Journal of the Operations Research Society of Japan
Vol. 57, No. 2, June 2014, pp. 75–86

A GENERALIZED FRAMEWORK FOR LISTING CUTS AND GRAPHS

Makoto Yaguchi
University of Tsukuba

(Received May 20, 2012; Revised May 21, 2014)

Abstract For the problem of enumerating cuts of graphs, Provan and Shier have provided a framework of enumeration that can be applied for the enumeration of various types of cuts such as minimal (s,t)-cuts in an undirected graph, minimal (s,t)-cuts in a directed graph, minimal (s,K)-cuts in an undirected graph, etc. In this paper we generalize their framework and provide an enumerating method that works in a looser condition, providing a possibility of its application easier. We demonstrate its use by giving an algorithm for the enumeration problem that contains the problem of listing minimal (s,K)-cuts in an undirected graph.

Keywords: Graph theory, (s,K)-cut, enumeration

1. Introduction

Let $G=(V,E)$ be a graph (undirected or directed) with vertex set V and edge set E. For $X,Y\subseteq V$, let $E(X,Y)=\{(u,v)\in E \mid u\in X, v\in Y\}$. For $X\subseteq V$, the induced subgraph $G[X]$ is the graph with vertex set X and edge set $E(X)=E(X,X)$. For a vertex v of G, let $G-v$ be the graph obtained by the deletion of v from G. For $E'\subseteq E$, define the graph $G-E'$ as the graph obtained from G by deleting all edges of E'. The open neighborhood $\Gamma(X)$ of $X\subseteq V$ is defined by $\Gamma(X)=\{v\in V\setminus X \mid \exists u\in X;(u,v)\in E\}$.

A set of edges $E'\subseteq E$ is an (s,t)-cut if there are no paths from the vertex s to the vertex t in $G-E'$, and it is an (s,K)-cut for a given vertex s and a set $K\subseteq V\setminus\{s\}$ if there are no paths from s to t in $G-E'$ for some $t\in K$. An edge set separating s from every element of K is called a strong (s,K)-cut. We discuss in this paper the problem of listing minimal cuts of a given graph. This problem, for example, has an application in network reliability, see Colbourn [4].

Abel and Bicker [1], Bellmore and Jensen [3] and Tsukiyama et al. [6] proposed algorithms of listing all minimal (s,t)-cuts of an undirected graph. The most efficient algorithm is that proposed by Tsukiyama et al. [6], which requires $O(|E|)$ time per one cut listed. After these individual results, Provan and Shier [5] gave a general framework of enumeration that can be applied for the enumeration of various types of cuts. It provides algorithms for listing minimal (s,t)-cuts in an undirected graph, minimal (s,K)-cuts in an undirected graph, and minimal (s,t)-cuts in a directed graph, respectively, in $O(|E|)$ time per one cut listed. But it is difficult to list minimal (s,K)-cuts of an arbitrary directed graph by the framework.

In this paper, we propose a generalized framework and discuss its application.

2. The Framework of Provan and Shier

First, let us review the enumeration framework of Provan and Shier [5].

Let $\mathcal{I}$ be a collection of subsets of the vertex set V of $G=(V,E)$. A vertex $v\in V\setminus S$ is

a *pivot element* for the set $S \in \mathcal{I}$ if there exists $Y \in \mathcal{I}$ such that $S \cup \{v\} \subseteq Y$ and $Y \subseteq Z$ for any $Z \in \mathcal{I}$ with $S \cup \{v\} \subseteq Z$. Namely, a vertex v is a pivot element if there exists a unique minimal element $Y \in \mathcal{I}$ which contains $S \cup \{v\}$. Here, the set $I(S, v) = Y \setminus S$ is called the *pivot set* for S and v. For example, consider $V = \{1, 2, 3, 4, 5, 6, 7, 8\}$ and the collection $\mathcal{I}$ which consists of $\{1, 2\}$, $\{1, 2, 3\}$, $\{1, 2, 3, 4, 8\}$, $\{1, 2, 3, 5, 6\}$, $\{1, 2, 3, 5, 7, 8\}$, and $\{2, 3\}$. Let $S = \{2, 3\}$. Then $\{1, 2, 3, 5, 7, 8\}$ is a minimal element of $\mathcal{I}$ which contains $S \cup \{7\}$. No other element of $\mathcal{I}$ is a minimal element of $\mathcal{I}$ which contains $S \cup \{7\}$. So $v = 7$ is a pivot element for S and $I(S, v) = \{1, 5, 7, 8\}$.

Define the following property (P2) for a collection $\mathcal{I}$.

(P2) For any $X, S \in \mathcal{I}$ with $X \supsetneq S$ there is a pivot element v for S with $v \in X$.

For disjoint $S, T \subseteq V$, define $\mathcal{I}(S, T) = \{X \in \mathcal{I} \mid S \subseteq X \subseteq V \setminus T\}$. The property (P2) gives the following lemma, where $\sqcup$ denotes the disjoint union.

Lemma 2.1 (Provan and Shier [5])**.** *Suppose the collection $\mathcal{I}$ satisfies property (P2). Let $S \in \mathcal{I}$ and $T \subseteq V \setminus S$ such that $|\mathcal{I}(S, T)| > 1$. Then there is a pivot element v for S with $I(S, v) \subseteq V \setminus T$. Moreover, $\mathcal{I}(S, T \cup \{v\})$ and $\mathcal{I}(S \cup I(S, v), T)$ form a nontrivial partition of $\mathcal{I}(S, T)$ i.e.,*

$$\mathcal{I}(S, T \cup \{v\}) \neq \emptyset \quad \textit{and} \quad \mathcal{I}(S \cup I(S, v), T) \neq \emptyset,$$
$$\mathcal{I}(S, T) = \mathcal{I}(S, T \cup \{v\}) \sqcup \mathcal{I}(S \cup I(S, v), T).$$

Lemma 2.1 gives the following recursive procedure LIST(S, T) that lists all elements of $\mathcal{I}(S, T)$, where $S \in \mathcal{I}$ and $T \subseteq V \setminus S$. PIVOT$(S, T; v, I(S, v))$ is a subroutine that takes two sets S and T as input, and outputs a pivot element v and the associated pivot set $I(S, v)$ such that $I(S, v)$ is a subset of $V \setminus T$ if possible; such a pivot element will exist if $|\mathcal{I}(S, T)| > 1$ by Lemma 2.1. Otherwise PIVOT returns $I(S, v) = \emptyset$. LIST(S, T) runs in time $O(\tau(\text{PIVOT}))$ per one element, where $\tau(\text{PIVOT})$ is the time complexity of the procedure PIVOT.

Procedure LIST(S, T)

```
PIVOT(S, T; v, I(S, v)) ;
if I(S, v) = ∅ then
    output S ;
else
    LIST(S, T ∪ {v}) ;
    LIST(S ∪ I(S, v), T) ;
```

Especially, if $\mathcal{I}$ contains the empty set, LIST$(\emptyset, \emptyset)$ lists all the members of $\mathcal{I}$. Let $\mathcal{C}$ be a family of cuts of G to be listed. Assume there exists a collection $\mathcal{I} \subseteq 2^V$ associated to $\mathcal{C}$ satisfying the following properties:

(P1) There is a one-to-one correspondence between cuts $C \in \mathcal{C}$ and sets $X \in \mathcal{I}$, defined by $X \in \mathcal{I} \Leftrightarrow C = E(X, V \setminus X) \in \mathcal{C}$.

(P2)′ For any $X, S \in \mathcal{I} \cup \{\emptyset\}$ with $X \supsetneq S$ there is a pivot element v for S with $v \in X$.

Property (P1) ensures that the elements of $\mathcal{C}$ are generated without duplication by listing the elements of $\mathcal{I}$. So any procedure to generate the elements of $\mathcal{I}$ will generate the elements of $\mathcal{C}$ with the same computational complexity. Property (P2)′ enables us to list all the members of $\mathcal{I}$ by the recursive procedure LIST$(\emptyset, \emptyset)$.

In order to apply the framework to a given collection of cuts $\mathcal{C}$, the following steps are required:

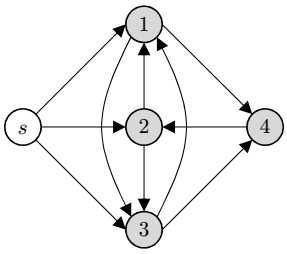


Figure 1: The graph that $\mathcal{I}$ does not satisfy (P2)′.

1. Give a collection $\mathcal{I} \subseteq 2^V$ satisfying property (P1).
2. Identify pivot elements and show that $\mathcal{I}$ satisfies property (P2)′.
3. Produce a corresponding PIVOT routine and analyze its complexity.

For listing minimal (s,t)-cuts in an undirected graph, Provan and Shier [5] showed that the collection $\mathcal{I}$ of $X \subseteq V$ satisfying (2.1) satisfies (P1).

$$\begin{aligned} &s \in X, \quad t \notin X, \\ &G[X] \text{ is connected}, \\ &G[V \setminus X] \text{ is connected}. \end{aligned} \tag{2.1}$$

And $\mathcal{I}$ satisfies (P2)′ because for any $X, S \in \mathcal{I} \cup \{\emptyset\}$ with $X \supsetneq S$, if $S = \emptyset$ then s is a pivot element for S with $s \in X$, and if $S \neq \emptyset$ then any vertex $v \neq t$ with $v \in \Gamma(S) \cap X$ is a pivot element for S with $v \in X$ as Provan and Shier [5] showed. Provan and Shier [5] provided an implementation of PIVOT that takes $O(|E|)$. Also Provan and Shier [5] provided algorithms of listing minimal (s,t)-cuts in a directed graph and minimal (s,K)-cuts in an undirected graph, that run in time $O(|E|)$ per cut, respectively, by giving a suitable implementations of PIVOT for each problem.

For listing minimal (s,K)-cuts in a directed graph, Provan and Shier [5] showed that the collection $\mathcal{I}$ of $X \subseteq V$ satisfying (2.2) satisfies (P1).

$$\begin{aligned} &s \in X, \quad K \setminus X \neq \emptyset, \\ &\text{there is a path from } s \text{ to } x \text{ in } G[X] \text{ for all } x \in X, \\ &\text{there is a path from } y \text{ to } t \text{ in } G[V \setminus X] \text{ for all } y \in \Gamma(X), t \in K \setminus X. \end{aligned} \tag{2.2}$$

But they showed that $\mathcal{I}$ does not necessarily satisfy property (P2)′. To see this, they considered the graph of Figure 1 in which $K = \{1,2,3,4\}$. $\mathcal{I}$ consists of {s}, $\{s,1\}$, $\{s,2\}$, $\{s,3\}$, $\{s,1,2,3\}$, $\{s,1,2,4\}$, $\{s,1,3,4\}$, and $\{s,2,3,4\}$. Let $S = \{s,1\}$. Then S has no pivot elements at all and thus $\mathcal{I}$ does not satisfy (P2)′. So it is difficult to list minimal (s,K)-cuts of an arbitrary directed graph by the framework.

3. A Generalized Framework for Listing

In this section, we give a generalization of the framework of the previous section.

Let $\mathcal{I}$ be a collection of subsets of the vertex set V of $G = (V,E)$. We relax the condition of a pivot element. For $S \in \mathcal{I}$ and $v \in V \setminus S$, let $\mathcal{Y}(S,v)$ be a set of all $Y \in \mathcal{I}$ satisfying

$$\begin{aligned} &S \cup \{v\} \subseteq Y, \\ &\nexists Z \in \mathcal{I}; S \cup \{v\} \subseteq Z \text{ and } Z \subsetneq Y. \end{aligned}$$

Namely, $\mathcal{Y}(S,v)$ is a set of minimal elements of $\mathcal{I}$ which contain $S \cup \{v\}$.

We define a pseudopivot element as a generalization of a pivot element.

Definition 3.1. *An element $v \in V \setminus S$ is a pseudopivot element for the set $S \in \mathcal{I}$ if the following conditions hold.*

1. *There is a minimal element of $\mathcal{I}$ which contains $S \cup \{v\}$.*
2. *For distinct $Y_i, Y_j \in \mathcal{Y}(S, v)$, there is no $Z \in \mathcal{I}$ such that $Y_i \cup Y_j \subseteq Z$.*

For example, consider $V = \{1, 2, 3, 4, 5, 6, 7\}$ and the collection $\mathcal{I}$ which consists of $\{1, 2\}$, $\{1, 2, 4\}$, $\{1, 2, 3, 4\}$, $\{1, 2, 3, 5, 6\}$, $\{1, 2, 3, 5, 7\}$ and $\{2, 3, 5\}$. Let $S = \{1, 2\}$ and $v = 3$. Then $\{1, 2, 3, 4\}$ is a minimal element of $\mathcal{I}$ which contains $S \cup \{v\}$. Also $\{1, 2, 3, 5, 6\}$ and $\{1, 2, 3, 5, 7\}$ are minimal elements of $\mathcal{I}$ which contains $S \cup \{v\}$. Hence $\mathcal{Y}(S, 3)$ consists of $\{1, 2, 3, 4\}$, $\{1, 2, 3, 5, 6\}$ and $\{1, 2, 3, 5, 7\}$. There does not exist $Z \in \mathcal{I}$ which contains any two distinct elements of $\mathcal{Y}(S, 3)$. So v is a pseudopivot element for S.

Note that if v is a pivot element for $S \in \mathcal{I}$, then v is a pseudopivot element for $S \in \mathcal{I}$.

Now we define a new property (P3) for $\mathcal{I}$.

(P3) For any $X, S \in \mathcal{I}$ with $X \supsetneq S$ there is a pseudopivot element v for S with $v \in X$.

For disjoint $S, T \subseteq V$ and $v \in V \setminus S$, define $\mathcal{J}(S, T, v) = \{Y \setminus S \mid Y \in \mathcal{Y}(S, v) \text{ and } Y \subseteq V \setminus T\}$. Using property (P3) we obtain the following lemma.

Lemma 3.1. *Suppose the collection $\mathcal{I}$ satisfies property (P3). Let $S \in \mathcal{I}$ and let $T \subseteq V \setminus S$ such that $|\mathcal{I}(S, T)| > 1$. Then there is a pseudopivot element v for S such that $Y \subseteq V \setminus T$ for some $Y \in \mathcal{Y}(S, v)$. Moreover, $\mathcal{I}(S, T \cup \{v\})$ and $\mathcal{I}(S \cup J, T)$ for all $J \in \mathcal{J}(S, T, v)$ are nonempty and they form a partition of $\mathcal{I}(S, T)$: i.e.,*

$$\mathcal{I}(S, T) = \mathcal{I}(S, T \cup \{v\}) \sqcup \bigsqcup_{J \in \mathcal{J}(S,T,v)} \mathcal{I}(S \cup J, T).$$

Proof. Since $|\mathcal{I}(S, T)| > 1$, there is $X \in \mathcal{I}(S, T)$ such that $X \supsetneq S$. By property (P3) there is a pseudopivot element v for S with $v \in X$. Since $X \in \mathcal{I}$ and $S \cup \{v\} \subseteq X$, there exists $Y \in \mathcal{Y}(S, v)$ such that $Y \subseteq X \subseteq V \setminus T$. Thus there is a pseudopivot element v for S such that $Y \subseteq V \setminus T$ for some $Y \in \mathcal{Y}(S, v)$.

We show that $\mathcal{I}(S, T \cup \{v\})$ and $\mathcal{I}(S \cup J, T)$ for all $J \in \mathcal{J}(S, T, v)$ form a partition of $\mathcal{I}(S, T)$. Consider any $U \in \mathcal{I}(S, T)$. If $v \notin U$, then we have $U \in \mathcal{I}(S, T \cup \{v\})$, and $U \notin \mathcal{I}(S \cup J, T)$ for any $J \in \mathcal{J}(S, T, v)$ by $S \cup \{v\} \subseteq S \cup J$. On the other hands, if $v \in U$, then we have $U \notin \mathcal{I}(S, T \cup \{v\})$, and there exists $J_U \in \mathcal{J}(S, T, v)$ such that $S \cup J_U \subseteq U$, thus $U \in \mathcal{I}(S \cup J_U, T)$, by $U \in \mathcal{I}(S, T)$ and $S \cup \{v\} \subseteq U$. Thus we observe that

$$\mathcal{I}(S, T) = \mathcal{I}(S, T \cup \{v\}) \sqcup \bigcup_{J \in \mathcal{J}(S,T,v)} \mathcal{I}(S \cup J, T).$$

For two distinct elements $J_i, J_j \in \mathcal{J}(S, T, v)$, we show $\mathcal{I}(S \cup J_i, T) \cap \mathcal{I}(S \cup J_j, T) = \emptyset$. By the definition of $\mathcal{I}(\cdot, \cdot)$, $\mathcal{I}(S \cup J_i, T) \cap \mathcal{I}(S \cup J_j, T) = \mathcal{I}(S \cup J_i \cup J_j, T)$. Since $J_i, J_j \in \mathcal{J}(S, T, v)$, there exist $Y_i, Y_j \in \mathcal{Y}(S, v)$ with $Y_i = S \cup J_i$ and $Y_j = S \cup J_j$. Then $\mathcal{I}(S \cup J_i \cup J_j, T) = \mathcal{I}(Y_i \cup Y_j, T)$. Since v is a pseudopivot element for S, there is no $Z \in \mathcal{I}$ such that $Y_i \cup Y_j \subseteq Z$, and hence we have $\mathcal{I}(Y_i \cup Y_j, T) = \emptyset$. Hence we get $\mathcal{I}(S \cup J_i, T) \cap \mathcal{I}(S \cup J_j, T) = \emptyset$. Thus $\mathcal{I}(S, T)$ is partitioned as

$$\mathcal{I}(S, T) = \mathcal{I}(S, T \cup \{v\}) \sqcup \bigsqcup_{J \in \mathcal{J}(S,T,v)} \mathcal{I}(S \cup J, T).$$

Finally, it is assured that $\mathcal{I}(S, T \cup \{v\})$ is nonempty since it contains S, and $\mathcal{I}(S \cup J, T)$ is nonempty for each $J \in \mathcal{J}(S, T, v)$ because it contains $S \cup J$. □

By Lemma 3.1, the recursive procedure pLIST(S,T) shown below lists all the elements of $\mathcal{I}(S,T)$, where $S \in \mathcal{I}$ and $T \subseteq V \setminus S$. In the procedure, pPIVOT$(S,T;v,\mathcal{J}(S,T,v))$ is a subroutine which takes the two sets S and T as input. The routine outputs a pseudopivot element v and the associated $\mathcal{J}(S,T,v)$ such that $Y \subseteq V \setminus T$ for some $Y \in \mathcal{Y}(S,v)$, if exists any. Such a pseudopivot element exists if $|\mathcal{I}(S,T)| > 1$ by Lemma 3.1. If there is no such pseudopivot element, then pPIVOT outputs $\mathcal{J}(S,T,v) = \emptyset$. In this case, $|\mathcal{I}(S,T)| = 1$ by $S \in \mathcal{I}$, and hence $\mathcal{I}(S,T) = \{S\}$. Thus pLIST(S,T) correctly lists all the elements of $\mathcal{I}(S,T)$.

Procedure pLIST(S,T)

```
pPIVOT(S,T;v,J(S,T,v)) ;
if J(S,T,v) = ∅ then
    output S ;
else
    pLIST(S,T∪{v}) ;
    foreach J ∈ J(S,T,v) do
        pLIST(S∪J,T) ;
```

The following theorem shows that if $\mathcal{I}$ contains the empty set, pLIST$(\emptyset,\emptyset)$ lists all the elements of $\mathcal{I}$.

Theorem 3.1. *Suppose that the collection $\mathcal{I}$ satisfies property (P3) and $\emptyset \in \mathcal{I}$. Then pLIST$(\emptyset,\emptyset)$ correctly lists all the elements of $\mathcal{I}$ in time $O(\tau(pPIVOT))$ per one element, where $\tau(pPIVOT)$ is the time complexity of the procedure pPIVOT.*

Proof. Once pLIST$(\emptyset,\emptyset)$ is carried out, the computation proceeds along a rooted tree $\mathbf{R}$ whose vertices represent calls to pLIST$(\cdot,\cdot)$. The root of $\mathbf{R}$ corresponds to $\mathcal{I}(\emptyset,\emptyset)$, and each vertex w_{ST} of $\mathbf{R}$ corresponds to some $\mathcal{I}(S,T)$, where $S \in \mathcal{I}$ and $T \subseteq V \setminus S$. At each vertex of $\mathbf{R}$, the procedure pPIVOT is invoked. In the case $|\mathcal{I}(S,T)| > 1$, pPIVOT$(S,T;v,\mathcal{J}(S,T,v))$ produces a pseudopivot element v and $\mathcal{J}(S,T,v)$ by Lemma 3.1. The children of w_{ST} in $\mathbf{R}$ correspond to $\mathcal{I}(S,T\cup\{v\}) \neq \emptyset$ and $\mathcal{I}(S\cup J,T) \neq \emptyset$ for all $J \in \mathcal{J}(S,T,v)$, respectively, forming a partition of $\mathcal{I}(S,T)$. Thus w_{ST} has at least two children in $\mathbf{R}$. In the case $|\mathcal{I}(S,T)| = 1$, pPIVOT$(S,T;v,\mathcal{J}(S,T,v))$ returns $\mathcal{J}(S,T,v) = \emptyset$, so w_{ST} is a leaf of $\mathbf{R}$ and pLIST(S,T) outputs $S \in \mathcal{I}$. Thus the leaves of $\mathbf{R}$ and the elements of $\mathcal{I}$ are in a one-to-one correspondence, so the number of leaves in $\mathbf{R}$ is $|\mathcal{I}|$. Since the vertices which are not leaves have at least two children, the number of vertices in $\mathbf{R}$, and hence the number of calls to pPIVOT, is less than $2|\mathcal{I}|$. Thus pLIST$(\emptyset,\emptyset)$ lists all the elements of $\mathcal{I}$ in time per element equal to the complexity of pPIVOT. □

Let $\mathcal{C}$ be a collection of cuts of G to be listed. Assume there exists a collection $\mathcal{I} \subseteq 2^V$ associated to $\mathcal{C}$ satisfying the following properties:

(P1) There is a one-to-one correspondence between cuts $C \in \mathcal{C}$ and sets $X \in \mathcal{I}$, defined by $X \in \mathcal{I} \Leftrightarrow C = E(X, V \setminus X) \in \mathcal{C}$.

(P3)′ For any $X, S \in \mathcal{I} \cup \{\emptyset\}$ with $X \supsetneq S$ there is a pseudopivot element v for S with $v \in X$.

Property (P3)′ enables us to list all the members of $\mathcal{I}$ by the recursive procedure pLIST$(\emptyset,\emptyset)$. Then pLIST$(\emptyset,\emptyset)$ lists all the members of $\mathcal{C}$.

In order to apply this framework to a given collection of cuts $\mathcal{C}$, the following steps are required:

1. Give a collection $\mathcal{I} \subseteq 2^V$ satisfying property (P1).
2. Identify pseudopivot elements and show that $\mathcal{I}$ satisfies property (P3)$'$.
3. Produce an implementation of pPIVOT routine and analyze its complexity.

4. Applications of the Framework

In this section we consider applications of the proposed framework.

At first, we consider the enumeration problems which can be computed by the framework of Provan and Shier. Suppose that the collection $\mathcal{C}$ of cuts of G has an associated collection $\mathcal{I}$ satisfying (P1) and (P2)$'$. We can show that $\mathcal{I}$ satisfies (P3)$'$ as follows. Let $X, S \in \mathcal{I} \cup \{\emptyset\}$ with $X \supsetneq S$. There is a pivot element v for S with $v \in X$ by (P2)$'$. By the definition, v is also a pseudopivot element for S with $v \in X$. Hence $\mathcal{I}$ satisfies (P3)$'$. In this case, we can use an implementation of PIVOT$(S, T; v, I(S, v))$ as an implementation of pPIVOT$(S, T; v, \mathcal{J}(S, T, v))$ by changing outputs from v and $I(S, v)$ to v and $\mathcal{J}(S, T, v)$. Thus the enumeration problems which can be computed by the framework of Provan and Shier can be computed by the proposed framework in the same time complexity.

Next we give the enumeration problem which can be computed by the proposed framework, though the existing framework cannot compute it.

Let $G = (V, E)$ be an undirected graph. We assume that the graph G is connected. Let $s \in V$ and $K \subseteq V \setminus \{s\}$. Let h be an integer with $1 \leq h \leq |K|$. We define that a set of edges $E' \subseteq E$ is an sh-cut of G if there are no paths from s to all vertices of H and all vertices of H are connected in $G - E'$ for some $H \subseteq K$ with $|H| \geq h$. We consider an enumeration of the collection $\mathcal{C}$ of minimal sh-cuts of G. When $h = 1$, sh-cuts are equal to (s, K)-cuts. Let $\mathcal{I}$ be the collection of $X \subseteq V$ satisfying (4.1). We can easily check $\emptyset, V \notin \mathcal{I}$.

$$\begin{aligned} &s \in X, \quad |K \setminus X| \geq h, \\ &G[X] \text{ is connected}, \\ &G[V \setminus X] \text{ is connected}. \end{aligned} \tag{4.1}$$

We show the collection $\mathcal{I}$ of (4.1) satisfies (P1).

Theorem 4.1. *Let $G = (V, E)$ be a connected undirected graph with $s \in V$ and $K \subseteq V \setminus \{s\}$. Let h be an integer with $1 \leq h \leq |K|$. Then $C \subseteq E$ is a minimal sh-cut of G if and only if $C = E(X, V \setminus X)$ where $X \subseteq V$ satisfies (4.1). Moreover, (P1) holds for the collection $\mathcal{I}$ of $X \subseteq V$ satisfying (4.1), by using the correspondence $C = E(X, V \setminus X)$.*

Proof. Suppose that $C \subseteq E$ is a minimal sh-cut of G, so that there are no paths from s to all vertices of H and all vertices of H are connected in $G - C$ for some $H \subseteq K$ with $|H| \geq h$. Some connected component G_1 contains s and another connected component G_2 contains all vertices of H. We show that $G - C$ has two connected components. Assume that $G - C$ has more than two connected components. Then there is $e_1 \in E$ from G_3 to G_1 or $e_2 \in E$ from G_3 to G_2 for some connected component G_3 which is not G_1 and G_2. If e_1 exists, the graph G' obtained from G by deleting $C \setminus \{e_1\}$ is the same as $G - C$ except that G_1 and G_3 are connected. So there are no paths from s to all vertices of H in G', and $C \setminus \{e_1\}$ is also an sh-cut of G. This is a contradiction. If e_2 exists, we can show $C \setminus \{e_2\}$ is an sh-cut of G by the same way. This is a contradiction. So $G - C$ has two connected components. Let X be the vertex set of G_1. Then the vertex set of G_2 is $V \setminus X$ and X satisfies $s \in X$ and $|K \setminus X| \geq h$. Also $E(X, V \setminus X) \subseteq C$. If C contains $e \in E(X)$, adding e to $G - C$ does not

cause a change about connectivity between vertices, and hence $C \setminus \{e\}$ is also an sh-cut of G. So C does not contain $e \in E(X)$. And C does not contain $e \in E(V \setminus X)$ for the same reason. Hence $C = E(X, V \setminus X)$, $G_1 = G[X]$ and $G_2 = G[V \setminus X]$. Thus $G[X]$ is connected and $G[V \setminus X]$ is connected. As a result X satisfies all conditions in (4.1).

Conversely, suppose X satisfies (4.1). Let $C = E(X, V \setminus X)$. Then there are no paths from any vertex $u \in X$ to any vertex $v \in V \setminus X$ in $G - C$. Let H be a subset of $K \setminus X$ such that $|H| \geq h$. Then all vertices of H are connected in $G - C$. So there are no paths from s to all vertices of H and all vertices of H are connected in $G - C$. Hence C is an sh-cut of G. For $e \in E(X, V \setminus X)$ let G'' be the graph obtained from G by deleting $C \setminus \{e\}$. Then there is a path using e from any vertex $u \in X$ to any vertex $v \in V \setminus X$ in G''. So there is a path from s to some vertex of H, and hence $C \setminus \{e\}$ is not an sh-cut of G. Thus C is a minimal sh-cut of G.

To show (P1), suppose that $X \neq Y$ define the same minimal sh-cut C of G; without loss of generality assume that there is some $y \in Y \setminus X$. Since $E[Y]$ is connected, there is a path from s to y in $E[Y]$. Let $(v_0, v_1, \ldots, v_k)$ be such path where $v_0 = s$, $v_k = y$ and $(v_{i-1}, v_i) \in E$ for $i = 1, 2, \ldots, k$. Let q be the smallest index such that $v_{q-1} \in X$ and $v_q \notin X$. Then edge $e = (v_{i-1}, v_i)$ is in $E(X, V \setminus X)$ but e is not in $E(Y, V \setminus Y)$. This is a contradiction. Thus property (P1) holds. □

To carry out Step 2 of the framework, we identify pseudopivot elements for $S \in \mathcal{I} \cup \{\emptyset\}$. For $S \in \mathcal{I} \cup \{\emptyset\}$, we define $W(S)$ as follows, where $\Gamma(S) = \{v \in V \setminus S \mid \exists u \in S; (u, v) \in E\}$.

$$W(S) = \begin{cases} \{s\} & \text{if } S = \emptyset, \\ \Gamma(S) & \text{if } S \neq \emptyset \text{ and } |K \setminus S| > h, \\ \Gamma(S) \setminus K & \text{if } S \neq \emptyset \text{ and } |K \setminus S| = h. \end{cases}$$

We defined $W(S)$ to satisfy $s \in S \cup \{v\}$, $|K \setminus (S \cup \{v\})| \geq h$ and $G[S \cup \{v\}]$ is connected for any $v \in W(S)$. Note that if $S = \emptyset$, $v = s$ satisfies $s \in S \cup \{v\}$, $|K \setminus (S \cup \{v\})| \geq h$ and $G[S \cup \{v\}]$ is connected. If $S \neq \emptyset$ and $|K \setminus S| > h$ then $S \in \mathcal{I}$. So $s \in S \cup \{v\}$ for any $v \in W(S)$. Since $|K \setminus S| > h$, $|K \setminus (S \cup \{v\})| \geq h$. Since $G[S]$ is connected and $v \in \Gamma(S)$, $G[S \cup \{v\}]$ is connected. If $S \neq \emptyset$ and $|K \setminus S| = h$ then $S \in \mathcal{I}$. So $s \in S \cup \{v\}$ for any $v \in W(S)$. Since $|K \setminus S| = h$ and $v \notin K$, $|K \setminus (S \cup \{v\})| = h$. Since $G[S]$ is connected and $v \in \Gamma(S)$, $G[S \cup \{v\}]$ is connected.

In Lemma 4.1, for $S, X \in \mathcal{I} \cup \{\emptyset\}$ with $S \subsetneq X$, we show that any $v \in W(S)$ is a pseudopivot element for S.

Lemma 4.1. *Let $G = (V, E)$ be an undirected graph with $s \in V$ and $K \subseteq V \setminus \{s\}$. Let h be an integer with $1 \leq h \leq |K|$. Let $\mathcal{I}$ be a collection characterized by (4.1). Then for $S, X \in \mathcal{I} \cup \{\emptyset\}$ with $S \subsetneq X$, any $v \in W(S)$ is a pseudopivot element for S.*

Proof. Let v be any vertex in $W(S)$. $s \in S \cup \{v\}$, $|K \setminus (S \cup \{v\})| \geq h$ and $G[S \cup \{v\}]$ is connected by the definition of $W(S)$. We separate the argument in two cases accordingly whether $G[V \setminus (S \cup \{v\})]$ is connected or not. First, consider the case $G[V \setminus (S \cup \{v\})]$ is connected. Since $s \in S \cup \{v\}$, $|K \setminus (S \cup \{v\})| \geq h$ and $G[S \cup \{v\}]$ is connected, $S \cup \{v\} \in \mathcal{I} \cup \{\emptyset\}$. So $\mathcal{Y}(S, v) = \{S \cup \{v\}\}$. Thus v is a pseudopivot element for S.

In the rest, discuss the case $G[V \setminus (S \cup \{v\})]$ is not connected. Let $U = V \setminus (S \cup \{v\})$. Let $\mathcal{U}$ be a collection of vertex sets of each connected component of $G[U]$. Consider any $N \in \mathcal{U}$. $G[N]$ is connected by definition. Since $G[V \setminus S]$ is connected and $G[U] = G[(V \setminus S) \setminus \{v\}]$ has more than one connected components, there exists $u \in N$ such that $(u, v) \in E$. Hence $G[\{v\} \cup N]$ is connected.

Consider $Y \in \mathcal{I} \cup \{\emptyset\}$ such that $S \cup \{v\} \subseteq Y$. Since $V \setminus Y \subseteq U$, $G[V \setminus Y]$ is a subgraph of $G[U]$. In order that $G[V \setminus Y]$ is connected, $G[Y]$ should contain all connected components of $G[U]$ except one connected component. Namely, $U \setminus N \subseteq Y$ for some $N \in \mathcal{U}$. So if $Y \in \mathcal{Y}(S, v)$, then there exists $N \in \mathcal{U}$ such that $S \cup \{v\} \cup (U \setminus N) \subseteq Y$.

Let $N \in \mathcal{U}$ and $Y_N = S \cup \{v\} \cup (U \setminus N) = V \setminus N$. We consider $Y_N \in \mathcal{I} \cup \{\emptyset\}$ or not. Here, $G[Y_N]$ is connected since $G[S \cup \{v\}]$ and $G[\{v\} \cup N']$ for $N' \in \mathcal{U}$ are all connected. Also $G[V \setminus Y_N]$ is connected since $V \setminus Y_N = N$. From $s \in S \subseteq Y_N$, we conclude that $Y_N \in \mathcal{I} \cup \{\emptyset\}$ if $|K \setminus Y_N| \geq h$.

Let $L_1 = \{N \in \mathcal{U} \mid |K \setminus Y_N| \geq h\}$ and $L_2 = \{N \in \mathcal{U} \mid |K \setminus Y_N| < h\}$. Here we have $L_1 \cup L_2 = \mathcal{U}$ and $L_1 \cap L_2 = \emptyset$. Note that $L_1 \neq \emptyset$ by $S \subsetneq X$ for $X \in \mathcal{I} \cup \{\emptyset\}$. For each $N \in L_1$, we have $Y_N \in \mathcal{I} \cup \{\emptyset\}$ by $|K \setminus Y_N| \geq h$. Let Z be any proper subset of Y_N with $S \cup \{v\} \subseteq Z \subsetneq Y_N$. Since $N = V \setminus Y_N \subsetneq V \setminus Z$, $G[V \setminus Z]$ has more than one connected components, thus $G[V \setminus Z]$ is not connected. So $Z \notin \mathcal{I} \cup \{\emptyset\}$, and thus we have $Y_N \in \mathcal{Y}(S, v)$. On the other hands, for each $N \in L_2$, we have $Y_N \notin \mathcal{I} \cup \{\emptyset\}$ by $|K \setminus Y_N| < h$, and hence $Y_N \notin \mathcal{Y}(S, v)$. It is obvious that there are no other $W \subseteq V$ in $\mathcal{Y}(S, v)$, and we have $\mathcal{Y}(S, v) = \{Y_N = V \setminus N \mid N \in L_1\}$. Since $\mathcal{Y}(S, v) \neq \emptyset$ by $L_1 \neq \emptyset$, v satisfies the first condition of a pseudopivot element for S.

Now we show that v satisfies the second condition of a pseudopivot element for S. We consider two distinct elements $Y_i, Y_j \in \mathcal{Y}(S, v)$. For some $N_i, N_j \in L_1$ with $N_i \neq N_j$, $Y_i = S \cup \{v\} \cup (U \setminus N_i)$ and $Y_j = S \cup \{v\} \cup (U \setminus N_j)$, and hence $Y_i \cup Y_j = S \cup \{v\} \cup (U \setminus N_i) \cup (U \setminus N_j) = S \cup \{v\} \cup U = V$. Since $V \notin \mathcal{I}$, there is no $Z \in \mathcal{I} \cup \{\emptyset\}$ with $Y_i \cup Y_j \subseteq Z$.

Thus v is a pseudopivot element for S. □

We can show the collection $\mathcal{I}$ of (4.1) satisfies (P3)′ by using Lemma 4.1.

Theorem 4.2. *Let $G = (V, E)$ be an undirected graph with $s \in V$ and $K \subseteq V \setminus \{s\}$. Let h be an integer with $1 \leq h \leq |K|$. Then the collection $\mathcal{I}$ characterized by (4.1) satisfies property* (P3)′.

Proof. Suppose that $S, X \in \mathcal{I} \cup \{\emptyset\}$ with $S \subsetneq X$. We show $W(S) \cap X \neq \emptyset$. Since $S \subsetneq X$, $G[X]$ is connected and $s \in X$, $\Gamma(S) \cap X \neq \emptyset$. If $|K \setminus S| = h$, then $W(S) = \Gamma(S) \setminus K$. By $|K \setminus X| = h$ and $\Gamma(S) \cap X \subseteq X \setminus S$, $(\Gamma(S) \cap X) \cap K = \emptyset$. Thus $W(S) \cap X = (\Gamma(S) \setminus K) \cap X = \Gamma(S) \cap X \neq \emptyset$. If $|K \setminus S| > h$, then $W(S) = \Gamma(S)$. Thus $W(S) \cap X = \Gamma(S) \cap X \neq \emptyset$.

Since $W(S) \cap X \neq \emptyset$, we can choose some $v \in W(S) \cap X$, and this v is a pseudopivot element for S by Lemma 4.1. Since $v \in X$, $\mathcal{I}$ satisfies property (P3)′. □

Theorem 4.2 shows the collection $\mathcal{I}$ of (4.1) satisfies (P3)′. To give the enumeration algorithm for $\mathcal{I}$ of (4.1), we need the procedure pPIVOT relative to given $S \in \mathcal{I} \cup \{\emptyset\}$ and $T \subseteq V \setminus S$. We use the following corollary of the proof of Lemma 4.1 that identifies $\mathcal{Y}(S, v)$ for $S \in \mathcal{I} \cup \{\emptyset\}$ and $v \in W(S)$.

Corollary 4.1. *Let $G = (V, E)$ be an undirected graph with $s \in V$ and $K \subseteq V \setminus \{s\}$, h an integer with $1 \leq h \leq |K|$, $\mathcal{I}$ a collection characterized by (4.1). Suppose that $S \in \mathcal{I} \cup \{\emptyset\}$ and $v \in W(S)$. Let $\mathcal{U}$ be the collection of the vertex sets of the connected components of $G[V \setminus (S \cup \{v\})]$. Then $\mathcal{Y}(S, v) = \{V \setminus N \mid N \in \mathcal{U}$ and $|K \cap N| \geq h\}$.*

By Corollary 4.1, for $S \in \mathcal{I} \cup \{\emptyset\}$, $T \subseteq V \setminus S$ and $v \in W(S)$,

$$\mathcal{J}(S, T, v) = \{(V \setminus N) \setminus S \mid N \in \mathcal{U},\ |K \cap N| \geq h \text{ and } T \subseteq N\}, \tag{4.2}$$

where $\mathcal{U}$ is defined in Corollary 4.1 and $\mathcal{J}(S, T, v) = \{Y \setminus S \mid Y \in \mathcal{Y}(S, v) \text{ and } Y \subseteq V \setminus T\}$.

We now discuss Step 3 of the framework. We consider pPIVOT$(S, T; v, \mathcal{J}(S, T, v))$. We only have to identify a pseudopivot element $v \in W(S)$ for S that satisfies $Y \subseteq V \setminus T$ for

some $Y \in \mathcal{Y}(S, v)$, since if there exists any pseudopivot element x for S such that $Y \subseteq V \setminus T$ for some $Y \in \mathcal{Y}(S, x)$ then there must exist one which lies in $W(S)$. By (4.2), for $v \in W(S)$ we count the number of K of each connected component of $G[V \setminus (S \cup \{v\})]$ and search the connected component of $G[V \setminus (S \cup \{v\})]$ which satisfies the condition of $\mathcal{J}(S, T, v)$. We can compute the number of K efficiently by using the biconnected components of $G[V \setminus S]$. If there exists the connected component of $G[V \setminus (S \cup \{v\})]$ which satisfies the condition of $\mathcal{J}(S, T, v)$ for some $v \in W(S)$, we compute $\mathcal{J}(S, T, v)$ and output v and $\mathcal{J}(S, T, v)$. Otherwise $\mathcal{J}(S, T, v) = \emptyset$. Below we describe an implementation of pPIVOT$(S, T; v, \mathcal{J}(S, T, v))$ that takes $O(|E|)$ time.

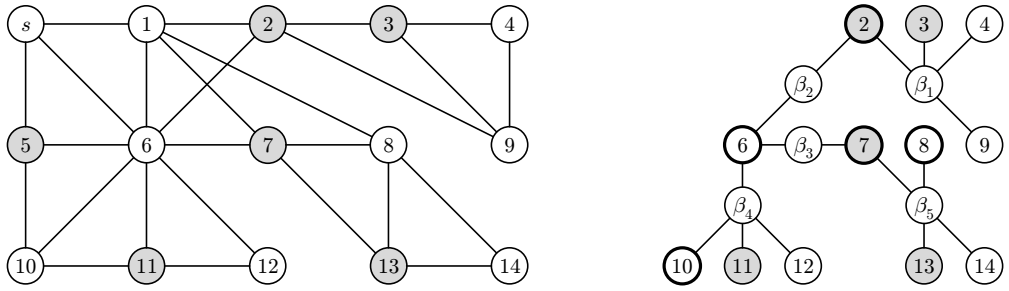
1. If $S = \emptyset$ then $W(S) := \{s\}$ and go to the next step. Else compute $\Gamma(S)$ and $|K \setminus S|$. Let $W(S) := \Gamma(S)$. If $|K \setminus S| = h$ then $W(S) := \Gamma(S) \setminus K$.
2. Construct the graph $G[V \setminus S]$.
3. Compute the biconnected components of $G[V \setminus S]$. Let $B_1, B_2, \ldots, B_b$ be the biconnected components of $G[V \setminus S]$.
4. Construct the graph $\mathbf{D} = (V_T, E_T)$, where the vertex set V_T consists of $V \setminus S$ and a vertex β_i for each biconnected component B_i of $G[V \setminus S]$ and the edge set E_T consists of all pairs (u, β_i) such that u is a vertex of B_i. It is easy to see that $\mathbf{D}$ is a tree.
5. For each edge $e = (u, \beta_i)$ of $\mathbf{D}$, compute $k(e)$ and $t(e)$, where $k(e)$ is the number of vertices of K in the connected component of $\mathbf{D} - \{e\}$ that β_i belongs to, and $t(e)$ is the number of vertices of T in the same connected component.
6. For each $v \in W(S)$, check whether there exists an edge e incident to v in $\mathbf{D}$ such that $k(e) \geq h$ and $t(e) = |T|$. If it holds for some v, choose v as a pseudopivot element to output, then go to the next step. Otherwise output $\mathcal{J}(S, T, v) = \emptyset$ and end the procedure.
7. Let $\mathcal{A} := \emptyset$.
8. For each edge $e = (v, w)$ incident to v in $\mathbf{D}$, if $k(e) \geq h$ and $t(e) = |T|$, then compute the vertex set U_w of the connected component of $\mathbf{D} - v$ that w belongs to. And let $\mathcal{A} := \mathcal{A} \cup \{(V \setminus S) \setminus U_w\}$.
9. Output v and $\mathcal{J}(S, T, v) = \mathcal{A}$ and end the procedure.

This implementation uses the biconnected components and the associated tree $\mathbf{D}$, borrowing the idea used in Provan and Shier [5]. If the deletion of vertex a from graph G' is not connected, then a is called an articulation point of G'. G' is biconnected if and only if there are no articulation points. A maximal subgraph of G' that is biconnected is called a biconnected component of G'.

We consider the correctness of our implementation of pPIVOT. For any $v \in V \setminus S$, let V_v be the set of vertices adjacent to v in $\mathbf{D}$. Let $V_B = \{\beta_1, \beta_2, \ldots, \beta_b\}$. We can check $V_v \subseteq V_B$. Note that, for any $v \in W(S)$, $U_w \cap V$ for some $w \in V_v$ is a vertex set of some connected component of $G[V \setminus S] - v$. And the collection of $U_w \cap V$ of all $w \in V_v$ is equal to the collection of vertex sets of all connected components of $G[V \setminus S] - v$. Consider any $v \in W(S)$. For any $w \in V_v$, let $e = (v, w)$. If $k(e) \geq h$, then $|K \cap (U_w \cap V)| \geq h$. If not $|K \cap (U_w \cap V)| < h$. Hence $\mathcal{Y}(S, v) = \{V \setminus U_w \mid k(e) \geq h \text{ with } e = (v, w) \text{ and } w \in V_v\}$. Moreover, if $t(e) = |T|$, then $T \subseteq U_w$ and hence $V \setminus U_w \subseteq V \setminus T$. On the other hand, if $t(e) < |T|$, then $(V \setminus U_w) \cap T \neq \emptyset$. So we get

$$\mathcal{J}(S, T, v) = \{(V \setminus S) \setminus U_w \mid k(e) \geq h \text{ and } t(e) = |T|, e = (v, w), w \in V_v\}.$$

In Step 6, identify the vertex $v \in W(S)$ that is a pseudopivot element for S such that $Y \subseteq V \setminus T$ for some $Y \in \mathcal{Y}(S, v)$. The discussion presented so far shows the correctness of our implementation of pPIVOT.



(a) (b)

Figure 2: The graph G and its associated tree $\mathbf{D}$.

To illustrate this procedure, consider the graph G shown in Figure 2 (a), where $K = \{2, 3, 5, 7, 11, 13\}$ and $h = 2$. Let $S = \{s, 1, 5\}$ and $T = \{7, 8\}$. The biconnected components of $G[V \setminus S]$ are $B_1 = \{2, 3, 4, 9\}$, $B_2 = \{2, 6\}$, $B_3 = \{6, 7\}$, $B_4 = \{6, 10, 11, 12\}$, and $B_5 = \{7, 8, 13, 14\}$. The tree $\mathbf{D}$ is shown in Figure 2 (b); the elements of $W(S) = \{2, 6, 7, 8, 10\}$ are shown in bold. For $6 \in W(S)$, the edge $e = (6, \beta_3)$ satisfies $k(e) = 2 \geq h$ and $t(e) = 2 = |T|$. So we can choose 6 as a pseudopivot element to output. Since there is no other edge e' incident to 6 in $\mathbf{D}$ such that $k(e') \geq h$ and $t(e') = |T|$, $\mathcal{J} = \{\{2, 3, 4, 6, 9, 10, 11, 12\}\}$.

Now we analyze the complexity of pPIVOT. Steps 1 and 2 can both be computed in $O(|E|)$ time. Steps 3 and 4 can be carried out in $O(|E|)$ time using the biconnected component algorithm [2]. For Step 5, we give an algorithm that runs in $O(|V|)$ time as follows. For vertices $v \in V_T$, we define a function f as:

$$f(v) = \begin{cases} 1 & \text{if } v \in K, \\ 0 & \text{otherwise.} \end{cases}$$

$f(\beta) = 0$ for all $\beta \in V_B$. Let A be the sum of $f(v)$ over all $v \in V_T$. We have $A = |K \setminus S|$. For an edge $e \in E_T$, let $u \in V \setminus S$ and $\beta \in V_B$ be the endpoints of e. $k(e)$ equals to the sum of $f(v)$ over all vertices v of the connected component of $\mathbf{D} - \{e\}$ that β belongs to. Since $\mathbf{D}$ is a tree, $A - k(e)$ is the sum of $f(v)$ over all vertices v of the connected component of $\mathbf{D} - \{e\}$ that u belongs to. The following algorithm computes $k(e)$ for all $e \in E_T$. The algorithm repeats over all vertices in $\mathbf{D}$ to update $k(e)$ for all $e \in E_T$.

1. Let $\mathbf{D}'$ be a tree obtained by copying $\mathbf{D}$.
2. For each $e \in E_T$, set $k(e) := 0$.
3. Choose a leaf v of $\mathbf{D}'$.
   If $v \in V \setminus S$ then there is a unique vertex $\beta \in V_B$ adjacent to v in $\mathbf{D}'$. Let $e = (v, \beta)$, and do the following:
   $k(e) := A - f(v)$.
   For each edge $e' \neq e$ which is incident to v in $\mathbf{D}$, $k(e) := k(e) - k(e')$.
   Delete v and e from $\mathbf{D}'$.
   If $v \in V_B$ then there is a unique vertex $y \in V \setminus S$ adjacent to v in $\mathbf{D}'$. Let $e = (v, y)$, and do the following:
   For each edge $e' \neq e$ which is incident to v in $\mathbf{D}$, $k(e) := k(e) + (A - k(e'))$.
   Delete v and e from $\mathbf{D}'$.
4. If $\mathbf{D}'$ has only one vertex, end the procedure. Otherwise, go to 3.

In the above algorithm, each vertex is chosen only one time. In Step 3, suppose that a leaf v of $\mathbf{D}'$ is chosen. All the vertices adjacent to v in $\mathbf{D}$ are already chosen except one.

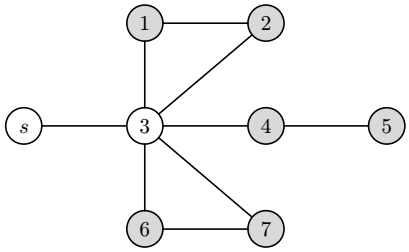


Figure 3: The graph G_1: $\mathcal{I}$ does not satisfy (P2)′ for $K = \{1, 2, 4, 5, 6, 7\}$.

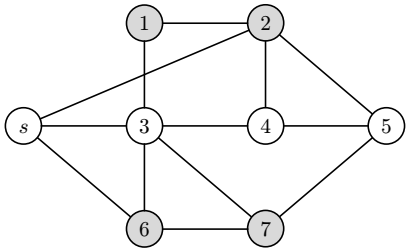


Figure 4: The graph G_2: $\mathcal{I}'$ does not satisfy (P2)′ for $K = \{1, 2, 6, 7\}$.

Let E_v be the set of edges which are incident to v in $\mathbf{D}$. When $v \in V \setminus S$, v is adjacent only to $\beta \in V_B$ in $\mathbf{D}'$. Let $e = (v, \beta)$. The vertices of the connected component of $\mathbf{D} - \{e\}$ that v belongs to are v and the vertices of the connected component of $\mathbf{D} - \{e'\}$ that $\gamma \in V_B$ belonged to for each $e' = (v, \gamma) \in E_v \setminus \{e\}$. Thus we have

$$A - k(e) = f(v) + \sum_{e' \in E_v \setminus \{e\}} k(e').$$

When $v \in V_B$, v is adjacent only to $y \in V \setminus S$ in $\mathbf{D}'$. Let $e = (v, y)$. The vertices of the connected component of $\mathbf{D} - \{e\}$ that v belongs to are v and the vertices of the connected component of $\mathbf{D} - e'$ that $w \in V \setminus S$ belonged to for each $e' = (v, w) \in E_v \setminus \{e\}$. Thus we have

$$k(e) = \sum_{e' \in E_v \setminus \{e\}} (A - k(e')).$$

Thus the above algorithm correctly computes $k(e)$ for all $e \in E_T$. The complexity of the above algorithm is $O(|V|)$ since the number of vertices of $\mathbf{D}$ is at most $2|V|$. For vertices $v \in V_T$, we define a function t as:

$$t(v) = \begin{cases} 1 & \text{if } v \in T, \\ 0 & \text{otherwise.} \end{cases}$$

Since $T \subseteq V$, $t(v) = 0$ for all $v \in V_B$. By replacing the function f with the function t, the above algorithm computes $t(e)$ for all $e \in E_T$.

Step 6 and Step 7 of our implementation of pPIVOT can be carried out in $O(|V|)$ because $\mathbf{D}$ is a tree with at most $2|V|$ vertices. Step 8 can also be carried out in $O(|V|)$ by applying the depth-first search to $\mathbf{D} - v$ because $\mathbf{D} - v$ is a forest with at most $2|V|$ vertices. Overall, the complexity of pPIVOT is $O(|E|)$. Thus our framework provides an algorithm for an enumeration of the collection $\mathcal{I}$ of $X \subseteq V$ satisfying (4.1), which requires $O(|E|)$ time per one element listed.

In the last we consider whether the framework of Provan and Shier [5] can be applied for an enumeration of the collection $\mathcal{I}$ of $X \subseteq V$ satisfying (4.1).

We can easily check that $\mathcal{I}$ does not necessarily satisfy property (P2)′. For example, consider the graph G_1 of Figure 3, where $K = \{1, 2, 4, 5, 6, 7\}$ and $h = 2$. $\mathcal{I}$ consists of the sets $\{s\}$, $\{s, 1, 2, 3, 4, 5\}$, $\{s, 1, 2, 3, 6, 7\}$, and $\{s, 3, 4, 5, 6, 7\}$. Let $S = \{s\}$ and $X = \{s, 1, 2, 3, 4, 5\}$. There are no pivot elements for S in X as follows. Since $S \cup \{1\}$ is contained in X and $\{s, 1, 2, 3, 6, 7\}$, the vertex 1 is not a pivot element for S. The vertex 2 is not a pivot element for S, $S \cup \{2\}$ is contained in X and $\{s, 1, 2, 3, 6, 7\}$. The vertices 3, 4 and 5

are not pivot elements for S, since both $S \cup \{3\}$, $S \cup \{4\}$ and $S \cup \{5\}$ are contained in X and $\{s, 3, 4, 5, 6, 7\}$. Thus the collection $\mathcal{I}$ does not satisfy (P2)′.

We can also consider the collection $\mathcal{I}'$ of $X \subseteq V$ satisfying (4.3) instead of $\mathcal{I}$ in the same way as the enumeration of minimal (s, K)-cut in Provan and Shier [5].

$$\begin{aligned} &|X \cap K| \geq h, \quad s \in V \setminus X, \\ &G[X] \text{ is connected}, \\ &G[V \setminus X] \text{ is connected}. \end{aligned} \tag{4.3}$$

But we can easily check that the collection $\mathcal{I}'$ does not necessarily satisfy property (P2)′. For example, consider the graph G_2 of Figure 4, where $K = \{1, 2, 6, 7\}$ and $h = 2$. Let $S = \emptyset$ and $X = \{1, 3, 6\} \in \mathcal{I}'$. There are no pivot elements for S in X as follows. The vertex 1 is not a pivot element for S, since $S \cup \{1\} \notin \mathcal{I}'$ is contained in X and $\{1, 2\} \in \mathcal{I}'$. The vertex 3 is not pivot elements for S, since $S \cup \{3\} \notin \mathcal{I}'$ is contained in X and $\{1, 3, 7\} \in \mathcal{I}'$. The vertex 6 is not a pivot element for S, since $S \cup \{6\} \notin \mathcal{I}'$ is contained in X and $\{6, 7\} \in \mathcal{I}'$. Thus the collection $\mathcal{I}'$ does not satisfy (P2)′.

Since $\mathcal{I}$ and $\mathcal{I}'$ do not satisfy (P2)′, the framework of Provan and Shier [5] can not be applied for an enumeration of the collection $\mathcal{I}$ without a new idea.

References

[1] U. Abel and R. Bicker: Determination of all minimal cut-sets between a vertex pair in an undirected graph. *IEEE Transactions on Reliability*, **31** (1982), 167–171.

[2] A.V. Aho, J.E. Hopcroft, and J.D. Ullman: *The Design and Analysis of Computer Algorithms* (Addison-Wesley, Boston, 1974).

[3] M. Bellmore and P.A. Jensen: An implicit enumeration scheme for proper cut generation. *Technometrics*, **12** (1970), 775–788.

[4] C.J. Colbourn: *The Combinatorics of Network Reliability* (Oxford University Press, New York, 1987).

[5] J.S. Provan and D.R. Shier: A Paradigm for Listing (s, t)-Cuts in Graphs. *Algorithmica*, **15** (1996), 351–372.

[6] S. Tsukiyama, I. Shirakawa, H. Ozaki, and H. Ariyoshi: An algorithm to enumerate all cutsets of a graph in linear time per cutset. *Journal of the ACM*, **27** (1980), 619–632.

Makoto Yaguchi
Graduate School of Systems and Information Engineering
University of Tsukuba
1-1-1 Tennoudai Tsukuba
Ibaraki 305-8573, Japan
E-mail: yaguch40@sk.tsukuba.ac.jp