

# Generating Cut Conjunctions and Bridge Avoiding Extensions in Graphs <sup>★</sup>

L. Khachiyan<sup>1</sup>, E. Boros<sup>2</sup>, K. Borys<sup>2</sup>, K. Elbassioni<sup>3</sup>, V. Gurvich<sup>2</sup>, and K. Makino<sup>4</sup>

<sup>1</sup> Department of Computer Science, Rutgers University, 110 Frelinghuysen Road, Piscataway NJ 08854;<sup>★★</sup>

<sup>2</sup> RUTCOR, Rutgers University, 640 Bartholomew Road, Piscataway NJ 08854; {boros,kborys,gurvich}@rutcor.rutgers.edu

<sup>3</sup> Max-Planck-Institut für Informatik, Saarbrücken, Germany; elbassio@mpi-sb.mpg.de

<sup>4</sup> Division of Mathematical Science for Social Systems, Graduate School of Engineering Science, Osaka University, Toyonaka, Osaka, 560-8531, Japan; makino@sys.es.osaka-u.ac.jp

**Abstract.** Let  $G = (V, E)$  be an undirected graph, and let  $B \subseteq V \times V$  be a collection of vertex pairs. We give an incremental polynomial time algorithm to enumerate all minimal edge sets  $X \subseteq E$  such that every vertex pair  $(s, t) \in B$  is disconnected in  $(V, E \setminus X)$ , generalizing well-known efficient algorithms for enumerating all minimal  $s$ - $t$  cuts, for a given pair  $s, t \in V$  of vertices. We also present an incremental polynomial time algorithm for enumerating all minimal subsets  $X \subseteq E$  such that no  $(s, t) \in B$  is a bridge in  $(V, X \cup B)$ . These two enumeration problems are special cases of the more general cut conjunction problem in matroids: given a matroid  $M$  on ground set  $S = E \cup B$ , enumerate all minimal subsets  $X \subseteq E$  such that no element  $b \in B$  is spanned by  $E \setminus X$ . Unlike the above special cases, corresponding to the cycle and cocycle matroids of the graph  $(V, E \cup B)$ , the enumeration of cut conjunctions for vectorial matroids turns out to be NP-hard.

## 1 Introduction

The *cut enumeration problem* for graphs calls for listing all minimal subsets of edges whose removal disconnects two specified vertices of a given graph. This so called *two-terminal cut enumeration problem* is known to be solvable in  $O(Nm + m + n)$  time and  $O(n + m)$  space [9], where  $n$  and  $m$  are the numbers of vertices and edges in the input graph, and  $N$  is the number of cuts. In this paper, we study the following extension of the two-terminal cut enumeration problem:

---

<sup>\*</sup> This research was partially supported by the National Science Foundation (Grant IIS-0118635), and by DIMACS, the National Science Foundation's Center for Discrete Mathematics and Theoretical Computer Science.

<sup>\*\*</sup> Our friend and co-author, Leonid Khachiyan passed away with tragic suddenness, while we were working on the final version of this paper.

**Cut Conjunctions in Graphs:** *Given an undirected graph  $G = (V, E)$ , and a collection  $B = \{(s_1, t_1), \dots, (s_k, t_k)\}$  of  $k$  vertex pairs  $s_i, t_i \in V$ , enumerate all minimal edge sets  $X \subseteq E$  such that for all  $i = 1, \dots, k$ , vertices  $s_i$  and  $t_i$  are disconnected in  $G' = (V, E \setminus X)$ .*

We call a minimal edge set  $X \subseteq E$  for which all pairs of vertices  $(s_i, t_i) \in B$  are disconnected in the subgraph  $G' = (V, E \setminus X)$ , a *minimal  $B$ -cut*, or simply a *cut conjunction* if  $B$  is clear from the context, and we denote by  $\mathcal{F}_{G,B}$  the family of all minimal  $B$ -cuts. Observe that each edge set  $X \in \mathcal{F}_{G,B}$  must indeed be the union (conjunction) of some minimal  $s_i$ - $t_i$  cuts for  $i = 1, \dots, k$ , justifying the naming of these edge sets. Let us also note that the enumeration of cut conjunctions cannot efficiently be reduced to two-terminal cut enumeration, because the majority of conjunctions of minimal  $s_i$ - $t_i$  cuts, for  $i = 1, \dots, k$ , may be nonminimal  $B$ -cuts (see [1] for an example).

In what follows, we assume without any loss of generality that in the above cut conjunction problem, no pair of vertices  $(s_i, t_i)$  is connected by an edge of  $G$ , i.e.  $E \cap B = \emptyset$  (since all such edges would have to belong to all cut conjunctions).

When  $B$  is the collection of all pairs of distinct vertices drawn from some vertex set  $V' \subseteq V$ , minimal  $B$ -cuts are known as *multiway cuts*, see e.g. [10]. The enumeration of cut conjunctions in graphs thus also includes the enumeration of multiway cuts.

It will be convenient to consider the cut conjunction problem for graphs in the context of the more general cut enumeration problem for (vectorial) matroids (see e.g. [11] for a thorough introduction to the matroid theory).

**Cut Conjunctions in Matroids:** *Given a matroid  $M$  on ground set  $S$  and a set  $B \subseteq S$ , enumerate all maximal sets  $X \subseteq A = S \setminus B$  that span no element of  $B$ .*

When  $M$  is the cycle matroid of a graph  $G = (V, E \cup B)$ , where  $E \cap B = \emptyset$ , we can let  $S = E \cup B$ , and then by definition, an edge set  $Y \subseteq A = E$  spans  $b = (s_i, t_i) \in B$  if and only if  $Y$  contains an  $s_i$ - $t_i$  path. This means that a maximal edge set  $Y \subseteq E$  spans no edge  $b \in B$  if and only if  $X = E \setminus Y$  is a minimal  $B$ -cut in the graph  $(V, E)$ . Thus, in this special case the cut conjunction problem in matroids is equivalent with the cut conjunction problem in graphs.

Let  $r : S \rightarrow \mathbb{Z}_+$  be the rank function of a matroid  $M$  on  $S$ . The *dual* matroid  $M^*$  on  $S$  is defined by the rank function  $r^*(X) = r(S \setminus X) + |X| - r(S)$ , see e.g. [11]. In particular,  $Y \subseteq A = S \setminus B$  spans  $b \in B$  in  $M^*$  if and only if  $r^*(Y \cup \{b\}) = r^*(Y)$ , which is equivalent to  $r(X \cup B) = r(X \cup (B \setminus b)) + 1$ , where as before,  $X = A \setminus Y$  denotes the set complementary to  $Y$  in  $A$ . This means that the cut conjunction problem for the dual matroid  $M^*$  is equivalent to the following enumeration problem:

**Dual Formulation of Cut Conjunctions in Matroids:** *Given a matroid  $M$  on ground set  $S$  and a set  $B \subseteq S$ , enumerate all minimal sets  $X \subseteq A = S \setminus B$  such that each element  $b \in B$  is spanned by  $X \cup (B \setminus b)$ .*

In particular, when  $M$  is the cycle matroid of a graph  $G = (V, E)$  (and consequently,  $M^*$  is the cocycle matroid of  $G$ ), the dual formulation of the cut conjunction problem for matroids leads to the following enumeration problem:

**Bridge-Avoiding Extensions:** *Given an undirected graph  $G = (V, E)$ , and a collection of edges  $B \subseteq E$ , enumerate all minimal edge sets  $X \subseteq E \setminus B$  such that no edge  $b \in B$  is a bridge in  $G' = (V, B \cup X)$ .*

Let us note that in all of the mentioned problems, the output may consist of exponentially many sets, in terms of the input size. Thus, the efficiency of such enumeration algorithms customarily is measured in both the input and output sizes (see e.g., [7]). In particular, it is said that a family  $\mathcal{F}$  can be enumerated in *incremental polynomial time*, if for any subfamily  $\mathcal{F}' \subseteq \mathcal{F}$  the problem of finding  $e \in \mathcal{F} \setminus \mathcal{F}'$  or proving that  $\mathcal{F}' = \mathcal{F}$  can be solved in  $\text{poly}(n, |\mathcal{F}'|)$  time, where  $n$  denotes the input size of the problem. The enumeration problem of  $\mathcal{F}$  is called NP-hard, if deciding  $\mathcal{F}' \neq \mathcal{F}$  for subfamilies  $\mathcal{F}' \subseteq \mathcal{F}$  is NP-hard, in general. It can be shown that if the enumeration problem for  $\mathcal{F}$  is NP-hard, then no algorithm can generate all elements of  $\mathcal{F}$  in time  $\text{poly}(n, |\mathcal{F}|)$ , unless  $P=NP$ .

### 1.1 Our results

We show that all of the above enumeration problems *for graphs* can be solved efficiently, i.e. in incremental polynomial time.

**Theorem 1.** *All cut conjunctions for a given set of vertex pairs in a graph can be enumerated in incremental polynomial time.*

**Theorem 2.** *All minimal bridge-avoiding extensions for a given set of edges in a graph can be enumerated in incremental polynomial time.*

In contrast, we can recall that the more general cut conjunction problem for vectorial matroids is NP-hard:

**Proposition 1.** [4] *Let  $M$  be a vectorial matroid defined by a collection  $S$  of  $n$ -dimensional vectors over a field of characteristic zero or of large enough characteristic (at least  $8n$ ), let  $B$  be a given subset of  $S$  and let  $\mathcal{F}$  be the family of all maximal subsets of  $A = S \setminus B$  that span no vector  $b \in B$ . Given a subfamily  $\mathcal{X} \subseteq \mathcal{F}$ , it is NP-hard to decide if  $\mathcal{X} \neq \mathcal{F}$ .*

In addition to indicating that the enumeration of cut conjunctions in vectorial matroids cannot be solved in incremental (or output) polynomial time, unless  $P=NP$ , the above result also implies that the dual formulation of the cut conjunction problem for vectorial matroids is similarly NP-hard. This is because the dual of an explicitly given vectorial matroid over a field  $\mathbf{F}$  is again an explicitly given vectorial matroid over the same field (see e.g., [11]).

As stated in Proposition 1, our NP-hardness result for cut conjunctions in vectorial matroids is valid for vectorial matroids over sufficiently large fields. In particular, the complexity of enumerating cut conjunctions in *binary* matroids remains open. We can only show that this problem is tractable for  $|B| = 2$ :

**Proposition 2.** *Let  $M$  be a binary matroid on ground set  $S$  and let  $B = \{b_1, b_2\} \subseteq S$ . All maximal subsets  $X$  of  $A = S \setminus B$  which span neither  $b_1$  nor  $b_2$  can be enumerated in incremental polynomial time.*

Finally, it is worth mentioning that for an arbitrary  $B$ , the cut conjunction problem in binary matroids includes, as a special case, the well-known *hypergraph transversal* (aka *hypergraph dualization*) problem [5, 6]: *enumerate all maximal independent sets (equivalently, minimal transversals) for an explicitly given hypergraph  $\mathcal{H} \subseteq 2^V$ .* To see this inclusion, let  $B$  be the  $n \times |\mathcal{H}|$  binary matrix whose columns are the characteristic vectors of the hyperedges of  $\mathcal{H}$ , and let  $I$  be the  $n \times n$  identity matrix. Letting  $M = [I, B]$  and denoting by  $A$  the columns set of  $I$ , we can identify each maximal subset of  $A$  which spans no columns of  $B$  with a maximal independent vertex set for  $\mathcal{H}$ . This shows that listing cut conjunctions for a binary matroid is at least as hard as listing all maximal independent sets for a hypergraph. The fastest algorithm for hypergraph dualization runs in quasi-polynomial time  $\text{poly}(n) + N^{o(\log N)}$ , where  $N$  is the number of generated maximal independent sets [6, 2].

## 1.2 The $X - e + Y$ method

We prove Theorems 1, 2 by using a generic approach discussed below. Let  $E$  be a finite set and let  $\pi : 2^E \rightarrow \{0, 1\}$  be a monotone Boolean function defined on the subsets of  $E$ , i.e., for which  $\pi(X) \leq \pi(Y)$  whenever  $X \subseteq Y$ . Suppose that an efficient algorithm is available for evaluating  $\pi(X)$  in  $\text{poly}(|E|)$  time for any  $X \subseteq E$ , and that our goal is to enumerate all minimal subsets  $X \subseteq S$  for which  $\pi(X) = 1$ . In particular, the enumeration of cut conjunctions (CC) and bridge avoiding subsets (BA) can all be embedded into this general scheme by letting  $\pi_{CC}(X) = 1$  iff vertex  $s_i$  is disconnected from  $t_i$  in  $(V, E \setminus X)$  for all  $i = 1, \dots, k$  and  $\pi_{BA}(X) = 1$  iff no  $b \in B$  is a bridge in  $(V, B \cup X)$ .

Returning to the general scheme, given a monotone Boolean function  $\pi$ , let  $\mathcal{F} = \{X \mid X \subseteq E \text{ is a minimal set satisfying } \pi(X) = 1\}$ , and let  $\mathcal{G} = (\mathcal{F}, \mathcal{E})$  be the directed “supergraph” with the vertex set  $\mathcal{F}$  in which two vertices  $X, X' \in \mathcal{F}$  are connected by an arc  $(X, X') \in \mathcal{E}$  if and only if  $X'$  can be obtained from  $X$  by the following process:

- (p1) Delete an element  $e$  from  $X$  ( $\pi(X \setminus e) = 0$  by the minimality of  $X$ ).
- (p2) Add an minimal set  $Y \subseteq E \setminus X$  to restore the property  $\pi((X \setminus e) \cup Y) = 1$ .
- (p3) Assuming an linear order on the elements of  $E$ , delete the lexicographically first minimal set  $Z \subseteq X \setminus e$  for which  $X' = (X \setminus (Z \cup e)) \cup Y \in \mathcal{F}$ .

Note that in step (p3) we use an arbitrary fixed linear order on  $E$ , and that the lexicographic minimization performed in that step can be done in time polynomial in  $|E|$  because we assume that evaluating  $\pi(\cdot)$  takes  $\text{poly}(|E|)$  time. Note also that in step (p3) we can always find a subset  $Z \subseteq X \setminus e$  for which  $X' = (X \setminus (Z \cup e)) \cup Y$  belongs to  $\mathcal{F}$ , due to the minimality of  $Y$  chosen in step (p2). Since the out-neighborhood of vertex  $X$  in  $\mathcal{G}$  is obtained by deleting all elements  $e \in X$  in step (p1), and by considering all possible minimal sets  $Y$  in step (p2), the resulting supergraph  $\mathcal{G} = (\mathcal{F}, \mathcal{E})$  is always strongly connected.

**Proposition 3.** *For any monotone Boolean function  $\pi : 2^E \rightarrow \{0, 1\}$ , and any linear ordering of  $E$ , the supergraph  $\mathcal{G} = (\mathcal{F}, \mathcal{E})$  is strongly connected.*

*Proof.* Let  $X, X' \in \mathcal{F}$  be two vertices of  $\mathcal{G}$ . We show by induction on  $|X \setminus X'|$  that  $\mathcal{G}$  contains a directed path from  $X$  to  $X'$ . If  $X \setminus X' = \emptyset$  then  $X \subseteq X'$ , but since  $X'$  is minimal,  $X = X'$  must follow. Suppose that  $|X \setminus X'| > 0$  and let us show that there is an out-neighbor  $X''$  of  $X$  such that  $|X'' \setminus X'|$  is smaller than  $|X \setminus X'|$ . Choose an arbitrary  $e \in X \setminus X'$ . Since  $(X \setminus e) \cup X'$  contains  $X'$  and  $\pi(X') = 1$ , we have  $\pi((X \setminus e) \cup X') = 1$  by the monotonicity of  $\pi$ , and hence there is a minimal nonempty set  $Y \subseteq X' \setminus X$  such that  $\pi((X \setminus e) \cup Y) = 1$ . Now we can lexicographically delete some elements  $Z$  of  $X \setminus e$  and obtain an out-neighbor  $X'' = (X \setminus (Z \cup e)) \cup Y \in \mathcal{F}$  for which  $|X'' \setminus X'| \leq |X \setminus (X' \cup e)| < |X \setminus X'|$ . Such a set  $Z$  exists because we have  $\pi((X \setminus e) \cup (Y \setminus y)) = 0$  for all  $y \in Y$  by the minimality of  $Y$ , and thus any minimal set  $\tilde{X} \in \mathcal{F}$  contained in  $(X \setminus e) \cup Y$  must contain  $Y$ .  $\square$

Let us remark that the number of minimal sets  $Y$  in **(p2)** may be exponential. For a given set  $X \in \mathcal{F}$  and element  $e \in X$ , any two distinct minimal sets,  $Y$  and  $Y'$ , corresponding to  $X$  and  $e$ , produce different neighbors of  $X$  in  $\mathcal{G}$ . Hence, every neighbor of  $X$  in  $\mathcal{G}$  may be generated at most  $|X|$  times.

Since the supergraph  $\mathcal{G} = (\mathcal{F}, \mathcal{E})$  is always strongly connected, we can generate  $\mathcal{F}$  by first computing an initial vertex  $X^o \in \mathcal{F}$  and then performing a traversal (say, breadth-first search) of  $\mathcal{G}$ . Given our assumption that  $\pi$  can be evaluated in  $\text{poly}(|E|)$  time, computing an initial vertex of  $\mathcal{G}$  can be done in polynomial time. Steps **(p1)** and **(p3)** can also be performed in  $\text{poly}(|E|)$  time. Hence we can conclude that the enumeration problem for  $\mathcal{F}$  reduces to the enumeration of sets  $Y$  in step **(p2)**. In particular, due to the above remark we get the following statement:

**Proposition 4.** *All elements of  $\mathcal{F}$  can be enumerated in incremental polynomial time whenever the enumeration problem **(p2)** can be done in incremental polynomial time.*  $\square$

As an illustration for the  $X - e + Y$  method, consider the *path conjunction problem [3]*: given an undirected graph  $G = (V, E)$  and a collection of  $k$  vertex pairs  $s_i, t_i \in V$ , enumerate all minimal edge sets  $X \subseteq E$  such that for all  $i = 1, \dots, k$ , vertices  $s_i$  and  $t_i$  are connected in  $(V, X)$ .

Let  $\pi(X) = 1$  if and only if every  $s_i$  is connected to  $t_i$  in  $(V, X)$ . Then the minimal edge sets  $X$  for which  $\pi(X) = 1$  are exactly the minimal path conjunctions. Furthermore, any such minimal path conjunction is a collection of trees  $T_1, \dots, T_l$  such that each vertex pair  $(s_i, t_i)$  belongs to a common tree  $T_j$ . Removing an edge  $e$  from  $X$  splits one of the trees into two sub-trees  $T'_j, T''_j$  such that there is at least one pair  $(s_i, t_i)$  with one vertex belonging to  $T'_j$  and the other to  $T''_j$ . Let  $G'$  be the graph obtained from  $G$  by contracting each tree of  $T_1, \dots, T'_j, T''_j, \dots, T_l$  into a vertex, and let  $u$  and  $v$  denote the vertices corresponding to  $T'_j$  and  $T''_j$ . A minimal edge set  $Y$  restores the property that all  $s_i$  and  $t_i$  are connected in  $(V, (X \setminus e) \cup Y)$  if and only if  $Y$  is a simple path from  $u$

to  $v$  in  $G'$ . Hence the  $X - e + Y$  method reduces the path conjunction enumeration problem to the enumeration of all  $u$ - $v$  paths in  $G'$ , which can be done via backtracking [8] incrementally efficiently. Thus by Proposition 4 enumerating all minimal path conjunctions can be done in incremental polynomial time [3].

Our proofs of Theorems 1, 2 follow this approach using the two monotone Boolean functions  $\pi_{CC}$ ,  $\pi_{BA}$  defined at the beginning of this section.

The remainder of the paper is organized as follows. We prove Theorems 1, 2 in Section 2, 3, respectively. For a full version with all related results and proofs see [1].

## 2 Proof of Theorem 1

In this section we apply the  $X - e + Y$  method to the Boolean function  $\pi_{CC}$  in order to enumerate all cut conjunctions in graphs. First, in Subsection 2.1 we state a characterization of cut conjunctions in graphs. In Subsection 2.2 we reduce the problem of enumerating all minimal sets  $Y$  in **(p2)** to the enumeration of all cut conjunctions in a graph of a simpler structure. In Subsection 2.3 we show that the latter problem can be solved efficiently by using a variant of the supergraph approach.

In this section we use the following notation. Let  $G = (V, E)$  be a graph, let  $U$  be a subset of its vertices, let  $F$  be a subset of its edges, and let  $G' = (V', E')$  and  $G'' = (V'', E'')$  denote subgraphs of  $G$  (i.e.,  $V', V'' \subseteq V$  and  $E', E'' \subseteq E$ ). We denote by  $G[U]$  a subgraph of  $G$  induced on the vertex set  $U$ . Then  $G - U = G[V \setminus U]$  is a graph obtained from  $G$  by deleting all the vertices of  $U$  and their incident edges,  $G - F = (V, E \setminus F)$  is obtained by deleting all the edges of  $F$  from  $E$  and  $G - G' = G - V'$ . We also define  $G + U = (V \cup U, E)$ ,  $G + F = (V, E \cup F)$ , and  $G' + G'' = (V' \cup V'', E' \cup E'')$ .

### 2.1 Characterization of Minimal Cut Conjunctions in Graphs

Let  $G = (V, E)$  be an undirected connected graph. It will be convenient to define a *cut* to be a set of edges  $E(G_1, \dots, G_l) = \bigcup_{i \neq j} \{uv \in E : u \in G_i, v \in G_j\}$  where  $G_1, \dots, G_l$  are induced subgraphs of  $G$  such that their vertex sets partition  $V$ , and  $G_i$  is connected for each  $i = 1, \dots, l$ . Let  $B = \{(s_1, t_1), \dots, (s_k, t_k)\}$  be a set of distinct source-sink pairs of  $G$ . A *B-cut* is a cut  $E(G_1, \dots, G_l)$  such that, for each  $i$ ,  $s_i$  and  $t_i$  do not belong to the same  $G_j$ . If the set  $B$  is clear from the context we shall call the minimal  $B$ -cut a *cut conjunction*. The following characterization of cut conjunctions follows directly from their definition.

**Proposition 5.** *Let  $E(G_1, G_2, \dots, G_l)$  be a  $B$ -cut. Then,  $E(G_1, G_2, \dots, G_l)$  is a minimal  $B$ -cut if and only if for every  $x, y \in \{1, \dots, l\}$  with  $x \neq y$ , if there is an edge of  $G$  between  $G_x$  and  $G_y$  then there must exist a source-sink pair  $(s_i, t_i)$  with exactly one vertex in  $G_x$  and the other in  $G_y$ .*

## 2.2 Reduction

In this section we reduce the problem of generating all minimal sets  $Y$  in **(p2)** to generating all cut conjunctions in a graph of a simpler structure.

Let  $F$  be a subset of edges of  $G$  and let  $(s_i, t_i) \in B$ . Suppose that  $s_i$  and  $t_i$  are in the same component of  $G - F$ . Then we say that  $(s_i, t_i)$  is *F-conflicting*. Let  $X = E(G_1, G_2, \dots, G_l)$  be a minimal  $B$ -cut of  $G$  and let  $b \in X$ . The removing  $b$  from  $X$  reconnects some two components,  $G_x$  and  $G_y$ , of  $G - X$ , where one endpoint of  $b$  is in  $G_x$  and the other in  $G_y$ . Thus  $G - (X \setminus b)$  contains at least one  $(X \setminus b)$ -conflicting pair. Hence generating all minimal sets  $Y \subseteq E \setminus X$  which restore the property that no  $s_i$  is connected to  $t_i$ , is equivalent to generating all minimal  $B'$ -cuts in the graph  $G_x + G_y + b$  where  $B'$  is the set of  $(X \setminus b)$ -conflicting pairs. Let  $L = G_x$  and  $R = G_y$ . We can always relabel the  $(X \setminus b)$ -conflicting pairs to guarantee that the conflicting  $s_i$ 's are in  $L$  and the conflicting  $t_i$ 's are in  $R$ . We denote the resulting graph by  $H(X, b)$ . Note that we have reduced our enumeration problem to listing all cut conjunctions in  $H(X, b)$ .

## 2.3 Enumerating Minimal Cut Conjunctions in $H(X, b)$

Let  $H = H(X, b) = (V, E)$  be the graph defined at the end of Section 2.2, that is:  $H = L + R + b$ , where  $b = v_L v_R$  is a bridge,  $L$  contains the sources  $s_1, \dots, s_k$ , and  $R$  contains the sinks  $t_1, \dots, t_k$ . Let  $B = \{(s_1, t_1), \dots, (s_k, t_k)\}$  and let  $K = E(G_1, \dots, G_l)$  be a cut conjunction of  $H$ , such that  $K \neq \{b\}$ . Without loss of generality, assume that  $b$  is in  $G_1$ . Note that every other  $G_j$  is contained either in  $L$  or in  $R$  (since  $G_j$  is connected and all paths from  $L$  to  $R$  go through  $b$ ). We denote by  $M = G_1$  the component containing  $b$  and call it the *root component of  $K$* . The other components will be called *leaf components of  $K$* . Denote the  $G_j$ 's contained in  $L$  by  $L_1, \dots, L_m$  and those in  $R$  by  $R_1, \dots, R_n$ . Observe that all edges of  $K = E(M, L_1, \dots, L_m, R_1, \dots, R_n)$  lie between the root and leaf components. Hence  $M$  uniquely determines the leaf components of  $K$ . Now we define the digraph  $\mathcal{H}$ , the *supergraph of cut conjunctions of  $H$* . The vertex set of  $\mathcal{H}$  is the family of all cut conjunctions of  $H$  other than  $\{b\}$ . For each cut conjunction  $K = E(M, L_1, \dots, L_m, R_1, \dots, R_n)$  of  $H$  we define its out-neighborhood to consist of all cut conjunctions which can be obtained from  $K$  by the following sequence of steps:

- (q1) Choose a vertex  $v$  incident to  $e \in K$  such that  $v \notin \{v_L, v_R\}$ . Depending on  $v$  we have the following three cases.
- (q2-a) Suppose  $v \in R_j$  is in a leaf component of  $K$  and  $M + v + e$  does not contain a source-sink pair  $(s_i, t_i)$ . Thus either  $v$  is not a sink, or  $v = t_i$  and  $s_i \notin M$ . Let  $W_1, \dots, W_p$  be the components of  $R_j - v$ , and let  $M_D = M + v + \bigcup_{u \in M} \{uv \in E\}$ . Then  $D = E(M_D, L_1, \dots, L_m, R_1, \dots, R_{j-1}, W_1, \dots, W_p, R_{j+1}, \dots, R_n)$  is a  $B$ -cut. Note that we have moved  $v$  from  $R_j$  to  $M$ . Removing  $v$  from  $R_j$  splits  $R_j$  into components  $W_1, \dots, W_p$ .
- (q2-b) Suppose  $v \in R_j$  is in a leaf component of  $K$  and  $M + v + e$  contains a source-sink pair  $(s_i, t_i)$ . Thus  $v = t_i$ ,  $s_i \in M$  and  $v_L \neq s_i$ . Let  $W_1, \dots, W_p$

be the components of  $R_j - t_i$  and let  $U_1, \dots, U_r$  be the components of  $M - s_i$  not containing  $b$ . Denote  $M_D = (M + t_i + \bigcup_{u \in M} \{ut_i \in E\}) - (s_i + U_1 + \dots + U_r)$ . Then  $D = E(M_D, L_1, \dots, L_m, s_i, U_1, \dots, U_r, R_1, \dots, R_{j-1}, W_1, \dots, W_p, R_{j+1}, \dots, R_n)$  is a  $B$ -cut. Note that we have moved  $t_i$  from  $R_j$  to  $M$ . To restore the property that no  $s_i$  is connected to  $t_i$ , we have removed  $s_i$  from  $M$ . Removing  $v$  from  $R_j$  splits  $R_j$  into components  $W_1, \dots, W_p$ , and removing  $s_i$  from  $M$  splits  $M$  into components  $U_1, \dots, U_r$  and  $M_D$ , the component containing  $b$ .

- (q2-c) Suppose  $v \in M - \{v_L, v_R\}$  and  $v$  is adjacent to  $L_j$ . Note that  $v \notin \{t_1, \dots, t_k\}$ . Let  $U_1, \dots, U_r$  be the components of  $M - v$  not containing  $b$ , and let  $M_D = M - (v + U_1 + \dots + U_r)$ . Then  $D = E(M_D, L_1, \dots, L_{j-1}, L_j + v + \bigcup_{u \in L_j} uv \in E, L_{j+1}, \dots, L_m, U_1, \dots, U_r, R_1, \dots, R_n)$  is a  $B$ -cut. Note that we have moved  $v$  from  $M$  to  $L_j$  splitting  $M$  into components  $U_1, \dots, U_r$  and  $M_D$ .
- (q3) Let  $D = E(G_1, \dots, G_l)$  be the  $B$ -cut obtained in the previous step. Choose the lexicographically first two sets  $G_x$  and  $G_y$  such that there is an edge  $e \in D$  connecting  $G_x$  and  $G_y$  and there is no  $(D \setminus e)$ -conflicting pair. Replace  $G_x$  and  $G_y$  in  $D$  by  $G_x + G_y$ . Repeat until no such edge exists, thus the  $B$ -cut is minimal. Let  $K' = E(M', L'_1, \dots, L'_{m'}, R'_1, \dots, R'_{n'})$  be the resulting cut conjunction. Then  $K'$  is a neighbor of  $K$  in  $\mathcal{H}$ .

**Proposition 6.** *The supergraph  $\mathcal{H}$  is strongly connected.*

Proposition 6 follows immediately from Lemma 1 and Lemma 2.

Let  $K_1, K_3$  be cut conjunctions and let  $M_1, M_3$  be their root components. We call the vertices of  $M_3$  *blue vertices*, and all other vertices *green vertices*. Let  $\mathcal{K}$  be an induced subgraph of  $\mathcal{H}$ , whose vertices are the cut conjunctions with root components containing all the blue vertices.

**Lemma 1.** *There exists a cut conjunction  $K_2 \in \mathcal{K}$  such that there is a path from  $K_1$  to  $K_2$  in  $\mathcal{H}$ .*

*Proof.* Let  $T$  be an arbitrary spanning tree of  $M_3$  containing the bridge  $b$ . For a  $B$ -cut  $D$  of  $H$  with  $M$  as its root component, we partition the edges of  $T$  into two groups. We call an edge  $e$  of  $T$  a  *$D$ -solid edge*, if  $e \in M$  and  $e$  is reachable from  $b$  by using only edges of  $T$  that are in  $M$ . Otherwise  $e$  is called a  *$D$ -dashed edge*. Note that  $b$  is  $D$ -solid edge. We denote the set of  $D$ -solid edges by  $S_D$  and the set of  $D$ -dashed edges by  $D_D$ . Clearly,  $|S_D| + |D_D| = |T|$ .

Let  $K_1 = E(M_1, L_1, \dots, L_m, R_1, \dots, R_n)$ . We will show by induction on  $|S_{K_1}|$  that there is a path from  $K_1$  to  $K_2$ . If  $|S_{K_1}| = |T|$ , then  $K_1 \in \mathcal{K}$ . If  $|S_{K_1}| < |T|$ , then there exists a  $K_1$ -dashed edge  $vw$  between two blue vertices  $v$  and  $w$  such that  $v$  is in a leaf component of  $K_1$ ,  $w \in M_1$  and  $w$  is incident to a  $K_1$ -solid edge. Suppose that  $v \in R_j$ . We now show that  $K'_1$ , a neighbor of  $K_1$ , obtained by moving  $v$  from the leaf to the root component, has  $|S_{K'_1}| \geq |S_{K_1}| + 1$ .

**Case 1:**  $v$  is not a sink or  $v = t_i$  and  $s_i \notin M_1$ . Let  $D$  be a  $B$ -cut obtained in (q2-a) and  $M_D = M_1 + v$  be its root component. Since  $M_D$  contains both  $v$

and  $w, vw$  is a  $D$ -solid edge, so  $|S_D| = |S_{K_1}| + 1$ . In **(q3)**  $M_D$  can only merge with leaf components, hence  $|S_{K'_1}| \geq |S_D|$ .

**Case 2:**  $v = t_i, s_i \in M$ . Note that  $t_i$  is a blue vertex, so  $s_i$  must be green, since  $M_3$  does not contain any source-sink pair, and in particular  $s_i$  cannot be an endpoint of  $b$ . Let  $D$  be a  $B$ -cut obtained in **(q2-b)** and  $M_D$  be its root component. Recall that  $M_D = (M_1 + t_i) - (s_i + U_1 + \dots + U_r)$ , where  $U_1, \dots, U_r$  are the components of  $M - s_i$  not containing  $b$ . Observe that in **(q2-b)** we did not remove any  $K_1$ -solid edge from  $M_1$ . Since  $s_i$  is a green vertex, all edges incident to  $s_i$  do not belong to  $T$ . Edges in  $U_1, \dots, U_r$  and incident to these components are also not  $K_1$ -solid, because all paths from  $b$  to  $U_1, \dots, U_r$ , which use edges of  $T$  that are in  $M_1$ , must go through  $s_i$ . Thus  $|S_D| = |S_{K_1}| + 1$ . In **(q3)**  $M_D$  can only merge with leaf components, hence  $|S_{K'_1}| \geq |S_D|$ .  $\square$

**Lemma 2.** *For every  $K_2 \in \mathcal{K}$  there is a path from  $K_2$  to  $K_3$  in  $\mathcal{K}$ .*

Since  $\mathcal{H}$  is strongly connected and finding an initial vertex of  $\mathcal{H}$  is easy, we can enumerate all sets  $Y$  in **(p2)** in incremental polynomial time.

### 3 Proof of Theorem 2

In this section we assume that  $G = (V, E)$  is a multigraph. Given  $B \subseteq E$ , let  $\mathcal{F} = \text{minimal } \{X \subseteq E \setminus B \mid \text{no } b \in B \text{ is a bridge of } (V, B \cup X)\}$ . We enumerate  $\mathcal{F}$  by using the  $X - e + Y$  method stated in Section 1.2. Proposition 7 below implies that this can be accomplished in incremental polynomial time.

**Proposition 7.** *Given a set  $X \in \mathcal{F}$  and an edge  $e \in X$ , all sets  $Y$  in **(p2)** can be enumerated with polynomial delay.*

*Proof.* Let  $B' = \{b_1, \dots, b_k\}$  be the subset of edges of  $B$  that are bridges in  $(V, B \cup (X \setminus e))$ . First observe that for each edge  $b_i \in B'$  there is a cycle  $C_i$  in  $(V, B \cup X)$  containing  $e$  and  $b_i$ . Suppose  $b_i \in C_i \setminus C_j$  for some  $i, j \in \{1, \dots, k\}$ . Then there is a cycle  $C' \subseteq (C_i \cup C_j) \setminus e$  such that  $b_i \in C'$ .  $C'$  is also the cycle in  $(V, B \cup (X \setminus e))$ . This would contradict the definition of  $B'$ . Hence all edges of  $B'$  lie on a common cycle in  $(V, B \cup X)$  containing  $e$ , and accordingly, all edges of  $B'$  belong to a common path in  $(V, B \cup (X \setminus e))$ .

Let  $G' = (V', E')$  be the multigraph obtained from  $(V, E \setminus e)$  by contracting all edges in  $(B \setminus B') \cup (X \setminus e)$ . Then  $B'$  is a path in  $G'$ . Furthermore, the enumeration of all sets  $Y$  in **(p2)** now reduces to the enumeration of all minimal edge sets  $Y'$  of  $G'$  for which no edge  $b$  on the path  $B'$  is a bridge in  $(V', B' \cup Y')$ . Thus the enumeration of cut conjunctions in cocycle matroids reduces to the special case of the same problem for multigraphs in which  $B$  is a path.

Now we argue that the latter problem is equivalent to the enumeration of all directed paths between a pair of vertices in some explicitly given directed multigraph. To see this, denote by  $u_1, \dots, u_{k+1}$  the  $k + 1$  vertices on the path  $B' = \{b_1, \dots, b_k\}$  in  $G'$ , and assume that  $b_i = u_i u_{i+1}$  for  $i = 1, \dots, k$ . If no edge  $b \in B'$  is a bridge in  $(V', B' \cup Y')$ , then for each  $i = 1, \dots, k$  there must exist

a path  $P \subseteq Y'$  such that  $P$  and  $B'$  are edge disjoint and the vertex set of  $P$  contains exactly two vertices  $u_\alpha, u_\beta$  of  $B'$  such that  $\alpha \leq i$  and  $\beta \geq i + 1$ . Thus  $Y' = P_1 \cup \dots \cup P_s$ , for some paths  $P_1, \dots, P_s$  satisfying conditions above, where no two distinct paths in  $Y'$  have a common vertex outside of  $B'$ . Denoting by  $u_{\alpha_i}$  and  $u_{\beta_i}$  the intersection of the vertex set of  $P_i$  with  $B'$ , we conclude  $u_1 = \alpha_1 < \alpha_2 \leq \beta_1 < \alpha_3 \leq \beta_2 < \alpha_4 \leq \dots < \alpha_s \leq \beta_{s-1} < \beta_s = u_{k+1}$ . Let us now consider the directed multigraph  $\vec{G}' = (V', \vec{E}')$  obtained from the multigraph  $G' = (V', E')$  by replacing the undirected path  $B'$  by the directed path  $\vec{B}' = u_1 \leftarrow u_2 \leftarrow \dots \leftarrow u_{k+1}$  and by adding two opposite arcs  $u \rightarrow v$  and  $v \rightarrow u$  for each edge  $uv \in E' \setminus B'$ . From the above discussion it follows that there exists a one to one correspondence between all minimal sets  $Y'$  and all minimal directed paths from  $u_1$  to  $u_{k+1}$  in  $\vec{G}'$ . Since it is well known that all minimal directed paths between a given pair of vertices can be enumerated via backtracking [8] with polynomial delay, Proposition 7 follows.  $\square$

## References

1. E. Boros, K. Borys, K. Elbassioni, V. Gurvich, L. Khachiyan, and K. Makino. Enumerating cut conjunctions in graphs and related problems. Rutcor Research Report RRR 19-2005, Rutgers University.
2. E. Boros, K. Elbassioni, V. Gurvich, and L. Khachiyan. An efficient implementation of a joint generation algorithm, experimental and efficient algorithms. In C. C. Ribeiro and S. L. Martins, editors, *Proceedings of the Third International Workshop, WEA*, volume 3059 of *Lecture Notes in Computer Science*, pages 114–128, Angra Dos Reis, Brazil, May 25-28 2004. Springer Verlag, Berlin, Heidelberg, New York.
3. E. Boros, K. Elbassioni, V. Gurvich, L. Khachiyan, and K. Makino. Generating paths and cuts in multi-pole (di)graphs. In J. Fiala, V. Koubek, and J. Kratochvil, editors, *Mathematical Foundations of Computer Science MFCS*, volume 3153 of *Lecture Notes in Computer Science*, pages 298–309, Prague, Czech Republic, August 22-27 2004. Springer Verlag, Berlin, Heidelberg, New York.
4. E. Boros, K. Elbassioni, V. Gurvich, L. Khachiyan, and K. Makino. On the complexity of some enumeration problems for matroids. Manuscript, 2005.
5. T. Eiter and G. Gottlob. Identifying the minimal transversals of a hypergraph and related problems. *SIAM Journal on Computing*, 24:1278–1304, 1995.
6. M. Fredman and L. Khachiyan. On the complexity of dualization of monotone disjunctive normal forms. *Journal of Algorithms*, 21:618–628, 1996.
7. E. Lawler, J. K. Lenstra, and A. H. G. Rinnooy Kan. Generating all maximal independent sets: NP-hardness and polynomial-time algorithms. *SIAM Journal on Computing*, 9:558–565, 1980.
8. R. C. Read and R. E. Tarjan. Bounds on backtrack algorithms for listing cycles, paths, and spanning trees. *Networks*, 5:237–252, 1975.
9. 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 Association for Computing Machinery*, 27:619–632, 1980.
10. V. Vazirani. *Approximation Algorithms*. Springer Verlag, Berlin, Heidelberg, New York, 2001.
11. D.J.A. Welsh. *Matroid Theory*. Academic Press, London, 1976.