Data Structures for Node Connectivity Queries

Zeev Nutov
The Open University of Israel, Ra'anana, Israel

Abstract

Let $\kappa(s,t)$ denote the maximum number of internally disjoint st-paths in an undirected graph G. We consider designing a data structure that includes a list of cuts, and answers the following query: given $s,t \in V$, determine whether $\kappa(s,t) \leq k$, and if so, return a pointer to an st-cut of size $\leq k$ (or to a minimum st-cut) in the list. A trivial data structure that includes a list of $n(n-1)/2$ cuts and requires $\Theta(kn^2)$ space can answer each query in $O(1)$ time. We obtain the following results.

- In the case when G is k-connected, we show that $2n$ cuts suffice, and that these cuts can be partitioned into $2k+1$ laminar families. Thus using space $O(kn)$ we can answers each min-cut query in $O(1)$ time, slightly improving and substantially simplifying the proof of a recent result of Pettie and Yin [18]. We then extend this data structure to subset k-connectivity.
- In the general case we show that $(2k+1)n$ cuts suffice to return an st-cut of size $\leq k$, and a list of size $k(k+2)n$ contains a minimum st-cut for every $s,t \in V$. Combining our subset k-connectivity data structure with the data structure of Hsu and Lu [7] for checking k-connectivity, we give an $O(k^2n)$ space data structure that returns an st-cut of size $\leq k$ in $O(\log k)$ time, while $O(k^3n)$ space enables to return a minimum st-cut.

2012 ACM Subject Classification Theory of computation → Design and analysis of algorithms

Keywords and phrases node connectivity, minimum cuts, data structure, connectivity queries

Digital Object Identifier 10.4230/LIPIcs.ESA.2022.82

Related Version *Previous Version*: https://arxiv.org/abs/2110.09102

1 Introduction

Let $\kappa(s,t) = \kappa_G(s,t)$ denote the maximum number of internally disjoint st-paths in a graph $G = (V,E)$. W.l.o.g. we will assume that G is connected. An ***st*-cut** is a subset $Q \subseteq V \cup E$ such that $G \setminus Q$ has no st-path. By Menger's Theorem, $\kappa(s,t)$ equals to the minimum size of an st-cut, and there always exists a minimum st-cut that contains no edge except of st. We consider designing a compact data structure that given $s,t \in V$ and $k < n = |V|$ answers the following k-bounded connectivity/cut queries.

$\text{PCON}_k(s,t)$	(partial connectivity query):	Determine whether $\kappa(s,t) \leq k$.
$\text{PCUT}_k(s,t)$	(partial cut query):	If $\kappa(s,t) \leq k$ then return an st-cut of size $\leq k$.
$\text{CON}_k(s,t)$	(connectivity query):	Return $\min\{\kappa(s,t), k+1\}$.
$\text{CUT}_k(s,t)$	(min-cut query):	If $\kappa(s,t) \leq k$ then return a minimum st-cut.

The query $\text{PCUT}_k(s,t)$ requires $\Theta(k)$ time just to write an st-cut. However, by slightly relaxing the definition, we allow the data structure to include a list of cuts, and to return just a pointer to an st-cut of size $\leq k$ in the list. How short can this list be? By choosing a minimum st-cut for each pair $\{s,t\}$, one gets a list of $n(n-1)/2$ cuts. This gives a trivial data structure, that answers both queries in $O(1)$ time, but requires $\Theta(kn^2)$ space – just store the pairwise connectivities in an $n \times n$ matrix, with pointers to the relevant $O(n^2)$ cuts. For edge connectivity, the Gomory-Hu Cut-Tree [5] shows that there exists such a list of $n-1$ cuts that form a laminar family. However, no similar result is known for the node connectivity case considered here.

30th Annual European Symposium on Algorithms (ESA 2022).
Editors: Shiri Chechik, Gonzalo Navarro, Eva Rotenberg, and Grzegorz Herman; Article No. 82; pp. 82:1–82:12
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

A graph is ***k*-connected** if $\kappa(s,t) \geq k$ for all $s,t \in V$. Recently, Pettie and Yin [18], and earlier in the 90's Cohen, Di Battista, Kanevsky, and Tamassia [3], considered the above problem in k-connected graphs. Pettie and Yin [18] suggested for $n \geq 4k$ an $O(kn)$ space data structure, that answers $\text{CON}_k(s,t)$ in $O(1)$ time and $\text{CUT}_k(s,t)$ in $O(k)$ time; they showed that it can be constructed in $\tilde{O}(m + \mathsf{poly}(k)n)$ time. The arguments in [18] are complex, and here by a simpler proof we obtain the following improvement on the $\text{CUT}_k(s,t)$ query.

▶ **Theorem 1.** *For any k-connected graph, there exists an $O(kn)$ space data structure, that includes a list of $2n$ cuts, that answers $\text{CON}_k(s,t)$ and $\text{CUT}_k(s,t)$ queries in $O(1)$ time.*

All our data structures can be constructed in polynomial time; we will not discuss designing efficient construction algorithms here. We note that the Pettie-Yin [18] data structure also includes a list of $O(n)$ cuts, and maybe it can be modified to return a pointer to a minimum st-cut in $O(1)$ time. In any case, our data structure and arguments are simpler, and we sketch the main ideas below.

- We observe that there exists a forest F, such that for any $st \in E$ with $\kappa(s,t) = k$, either one of s,t has degree k (and a minimum st-cut is the set of k edges incident to s or to t), or $st \in F$; this follows from Mader's Critical Cycle Theorem [12]. Thus for pairs $st \in E$ there are at most n relevant cuts, and it is not hard to see that matching between a pair s,t and its cut can be done in $O(1)$ time.
- For $A \subseteq B$ let ∂A denote the set of neighbors of A in G. Let $\mathcal{C}$ be a family of sets obtained by picking for every node v an inclusion minimal set C_v with $v \in C_v$, $|\partial C_v| = k$, and $|C_v| \leq (n-k)/2$, if such a set exists (we show that C_v is unique, if exists). For any $st \notin E$ with $\kappa(s,t) = k$, a minimum st-cut $Q \subset V$ has a connected component of $G \setminus Q$ of size $\leq (n-k)/2$ that contains s or t; thus $C_s \in \mathcal{C}$ or $C_t \in \mathcal{C}$. To store $\mathcal{C}$ in $O(kn)$ space, we will show that $\mathcal{C}$ can be partitioned into $O(k)$ laminar families (each laminar family can be represented by a tree). To check whether $\kappa(s,t) = k$ we just need to check that $t \notin C_s \cup \partial C_s$ or $s \notin C_t \cup \partial C_s$; this can be done in $O(1)$ time, using a data structure that answers in $O(1)$ time ancestor/descendant queries in trees.

For a set $S \subseteq V$ of terminals we say that a graph is ***k*-*S*-connected** if $\kappa(s,t) \geq k$ for all $s,t \in S$. We will extend Theorem 1 to k-S-connected graphs as follows.

▶ **Theorem 2.** *For any k-S-connected graph with $|S| \geq 3k$, there exists an $O(k|S|)$ space data structure, that includes a list of $3|S|$ cuts, and answers $\text{CON}_k(s,t)$ and $\text{CUT}_k(s,t)$ queries for node pairs in S in $O(1)$ time.*

For arbitrary graphs, a trivial data structure that answers $\text{CON}_k(s,t)$ and $\text{CUT}_k(s,t)$ queries in $O(1)$ time uses $\Theta(kn^2)$ space. Hsu and Lu [7] showed that there exists an auxiliary directed graph $H = (V,F)$ and an ordered partition $S_1, S_2, \ldots$ of V, such that:

- Every part S_i has at most $2k-1$ neighbors in H, and all of them are in $S_{i+1} \cup S_{i+2} \cup \cdots$.
- $\kappa(s,t) \geq k+1$ iff s,t belong to the same part, or $st \in F$, or $ts \in F$.

They also gave a polynomial time algorithm for constructing such H. Augmenting H by a perfect hashing data structure enables to answer "$st \in F$?" queries in $O(1)$ time. Since $|F| = O(kn)$, this gives an $O(kn)$ space[1] data structure that determines whether $\kappa(s,t) \geq k+1$ in $O(1)$ time. Furthermore, a collection of such data structures for each $k' = 1, \ldots, k+1$ enables to find $\min\{\kappa(s,t), k+1\}$ in $O(\log k)$ time, using binary search. However, this data

[1] As in previous works, we ignore the unavoidable $O(\log n)$ factor invoked by storing the indexes of nodes, and assume that any basic arithmetic or comparison operation with indexes can be done in $O(1)$ time.

structure alone cannot answer cut queries for pairs that belong to the same part S_i of the partition. Using our data structure for k-S-connectivity from Theorem 2, and a new bound on the number of relevant cuts, we get the following, see also Table 1.

▶ **Theorem 3.** *There exists an $O(k^2n)$ space data structure that includes a list of $(2k+1)n$ cuts, that answers $\text{CON}_k(s,t)$ and $\text{PCUT}_k(s,t)$ queries in $O(\log k)$ time; a list of $k(k+2)n$ cuts and $O(k^3n)$ space allows to answer also $\text{CUT}_k(s,t)$ in $O(\log k)$ time. Furthermore, space $O(k^2n+n^2)$ allows to answer $\text{CON}_k(s,t)$ and $\text{PCUT}_k(s,t)$ in $O(1)$ time, and space $O(k^3n+n^2)$ allows to answer $\text{CON}_k(s,t)$ and $\text{CUT}_k(s,t)$ in $O(1)$ time.*

Theorems 1, 2, and 3 are proves in section 2, 3, and 4, respectively.

Table 1 Summary of the results in Theorem 3. Note that when k is bounded by a constant, the last row data structure has linear space and answers all queries in $O(1)$ time, which is optimal.

list size	space	$\text{PCON}_k(s,t)$	$\text{PCUT}_k(s,t)$	$\text{CON}_k(s,t)$	$\text{CUT}_k(s,t)$	reference
$n(n-1)/2$	$O(kn^2)$	$O(1)$	$O(1)$	$O(1)$	$O(1)$	folklore
-	$O(n^2)$	$O(1)$	-	$O(1)$	-	folklore
-	$O(kn)$	$O(1)$	-	-	-	[7]
-	$O(k^2n)$	$O(1)$	-	$O(\log k)$	-	[7]
$(2k+1)n$	$O(k^2n+n^2)$	$O(1)$	$O(1)$	$O(1)$	-	this paper
$k(k+2)n$	$O(k^3n+n^2)$	$O(1)$	$O(1)$	$O(1)$	$O(1)$	this paper
$(2k+1)n$	$O(k^2n)$	$O(\log k)$	$O(\log k)$	$O(\log k)$	-	this paper
$k(k+2)n$	$O(k^3n)$	$O(\log k)$	$O(\log k)$	$O(\log k)$	$O(\log k)$	this paper

In parallel to our work, Pettie, Saranurak, and Yin [17] gave a randomized $O(kn\log n)$ space data structure that answer $\text{CON}_k(s,t)$ queries in time $O(\log n)$. This data structure extends the data structure of Iszak and Nutov [8], that in turn is based on an idea of Chuzhoy and Khanna [2]. We briefly describe these results. Given a set $S \subseteq V$ of **terminals**, the edges and the nodes in $V \setminus S$ are called **elements**. The **element connectivity** between $s,t \in S$ is the maximum number of pairwise element disjoint st-paths. The Gomory-Hu tree extends to element connectivity (c.f. [19, 1]), and implies an $O(|S|)$ space data structure that answers element connectivity queries between terminals in $O(1)$ time. The data structure of [8] decomposes a node connectivity instance into $O(k^2\log n)$ element connectivity instances with $\Theta(n/k)$ terminals each; we will give a generalization of this decomposition in Section 5. For any element connectivity instance G_S with terminal set S and $s,t \in S$, $\kappa(s,t)$ is at most the element st-connectivity in G_S, and for at least one instance an equality holds (an instance with $s,t \in S$ and $Q \cap S = \emptyset$ for some minimum st-cut Q). So, to find $\kappa(s,t)$, one has to find the minimum element st-connectivity, among all instances in which s,t are terminals. There are $O(k^2\log n)$ element connectivity instances, with $\Theta(n/k)$ terminals each, hence the overall number of terminals is $O(k^2\log n)\cdot(n/k) = O(kn\log n)$. Iszak and Nutov [8] considered designing a labeling scheme[2], where efficient query time is not required, and used the connectivity classes data structure in each instance; this enables answering $\text{CON}_k(s,t)$ queries in $O(k\log n)$ time. Pettie, Saranurak, and Yin [17] used element connectivity Gomory-Hu trees instead, and also designed a novel data structure that given s,t finds an instance with the minimum element st-connectivity in $O(\log n)$ time. They also

[2] For additional work on labeling schemes for node-connectivity, see for example, [10, 11, 7].

showed that for large values of k, any data structure for answering node connectivity queries needs at least $\Omega(kn/\log n)$ space, matching up to an $O(\log n)$ factor the $O(kn)$ space of a sparse certificate graph [14].

Let us compare between the [17] $O(kn \log n)$ space data structure that answers $\textsc{con}_k(s,t)$ queries in $O(\log n)$ time, to our results. First thing to observe is that the [17] data structure can be augmented by a list of $O(kn \log n)$ cuts to support $\textsc{cut}_k(s,t)$ queries. Our cut list has size $O(k^2 n)$, which is better when $k < \log n$. Moreover, our query time is $O(\log k)$. For example for $k = \log n$ both data structures have space $O(n \log^2 n)$, but our query time is $O(\log \log n)$ while that of [17] is $O(\log n)$. Note that for $k \leq 4$ are known linear space data structures that can answer cut queries in $O(1)$ time; see [6] and [9] for the cases $k = 3$ and $k = 4$, respectively. Our work, that was done independently from [17] and uses totally different techniques, extend this to any constant k, bridging the gap between the result of [17] and the results known for $k \leq 4$.

2 *k*-connected graphs (Theorem 1)

In this section let $G = (V,E)$ be a k-connected graph. We first explain how to answer the queries for pairs s,t with $st \in E$. Let K denote the set of nodes of degree k in G. If $s \in K$ then the set of edges incident to s is a minimum st-cut for all t. There are $|K| \leq n$ such minimum cuts. This situation can be recognized in $O(1)$ time, hence we may omit such pairs from our analysis and assume that each of s,t has degree $\geq k+1$.

We say that $st \in E$ is a **critical edge** if $\kappa(s,t) = k$. Let F be the set of critical edges $st \in E$ such that $s,t \in V \setminus K$. Mader's Critical Cycle Theorem [12] states that any cycle of critical edges contains a node of degree k, hence F is a forest. Thus just specifying the edges in F and a list of $|F| \leq n - |K| - 1$ minimum st-cuts for every $st \in F$, gives an $O(kn)$ space data structure that answers the relevant queries in $O(1)$ time.

Henceforth assume that $s,t \in V \setminus K$ and $st \notin E$. We will show that then there exists a list of $n - |K|$ cuts, such that whenever $\kappa(s,t) = k$, there exists a minimum st-cut in the list. However, this is not enough to answer the relevant queries in $O(1)$ time, since we still need to choose the right minimum cut from the list.

For a node subset $A \subseteq V$ let ∂A denote the set of neighbors of A in G. Let $A^* = V \setminus (A \cup \partial A)$ denote the "node complement" of A. We say that A is: a **tight set** if $|\partial A| = k$ and $A^* \neq \emptyset$, an ***st*-set** if $s \in A$ and $t \in A^*$, and a **small set** if $|A| \leq \frac{n-k}{2}$. Note that A is tight if and only if ∂A is a minimum cut, and A is a union of some, but not all, connected components of $G \setminus \partial A$. The following statement is a folklore, c.f. [15, 16].

▶ **Lemma 4.** *Let A,B be tight sets. If the sets $A \cap B^*, B \cap A^*$ are both nonempty then they are both tight. If A,B are small and $A \cap B \neq \emptyset$ then $A \cap B$ is tight.*

Let $R = \{s \in V \setminus K : \text{there exist a small tight set containing } s\}$. For $s \in R$ let C_s be the (unique, by Lemma 4) inclusion minimal small tight set that contains s. Let $\mathcal{C} = \{C_s : s \in R\}$. The following lemma shows that the family $\{\partial C : C \in \mathcal{C}\}$ is a "short" list of $n - |K|$ minimum cuts, that for every $s,t \in V \setminus K$ with $st \notin E$ includes a minimum st-cut.

▶ **Lemma 5.** *Let $s,t \in V \setminus K$ with $st \notin E$. Then $\kappa(s,t) = k$ if an only if at least one of the following holds:* (i) *$s \in R$ and C_s is an st-set, or* (ii) *$t \in R$ and C_t is a ts-set. Consequently, $\kappa(s,t) = k$ if and only if the family $\{\partial C : C \in \mathcal{C}\}$ contains a minimum st-cut.*

Proof. If (i) holds then ∂C_s is a minimum st-cut, while if (ii) holds then ∂C_t is a minimum st-cut. Thus $\kappa(s,t) = k$ if (i) or (ii) holds.

Assume now that $\kappa(s,t)=k$ and we will show that (i) or (ii) holds. Let $Q \subset V \setminus \{s,t\}$ be a minimum st-cut. Then one component A of $G \setminus Q$ contains s and the other B contains t. Since $|A|+|B| \leq n-|Q|=n-k$, one of A,B, say A is small. Thus $s \in R$. Since $C_s \subseteq A$ and since $t \in A^*$, we have $t \in C_s^*$. Consequently, ∂C_s is a minimum st-cut, as required. ◀

Two sets A,B are **laminar** if they are disjoint or one of them contains the other; equivalently, A,B are not laminar if they intersect but none of them contains the other. A set family is laminar if its members are pairwise laminar. A laminar family $\mathcal{L}$ on V can be represented by a rooted tree $T=(\mathcal{L} \cup \{V\}, J)$ and a mapping $\psi : V \rightarrow \mathcal{L} \cup \{V\}$. Here B is a child of A in T if B is a maximal set in $\mathcal{L} \setminus \{A\}$ contained A, and for every $v \in V$, $\psi(v)$ is a minimal set in $\mathcal{L}$ that contains v.

▶ **Lemma 6.** *$\mathcal{C}$ can be partitioned into at most $2k+1$ laminar families.*

Proof. Consider two sets $A=C_a$ and $B=C_b$ that are not laminar. Then $a \notin A \cap B$ since otherwise by Lemma 4 $A \cap B$ is a (small) tight set that contains a, contradicting the minimality of $A=C_a$. By a similar argument, $b \notin A \cap B$. We also cannot have both $a \in A \cap B^*$ and $b \in B \cap A^*$, as then by Lemma 4 $A \cap B^*$ is a tight set that contains a, contradicting the minimality of $A=C_a$. Consequently, $a \in \partial B$ or $b \in \partial A$.

Construct an auxiliary directed graph H on node set R and edges set $\{ab : a \in \partial C_b\}$. The indegree of every node in H is at most k. This implies that every subgraph of the underlying graph of H has a node of degree $2k$. A graph is d-degenerate if every subgraph of it has a node of degree d. It is known that any d-degenerate graph can be colored with $d+1$ colors, in linear time, see [4, 13]. Hence H is $(2k+1)$-colorable. Consequently, we can compute in polynomial time a partition of R into at most $2k+1$ independent sets. For each independent set R_i, the family $\{C_s : s \in R_i\}$ is laminar. ◀

Our data structure for pairs $s,t \in V \setminus K$ with $st \notin E$ consists of:

- A family $\mathcal{T}$ of at most $2k+1$ trees, where each tree $T \in \mathcal{T}$ with a mapping $\psi_T : V \rightarrow V(T)$ represents one of the at most $2k+1$ laminar families of tight sets as in Lemma 6; the total number of edges in all trees in $\mathcal{T}$ is at most $n-|K|$.
- For each tree $T \in \mathcal{T}$, a linear space data structure that answers ancestor/descendant queries in $O(1)$ time. This can be done by assigning to each node of T the in-time and the out-time in a DFS search on T.
- A list $\{\partial C_s : s \in R\}$ of $|R|=n-|K|$ minimum cuts; this can be also encoded by an auxiliary directed graph $H=(V,F)$ with edge set $F=\{ts : t \in \partial C_s\}$. Using perfect hashing data structure we can check whether $ts \in F$ in $O(1)$ time.

For every $s \in S$ let T_s be the (unique) tree in $\mathcal{T}$ where C_s is represented. The next statement, that is a direct consequence of Lemma 5, specifies how we answer the queries.

▶ **Lemma 7.** *Let $s,t \in V \setminus K$ with $st \notin E$.*

(i) *If in T_s, $\psi_{T_s}(t)$ is not a descendant of $\psi_{T_s}(s)$ and $t \notin \partial C_s$, then ∂C_s is a minimum st-cut.*

(ii) *If in T_t, $\psi_{T_t}(s)$ is not a descendant of $\psi_{T_t}(t)$ and $s \notin \partial C_t$, then ∂C_t is a minimum st-cut.*

Furthermore, if none of (i),(ii) *holds then $\kappa(s,t) \geq k+1$.*

It is easy to see that with appropriate pointers, and using perfect hashing data structure to check adjacency in the auxiliary directed graph H, we get an $O(kn)$ space data structure that checks the two conditions in Lemma 7 in $O(1)$ time. If one of the conditions holds, the data structure return a pointer to one of ∂C_s or ∂C_t. Else, it reports that $\kappa(s,t) \geq k+1$.

This concludes the proof of Theorem 1.

3 Subset connectivity (Theorem 2)

In this section let $G = (V, E)$ be a k-S-connected graph with $|S| \geq 3k$. For the simpler case $S = V$ we related cuts to node subsets. Unfortunately, for the more general subset k-connectivity case, we need slightly more complex objects than sets, as follows.

▶ **Definition 8.** *An ordered pair $\mathbb{A} = (A, A^+)$ of subsets of a groundset V is called a* **biset** *if $A \subseteq A^+$; A is the* **inner set** *and A^+ is the* **outer set** *of $\mathbb{A}$, and $\partial\mathbb{A} = A^+ \setminus A$ is the* **boundary** *of $\mathbb{A}$. $A^* = V \setminus A^+$ is the* **co-set** *of $\mathbb{A}$ and $\mathbb{A}^* = (A^*, V \setminus A)$ is the* **co-biset** *of $\mathbb{A}$. We say that $\mathbb{A}$ is an* ***st*-biset** *if $s \in A$ and $t \in A^*$.*

In the case $S = V$, the relevant bisets were $(A, A^+) = (A, A \cup \partial A)$, where A was a tight set. Here we will say that $\mathbb{A}$ is a **tight biset** if $A \cap S \neq \emptyset$, $A^* \cap S \neq \emptyset$, and $|\partial\mathbb{A}| + |\delta(\mathbb{A})| = k$, where $\delta(\mathbb{A})$ is the set of edges in G that go from A to A^*. Note that $\mathbb{A}$ is tight if and only if $\partial\mathbb{A} \cup \delta(\mathbb{A})$ is a minimum st-cut for some $s \in A \cap S$ and $t \in A^* \cap S$. We will consider the family $\mathcal{F} = \{(A \cap S, A^+ \cap S) : \mathbb{A} \text{ is tight}\}$ obtained by projecting the tight bisets on S. Note that for $\mathbb{A} \in \mathcal{F}$ there might be many tight bisets in G whose projection on S is $\mathbb{A}$, and that there always exists at least one such biset.

▶ **Definition 9.** *The* **intersection** *and the* **union** *of two bisets $\mathbb{A}, \mathbb{B}$ are the bisets defined by $\mathbb{A} \cap \mathbb{B} = (A \cap B, A^+ \cap B^+)$ and $\mathbb{A} \cup \mathbb{B} = (A \cup B, A^+ \cup B^+)$. The biset $\mathbb{A} \setminus \mathbb{B}$ is defined by $\mathbb{A} \setminus \mathbb{B} = (A \setminus B^+, A^+ \setminus B)$. We say that $\mathbb{A}, \mathbb{B}$:* **intersect** *if $A \cap B \neq \emptyset$,* **cross** *if $A \cap B \neq \emptyset$ and $A^* \cap B^* \neq \emptyset$, and* **co-cross** *if $A \cap B^* \neq \emptyset$ and $B \cap A^* \neq \emptyset$.*

We say that $\mathbb{A} \in \mathcal{F}$ is a **small biset** if $|A| \leq \frac{|S|-k}{2}$, and $\mathbb{A}$ is a **large biset** otherwise. Clearly, $|\partial\mathbb{A}| \leq k$ for all $\mathbb{A} \in \mathcal{F}$. The family $\mathcal{F}$ has the following properties, c.f. [15, 16].

▶ **Lemma 10.** *The family $\mathcal{F} = \{(A \cap S, A^+ \cap S) : \mathbb{A} \text{ is tight}\}$ has the following properties:*
1. *$\mathcal{F}$ is* **symmetric***: $\mathbb{A}^* \in \mathcal{F}$ whenever $\mathbb{A} \in \mathcal{F}$.*
2. *$\mathcal{F}$ is* **crossing***: $\mathbb{A} \cap \mathbb{B}, \mathbb{A} \cup \mathbb{B} \in \mathcal{F}$ whenever $\mathbb{A}, \mathbb{B} \in \mathcal{F}$ cross.*
3. *$\mathcal{F}$ is* **co-crossing***: $\mathbb{A} \setminus \mathbb{B}, \mathbb{B} \setminus \mathbb{A} \in \mathcal{F}$ whenever $\mathbb{A}, \mathbb{B} \in \mathcal{F}$ co-cross.*
4. *If $\mathbb{A}, \mathbb{B} \in \mathcal{F}$ are small intersecting bisets then $\mathbb{A} \cap \mathbb{B}, \mathbb{A} \cup \mathbb{B} \in \mathcal{F}$.*

▶ **Definition 11.** *We say that a biset $\mathbb{B}$* **contains a biset** *$\mathbb{A}$ and write $\mathbb{A} \subseteq \mathbb{B}$ if $A \subseteq B$ and $A^+ \subseteq B^+$. $\mathbb{A}, \mathbb{B}$ are* **laminar** *if one of them contains the other or if $A \cap B = \emptyset$. A biset family is laminar if its members are pairwise laminar.*

For every $s \in S$ let $\mathcal{C}_s$ be the family of all inclusion minimal bisets $\mathbb{C} \in \mathcal{F}$ with $s \in C$ and $|C| \leq |C^*|$. Let $\mathcal{C} = \cup_{s \in S}\mathcal{C}_s$. Note that $|\mathcal{C}_s| \leq |S| - 1$ and that $\mathcal{C}_s$ can be computed using $|S| - 1$ min-cut computations. The following observation follows from the symmetry of $\mathcal{F}$.

▶ **Lemma 12.** *Let $\mathbb{A} \in \mathcal{F}$ be an st-biset. If $|A| \leq |A^*|$ then there is an st-biset in $\mathcal{C}_s$, and if $|A^*| \leq |A|$ then there is a ts-biset in $\mathcal{C}_t$. Consequently, for any $s, t \in S$, $\mathcal{C}$ contains an st-biset or a ts-biset.*

The next lemma shows that $|\mathcal{C}_s| \leq 3$ if $|S| \geq 3k$.

▶ **Lemma 13.** *For any $s \in S$, $\mathcal{C}_s$ contains at most one small biset and at most $\frac{2(|S|-1)}{|S|-k}$ large bisets. In particular, if $|S| \geq 3k$ then in $\mathcal{C}_s$ there are at most two large bisets and $|\mathcal{C}| \leq 3|S|$.*

Proof. In $\mathcal{C}_s$ there is at most one small biset, by property 4 in Lemma 10. No two large bisets $\mathbb{A}, \mathbb{B} \in \mathcal{C}_s$ cross, as otherwise $s \in \mathbb{A} \cap \mathbb{B}$ and (by property 2 in Lemma 10) $\mathbb{A} \cap \mathbb{B} \in \mathcal{F}$, contradicting the minimality of $\mathbb{A}, \mathbb{B}$. Thus the sets in $\{C^* : \mathbb{C} \in \mathcal{C}_s\}$ are pairwise disjoint. Furthermore, $|C^*| \geq |C| \geq \frac{|S|-k}{2}$ for every large biset $\mathbb{C} \in \mathcal{C}_s$. This implies that the number of large bisets in $\mathcal{C}_s$ is at most $\frac{|S|-1}{(|S|-k)/2} \leq \frac{2(3k-1)}{2k} < 3$ if $|S| \geq 3k$. ◀

By a proof identical to that of Lemma 6 we have the following.

▶ **Lemma 14.** *The family of small bisets in $\mathcal{C}$ can be partitioned in polynomial time into at most $2k+1$ laminar families.*

Proof. Consider two small bisets $\mathbb{A} \in \mathcal{C}_a$ and $\mathbb{B} \in \mathcal{C}_b$ that are not laminar. Then $a \notin A \cap B$ since otherwise $\mathbb{A} \cap \mathbb{B} \in \mathcal{F}$, contradicting the minimality of $\mathbb{A}$. Similarly, $b \notin A \cap B$. We also cannot have both $a \in A \cap B^*$ and $b \in B \cap A^*$, as then $\mathbb{A}, \mathbb{B}$ co-cross and thus $\mathbb{A} \setminus \mathbb{B}, \mathbb{B} \setminus \mathbb{A} \in \mathcal{F}$, contradicting the minimality of $\mathbb{A}, \mathbb{B}$. Consequently, $a \in \partial B$ or $b \in \partial A$. The rest of the proof coincides with that of Lemma 6. ◀

Later, we will prove the following.

▶ **Lemma 15.** *If $|S| \geq 3k$ then the family of large bisets in $\mathcal{C}$ can be partitioned in polynomial time into at most $3(2k+1)$ laminar families.*

Lemmas 14 and 15 imply that $\mathcal{C}$ can be partitioned into at most $4(2k+1)$ laminar families. For our purposes, we just need the family of the inner sets of each family to be laminar. Together with Lemma 13 this implies Theorem 2 in the same way as Lemma 6 implies Theorem 1, except the following minor differences.

- Here we have $4(2k+1)$ laminar families instead of $2k+1$ laminar families, and for each $s \in S$ we have by Lemma 13 $|\mathcal{C}_s| \leq 3$ instead of $|\mathcal{C}_s| = 1$.
- For each biset $\mathbb{C} \in \mathcal{C}$, our list of cuts will include a mixed cut $\partial\mathbb{A} \cup \delta(\mathbb{A})$ of some tight biset of G whose projection on S is $\mathbb{C}$.

These differences affect space and query time only by a small constant factor. Thus all we need is to prove Lemma 15, which we will do in the rest of this section.

▶ **Lemma 16.** *Let $\mathbb{A} \in \mathcal{C}_a$ and $\mathbb{B} \in \mathcal{C}_b$ be two non-laminar large bisets in $\mathcal{C}$. If $|S| \geq 3k$ then $a \in \partial\mathbb{B}$ or $b \in \partial\mathbb{A}$, or (see Fig. 1(a)): $a, b \in A \cap B$, $A^* \cap B^* = \emptyset$, and both $A \cap B^*, B \cap A^*$ are non-empty.*

Proof. Assume that $a \notin \partial\mathbb{B}$ and $b \notin \partial\mathbb{A}$. We will show that then the case in Fig. 1(a) holds.

Suppose that $a \in A \cap B$; the analysis of the case $b \in A \cap B$ is similar. Then $A^* \cap B^* = \emptyset$; otherwise, $\mathbb{A}, \mathbb{B}$ cross and (by property 2 in Lemma 10) we get $\mathbb{A} \cap \mathbb{B} \in \mathcal{F}$, contradicting the minimality of $\mathbb{A}$. Furthermore, if $B \cap A^* = \emptyset$ we get $\frac{|S|-k}{2} < |A| \leq |A^*| = |\partial\mathbb{B} \cap A^*| \leq k$, contradicting that $|S| \geq 3k$. By a similar argument, $A \cap B^* \neq \emptyset$. If $a \in A \cap B$ and $b \in B \cap A^*$ (Fig. 1(b)), then $\mathbb{A}, \mathbb{B}$ co-cross (by property 3 in Lemma 10) and thus $\mathbb{B} \setminus \mathbb{A} \in \mathcal{F}$; this contradicts the minimality of $\mathbb{B}$.

If none of a, b is in $A \cap B$, then (see Fig. 1(c)) $a \in A \cap B^*$ and $b \in B \cap A^*$. Consequently, $\mathbb{A}, \mathbb{B}$-co-cross, and thus (by property 3 in Lemma 10) $\mathbb{A} \setminus \mathbb{B}, \mathbb{B} \setminus \mathbb{A} \in \mathcal{F}$, contradicting the minimality of $\mathbb{A}, \mathbb{B}$.

Thus the only possible case is the one in Fig. 1(a), completing the proof of the lemma. ◀

From Lemma 16, by a proof identical to that of Lemma 6 we get the following.

▶ **Corollary 17.** *The family of large bisets in $\mathcal{C}$ can be partitioned in polynomial time into at most $2k+1$ parts such that any two bisets $\mathbb{A} \in \mathcal{C}_a$ and $\mathbb{B} \in \mathcal{C}_b$ that belong to the same part $\mathcal{P}$ are either laminar, or have the following property: $a, b \in A \cap B$ and $A^* \cap B^* = \emptyset$.*

Thus the following lemma finishes the proof of Lemma 15, and also of Theorem 2.

▶ **Lemma 18.** *Let $\mathcal{P}$ be one of the $2k+1$ parts as in Corollary 17; in particular, if $\mathbb{A}, \mathbb{B} \in \mathcal{P}$ are not laminar then $A \cap B \neq \emptyset$ and $A^* \cap B^* = \emptyset$. Then $\mathcal{P}$ can be partitioned in polynomial time into at most 3 laminar families.*

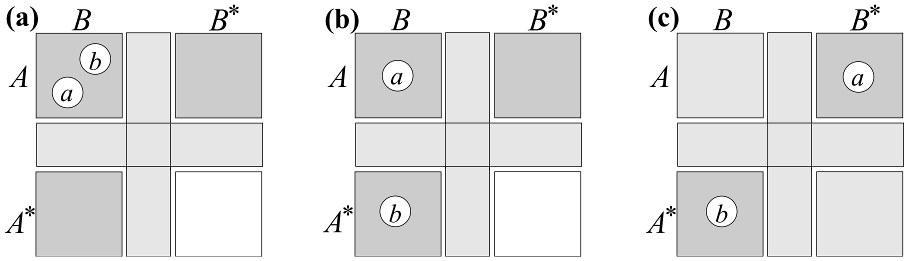

Figure 1 Illustration to the proof of Lemma 16; dark gray sets are non-empty.

Proof. Let $\mathcal{M}$ be the family of maximal members in $\mathcal{P}$. We will show later that the set family $\mathcal{H} = \{A^* : \mathbb{A} \in \mathcal{M}\}$ has a hitting set U of size $|U| \leq 3$. Now note that:

- For every $v \in S$ the family $\mathcal{P}_v = \{\mathbb{A} \in \mathcal{P} : v \in A^*\}$ is laminar, since $v \in A^* \cap B^*$ for any $\mathbb{A}, \mathbb{B} \in \mathcal{P}_v$, while $A^* \cap B^* = \emptyset$ for any non-laminar $\mathbb{A}, \mathbb{B} \in \mathcal{P}$.
- Since U is a hitting set of $\mathcal{H}$, for any $\mathbb{A} \in \mathcal{P}$ there is $v \in U \cap A^*$, and then $\mathbb{A} \in \mathcal{P}_v$.

Summarizing, each one of the families $\mathcal{P}_v$ is laminar and $\cup_{v \in U} \mathcal{P}_v = \mathcal{P}$. By removing bisets that appear more than once, we get a partition of $\mathcal{P}$ into $|U| \leq 3$ laminar families.

It remains to show that $\mathcal{H}$ has a hitting set of size ≤ 3. A **fractional hitting set** of $\mathcal{H}$ is a function $h : S \to [0,1]$ such that $h(A) = \sum_{v \in A} h(v) \geq 1$ for all $A \in \mathcal{H}$. For $v \in S$ let $\mathcal{H}_v$ be the family of sets in $\mathcal{H}$ that contain v, and let $\mathcal{M}_v = \{\mathbb{A} \in \mathcal{M} : A^* \in \mathcal{H}_v\}$. Note that:

- $h(v) = \frac{2}{|S|-k+1}$ for all $v \in S$ is a fractional hitting set of $\mathcal{H}$ and $h(S) = \frac{2|S|}{|S|-k+1} < 3$.
- No two bisets in $\mathcal{M}_v$ intersect and $|\mathcal{H}_v| \leq |\mathcal{M}_v|$. This implies $|\mathcal{H}_v| \leq |\mathcal{M}_v| \leq \frac{2|S|}{|S|-k+1} < 3$, so $|\mathcal{H}_v| \leq 2$ for all $v \in S$.

Since $|\mathcal{H}_v| \leq 2$ for all $v \in S$, computing a minimum hitting set of $\mathcal{H}$ reduces to the minimum Edge-Cover problem, and $\mathcal{H}$ has a hitting set U of size $|U| \leq \frac{4}{3} \cdot h(S)$ ($\frac{4}{3}$ is the integrality gap of the Edge-Cover problem). Since $\frac{4}{3} \cdot h(S) < \frac{4}{3} \cdot 3$, $\mathcal{H}$ has a hitting set U of size $|U| \leq 3$. ◀

4 Arbitrary graphs (Theorem 3)

For the proof of Theorem 3, we will later prove the following.

▶ **Theorem 19.** *There exists a list of at most $(2k+1)n$ cuts that contains an st-cut of size $\leq k$ for any $s, t \in V$ with $\kappa(s,t) \leq k$.*

A union of list as in Theorem 19 for every $k' = 1, \ldots k$ is a list that includes a minimum st-cut for any $s, t \in V$ with $\kappa(s,t) \leq k$. The size of this list is $\sum_{k'=1}^{k}(2k'+1)n = k(k+2)n$. Thus we have the following.

▶ **Corollary 20.** *There exists a list of at most $k(k+2)n$ cuts that contains a minimum st-cut for any $s, t \in V$ with $\kappa(s,t) \leq k$.*

We can combine these bounds with the trivial data structure. The combined data structure will include a a list of cuts, a $V \times V$ matrix, and for each matrix entry (s,t) the number $\min\{\kappa(s,t), k+1\}$ and a pointer to an st-cut in the list. We can answer in $O(1)$ time $\textsc{pcut}_k(s,t)$ queries using list of size $O(kn)$ and $\textsc{cut}_k(s,t)$ queries using list of size $O(k^2 n)$. This proves the second part of Theorem 3.

For the first part of Theorem 3 we combine our bounds with the Hsu-Lu [7] data structure. Recall that this data structure has two ingredients:

- An ordered partition $\mathcal{P} = (S_1, S_2, \ldots)$ of V.
- A directed graph $H = (V, F)$ such that every S_i has at most $2k-1$ neighbors in H, and all of them are in $S_{i+1} \cup S_{i+2} \cup \cdots$; this implies that $|F| \leq (2k-1)n$.

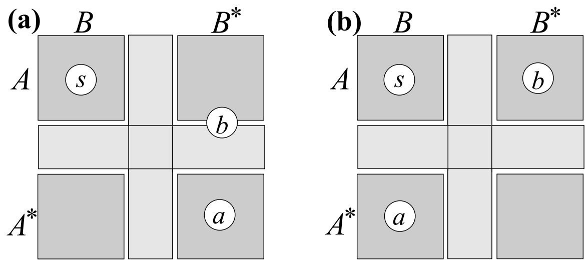

Figure 2 Illustration to the proof of Lemma 21.

Then $\kappa(s,t) \geq k+1$ iff s,t belong to the same part, or $st \in F$, or $ts \in F$. Overall, this data structure can be implemented using $O(kn)$ space and answers $\textsc{pcon}_k(s,t)$ queries in $O(1)$ time.

We augment this data structure by adding to each part $S \in \mathcal{P}$ a data structure for subset k-S-connectivity, with a unified cut list as in Theorem 19; we add Theorem 2 data structure if $|S| \geq 3k$, and the trivial data structure (an $S \times S$ matrix) if $|S| < 3k$.

The dominating space of the combined data structure is $O(k^2 n)$, due to storing the Theorem 19 cut list; other parts need substantially lower space. Ignoring the space of the cut list, subset k-connectivity data structures for $S \in \mathcal{P}$ with $|S| \geq 3k$ need total space $\sum_{S \in \mathcal{P}} O(|S|) = O(n)$. For $|S| < 3k$ the trivial data structure has space $O(|S|^2) = O(k|S|)$. It is not hard to see that the worse case is when there are $\Theta(n/k)$ parts in $\mathcal{P}$ of size $\Theta(k)$ each. Thus the total space invoked by parts with $|S| < 3k$ is also $O(n)$. Finally, the space required to store F is $O(|F|) = O(kn)$.

Now we observe that a collection of such data structures for each $k' = 1, \ldots, k+1$ enables to find $\min\{\kappa(s,t), k+1\}$ in $O(\log k)$ time, using binary search. Once $k' = \kappa(s,t)$ is determined, we can also answer any $\textsc{cut}_k(s,t)$ query in $O(1)$ time. The dominating space of this data structure is $O(k^3 n)$, due to storing the Corollary 20 cut list.

This concludes the proof of Theorem 2, except that we need to prove Theorem 19, which we will do in the rest of this section.

For a biset $\mathbb{A}$ let $\psi(\mathbb{A}) = |\partial\mathbb{A}| + |\delta(\mathbb{A})|$. Here we will say that $\mathbb{A}$ is an ***st*-tight biset** if $\mathbb{A}$ is an st-biset and $\psi(\mathbb{A}) = \kappa(s,t)$. Note that then $\partial\mathbb{A} \cup \delta(\mathbb{A})$ is a minimum st-cut, and that for any minimum st-cut Q there exists an st-tight biset $\mathbb{A}$ with $\partial\mathbb{A} \cup \delta(\mathbb{A}) = Q$. It is known that the function ψ satisfies the submodular inequality $\psi(\mathbb{A}) + \psi(\mathbb{B}) \geq \psi(\mathbb{A} \cap \mathbb{B}) + \psi(\mathbb{A} \cup \mathbb{B})$, and (by symmetry) also the co-submodular inequality $\psi(\mathbb{A}) + \psi(\mathbb{B}) \geq \psi(\mathbb{A} \setminus \mathbb{B}) + \psi(\mathbb{B} \setminus \mathbb{A})$.

It is known that if $\mathbb{A}, \mathbb{B}$ are both st-tight then so are $\mathbb{A} \cap \mathbb{B}, \mathbb{A} \cup \mathbb{B}$. Let $\mathbb{C}_{st}$ denote the (unique) inclusion minimal st-tight biset. For $s \in S$ let $T_s = \{t \in V : \kappa(s,t) \leq k, |C_{st}| \leq |C_{ts}|\}$. Let $\mathcal{C}_s$ be the family of all inclusion minimal bisets in the family $\{\mathbb{C}_{st} : t \in T_s\}$. Let $\mathcal{C} = \cup_{s \in S} \mathcal{C}_s$. One can verify that for any $s,t \in V$ with $\kappa_(s,t) \leq k$, $\mathcal{C}$ contains and st-biset or a ts-biset $\mathbb{C}$ with $\psi(\mathbb{C}) \leq k$. We will show that $|\mathcal{C}_s| \leq 2k+1$ for all $s \in V$. For that, we need the following lemma.

▶ **Lemma 21.** *Let $\mathbb{A} = \mathbb{C}_{sa}$ and $\mathbb{B} = \mathbb{C}_{sb}$ be distinct bisets in $\mathcal{C}_s$. Then $a \in \partial B$ or $b \in \partial A$.*

Proof. Suppose to the contrary that $a \notin \partial B$ and $b \notin \partial A$. If one of a, b is in $A^* \cap B^*$, say $a \in A^* \cap B^*$ (see Fig. 2(a)), then $\mathbb{A} \cup \mathbb{B}$ is an sa-biset and $\mathbb{A} \cap \mathbb{B}$ is an sb-biset.Thus

$$\kappa(s,a) + \kappa(s,b) = \psi(\mathbb{A}) + \psi(\mathbb{B}) \geq \psi(\mathbb{A} \cap \mathbb{B}) + \psi(\mathbb{A} \cup \mathbb{B}) \geq \kappa(s,b) + \kappa(s,a) \ .$$

Hence equality holds everywhere, so $\mathbb{A} \cap \mathbb{B}$ is sb-tight. This contradicts the minimality of $\mathbb{B}$.

Else, $a \in A^* \cap B$ and $b \in B^* \cap A$ (see Fig. 2(b)). Then $\mathbb{A} \setminus \mathbb{B}$ is a bs-biset and $\mathbb{B} \setminus \mathbb{A}$ is an as-biset.Thus

$$\kappa(s,a) + \kappa(s,b) = \psi(\mathbb{A}) + \psi(\mathbb{B}) \geq \psi(\mathbb{A} \setminus \mathbb{B}) + \psi(\mathbb{B} \setminus \mathbb{A}) \geq \kappa(b,s) + \kappa(a,s) \ .$$

Hence equality holds everywhere, so $\mathbb{A} \setminus \mathbb{B}$ is as-tight and $\mathbb{B} \setminus \mathbb{A}$ is bs-tight. This implies $|C_{sa}| > |C_{bs}|$ and $|C_{sb}| > |C_{as}|$, and we get the contradiction $|C_{sa}| + |C_{sb}| > |C_{bs}| + |C_{as}|$. ◀

▶ **Lemma 22.** *$|\mathcal{C}_s| \leq 2k+1$ for all $s \in V$.*

Proof. The proof is similar to the one of Lemma 6. Construct an auxiliary directed graph H on node set T_s and edge set $\{ab : a \in \partial \mathbb{C}_{sb}\}$. Note that if H has no edge between a and b then $\mathbb{C}_{sa} = \mathbb{C}_{sb}$. The indegree of every node in H is at most k. Thus by the same argument as in Lemma 6 we get that the underlying graph of H is $(2k+1)$-colorable, and thus T_s can be partitioned into at most $2k+1$ independent sets. For each independent set T', the family $\{\mathbb{C}_{st} : t \in T'\}$ consists of a single biset. ◀

This concludes the proof of Theorem 19, and thus also of Theorem 3.

5 Decomposition of node connectivity into element connectivity

Recall that given a set $S \subseteq V$ of terminals, The element connectivity between $s,t \in S$ is the maximum number of pairwise element disjoint st-paths, where elements are the edges and the nodes in $V \setminus S$. Let $\kappa^S_G(s,t)$ denote the st-element-connectivity in G. By Menger's Theorem, $\kappa^S_G(s,t)$ equals the minimum size $|C|$ of a set C of elements with $C \cap S = \emptyset$ such $G \setminus C$ has not st-path. It is easy to see that $\kappa^S_G(s,t) \geq \kappa_G(s,t)$, and that an equality holds iff there exists a minimum st-cut C with $C \cap S = \emptyset$. We thus will consider the following problem: given a family $\mathcal{C}$ of subsets of V, find a "small" family $\mathcal{S}$ of subsets of V such that for every $s,t \in V$ and $C \in \mathcal{C}$ with $s,t \notin C$, there is $S \in \mathcal{S}$ with $s,t \in S$ and $C \cap S = \emptyset$; following [2], we will call such a family **$\mathcal{C}$-resilient**. The objective can be also to minimize $\sum_{S \in \mathcal{S}} |S|$. Chuzhoy and Khanna [2] showed that if $\{C \subset V : |C| \leq k\}$ is the family of all subsets of size $\leq k$, then there exists a $\mathcal{C}$-resilient family $\mathcal{S}$ of size $|\mathcal{S}| = O(k^3 \ln n)$. They also gave a randomized polynomial time algorithm for finding such $\mathcal{S}$. The number of subsets of size $\leq k$ is $\binom{n}{k} \approx n^k$, while the relevant family in our case – as in Corollary 20, has a much smaller size $|\mathcal{C}| \leq k(k+2)n$. We will consider the case of an arbitrary family $\mathcal{C} \subseteq \{C \subseteq V : |C| \leq k\}$, and prove the following.

▶ **Lemma 23.** *Let $\mathcal{C}$ be a family of sets of size at most k each on a groundset V of size n. Then there exists a $\mathcal{C}$-resilient family $\mathcal{S}$ of $O(k^2 \ln(n|\mathcal{C}|))$ subsets of V of size $r = \frac{n-k}{k+1}$ each. Furthermore, assigning to each set in $\mathcal{A} = \{S \subseteq V : |S| = r\}$ probability $\Delta = 1/\binom{n-k-2}{r-2}$ and applying randomized rounding $4\ln(n|\mathcal{C}|)$ times gives such $\mathcal{S}$ w.h.p.*

Proof. If $n \leq 3k+1$, then the subsets of V of size 2 is a family as required of size $\frac{3k(3k+1)}{2}$, so assume that $n \geq 3k+2$.

Let $\mathcal{A} = \{S \subseteq V : |S| = r\}$ and $\mathcal{B} = \{(\{s,t\},C) : s,t \in V, C \in \mathcal{C}\}$. Define a bipartite graph with sides $\mathcal{A}, \mathcal{B}$ by connecting $S \in \mathcal{A}$ to $(\{s,t\},C) \in \mathcal{B}$ if $s,t \in S$ and $C \cap S = \emptyset$; in this case we will say that S **covers** $(\{s,t\},C)$. This defines an instance of the Set Cover problem, where $\mathcal{A}$ are the sets and $\mathcal{B}$ are the elements. The lemma says that there exists a cover $\mathcal{S} \subseteq \mathcal{A}$ of $\mathcal{B}$ that has size $|\mathcal{S}| = O(k^2 \log(n|\mathcal{C}|))$.

A **fractional cover** of $\mathcal{B}$ is a function $h : \mathcal{A} \longrightarrow [0,1]$ such that

$$\sum \{h(S) : S \in \mathcal{A} \text{ covers } (\{s,t\},C)\} \geq 1 \qquad \forall (\{s,t\},C) \in \mathcal{B} \ .$$

The **value** of a fractional cover h is $\sum_{S\in\mathcal{A}} h(S)$. It is known that if there is a fractional cover of value τ, then there is a cover of size $\tau(1+\ln|\mathcal{B}|)$. We have $|\mathcal{B}| = \frac{n(n-1)}{2}|\mathcal{C}|$, hence $\ln|\mathcal{B}| \leq 2\ln(n|\mathcal{C}|) - \ln 2$ and $\lceil 2\ln|\mathcal{B}|\rceil \leq 4\ln(n|\mathcal{C}|)$.

Our next goal is to show that there is a fractional cover of value $O(k^2)$. We have $|\mathcal{A}| = \binom{n}{r}$. The number of sets in $\mathcal{A}$ that cover a given member $(\{s,t\},C) \in \mathcal{B}$ is $\Delta = \binom{n-k-2}{r-2}$, which is the number of choices of a set $S \setminus \{s,t\}$ of size $r-2$ from the set $V \setminus (C \cup \{s,t\})$ of size $n-k-2$. Defining $h(S) = 1/\Delta$ for all $S \in \mathcal{A}$ gives a fractional cover of value $|\mathcal{A}|/\Delta$. Denote $m = n-k$. Then:

$$\frac{|\mathcal{A}|}{\Delta} = \frac{\binom{n}{r}}{\binom{m-2}{r-2}} = \frac{m(m-1)}{r(r-1)} \cdot \frac{n!}{(n-r)!} \cdot \frac{(m-r)!}{m!} \leq \frac{m^2}{(r-1)^2} \prod_{i=1}^{r} \frac{n-i+1}{m-i+1} \; .$$

Note that for $1 \leq i \leq r$ we have $\frac{n-i+1}{m-i+1} = 1 + \frac{n-m}{m-i+1} \leq 1 + \frac{k}{n-k-r}$. Let us choose r such that $\frac{k}{n-k-r} = \frac{1}{r}$, so $r = \frac{n-k}{k+1}$; assume that r is an integer, as adjustment to floors and ceilings only affects by a small amount the constant hidden in the $O(\cdot)$ term. Since $(1+1/r)^r \leq e$ we obtain

$$\prod_{i=1}^{r} \frac{n-i+1}{m-i+1} \leq \left(1+\frac{1}{r}\right)^r \leq e \; .$$

Since we assume that $n \geq 3k+2$, we have $\frac{n-k}{k+1} \geq 2$ and thus $\frac{m}{r-1} \leq 2(k+1)$. Consequently, we get that $\frac{|\mathcal{A}|}{\Delta} \cdot (1+\ln|\mathcal{B}|) = O(k^2 \ln(n|\mathcal{C}|))$. This implies that a standard greedy algorithm for Set Cover, produces the required family of size $O\left(k^2 \ln n|\mathcal{C}|\right)$. There is a difficulty to implement this algorithm in time polynomial in n (unless $r = \frac{n-k}{k+1}$ is a constant), since $|\mathcal{A}| = \binom{n}{r}$ may not be polynomial in n. Thus we use a randomized algorithm for Set Cover, by rounding each entry to 1 with probability determined by our fractional cover. It is known that repeating this rounding $2\lceil\ln|\mathcal{B}|\rceil \leq 4\ln(n|\mathcal{C}|)$ times gives a cover w.h.p., and clearly its expected size is $2\lceil\ln|\mathcal{B}|\rceil$ times the value of the fractional hitting set. In our case, $h(S) = 1/\Delta = 1/\binom{n-k-2}{r-2}$ for all $S \in \mathcal{A}$. Thus we just need to assign to each set in $\mathcal{A}$ probability $1/\Delta$, and apply randomized rounding $4\ln(n|\mathcal{C}|)$ times. ◀

Applying Lemma 23 on the family $\mathcal{C}$ as in as in Corollary 20, that has size $|\mathcal{C}| \leq k(k+2)n$, we get that that there exists a $\mathcal{C}$-resilient family $\mathcal{S}$ of $O(k^2\ln(n|\mathcal{C}|)) = O(k^2 \ln n)$ subsets of V of size $r = \frac{n-k}{k+1}$ each. On the other hand, if $|\mathcal{C}|$ is the family of all subsets of V of size k, then $|\mathcal{C}| = \binom{n}{k} < (ne/k)^k$ and we get the bound $O(k^2\ln(n|\mathcal{C}|)) = O(k^3 \ln n)$ of Chuzhoy and Khanna [2].

References

1 C. Chekuri, T. Rukkanchanunt, and C. Xu. On element-connectivity preserving graph simplification. In *23rd European Symposium on Algorithms (ESA)*, pages 313–324, 2015.

2 J. Chuzhoy and S. Khanna. An $O(k^3 \log n)$-approximation algorithm for vertex-connectivity survivable network design. *Theory of Computing*, 8(1):401–413, 2012.

3 R. F. Cohen, G. Di Battista, A. Kanevsky, and R. Tamassia. Reinventing the wheel: an optimal data structure for connectivity queries. In *25th Symposium on Theory of Computing (STOC)*, pages 194–200, 1993.

4 P. Erdös and A. Hajnal. On chromatic number of graphs and set-systems. *Acta Mathematica Hungarica*, 17(1–2):61–99, 1966.

5 R. E. Gomory and T. C. Hu. Multi-terminal network flows. *Journal of the Society for Industrial and Applied Mathematics*, 9, 1961.

6 J. E. Hopcroft and R. E. Tarjan. Dividing a graph into triconnected components. *SIAM J. Computing*, 2(3):135–158, 1973.

7 T-H. Hsu and H-I. Lu. An optimal labeling for node connectivity. In *20th International Symposium Algorithms and Computation (ISAAC)*, pages 303–310, 2009.

8 R. Izsak and Z. Nutov. A note on labeling schemes for graph connectivity. *Information Processing Letters*, 112(1-2):39–43, 2012.

9 A. Kanevsky, R. Tamassia, G. Di Battista, and J. Chen. On-line maintenance of the four-connected components of a graph. In *32nd Annual Symposium on Foundations of Computer Science (FOCS)*, pages 793–801, 1991.

10 M. Katz, N. A. Katz, A. Korman, and D. Peleg. Labeling schemes for flow and connectivity. *SIAM J. Comput.*, 34(1):23–40, 2004.

11 A. Korman. Labeling schemes for vertex connectivity. *ACM Transactions on Algorithms*, 6(2), 2010.

12 W. Mader. Ecken vom grad n in minimalen n-fach zusammenhängenden graphen. *Archive der Mathematik*, 23:219–224, 1972.

13 D. W. Matula and L. L. Beck. Smallest-last ordering and clustering and graph coloring algorithms. *Journal of the ACM*, 30(3):417–427, 1983.

14 H. Nagamochi and T. Ibaraki. A linear-time algorithm for finding a sparse k-connected spanning subgraph of a k-connected graph. *Algorithmica*, 7(5&6):583–596, 1992.

15 Z. Nutov. Approximating subset k-connectivity problems. *J. Discrete Algorithms*, 17:51–59, 2012.

16 Z. Nutov. Improved approximation algorithms for minimum cost node-connectivity augmentation problems. *Theory of Computing Systems*, 62(3):510–532, 2018.

17 S. Pettie, T. Saranurak, and L. Yin. Optimal vertex connectivity oracles. In *54th Symposium on Theory of Computing (STOC)*, pages 151–161, 2022.

18 S. Pettie and L. Yin. The structure of minimum vertex cuts. In *48th International Colloquium on Automata, Languages, and Programming (ICALP)*, pages 105:1–105:20, 2021.

19 A. Schrijver. *Combinatorial Optimization: Polyhedra and Efficiency*. Springer Verlag, Berlin Heidelberg, 2003.