Generating Minimal k-Vertex Connected Spanning Subgraphs

Endre Boros[1], Konrad Borys[1], Khaled Elbassioni[2], Vladimir Gurvich[1], Kazuhisa Makino[3], and Gabor Rudolf[1]

[1] RUTCOR, Rutgers University, 640 Bartholomew Road, Piscataway NJ 08854-8003; ({boros,kborys,gurvich,grudolf}@rutcor.rutgers.edu)
[2] Max-Planck-Institut für Informatik, Saarbrücken, Germany; (elbassio@mpi-sb.mpg.de)
[3] Division of Mathematical Science for Social Systems, Graduate School of Engineering Science, Osaka University, Toyonaka, Osaka, 560-8531, Japan; (makino@sys.es.osaka-u.ac.jp)

Abstract. We show that minimal k-vertex connected spanning subgraphs of a given graph can be generated in incremental polynomial time for any fixed k.

1 Introduction

Vertex and edge connectivity are two of the most fundamental concepts in network reliability theory. While in the simplest case only the connectedness of an undirected graph, that is, the presence of a spanning tree, is required, in practical applications higher levels of connectivity are often desirable. Given the possibility that the edges of the network can randomly fail the reliability of the network is defined as the probability that the operating edges provide a certain level of connectivity. Most methods computing network reliability depend on the efficient generation of all (or many) minimal subsets of network edges which guarantee the required connectivity [Cou87,Val79].

In this paper we consider the problems of generating minimal k-vertex connected spanning subgraphs. An undirected graph G on at least $k+1$ vertices is *k-vertex connected* if every subgraph of G obtained by removing at most $k-1$ vertices is connected. A subgraph of a graph G is *spanning* if it has the same vertex set as G.

For a fixed integer k we define the problem of generating minimal k-vertex connected spanning subgraphs as follows:

Input: A k-vertex connected graph G

Output: The list of all minimal k-vertex connected spanning subgraphs of G

Note that the output of the above problem may consist of exponentially many subgraphs in terms of the input size. Thus, the efficiency of generation algorithms is measured customarily in both the input and output size (see e.g., [Val79,LLK80,JP88]). An algorithm generating all elements of a family $\mathcal{F}$ is said to run in *incremental polynomial time* if generating K elements of $\mathcal{F}$ (or all if $\mathcal{F}$ has less than K elements) can be done in time polynomial in K and the size of the input, for an arbitrary integer K.

Our problems include as a special case the problem of generating spanning trees ($k = 1$), which can be solved efficiently [RT75,GM78,Mat93,ASU97]. The problem of generating 2-vertex connected subgraphs and its generalization for matroids has been considered in [KBB$^+$].

1.1 Main Results

We show that this generation problem can be solved in incremental polynomial time.

Theorem 1. *For every K we can generate K minimal k-vertex connected spanning subgraphs of a given graph in $O(K^3m^3n + K^2m^5n^4 + Kn^km^2)$ time, where $n = |V|$, $m = |E|$.*

We remark that the running time of our algorithm depends exponentially on k. The complexity of the above problem when k is also part of the input remains an open question.

1.2 The $X - e + Y$ method

In this section we recall a technique from [KBB$^+$05], which is a variant of the supergraph approach introduced by [SS02]. Let $\mathcal{C}$ be a class of finite sets and for every $E \in \mathcal{C}$ let $\pi : 2^E \to \{0, 1\}$ be a monotone Boolean function, i.e., one for which $X \subseteq Y$ implies $\pi(X) \le \pi(Y)$. We assume that $\pi(\emptyset) = 0$ and $\pi(E) = 1$. Let

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

Our goal is to generate all sets belonging to $\mathcal{F}$.

We remark that for every $X \subseteq E$ for which $\pi(X) = 1$ we can derive a subset $Y \subseteq X$ such that $Y \in \mathcal{F}$, by evaluating π exactly $|X|$ times. This can be accomplished by deleting one-by-one elements of X whose removal does not change the value of π. To formalize this, we can fix an arbitrary linear order $\prec$ on elements of E, without any loss of generality, and define a mapping $Project : \{X \subseteq E \mid \pi(X) = 1\} \to \mathcal{F}$ by $Project(X) = X \setminus Z$, where Z is the lexicographically first subset of X, with respect to $\prec$, such that $\pi(X \setminus Z) = 1$ and $\pi(X \setminus (Z \cup e)) = 0$ for every $e \in X \setminus Z$. Clearly, by trying to delete elements of X in their $\prec$-order, we can compute $Project(X)$, as we remarked above, by evaluating π exactly $|X|$ times.

We next introduce a directed graph $\mathcal{G} = (\mathcal{F}, \mathcal{E})$ on vertex set $\mathcal{F}$. We define the neighborhood $N(X)$ of a vertex $X \in \mathcal{F}$ as follows $N(X) = \{Project((X \setminus e) \cup Y) \mid e \in X, Y \in \mathcal{Y}_{X,e}\}$, where $\mathcal{Y}_{X,e}$ is defined by $\mathcal{Y}_{X,e} = \{Y \mid Y$ is a minimal subset of $E \setminus X$ satisfying $\pi((X \setminus e) \cup Y) = 1\}$.

In other words, for every set $X \in \mathcal{F}$ and for every element $e \in X$ we extend $X \setminus e$ in all possible minimal ways to a set $X' = (X \setminus e) \cup Y$ for which $\pi(X') = 1$

(since $X \in \mathcal{F}$, we have $\pi(X \setminus e) = 0$), and introduce each time a directed arc from X to $Project(X')$. We call the obtained directed graph $\mathcal{G}$ the *supergraph* of our generation problem.

Proposition 1 ([KBB+05]). *The supergraph $\mathcal{G} = (\mathcal{F}, \mathcal{E})$ is strongly connected.* □

Since $\mathcal{G}$ is strongly connected by performing a breadth-first search in $\mathcal{G}$ we can generate all elements of $\mathcal{F}$. Thus, given two procedures:

- $First(X, e)$, which for every $X \in \mathcal{F}$ and $e \in X$ returns an element of $\mathcal{Y}_{X,e}$ if $\mathcal{Y}_{X,e} \neq \emptyset$ and $\emptyset$ otherwise,
- $Next(\mathcal{Y}, X, e)$, which return an element of $\mathcal{Y}_{X,e} \setminus \mathcal{Y}$ if $\mathcal{Y}_{X,e} \neq \mathcal{Y}$ and $\emptyset$ otherwise,

the procedure $Transversal(\mathcal{G})$, defined below, generates all elements of $\mathcal{F}$.

$Traversal(\mathcal{G})$

Find an initial vertex $X^0 \leftarrow Project(E)$, initialize a queue $\mathcal{Q} = \emptyset$ and a dictionary of output vertices $\mathcal{D} = \emptyset$.

Perform a breadth-first search of $\mathcal{G}$ starting from X^o:

output X^0 and insert it to $\mathcal{Q}$ and to $\mathcal{D}$
while $\mathcal{Q} \neq \emptyset$ **do**
  take the first vertex X out of the queue $\mathcal{Q}$
  for every $e \in X$ **do**
    $\mathcal{Y} \leftarrow \emptyset$, $Y \leftarrow First(X, e)$
    while $Y \neq \emptyset$ **do**
      compute the neighbor $X' \leftarrow Project((X \setminus e) \cup Y)$
      if $X' \notin \mathcal{D}$ **then output** X' and insert it to $\mathcal{Q}$ and to $\mathcal{D}$
      add Y to $\mathcal{Y}$, $Y \leftarrow Next(\mathcal{Y}, X, e)$

Proposition 2. *Assume that the procedure $First(X, e)$ works in time $O(\phi_1(E))$, the for every K procedure $Next(\mathcal{Y}, X, e)$ outputs K elements of $\mathcal{Y}_{X,e}$ in time $\phi_2(K, E)$ and there is an algorithm evaluating π in time $O(\gamma(E))$. Then $Traversal(\mathcal{G})$ outputs K elements of $\mathcal{F}$ in time $O(K^2|E|^2\gamma(E) + K^2 log(K)|E|^2 + K|E|\phi_2(K, E) + K|E|\phi_1(E))$.*

2 Proof of Theorem 1

In this section we apply the $X - e + Y$ method to the generation of all minimal k-vertex connected spanning subgraphs.

For a given k-vertex connected graph (V, E) we define a Boolean function π as follows: for a subset $X \subseteq E$ let

$$\pi(X) = \begin{cases} 1, \text{ if (V,X) is } k\text{-vertex connected;} \\ 0, \text{ otherwise.} \end{cases}$$

Clearly π is monotone, $\pi(\emptyset) = 0$, $\pi(E) = 1$. Then $\mathcal{F} = \{X \mid X \subseteq E$ is a minimal set satisfying $\pi(X) = 1\}$ is the family of edge sets of all minimal k-vertex connected spanning subgraphs of (V, E).

2.1 $(k-1)$-separators of $(V, X \setminus e)$

Before describing procedures $First(X, e)$ and $Next(\mathcal{Y}, X, e)$ we need the additional notions and elementary results.

A *k-separator* of a graph is a set of k vertices whose removal (simultaneously removing all edges adjacent to those vertices) makes the graph no longer connected. Note that a k-vertex connected graph has no k'-separators for $k' < k$.

Let $G = (V, X)$ be a minimal k-vertex connected spanning subgraph of a k-vertex connected graph (V, E) (see Figure 1).

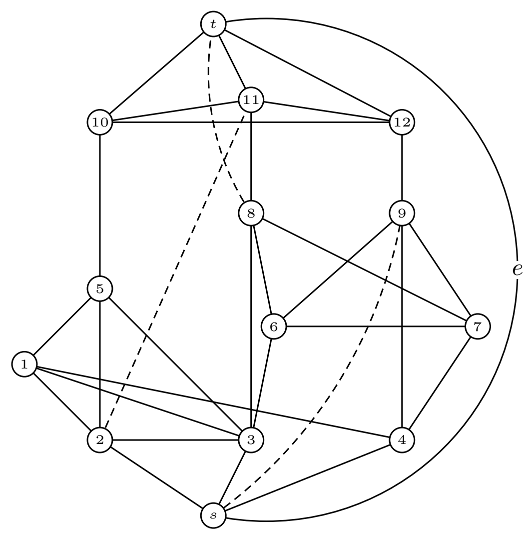

Fig. 1. 4-vertex connected graph (V, E) and its minimal 4-vertex connected subgraph $G = (V, X)$. Solid lines are edges in X.

Let $e = st$ be an arbitrary edge of G and let W be a $(k-1)$-separator of $G_e = (V, X \setminus e)$. Note that W contains neither s nor t, since otherwise W would also be a $(k-1)$-separator of G. We denote by S_W and T_W the vertex sets of the components (i.e., maximal connected subgraphs) of $G_e[V \setminus W]$ containing s and t, respectively.

Claim. $G_e[V \setminus W]$ consists of two components, $G_e[S_W]$ and $G_e[T_W]$ (see Figure 2).

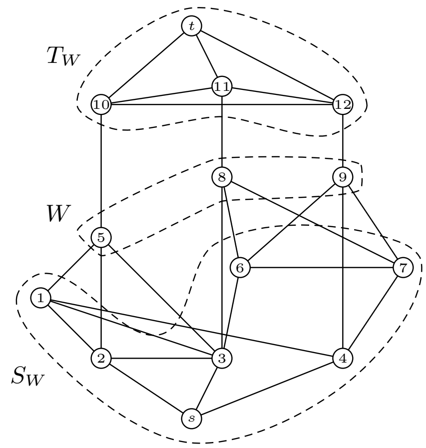

Fig. 2. 3-separator $W = \{5, 8, 9\}$ and the corresponding 3-source $S_W = \{s, 1, 2, 3, 4, 6, 7\}$.

We denote by $N(\cdot)$ a neighborhood in the graph G_e. Let $\mathcal{W}$ be the set of all $(k-1)$-separators of $G_e = (V, X \setminus e)$ and let $\mathcal{S} = \{S \subseteq V \mid |N(S)| = k-1, s \in S, t \notin S \cup N(S)\}$. We call an element of $\mathcal{S}$ a $(k-1)$-*source*. Note that the mapping $W \longmapsto S_W$ is a bijection between $\mathcal{W}$ and $\mathcal{S}$ whose inverse is $S \longmapsto N(S)$.

For two vertices $u, v \in V$ let $D_{u,v} = \{S_W \in \mathcal{S} \mid u \in S_W, v \in T_W\}$. For an edge $f = uv$ let $D_f = D_{u,v} \cup D_{v,u}$

We call a set of hyperedges whose union contains every vertex a *hyperedge cover*. We show that elements of $\mathcal{Y}_{X,e}$ are in one to one correspondence with the minimal hyperedge covers of $\mathcal{H}_{X,e}$.

Claim. Let $Y \subseteq E \setminus X$. The graph $(V, X \setminus e \cup Y)$ is k-vertex connected if and only if $\bigcup_{f \in Y} D_f = \mathcal{S}$.

2.2 Procedures $\boldsymbol{First(X, e)}$ and $\boldsymbol{Next(\mathcal{Y}, X, e)}$

We describe $First(X, e)$ and $Next(\mathcal{Y}, X, e)$, procedures generating all elements of $\mathcal{Y}_{X,e}$.

$First(X, e)$

1 construct a hypergraph $\mathcal{H}_{X,e}$ on vertex set $\mathcal{S}$ with edge set $\mathcal{E} = \{D_f \mid f \in E \setminus X\}$
2 find a minimal hyperedge cover $\mathcal{C}$ of $\mathcal{H}_{X,e}$
3 return a set $\{f \mid D_f \in \mathcal{C}\}$

$Next(\mathcal{Y}, X, e)$

1 find a a minimal hyperedge cover $\mathcal{C}$ of $\mathcal{H}_{X,e}$ not in $\{D_f \mid f \in Y, Y \in \mathcal{Y}\}$
2 return a set $\{f \mid D_f \in \mathcal{C}\}$

In the remainder of this section we show that we can generate minimal hyperedge covers of $\mathcal{H}_{X,e}$ efficiently.

2.3 Structure of $(k-1)$-separators

Consider the poset $L = (\mathcal{S}, \subseteq)$ of the $(k-1)$-sources ordered by inclusion.

Proposition 3. *The poset L with operations $\cap$ and $\cup$ is a lattice.*

We show that the ordering of $(k-1)$-sources in L has a natural interpretation for the corresponding $(k-1)$-separators.

Since the graph G_e is $(k-1)$-vertex connected, by Menger's Theorem it contains $k-1$ internally vertex disjoint s-t paths. Let $P_1 = sv_1^1 \dots v_{l_1}^1 t$, $P_2 = sv_1^2 \dots v_{l_2}^2 t$, $\dots$, $P_{k-1} = sv_1^{k-1} \dots v_{l_{k-1}}^{k-1} t$ denote such a collection of paths (see Figure 3). We denote by V_P the set of all vertices belonging to the paths $P_1, \dots, P_{k-1}$. Note that not all vertices in V necessarily belong to V_P.

Consider a $(k-1)$-separator W. Since the removal of W disconnects G_e, W contains at least one internal vertex from each path P_i, $i = 1, \dots, k-1$. As W has $k-1$ vertices, $W = \{v_{\alpha(W,1)}^1, \dots, v_{\alpha(W,k-1)}^{k-1}\}$, where $\alpha(W, i)$ is the index of the vertex of P_i belonging to W.

Claim. Let $W, U \in \mathcal{W}$. $S_W \subseteq S_U$ if and only if $\alpha(W, i) \leq \alpha(U, i)$ for all $i = 1, \dots, k-1$.

Lemma 1. *Let S_W, S_U be $(k-1)$-sources of G_e. Either $S_W \cap T_U = \emptyset$ or $T_W \cap S_U = \emptyset$.*

Proof. We partition $\{1, \dots, k-1\}$ into sets I, J and K as follows: $I = \{i \mid \alpha(W, i) > \alpha(U, i)\}$, $J = \{i \mid \alpha(W, i) = \alpha(U, i)\}$, $K = \{i \mid \alpha(W, i) < \alpha(U, i)\}$.

Let $C = \{v_{\alpha(U,i)}^i \mid i \in I\} \cup \{v_{\alpha(W,i)}^i \mid i \in I\} \cup \{v_{\alpha(W,i)}^i \mid i \in J\}$. Observe that $|C| = 2|I| + |J|$.

We show that $N(S_W \cap T_U) \subseteq C$. Note that $V \setminus ((S_W \cap T_U) \cup C) = T_W \cup S_U$ (see Figure 4). Since W and U are $(k-1)$-separators of G_e, there is no edge between $S_W \cap T_U$ and $T_W \cup S_U$, thus $N(S_W \cap T_U) \subseteq C$.

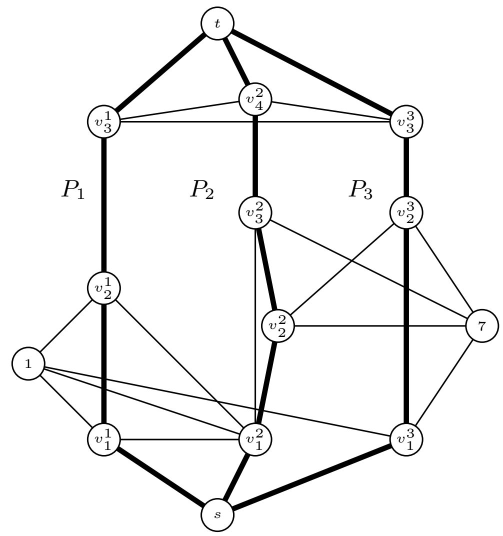

Fig. 3. Internally vertex disjoint paths P_1, P_2, P_3 of G_e represented by thick edges.

Let $D = \{v^i_{\alpha(U,i)} \mid i \in K\} \cup \{v^i_{\alpha(W,i)} \mid i \in K\} \cup \{v^i_{\alpha(W,i)} \mid i \in J\}$. Similarly, we obtain that $N(T_W \cap S_U) \subseteq D$.

Suppose for contradiction that $S_W \cap T_U \neq \emptyset$ and $T_W \cap S_U \neq \emptyset$. Since $S_W \cap T_U$ contains neither s nor t, the removal of $N(S_W \cap T_U)$ disconnects G. As G is k-vertex connected, we obtain $k \leq |N(S_W \cap T_U)| \leq |C|$, thus $2|I| + |J| \geq k$. Similarly, we have $2|K| + |J| \geq k$. Recall that I, J and K partition $\{1, \ldots, k-1\}$, thus $k - 1 = |I| + |J| + |K|$.

Combining this with the above inequalities we obtain $2((k-1) + |I| + |J| + |K|) \geq 2(k + |I| + |J| + |K|)$, a contradiction. □

2.4 Bounding The Number of $(k-1)$-sources

It is easy to see that the numebr of $(k-1)$-sources is at most $\binom{|V|}{k-1}$, since each one corresponds to different $(k-1)$-separators. In this section we provide a better bound on this number.

Corollary 1. *If S_W and S_U are incomparable in L then there exists some $i \in \{1, \ldots, k-1\}$ such that $|\alpha(W,i) - \alpha(U,i)| = 1$, i.e., the vertices $v^i_{\alpha(W,i)}$ and $v^i_{\alpha(U,i)}$ are adjacent on the path P_i.*

Proof. Suppose on the contrary that $|\alpha(W,i) - \alpha(U,i)| > 1$ for all $i = 1, \ldots, k-1$. Then since S_W and S_U are incomparable, by Claim 2.3 there exist $j, l \in \{1, \ldots, k-1\}$ such that $\alpha(U,j) + 1 < \alpha(W,j)$ and $\alpha(W,l) + 1 < \alpha(U,l)$. Then $v^j_{\alpha(U,j)+1} \in S_W \cap T_U$, $v^l_{\alpha(W,l)+1} \in T_W \cap S_U$ contradicting Lemma 1. □

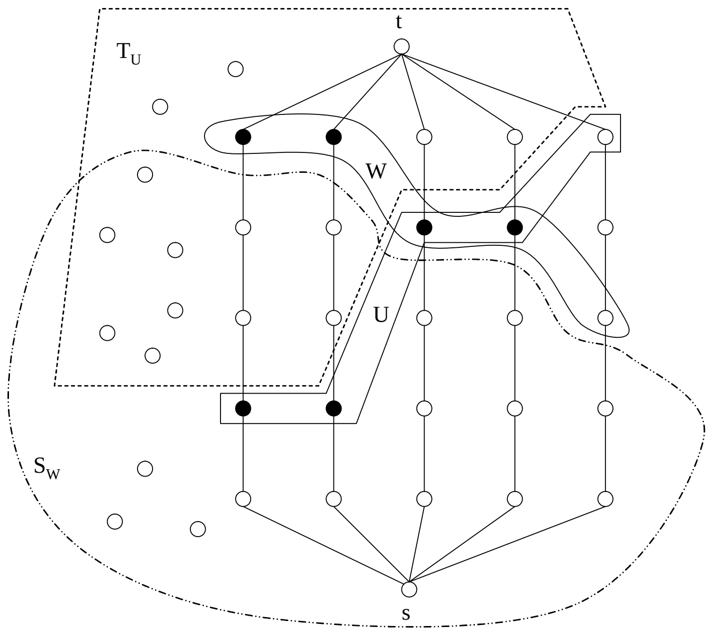

Fig. 4. $(k-1)$-separators W and U. Black nodes are vertices of C.

The *width* of a poset is the size of its largest antichain. We show that the width of L is bounded.

Proposition 4. *The width of L is at most 2^{k-1}.*

Proof. We associate to every $(k-1)$-separator W a 0-1 vector $\pi(W) = (\alpha(W,1) \mod 2, \ \dots \ , \alpha(W, k-1) \mod 2)$. By Corollary 1, if two $(k-1)$-separators W, U are incomparable, there exists some $i \in \{1, \dots, k-1\}$ such that $|\alpha(W,i) - \alpha(U,i)| = 1$, implying $\pi(W) \neq \pi(U)$.

Since the number of different 0-1 vectors of length $k-1$ is 2^{k-1}, every antichain in P has size at most 2^{k-1}. □

Corollary 2. *For every fixed k the number of $(k-1)$-sources is $O(|V|)$.*

2.5 Generating Minimal Hyperedge Covers of $\mathcal{H}_{X,e}$

In this section we reduce the problem of generating minimal hyperedge covers of $\mathcal{H}_{X,e}$ to the problem of generating minimal transversals of 2-conformal hypergraphs. For the latter problem the algorithm is provided in [BEGK04].

A *transversal* is a set of vertices intersecting every hyperedge. A hypergraph is δ-*conformal* if its transpose is δ-Helly (see [Ber89] for other equivalent definitions).

First we show that the hypergraphs $\mathcal{H}_{X,e}$ are 2-Helly.

Claim. Either $D_{u,v} = \emptyset$ or $D_{v,u} = \emptyset$ for all $u, v \in V$.

Proof. Suppose on the contrary that we have $S_W \in D_{u,v}$ and $S_U \in D_{v,u}$. Then $u \in S_W \cap T_U$ and $v \in T_W \cap S_U$, contradicting Lemma 1. □

Claim. D_f is a sublattice of L (see Figure 5).

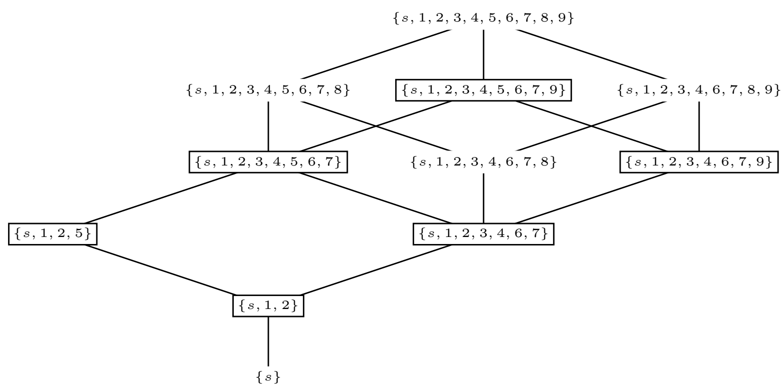

Fig. 5. Elements of $D_{2,11}$ are in black rectangles. Note that $D_{11,2} = \emptyset$.

Proof. Let $f = uv$. Without loss of generality we can assume that $D_f = D_{u,v}$. Let $S, S' \in D_{u,v}$. Then $u \in S \cap S'$ and $v \notin (S \cap S') \cup N(S \cap S')$. Similarly, $u \in S \cup S'$ and $v \notin S \cup S' \cup N(S \cup S')$. Thus $S \cap S', S \cup S' \in D_f$. □

Since the edges of $\mathcal{H}_{X,e}$ are sublattices of L, the hypergraphs $\mathcal{H}_{X,e}$ are 2-Helly ([Ber89, Example 2 on page 21]). Thus the hypergraphs $\mathcal{H}^T_{X,e}$ are 2-conformal.

Note that minimal hyperedge covers of $\mathcal{H}_{X,e}$ are minimal transversals of $\mathcal{H}^T_{X,e}$. An algorithm from [BEGK04] generates K minimal transversals of δ-conformal hypergraph in $O(K^2 i^2 j + K i^{\delta+2} j^{\delta+2})$, where i and j are the numeber of vertices and number of hyperedges, respectively.

2.6 Complexity

In this section we analyze the complexity of $Traversal(\mathcal{G})$. Let $n = |V|$, $m = |E|$. Since G is k-vertex connected we have $m \geq n$.

Note that $\pi(X)$ can be evaluated in $O(k^3|V|^2)$ time [CKT93], thus $\gamma(E) = n^2$.

Claim. For every $X \in \mathcal{F}$ and $e \in X$ the hypergraph $\mathcal{H}_{X,e} = (\mathcal{S}, \mathcal{E})$ has $O(n)$ vertices and $O(m)$ edges and it can be constructed in $O(n^k m)$ time.

Proof of Claim 6: By Corollary 2 the number of vertices of $\mathcal{H}_{X,e} = (\mathcal{S}, \mathcal{E})$ is at most $O(n)$. The number of edges of $\mathcal{H}_{X,e} = (\mathcal{S}, \mathcal{E})$ is exactly $|E \setminus X| \leq m$ since we add an edge to $\mathcal{H}_{X,e} = (\mathcal{S}, \mathcal{E})$ for every edge of $E \setminus X$.

To construct $\mathcal{H}_{X,e} = (\mathcal{S}, \mathcal{E})$ we first need to find all $(k-1)$-sources and $(k-1)$-separators. We can check if after removing a given set of $k-1$ vertices the graph G is still connected in $O(n+m)$ time using, e.g., depth first search. Thus we can find all $(k-1)$-separators by repeating the above procedure for every $(k-1)$-element subset of V. The number of such subsets is $\binom{n}{k-1} \leq n^{k-1}$. Thus we can compute all $(k-1)$-sources and $(k-1)$-separators in $O(n^{k-1}m)$ time.

To add edges we need to check for every $f \in E \setminus X$ and every $(k-1)$-separator W if S_W belongs to D_f, which can be done in $O(n)$ time for each pair f and W. Thus the complexity of constructing edges of $\mathcal{H}$ is $O(n^2 m)$. $\square$

Since we can find a minimal transversal of $\mathcal{H}^T_{X,e}$ in $O(|\mathcal{E}|)$ time and by Claim 2.6 we have $\phi_1(E) = n^k m$. Recall that $\phi_2(K, E) = K^2 m^2 n + K m^4 n^4$ (see Section 2.5). Thus by Proposition 2 the complexity of $Traversal(\mathcal{G})$ is $O(K^3 m^3 n + K^2 m^5 n^4 + K n^k m^2)$.

References

[ASU97] A. Tamura A. Shioura and T. Uno. An optimal algorithm for scanning all spanning trees of undirected graphs. *SIAM Journal on Computing*, 26(3):678–692, 1997.

[BEGK04] E. Boros, K. Elbassioni, V. Gurvich, and L. Khachiyan. Generating maximal independent sets for hypergraphs with bounded edge-intersections. In Martin Farach-Colton, editor, *LATIN 2004: Theoretical Informatics, 6th Latin American Symposium*, volume 2976 of *Lecture Notes in Computer Science*, pages 488–498, Buenos Aires, Argentina, April 2004. Springer.

[Ber89] C. Berge. *Hypergraphs*. Elsevier-North Holand, Amsterdam, 1989.

[CKT93] J. Cheriyan, M. Y. Kao, and R. Thurimella. Algorithms for parallel k-vertex connectivity and sparse certificates. 22:157–174, 1993.

[Cou87] C.J. Coulbourn. *The Combinatorics of Network Reliability*. Oxford University Press, 1987.

[GM78] H.N. Gabow and E.W. Myers. Finding all spanning trees of directed and undirected trees. *SIAM Journal on Computing*, 117:280–287, 1978.

[JP88] D.S. Johnson and Ch. H. Papadimitriou. On generating all maximal independent sets. *Information Processing Letters*, 27:119–123, 1988.

[KBB+] L. Khachiyan, E. Boros, K. Borys, K. Elbassioni, V. Gurvich, and K. Makino. Enumerating spanning and connected subsets in graphs and matroids. Manuscript.

[KBB+05] L. Khachiyan, E. Boros, K. Borys, K. Elbassioni, V. Gurvich, and K. Makino. Generating cut conjunctions and bridge avoiding extensions in graphs. In *Algorithms and Computation: 16th International Symposium, ISAAC 2005, Sanya, Hainan, China, December 19-21, 2005*, pages 156–165, 2005.

[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.

[Mat93] T. Matsui. Algorithms for finding all the spanning trees in undirected graphs. Technical report, Department of Mathematical Engineering and Information Physics, Faculty of Engineering, University of Tokyo, 1993. Report METR93-08.

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

[SS02] B. Schwikowski and E. Speckenmeyer. On enumerating all minimal solutions of feedback problems. *Discrete Applied Mathematics*, 117:253–265, 2002.

[Val79] L. Valiant. The complexity of enumeration and reliability problems. *SIAM Journal on Computing*, 8:410–421, 1979.