# Generating Paths and Cuts in Multi-pole (Di)graphs[⋆]

Endre Boros[1], Khaled Elbassioni[1], Vladimir Gurvich[1], Leonid Khachiyan[2], and Kazuhisa Makino[3]

[1] RUTCOR, Rutgers University, 640 Bartholomew Road, Piscataway NJ 08854-8003, {boros,elbassio,gurvich}@rutcor.rutgers.edu

[2] Department of Computer Science, Rutgers University, 110 Frelinghuysen Road, Piscataway NJ 08854-8003, leonid@cs.rutgers.edu

[3] Division of Systems Science, 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 a (directed) graph with vertex set $V$ and edge (arc) set $E$. Given a set $\mathcal{P}$ of (source-sink) pairs of vertices of $G$, an important problem that arises in the computation of network reliability is the enumeration of minimal subsets of edges (arcs) that connect/disconnect all/at least one of the given source-sink pairs of $\mathcal{P}$. For undirected graphs, we show that the enumeration problems for conjunctions of paths and disjunctions of cuts can be solved in incremental polynomial time. For directed graphs both of these problems are NP-hard. We also give a polynomial delay algorithm for enumerating minimal sets of arcs connecting respectively two given nodes $s_1$ and $s_2$ to a given vertex $t_1$, and each vertex of a given subset of vertices $T_2$.

## 1 Introduction

Let $G = (V, E)$ be a (directed) graph on vertex set $V$ and edge (arc) set $E$. Let $\mathcal{P}$ be an arbitrary family of (*source-sink*) pairs of vertices of $G$: $\mathcal{P} = \{(s_i, t_i) \in V \times V \mid i \in [k] = \{1, \ldots, k\}\}$. We assume that $s_i \neq t_i$ for all $i \in [k]$, however sources and sinks may coincide (i.e. for $i \neq j$, we may have any of $s_i = s_j$, $t_i = t_j$ or $s_i = t_j$).

Consider the following two dual enumeration problems, stated as the generation of all minimal subsets of edges (arcs) $E' \subseteq E$ such that

**Path-conjunction (PC):** $s_i$ is connected to $t_i$ in the (di)graph $G' = (V, E')$ for all $i \in [k]$;

[⋆] This research was supported by the National Science Foundation (Grant IIS-0118635). The second and third authors are also grateful for the partial support by DIMACS, the National Science Foundation's Center for Discrete Mathematics and Theoretical Computer Science.

**Cut-disjunction (CD):** $s_i$ is not connected to $t_i$ in the (di)graph $G' = (V, E \setminus E')$ for some $i \in [k]$.

These problems play an important role in network reliability, where edges or arcs represent communication or transportation links, which may work or fail independently, and where the goal is to determine the probability that the network is working, based on the individual edge/arc failure probabilities. It turns out that such network reliability computations require in the general case the list of all multipolar (directed) paths and cuts, depending on the type of connectivity the network ought to maintain, see e.g. [1,4,5,11].

It is easy to see that for the path-conjunction and cut-disjunction problems on undirected graphs, we can assume without loss of generality that the given family of pairs $\mathcal{P}$ corresponds to all pairs in a given family of disjoint vertex sets $V_1, \ldots, V_k \subseteq V$, i.e., $\mathcal{P} = \frac{V_1}{2} \cup \ldots \cup \frac{V_k}{2}$. Such path-conjunctions are usually referred to in the literature as *generalized Steiner trees*. In other words, a generalized Steiner tree is a minimal set of edges $E' \subseteq E$ connecting all vertices within each set $V_i$, i.e. for each $i = 1, \ldots, k$, all vertices of $V_i$ must belong to a single connected component of $(V, E')$. In particular, for $k = 1$ we obtain the usual definition of Steiner trees. When each set $V_i$ consists of two vertices $\{u_i, v_i\}$, generalized Steiner trees are called *point-to-point* connections. By the same token, the general case of the cut-disjunction problem for undirected graphs calls for enumerating all minimal subsets of edges whose removal disconnects at least one of the given vertex sets $V_1, \ldots, V_k \subseteq V$.

Given a (di)graph $G = (V, E)$, we consider the problem of listing all subgraphs of $G$, or correspondingly, the family $\mathcal{F}_\pi \subseteq 2^E$ of all minimal subsets of $E$, satisfying a certain monotone connectivity property $\pi : 2^E \mapsto \{0, 1\}$:

***GEN***$(\mathcal{F}_\pi, \mathcal{X})$**:** *Given a monotone connectivity property $\pi$ and a subfamily $\mathcal{X} \subseteq \mathcal{F}_\pi$ of subgraphs of a (di)graph $G$ satisfying $\pi$, either find a new subgraph $X \in \mathcal{F}_\pi \setminus \mathcal{X}$, or prove that the given partial list is complete: $\mathcal{X} = \mathcal{F}_\pi$.*

For instance, if $\pi(X)$ is the property that the subgraph with edge set $X \subseteq E$ is connected, then $\mathcal{F}_\pi$ is the family of spanning trees of $G$. In this paper, we shall represent subgraphs of a given graph $G = (V, E)$ as subsets of edges. Enumeration algorithms for listing subgraphs satisfying a number of monotone properties are well known. For instance, it is known [13] that the problems of listing all minimal cuts or all spanning trees of an undirected graph $G = (V, E)$ can be solved with delay $O(|E|)$ per generated set. It is also known (see e.g., [6, 7,12]) that all minimal $(s, t)$-cuts or $(s, t)$-paths, can be listed with delay $O(|E|)$ per cut or path. Furthermore, polynomial delay algorithms also exist for listing directed spanning trees in a directed graph [9,14]. We consider the families of minimal path conjunctions and cut disjunctions for a given graph $G$ and family of pairs $\mathcal{P}$, denoted respectively by $\mathcal{F}_{PC}(G, \mathcal{P})$, and $\mathcal{F}_{CD}(G, \mathcal{P})$ .

**Theorem 1.** *For any undirected graph $G$ and any family of pairs of vertices $\mathcal{P}$, the two generation problems GEN$(\mathcal{F}_{PC}(G, \mathcal{P}), \mathcal{X})$ and GEN$(\mathcal{F}_{CD}(G, \mathcal{P}), \mathcal{X})$ can be solved in polynomial time.*

Theorem 1 can be stated in the more general setting of generation problems in matroids. These more general results will be described in Section 2. More generally, the above two problems GEN($\mathcal{F}_{PC}(G,\mathcal{P}),\mathcal{X}$) and GEN($\mathcal{F}_{CD}$ $(G,\mathcal{P}),\mathcal{X}$) remain tractable if we fix some edges, and ask for minimal extensions that form path conjunctions or cut disjunctions. The reduction is trivial: just contract these edges and consider the problem on the resulting graph.

Now let us consider the analogous problems for directed graphs. In the special case when $\mathcal{P}$ is the complete set of pairs of vertices of the graph $G$, problem GEN($\mathcal{F}_{PC}$ $(G,\mathcal{P}),\mathcal{X}$) calls for enumerating minimal strongly connected subgraphs of $G$, the status of which was determined to be incrementally polynomially solvable in [3]. Problem GEN($\mathcal{F}_{CD}(G,\mathcal{P}),\mathcal{X}$), on the other hand, in this special case is known to be NP-hard [3]. We show here that the general path conjunction problem with arbitrary pairs is also NP-hard.

**Theorem 2.** *Given a digraph $G$, a family of $k$ disjoint pairs $\mathcal{P}$ of vertices of $G$, and a subfamily $\mathcal{X} \subseteq \mathcal{F}_{PC}(G,\mathcal{P})$ of path conjunctions of $G$ with respect to $\mathcal{P}$, it is NP-hard to decide if this subfamily is complete: $\mathcal{X} = \mathcal{F}_{PC}(G,\mathcal{P})$.*

As for the *disjunction* of directed paths, it is still open whether their enumeration can be done in incremental polynomial time. However, we prove in Section 5 that the following related extension problem is NP-hard.

**Theorem 3.** *Let $G = (V,E)$ be a digraph and $\mathcal{P}$ be a family of pairs of vertices of $G$. Fix a subset of arcs $Y \subseteq E$ and let $\mathcal{F}_{PD}(G,\mathcal{P},Y)$ be the family of minimal subsets of arcs $X \subseteq E$ such that the graph $(V, X \cup Y)$ contains a directed path between at least one of the pairs of vertices in $\mathcal{P}$. Given a sublist $\mathcal{X} \subseteq \mathcal{F}_{PD}(G,\mathcal{P},Y)$ it is NP-hard to check if the given sublist is complete: $\mathcal{X} = \mathcal{F}_{PD}(G,\mathcal{P},Y)$.*

The same statement holds for the extension problem for disjunction of cuts. These results should be contrasted with the corresponding results for undirected graphs, where the extension problems are polynomially solvable.

Finally, we shall also prove the following positive result which generalizes previously known results on generating directed spanning trees [9,14].

**Theorem 4.** *Let $G = (V,E)$ be a directed graph, $s_1, s_2 \in V$ be two arbitrary (not necessarily distinct) vertices, and $t_1, T_2 \subseteq V$ be arbitrary vertex and subset of vertices of $V$. Let $\mathcal{P} = \{(s_1,t_1)\} \bigcup \{(s_2,t) \ : \ t \in T_2\}$. Then problem GEN($\mathcal{F}_{PC}(G,\mathcal{P}),\mathcal{X}$) can be solved with polynomial delay.*

To prove Theorem 4, we use the *backtracking* method for enumeration (see [13]). This method can generally be explained as follows. Suppose that we want to enumerate all elements of a family $\mathcal{F} \subseteq 2^E$, where $E = \{1,2,\ldots,|E|\}$. The algorithm works by building a search tree of depth $|E|$ whose leaves contain the elements of the family $\mathcal{F}$. Each node of the tree is identified with two disjoint subsets $S_1, S_2 \subseteq E$, and have at most two children. At the root of the tree, we have $S_1 = S_2 = \emptyset$. The left child of any node $(S_1,S_2)$ of the tree at level $i$ is $(S_1 \cup \{i\}, S_2)$ provided that there is an $X \in \mathcal{F}$, such that $X \supseteq S_1 \cup \{i\}$ and $X \cap S_2 = \emptyset$. The right child of any node $(S_1,S_2)$ of the tree at level $i$

is $(S_1, S_2 \cup \{i\})$ provided that there is an $X \in \mathcal{F}$, such that $X \supseteq S_1$ and $X \cap (S_2 \cup \{i\}) = \emptyset$. When we extend the set $S_1$ by a new element $i \in E$ to get a new set $S_1'$, we may restrict our attention to subsets $S_1'$ satisfying certain properties. More precisely, let $\mathcal{F}' \subseteq 2^E$ be a family of sets such that we can test in polynomial time if a set $X \in \mathcal{F}'$, and such that for every $X \in \mathcal{F}$, there is a set $X' \in \mathcal{F}'$ contained in $X$. The following is a sufficient condition for this method to work in polynomial time:

**(F)** For any two disjoint subsets $S_1 \in \mathcal{F}'$ and $S_2 \subseteq E$, and for any $i \in E \setminus (S_1 \cup S_2)$ such that $S_1 \cup \{i\} \in \mathcal{F}'$, we can check in polynomial time if there is an element $X \in \mathcal{F}$, such that $X \supseteq S_1 \cup \{i\}$ and $X \cap S_2 = \emptyset$.

This way, under assumption (F), we obtain a polynomial delay, polynomial space algorithm for enumerating the elements of $\mathcal{F}$ in lexicographic order, by performing a *depth-first search* traversal on the nodes of the backtracking tree constructed as above.

The remainder of the paper consists of the proof of a more general result for matroids in Section 2 instrumental in the proof of Theorem 1 in Section 3. The proofs of Theorems 2, 3 and 4 are presented in Sections 4, 5 and 6, respectively.

## 2 Generation Problems in Matroids

Many generation problems for undirected graphs can be viewed more generally as generation problems in graphical or co-graphical matroids, see [10,16] for general background on Matroid Theory.

Let $M$ be a matroid on ground set $S$ of cardinality $|S| = n$. In general, we assume that $M$ is defined by an *independence oracle*, i.e. an algorithm $\mathcal{O}$ which, given a subset $X$ of $S$, can determine in unit time whether or not $X$ is independent in $M$. This assumption implies that the rank of any set $X \subseteq S$, $r(X) = \max\{|I| \mid I \text{ independent subset of } X\}$, and in particular, the rank of the matroid $r(M) \overset{\text{def}}{=} r(S)$ can be determined in $O(n)$ time by the well-known greedy algorithm. A set $U \subseteq S$ is said to *span* an element $s \in S$ if $r(U \cup \{s\}) = r(U)$, to span a set $W \subseteq S$ if $r(U \cup W) = r(U)$, and to span the whole matroid $M$ if $r(U) = r(S) = r(M)$. In the last case $S$ is called a *spanning set*. Minimal spanning sets for $S$ are called *bases*. Equivalently, a base $B$ is a maximal independent set. Further, $|B| = r(M)$ for every base $B$ in $M$. Minimal dependent sets in $M$ are called *circuits*.

Denote the hypergraphs of all bases and circuits of $M$ by $\mathcal{B}(M)$ and $\mathcal{C}(M)$ respectively. Given a matroid $M$, it is known that the complementary set $\mathcal{B}^c = \{S \setminus B \mid B \in \mathcal{B}(M)\}$ is the set of bases of another matroid $M^*$ on the same ground set $S$. The matroid $M^*$ is called the *dual matroid* of $M$. The bases of $M^*$ are called the *co-bases* of $M$ and the circuits of $M^*$ are called the *co-circuits* of $M$. Note that the rank of a set $X \subseteq S$ in the dual matroid $M^*$, $r^*(X) = r(S \setminus X) + |X| - r(M)$, can also be computed in $O(n)$ oracle time, and thus, $\mathcal{M}$ can be used as an independence oracle for the dual matroid.

In reliability theory we apply the above definitions to graphical and co-graphical matroids. Given a connected multi-graph $G = (V, E)$, we denote by $M_G$ the graphical matroid for $G$. The elements of $M_G$ are the edges of $G$, and a set $E' \subseteq E$ spans $e = (v', v'')$ iff $E'$ contains a path between $v'$ and $v''$. In particular, $E'$ is independent iff it is a forest, that is if $(V, E')$ contains no cycle. This means that the bases $M_G$ are the spanning trees of $G$, and the circuits of $M_G$ are the simple cycles of $G$. Thus the paths between vertices $v'$ and $v''$ in $G$ are identified with the circuits through the edge $e = (v', v'')$ in $M_G$. (If $e = (v', v'') \notin E$, we can add $e$ to $M_G$.) The bases of the dual (co-graphical) matroid $M_G^*$ are the complements to the spanning trees, i.e. the minimal transversals to the simple cycles, so called *feedbacks*. The circuits are the minimal transversals to the spanning trees, that is minimal *cuts* for $G$.

We shall need the following Theorem from [2]

**Theorem 5.** *Let $M$ be a matroid with ground set $S$, let $a \in S$, and let $\mathcal{C}(M, a)$ be the set of circuits $C$ of $M$ such that $a \in C$. Assuming that $M$ is defined by an independence oracle, all elements of $\mathcal{C}(M, a)$ can be enumerated in incremental polynomial time.*

The above theorem can be extended as follows.

**Theorem 6.** *Given a matroid $M$ with ground set $S$ and two non-empty sets $D, A \subseteq V$, all minimal subsets of $\mathcal{D}$ which span $A$ can be enumerated in incremental polynomial time. All maximal subsets of $D$ which do not span $A$ can also be enumerated in incremental polynomial time.*

*Proof.* Let $\alpha$ be a new element representing $A$, and let $M_\alpha$ be the matroid on $D \cup \alpha$ with the following rank function:

$$\rho(X) = \begin{cases} r(X), & \text{if} \quad \alpha \notin X \\ \max\{r((X \setminus \alpha) \cup a) \mid a \in A\}, & \text{otherwise.} \end{cases} \tag{1}$$

It is easy to check that $M_\alpha$ is indeed a matroid. When $M$ is a vectorial matroid over a large field, $\alpha$ can be interpreted as the "general linear combination" of all elements of $A$; in general, $\rho(X)$ is the so-called *principal extension of* $r(X)$ *on* $A$ *with value* 1 (see e.g. [8]). Let $SPAN(D, A)$ be the collection of all minimal subsets of $\mathcal{D}$ which span $A$. When $I \in \mathcal{SPAN}(D, A)$ then $I \cup \alpha$ is a circuit in $M_\alpha$ and conversely, for any circuit $C$ in $M_\alpha$ containing $\alpha$, the set $C \setminus \alpha$ belongs to $\mathcal{SPAN}(D, A)$. Hence the enumeration problem for $\mathcal{SPAN}(D, A)$ is equivalent with that for the set of all circuits through $\alpha$ in $M_\alpha$. Given an independence oracle for $M$, the rank function (1) of the extended matroid can be trivially evaluated in oracle-polynomial time. Therefore the first claim of Theorem 6 directly follows from Theorem 5. To see the second claim note that the maximal subsets of $D$ which do not span $A$ are in one-to-one correspondence with the circuits of the dual matroid $M_\alpha^*$ which contain $\alpha$. □

## 3 Proof of Theorem 1

Let us consider first path-conjunctions (PC): In order to generate all minimal subsets $X$ of edges of a given (multi) graph $G = (V, E)$ such that $G(X) = (V, X)$ contains paths between given pairs of vertices $s_i, t_i$, for $i \in [k]$, we add the set $A$ of $k$ of "new" edges $e_i = (s_i, t_i)$ to $E$, and let $D = E$. Then each minimal subset of $D$ which spans $A$ in the graphical matroid of $G' = (V, D \cup A)$ is a minimal set of edges of $G$ connecting all pairs of vertices $s_i, t_i$, and vice versa. Consequently, all such minimal path conjunctions can be enumerated in incremental polynomial time by the first part of Theorem 6.

Let us consider next cut-disjunctions (CD): Generating all minimal subsets of edges of $G = (V, E)$ which disconnect at least one pair of vertices $s_i, t_i$ for $k \in [k]$ is equivalent to the generation of all maximal subsets of $E$ which do not connect all the pairs. Letting as before $D = E$ and $A = \{(s_i, t_i) : i \in [k]\}$, we conclude that all such maximal cut disjunction can be enumerated in incremental polynomial time by the second part of Theorem 6. □

**Remark.** Suppose we are given a graph $G = (V, E)$, a set of $k$ terminals $\{s_1, \ldots, s_k\} \subseteq V$, and it is sought to enumerate all minimal sets of edges whose removal disconnects the terminals $\{s_1, \ldots, s_k\}$ from each other. Such a minimal set of edges is commonly referred to as a *multiway cut*, see [15].

Interestingly, the special case of enumerating cut conjunctions can be solved in incremental polynomial time as follows. First of all, we can assume without loss of generality that the input graph $G = (V, E)$ contains no edges between the given terminals $s_1, \ldots, s_k$: if such edges exist, then any multiway cut must remove all of them. Let $A$ be the set of $k-1$ "new" edges forming a spanning tree on the terminals, say $A = \{a_1 = (s_1, s_2), a_2 = (s_2, s_3), ..., a_{k-1} = (s_{k-1}, s_k)\}$. Let $G' = (V, E')$, where $E' = E \cup A$, and let $M^*$ be the co-graphical matroid of $G'$. An edge set $X \subseteq E$ spans $A$ in $M^*$ if and only if

$$r^*(X \cup A) = r^*(X), \tag{2}$$

where $r^*(\cdot)$ is the rank function of $M^*$. By definition, $r^*(Z) = r(E' \setminus Z) + |Z| - r(M)$, where $M$ is the graphical matroid of $G'$ and $r(\cdot)$ is the rank function of $M$. So condition (2) can be written as follows: $r(E \setminus X) + |A| = r(E' \setminus X)$, or equivalently, $r(Y) + |A| = r(Y \cup A)$ where $Y = E \setminus X$ is the complement of $X$ in $G = (V, E)$. So if we remove $X$ from $G$, and then start adding the edges of $A$ to the resulting graph $(V, Y)$, then each new edge from $A$ should be decreasing the number of connected components. But this is the same as saying that the terminals $s_1, ..., s_k$ are all in distinct connected components of $(V, Y)$, i.e. that $X$ is a multiway cut. So the enumeration problem for multiway cuts, in this case, is equivalent to the enumeration of all minimal subsets $X$ of $E$ such that $X$ spans $A$ in $M^*$, which, by Theorem 6, can be done in incremental polynomial time. □

Analogously, the special case of path disjunction, when the goal is to enumerate all minimal subsets of edges which connect at least one pair of vertices in $\{s_1, s_2, \ldots, s_k\}$, can also be solved in incremental polynomial time.

## 4 Proof of Theorem 2

We use a polynomial-time transformation from the satisfiability problem. Let $\phi = C_1 \wedge \ldots \wedge C_m$ be a conjunctive normal form on $2n$ literals $x_1, \overline{x}_1, \ldots, x_n, \overline{x_n}$. We construct a digraph $G = (V, E)$ on $|V| = 6m + n + 2$ vertices and $|E| = 6m + 2n + \sum_{j=1}^{m} |C_j|$ arcs. An example of our construction is given in Figure 1.
*Vertices:* We associate with each clause $C_j$, for $j = 1, \ldots, m$, 6 vertices of $G$: $s_j, s'_j, s''_j$ and $t_j, t'_j$, and $t''_j$. We associate with each variable $x_i$, for $i = 1, \ldots, n$ a vertex $x_i$ of $V$. In addition there are two other vertices $u$ and $v$.
*Arcs:* For each clause $C_j$, for $j = 1, \ldots, m$, we have 6 arcs $(s'_j, s_j), (s_j, u), (u, s''_j)$ and $(t'_j, v), (v, t_j), (v, t''_j)$. For each $i = 1, \ldots, n$, we also have two arcs $(u, x_i)$ and $(x_i, v)$ representing positive and negative literals $x_i$ and $\overline{x}_i$ respectively. Finally, if $x_i$ appears in $C_j$ we add an arc between $s_j$ and $x_i$ and if $\overline{x}_i$ appears in $C_j$ we add an arc between $x_i$ and $t_j$.
*Pairs:* We use the following set $\mathcal{P}$ of $3m$ disjoint pairs:

$$\mathcal{P} = \{(s_j, t_j) : j \in [m] = \{1, \ldots, m\}\} \bigcup \{(s'_j, s''_j) : j \in [m]\} \bigcup \{(t'_j, t''_j) : j \in [m]\}.$$

*Trivial path conjunctions:* Note that any path conjunction must include the sets of arcs $E_j = \{(s'_j, s_j), (s_j, u), (u, s''_j), (t'_j, v), (v, t_j), (t_j, t''_j)\}$, for $j = 1, \ldots, m$. Thus, any path conjunction that connects the family of pairs specified by $\mathcal{P}$ must be an extension of this set of arcs. We call such an extension trivial if it includes both arcs $(u, x_i)$ and $(x_i, v)$ for some $i \in [n]$. This way, we define a set of $n$ trivial path conjunctions.
*Non-trivial minimal path conjunctions:* Let us now show that any non-trivial minimal path conjunction yields a satisfying assignment for $C$ and conversely, any satisfying assignment for $\phi$ gives a non-trivial minimal path conjunction for $G$ with respect to $\mathcal{P}$. This will prove Theorem 2. Note that any such non-trivial path conjunction must avoid one of the arcs $(u, x_i), (x_i, v)$ for each $i = 1, \ldots, n$.

Let $\sigma = (\ell_1, \ell_2, \ldots, \ell_n)$ be the set of literals assigned the value $True$ in a satisfying truth assignment for $C$. We define a non-trivial minimal path conjunction

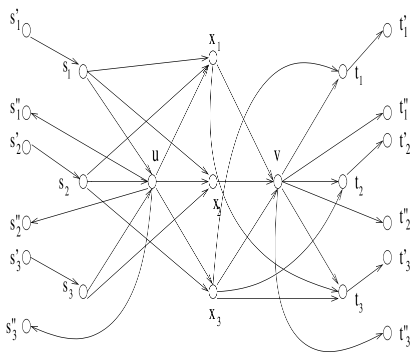

**Fig. 1.** An example of the NP-hard construction proof for Theorem 2 with CNF $\phi = (x_1 \vee x_2 \vee \overline{x}_3) \wedge (x_1 \vee \overline{x}_2 \vee x_3) \wedge (\overline{x}_1 \vee x_2 \vee \overline{x}_3)$.

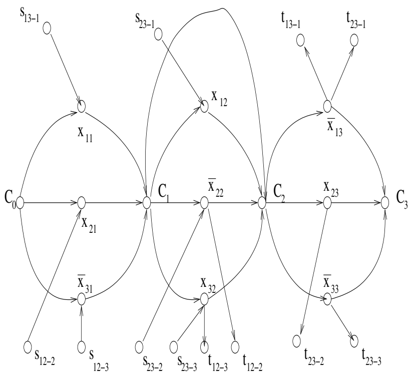

**Fig. 2.** An example of the NP-hard construction proof for Theorem 3 with CNF $\phi = (x_1 \vee x_2 \vee \overline{x}_3) \wedge (x_1 \vee \overline{x_2} \vee x_3) \wedge (\overline{x}_1 \vee x_2 \vee \overline{x}_3)$.

$X$ of $G$ corresponding to $\sigma$ as follows. First we include in $X$ all sets of arcs belonging to $E_j$, for $j = 1, \ldots, m$. Next, we append to $X$ all arcs corresponding to literals *not* appearing in $\sigma$. Then for each clause $C_j$, for $j = 1, \ldots, m$, we include one of the arcs $(s_j, x_i)$ or $(x_i, t_j)$ depending on which of the literals $x_i$ or $\bar{x}_i$ satisfies $C_j$, respectively. Now, the fact that $\sigma$ is satisfying implies that there is a path from $s_j$ to $t_j$, for each $j = 1, \ldots, m$.

Conversely, let $X$ be a non-trivial minimal path conjunction connecting the pairs of $\mathcal{P}$. The non-triviality of $X$ implies that, for each $i = 1, \ldots, n$, one of the arcs $(u, x_i)$ or $(x_i, v)$ is not included in $X$. Let us define a satisfying truth assignment $\sigma$ that assigns True to the literals corresponding to arcs *not* included in $X$. Since $X$ contains a path from $s_j$ to $t_j$ for each $j \in [m]$, one of the arcs $(s_j, x_i)$ or $(x_i, t_j)$ must be included in $X$, implying that $\phi$ is satisfiable.

## 5 Proof of Theorem 3

Again, we use a polynomial-time transformation from the satisfiability problem. Let $\phi = C_1 \wedge \ldots \wedge C_m$ be a conjunctive normal form (CNF) on $2n$ literals $x_1, \overline{x}_1, \ldots, x_n, \overline{x}_n$. Assume that no clause contains a literal and its negation, and that each of the $2n$ literals appears in at least one of the clauses. For a literal $\ell_i$, let $d(\ell_i)$ be the number of clauses in which the literal appears. We construct a digraph $G = (V, E)$ on $|V| = m + 1 + \sum_{j=1}^{m} |C_j| + 2\sum_{i=1}^{n} d(x_i)d(\overline{x}_i)$ vertices and $|E| \leq 2\sum_{j=1}^{m} |C_j| + 3\sum_{i=1}^{n} d(x_i)d(\overline{x}_i)$ arcs. An example of our construction is given in Figure 2.

*Vertices:* In addition to vertex $C_0$, we associate a vertex $C_j \in V$ with each clause $C_j$, for $j = 1, \ldots, m$. For each pair of clauses $(C_j, C_k)$, $j < k$, such that literal $\ell_i$ appears in $C_j$ and $\overline{\ell}_i$ appears in $C_k$, we define two vertices $s_{jk-i}, t_{jk-i}$. Finally, for each clause $C_j$, for $j = 1, \ldots, m$, and for each literal $\ell_i$ appearing in $C_j$, we define a vertex $\ell_{ij}$ of $V$.

*Arcs:* For each clause $C_j$ and each literal $\ell_i$ appearing in $C_j$ we define two arcs $(C_{j-1}, \ell_{ij})$ and $(\ell_{ij}, C_j)$. In addition, for each pair of clauses $(C_j, C_k)$, $j < k$, such that literal $\ell_i$ appears in $C_j$ and $\overline{\ell}_i$ appears in $C_k$, we include the arcs $(s_{jk-i}, \ell_{ij}), (C_j, C_{k-1})$ and $(\overline{\ell}_{ik}, t_{jk-i})$.
*Pairs:* We use a set $\mathcal{P}$ of $\sum_{i=1}^{n} d(x_i)d(\overline{x}_i)+1$ disjoint pairs. $\mathcal{P}$ contains $(C_0, C_m)$ and $(s_{jk-i}, t_{jk-i})$ for every pair of clauses $(C_j, C_k)$, $j < k$, such that literal $\ell_i$ appears in $C_j$ and $\overline{\ell}_i$ appears in $C_k$.
*Fixed set of arcs $Y$:* We include in $Y$ all arcs of the form $(s_{jk-i}, \ell_{ij}), (\ell_{ij}, t_{jk-i})$ and $(C_j, C_{k-1})$ for all $j, k \in [m]$ and all literals $\ell_i$ appearing in $C_j$ or $C_k$.

Clearly, for every pair of clauses $(C_j, C_k)$, $j < k$, such that literal $\ell_i$ appears in $C_j$ and $\overline{\ell}_i$ appears in $C_k$, the two arcs $(\ell_j i, C_j)$ and $(\overline{\ell}_{ik}, C_k)$ extend $Y$ to a dipath connecting $s_{jk-i}$ to $t_{jk-i}$. Thus apart form this set of $\sum_{i=1}^{n} d(x_i)d(\overline{x}_i)$ paths, any other path disjunction must avoid one of the arcs $(\ell_{ij}, C_j)$ or $(\overline{\ell}_{ik}, C_k)$ for every $j < k$, such that literal $\ell_i$ appears in $C_j$ and $\overline{\ell}_i$ appears in $C_k$. The only path that remains to be connected is the one from $C_0$ to $C_m$ and it is easy to see that such path corresponds to a satisfying truth assignment for $\phi$.

## 6 Proof of Theorem 4

Let $G = (V, E)$ be a digraph. Given a node $s \in V$ and a subset of nodes $T \subseteq V$, let us call an $(s, T)$-*directed Steiner tree* any minimal subset of arcs $X \subseteq E$ such that there is a dipath in $(V, X)$ from $s$ to every node in $T$. Note that the minimality of $X$ implies indeed that the underlying graph of $(V, X)$ must be a tree, whose leaves belong to $T$.

Given $s \in V$, [9,14] give algorithms for enumerating all directed trees connecting $s$ to all other nodes in the graph. We begin first by showing how to use the backtracking approach to enumerate all $(s, T)$-directed Steiner trees connecting a given node $s \in V$ to a given subset of nodes $T \subseteq V$. For this we need to verify that (F) is satisfied. Let $\mathcal{F}$ be the family of directed $(s, T)$-Steiner trees, and $\mathcal{F}'$ be the family of directed Steiner subtrees $\mathbf{T}'$ connecting $s$ to some subset of vertices $T' \subseteq T$, and such that every leaf of $\mathbf{T}'$ belongs to $T'$, except possibly one leaf $u$ which is not in $T$, but there is a path from $u$ to some $t \in T \setminus T'$ that does not use any vertex of $\mathbf{T}'$. Consider any two disjoint subsets $S_1 \in \mathcal{F}'$ and $S_2 \subseteq E$. Let $T'$ be the subset of $T$ reachable from $s$ by using only arcs from $S_1$. If the tree $(V, S_1)$ has a leaf vertex $u \notin T$, then any arc $a \in E \setminus (S_1 \cup S_2)$ satisfying $S_1 \cup \{a\} \in \mathcal{F}'$ must be of the form $(u, v)$. Otherwise, any arc $a = (u, v) \in E$ can be used to extend $S_1$ as long as $u$ is reachable from $s$ by a dipath in the graph $(V, S_1)$. In both cases, the check whether $S_1 \cup \{a\}$ can be extended to an $(s, T)$-directed Steiner tree that avoids $S_2$ can be done by simply deleting all the vertices appearing in $S_1$ and all the arcs appearing in $S_2$ from $G$ and checking the reachability of some vertex in $T \setminus T'$ from $v$.

Now we turn our attention to the case with two source vertices $s_1, s_2$ and sinks $t_1 \in V$ and $T_2 \subseteq V$. We assume that the graph $(V, E)$ contains a dipath connecting $s_1$ to $t_1$ and a directed Steiner tree connecting $s_2$ to every vertex in $T_2$. Otherwise there is nothing to generate. Such a condition is easy to verify.

Let us call a *minimal $\mathcal{P}$-linked subgraph*, MPLS in short, any minimal collection of arcs $X$ such that there is a directed path in $(V, X)$ from $s_1$ to $t_1$, and from $s_2$ to any node in $T_2$. Given a digraph $G$ and an $(s, T)$-Steiner tree $\mathbf{T}$ in $G$, we call a *cross path* any $(v_1, v_2)$-dipath in $(V, E \setminus E(\mathbf{T}))$, where $v_1$ and $v_2$ are two vertices such that there is no dipath from $v_2$ to $v_1$ in $\mathbf{T}$. The following property of MPLS holds.

**Lemma 1.** *Let $X \subseteq E$ be the union of an $(s_1, t_1)$-dipath $\mathbf{P}_1$ and an $(s_2, T_2)$-Steiner tree $\mathbf{T}_2$. Then $X$ is an MPLS if and only if $X$ has no cross paths.*

*Proof.* Suppose that $X$ is an MPLS, but has a cross path from $v_1$ to $v_2$. Consider the unique dipath $s_2 = x_0, x_1, \ldots, x_t = v_2$, in $\mathbf{T}_2$ from $s_2$ to $v_2$ and let $x_r$ be the last vertex on this path that belongs to $T_2 \cup \{v_1\}$, or lies on a path from $s_2$ to some vertex $t \in T_2$ that does not pass through $v_2$. Then, if we delete from $X$ all the arcs $(x_r, x_{r+1}), \ldots, (x_{t-1}, x_t)$, we would still have a path between $s_1$ and $t_1$ and between $s_2$ and every vertex in $T_2$. This contradicts the minimality of $X$.

Conversely, suppose that we drop an arc $a \in X = E(\mathbf{P}_1) \cup E(\mathbf{T}_2)$, and the graph $(V, X \setminus \{a\})$ still contains an $(s_1, t_1)$-dipath and an $(s_2, T_2)$-directed Steiner tree. If $a$ belongs to $\mathbf{P}_1$, then $X$ contains two distinct dipaths between $s_1$ and $t_1$, and if $a$ belongs to $\mathbf{T}_2$, then $X$ contains two distinct dipaths between $s_2$ and some $t \in T_2$. In both cases, it easy to see that $X$ must have a cross path. $\square$

The two following corollaries are immediate from Lemma 1.

**Corollary 1.** *Let $X$ be an MPLS that is composed of the union of a directed $(s_1, t_1)$-path $\mathbf{P}_1$ and a directed Steiner tree $\mathbf{T}_2$ connecting $s_2$ to every node in $T_2$. Suppose that $\mathbf{P}_1$ and $\mathbf{T}_2$ contain common vertices $v_1$ and $v_2$ such that $v_1$ lies on the path between $s_1$ and $v_2$ in $\mathbf{P}_1$, and on the path between $s_2$ and $v_2$ in $\mathbf{T}_2$, then $\mathbf{P}_1$ and $\mathbf{T}_2$ must contain exactly the same set of vertices between $v_1$ and $v_2$.*

**Corollary 2.** *In every MPLS $X$, the dipath from $s_1$ to $t_1$ is unique, and the dipath from $s_2$ to any $t \in T_2$ is unique.*

Yet, for more than two sources, Corollary 2 may not hold, as illustrated by the following example with three sources $\{s_1, s_2, s_3\}$ and three sinks $\{t_1, t_2, t_3\}$ (see Figure 3). Let $G$ be a digraph on 10 vertices $\{s_1, s_2, s_3, a, b, c, d, t_1, t_2, t_3\}$ and with arcs $\{(s_1, a), (s_2, b), (s_3, d), (b, t_3), (c, t_1), (d, t_2), (a, b), (b, c), (a, d), (d, c), (c, a)\}$. Clearly, in this example, there exists two dipaths between $s_1$ and $t_1$ namely, $s_1, a, b, c, t_1$ and $s_1, a, d, c, t_1$. Yet, all the arcs are needed to maintain the connectivity between $s_2$ and $t_2$ or between $s_3$ and $t_3$.

**Lemma 2.** *Every $(s_1, t_1)$-dipath can be extended to an MPLS, and every MPLS is an extension of some $(s_1, t_1)$-dipath.*

*Proof.* The first part of the lemma is a consequence of Lemma 1. Indeed, fix an $(s_1, t_1)$-dipath $\mathbf{P}_1$ and let $\mathbf{T}_2$ be an arbitrary directed $(s_2, T_2)$-Steiner tree. Let us modify the union $F = E(\mathbf{P}_1) \cup E(\mathbf{T}_2)$ by deleting arcs from $\mathbf{T}_2$ to obtain an MPLS $X \subseteq F$ as follows. For every two vertices $v_1$ and $v_2$ in $F$ such that there is

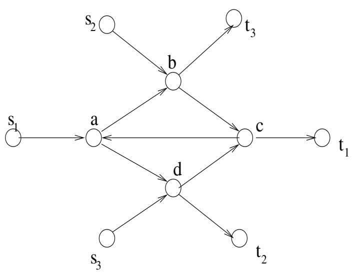

**Fig. 3.** An example showing that Corollary 2 does not hold with 3 sources.

a cross path from $v_1$ to $v_2$, consider the unique dipath $s_2 = x_1, x_2, \ldots, x_t = v_2$, in $\mathbf{T}_2$ from $s_2$ to $v_2$ and let $x_r \neq v_2$ be the last vertex on this path that belongs to $T_2 \cup \{v_1\}$, or lies on a path, in $\mathbf{T}_2$, from $s_2$ to some vertex $u \in T_2$ that does not pass through $v_2$. Then, we delete from $F$ all the arcs $(x_r, x_{r+1}), \ldots, (x_{t-1}, x_t)$. Clearly the resulting graph still contains an MPLS. We repeat this operation until there are no cross paths left. The resulting graph will be an MPLS by Lemma 1. To see the second part, note by Corollary 2 that an MPLS $X$ contains a unique path from $s_1$ to every $t \in T_2$. This readily defines an $(s_1, t_1)$-dipath of which $X$ is an extension. □

To prove Theorem 4, we apply backtracking as follows. First, we generate an $(s_1, t_1)$-dipath $\mathbf{P}_1$ connecting $s_1$ to $t_1$. We can generate all such paths in lexicographic order as explained above, and by Lemma 2 we know that each of them can be extended to an MPLS, and that all such possible extensions are exactly the set of MPLS's. Furthermore, the MPLS extensions $X$ and $X'$ of any two distinct $(s_1, t_1)$-dipaths $\mathbf{P}_1$ and $\mathbf{P}'_1$ are distinct. This follows from the fact that if $X = X'$ then both of them contain $\mathbf{P}_1 \cup \mathbf{P}'_1$, and hence they contain at least two different dipaths between $s_1$ and $t_1$, in contradiction to Corollary 2.

For each generated $(s_1, t_1)$-dipath $\mathbf{P}_1$, let us generate all possible extensions to an MPLS using backtracking. Let $\mathcal{F}$ be the family of all such possible extensions, and $\mathcal{F}'$ be the family of all subsets $X \subseteq E \setminus E(\mathbf{P}_1)$ of arcs such that (i) $X$ forms with some subset of arcs $Y \subseteq E(\mathbf{P}_1)$ a directed Steiner subtree $\mathbf{T}'_2$ connecting $s_2$ to some subset of vertices $T'_2 \subseteq T_2$, (ii) every leaf of $\mathbf{T}'_2$ belongs to $T'_2$, except possibly one leaf $u$ which is not in $T_2$, but there is a dipath in $G$ from $u$ to some $t \in T_2 \setminus T'_2$ that does not use any vertex of $\mathbf{T}'_2$, and (iii) there are no cross paths in $\mathbf{P}_1 \cup \mathbf{T}'_2$. By Lemma 1, any set $X \subseteq E$ satisfying (i), (ii) and (iii) can be extended to an MPLS, and therefore, all we need to check is that condition (F) is satisfied. Consider any two disjoint subsets $S_1 \in \mathcal{F}'$ and $S_2 \subseteq E$. Let $T'_2$ be the subset of $T_2$ reachable from $s_2$ by using only arcs from $S_1$. If the subgraph $(V, S_1 \cup E(\mathbf{P}_1))$ has a leaf vertex (that is a vertex with out-degree zero) $u \notin T_2 \cup \{t_1\}$, then any arc $a \in E \setminus (S_1 \cup S_2)$ satisfying $S_1 \cup \{a\} \in \mathcal{F}'$ must be of the form $(u, v)$. Otherwise, any arc $a = (u, v) \in E$ can be used to extend $S_1$ as long as $u$ is reachable from $s_2$ by a dipath in the graph $(V, S_1 \cup E(\mathbf{P}_1))$. In both cases, the check whether $S_1 \cup \{a\}$ can be extended to an MPLS that avoids $S_2$ can be done as follows. First, assume the path $\mathbf{P}_1$ consists of the vertices $s_1 = x_1, x_2, \ldots, x_r = t_1$ in that order, and let $\mathbf{P}$ be the path $s_2 = y_1, y_2, \ldots, y_k = v$ from $s_2$ to $v$ in $(V, S_1 \cup E(\mathbf{P}_1))$. If the paths $\mathbf{P}_1$ and $\mathbf{P}$ intersect and $\mathbf{P}$ is the *only* path, from $s_2$ to a node in $\mathbf{T}'_2$, that intersects $\mathbf{P}_1$, let $l \in \{1, \ldots, k\}$ be the largest index such that $y_l = x_t$ for some $t \in \{1, \ldots, r\}$. Then all vertices on the sub-path $x_{t+1}, \ldots, x_r = t_1$ must be deleted from $G$ to avoid cross paths. Otherwise, if there is a path in $\mathbf{T}'_2$, from $s_2$ to some $v' \neq v$, that intersects $\mathbf{P}_1$, then all the nodes on $\mathbf{P}_1$ must be deleted from $G$ to avoid cross paths. Next, we delete also from $G$ all the vertices appearing in $S_1$ and all the arcs in $S_2$. Finally, checking (F) reduces now to checking the reachability, from $v$, of some vertex in $T_2 \setminus T'_2$ in the remaining graph.

## References

1. U. Abel and R. Bicker, Determination of All Cutsets Between a Node Pair in an Undirected Graph, *IEEE Transactions on Reliability* **31** (1986) pp. 167–171.
2. E. Boros, K. Elbassioni, V. Gurvich and L, Khachiyan. Algorithms for Enumerating Circuits in Matroids, in *the Proceedings of the 14th Annual International Symposium on Algorithms and Computation (ISAAC 2003)*, *LNCS* 2906, pp. 485–494, Kyoto, Japan, December 2003.
3. E. Boros, K. Elbassioni, V. Gurvich and L. Khachiyan, On enumerating minimal dicuts and strongly connected subgraphs, to appear in *Proc. 10th Conf. on Integer Programming and Combinatorial Optimization (IPCO X)*, June 2004.
4. V.K. Bansal, K.B. Misra and M.P. Jain, Minimal Pathset and Minimal Cutset Using Search Technique, *Microelectr. Reliability* **22** (1982), pp. 1067–1075.
5. C. J. Colburn, The Combinatorics of Network Reliability, Oxford Univ. Press, New York, 1987.
6. N. D. Curet, J. DeVinney and M. E. Gaston, An efficient network flow code for finding all minimum cost *s-t* cutsets, *Comp. and Oper. Res.* **29** (2002), pp. 205–219.
7. D. Gusfield and D. Naor, Extracting maximum information about sets of minimum cuts, *Algorithmica* **10** (1993), pp. 64–89.
8. L. Lovász, Submodular functions and convexity, in *Mathematical Programming: The State of the Art*, Bonn 1982, pp. 235–257, (Springer Verlag, 1983).
9. S. Kapoor and H. Ramesh, An Algorithm for Enumerating All Spanning Trees of a Directed Graph, *Algorithmica* 27(2), pp. 120–130, 2000.
10. J. Oxley, Matroid Theory, Oxford University Press, Oxford, 1992.
11. J. S. Provan and M. O. Ball, Computing Network Reliability in Time Polynomial in the Number of Cuts, *Operations Research* **32** (1984), pp. 516–526.
12. J. S. Provan and D. R. Shier, A paradigm for listing $(s, t)$ cuts in graphs, *Algorithmica* **15**, (1996), pp. 351–372.
13. R. C. Read and R. E. Tarjan, Bounds on backtrack algorithms for listing cycles, paths, and spanning trees, *Networks*, 5 (1975) pp. 237–252.
14. T. Uno, An Algorithm for Enumerating all Directed Spanning Trees in a Directed Graph, in *the Proceedings of the 7th Annual International Symposium on Algorithms and Computation ISAAC*, 1996, pp. 166–173
15. V. Vazirani, Approximation Algorithms, Springer-Verlag, Berlin, 2001.
16. D.J.A. Welsh, Matroid Theory, Academic Press, London, New York, San Francisco, 1976.