DIMACS Technical Report 2005-24
June 2005

Enumerating Cut Conjunctions in Graphs and Related Problems [1,2]

by

Leonid Khachiyan
Department of Computer Science, Rutgers University

Endre Boros[3]
RUTCOR, Rutgers University

Konrad Borys[4]
RUTCOR, Rutgers University Piscataway NJ 08854

Khaled Elbassioni
Max-Planck-Institut für Informatik
Saarbrücken, Germany

Vladimir Gurvich [5]
RUTCOR, Rutgers University

Kazuhisa Makino
Division of Mathematical Science for Social Systems, Osaka University
Toyonaka, Osaka, Japan

[1]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.

[2]Our friend and colleague, Leo Khachiyan passed away with tragic suddenness while we were preparing this manuscript.

[3]Permanent Member

[4]Affiliated Graduate Student Member

[5]Permanent Member

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 [TSOA80], where n and m are the numbers of vertices and edges in the input graph, and N is the total number of cuts. In this paper, we study the following natural extension of the two-terminal cut enumeration problem:

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

Note that for $i \neq j$, s_i and s_j or s_i and t_j or t_i and t_j may coincide. 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 $i = 1, \ldots, k$, justifying the naming of these edge sets. Note also that not all conjunctions of minimal s_i-t_i cuts for $i = 1, \ldots, k$ are minimal B-cuts. Figure 1 depicts a graph with the number of minimal s_k-t_k cuts not polynomially bounded by $|V|$ and $|\mathcal{F}_{G,B}|$, showing that the enumeration of cut conjunctions cannot efficiently be reduced to two-terminal cut enumeration.

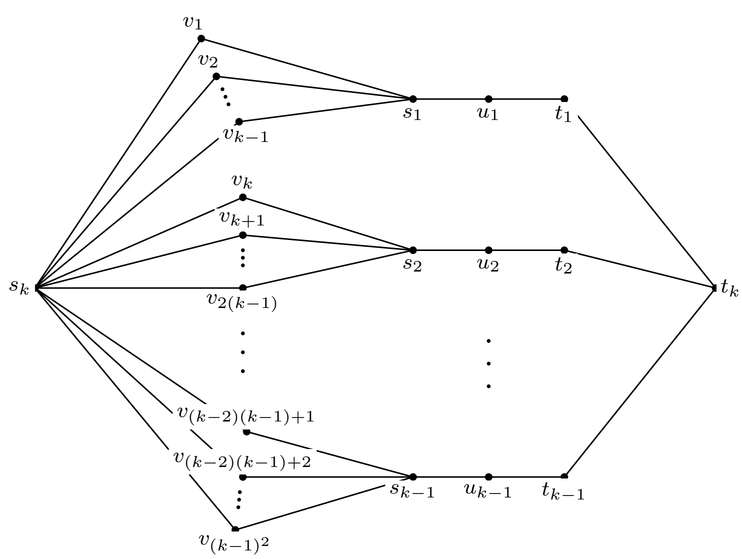

Figure 1: Minimal B-cuts contain exactly one edge of each pair s_iu_i and u_it_i, for $i = 1, \ldots, k-1$, thus we have $|\mathcal{F}_{G,B}| = 2^{k-1}$. While the number of minimal s_k-t_k cuts is more than $2^{(k-1)^2}$, i.e. it is not polynomially bounded by $|V| = k^2 + k$ and $|\mathcal{F}_{G,B}|$.

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. [Vaz01]. 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. In what follows we assume familiarity with matroid theory (see e.g. [Wel76] for a thorough introduction).

> **Cut Conjunctions in Matroids**: *Given a matroid M on ground set S and a set $B \subseteq S$, enumerate all maximal sets $X \subseteq A \overset{\text{def}}{=} S \setminus B$ 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 $S \overset{\text{def}}{=} E \cup B$, 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 $r : S \to \mathbb{Z}_+$ be the rank function of a matroid M on S (where $\mathbb{Z}_+$ denotes the set of nonnegative integers). The *dual* matroid M^* on S is defined by the rank function $r^*(X) = r(S \setminus X) + |X| - r(S)$, see e.g. [Wel76]. 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 complimentary 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 $X \subseteq A \overset{\text{def}}{=} S \setminus B$ 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 cocyle 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., [LLK80]). 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 $poly(n, |\mathcal{F}'|)$ 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 $poly(n, |\mathcal{F}|)$, 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 *[BEG+05] 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 $A \stackrel{\text{def}}{=} S \setminus B$ 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., [Sch03]).

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 $A \stackrel{\text{def}}{=} S \setminus B$ 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 [EG95, FK96]:

> *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 readily 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 theoretically fastest currently available algorithm for hypergraph dualization runs in quasi-polynomial time $poly(n) + N^{o(\log N)}$, where N is the sum of $|\mathcal{H}|$ and the number of generated maximal independent sets [FK96].

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 $\pi : 2^E \to \{0, 1\}$ 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 $poly(|E|)$ time for any $X \subseteq E$, and that our goal is to enumerate all (inclusionwise) minimal subsets $X \subseteq S$ for which $\pi(X) = 1$. In particular, the enumeration of cut conjunctions and bridge avoiding subsets can all be embedded into this general scheme by letting

$$\pi_{CC}(X) = 1 \iff \begin{array}{c}\text{vertex } s_i \text{ is disconnected from } t_i \\ \text{in } (V, E \setminus X) \text{ for all } i = 1, \ldots, k,\end{array}$$

and

$$\pi_{BA}(X) = 1 \iff \text{no } b \in B \text{ is a bridge in } (V, B \cup X).$$

Returning to the general scheme, given a monotone Boolean function π, let

$$\mathcal{F} \stackrel{\text{def}}{=} \{X \mid X \subseteq E \text{ is a minimal set satisfying } \pi(X) = 1\},$$

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 (since X is an minimal set satisfying $\pi(X) = 1$, this implies $\pi(X \setminus e) = 0$).

(p2) Add a minimal set $Y \subseteq E \setminus X$ to restore the underlying property $\pi((X \setminus e) \cup Y) = 1$.

(p3) Assuming a fixed linear order on the elements of E, delete the lexicographically first minimal set $Z \subseteq X \setminus e$ to restore the minimality of $X' = (X \setminus (Z \cup e)) \cup Y$ with respect to $\pi(X') = 1$.

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 $poly(|E|)$ 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 possible elements $e \in X$ in step (**p1**), and by considering all possible minimal sets Y in step (**p2**), it can be shown that the resulting supergraph $\mathcal{G} = (\mathcal{F}, \mathcal{E})$ is always strongly connected.

Proposition 3 *For any monotone Boolean function $\pi : 2^E \to \{0, 1\}$, 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 $|X \setminus X'|$. For this, let us choose an arbitrary element $e \in X \setminus X'$. 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 π, 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. □

Remark 1 *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 $\mathcal{G}$. For a given X, two elements of X, e and e' can produce the same neighbor in $\mathcal{G}$. Hence, every neighbor of X in $\mathcal{G}$ can 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 $\pi(\cdot)$ can be evaluated in $poly(|E|)$ time, computing an initial vertex of $\mathcal{G}$ can be done in polynomial time. Steps (**p1**) and (**p3**) can also be performed in $poly(|E|)$ 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.* □

As an illustration for the $X - e + Y$ method, consider the following *path conjunction problem* [BEG+04]:

> *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 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 to be listed in the above problem. Furthermore, it is easy to see that 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 $(V, (X \smallsetminus e) \cup Y)$ 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 [RT75] incrementally efficiently. Thus by Proposition 4 enumerating all minimal path conjunctions can be done in incremental polynomial time [BEG+04].

Our proofs of Theorems 1, 2 follow this approach using the two monotone Boolean functions π_{CC}, π_{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. Then in Section 4 we show the NP-hardness of enumerating cut conjunctions in vectorial matroids, as stated in Proposition 1. Finally, in Section 5 we also prove Proposition 2.

2 Proof of Theorem 1

In this section we apply the $X - e + Y$ method to the Boolean function π_{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 $G - U \stackrel{\text{def}}{=} G[V \setminus U]$ is a graph obtained from G by deleting all the vertices of U and their incident edges, $G - F \stackrel{\text{def}}{=} (V, E \setminus F)$ is obtained by deleting all the edges of F from E and $G - G' \stackrel{\text{def}}{=} G - V'$. We also define $G + U \stackrel{\text{def}}{=} (V \cup U, E)$, $G + F \stackrel{\text{def}}{=} (V, E \cup F)$, and $G' + G'' \stackrel{\text{def}}{=} (V' \cup V'', E' \cup E'')$.

2.1 Characterization of Minimal Cut Conjunctions in Graphs

Let G be an undirected connected graph with vertex set V and edge set E. It will be convenient to define a *cut* to be a set of edges $E(G_1, \ldots, G_l) = \bigcup_{i \neq j} \{uv \in E : u \in G_i, v \in G_j\}$ where $G_1, \ldots, G_l$ are induced subgraphs of G such that their vertex sets partition V, and G_i is connected for each $i = 1, \ldots, l$.

Let $B = \{(s_1, t_1), \ldots, (s_k, t_k)\}$ be a set of distinct source-sink pairs of G. A *B-cut* is a cut $E(G_1, \ldots, G_l)$ 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.

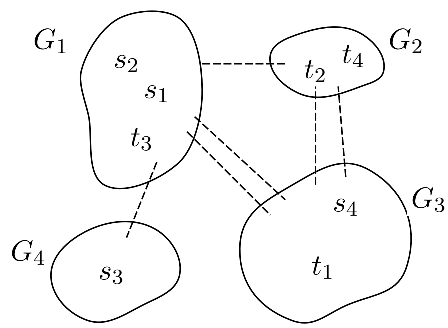

Figure 2: Minimal B-cut $E(G_1, G_2, G_3, G_4)$. The dashed lines are the edges of the B-cut.

Proposition 5 *Let $E(G_1, G_2, \ldots, G_l)$ be a B-cut. Then, $E(G_1, G_2, \ldots, G_l)$ is a minimal B-cut if and only if for every $x, y \in \{1, \ldots, l\}$ 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 (see Figure 2).*

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 $X = E(G_1, G_2, \ldots, G_l)$ 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 (see Figure 3). 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.

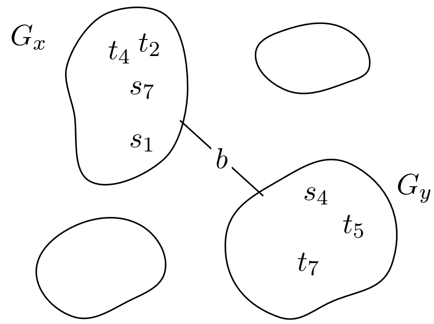

Figure 3: Graph $G - (X \setminus b)$ contains two $(X \setminus b)$-conflicting pairs (s_4, t_4) and (s_7, t_7).

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 $H(X, b)$ (see Figure 4). Note that we have reduced our enumeration problem to listing all cut conjunctions in $H(X, b)$. As we discuss in the next section, the latter problem can be efficiently solved by traversing an appropriately defined supergraph of cut conjunctions of $H(X, b)$.

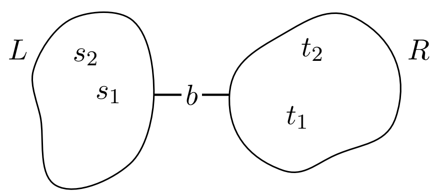

Figure 4: Graph $H(X, b)$ with all sources in L and sinks in R.

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:

- $H = L + R + b$,
- $b = v_L v_R$ is a bridge (note that v_L can be a source and v_R can be a sink, but $b \neq s_i t_i$ for all i),
- L contains the sources $s_1, \ldots, s_k$, and

- R contains the sinks $t_1, \ldots, t_k$ (see Figure 4).

Let $B = \{(s_1, t_1), \ldots, (s_k, t_k)\}$ and let $K = E(G_1, \ldots, G_l)$ 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 $L_1, \ldots, L_m$ and those in R by $R_1, \ldots, R_n$ (see Figure 4).

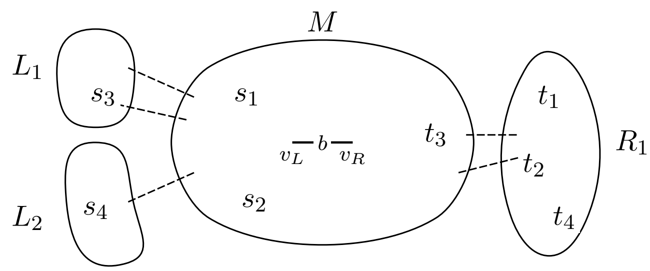

Figure 5: Minimal B-cut $E(M, L_1, L_2, R_1)$. Dashed lines are the edges of the B-cut.

Proposition 6 *All edges of $K = E(M, L_1, \ldots, L_m, R_1, \ldots, R_n)$ lie between the root and leaf components. Hence M uniquely determines the leaf components of K.*

Proof: Suppose that there is an edge $e \in K$ between two leaf components. Since there is no edge between L_i and R_j, without loss of generality suppose that e connects L_i and L_j. But L_i and L_j contain only sources. Thus, by Proposition 5 , K is not minimal, a contradiction. □

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 $K = E(M, L_1, \ldots, L_m, R_1, \ldots, R_n)$ 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 (see example in Figure 11):

(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 is in a leaf component of K and $M + v + e$ does not contain a source-sink pair (s_i, t_i). Without loss of generality, assume that $v \in R_j$ and either v is not a sink, or $v = t_i$ and $s_i \notin M$ (see Figure 6).

Let $W_1, \ldots, W_p$ be the components of $R_j - v$, and let $\widehat{M} = M + v + \bigcup_{u \in M}\{uv \in E\}$. Then

$$D = E(\widehat{M}, L_1, \ldots, L_m, R_1, \ldots, R_{j-1}, W_1, \ldots, W_p, R_{j+1}, \ldots, R_n)$$

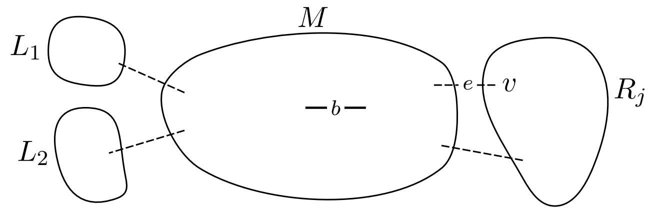

Figure 6: Cut conjunction K in **(q2-a)**

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 $W_1, \ldots, W_p$ (see Figure 7). We will remove non minimal edges of D in **(q3)**.

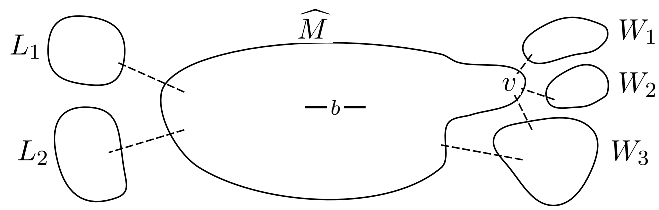

Figure 7: B-cut D in **(q2-a)**

(q2-b) Suppose v is in a leaf component of K and $M + v + e$ contains a source-sink pair (s_i, t_i). Without loss of generality, assume that $v \in R_j$ and $v = t_i$, $s_i \in M$ and $v_L \neq s_i$ (If $v_L = s_i$ we do not allow to include t_i to M). Let $W_1, \ldots, W_p$ be the components of $R_j - t_i$ and let $U_1, \ldots, U_r$ be the components of $M - s_i$ not containing b. Denote $\widehat{M} = (M + t_i + \bigcup_{u \in M}\{ut_i \in E\}) - (s_i + U_1 + \ldots + U_r)$. Then

$$D = E(\widehat{M}, L_1, \ldots, L_m, s_i, U_1, \ldots, U_r, R_1, \ldots, R_{j-1}, W_1, \ldots, W_p, R_{j+1}, \ldots, R_n)$$

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 $W_1, \ldots, W_p$, and removing s_i from M splits M into components $U_1, \ldots, U_r$ and $\widehat{M}$, the component containing b (see Figure 8). We will remove non minimal edges of D in **(q3)**.

(q2-c) Suppose $v \in M - \{v_L, v_R\}$. Without loss of generality, assume that v is adjacent to L_j (see Figure 9). Note that $v \notin \{t_1, \ldots, t_k\}$.

Let $U_1, \ldots, U_r$ be the components of $M - v$ not containing b, and let $\widehat{M} = M - (v + U_1 + \ldots + U_r)$. Then

$$D = E(\widehat{M}, L_1, \ldots, L_{j-1}, L_j + v + \bigcup_{u \in L_j} uv \in E, L_{j+1}, \ldots, L_m, U_1, \ldots, U_r, R_1, \ldots, R_n)$$

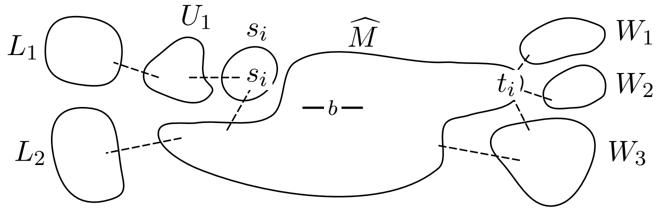

Figure 8: B-cut D in **(q2-b)**

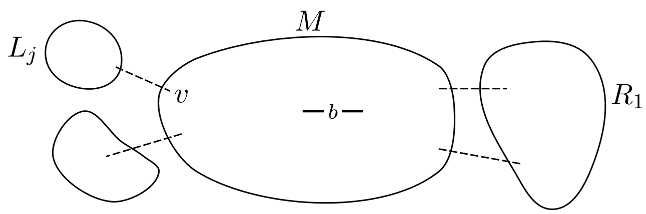

Figure 9: Cut conjunction K in **(q2-c)**

is a B-cut. Note that we have moved v from M to L_j splitting M into components $U_1, \ldots, U_r$ and $\widehat{M}$ (see Figure 10).

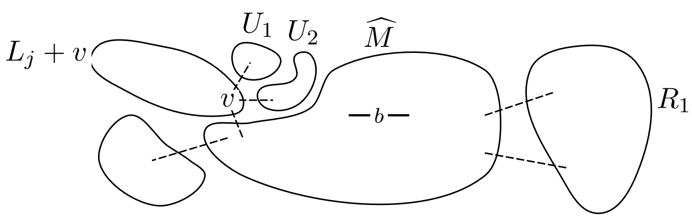

Figure 10: B-cut D in **(q2-c)**

(q3) Let $D = E(G_1, \ldots, G_l)$ 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 $K' = E(M', L'_1, \ldots, L'_{m'}, R'_1, \ldots, R'_{n'})$ be the resulting cut conjunction. Then K' is a neighbor of K in $\mathcal{H}$.

Proposition 7 *The supergraph $\mathcal{H}$ is strongly connected.*

To prove Proposition 7 we need two lemmas.

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. Note that $\mathcal{K}$ has at least one vertex, K_3.

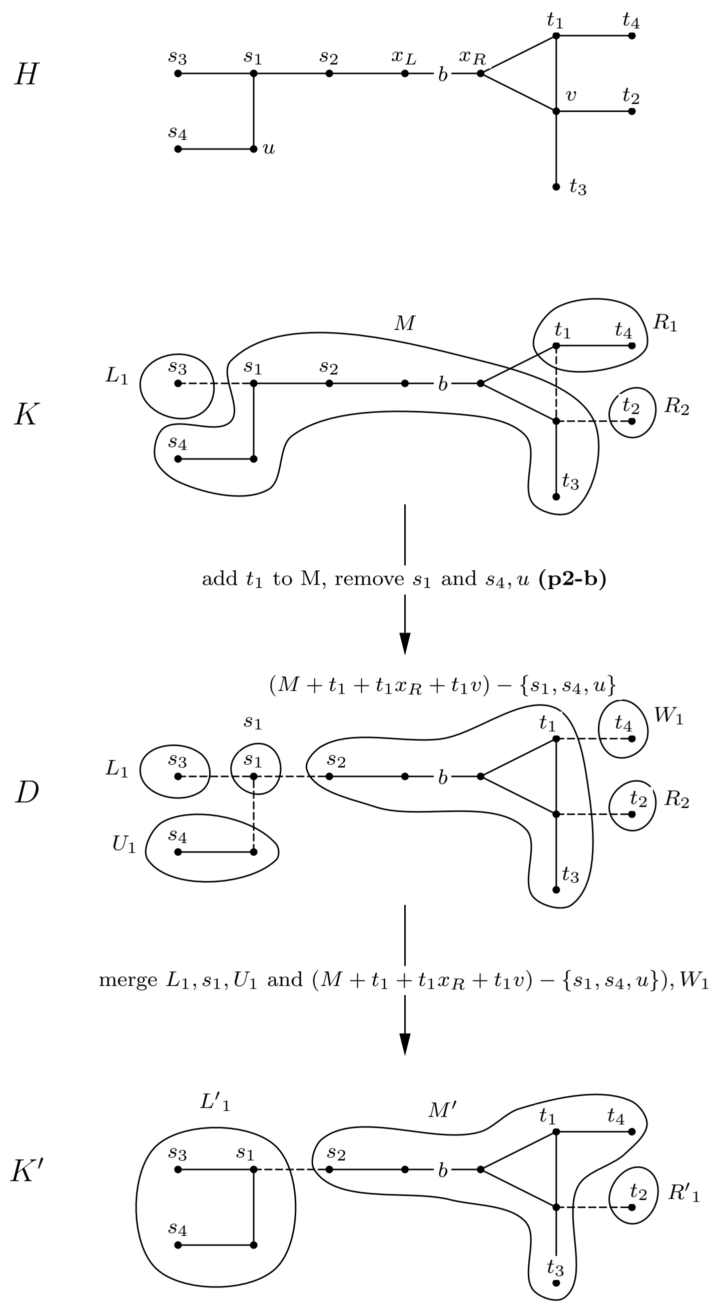

Figure 11: Consider the graph H above and the cut conjunction $K = E(M, L_1, R_1, R_2) = E(\{s_1, s_2, s_4, u, x_L, x_R, t_3, v\}, \{s_3\}, \{t_1, t_4\}, \{t_2\})$. Then $K' = E(M', L'_1, R'_1) = E(\{s_2, x_L, x_R, t_1, t_3, t_4, v\}, \{s_1, s_3, s_4, u\}, \{t_2\})$ is a neighbor of K obtained by moving t_1 to M.

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. Edges that form a contiguous part within M will be called D-solid edges, and the remaining edges will be called D-dashed edges. More precisely, we call an edge e of T a *D-solid edge*, if

- $e \in M$,
- e is reachable from b by using only edges of T that are in M.

Otherwise e is called a *D-dashed edge* (see Figure 12). 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 $K_1 = E(M_1, L_1, \ldots, L_m, R_1, \ldots, R_n)$. We will show by induction on the number of K_1-solid edges $|S_{K_1}|$ that there is a path from K_1 to K_2.

If $|S_{K_1}| = |T|$, then M_1 contains the spanning tree T of blue vertices. Hence $K_1 \in \mathcal{K}$.

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. Without loss of generality, suppose that $v \in R_j$ (see Figure 12). Such an edge exists because K_1-dashed and K_1-solid edges form the spanning tree of blue vertices.

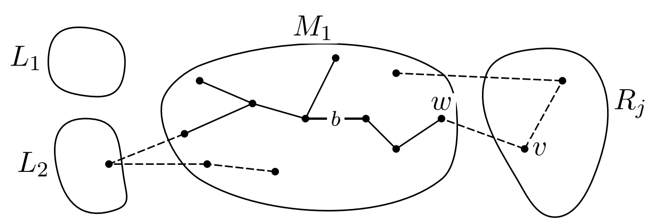

Figure 12: Cut conjunction K_1. Solid lines are the K_1-solid edges, dashed lines are the K_1-dashed edges.

We now show that K_1', a neighbor of K_1, obtained by moving the blue vertex v from the leaf to the root component, has $|S_{K_1'}| \geq |S_{K_1}| + 1$. Depending on v there are two cases.

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 be its root component. Recall that $M_D = M_1 + v$. Thus S_D contains all K_1-solid edges. 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 $|S_{K_1'}| \geq |S_D|$. This implies that $|S_{K_1'}| \geq |S_{K_1}| + 1$.

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 $M_D = (M_1 + t_i) - (s_i + U_1 + \ldots + U_r)$, where $U_1, \ldots, U_r$ 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 $U_1, \ldots, U_r$ and incident to these components are also not K_1-solid, because all paths from b to $U_1, \ldots, U_r$, 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 increase its size after merging with leaf components, hence $|S_{K_1'}| \geq |S_D|$. This implies that $|S_{K_1'}| \geq |S_{K_1}| + 1$.

□

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

Proof: Let $W_1, \ldots, W_q$ be the leaf components of K_3 and $T_1, \ldots, T_q$ be arbitrary spanning trees of $W_1, \ldots, W_q$. Recall that vertices of $W_1, \ldots, W_q$ are called green vertices.

For every leaf W_j there is at least one source-sink pair (s_i, t_i) such that one of s_i and t_i belongs to W_j and the other to the root component of K_3. Choose one such source or sink for every W_j and denote this set by $P = \{p_1, \ldots, p_q\}$.

Let $D = E(M, G_1, \ldots, G_l)$ be a B-cut of H such that all vertices of P are in the leaf components. Let $e \in T_i$ for some $i \in \{1, \ldots, q\}$. We call e a *D-solid edge* if there is $j \in \{1, \ldots, l\}$ such that $e \in G_j$, $p_i \in G_j$ and e is reachable from p_i by using only edges of T_i that are in G_j. Otherwise e is called a *D-dashed edge* (see Figure 13). We denote the set of D-solid edges by S_D and the set of D-dashed edges by D_D. Note that $|S_D| + |D_D| = |T_1| + \ldots + |T_q|$.

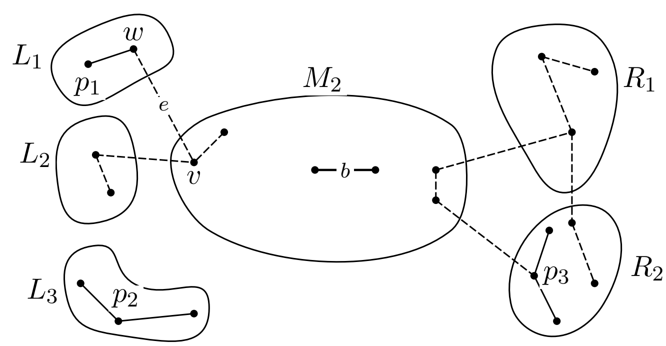

Figure 13: Cut conjunction K_2. The solid lines are K_2-solid edges, the dashed lines are K_2-dashed edges.

Let $K_2 = E(M_2, L_1, \ldots, L_m, R_1, \ldots, R_n)$. Recall that M_3 is the root component of K_3 and its vertices are called blue vertices. Since $M_3 \subseteq M_2$, all elements of P must belong to leaf components of K_2 and thus the notion of K_2-solid edges is well defined. We will show by induction on the number of K_2-solid edges $|S_{K_2}|$, that there is a path in $\mathcal{K}$ from K_2 to K_3 (note that since this path is in $\mathcal{K}$, the root components of vertices on that path must contain all the blue vertices).

If $|S_{K_2}| = |T_1| + \ldots + |T_q|$, all green vertices are in leaf components, so M_2 contains only blue vertices, thus $M_2 = M_3$ and by Proposition 6, we have $K_2 = K_3$.

If $|S_{K_2}| < |T_1| + \ldots + |T_q|$, then there exists a K_2-dashed edge $e = vw$ between two green vertices v and w such that w is in a leaf component, $v \in M_2$ and w is incident to a K_2-solid edge or $w = p_i$. Without loss of generality, suppose that $e \in T_i$ and $w \in L_j$ (see Figure 13). Such an edge exists because K_2-dashed and K_2-solid edges form a spanning forest of green vertices.

We show that K_2', a neighbor of K_2 obtained by moving v from M_2 to L_j, has $|S_{K_2'}| \geq |S_{K_2}| + 1$ and $K_2' \in \mathcal{K}$.

Let $D = E(\widehat{M}, L_1, \ldots, L_j + v, \ldots, L_m, U_1, \ldots, U_r, R_1, \ldots, R_n)$ be a B-cut obtained in **(q2-c)**. Recall that $\widehat{M} = M_2 - (v + U_1 + \ldots + U_r)$, where $U_1, \ldots, U_r$ are the components of $M_2 - v$ not containing b. Note also that $U_1, \ldots, U_r$ cannot contain any blue vertices, since M_2 contains M_3, which is connected, thus removing a green vertex v cannot disconnect any blue vertex from b. Hence $M_3 \subseteq \widehat{M}$. Since in **(q3)** $\widehat{M}$ can only increase its size, the root component of K_2' contains M_3.

Since $L_j + v$ contains both v and w, e is a D-solid edge. Thus $|S_D| = |S_{K_2}| + 1$. In **(q3)** only leaf components not containing vertices of P can merge with $\widehat{M}$. Since these leaf components do not not contain any solid edges, we obtain $|S_{K_2'}| \geq |S_D|$. This implies that $|S_{K_2'}| \geq |S_{K_2}| + 1$.

□

Proof of Proposition 7. Let K_1 and K_3 be arbitrary cut conjunctions and $\mathcal{K}$ be the induced subgraph of $\mathcal{H}$ defined above. By Lemma 1 there is a path in $\mathcal{H}$ from K_1 to some cut conjunction K_2 in $\mathcal{K}$. By Lemma 2 there is a path from any cut conjunction of $\mathcal{K}$ to K_3. The proposition follows. □

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

It will be convenient to assume in this section that the input graph $G = (V, E)$ may contain parallel edges, i.e. that G is a *multigraph*. Given a nonempty set $B \subseteq E$, let

$$\mathcal{F} \overset{\text{def}}{=} \text{minimal } \{X \subseteq E \setminus B \mid \text{no } b \in B \text{ is a bridge of } (V, B \cup X)\}$$

be the family of all minimal bridge-avoiding extensions of B. We enumerate $\mathcal{F}$ by using the $X - e + Y$ method stated in the Introduction. Proposition 8 below implies that this can be accomplished in incremental polynomial time.

Proposition 8 *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 $B' = \{b_1, \ldots, b_k\}$ 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 $i, j \in \{1, \ldots, k\}$. 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 $(V', B' \cup Y')$. In other words, the general enumeration problem for 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 in turn equivalent to the enumeration of all directed paths between a pair of vertices in some explicitly given directed multigraph. To see this, denote by $u_1, \ldots, u_{k+1}$ the $k+1$ vertices on the path $B' = \{b_1, \ldots, b_k\}$ in G', and assume without loss of generatity that $b_i = u_i u_{i+1}$ for $i = 1, \ldots, k$. If no edge $b \in B'$ is a bridge in $(V', B' \cup Y')$, then for each $i = 1, \ldots, k$ there must exist a path $P \subseteq Y'$ such that

($\mathcal{P}'$) P and B' are edge disjoint and

($\mathcal{P}''$) the vertex set of P contains exactly two vertices u_α, u_β of B' such that $\alpha \le i$ and $\beta \ge i + 1$.

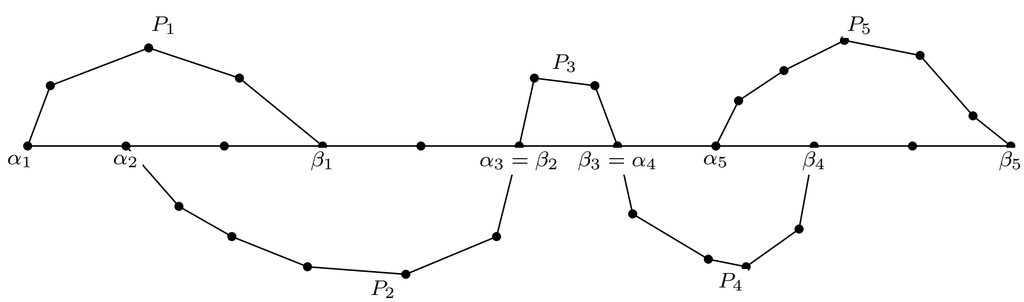

Figure 14: Subgraph $(V', B' \cup Y')$.

By the minimality of Y' we conclude that

$$Y' = P_1 \cup \ldots \cup P_s \tag{1}$$

for some paths $P_1, \ldots, P_s$ satisfying conditions ($\mathcal{P}'$) and ($\mathcal{P}''$) above, where no two distinct paths in the above decomposition have a common vertex outside of B'. Denoting by u_{α_i} and u_{β_i} the intersection of the vertex set of P_i with B', we may also assume without loss of generality that

$$u_1 = \alpha_1 < \alpha_2 \le \beta_1 < \alpha_3 \le \beta_2 < \alpha_4 \le \ldots < \alpha_s \le \beta_{s-1} < \beta_s = u_{k+1}, \tag{2}$$

where some pairs of consecutive paths P_j and P_{j+1} may share a common endpoint on B' (see Figure 14).

Let us now consider the directed multigraph $\overrightarrow{G}' = (V', \overrightarrow{E}')$ obtained from the multigraph $G' = (V', E')$ by replacing the undirected path B' by the directed path $\overrightarrow{B}' = u_1 \leftarrow u_2 \leftarrow \ldots \leftarrow u_k \leftarrow u_{k+1}$ and by adding two opposite arcs $u \to v$ and $v \to u$ for each of the remaining edges $uv \in E' \setminus B'$. From the above discussion it follows that there exists a one to one correspondence between all minimal sets Y' admitting decomposition (1) which satisfies (2) and all (inclusionwise) minimal directed paths from u_1 to u_{k+1} in $\overrightarrow{G}'$ (see Figure 15).

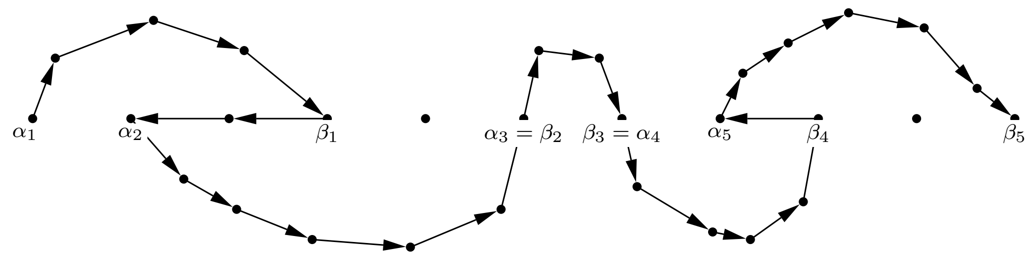

Figure 15: Directed path in $\overrightarrow{G'}$.

Since it is well known that all minimal directed paths between a given pair of vertices can be enumerated via backtracking [RT75] with polynomial delay, Proposition 8 follows. □

4 Proof of Proposition 1

For the sake of completeness we present the proof of Proposition 1 from [BEG+05].

Let M be a vectorial matroid on ground set S, let $B \subseteq S$ and $A \stackrel{\text{def}}{=} S \setminus B$ be a partition of S, and let

$$\mathcal{F} \stackrel{\text{def}}{=} \text{maximal } \{X \subseteq A \mid \text{no } b \in B \text{ is spanned by } X\}.$$

In this section we show that given a subfamily $\mathcal{X} \subseteq \mathcal{F}$, it is NP-hard to decide whether $\mathcal{X} = \mathcal{F}$. We reduce our problem from the well known 3-satisfiability.

Let $\phi = C_1 \wedge C_2 \ldots \wedge C_m$ be a given CNF on n variables with exactly three literals per clause. We may represent the sets A and B as matrices. We let

$$A = (\mathbf{a}^{\bar{x}_1}, \mathbf{a}^{\bar{x}_2}, \ldots, \mathbf{a}^{\bar{x}_n}, \mathbf{a}^{x_1}, \mathbf{a}^{x_2}, \ldots, \mathbf{a}^{x_n}),$$

where $\mathbf{a}^{\bar{x}_i}$ and $\mathbf{a}^{x_i}$ are $(n+1)$-dimensional vectors defined as

$$(\mathbf{a}^{\bar{x}_i})_j = \begin{cases} 1, & \text{if } i = j; \\ 0, & \text{otherwise,} \end{cases}$$

$$(\mathbf{a}^{x_i})_j = \begin{cases} 1, & \text{if } i = j \text{ or } i = n+1; \\ 0, & \text{otherwise.} \end{cases}$$

For every clause $C_p = l_{i_1} \vee l_{i_2} \vee l_{i_3}$, where $l_{i_j} \in \{x_{i_j}, \bar{x}_{i_j}\}$, and $\alpha \in \{0, \dots, n-3\}$, we define

$$\mathbf{b}^{p,\alpha} = 4n\mathbf{a}^{l_{i_1}} + 2n\mathbf{a}^{l_{i_2}} + n\mathbf{a}^{l_{i_3}} + \mathbf{f}^p + \alpha\mathbf{e},$$

where $\mathbf{f}^p$ and $\mathbf{e}$ are $(n+1)$-dimensional vectors defined as

$$f_i^p = \begin{cases} 0, & \text{if } i \in \{i_1, i_2, i_3, n+1\}; \\ 1, & \text{otherwise,} \end{cases}$$

and

$$\mathbf{e} = (0, \dots, 0, 1)^T.$$

Then $B = (\mathbf{b}^{p,\alpha})$, for $p = 1, \dots, m$ and $\alpha = 0, \dots, n-3$ (see Example 4.1).

Example 4.1 *Consider* $\phi = C_1 \wedge C_2 = (x_1 \vee \bar{x}_2 \vee x_3)(x_1 \vee \bar{x}_4 \vee \bar{x}_5)$.
Then

$$A = (\boldsymbol{a}^{\bar{x}_1}, \boldsymbol{a}^{\bar{x}_2}, \dots, \boldsymbol{a}^{\bar{x}_5}, \boldsymbol{a}^{x_1}, \boldsymbol{a}^{x_2}, \dots, \boldsymbol{a}^{x_5}) = \left(\begin{array}{ccccc|ccccc} 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 \\ \hline 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 \end{array}\right)$$

$$\boldsymbol{b}^{1,\alpha} = \left(\begin{array}{r} 4 \cdot 5 \\ 2 \cdot 5 \\ 5 \\ 1 \\ 1 \\ \hline 4 \cdot 5 + 5 + \alpha \end{array}\right)$$

$$\boldsymbol{b}^{2,\alpha} = \left(\begin{array}{r} 5 \\ 1 \\ 1 \\ 4 \cdot 5 \\ 2 \cdot 5 \\ \hline 5 + \alpha \end{array}\right)$$

$$B = \{\boldsymbol{b}^{1,0},\ \boldsymbol{b}^{1,1}, \boldsymbol{b}^{1,2}, \boldsymbol{b}^{2,0}, \boldsymbol{b}^{2,1}, \boldsymbol{b}^{2,2}\}.$$

Claim 1 *For each* $i \in \{1, \dots, n\}$, $A \setminus \{\boldsymbol{a}^{\bar{x}_i}, \boldsymbol{a}^{x_i}\}$ *is a maximal subset of* A *spanning no* $\boldsymbol{b} \in B$.

Proof: Observe that all vectors of $A \setminus \{\mathbf{a}^{\bar{x}_i}, \mathbf{a}^{x_i}\}$ have i^{th} entry zero and every $\mathbf{b} \in B$ has all entries nonzero. Both $A \setminus \{\mathbf{a}^{\bar{x}_i}\}$ and $A \setminus \{\mathbf{a}^{x_i}\}$ span all $b \in B$, since $rank(A \setminus \{\mathbf{a}^{\bar{x}_i}\}) = rank(A \setminus \{\mathbf{a}^{x_i}\}) = n+1$. Thus $A \setminus \{\mathbf{a}^{\bar{x}_i}, \mathbf{a}^{x_i}\}$ is maximal subset of A spanning no $\mathbf{b} \in B$. □

Let $\mathcal{X} = \{A \setminus \{\mathbf{a}^{\bar{x}_1}, \mathbf{a}^{x_1}\}, \dots, A \setminus \{\mathbf{a}^{\bar{x}_n}, \mathbf{a}^{x_n}\}\} \subseteq \mathcal{F}$. We shall call elements of $\mathcal{F} \setminus \mathcal{X}$ *nontrivial*. Let $\mathcal{H}$ be a family of subsets of A of the form $(\mathbf{a}^{l_1}, \mathbf{a}^{l_2}, \dots, \mathbf{a}^{l_n})$, where $l_i \in \{x_i, \bar{x}_i\}$, i.e. subsets of A that contain exactly one of each pair $\mathbf{a}^{\bar{x}_i}$, $\mathbf{a}^{x_i}$, for $i \in \{1, \dots, n\}$.

Claim 2 *Every nontrivial element X of $\mathcal{F}$ belongs to $\mathcal{H}$.*

Proof: X is a maximal subset of A spanning no $\mathbf{b} \in B$ and is not a subset of an element of $\mathcal{X}$, thus X must contain at least one of each pair $\mathbf{a}^{\bar{x}_i}$, $\mathbf{a}^{x_i}$. Suppose that for some j, X contains both $\mathbf{a}^{\bar{x}_j}$, $\mathbf{a}^{x_j}$. Then $rank(X) = n+1$, thus X spans all $\mathbf{b} \in B$, a contradiction. Hence X contains exactly one of $\mathbf{a}^{\bar{x}_i}$, $\mathbf{a}^{x_i}$, for $i \in \{1, \ldots, n\}$. □

Now let $X = (\mathbf{a}^{l_1}, \mathbf{a}^{l_2}, \ldots, \mathbf{a}^{l_n}) \in \mathcal{H}$ and $\mathbf{x} = (x_1, \ldots, x_n)$ be an assignment of ϕ. We define a bijection between elements of $\mathcal{H}$ and assignments of ϕ as follows: $x_i = 0$ if and only if $\mathbf{a}^{x_i} \in X$, $x_i = 1$ if and only if $\mathbf{a}^{\bar{x}_i} \in X$.

Claim 3 *X is nontrivial element of $\mathcal{F}$ if and only if $\boldsymbol{x}$ is a satisfying assignment of ϕ.*

Proof: Let X be nontrivial element of $\mathcal{F}$. By Claim 2, $X \in \mathcal{H}$, so there exists an assignment $\mathbf{x}$ corresponding to X. Suppose that $\mathbf{x}$ is not a satisfying assignment, then $\mathbf{x}$ does not satisfy a clause $C_p = l_{i_1} \vee l_{i_2} \vee l_{i_3}$. Thus l_{i_1}, l_{i_2}, l_{i_3} are assigned 0. Then $\{\mathbf{a}^{l_{i_1}}, \mathbf{a}^{l_{i_2}}, \mathbf{a}^{l_{i_3}}\} \in X$. Let $\alpha = \sum_{j \notin \{i_1, i_2, i_3\}} (1 - x_j)$ be the number of 0's in entries of $\mathbf{x}$ different than i_1, i_2, i_3.

Then $\sum_{i \notin \{i_1, i_2, i_3\}} \mathbf{a}^{l_i} = \mathbf{f} + \alpha \mathbf{e}$, hence $\mathbf{b}^{p,\alpha} = 4n\mathbf{a}^{l_{i_1}} + 2n\mathbf{a}^{l_{i_2}} + n\mathbf{a}^{l_{i_3}} + \sum_{i \notin \{i_1, i_2, i_3\}} \mathbf{a}^{l_i}$. Thus $\mathbf{b}^{p,\alpha}$ is spanned by X, a contradiction (see Example 4.2).

Now let $\mathbf{x}$ be a satisfying assignment. We will show that X spans no $b \in B$. Choose $\mathbf{b}^{p,\alpha} = (b_1, \ldots, b_{n+1}) \in B$ corresponding to the clause $C_p = l_{i_1} \vee l_{i_2} \vee l_{i_3}$. Observe that $X = \left(\frac{I_n}{\mathbf{r}} \right)$, where I_n is $n \times n$ identity matrix and $\mathbf{r} = (r_{l_1}, \ldots, r_{l_n})$ is a n-dimensional vector. Then the system $I_n \mathbf{y} = (b_1, \ldots, b_n)$ has a unique solution

$$y_i = b_i = \begin{cases} 4n, & \text{if } i = i_1; \\ 2n, & \text{if } i = i_2; \\ n, & \text{if } i = i_3; \\ 1, & \text{otherwise.} \end{cases}$$

However the linear combination, with coefficients y_i, of entries of the last row of A cannot be equal to b_{n+1}, the last entry of $\mathbf{b}^{p,\alpha}$, for any $\alpha \in \{0, \ldots, n-3\}$ (see Example 4.3), because

- the linear combination is $\sum_{i=1 \ldots n} y_i r_{l_i} = 4nr_{i_1} + 2nr_{i_2} + nr_{i_3} + \beta$, where $\beta = \sum_{i \notin \{i_1, i_2, i_3\}} (1 - x_i)$ is the number of zero entries of $\mathbf{x}$ different than i_1, i_2, i_3,
- $b_{n+1} = 4n(\mathbf{a}^{l_{i_1}})_{n+1} + 2n(\mathbf{a}^{l_{i_2}})_{n+1} + n(\mathbf{a}^{l_{i_3}})_{n+1} + \alpha$,
- there is at least one index j of $\{i_1, i_2, i_3\}$ such that it satisfies $(\mathbf{a}^{l_j})_{n+1} \neq r_j$ (since $\mathbf{x}$ is a satisfying assignment, it must satisfy every clause).

Hence X is nontrivial element of $\mathcal{F}$. □

Example 4.2 *Let ϕ, A, B be as defined in Example 4.1. A nonsatisfying assignment $\boldsymbol{x} = (0,1,0,0,1)$ of ϕ corresponds to*

$$X = (\boldsymbol{a}^{x_1}, \boldsymbol{a}^{\bar{x}_2}, \boldsymbol{a}^{x_3}, \boldsymbol{a}^{x_4}, \boldsymbol{a}^{\bar{x}_5}) = \left(\begin{array}{ccccc} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ \hline 1 & 0 & 1 & 1 & 0 \end{array}\right)$$

$\boldsymbol{x}$ does not satisfy the first clause $x_1 \vee \bar{x}_2 \vee x_3$, number of 0's not in the first, second or third entry of $\boldsymbol{x}$ is 1, thus X spans $\boldsymbol{b}^{1,1}$:

$$4 \cdot 5 \left(\begin{array}{c} 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ \hline 1 \end{array}\right) + 2 \cdot 5 \left(\begin{array}{c} 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ \hline 0 \end{array}\right) + 5 \left(\begin{array}{c} 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ \hline 1 \end{array}\right) + \left(\begin{array}{c} 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ \hline 1 \end{array}\right) + \left(\begin{array}{c} 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ \hline 0 \end{array}\right) = \left(\begin{array}{r} 4 \cdot 5 \\ 2 \cdot 5 \\ 5 \\ 1 \\ 1 \\ \hline 4 \cdot 5 + 5 + 1 \end{array}\right)$$

Example 4.3 *A satisfying assignment $\boldsymbol{x} = (1,0,0,0,1)$ of ϕ corresponds to*

$$X = (\boldsymbol{a}^{\bar{x}_1}, \boldsymbol{a}^{x_2}, \boldsymbol{a}^{x_3}, \boldsymbol{a}^{x_4}, \boldsymbol{a}^{\bar{x}_5}) = \left(\begin{array}{c} I_5 \\ \hline \boldsymbol{r} \end{array}\right) = \left(\begin{array}{ccccc} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ \hline 0 & 1 & 1 & 1 & 0 \end{array}\right)$$

Choose

$$\boldsymbol{b}^{1,\alpha} = \left(\begin{array}{c} b_1 \\ b_2 \\ b_3 \\ b_4 \\ b_5 \\ \hline b_6 \end{array}\right) = \left(\begin{array}{r} 4 \cdot 5 \\ 2 \cdot 5 \\ 5 \\ 1 \\ 1 \\ \hline 4 \cdot 5 + 5 + \alpha \end{array}\right)$$

corresponding to the first clause $x_1 \vee \bar{x}_2 \vee x_3$. Then the system $I_5 \boldsymbol{y} = (b_1, \ldots, b_5)$ has a unique solution

$$\boldsymbol{y} = \left(\begin{array}{r} 4 \cdot 5 \\ 2 \cdot 5 \\ 5 \\ 1 \\ 1 \end{array}\right)$$

However $\sum_{i=1,\ldots,5} y_i r_{l_i} = 2 \cdot 5 + 5 + 1 \neq 4 \cdot 5 + 5 + \alpha = b_6^{1,\alpha}$, for any $\alpha \in \{0,1,2\}$. Thus X does not span $\boldsymbol{b}^{1,0}$, $\boldsymbol{b}^{1,1}$, $\boldsymbol{b}^{1,2}$. Similarly X does not span $\boldsymbol{b}^{2,0}$, $\boldsymbol{b}^{2,1}$, $\boldsymbol{b}^{2,2}$.

5 Proof of Proposition 2

Let us consider a binary matroid M on ground set $S = A \cup B$, where $B = \{b_1, b_2\}$. As we mentioned in the Introduction, it is enough to consider the dual formulation of the cut conjunction problem:

> *Generate all minimal subsets $X \subseteq A \stackrel{\text{def}}{=} S \setminus B$ such that $X \cup \{b_2\}$ spans b_1 and $X \cup \{b_1\}$ spans b_2 in the dual matroid M^*.*

To see that this enumeration problem is tractable, we show first that for a subset X of A, b_1 is a linear combination of vectors of $X \cup \{b_2\}$ and b_2 is a linear combination of vectors of $X \cup \{b_1\}$ if and only if $b_1 + b_2$ is a linear combination of vectors of X.

If $\sum_{a \in Y} a = b_1 + b_2$, where $Y \subseteq X$, then $\sum_{a \in Y} a + b_1 = b_2$ and $\sum_{a \in Y} a + b_2 = b_1$.

For the converse direction of the above claim, let us consider a subset X of A such that b_1 is a linear combination of $X \cup \{b_2\}$ and b_2 is a linear combination of $X \cup \{b_1\}$. Depending on whether these linear combination include b_2 and b_1, respectively, we have two cases:

Case 1: b_2, b_1 do not appear in either of the linear combinations, i.e. $\sum_{a \in X_1} a = b_1$, $\sum_{a \in X_2} a = b_2$, where $X_1, X_2 \subseteq X$. Then $\sum_{a \in (X_1 \cup X_2) \setminus (X_1 \cap X_2)} a = b_1 + b_2$.

Case 2: Without loss of generality suppose b_2 appears in the first linear combination, i.e. $\sum_{a \in Y} a + b_2 = b_1$, where $Y \subseteq X$. Then $\sum_{a \in Y} a = b_1 + b_2$.

Hence X is a minimal subset of A such that $X \cup \{b_2\}$ spans b_1 and $X \cup \{b_1\}$ spans b_2 in M^* if and only if X is a minimal subset of A spanning $b_1 + b_2$ in the matroid on ground set $A \cup \{b_1 + b_2\}$. Thus our problem reduces to the enumeration of all circuits containing $b_1 + b_2$ in the matroid on ground set $A \cup \{b_1 + b_2\}$, which can be done in incremental polynomial time [BEG+05]. □

Let us remark that similar simplification cannot work for $|B| > 2$. For instance, for $B = \{b_1, b_2, b_3\}$, the facts that b_i is a linear combination of vectors of $X \cup \{B \setminus b_i\}$, for $i = 1, 2, 3$, do not imply that $b_1 + b_2 + b_3$ is a linear combination of vectors of X. Consider e.g., the vectors a_1, a_2, a_3, a_4, a_5 satisfying $a_1 + a_2 = b_1, a_3 + a_4 = b_2, a_5 + b_1 = b_3$.

References

[BEG+04] 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.

[BEG+05] E. Boros, K. Elbassioni, V. Gurvich, L. Khachiyan, and K. Makino. On the complexity of some enumeration problems for matroids. To appear in SIAM Journal on Discrete Mathematics, 2005.

[EG95] T. Eiter and G. Gottlob. Identifyig the minimal transversals of a hypergraph and related problems. *SIAM Journal on Computing*, 24:1278–1304, 1995.

[FK96] M. Fredman and L. Khachiyan. On the complexity of dualization of monotone disjunctive normal forms. *Journal of Algorithms*, 21:618–628, 1996.

[LLK80] 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.

[RT75] R. C. Read and R. E. Tarjan. Bounds on backtrack algorithms for listing cycles, paths, and spanning trees. *Networks*, 5:237–252, 1975.

[Sch03] A. Schrijver. *Combinatorial Optimization: Polyhedra and Efficiency*, volume B. Springer Verlag, Berlin, Heidelberg, New York, 2003. page 654.

[TSOA80] 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.

[Vaz01] V. Vazirani. *Approximation Algorithms*. Springer Verlag, Berlin, Heidelberg, New York, 2001.

[Wel76] D.J.A. Welsh. *Matroid Theory*. Academic Press, London, 1976.