

# Near-Optimal Vertex Fault-Tolerant Labels for Steiner Connectivity

Koustav Bhanja 

Weizmann Institute of Science, Rehovot, Israel

Asaf Petruschka 

Weizmann Institute of Science, Rehovot, Israel

---

## Abstract

We present a compact labeling scheme for determining whether a designated set of terminals in a graph remains connected after any  $f$  (or less) vertex failures occur. An  $f$ -FT Steiner connectivity labeling scheme for an  $n$ -vertex graph  $G = (V, E)$  with terminal set  $U \subseteq V$  provides labels to the vertices of  $G$ , such that given only the labels of any subset  $F \subseteq V$  with  $|F| \leq f$ , one can determine if  $U$  remains connected in  $G - F$ . The main complexity measure is the maximum label length.

The special case  $U = V$  of global connectivity has been recently studied by Jiang, Parter, and Petruschka [33], who provided labels of  $n^{1-1/f} \cdot \text{poly}(f, \log n)$  bits. This is near-optimal (up to  $\text{poly}(f, \log n)$  factors) by a lower bound of Long, Pettie and Saranurak [37]. Our scheme achieves labels of  $|U|^{1-1/f} \cdot \text{poly}(f, \log n)$  for general  $U \subseteq V$ , which is near-optimal for any given size  $|U|$  of the terminal set. To handle terminal sets, our approach differs from [33]. We use a well-structured Steiner tree for  $U$  produced by a decomposition theorem of Duan and Pettie [24], and bypass the need for Nagamochi-Ibaraki sparsification [40].

**2012 ACM Subject Classification** Theory of computation  $\rightarrow$  Graph algorithms analysis; Theory of computation  $\rightarrow$  Data structures design and analysis; Theory of computation  $\rightarrow$  Distributed algorithms

**Keywords and phrases** Fault Tolerance, Labeling Schemes, Steiner Connectivity

**Digital Object Identifier** 10.4230/LIPIcs.ESA.2025.44

**Related Version** Full Version: <https://arxiv.org/abs/2506.23215>

**Funding** Koustav Bhanja: Supported by Merav Parter’s European Research Council (ERC) grant under the European Union’s Horizon 2020 research and innovation programme, grant agreement No. 949083.

Asaf Petruschka: Supported by an Azrieli Foundation fellowship, and by Merav Parter’s European Research Council (ERC) grant under the European Union’s Horizon 2020 research and innovation programme, grant agreement No. 949083.

**Acknowledgements** We are thankful to Merav Parter for encouraging this collaboration and for valuable discussions.

## 1 Introduction

Labeling schemes are distributed graph data structures with many applications in graph algorithms and distributed computing. The goal is to assign the vertices (and/or edges) of the graph with succinct yet informative labels, so that queries of interest can be answered from only the labels of the query arguments, without any access to the graph or to any other centralized storage. The main complexity measure is the *label length*, which is the maximum number of bits stored in a single label; construction and query time are secondary measures. Diverse types of labeling have been studied, with notable examples being adjacency [14, 15, 34, 4, 7], ancestry and least common ancestors in trees [1, 6], exact or approximate distances [46, 26, 5, 12, 27], and pairwise edge- or vertex-connectivity [35, 30, 31, 44].



© Koustav Bhanja and Asaf Petruschka;  
licensed under Creative Commons License CC-BY 4.0

33rd Annual European Symposium on Algorithms (ESA 2025).

Editors: Anne Benoit, Haim Kaplan, Sebastian Wild, and Grzegorz Herman; Article No. 44; pp. 44:1–44:13

Leibniz International Proceedings in Informatics



LIPICs Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

The error-prone nature of many real-life networks has further motivated researchers to consider labeling schemes designed to support events of failures in the graph. Courcelle and Twig [19] were the first to explicitly define and study *fault-tolerant (FT)* (a.k.a *forbidden set*) labeling schemes, where the goal is to answer queries subject to the failure (i.e., deletion) of a set  $F$  of vertices or edges in the graph. Here, the set  $F$  is part of the query arguments, so the labels assigned to  $F$  are available to the query algorithm. Usually,  $|F|$  is assumed to be bounded by some integer parameter  $f \geq 1$  (this is abbreviated to the  $f$ -FT model). A lot of the works on FT labeling studied special graph families, such as planar or bounded doubling-dimension graphs [19, 18, 2, 3, 8, 13]. In this work, we consider a general  $n$ -vertex (undirected) graph  $G = (V, E)$  and focus on connectivity-related  $f$ -FT labeling schemes.

**$s$ - $t$  connectivity.** The “ $s$ - $t$  variant” of this problem has been extensively studied in recent years. Here, a query consists of a triplet  $(s, t, F)$ , and it is required to determine if  $s, t$  are connected in  $G - F$ . The study of  $f$ -FT  $s$ - $t$  connectivity labeling was initiated by Dory and Parter [23] with subsequent works in [32, 41, 42, 37], rendering it reasonably well-understood by now: it is known to admit very succinct labels of only  $\text{poly}(f, \log n)$  bits, either with edge failures or with vertex failures; we refer to [37] for a detailed presentation of the state-of-the-art bounds.

**Global connectivity.** Parter, Petruschka and Pettie [42] have proposed the “global connectivity” variant, where the query consists only of  $F$ , and it is required to determine if  $G - F$  is a connected graph. Unlike with  $s$ - $t$  connectivity, the global variant turns out to behave very differently with edge failures ( $F \subseteq E$ ) or with vertex failures ( $F \subseteq V$ ). Essentially all known labeling schemes for the  $s$ - $t$  connectivity variant under edge failures [23, 32, 37] also solve the global variant en route, hence the latter also admits succinct  $\text{poly}(f, \log n)$ -bit labels. However, in case of vertex failures, it was shown in [42] that polynomial dependency in  $f$  is impossible, and their lower bound was improved to  $\Omega(n^{1-1/f}/f)$  by Long, Pettie and Saranurak [37]. Very recently, Jiang, Parter and Petruschka [33] gave a nearly matching upper bound by providing a construction of  $f$ -FT global connectivity labels of  $n^{1-1/f} \cdot \text{poly}(f, \log n)$  bits, essentially settling the problem.

**Steiner connectivity.** In this work, we study the natural generalization of the problem to *Steiner connectivity*. In this variant, the graph  $G = (V, E)$  comes with a (fixed) subset of designated vertices  $U \subseteq V$  called *terminals*. Given a query  $F \subseteq V \cup E$  of up to  $f$  faulty vertices/edges, it is required to report if  $F$  is a  $U$ -Steiner cut in  $G$ , i.e., if there is a pair of disconnected terminals in  $G - F$ . Note that  $U$  is fixed and not a part of the query, so we only get to see the labels of  $F$ .<sup>1</sup> We call this variant  $f$ -FT Steiner connectivity labeling scheme.

We first note that the lower bound for the global connectivity variant of [37] immediately yields a lower bound for the Steiner setting as well:<sup>2</sup>

► **Theorem 1** ([37]). *Any  $f$ -FT Steiner connectivity labeling schemes for  $n$ -vertex graphs with  $k$  terminals must have label length of  $\Omega(k^{1-1/f}/f)$  bits (in the worst case).*

Our technical contribution is providing a nearly matching upper bound which is optimal up to  $\text{poly}(f, \log n)$  factors, hence generalizing the result of [33] (recovered when  $U = V$ ):

<sup>1</sup> If we also get the labels of  $U$ , we can just use the  $\text{poly}(f, \log n)$ -bit labels of the  $s$ - $t$  variant, and apply  $|U| - 1$  pairwise connectivity queries (between one terminal in  $U$  to all the others) in  $G - F$ .

<sup>2</sup> Construct the lower bound instance of [37] on  $k$  vertices set as terminals, and add  $n - k$  isolated vertices.

► **Theorem 2.** *For every  $n$ -vertex graph  $G = (V, E)$  with terminal set  $U \subseteq V$  and  $f \geq 1$ , there is an  $f$ -FT Steiner connectivity labeling scheme for  $(G, U)$  with labels of  $|U|^{1-1/f} \cdot \text{poly}(f, \log n)$  bits. The labels are computed deterministically in polynomial time.*

The  $f$ -FT Steiner connectivity labeling scheme of Theorem 2 supports any query  $F \subseteq V \cup E$  of vertices and edges with  $|F| \leq f$ . However, the challenge lies entirely in the case of vertex failures, i.e.,  $F \subseteq V$ : proving Theorem 2 for this case implies also the general case. Indeed, we can construct a new instance  $(G', U)$  where  $G'$  is obtained from  $G$  by splitting each edge  $e$  of  $G$  by a new middle  $v_e$ , so the failure of  $e$  in  $G$  is equivalent to the failure of  $v_e$  in  $G'$  (in terms of affect on connectivity between original vertices).

As for the case of only edge failures  $F \subseteq E$ , much like in the global connectivity variant, it behaves very differently and admits labels of  $\text{poly}(f, \log n)$  bits, implicit by known constructions of  $f$ -FT connectivity labels for pairwise connectivity queries (the standard  $s$ - $t$  variant in the literature). Specifically, both the labels of [23] based on linear graph sketches and its derandomized version in [32] are straightforward to augment to answer Steiner connectivity queries,<sup>3</sup> and other constructions in [23, 37] seem plausibly amenable to such augmentations as well. In light of the discussions above, in the rest of the paper we will only consider the case of vertex failures, i.e., queries are of the form  $F \subseteq V$  with  $|F| \leq f$ .

**Related work: Steiner variants.** We mention that providing “Steiner generalizations” that interpolate between the two extreme scenarios of the terminal set ( $|U| = n$  and  $|U| = 2$ ) is a common objective in various algorithmic and graph-theoretic problems, such as computing Steiner mincut [22, 17, 29, 20], data structures for Steiner mincuts [21, 22, 9] and minimum+1 Steiner cuts [11], sensitivity oracles for Steiner mincuts [10, 11], Steiner connectivity augmentation and splitting-off [16] and construction of cactus graphs for compactly storing and characterizing Steiner mincuts [22, 29].

**Related work: connectivity oracles.** FT connectivity labeling schemes can be seen as distributed versions of *connectivity oracles under failures*, which are centralized data structures designed to answer the same kind of queries. These have been extensively studied especially in the  $s$ - $t$  variant, with both edge faults [43, 24, 28] and vertex faults [24, 47, 38, 45, 36, 39]. Much like with labeling schemes, in the global variant of connectivity oracles, vertex failures require much larger complexity than edge failures, as was first observed by Long and Saranurak [38]. Very recently, Jiang, Parter and Petruschka [33] provided almost-optimal oracles for global connectivity under vertex failures, and their approach seems plausible to extend also to the Steiner variant.

## 1.1 Our Approach

While our scheme of Theorem 2 and its formal analysis are not complicated, the intuitions behind them might seem illusive. In this section we aim to clarify these. We first introduce a warm-up suboptimal labeling scheme for the Steiner case, then survey the solution of [33] for the global variant, and finally explain how we “glue together” ideas from both to get our solution. For notational convenience, we use here  $\tilde{O}(\cdot)$  to hide  $\text{poly}(f, \log n)$  factors.

<sup>3</sup> These are based on fixing some spanning tree  $T$  for  $G$ , and reconnecting the connected components of  $T - F$  to obtain those of  $G - F$ . Augment the label of each tree edge with the number of terminals in the subtree below it. Then we can invoke a similar strategy to the sketch computation in [23, 32] to determine the number of terminals in each component of  $T - F$ , and hence also of  $G - F$ .

A basic building block are the previously-introduced  $f$ -FT labels for the  $s$ - $t$  connectivity variant, with length  $\tilde{O}(1)$  bits [42, 37]. Their construction is highly complicated, but we use them as a black-box. Let  $\ell(\cdot)$  denote these labels, and  $L(\cdot)$  the labels that we construct; each  $L(\cdot)$ -label will store many  $\ell(\cdot)$ -labels.

**Warm-up: using degree classification.** To gain intuition, we first sketch a *suboptimal* construction based on a strategy introduced in [41]. Assume first  $U = V$ . Choose an arbitrary spanning tree  $T$  in  $G$ . Let  $r$  be some degree threshold. Let  $B$  be the set of vertices with  $T$ -degree  $\geq r$  (high-deg), and  $V - B$  be the rest (low-deg). For a low-deg  $x \in V - B$ , let  $L(x)$  store  $\{\ell(x)\} \cup \{\ell(y) \mid y \text{ is } T\text{-neighbor of } x\}$ . This handles the following queries:

- **(all-low-deg)** If  $F \subseteq V - B$ , use the  $\ell(\cdot)$ -labels to check pairwise connectivity between all  $T$ -neighbors of  $F$ ; one observes that  $F$  is a cut in  $G$  iff we find some disconnected pair.

To handle the remaining queries where  $F \cap B \neq \emptyset$ , we use recursion. For each  $x \in V$  let  $L(x)$  store, for every  $y \in B$ ,  $x$ 's label to handle  $f - 1$  faults in  $G - \{y\}$  (recursively). Then,

- **(one-high-deg)** If there is some  $y \in F \cap B$ , we answer the query  $F - \{y\}$  in  $G - y$ .

Let  $b_f$  denote the resulting label length. As  $|B| = O(n/r)$ , we get  $b_f = \tilde{O}(r + (n/r) \cdot b_{f-1})$ . Optimizing  $r$  and solving the recurrence (with base case  $b_1 = 1$ ) yields  $b_f = \tilde{O}(n^{1-1/2^{f-1}})$ .

Generalizing to the Steiner variant with arbitrary terminals  $U \subseteq V$  is rather smooth. Essentially, one takes  $T$  as a *minimal Steiner tree* that connects all terminals  $U$ . Then high-deg vertices in  $T$  are only  $|B| \leq O(|U|/r)$ , which eventually results in  $\tilde{O}(|U|^{1-1/2^{f-1}})$ -bit labels. The details are provided in the full version of the paper.

**Existing solution: the global case.** As previously mentioned, Jiang, Parter and Petruschka [33] gave near-optimal labels for the  $f$ -FT global connectivity variant, i.e., the case  $U = V$ . Roughly speaking, they also use a high/low-deg partition  $B$  and  $V - B$  (with threshold  $r$ ), but divide queries  $F$  in a dual manner to the warm-up above: **(all-high-deg)**  $F \subseteq B$ , or **(one-low-deg)**  $F \cap (V - B) \neq \emptyset$ . The high/low-deg partition is w.r.t. to degrees in  $G$ , not in a spanning tree. To enable this, they use *Nagamochi-Ibaraki sparsification* [40], yielding a subgraph  $H$  of  $G$  with  $\tilde{O}(n)$  edges such that  $G - F$  and  $H - F$  have the same connected components, for every  $F \subseteq V$ ,  $|F| \leq f$ . So w.l.o.g.,  $G$  has only  $m = \tilde{O}(n)$  edges.

To see the broad strokes, we explain their solution in the special case that  $G$  is  $f$ -vertex-connected. As [33] observe, in this case, if the query  $F$  is indeed a cut, then every  $x \in F$  is adjacent to every connected component of  $G - F$ . Thus, for a low-deg  $x \in V - B$ , one lets  $L(x)$  store  $\{\ell(y) \mid (x, y) \in E\}$ , and answers a query with  $x \in F$  by looking for a pair of neighbors disconnected in  $G - F$ . Note that this argument crucially hinges on the fact that these are all of  $x$ 's neighbors in the (sparsified) graph  $G$ , and not in a spanning tree. To handle queries  $F \subseteq B$ , one observes that a high-deg  $x \in B$  can participate in at most  $O(|B|^{f-1})$  such queries, and simply writes the answers to these inside  $L(x)$ . As  $|B| \leq O(m/r)$ , the label length is  $\tilde{O}(r + (m/r)^{f-1})$ . Setting  $r = m^{1-1/f}$  gives  $\tilde{O}(m^{1-1/f}) = \tilde{O}(n^{1-1/f})$ .

**Our new solution: the Steiner case.** The main issue that prevents extending the solution of [33] to the Steiner case with arbitrary terminals  $U \subseteq V$  lies in the sparsification step. Intuitively, to get  $\tilde{O}(|U|^{1-1/f})$  label length, we would like to have  $\tilde{O}(|U|)$  edges in the sparsified subgraph  $H$ , but then preserving connected components under faults is clearly impossible:  $|U|$  can be much smaller than  $|V|$ , so such  $H$  cannot even be connected. To tackle this, one may wish to construct other meaningful sparsifiers for Steiner connectivity under faults. This seems to be a challenging task, as even a plausible definition is non-trivial.

Instead, we take a different route and combine the “best-of-both-worlds”: we use a Steiner tree for high/low-deg partition as in the warm-up, but (roughly speaking) divide queries  $F$  to *all-high-deg* or *one-low-deg* as in [33]. Thus, even in the global case  $U = V$ , our approach yields a near-optimal scheme which is different than that of [33].

On a high level, the key is not using just any (minimal) Steiner tree  $T$ , but one where high-degree vertices  $B$  admit a lot of structure, by using a powerful graph decomposition of Duan and Pettie [24]. While it still holds that  $|B| \leq O(|U|/r)$ , the additional structure is that after deleting  $B$  from  $G$ , the remaining forest  $T - B$  is still a Steiner forest for  $U - B$  (i.e., connected terminals in  $G - B$  are also connected in  $T - B$ ). The main technical part of our solution lies in a structural analysis that studies the interplay between the (known-in-advance)  $G - B$  and  $T - B$ , and the terminals in the graph  $G - F$  defined by a given query  $F$  (this is mostly reflected in the correctness proof, specifically Claim 10). Essentially, this extra structure lets us design labels where it is enough for low-deg vertices to be defined w.r.t. the tree, and store the  $\ell(\cdot)$ -labels of only their tree neighbors.

## 2 Preliminaries

Throughout the paper,  $G = (V, E)$  is an undirected graph with  $n$  vertices,  $U \subseteq V$  is a designated terminal set in  $G$ , and  $f \geq 1$  is an integer parameter.

For  $K, W \subseteq V$ , we say that  $K$  *separates*  $W$  if there exist two vertices  $w, w' \in W - K$  that are disconnected in  $G - K$ . A vertex set  $F \subseteq V$  is called a *cut* in  $G$  if it separates  $V$ , and a *Steiner cut* if it separates the terminals  $U$ .

To shorten statements of theorems, lemmas, etc., we always use the term “algorithm” to mean a deterministic polynomial-time algorithm.

## 3 Basic Tools

As mentioned in Section 1.1, the most basic tool, which we use as a black box, is the succinct  $f$ -FT labels for  $s$ - $t$  connectivity:

► **Theorem 3** ([37]). *There is an algorithm that outputs an  $O(f^4 \log^{7.5}(n))$ -bit label  $\ell(v)$  to every  $v \in V$  with the following property:*

*For every  $F \subseteq V$  with  $|F| \leq f$  and  $s, t \in V$ , one can determine if  $s, t$  are connected in  $G - F$ , by only inspecting the  $\ell(\cdot)$ -labels of  $F \cup \{s, t\}$ .*

As explained in Section 1.1, our labels  $L(\cdot)$  of Theorem 2 store many  $\ell(\cdot)$  labels, and to answer a query  $F$ , we make many  $s$ - $t$  connectivity queries in  $G - F$  using these  $\ell(\cdot)$ -labels. Suppose that during the query algorithm, we found in this way a pair  $s, t$  which is disconnected in  $G - F$ . Then clearly  $F$  is a cut in  $G$ , but we still cannot determine that is a *Steiner cut*, as one of  $s, t$  might lie in a connected component of  $G - F$  without any terminals. We now present a simple tool that augments Theorem 3 to help us detect this situation.

► **Lemma 4.** *There is an algorithm that outputs an  $O(f^4 \log^{7.5})$ -bit label  $\bar{\ell}(v)$  to every  $v \in V$  with the following property:*

*For every  $F \subseteq V$  with  $|F| \leq f$  and  $x \in V$ , one can determine there exists some  $u \in U$  such that  $x, u$  are connected in  $G - F$ , by only inspecting the  $\bar{\ell}(\cdot)$ -labels of  $F \cup \{x\}$ .*

**Proof.** Construct an auxiliary graph  $G'$  from  $G$  by adding a new vertex  $z$  and adding an edge  $(z, u)$  for every terminal  $u \in U$ . Let  $\ell'(\cdot)$  be the labels computed by the algorithm of Theorem 3 on  $G'$ , and define  $\bar{\ell}(v) := (\ell'(v), \ell'(z))$ . Now, let  $F \subseteq V$  with  $|F| \leq f$  and

$x \in V$ . Note that there exists some  $u \in U$  which is connected to  $x$  in  $G - F$  iff  $x, z$  are connected in  $G' - F$ . The  $\bar{\ell}(\cdot)$ -labels of  $F \cup \{x\}$  contain all  $\ell'(\cdot)$ -labels of  $F \cup \{x, z\}$ , so we can use them to determine if the latter condition holds.  $\blacktriangleleft$

## 4 The Labeling Scheme: Proof of Theorem 2

In this section, we prove our main Theorem 2, giving  $f$ -FT Steiner connectivity labels of  $|U|^{1-1/f} \cdot \text{poly}(f, \log n)$  bits. Recall that, as discussed in the introduction, we only need to consider up to  $f$  vertex failures, i.e., queries  $F \subseteq V$  with  $|F| \leq f$ .

**Subset labels.** We first introduce a somewhat technical lemma. Roughly speaking, in terms of the high-level intuition of Section 1.1, it serves to treat the *all-high-degree* case. The idea is to store explicit information for some relatively small collection of vertex-subsets (of “high-deg vertices”, see Section 1.1), such we have the information on  $K \subseteq F$ , we can determine if  $K$  itself already separates  $U - F$ , which implies that  $F$  is a Steiner cut.

► **Lemma 5.** *There is an algorithm that given any  $K \subseteq V$ , outputs a subset label  $\hat{\ell}(K)$  of  $O(f \log n)$  bits with the following property:*

*For every  $F \subseteq V$  with  $|F| \leq f$ , one can determine if  $K$  separates  $U - F$ , by only inspecting  $\hat{\ell}(K)$  and the names/identifiers of the vertices in  $F$ .*

**Proof.** Let  $C_1, \dots, C_p$  be the connected components of  $G - K$  that contain some terminal, and denote  $U_i = C_i \cap U$ , where w.l.o.g.  $|U_1| \leq \dots \leq |U_p|$ . Our goal is that given  $\hat{\ell}(K)$  and any  $F \subseteq V$ ,  $|F| \leq f$ , we can decide if at least two sets in  $\{U_i - F \mid 1 \leq i \leq p\}$  are non-empty.

Suppose there exists some  $q$  such that  $\sum_{i=1}^q |U_i| > f$  and  $\sum_{i=q+1}^p |U_i| > f$ . Then for every  $F \subseteq V$  with  $|F| \leq f$  there is one non-empty set in  $\{U_i - F \mid i \leq q\}$  and another in  $\{U_i - F \mid i \geq q+1\}$ . Thus, we just indicate this situation in  $\hat{\ell}(K)$  and always answer YES.

Assume now that no such  $q$  exists.

- If  $|U_p| > f$ : Then  $\sum_{i=1}^{p-1} |U_i| < f$ , and we let  $\hat{\ell}(K)$  store  $U_1, \dots, U_{p-1}$  (and indicate that  $|U_p| > f$ ). This suffices as for every  $F \subseteq V$  with  $|F| \leq f$  we know that  $U_p - F \neq \emptyset$ , so we only need to decide if  $\{U_i - F \mid 1 \leq i \leq p-1\}$  has a non-empty set.
- If  $|U_p| \leq f$ : We assert that the sum  $\sum_{i=1}^p |U_i|$  is at most  $3f$ . Indeed, if it is more than  $f$ , let  $q$  be maximal such that  $\sum_{i=q+1}^p |U_i| > f$ . Then  $\sum_{i=1}^q |U_i| \leq f$ ,  $\sum_{i=q+2}^p |U_i| \leq f$  (by maximality), and  $|U_{q+1}| \leq |U_p| \leq f$ . Thus, we just let  $\hat{\ell}(K)$  store all of  $U_1, \dots, U_p$ .  $\blacktriangleleft$

**The Duan-Pettie Decomposition.** Our label construction crucially relies on a graph decomposition algorithm of Duan-Pettie [24] (as explained in Section 1.1). This algorithm is a recursive version of the Fürer-Raghavachari [25] algorithm for finding a Steiner tree with smallest possible maximum degree, up to  $+1$  error. The Duan-Pettie decomposition utilizes the structural properties of the Fürer-Raghavachari tree (rather than its strong approximation guarantee). In the following, a *Steiner forest* means a forest in the graph where each pair of connected terminals (in the graph) is also connected in the forest.

► **Theorem 6 ([24]).** *Let  $r \geq 3$  be a degree threshold. There is an algorithm  $\text{Decomp}(G, U, r)$  that outputs a pair  $(T, B)$  such that the following hold:*

1.  $T$  is a Steiner forest for  $U$  in  $G$ , and  $T - B$  is a Steiner forest for  $U - B$  in  $G - B$ .
2. The maximum degree in the forest  $T - B$  is at most  $r$ .
3.  $|B| < |U|/(r-2)$  and  $|B \cap U| < |U|/(r-1)$ .

*The running time of  $\text{Decomp}(G, U, r)$  is  $O(|U|m \log |U|)$ .*



In the following we assume that  $f \geq 2$ , as Theorem 2 is trivial when  $f = 1$ : the label of vertex  $x$  just stores a single bit indicating whether  $\{x\}$  is a Steiner cut.

**Labeling.** Our first step is to invoke  $\text{Decomp}(G, U, r := |U|^{1-1/f} + 2)$  of Theorem 6; let  $(T, B)$  be the output. For every  $x \in V$ , we choose an arbitrary terminal  $u_x \in U$  such that  $x, u_x$  are connected in  $G - B$  (where  $u_x := \perp$ , i.e., a null value, if there is no such terminal). Next, we construct the  $\ell(\cdot)$ -labels and  $\hat{\ell}(\cdot)$ -labels of Theorem 3 and Lemma 4; for every  $v \in V$ , define  $\ell^*(v) := (v, \ell(v), \hat{\ell}(v))$ . Finally, the  $L(\cdot)$  labels are constructed as follows:

■ **Algorithm 1** Creating the label  $L(x)$  of a  $x \in V$ .

---

```

1: store  $\ell^*(x)$ 
2: store  $\ell^*(u_x)$  (or  $\perp$  if  $u_x = \perp$ )
3: store  $\{\ell^*(b) \mid b \in B\}$ 
4: store  $\hat{\ell}(K = \emptyset)$  of Lemma 5 (i.e., applying this lemma with  $K = \emptyset$ )
5: if  $x \notin B$  then
6:   store  $\{\ell^*(y) \mid (x, y) \text{ is an edge of } T - B\}$ 
7: else  $\triangleright x \in B$ 
8:   for every  $K \subseteq B$  s.t.  $x \in K$  and  $|K| \leq f$  do
9:     store  $K$  along with  $\hat{\ell}(K)$  of Lemma 5

```

---

The label length analysis is rather straightforward, given in detail in the following claim.

▷ **Claim 7.** Each label  $L(x)$  generated by Algorithm 1 has  $|U|^{1-1/f} \cdot \text{poly}(f, \log n)$  bits, and takes polynomial time to compute.

**Proof.** By Theorem 3, Lemma 4 and Lemma 5 each  $\ell^*(\cdot)$ -label or  $\hat{\ell}(\cdot)$ -label consists of  $\text{poly}(f, \log n)$  bits, so we show that only  $O(|U|^{1-1/f})$  such labels are stored in  $L(x)$ . The first four lines only store  $O(|B|)$  such labels, and  $|B| \leq |U|/(r-2) = |U|^{1/f}$  by the properties of  $\text{Decomp}(G, U, r)$  in Theorem 6). As we assume  $f \geq 2$ ,  $|U|^{1/f} \leq |U|^{1-1/f}$ , so this is within budget. If  $x \notin B$ , then, line 6 stores  $\ell^*(y)$  for each neighbor  $y$  of  $x$  in  $T - B$ , which are at most  $r = O(|U|^{1-1/f})$  again by Theorem 6. Next suppose  $x \in B$ . The number of sets  $K \subseteq B$  s.t.  $x \in K$  and  $|K| \leq f$  can be bounded by

$$\sum_{k=0}^{f-1} \binom{|B|-1}{k} \leq 1 + \sum_{k=1}^{f-1} \left( \frac{|B| \cdot e}{k} \right)^k \leq 1 + |B|^{f-1} \cdot \sum_{k=1}^{f-1} \left( \frac{e}{k} \right)^k \leq O(|U|^{1-1/f})$$

(we used  $\binom{n}{k} \leq (n/k)^k$  for  $1 \leq k \leq n$ ,  $\sum_{k=1}^{\infty} (e/k)^k$  is convergent, and  $|B| \leq |U|^{1/f}$ ). Thus, lines 8 and 9 store  $O(|U|^{1-1/f})$   $\hat{\ell}(\cdot)$ -labels.

Note that constructing the  $\ell^*(\cdot)$  labels for all vertices takes polynomial time by Theorem 3 and Lemma 4. Also, we only need to construct  $\hat{\ell}(K)$  labels for sets  $K \subseteq B$  such that  $x \in B$  and  $|K| \leq f$  (or such that  $K = \emptyset$ ), and there are only  $O(|U|^{1-1/f})$  such sets  $K$  by the arguments above; constructing each  $\hat{\ell}(K)$  takes polynomial time by Lemma 5. ◁

► **Remark 8.** The subset labels of Lemma 5 are trivial to construct with  $O(f^2 \log n)$  bits: with notations from the proof, choose arbitrary  $f+1$  sets among  $U_1, \dots, U_p$  (or all if  $p \leq f$ ), take  $f+1$  terminals from each chosen  $U_i$  (or all if  $|U_i| \leq f$ ). We've included the more efficient version to demonstrate that the  $\text{poly}(f, \log n)$  factors in Claim 7 come from the  $f$ -FT labels of the  $s$ -t variant (Theorem 3). These have an  $\Omega(f + \log n)$ -bit lower bound bits [42], so even if the latter is matched, the contribution of the subset labels is lightweight.

**Query algorithm.** The pseudo-code for the query algorithm is given in the following Algorithm 2, designed to return *YES* if the query  $F$  is a Steiner cut, and *NO* otherwise.

■ **Algorithm 2** Answering a query  $F \subseteq V$ ,  $|F| \leq f$  from the labels  $\{L(x) \mid x \in F\}$ .

---

```

1: let  $K := F \cap B$ 
2: if  $K$  separates  $U - F$  in  $G$  then return YES
3: let  $S := \{v \in V - F \mid \ell^*(v) \text{ is stored in } L(x) \text{ of some } x \in F\}$ 
4: let  $W := \{w \in S \mid \exists u \in U \text{ s.t. } u, w \text{ are connected in } G - F\}$ 
5: if  $W \neq \emptyset$  then
6:   choose an arbitrary  $w^* \in W$ 
7:   for every  $w \in W - \{w^*\}$  do
8:     if  $w^*, w$  are disconnected in  $G - F$  then return YES
9: return NO

```

---

We now provide the implementation details:

- Line 2 is executed using  $\hat{\ell}(K)$  of Lemma 5 (and the names of the vertices in  $F$ , given in their labels). If  $K = F \cap B$  is non-empty, we can find  $\hat{\ell}(K)$  in the label  $L(x)$  of some  $x \in F \cap B$ . Otherwise,  $\hat{\ell}(K)$  is stored in  $L(x)$  for every  $x \in F$ .
- Line 4 is executed by using the  $\bar{\ell}(\cdot)$ -labels of the vertices in  $S \cup F$  (which appear in their  $\ell^*(\cdot)$ -labels). For every  $w \in S$ , we use  $\bar{\ell}(w)$  and  $\{\bar{\ell}(x) \mid x \in F\}$  to decide if  $w$  is connected to some  $u \in U$  in  $G - F$ , which is possible by Lemma 4.
- Line 8 is executed using the  $\ell(\cdot)$ -labels of  $w^*, w$  and  $F$  (which appear in their  $\ell^*(\cdot)$ -labels), which enable to determine if  $w^*, w$  are connected in  $G - F$  by Theorem 3.

**Correctness.** We need to show that *YES* is returned if and only if  $F$  is a Steiner cut, namely  $F$  separates  $U$  in  $G$ . The soundness direction (‘only if’) is trivial:

▷ **Claim 9.** If the query algorithm (Algorithm 2) returns *YES* then  $F$  is a Steiner cut in  $G$ .

*Proof.* If *YES* was returned in line 2, then  $F$  has a subset  $K$  which separates two terminals in  $U - F$ , so these are also separated by  $F$ . Thus,  $F$  is a Steiner cut.

Otherwise, *YES* was returned in line 8, so there are  $w^*, w \in W$  disconnected in  $G - F$ . By definition of  $W$  (line 4 of Algorithm 1), there exists terminals  $u^*, u \in U$  such that  $u^*, w^*$  and  $u, w$  are connected pairs in  $G - F$ . Thus,  $u^*, u$  are two terminals disconnected in  $G - F$ , so  $F$  is a Steiner cut.  $\triangleleft$

Next, we address the completeness (‘if’) direction:

▷ **Claim 10.** If  $F$  is a Steiner cut in  $G$ , then the query algorithm (Algorithm 2) returns *YES*.

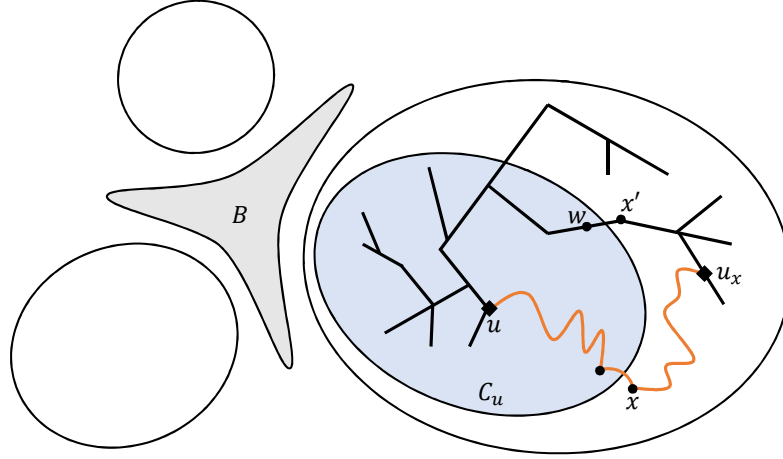
*Proof.* Let us first consider the trivial case when  $K = F \cap B$  separates  $U - F$  in  $G$ . Then Algorithm 2 returns *YES* in line 2 as needed. Henceforth, we assume that  $F$  satisfies the following property:

(P1)  $F \cap B$  does not separate  $U - F$  in  $G$ .

Note that given Property (P1), the only possible place where Algorithm 2 can return *YES* is in line 8, so we aim to show that this is indeed the case. We will crucially use Property (P1) later on in the proof.

Let  $u, u' \in U$  be two terminals disconnected in  $G - F$ , which exists as  $F$  is a Steiner cut. We will show there exists  $w, w' \in W$  such that (i)  $u, w$  are connected in  $G - F$  and (ii)  $u', w'$  are connected in  $G - F$ . Let us argue why this suffices to complete the proof. First, this





■ **Figure 1** Illustrating the proof of Claim 10. Given (P1) and (P2), the connected component  $C_u$  of  $u$  in  $G - F$  is fully contained in some connected component of  $G - B$ , and  $C_u$  has some neighbor  $x \in F$  within this bigger component. Thus, both  $u$  and the chosen terminal  $u_x$  for  $x$  are connected to  $x$  in this component. The forest  $T - B$  contains a tree that connects *all terminals* in this component, so it must connect  $u$  and  $u_x$ . If  $u_x \notin C_u$  (as shown in the figure), then the tree must contain a vertex  $w \in C_u$  which is a neighbor of some  $x' \in F$  in  $T - B$ , and so  $w \in S$ . Otherwise,  $u_x \in C_u \cap S$ . In either case, we find some vertex in  $C_u \cap S$  as required.

clearly implies that  $W \neq \emptyset$ . Second, one of  $w, w'$  must be disconnected from  $w^*$  in  $G - F$ , as otherwise  $w, w'$  would be connected in  $G - F$ , so by (i) and (ii) we get that  $u, u'$  are also connected in  $G - F$ , but this is a contradiction. Therefore, line 8 of Algorithm 2, when executed with either  $w$  or  $w'$  must return *YES* as required.

We thus focus on showing the existence of  $w \in W$  which is connected to  $u$  in  $G - F$  (and the proof for  $w'$  and  $u'$  is symmetric). Let  $C_u$  be the connected component of  $u$  in  $G - F$ . Recall the definitions of  $S$  and its subset  $W$  from lines 3 and 4 of Algorithm 2. Note that every vertex  $w \in C_u \cap S$  is automatically in  $W$ , since it belongs to  $S$  and is connected to the terminal  $u$  in  $G - F$ . So, our goal is simply to show that  $C_u \cap S \neq \emptyset$ , or in other words, that the  $\ell^*(\cdot)$ -label of some vertex in  $C_u$  is stored in some  $L(x)$  of  $x \in F$ .

Again, there is a trivial case: if  $C_u \cap B \neq \emptyset$  then we are done, because  $\{\ell^*(b) \mid b \in B\}$  is stored in every  $L(x)$  of  $x \in F$  (line 3 of Algorithm 1), so  $B \subseteq S$ . Thus, we assume that:

**(P2)** The connected component  $C_u$  of  $u$  in  $G - F$  does not intersect  $B$ .

In fact, the inclusion of the  $\ell^*(\cdot)$ -labels of  $B$  in every label  $L(x)$  is precisely aimed at enforcing Property (P2).

Our next step is to “capitalize on” properties (P1), (P2), and the important property that  $T - B$  is a Steiner forest for  $U - B$  in  $G - B$  (guaranteed by Theorem 6). Consider the set  $N_G(C_u)$  of all vertices outside  $C_u$  that are adjacent to  $C_u$  in  $G$ . Note that, as  $C_u$  is a connected component in  $G - F$ , we have  $N_G(C_u) \subseteq F$ . Observe that  $N_G(C_u)$  separates  $U - F$  in  $G$ , as  $u, u' \in U - F$  with  $u \in C_u$  and  $u' \notin C_u$ . Thus, by Property (P1), it must be that  $N_G(C_u) \not\subseteq F \cap B$ . Hence, there exists some  $x \in N_G(C_u) \cap (F - B)$ . Now, using Property (P2), we see that all vertices in  $C_u \cup \{x\}$  are connected in  $G - B$ , so  $u$  is connected to  $x$  in  $G - B$ . Recall that the terminal  $u_x \in U$  is also (by definition) connected to  $x$  in  $G - B$ . (Note that  $u_x$  exists since  $x$  is connected to the terminal  $u$  in  $G - B$ .) Hence,  $u, u_x$  are connected in  $G - B$ , and thus also in  $T - B$ . This is the crux for finishing the proof in the following argument. See Figure 1 for an illustration.

Observe that  $u_x \in S$ , since  $\ell^*(u_x)$  is stored in  $L(x)$  (line 2 of Algorithm 1). So, if  $u, u_x$  are connected in  $G - F$ , then  $u_x \in C_u \cap S$  and we are done. Otherwise, the path in  $T - B$  between  $u$  and  $u_x$  must go through  $F$ . Let  $x' \in F$  be the vertex closest to  $u$  on this path, and let  $w$  be its neighbor in the direction of  $u$  on the path. (It might be that  $x' = x$ .) Then the subpath between  $u$  and  $w$  avoids  $F$ , so  $w \in C_u$ . Also,  $x' \notin B$  and  $(x', w)$  is an edge of  $T - B$ , hence  $\ell^*(w)$  is stored in  $L(x')$  (line 6 of Algorithm 1), so  $w \in S$ . Namely, we have shown that  $C_u \cap S \neq \emptyset$ , and the proof is concluded.  $\triangleleft$

Theorem 2 follows immediately from the combination of Claim 7, Claim 9 and Claim 10.

---

## References

- 1 Serge Abiteboul, Stephen Alstrup, Haim Kaplan, Tova Milo, and Theis Rauhe. Compact labeling scheme for ancestor queries. *SIAM J. Comput.*, 35(6):1295–1309, 2006. doi:10.1137/S0097539703437211.
- 2 Ittai Abraham, Shiri Chechik, and Cyril Gavoille. Fully dynamic approximate distance oracles for planar graphs via forbidden-set distance labels. In *Proceedings of the 44th Symposium on Theory of Computing Conference, STOC 2012, New York, NY, USA, May 19 - 22, 2012*, pages 1199–1218. ACM, 2012. doi:10.1145/2213977.2214084.
- 3 Ittai Abraham, Shiri Chechik, Cyril Gavoille, and David Peleg. Forbidden-set distance labels for graphs of bounded doubling dimension. *ACM Trans. Algorithms*, 12(2):22:1–22:17, 2016. doi:10.1145/2818694.
- 4 Stephen Alstrup, Søren Dahlgaard, and Mathias Bæk Tejs Knudsen. Optimal induced universal graphs and adjacency labeling for trees. *J. ACM*, 64(4):27:1–27:22, 2017. doi:10.1145/3088513.
- 5 Stephen Alstrup, Cyril Gavoille, Esben Bistrup Halvorsen, and Holger Petersen. Simpler, faster and shorter labels for distances in graphs. In *Proceedings of the Twenty-Seventh Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2016, Arlington, VA, USA, January 10-12, 2016*, pages 338–350. SIAM, 2016. doi:10.1137/1.9781611974331.CH25.
- 6 Stephen Alstrup, Esben Bistrup Halvorsen, and Kasper Green Larsen. Near-optimal labeling schemes for nearest common ancestors. In *Proceedings of the Twenty-Fifth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2014, Portland, Oregon, USA, January 5-7, 2014*, pages 972–982. SIAM, 2014. doi:10.1137/1.9781611973402.72.
- 7 Stephen Alstrup, Haim Kaplan, Mikkel Thorup, and Uri Zwick. Adjacency labeling schemes and induced-universal graphs. *SIAM J. Discret. Math.*, 33(1):116–137, 2019. doi:10.1137/16M1105967.
- 8 Aviv Bar-Natan, Panagiotis Charalampopoulos, Pawel Gawrychowski, Shay Mozes, and Oren Weimann. Fault-tolerant distance labeling for planar graphs. *Theor. Comput. Sci.*, 918:48–59, 2022. doi:10.1016/J.TCS.2022.03.020.
- 9 Surender Baswana and Abhyuday Pandey. The connectivity carcass of a vertex subset in a graph: both odd and even case. In *2025 Symposium on Simplicity in Algorithms, SOSA 2025, New Orleans, LA, USA, January 13-15, 2025*, pages 385–422. SIAM, 2025. doi:10.1137/1.9781611978315.30.
- 10 Koustav Bhanja. Optimal sensitivity oracle for steiner mincut. In *35th International Symposium on Algorithms and Computation, ISAAC 2024, December 8-11, 2024, Sydney, Australia*, volume 322 of *LIPICs*, pages 10:1–10:18. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2024. doi:10.4230/LIPICs.ISAAC.2024.10.
- 11 Koustav Bhanja. Minimum+1 steiner cuts and dual edge sensitivity oracle: Bridging gap between global and (s,t)-cuts. In *The 52nd EATCS International Colloquium on Automata, Languages, and Programming (ICALP)*, 2025.
- 12 Marthe Bonamy, Cyril Gavoille, and Michal Pilipczuk. Shorter labeling schemes for planar graphs. *SIAM J. Discret. Math.*, 36(3):2082–2099, 2022. doi:10.1137/20M1330464.

- 13 Itai Boneh, Shiri Chechik, Shay Golan, Shay Mozes, and Oren Weimann. Optimal fault-tolerant labeling for reachability and approximate distances in directed planar graphs.
- 14 Melvin A. Breuer. Coding the vertexes of a graph. *IEEE Trans. Inf. Theory*, 12(2):148–153, 1966. doi:10.1109/TIT.1966.1053860.
- 15 Melvin A Breuer and Jon Folkman. An unexpected result in coding the vertices of a graph. *Journal of Mathematical Analysis and Applications*, 20(3):583–600, 1967. doi:10.1016/0022-247X(67)90082-0.
- 16 Ruoxu Cen, William He, Jason Li, and Debmalaya Panigrahi. Steiner connectivity augmentation and splitting-off in poly-logarithmic maximum flows. In *Proceedings of the 2023 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 2449–2488. SIAM, 2023. doi:10.1137/1.9781611977554.CH95.
- 17 Richard Cole and Ramesh Hariharan. A fast algorithm for computing steiner edge connectivity. In *Proceedings of the thirty-fifth annual ACM symposium on Theory of computing*, pages 167–176, 2003. doi:10.1145/780542.780568.
- 18 Bruno Courcelle, Cyril Gavoille, Mamadou Moustapha Kanté, and Andrew Twigg. Connectivity check in 3-connected planar graphs with obstacles. *Electron. Notes Discret. Math.*, 31:151–155, 2008. doi:10.1016/J.ENDM.2008.06.030.
- 19 Bruno Courcelle and Andrew Twigg. Compact forbidden-set routing. In *STACS 2007, 24th Annual Symposium on Theoretical Aspects of Computer Science, Aachen, Germany, February 22-24, 2007, Proceedings*, volume 4393 of *Lecture Notes in Computer Science*, pages 37–48. Springer, 2007. doi:10.1007/978-3-540-70918-3\_4.
- 20 Matthew Ding and Jason Li. Deterministic minimum steiner cut in maximum flow time. In *32nd Annual European Symposium on Algorithms, ESA 2024, September 2-4, 2024, Royal Holloway, London, United Kingdom*, volume 308 of *LIPICs*, pages 46:1–46:14. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2024. doi:10.4230/LIPICS.ESA.2024.46.
- 21 Yefim Dinitz and Alek Vainshtein. The connectivity carcass of a vertex subset in a graph and its incremental maintenance. In *Proceedings of the twenty-sixth annual ACM symposium on Theory of computing*, pages 716–725, 1994. doi:10.1145/195058.195442.
- 22 Yefim Dinitz and Alek Vainshtein. The general structure of edge-connectivity of a vertex subset in a graph and its incremental maintenance. odd case. *SIAM J. Comput.*, 30(3):753–808, 2000. doi:10.1137/S0097539797330045.
- 23 Michal Dory and Merav Parter. Fault-tolerant labeling and compact routing schemes. In *PODC '21: ACM Symposium on Principles of Distributed Computing, Virtual Event, Italy, July 26-30, 2021*, pages 445–455. ACM, 2021. doi:10.1145/3465084.3467929.
- 24 Ran Duan and Seth Pettie. Connectivity oracles for graphs subject to vertex failures. *SIAM J. Comput.*, 49(6):1363–1396, 2020. doi:10.1137/17M1146610.
- 25 M. Furer and B. Raghavachari. Approximating the minimum-degree steiner tree to within one of optimal. *Journal of Algorithms*, 17(3):409–423, 1994. doi:10.1006/jagm.1994.1042.
- 26 Cyril Gavoille, David Peleg, Stéphane Pérennes, and Ran Raz. Distance labeling in graphs. *J. Algorithms*, 53(1):85–112, 2004. doi:10.1016/J.JALGOR.2004.05.002.
- 27 Pawel Gawrychowski and Przemyslaw Uznanski. Better distance labeling for unweighted planar graphs. *Algorithmica*, 85(6):1805–1823, 2023. doi:10.1007/S00453-023-01133-Z.
- 28 David Gibb, Bruce M. Kapron, Valerie King, and Nolan Thorn. Dynamic graph connectivity with improved worst case update time and sublinear space. *CoRR*, abs/1509.06464, 2015. arXiv:1509.06464.
- 29 Zhongtian He, Shang-En Huang, and Thatchaphol Saranurak. Cactus representations in polylogarithmic max-flow via maximal isolating mincuts. In *Proceedings of the 2024 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 1465–1502. SIAM, 2024. doi:10.1137/1.9781611977912.60.
- 30 Tai-Hsin Hsu and Hsueh-I Lu. An optimal labeling for node connectivity. In *Algorithms and Computation, 20th International Symposium, ISAAC 2009, Honolulu, Hawaii, USA, December 16-18, 2009. Proceedings*, volume 5878 of *Lecture Notes in Computer Science*, pages 303–310. Springer, 2009. doi:10.1007/978-3-642-10631-6\_32.

- 31 Rani Izsak and Zeev Nutov. A note on labeling schemes for graph connectivity. *Inf. Process. Lett.*, 112(1-2):39–43, 2012. doi:10.1016/J.IPL.2011.10.001.
- 32 Taisuke Izumi, Yuval Emek, Tadashi Wadayama, and Toshimitsu Masuzawa. Deterministic fault-tolerant connectivity labeling scheme. In *Proceedings of the 2023 ACM Symposium on Principles of Distributed Computing, PODC 2023, Orlando, FL, USA, June 19-23, 2023*, pages 190–199. ACM, 2023. doi:10.1145/3583668.3594584.
- 33 Yonggang Jiang, Merav Parter, and Asaf Petruschka. New oracles and labeling schemes for vertex cut queries. *CoRR*, abs/2501.13596, 2025. doi:10.48550/arXiv.2501.13596.
- 34 Sampath Kannan, Moni Naor, and Steven Rudich. Implicit representation of graphs. *SIAM J. Discret. Math.*, 5(4):596–603, 1992. doi:10.1137/0405049.
- 35 Michal Katz, Nir A. Katz, Amos Korman, and David Peleg. Labeling schemes for flow and connectivity. *SIAM J. Comput.*, 34(1):23–40, 2004. doi:10.1137/S0097539703433912.
- 36 Evangelos Kosinas. Connectivity Queries Under Vertex Failures: Not Optimal, but Practical. In *31st Annual European Symposium on Algorithms (ESA 2023)*, volume 274 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 75:1–75:13, Dagstuhl, Germany, 2023. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. doi:10.4230/LIPIcs.ESA.2023.75.
- 37 Yaowei Long, Seth Pettie, and Thatchaphol Saranurak. Connectivity labeling schemes for edge and vertex faults via expander hierarchies. In *Proceedings of the 2025 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 1–47, 2025. doi:10.1137/1.9781611978322.1.
- 38 Yaowei Long and Thatchaphol Saranurak. Near-optimal deterministic vertex-failure connectivity oracles. In *63rd IEEE Annual Symposium on Foundations of Computer Science, FOCS 2022, Denver, CO, USA, October 31 - November 3, 2022*, pages 1002–1010. IEEE, 2022. doi:10.1109/FOCS54457.2022.00098.
- 39 Yaowei Long and Yunfan Wang. Better decremental and fully dynamic sensitivity oracles for subgraph connectivity. In *51st International Colloquium on Automata, Languages, and Programming, ICALP 2024, July 8-12, 2024, Tallinn, Estonia*, volume 297 of *LIPIcs*, pages 109:1–109:20. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2024. doi:10.4230/LIPICS.ICALP.2024.109.
- 40 Hiroshi Nagamochi and Toshihide 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. doi:10.1007/BF01758778.
- 41 Merav Parter and Asaf Petruschka. Optimal dual vertex failure connectivity labels. In Christian Scheideler, editor, *36th International Symposium on Distributed Computing, DISC 2022, October 25-27, 2022, Augusta, Georgia, USA*, volume 246 of *LIPIcs*, pages 32:1–32:19. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2022. doi:10.4230/LIPICS.DISC.2022.32.
- 42 Merav Parter, Asaf Petruschka, and Seth Pettie. Connectivity labeling and routing with multiple vertex failures. In *Proceedings of the 56th Annual ACM Symposium on Theory of Computing, STOC 2024, Vancouver, BC, Canada, June 24-28, 2024*, pages 823–834. ACM, 2024. doi:10.1145/3618260.3649729.
- 43 Mihai Pătraşcu and Mikkel Thorup. Planning for fast connectivity updates. In *48th Annual IEEE Symposium on Foundations of Computer Science (FOCS 2007), October 20-23, 2007, Providence, RI, USA, Proceedings*, pages 263–271. IEEE Computer Society, 2007. doi:10.1109/FOCS.2007.59.
- 44 Seth Pettie, Thatchaphol Saranurak, and Longhui Yin. Optimal vertex connectivity oracles. In *STOC '22: 54th Annual ACM SIGACT Symposium on Theory of Computing, Rome, Italy, June 20 - 24, 2022*, pages 151–161. ACM, 2022. doi:10.1145/3519935.3519945.
- 45 Michal Pilipczuk, Nicole Schirrmacher, Sebastian Siebertz, Szymon Torunczyk, and Alexandre Vigny. Algorithms and data structures for first-order logic with connectivity under vertex failures. In *49th International Colloquium on Automata, Languages, and Programming, ICALP 2022, July 4-8, 2022, Paris, France*, volume 229 of *LIPIcs*, pages 102:1–102:18. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2022. doi:10.4230/LIPICS.ICALP.2022.102.

- 46 Mikkel Thorup and Uri Zwick. Approximate distance oracles. *J. ACM*, 52(1):1–24, 2005. doi:10.1145/1044731.1044732.
- 47 Jan van den Brand and Thatchaphol Saranurak. Sensitive distance and reachability oracles for large batch updates. In *60th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2019, Baltimore, Maryland, USA, November 9-12, 2019*, pages 424–435. IEEE Computer Society, 2019. doi:10.1109/FOCS.2019.00034.