begin
11     Compute an $(s^{(i)}, t^{(i)})$-MC-partition $\pi_{(s^{(i)},t^{(i)})}$ over $\mathcal{C}(N^{(i)})$;
12     Compute a cactus representation $(\mathcal{R}^{(i)}, \varphi^{(i)})$ for $\mathcal{C}(N^{(i)})$ from $\pi_{(s^{(i)},t^{(i)})}$
       and $(\mathcal{R}^{(i-1)}, \varphi^{(i-1)})$
13   end
14 $(\mathcal{R}, \varphi) := (\mathcal{R}^{(|V|)}, \varphi^{(|V|)})$
15 end.
```

In the above algorithm $\pi_{(s^{(i)},t^{(i)})}$ must be computed for the specific pair of vertices $s^{(i)}, t^{(i)}$, chosen in line 5, to simplify the merging step in line 12. Therefore, we need an algorithm that can find a maximum flow between a *prescribed* pair of source and sink. The reason why the merging step would become much more complicated unless $s^{(i)}$ and $t^{(i)}$ were chosen carefully is that the algorithms in [11, 18] make use of only those min-cuts which belong to the $(s^{(i)}, t^{(i)})$-MC-partition, $\pi_{(s^{(i)},t^{(i)})}$. However, as shown in Lemma 3.7(ii) (and in Section 6 in more detail), $\pi_{(s^{(i)},t^{(i)})}$ may reveal many other min-cuts. In fact, it will turn out that those additional min-cuts appear both in $\pi_{(s^{(i)},t^{(i)})}$ and $(\mathcal{R}^{(i-1)}, \varphi^{(i-1)})$ of line 12. In other words, those min-cuts will give us an indication on how to combine $\pi_{(s^{(i)},t^{(i)})}$ and $(\mathcal{R}^{(i-1)}, \varphi^{(i-1)})$ properly. By making the maximum use of the $(s^{(i)}, t^{(i)})$-MC-partition, $\pi_{(s^{(i)},t^{(i)})}$, our new algorithm will choose the pair $s^{(i)}$ and $t^{(i)}$ arbitrarily in line 5 and have a simpler merging step in line 12.

5. Union of Representations

To implement step 12 of Algorithm CACTUS, this section introduces an operation for combining the representations for two different sets of min-cuts of N into a single representation.

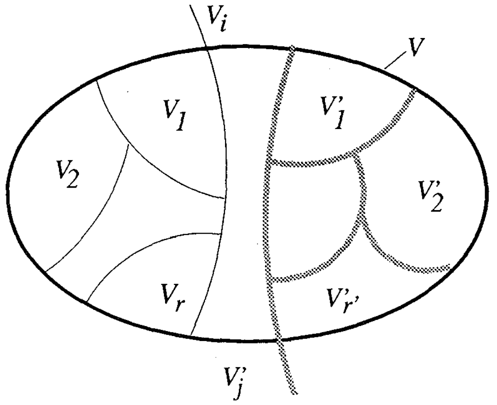


Figure 5: Two complementary partitions.

We say that a representation $(\mathcal{R}, \varphi)$ **induces** the partition $\{V_1, V_2, \ldots, V_r\}$ of V, if $\mathcal{R}$ has exactly r non-empty nodes, x_1, $x_2, \ldots,$ x_r such that $\varphi^{-1}(x_i) = V_i$. Two partitions $\{V_1, \ldots, V_r\}$ and $\{V'_1, \ldots, V'_{r'}\}$ (or two o-partitions $(V_1, \ldots, V_r)$ and $(V'_1, \ldots, V'_{r'})$) of V, where $r, r' \geq 2$, are said to be **complementary** (with respect to V_i and V'_j), if there are two subsets V_i, V'_j such that

$$V_i \cup V'_j = V.$$

(See Fig. 5.) Thus, two cuts either cross each other or are complementary. Similarly, a cut and a partition may cross each other, be complementary or compatible. They can be complementary and compatible at the same time.

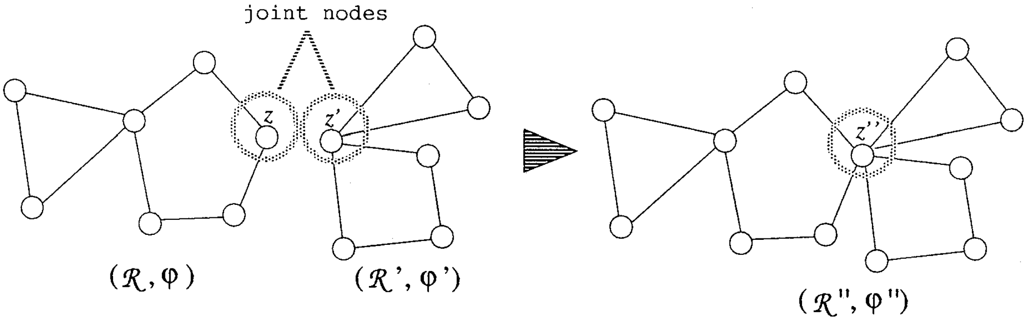


Figure 6: Union of two representations.

Consider two cacti $\mathcal{R} = (W, F)$ and $\mathcal{R}' = (W', F')$, where $W \cap W' = \emptyset$, such that $(\mathcal{R}, \varphi)$ and $(\mathcal{R}', \varphi')$ are representations for $\mathcal{C} \subseteq \mathcal{C}(N)$ and $\mathcal{C}' \subseteq \mathcal{C}(N)$, respectively. These two representations are said to be **complementary**, if the two partitions, $\{V_1, V_2, \ldots, V_r\}$ and $\{V'_1, V'_2, \ldots, V'_{r'}\}$, of V induced by $(\mathcal{R}, \varphi)$ and $(\mathcal{R}', \varphi')$, respectively, are complementary. (This implies $r, r' \geq 2$.) If $(\mathcal{R}, \varphi)$ and $(\mathcal{R}', \varphi')$ are complementary, then there are two nodes, $z \in W$ and $z' \in W'$, such that

$$\varphi^{-1}(z) \cup \varphi'^{-1}(z') = V.$$

We call these z and z' **joint nodes**. The **union**, denoted by

$$(\mathcal{R}'', \varphi'') = (\mathcal{R}, \varphi) \oplus (\mathcal{R}', \varphi'),$$

of two complementary representations, $(\mathcal{R}, \varphi)$ and $(\mathcal{R}', \varphi')$, is defined as the cactus obtained from $\mathcal{R}$ and $\mathcal{R}'$ by identifying the two joint nodes, z, z' as a new node, say z'', and defining a mapping $\varphi'' : V \to W \cup W' \cup \{z''\} - \{z, z'\}$ such that

$$\varphi''^{-1}(z'') = \varphi^{-1}(z) \cap \varphi'^{-1}(z'),$$

$$\varphi''^{-1}(x) = \varphi^{-1}(x) \text{ for } x \in W{-}z, \text{ and } \varphi''^{-1}(x') = \varphi'^{-1}(x') \text{ for } x' \in W'{-}z',$$

as shown in Fig. 6. Note that after the above union operation, the node z becomes an empty node if and only if the joint nodes z and z' satisfy $\varphi^{-1}(z) \cap \varphi'^{-1}(z') = \emptyset$, i.e., $\varphi^{-1}(W{-}z) = \varphi'^{-1}(z')$ or equivalently $\varphi'^{-1}(W'{-}z') = \varphi^{-1}(z)$. The significance of the union operation is given by the following lemma.

Lemma 5.1 *Let a representation $\mathcal{R} = (W, F)$ for $\mathcal{C} \subseteq \mathcal{C}(N)$ and a representation $\mathcal{R}' = (W', F')$ for $\mathcal{C}' \subseteq \mathcal{C}(N)$ be complementary. Then $(\mathcal{R}'', \varphi'') = (\mathcal{R}, \varphi) \oplus (\mathcal{R}', \varphi')$ is a representation for $\mathcal{C} \cup \mathcal{C}'$.*

Proof: Let $\mathcal{R}'' = (W'', F'')$, where $W'' = W \cup W' \cup \{z''\} - \{z, z'\}$. Let $\{X, \overline{X}\} \in \mathcal{C} \cup \mathcal{C}'$. Consider the case $\{X, \overline{X}\} \in \mathcal{C}$ (the other possibility, i.e., $\{X, \overline{X}\} \in \mathcal{C}'$ can be treated analogously), and let $\{S, W{-}S\}$ be the corresponding cut in $\mathcal{C}(\mathcal{R})$, where we assume $X = \varphi^{-1}(S)$ and $z \in W{-}S$ without loss of generality. By definition, we have $\{S, W''{-}S\} \in \mathcal{C}(\mathcal{R}'')$, $\varphi''^{-1}(S) = X$, and $\varphi''^{-1}(W''{-}S) = \overline{X}$. Conversely, let $\{T, W''{-}T\}$ be an arbitrary cut in $\mathcal{C}(\mathcal{R}'')$, where we assume $z'' \in W''{-}T$ without loss of generality. Then, $T \subseteq W{-}z$ or $T \subseteq W'{-}z'$ holds, because, otherwise, the network induced by T from $\mathcal{R}''$ would not be connected, implying a cut smaller than a min-cut, a contradiction. Therefore, $\{T, W''{-}T\}$ corresponds to a cut in $\mathcal{C} \cup \mathcal{C}'$. □

6. (s,t)-Cactus Representation

Throughout this section, we assume that the edge $(s,t) \in E$ in a network $N = (V, E, c)$ is critical, and $\pi_{(s,t)} = (V_1, \ldots, V_r)$ denotes the (s,t)-MC-partition over $\mathcal{C}(N)$. We will show how to construct a cactus representation for $\mathcal{C}_{comp}(\pi_{(s,t)})$ (i.e., the set of all min-cuts compatible with the (s,t)-MC-partition), which we call the "(s,t)-cactus representation."

6.1. Min-Cuts Compatible with (s,t)-MC-Partition

Using Lemma 3.7, one could enumerate all min-cuts in $\mathcal{C}_{comp}(\pi_{(s,t)})$ by checking whether $c(V_{(i,j)}, \overline{V_{(i,j)}}; N) = \lambda(N)$ for all possible pairs, (i,j). However, there are as many as $O(r^2) = O(|V|^2)$ such pairs. To find them more efficiently, we first show that the set $\mathcal{C}_{comp}(\pi_{(s,t)})$ can be represented by a collection of MC- and/or circular MC-partitions over $\mathcal{C}(N)$, based on which $\mathcal{C}_{comp}(\pi_{(s,t)})$ can be computed from $\pi_{(s,t)}$ in linear time.

In order to apply Lemma 3.7, we first partition $\{V_1, V_2, \ldots, V_r\}$ into

$$A = \{V_i \mid c(V_i, \overline{V_i}; N) = \lambda(N),\ 1 < i < r\},\ \text{and}$$

$$B = \{V_1, V_r\} \cup \{V_i \mid c(V_i, \overline{V_i}; N) > \lambda(N),\ 1 < i < r\}.$$

If we start with $\pi = (V_1, V_2, \ldots, V_r)$ and remove from π all $V_i \in A$, we are left with a sequence of *segments*, each consisting of one or more partition blocks from $\{V_i\}$ with contiguous indices. Suppose that there are $q{+}1$ such segments. For $j = 0, 1, \ldots, q$, we use B_j to denote the set of partition blocks from $\{V_i\}$ belonging to segment j, with $V_1 \in B_0$ and $V_r \in B_q$. Clearly, $\{B_j \mid j = 0, 1, \ldots, q\}$ is a partition of B.

A can also be considered as consisting of contiguous segments as follows: start with $\pi = (V_1, V_2, \ldots, V_r)$ and remove from π all $V_i \in B$. This gives rise to exactly q segments. We now proceed to partition A into at least p ($p \geq q$) subsets, $A_1, A_2 \ldots, A_p$, such that each segment gives rise to at least one subset A_i. More specifically, $A_i = \{V_{a_i}, V_{a_i+1}, \cdots, V_{b_i}\}$, $i = 1, 2, \ldots, p$, are defined as follows:

Let $b_0 = 0$, let V_{a_i} be the subset in A with the smallest index a_i satisfying $b_{i-1} < a_i$, and let V_{b_i} be the subset in A with the largest index b_i satisfying

$$c(V_{(a_i,k)}, \overline{V_{(a_i,k)}}; N) = \lambda(N) \text{ and } V_k \in A \ \text{ for any } k, a_i \leq k \leq b_i. \tag{6.1}$$

It is easy to see that, for each B_j, $j = 1, \ldots, q-1$, there is an index i such that

$$B_j = \{V_{b_i+1}, V_{b_i+2}, \ldots, V_{a_{i+1}-1}\}.$$

For example, Fig. 4 shows the partition blocks $\{A_1, A_2, A_3\}$ and $\{B_0, B_1, B_2\}$ for the (v_1, v_{19})-MC-partition $\pi_{(v_1,v_{19})} = (V_1, V_2, \ldots, V_{r=9})$ over $\mathcal{C}(N_1)$ for the critical edge (v_1, v_{19}) in N_1 of Fig. 1.

Lemma 6.1 *Let $\{A_i \mid i = 1, \ldots, p\}$ and $\{B_j \mid j = 0, 1, \ldots, q\}$ be as defined above for the (s,t)-MC-partition for a critical edge $(s,t) \in E$ in $N = (V, E, c)$. Then,* (i), (ii) *and* (iii) *below hold.*

(i) *The o-partition $\pi_{B_0} = (V_1, \ldots, V_{a_1-1}, V_{(a_1,r)})$ of V is an MC-partition over $\mathcal{C}(N)$, the o-partition $\pi_{B_q} = (V(1, b_p), V_{b_p+1}, \ldots, V_r)$ of V is an MC-partition over $\mathcal{C}(N)$, and, for each j $(1 \le j \le q-1)$, the o-partition of V associated with $B_j = \{V_{b_i+1}, V_{b_i+2}, \ldots, V_{a_{i+1}-1}\}$,*

$$\pi_{B_j} = (V_{(1,b_i)}, V_{b_i+1}, V_{b_i+2}, \ldots, V_{a_{i+1}-1}, V_{(a_{i+1},r)})$$

is an MC-partition over $\mathcal{C}(N)$.

(ii) *For each i $(1 \le i \le p)$, the o-partition of V associated with A_i,*

$$\pi_{A_i} = (V_{(1,a_i-1)}, V_{a_i}, V_{a_i+1}, \ldots, V_{b_i}, V_{(b_i+1,r)})$$

is a maximal circular MC-partition over $\mathcal{C}(N)$.

(iii) *$\mathcal{C}_{comp}(\pi_{(s,t)})$ consists only of the min-cuts belonging to the MC-partitions in* (i) *and* (ii) *above.*

Proof: (i) Immediate from the definition of (s,t)-MC-partitions.
(ii) and (iii) See Appendix 3. □

Note that a cut in $\mathcal{C}(N)$ may belong to more than one MC-partition in $\{\pi_{B_j} \mid j = 1, \ldots, q\} \cup \{\pi_{A_i} \mid i = 1, \ldots, p\}$ in the above lemma. For example, the cut $\{V_{(1,b_i)}, V_{(b_i+1,r)}\}$ belongs to the MC-partition π_{B_j} for some B_j as well as to the circular MC-partition π_{A_i}.

6.2. (s,t)-Cactus Construction

This section proves that there exists a cactus representation for $\mathcal{C}_{comp}(\pi_{(s,t)})$.

For each B_j, $j = 0, 1, \ldots, q$, let $(\ddot{\mathcal{R}}_{\pi_{B_j}}, \psi_{\pi_{B_j}})$ be the chain representation for the min-cuts belonging to the MC-partition π_{B_j} over $\mathcal{C}(N)$ defined in Section 3.1, and for each A_i, $i = 1, 2, \ldots, p$, let $(\mathring{\mathcal{R}}_{\pi_{A_i}}, \phi_{\pi A_i})$ be the cycle representation for the min-cuts belonging to the circular MC-partition π_{A_i} over $\mathcal{C}(N)$ defined in Section 3.1.

In our running example (Fig. 4), $\ddot{\mathcal{R}}_{\pi_{B_0}}$ has two nodes and represents the MC-partition $\pi_{B_0} = (V_1, V_{(2,9)})$. Similarly, $\mathring{\mathcal{R}}_{\pi_{A_1}}$ has five nodes and represents the MC-partition $\pi_{A_1} = (V_1, V_2, V_3, V_4, V_{(5,9)})$. Since $V_1 \cup V_{(2,9)} = V$, these two MC-partitions are complementary, and, therefore, their union $(\ddot{\mathcal{R}}_{\pi_{B_0}}, \psi_{\pi_{B_0}}) \oplus (\mathring{\mathcal{R}}_{\pi_{A_1}}, \phi_{\pi_{A_1}})$ can be formed, with the nodes corresponding to the partition blocks $V_{(2,9)}$ of π_{B_0} and V_1 of π_{A_1} as the joint nodes. By Lemma 5.1, the resulting cactus represents all min-cuts that are represented by $(\ddot{\mathcal{R}}_{\pi_{B_0}}, \psi_{\pi_{B_0}})$ or $(\mathring{\mathcal{R}}_{\pi_{A_1}}, \phi_{\pi_{A_1}})$.

Continuing as above, we can construct a single cactus which represents all min-cuts in $\mathcal{C}_{comp}(\pi_{(s,t)})$. More formally, a cactus representation $(\mathcal{R}_{(s,t)}, \varphi_{(s,t)})$ for $\mathcal{C}_{comp}(\pi_{(s,t)})$ is obtained as follows:

```
Let $(\mathcal{R}, \varphi) := (\ddot{\mathcal{R}}_{\pi_{B_0}}, \psi_{\pi_{B_0}})$.
For $i = 1, 2, \ldots, p$, do $\{(\mathcal{R}, \varphi) := (\mathcal{R}, \varphi) \oplus (\mathring{\mathcal{R}}_{\pi_{A_i}}, \phi_{\pi_{A_i}})$;
    if $b_i+1 \neq a_{i+1}$ then $(\mathcal{R}, \varphi) := (\mathcal{R}, \varphi) \oplus (\ddot{\mathcal{R}}_{\pi_{B_j}}, \psi_{\pi_{B_j}})$, where $V_{b_i+1} \in B_j$.$\}$
Let $(\mathcal{R}_{(s,t)}, \varphi_{(s,t)}) := (\mathcal{R}, \varphi)$.
```

Note that if b_i+1 = a_{i+1} then there is no intervening B-segment between A_i and A_{i+1}. Since the two operands in each union $(\mathcal{R},\varphi)\oplus(\mathring{\mathcal{R}}_{\pi_{A_i}},\phi_{\pi_{A_i}})$ (or $(\mathcal{R},\varphi)\oplus(\ddot{\mathcal{R}}_{\pi_{B_j}}\psi_{\pi_{B_j}})$) are complementary, Lemma 5.1 is applicable, and the obtained $(\mathcal{R}_{(s,t)},\varphi_{(s,t)})$ is a cactus representation for $\mathcal{C}_{comp}(\pi_{(s,t)})$. Note also that, in the above procedure, every time a union is formed, the two joint nodes turn into an empty node. This implies that the mapping $\varphi_{(s,t)}$ can be easily obtained from $\psi_{\pi_{B_i}},\phi_{\pi_{A_i}}$ without any extra computation. (See the paragraph just before Lemma 5.1.) It is easy to see that $\mathcal{R}_{(s,t)}$ contains p+q empty nodes.

For example, Fig. 7 shows the (v_1,v_{19})-cactus representation obtained by the above procedure from the (v_1,v_{19})-MC-partition $\pi_{(v_1,v_{19})}$ in Fig. 4. It is seen from Fig. 7 that, intuitively, the above procedure introduces p+q empty nodes, and insert cycles (for $\{A_i\}$) and chains (for $\{B_j\}$) in the order in which they appear in the (s,t)-MC-partition.

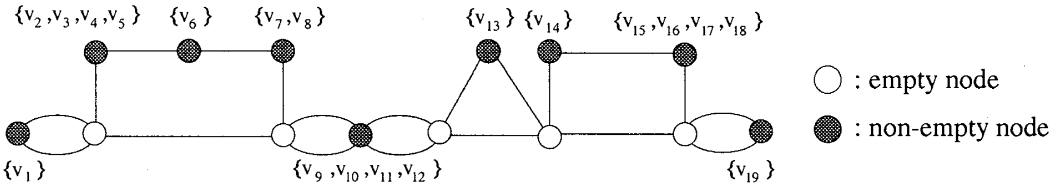


Figure 7: The (v_1,v_{19})-cactus representation for $\mathcal{C}_{comp}(\pi_{(v_1,v_{19})})$.

From the above discussions, we have the following theorem.

Theorem 6.2 *Let* (s,t) *be any critical edge in a network* $N=(V,E,c)$. *Then there is a cactus representation for the set of all min-cuts that are compatible with the* (s,t)*-MC-partition over* $\mathcal{C}(N)$. □

We call the cactus representation $(\mathcal{R}_{(s,t)},\varphi_{(s,t)})$ for $\mathcal{C}_{comp}(\pi_{(s,t)})$ obtained by the above procedure the (s,t)-**cactus representation**. The (s,t)-cactus representation may not be a cycle-type normal cactus representation. For example, the (v_1,v_{19})-cactus representation of Fig. 7 is not normal.

Theorem 6.3 *The* (s,t)*-cactus representation can be constructed from the* (s,t)*-MC-partition in* $O(|E|+|V|)$ *time and* $O(|E|+|V|)$ *space.*

Proof: A skeleton procedure for determining all A_i, $i=1,2,\ldots,p$ using (6.1) would look like:

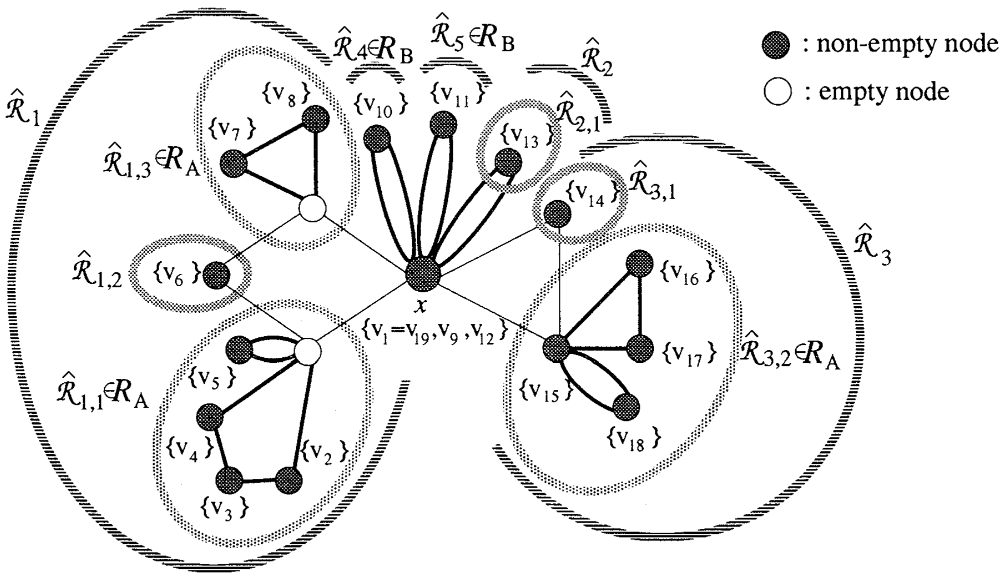
```
$a := k := 2$;
**while** $a < r$ **do** {
  **while** $c(V_{(a,k)},\overline{V_{(a,k)}})=\lambda(N)$ **do** {$k := k+1$};
  $a := k := k+1$};
```

Whenever the inner while-loop is executed one or more times consecutively, a new subset A_i is determined. To analyze the complexity of the above procedure, imagine that the sets $V_1, V_2,\ldots, V_r$ are arranged from left to right. Intuitively, as the above procedure is executed, the set $V_{(a,k)}$ moves from left to right. The edge set $E(V_{(a,k)},\overline{V_{(a,k)}})$ may expand and shrink. As a and k change during execution, each edge enters $E(V_{(a,k)},\overline{V_{(a,k)}})$ and leaves it at most twice. Therefore, the above procedure requires at most $O(|E|+|V|)$ time. We can then construct the cactus $\mathcal{R}_{(s,t)}$ in $O(|V|)$ time and $O(|V|)$ space, since $\mathcal{R}_{(s,t)}$ has at most $O(|V|)$ nodes and edges. Computing the mapping $\varphi_{(s,t)}$ for $\mathcal{R}_{(s,t)}$ can be done in $O(|V|)$ time and $O(|V|)$ space as commented in the paragraph just before Theorem 6.2. □

For a non-critical edge (u,v), we define the (u,v)-cactus representation to be $(\mathcal{R},\varphi)$, where $\mathcal{R}=(\{x\},\emptyset)$ and $\varphi(V)=x$. We will find this convention useful in the next section.

7. Complete Cactus Representation

In the previous section, given the (s,t)-MC-partition $\pi_{(s,t)}$ for a critical edge (s,t), we showed how to construct the (s,t)-cactus representation $(\mathcal{R}_{(s,t)}, \varphi_{(s,t)})$, i.e., a cactus representation for the set, $\mathcal{C}_{comp}(\pi_{(s,t)})$, of all min-cuts compatible with $\pi_{(s,t)}$.



$\hat{\mathcal{R}}_1$
$\hat{\mathcal{R}}_{1,3} \in \mathcal{R}_A$
$\{v_8\}$
$\{v_7\}$
$\hat{\mathcal{R}}_4 \in \mathcal{R}_B$
$\hat{\mathcal{R}}_5 \in \mathcal{R}_B$
$\{v_{10}\}$
$\{v_{11}\}$
$\hat{\mathcal{R}}_2$
$\hat{\mathcal{R}}_{2,1}$
$\{v_{13}\}$
$\{v_{14}\}$
$\hat{\mathcal{R}}_{3,1}$
● : non-empty node
○ : empty node
$\hat{\mathcal{R}}_3$
$\hat{\mathcal{R}}_{1,2}$
$\{v_6\}$
x
$\{v_1 = v_{19}, v_9, v_{12}\}$
$\{v_{16}\}$
$\{v_{17}\}$
$\hat{\mathcal{R}}_{3,2} \in \mathcal{R}_A$
$\{v_{15}\}$
$\{v_{18}\}$
$\hat{\mathcal{R}}_{1,1} \in \mathcal{R}_A$
$\{v_5\}$
$\{v_4\}$
$\{v_2\}$
$\{v_3\}$

Figure 8: The cycle-type normal cactus representation for $\mathcal{C}(N_{1 v_1 = v_{19}})$.

In general, $\mathcal{C}(N) - \mathcal{C}_{comp}(\pi_{(s,t)})$ is non-empty. According to Lemma 3.6, no min-cut in $\mathcal{C}(N) - \mathcal{C}_{comp}(\pi_{(s,t)})$ crosses the (s,t)-MC-partition. Therefore, all min-cuts in $\mathcal{C}(N) - \mathcal{C}_{comp}(\pi_{(s,t)})$ are complementary with $\pi_{(s,t)}$, and no min-cut in $\mathcal{C}(N) - \mathcal{C}_{comp}(\pi_{(s,t)})$ separates s and t. The latter implies that each min-cut in $\mathcal{C}(N) - \mathcal{C}_{comp}(\pi_{(s,t)})$ has its counterpart in $N_{t=s}$, i.e., the network obtained from N by identifying t and s (see Section 4.1). For example, Fig. 8 illustrates a cycle-type normal cactus representation for $\mathcal{C}(N_{1, v_{19} = v_1})$.

The main objective of this section is to show how to construct a cactus representation for $\mathcal{C}(N)$ from the (s,t)-cactus representation $(\mathcal{R}_{(s,t)}, \varphi_{(s,t)})$ for a critical edge (s,t) in $N = (V, E, c)$ and the cycle-type normal cactus representation for $\mathcal{C}(N_{t=s})$. Note that a cactus representation for $\mathcal{C}(N)$ for N with $|V| = 2$ can be easily constructed and $N_{t=s}$ has one less vertex than N. Therefore, the method can be recursively applied, together with a "normalization" operation (see the proof of Lemma 3.3 in [16]) and the (s,t)-cactus construction method discussed in Section 6, to construct the cycle-type normal cactus representation for $\mathcal{C}(N)$ for any network N.

If $\lambda(N_{t=s}) > \lambda(N)$, clearly, $\mathcal{C}(N) - \mathcal{C}_{comp}(\pi_{(s,t)}) = \emptyset$, and an (s,t)-cactus representation is a cactus representation for $\mathcal{C}(N)$. In the rest of this section, we assume that (s,t) is a critical edge and $\lambda(N_{t=s}) = \lambda(N)$. Let $\hat{N}$ denote $N_{t=s}$, and $(\hat{\mathcal{R}}, \hat{\varphi})$ be the cycle-type normal cactus representation for $\mathcal{C}(\hat{N})$, and x be the node in $\hat{\mathcal{R}}$ such that $s \in \hat{\varphi}^{-1}(x)$. Starting with the (s,t)-cactus representation $(\mathcal{R}_{(s,t)}, \varphi_{(s,t)})$, we "graft" subgraphs of the cactus representation $(\hat{\mathcal{R}}, \hat{\varphi})$ for $\mathcal{C}(\hat{N})$ one by one, using the union operation. In order to be "unioned" together, the two cactus representations must be complementary. This puts restrictions on which subgraphs of the cactus representation $(\hat{\mathcal{R}}, \hat{\varphi})$ for $\mathcal{C}(\hat{N})$ can be "grafted." In fact, our main efforts in this section will be on determining "qualified" candidate subgraphs. Not only do we have to find qualified candidates, but we also need to "decompose" the cactus representation $(\hat{\mathcal{R}}, \hat{\varphi})$ for $\mathcal{C}(\hat{N})$ in such a way that all information regarding the min-cuts in $\mathcal{C}(N) - \mathcal{C}_{comp}(\pi_{(s,t)})$ stored in it is extracted from it as a subgraph.

7.1. Maximal Circular MC-Partition over $\mathcal{C}(N_{t=s})$

Based on Lemma 3.3, this subsection investigates the relationship between the (s,t)-cactus representation $(\mathcal{R}_{(s,t)}, \varphi_{(s,t)})$ and the cycle-type normal cactus representation $(\hat{\mathcal{R}}, \hat{\varphi})$ for $\mathcal{C}(\hat{N})$.

For the (s,t)-MC-partition $\pi_{(s,t)} = (V_1, V_2, \ldots, V_r)$ over $\mathcal{C}(N)$, let a_i and b_i, $i = 1, 2, \ldots, p$, be the indices defined in Section 6. (See Fig. 4 for an example.) From Lemma 6.1(ii), we see that, for each i, the o-partition,

$$\pi_{A_i} = (V_{(1,a_i-1)}, V_{a_i}, V_{a_i+1}, \ldots, V_{b_i}, V_{(b_i+1,r)}),$$

of V is a maximal circular MC-partition over $\mathcal{C}(N)$. This implies that the o-partition,

$$\hat{\pi}_{A_i} = (V_{(1,a_i-1)} \cup V_{(b_i+1,r)} - t, V_{a_i}, V_{a_i+1}, \ldots, V_{b_i}), \tag{7.1}$$

of $V - t$ is a maximal circular MC-partition over $\mathcal{C}(\hat{N})$ if $a_i < b_i$ (in the case of $a_i = b_i$, $\hat{\pi}_{A_i} = (V_{(1,a_i-1)} \cup V_{(b_i+1,r)} - t, V_{a_i})$ may not be maximal if there is another maximal circular MC-partition $(V'_1, \ldots, V'_p)$ with $V'_1 = V - V_{a_i}$). Therefore, by Lemma 3.3, the cycle-type normal cactus representation $(\hat{\mathcal{R}}, \hat{\varphi})$ for $\mathcal{C}(\hat{N})$ has a $(b_i - a_i + 2)$-cycle C_i which yields $\hat{\pi}_{A_i}$. This fact plays a key role in Section 7.2 in finding from $\hat{\mathcal{R}}$ those min-cuts $(Y, \overline{Y}) \in \mathcal{C}(N) - \mathcal{C}_{comp}(\pi_{(s,t)})$ which are complementary with $\pi_{(s,t)}$ with respect to $\overline{Y}$ and some $V_k \in \{V_{a_i}, V_{a_i+1}, \ldots, V_{b_i}\}$, i.e., $Y \subset V_k$. Note, however, that the converse is not true. That is, a cycle in $\hat{\mathcal{R}}$ on which x lies may not always correspond to any maximal circular MC-partition over $\mathcal{C}(N)$ in Lemma 6.1(ii).

7.2. Decomposing Cactus Representation for $\mathcal{C}(N_{t=s})$

Again, the problem we face is the following:

- Given: the (s,t)-cactus representation $(\mathcal{R}_{(s,t)}, \varphi_{(s,t)})$ corresponding to the (s,t)-MC-partition $\pi_{(s,t)} = (V_1, V_2, \ldots, V_r)$, and the cycle-type normal cactus representation $(\hat{\mathcal{R}}, \hat{\varphi})$ for $\mathcal{C}(\hat{N})$.
- Find: a cactus representation for $\mathcal{C}(N)$.

As outlined in the introduction of Section 7, we need to "decompose" $(\hat{\mathcal{R}}, \hat{\varphi})$. We carry out the decomposition in two stages. In the first stage, we decompose $\hat{\mathcal{R}}$ into "blockcacti" at x (see below). Some of these blockcacti (i.e., those in R_B defined below) are "qualified," while others need to be decomposed further into "subcacti" (see (7.3)).

Let x be the node in $\hat{\mathcal{R}}$ such that $s \in \hat{\varphi}^{-1}(x)$, and remove x from $\hat{\mathcal{R}}$. Let $(\hat{W}'_i, \hat{F}'_i)$, $i = 1, 2, \ldots, h$, denote the connected components in $\hat{\mathcal{R}} - x$, where $\hat{W}'_i \subseteq \hat{W} - x$. For each $i = 1, 2, \ldots, h$, the network $\hat{\mathcal{R}}_i = (\hat{W}_i, \hat{F}_i)$ induced by $\hat{W}_i = \hat{W}'_i \cup \{x\}$ from $\hat{\mathcal{R}}$ is called a **blockcactus** at x of $\hat{\mathcal{R}}$. Intuitively, $\hat{\mathcal{R}}_i$ is obtained from $\hat{\mathcal{R}}$ by identifying all nodes not in $\hat{W}'_i$ with x. It is easy to see that it is a uniform cactus. Let us denote the node x in each $\hat{\mathcal{R}}_i$ by $x_i \in \hat{W}_i$. We now define a mapping $\hat{\varphi}_i : V \to \hat{W}_i$ by

$$\hat{\varphi}_i^{-1}(z) := \begin{cases} \hat{\varphi}^{-1}(z) & \text{if } z \in \hat{W}_i - x_i \\ V - \hat{\varphi}^{-1}(\hat{W}_i - x_i) & \text{if } z = x_i \end{cases} \tag{7.2}$$

(note that t now appears in $\hat{\varphi}_i^{-1}$). Clearly, each $(\hat{\mathcal{R}}_i, \hat{\varphi}_i)$, $i = 1, 2, \ldots, h$, is a cycle-type normal cactus representation for $\mathcal{C}_i$, where $\mathcal{C}_i$ denotes the set of all min-cuts in N that are compatible with the partition of V induced by $(\hat{\mathcal{R}}_i, \hat{\varphi}_i)$. We call each $(\hat{\mathcal{R}}_i, \hat{\varphi}_i)$ a **blockcactus representation** at x.

To state the next lemma, we need to define certain partition $A^* \cup B$ of V. For the (s,t)-MC-partition $\pi_{(s,t)} = (V_1, V_2, \ldots, V_r)$ in N, let a_i, b_i, $i = 1, 2, \ldots, p$, be the indices defined in the Section 6. (See Fig. 4 for an example.) Now define

$$A^* = \{V_{(a_i,b_i)} \mid i = 1, 2, \ldots, p\}.$$

B was defined in Section 6, i.e.,

$$B = \{V_1, \ldots, V_{a_1-1}\} \cup \{V_j \mid b_i < j < a_{i+1} \text{ for some } i = 2, \ldots, p-1\} \cup \{V_{b_p+1}, \ldots, V_r\}.$$

Clearly, $A^* \cup B$ is a partition of V. For example, A^* and B defined for the (v_1, v_{19})-MC-partition, $\pi_{(v_1,v_{19})} = (V_1, V_2, \ldots, V_9)$, in Fig. 4 are given by

$$A^* = \{V_{(2,4)} = \{v_2, v_3, v_4, v_5, v_6, v_7, v_8\}, V_6 = \{v_{13}\}, V_{(7,8)} = \{v_{14}, v_{15}, v_{16}, v_{17}, v_{18},\}\} \text{ and}$$

$$B = \{V_1 = \{v_1\}, V_5 = \{v_9, v_{10}, v_{11}, v_{12}\}, V_9 = \{v_{19}\}\}.$$

Lemma 7.1 *Given a network $N = (V, E, c)$ and the cycle-type normal cactus representation $(\hat{\mathcal{R}}, \hat{\varphi})$ for $\mathcal{C}(\hat{N})$, where $\hat{N} = N_{t=s}$ for some critical edge $(s,t) \in E$, let $(\hat{\mathcal{R}}_i, \hat{\varphi}_i)$, $i = 1, 2, \ldots, h$, be the blockcactus representations at x with $s \in \hat{\varphi}^{-1}(x)$, where $x_i \in \hat{W}_i$ denotes x in each $\hat{\mathcal{R}}_i = (\hat{W}_i, \hat{F}_i)$. Then, for $i = 1, 2, \ldots, h$, we have the following:*

(i) *$\hat{\varphi}_i^{-1}(\hat{W}_i - x_i) \subseteq X$ holds for some $X \in A^* \cup B$.*
(ii) *If $\hat{\varphi}_i^{-1}(\hat{W}_i - x_i) \subseteq X \in A^*$, then $\hat{\varphi}_i^{-1}(\hat{W}_i - x_i) = X$.*
(iii) *If $\hat{\varphi}_i^{-1}(\hat{W}_i - x_i) \subseteq X \in B$, then $\hat{\varphi}_i^{-1}(\hat{W}_i - x_i) \subset X$.*

Proof: (i) Because of our assumption that $\lambda(N) = \lambda(\hat{N})$, $\{\hat{W}_i - x_i, \hat{W} - (\hat{W}_i - x_i)\}$ is a min-cut of $\hat{\mathcal{R}}$, and hence $\{Y, \overline{Y}\}$, where $Y = \hat{\varphi}_i^{-1}(\hat{W}_i - x_i)$, is a min-cut in $\mathcal{C}(N)$. This cut does not separate s and t, since $\{s,t\} \subseteq \overline{Y}$. By Lemma 3.6, the cut $\{Y, \overline{Y}\}$ does not cross the (s,t)-MC-partition $\pi_{(s,t)} = (V_1, V_2, \ldots, V_r)$, i.e., it is either compatible or complementary with $\pi_{(s,t)}$. Assume the assertion (i) is not true, and let $Y \cap X_1 \neq \emptyset \neq Y \cap X_2$ for two distinct $X_1, X_2 \in A^* \cup B$ (note that $X_1 \cap X_2 = \emptyset$). Then, since $\{s,t\} \subseteq \overline{Y}$ implies $V_1 \cup V_r \subseteq \overline{Y}$, neither Y nor $\overline{Y}$ is contained in one $V_k \in \{V_1, V_2, \ldots, V_r\}$, i.e., $\{Y, \overline{Y}\}$ cannot be complementary with (s,t)-MC-partition $\pi_{(s,t)}$. Thus, the cut $\{Y, \overline{Y}\}$ must be compatible with the $\pi_{(s,t)}$. This, together with the fact that $\{Y, \overline{Y}\}$ does not separate s and t, implies, by Lemma 6.1, that there is an i such that, for all $V_k \in Y$, $k \in \{a_i, a_{i+1}, \ldots, b_i\}$. This contradicts the definition of X_1 and X_2.

(ii) Assume $X = V_{(a_i,b_i)}$ for some $i = 1, \ldots, p$. Let $(\mathcal{R}, \varphi)$ be the cycle type normal cactus representation for $\mathcal{C}(N)$, and z_s (resp. z_t) be the node in $\mathcal{R} = (W, F)$ such that $s \in \varphi^{-1}(z_s)$ (resp. $t \in \varphi^{-1}(z_t)$). To obtain a cactus representation for $\mathcal{C}(\hat{N})$, consider the set $Y_{\langle s,t \rangle}$ of all nodes $y \in W - \{z_s, z_t\}$ such that z_s and z_t are no longer connected in $\mathcal{R} - y$. Let $Z_{\langle s,t \rangle} = Y_{\langle s,t \rangle} \cup \{z_s, z_t\}$, and let $\mathcal{R}_{\langle s,t \rangle}$ denote the graph obtained from $\mathcal{R}$ by shrinking $Z_{\langle s,t \rangle}$ into a new single node $z_{\langle s,t \rangle}$, deleting any resulting self-loop. Clearly, $\mathcal{R}_{\langle s,t \rangle}$ is a cactus. Define $\varphi_{\langle s,t \rangle}$ to be the mapping such that $\varphi_{\langle s,t \rangle}^{-1}(z) = \varphi^{-1}(z)$ if $z \in W - Z_{\langle s,t \rangle}$ and $\varphi_{\langle s,t \rangle}^{-1}(z_{\langle s,t \rangle}) = \cup_{z \in Z_{\langle s,t \rangle}} \varphi^{-1}(z)$. Then $(\mathcal{R}_{\langle s,t \rangle}, \varphi_{\langle s,t \rangle})$ is a representation for $\mathcal{C}(\hat{N})$, since only those min-cuts in $\mathcal{R}$ that do not separate z_s and z_t remain in $\mathcal{R}_{\langle s,t \rangle}$. Note that the resulting representation $(\mathcal{R}_{\langle s,t \rangle}, \varphi_{\langle s,t \rangle})$ has a $(b_i - a_i + 2)$-cycle C_i which yields $\tilde{\pi}_{A_i}$, where C_i passes through $z_{\langle s,t \rangle}$. $(\mathcal{R}_{\langle s,t \rangle}, \varphi_{\langle s,t \rangle})$ may not be normal, but is of the cycle-type since no empty 3-junction node newly is created. $(\mathcal{R}_{\langle s,t \rangle}, \varphi_{\langle s,t \rangle})$ has an empty 2-junction node, say, z, belonging to a 2-cycle C'_z only when such cycle is a subpath of a longer cycle in $\mathcal{R}$ (i.e., C'_z passes though z and $z_{\langle s,t \rangle}$). For each such node z, merge z into $z_{\langle s,t \rangle}$, deleting the two edges of C'_z. The resulting representation is the cycle-type normal cactus representation for $\mathcal{C}(\hat{N})$, which is unique by Lemma 2.1(ii). We denote it by $(\hat{\mathcal{R}}, \hat{\varphi})$, where $x = z_{s,t}$. If a cycle C_i with $b_i - a_i + 2 \geq 3$ remains in $(\hat{\mathcal{R}}, \hat{\varphi})$, then (ii) holds. Assume that C_i is deleted from $(\mathcal{R}_{\langle s,t\rangle}, \varphi_{\langle s,t\rangle})$ by removing an empty node z that belongs to a 2-cycle $C'_z = C_i$. Let C''_z be another cycle in $(\mathcal{R}_{\langle s,t\rangle}, \varphi_{\langle s,t\rangle})$ that passes through z. Clearly, C''_z remains in $(\hat{\mathcal{R}}, \hat{\varphi})$, passing through $x = z_{s,t}$ instead of z. This implies that this C''_z gives rise to the bolckcactus representation $(\hat{\mathcal{R}}_i, \hat{\varphi}_i)$, proving (ii). (iii) From $\{s,t\} \subseteq \overline{X}$, $X \neq V_1, V_r$. Then, $\hat{\varphi}_i^{-1}(\hat{W}_i - x_i) = X \in B$ implies that $X = V_j$ for some $V_j \in B$. This, however, contradicts that $(V_j, \overline{V_j}) \notin \mathcal{C}(N)$. □

By Lemma 7.1(ii), we may assume without loss of generality that $(\hat{\mathcal{R}}_i, \hat{\varphi}_i)$, $i = 1, 2, \ldots, h$, are renumbered so that

$$\begin{array}{ll} \hat{\varphi}_i^{-1}(\hat{W}_i - x_i) = V_{(a_i,b_i)} & \text{for } i = 1, 2, \ldots, p, \quad \text{and} \\ \exists V_j \in B : \hat{\varphi}_i^{-1}(\hat{W}_i - x_i) \subset V_j & \text{for } i = p+1, p+2, \ldots, h \end{array}$$

hold. Since $\hat{\varphi}_i^{-1}(\hat{W}_i - x_i)$, for $i = p+1, p+2, \ldots, h$, is properly contained in a single element V_j by Lemma 7.1(iii), any representation in

$$R_B = \{(\hat{\mathcal{R}}_i, \hat{\varphi}_i) \mid i = p+1, p+2, \ldots, h\}$$

is complementary with the (s,t)-cactus representation $(\mathcal{R}_{(s,t)}, \varphi_{(s,t)})$ and represents min-cuts in $\mathcal{C}(N) - \mathcal{C}_{comp}(\pi_{(s,t)})$. In Fig. 8, we have five blockcacti $\hat{\mathcal{R}}_1, \ldots, \hat{\mathcal{R}}_5$ at x, where $(\hat{\mathcal{R}}_4, \hat{\varphi}_4), (\hat{\mathcal{R}}_5, \hat{\varphi}_5) \in R_B$.

We now proceed to the second decomposition stage for those blockcacti $\hat{\mathcal{R}}_i$, $i = 1, \ldots, p$. For each $i = 1, 2, \ldots, p$, let C_i be the cycle containing x_i in $\hat{\mathcal{R}}_i$. Then, as discussed in Section 7.1, C_i yields a maximal circular MC-partition,

$$\hat{\pi}_{A_i} = (V_{(1,a_i-1)} \cup V_{(b_i+1,r)} - t, V_{a_i}, V_{a_i+1}, \ldots, V_{b_i}),$$

over $\mathcal{C}(\hat{N})$. We decompose the blockcactus $\hat{\mathcal{R}}_i$ into a collection of subcacti in $\hat{\mathcal{R}}_i - F(C_i)$ where $F(C_i)$ denotes the set of edges on C_i (see Section 3.2). Let us denote the subcacti in $\hat{\mathcal{R}}_i$ by $\hat{\mathcal{R}}_{i,j} = (\hat{W}_{i,j}, \hat{F}_{i,j})$, $j = 0, 1, \ldots, b_i - a_i + 1$, and $y_{i,j}$ be the node in $W_{i,j} \cap W(C_i)$, where $W(C_i)$ denotes the set of nodes on C_i. Here, we denote the trivial cactus consisting of x_i by $\hat{\mathcal{R}}_{i,0}$. If some subcactus $\hat{\mathcal{R}}_{i,j}$ with $j \neq 0$ is trivial, it represents only one min-cut in $(\hat{\mathcal{R}}, \hat{\varphi})$, which is in $\mathcal{C}_{comp}(\pi_{(s,t)})$ and hence is already represented in the (s,t)-cactus representation $(\mathcal{R}_{(s,t)}, \varphi_{(s,t)})$. We therefore consider only nontrivial subcacti, and, for each such subcactus $\hat{\mathcal{R}}_{i,j}$, define its mapping $\hat{\varphi}_{i,j} : V \to \hat{W}_{i,j}$ by

$$\text{(7.3)} \qquad \hat{\varphi}_{i,j}^{-1}(z) := \begin{cases} \hat{\varphi}^{-1}(y) & \text{if } z \in \hat{W}_{i,j} - y_{i,j} \\ V - \hat{\varphi}^{-1}(\hat{W}_{i,j} - y_{i,j}) & \text{if } z = y_{i,j}. \end{cases}$$

Then, any cactus representation in

$$R_A = \{(\hat{\mathcal{R}}_{i,j}, \hat{\varphi}_{i,j}) \mid 1 \leq j \leq b_i - a_i + 1,\ i = 1, 2, \ldots, p,\ \hat{\mathcal{R}}_{i,j} \text{ is nontrivial}\}$$

is complementary with the (s,t)-cactus representation $(\mathcal{R}_{(s,t)}, \varphi_{(s,t)})$. In our running example of Fig. 8, we have three trivial subcacti $(\hat{\mathcal{R}}_{1,2}, \hat{\varphi}_{1,2})$, $(\hat{\mathcal{R}}_{2,1}, \hat{\varphi}_{2,1})$, $(\hat{\mathcal{R}}_{3,1}, \hat{\varphi}_{3,1})$, and $R_A = \{(\hat{\mathcal{R}}_{1,1}, \hat{\varphi}_{1,1}), (\hat{\mathcal{R}}_{1,3}, \hat{\varphi}_{1,3}), (\hat{\mathcal{R}}_{3,2}, \hat{\varphi}_{3,2})\}$.

7.3. Grafting Procedure

Making use of the sets R_B and R_A defined in Section 7.2, we can compute a cactus representation $(\mathcal{R}^*, \varphi^*)$ for $\mathcal{C}(N)$ in N as follows:

```
$(\mathcal{R}^*, \varphi^*) := (\mathcal{R}_{(s,t)}, \varphi_{(s,t)})$;
$R := R_B \cup R_A$;
while $R \neq \emptyset$ do
begin
    pick an element $(\mathcal{R}', \varphi')$ from $R$;
    $R := R - (\mathcal{R}', \varphi')$;
    $(\mathcal{R}^*, \varphi^*) := (\mathcal{R}^*, \varphi^*) \oplus (\mathcal{R}', \varphi')$.
end.
```

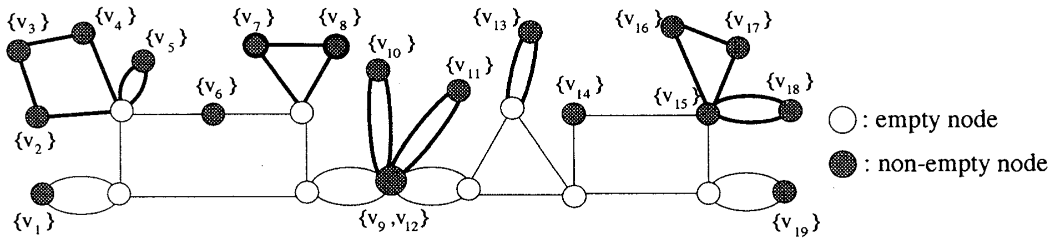


Figure 9: A cactus representation for $\mathcal{C}(N_1)$.

It is easy to see that an intermediate $(\mathcal{R}^*, \varphi^*)$ constructed during the above procedure is complementary with any $(\mathcal{R}', \varphi')$ remaining in R. Therefore, the successive union operations can be carried out and each $(\mathcal{R}^*, \varphi^*)$ is a cactus representation for some $\mathcal{C}' \subseteq \mathcal{C}(N)$. Clearly, the final $(\mathcal{R}^*, \varphi^*)$ represents the set of all min-cuts in $\mathcal{C}(N)$. Note that the cactus $\mathcal{R}^*$ contains the nodes in $\{\hat{W}_i - x_i \mid (\hat{\mathcal{R}}_i, \hat{\varphi}_i) \in R_B\} \cup \{\hat{W}_{i,j} - y_{i,j} \mid (\hat{\mathcal{R}}_{i,j}, \hat{\varphi}_{i,j}) \in R_A\}$, and the rest of the nodes in $\mathcal{R}^*$ correspond to the nodes in $\mathcal{R}_{(s,t)}$. For example, Fig. 9 shows the cactus representation obtained by applying the above procedure to the representations in Figs. 7 and 8.

Lemma 7.2 *The cycle-type normal cactus representation for $\mathcal{C}(N)$ in N can be obtained from the (s,t)-cactus representation $(\mathcal{R}_{(s,t)}, \varphi_{(s,t)})$ and the cycle-type normal cactus representation $(\hat{\mathcal{R}}, \hat{\varphi})$ for $\mathcal{C}(\hat{N})$ in $O(|V|)$ time and $O(|V|)$ space.*

Proof: A cactus representation $(\mathcal{R}^*, \varphi^*)$ obtained above represents all min-cuts in $\mathcal{C}(N)$, each of which is represented in a cactus representation $(\mathcal{R}', \varphi') \in R_B \cup R_A$ or the (s,t)-cactus representation $(\mathcal{R}_{(s,t)}, \varphi_{(s,t)})$. That is, all min-cuts represented by $(\hat{\mathcal{R}}, \hat{\varphi})$, except those represented by the circular MC-partitions in (7.1), are represented by $(\mathcal{R}^*, \varphi^*)$. Since all min-cuts represented by circular MC-partitions in (7.1) are already represented in $(\mathcal{R}_{(s,t)}, \varphi_{(s,t)})$ by Lemma 6.1(ii), the resulting $(\mathcal{R}^*, \varphi^*)$ is a cactus representation for $\mathcal{C}(N)$.

We now analyze the time and space complexities. Note that $\hat{\mathcal{R}}$ has $O(|V|)$ nodes and edges by Lemma 2.1(iii) since $(\hat{\mathcal{R}}, \hat{\varphi})$ is a cycle-type normal cactus representation. Clearly, $R_B \cup R_A$ can be prepared in $O(|V|)$ time and $O(|V|)$ space (note that we do not have to update mappings in (7.2) and (7.3) for the purpose of constructing $(\mathcal{R}^*, \varphi^*)$). Each time a union $(\mathcal{R}^*, \varphi^*) \oplus (\mathcal{R}', \varphi')$ is formed, a joint node z^* in $\mathcal{R}^*$ can be found in $O(1)$ time by using

$\hat{\varphi}^{-1}$ and $\varphi_{(s,t)}$ (z^* is determined by $\varphi_{(s,t)}(v)$ for any $v \in \hat{\varphi}^{-1}(z')$ after picking a 1-junction node z' of $\hat{\mathcal{R}}$ in $\mathcal{R}'$), and a cactus of $(\mathcal{R}^*, \varphi^*) \oplus (\mathcal{R}', \varphi')$ can be computed in $O(|F'|)$ time, where F' is the edge set of $\mathcal{R}'$. This ensures that the total time for all union operations is at most the size of $\hat{\mathcal{R}}$, i.e., $O(|V|)$. After the final cactus $\mathcal{R}^*$ is obtained, its mapping φ^* can be easily computed by using $\hat{\varphi}^{-1}$ and $\varphi_{(s,t)}$ as follows. Any node y except x in $\hat{\mathcal{R}}$ also appears in $\mathcal{R}^*$, and hence $\varphi^{*-1}(y) = \hat{\varphi}^{-1}(y)$ for such y. For each vertex v in $\hat{\varphi}^{-1}(x)$, we see that $\varphi^*(v) = \varphi_{(s,t)}(v)$. From these, φ^* can be determined in $O(|V|)$ time

The other operations including the simplification of $\mathcal{R}^*$ to make $\mathcal{R}^*$ cycle-type normal can be carried out in $O(|V|)$ time and $O(|V|)$ space. (See the proof of Lemma 3.3 in [16]). □

7.4. Complete Algorithm

Putting everything together according to the approach stated in Section 4, we now present our algorithm for constructing a cactus representation for $\mathcal{C}(N)$.

```
Algorithm NEWCACTUS {Input: a network $N = (V, E, c)$.
  Output: a cactus representation $(\mathcal{R}, \varphi)$ for $\mathcal{C}(N)$.}
begin
  Compute $\lambda(N)$; $N^{(|V|)} := N$;
  for $i = |V|, |V|-1, \ldots, 2$ do
    begin
      Choose an edge $(s^{(i)}, t^{(i)})$ in $N^{(i)}$;
      $N^{(i-1)} := N^{(i)}_{t^{(i)}=s^{(i)}}$
    end
  $\mathcal{R}^{(1)} := (x, \emptyset)$;  $\varphi^{(1)}(V) := x$;
  for $i = 2, 3, \ldots, |V|$ do
    begin
      Compute an $(s^{(i)}, t^{(i)})$-cactus representation $(\mathcal{R}_{(s^{(i)},t^{(i)})}, \varphi_{(s^{(i)},t^{(i)})})$ over $\mathcal{C}(N^{(i)})$;
      Compute a cycle-type normal cactus representation $(\mathcal{R}^{(i)}, \varphi^{(i)})$ for $\mathcal{C}(N^{(i)})$
        from $(\mathcal{R}_{(s^{(i)},t^{(i)})}, \varphi_{(s^{(i)},t^{(i)})})$ and $(\mathcal{R}^{(i-1)}, \varphi^{(i-1)})$
    end
  $(\mathcal{R}, \varphi) := (\mathcal{R}^{(|V|)}, \varphi^{(|V|)})$
end.
```

Theorem 7.3 *Algorithm NEWCACTUS computes the cycle-type normal cactus representation for $\mathcal{C}(N)$ in a network $N = (V, E, c)$ correctly and runs in $O(|V| \cdot M(|V|, |E|))$ time, where $M(|V|, |E|)$ is the time for computing a maximum flow between two vertices in $N = (V, E, c)$.*

Proof: The validity of NEWCACTUS follows immediately from Lemma 7.2. It is known that $\lambda(N)$ can be determined in $O(|V| \cdot M(|V|, |E|))$ time [9]. At each stage i, the (s,t)-cactus representation in $N^{(i)}$ can be found in $O(|E|+|V|)$ time, after computing a maximum flow between arbitrarily chosen s and t in $M(|V|, |E|)$ time. Then, NEWCACTUS runs in $O(|V| \cdot M(|V|, |E|))$ time. □

The above theorem implies the already known fact [3, 8] that, for any network N, $\mathcal{C}(N)$ has a cactus representation. We finally discuss a special case of networks, where all edges have unit weights.

Corollary 7.1 *A cactus representation for the set $\mathcal{C}(G)$ of all min-cuts in a multigraph $G = (V, E)$ can be constructed in $O(|E|+\lambda(G)|V|^2)$ time and $O(|V|+|E|)$ space.*

Proof: The edge-connectivity $\lambda(G)$ of G can be determined in $O(|E|+\lambda(G)|V|^2)$ time and $O(|V|+|E|)$ space [14]. We first find a subset $E' \subseteq E$ with $|E'| < (\lambda(G)+1)|V|$ such that the graph $G' = (V, E')$ satisfies $\lambda(G') = \lambda(G)$ and has exactly the same set of min-cuts as G. It is known that such E' can be found in $O(|V|+|E|)$ time and $O(|V|+|E|)$ space [6, 15]. By applying NEWCACTUS to this sparse graph G', a cactus representation for $\mathcal{C}(G)$ can be computed in $O(|E|+|V| \cdot |E'|) \le O(|E|+\lambda(G)|V|^2)$ time and $O(|V|+|E|)$ space because a maximum flow between a pair of vertices can be found in linear time [13]. This completes the proof. □

The currently best time bound for constructing a cactus representation in a multigraph $G = (V, E)$ is $O(|E|+\lambda(G)^2|V|\log|V|)$ [8]. When G has so many edges that $\lambda(G) \ge |V|/\log|V|$ holds, the time bound in the above corollary is slightly better than this bound.

Acknowledgements

The authors would like to thank the anonymous referees for their helpful comments, which have substantially improved the presentation. This work was supported in part by grants from the Natural Sciences and Engineering Research Council of Canada, the Advanced Systems Institute of British Columbia, and in part by Grant-in-Aid from the Ministry of Education, Science and Culture of Japan.

Appendix 1. Proof of Lemma 3.6

Let $\pi_{(s,t)} = (V_1, V_2, \ldots, V_r)$ be the (s,t)-MC-partition over $\mathcal{C}(N)$. Assume that a cut $\{X, \overline{X}\} \in \mathcal{C}(N)$ crosses $\pi_{(s,t)}$. By Lemma 3.4, any min-cut separating s and t never crosses $\pi_{(s,t)}$, and hence, we can assume $X \cap \{s,t\} = \emptyset$. Then there are two subsets V_k and V_l in $\pi_{(s,t)}$ such that $X \cap V_k \neq \emptyset \neq \overline{X} \cap V_k$ and $X \cap V_l \neq \emptyset$. Without loss of generality, assume $k < l$ (the case $k > l$ can be treated analogously), as shown in Fig. 10. Since the min-cut $\{V_{(1,k)}, V_{(k+1,r)}\}$ crosses $\{X, \overline{X}\}$, by Lemma 3.1(i), we have

$$c(V_{(1,k)} \cap X, V_{(k+1,r)} \cap X) = c(V_{(1,k)} \cap X, V_{(1,k)} - X) \quad (= \lambda(N)/2).$$

Clearly $E(V_{(1,k)} - X, \overline{V_{(1,k)} - X}) - E(V_{(1,k)} - X, V_{(1,k)} \cap X) = E(V_{(1,k)}, V_{(k+1,r)}) - E(V_{(1,k)} \cap X, V_{(k+1,r)} \cap X)$ holds (see Fig. 10).

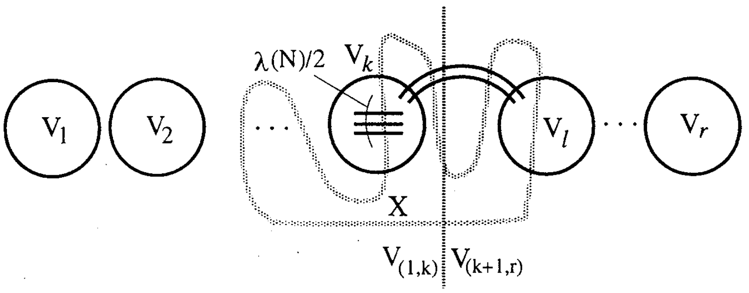


Figure 10: Cut $\{X, V-X\}$ crossing the (s,t)-MC partition $\pi_{(s,t)}$.

Therefore, from these equalities, we have

$$\begin{aligned}&c(V_{(1,k)}-X, \overline{V_{(1,k)}-X})\\ &= c(V_{(1,k)}, V_{(k+1,r)})-c(V_{(1,k)}\cap X, V_{(k+1,r)}\cap X)+c(V_{(1,k)}-X, V_{(1,k)}\cap X)\\ &= c(V_{(1,k)}, V_{(k+1,r)}) \quad (=\lambda(N)),\end{aligned}$$

implying that $\{V_{(1,k)}-X, \overline{V_{(1,k)}-X}\}$ is a min-cut separating s and t. This contradicts Lemma 3.4. □

Appendix 2. Proof of Lemma 3.7(ii)

Let $\{X, \overline{X}\} \in \mathcal{C}(N)$ be a min-cut compatible with $\pi_{(s,t)}$ which does not separate s and t, i.e., $\{s,t\} \subseteq \overline{X}$ holds without loss of generality. Then $V_1 \cup V_r \subseteq \overline{X}$ holds since $\{X, \overline{X}\}$ is compatible with $\pi_{(s,t)}$. We first show that X must have the form $V_{(i,j)}$ for some i, j $(1 < i \leq j < r)$. Assume that there are i, k with $2 \leq i < k \leq r-1$ such that

$$V_{i-1} \subseteq X, \quad V_i \subseteq \overline{X}, \text{ and } V_k \subseteq X \quad \text{(see Fig. 11).}$$

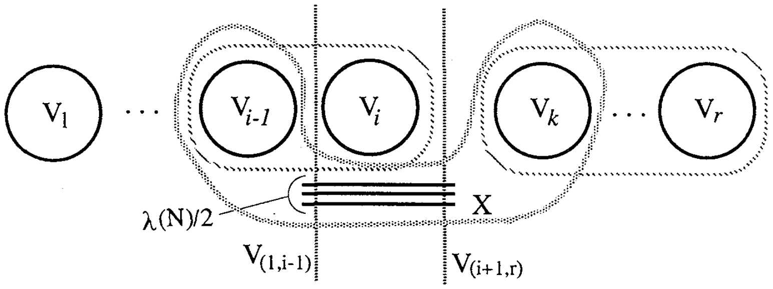


Figure 11: Cut $\{X, V-X\}$ compatible with the (s,t)-MC partition $\pi_{(s,t)}$.

Since the min-cut $\{V_{(1,i-1)}, V_{(i,r)}\} \in \mathcal{C}(N)$ crosses the cut $\{X, \overline{X}\}$, we have, by Lemma 3.1(i), $c(V_{(1,i-1)} \cap X, V_{(i,r)} \cap X) = \lambda(N)/2$, or equivalently,

$$c(V_{(1,i-1)} \cap X, V_{(i+1,r)} \cap X) = \lambda(N)/2,$$

since $V_i \cap X = \emptyset$ (see Fig. 11). Edge (s,t), which has a positive weight, belongs to $E(V_{(1,i-1)}, V_{(i+1,r)}) - E(V_{(1,i-1)} \cap X, V_{(i+1,r)} \cap X)$. Therefore, we have

$$c(V_{(1,i-1)}, V_{(i+1,r)}) \geq c(V_{(1,i-1)} \cap X, V_{(i+1,r)} \cap X) + c(s,t) > \lambda(N)/2.$$

From this and $c(V_{(1,i-1)}, V_{(i,r)}) = \lambda(N)$ (resp. $c(V_{(1,i)}, V_{(i+1,r)}) = \lambda(N)$), we see that

$$c(V_{(1,i-1)}, V_i) < \lambda(N)/2 \quad \text{(resp. } c(V_i, V_{(i+1,r)}) < \lambda(N)/2)$$

holds. This means $c(V_i, \overline{V_i}) < \lambda(N)$, contradicting the definition of $\lambda(N)$.

Next, we first show that $\{V_k, \overline{V_k}\}$ is a min-cut for each k such that $i \leq k \leq j$ with $X = V_{(i,j)}$. Obviously, $i = j$ implies that $\{X, \overline{X}\} = \{V_k, \overline{V_k}\}$ is a min-cut for $i = k = j$. Since $\{V_{(1,i)}, V_{(i+1,r)}\}$ and $\{V_{(i,j)}, \overline{V_{i,j}}\}$ are min-cuts crossing each other, $\{V_i, \overline{V_i}\}$ and $\{V_{(i+1,j)}, \overline{V_{(i+1,j)}}\}$ are also min-cuts by Lemma 3.1. Analogously, by repeating a similar argument for $X = V_{(i+1,j)}, X = V_{(i+2,j)}, \ldots$, the rest of $\{V_k, \overline{V_k}\}$ $(i+1 \leq k \leq j)$ can be shown to be a min-cut. □

Appendix 3. Proof of Lemma 6.1(ii),(iii)

We first show that (ii-a) the o-partition π_{A_i} of V defined in (ii) is a circular MC-partition in N. Then, after showing (iii), we prove that (ii-b) π_{A_i} is a maximal circular MC-partition over $\mathcal{C}(N)$.

(ii-a) It is sufficient to show that, for each $A_i = \{V_{a_i}, V_{a_i+1}, \ldots, V_{b_i}\}$, $i = 1, 2, \ldots, p$, any cut $\{V_{(k,l)}, \overline{V_{(k,l)}}\}$ with $a_i \le k \le l \le b_i$, is a min-cut, because the other cuts belonging to π_{A_i} are min-cuts that separate s and t by the definition of circular MC-partitions. In the case where either $k = a_i$ or $k = l$, by the definition of A_i, $\{V_{(k,l)}, \overline{V_{(k,l)}}\} \in \mathcal{C}(N)$. In the remaining case where $a_i < k < l$, the min-cut $\{V_{(1,k-1)}, \overline{V_{(1,k-1)}}\}$ separating s and t crosses the cut $\{V_{(a_i,l)}, \overline{V_{(a_i,l)}}\}$, which is a min-cut by the definition of A_i. By Lemma 3.1, $\{V_{(k,l)} = V_{(1,l)} - V_{(1,k-1)}, \overline{V_{(k,l)}}\}$ is also a min-cut in $\mathcal{C}(N)$.

(iii) Assume that there is a min-cut $\{X, \overline{X}\} \in \mathcal{C}_{comp}(\pi_{(s,t)})$ that *does not* belong to any of MC-partitions, $\pi_{B_0}, \ldots, \pi_{B_q}$, or circular MC-partitions, $\pi_{A_1}, \ldots, \pi_{A_p}$. Clearly, any min-cut separating s and t belongs to at least one of those MC-partitions or circular MC-partitions. Then, the cut $\{X, \overline{X}\}$ does not separate s and t, and hence has the form $X = V_{(k,l)}$ $(1 < k \le l < r)$ by Lemma 3.7. In analogy with the proof of Lemma 3.7(ii), we see that each $\{V_i, \overline{V_i}\}$ with $k \le i \le l$ is a min-cut in $\mathcal{C}(N)$, from which we have $\{V_k, V_{k+1}, \ldots, V_l\} \subseteq A$.

From the above discussion, there is an A_i with $a_i \le k \le b_i$, where $b_i < l$, because, otherwise, $\{V_{(k,l)}, \overline{V_{(k,l)}}\}$ would belong to π_{A_i}. Here we show that $\{V_{(k,b_i+1)}, \overline{V_{(k,b_i+1)}}\}$ is a min-cut in $\mathcal{C}(N)$. This is clear if $b_i + 1 = l$. If $b_i + 2 \le l$, then min-cut $\{V_{(1,b_i+1)}, \overline{V_{(1,b_i+1)}}\}$ crosses min-cut $\{V_{(k,l)}, \overline{V_{(k,l)}}\}$, from which we see that $\{V_{(k,b_i+1)}, \overline{V_{(k,b_i+1)}}\}$ is a min-cut by Lemma 3.2. This means $a_i < k$ from the maximality of A_i. In this case, however, the min-cut $\{V_{(a_i,b_i)}, \overline{V_{(a_i,b_i)}}\}$ crosses this min-cut $\{V_{(k,b_i+1)}, \overline{V_{(k,b_i+1)}}\}$, and, again by Lemma 3.2, we have a min-cut $\{V_{(a_i,b_i+1)}, \overline{V_{(a_i,b_i+1)}}\}$. This contradicts the maximality of A_i.

(ii-b) Assume that a circular MC-partition π_{A_i} over $\mathcal{C}(N)$ for an A_i $(1 \le i \le p)$ is not maximal. Since no min-cut crosses π_{A_i} by Lemma 3.6, any o-partition of V of the form $(V_{(1,a_i-1)}, V_{a_i}, \ldots, V_j - X, X, \ldots, V_{b_i}, V_{(b_i+1,r)})$ $(a_i \le j \le b_i)$ cannot be a circular MC-partition over $\mathcal{C}(N)$. Assume that $\mathcal{C}(N)$ has a circular MC-partition of V of the form

$$\pi_X = (V_{(1,a_i-1)} - X, X, V_{a_i}, \ldots, V_{b_i}, V_{(b_i+1,r)})$$

(the case where $X \subset V_{(b_i+1,r)}$ can be treated analogously). It follows from this that $\{V_{(1,a_i-1)} - X, \overline{V_{(1,a_i-1)} - X}\}$ is a min-cut separating s and t. By Lemma 3.7(i), $V_{(1,a_i-1)} - X$ and X must be of the form $V_{(1,k)}$ and $V_{(k+1,a_i-1)}$, respectively. Thus, $\{V_{(k+1,b_i)}, \overline{V_{(k+1,b_i)}}\}$, where $V_{(k+1,b_i)} = X \cup V_{a_i} \cup \cdots \cup V_{b_i}$, is a min-cut which belongs to π_X, and is compatible with the (s,t)-MC-partition $\pi_{(s,t)}$. From (iii) of this lemma, such a min-cut $\{V_{(k+1,b_i)}, \overline{V_{(k+1,b_i)}}\}$ must belong to the MC-partition in (i) or (ii) of this lemma. However, there is no such MC-partition, a contradiction. □

References

[1] M.O. Ball and J.S. Provan: "Calculating bounds on reachability in computer networks," *Networks* 13 (1983) pp.253–278.

[2] R.E. Bixby: "The minimum number of edges and vertices in a graph with edge connectivity n and m n-bonds," *Networks* 5 (1975) pp.253–298.

[3] E.A. Dinits, A.V. Karzanov and M.V. Lomonosov: "On the structure of a family of minimal weighted cuts in a graph," *Studies in Discrete Optimization* (in Russian), A.A. Fridman (Ed.), Nauka, Moscow, (1976) pp.290–306.

[4] S. Even and R.E. Tarjan: "Network flow and testing graph connectivity," *SIAM J. Computing* 4 (1975) pp.507–518.

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

[6] A. Frank, T. Ibaraki and H. Nagamochi: "On sparse subgraphs preserving connectivity properties," *J. Graph Theory* 17 (1993) 275–281.

[7] H.N. Gabow: "A matroid approach to finding edge connectivity and packing arborescences," *Proc. 23rd ACM Symp. Theory of Computing*, (1991) pp.112–122.

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

[9] R.E. Gomory and T.C. Hu: "Multi-terminal network network flows," *J. SIAM* 9 (1961) pp.551–570.

[10] J. Hao and J.B. Orlin: "A faster algorithm for finding the minimum cut in a graph," *Proc. 3rd ACM-SIAM Symp. on Discrete Algorithms* (1992) pp.165–174.

[11] A.V. Karzanov and E.A. Timofeev: "Efficient algorithm for finding all minimal edge cuts of a nonoriented graph," *Kibernetika* 2 (1986) pp.8-12; translated in Cybernetics (1986) pp.156–162.

[12] D.W. Matula: "Determining edge connectivity in $O(nm)$," *Proc. 28th IEEE Symp. Found. Comp. Sci.* (1987) pp.249–251.

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

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

[15] H. Nagamochi, Z. Sun and T. Ibaraki: "Counting the number of minimum cuts in undirected multigraphs," *IEEE Trans. on Reliability* 40 (1991) pp.610–614.

[16] H. Nagamochi and T. Kameda: "Canonical cactus representation for minimum cuts," *Japan J. of Industrial and Applied Mathematics* 11 (1994) pp.343–361.

[17] D. Naor, D. Gusfield and C. Martel: "A fast algorithm for optimally increasing the edge connectivity," *Proc. 31st Annual IEEE Symp. Found. Comp. Sci.* (1990) pp.698–707.

[18] D. Naor and V.V. Vazirani: "Representing and enumerating edge connectivity cuts in *RNC*," *Proc. 2nd Workshop on Algorithms and Data Structures* (F.Dehne, J.-R. Sack and N. Santoro, eds.), Lecture Notes in Computer Science 519, Springer Verlag, (1991) pp.273–285.

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

Hiroshi Nagamochi
Department of Applied Mathematics and Physics
Graduate School of Engineering
Kyoto University
Kyoto 606-01, Japan
Netaddress: naga@kuamp.kyoto-u.ac.jp

Tiko Kameda
School of Computing Science
Simon Fraser University
Burnaby, B.C., Canada V5A 1S6
Netaddress: tiko@cs.sfu.ca