Sparse Fault-Tolerant BFS Trees

Merav Parter *† David Peleg * ‡

October 15, 2013

Abstract

A *fault-tolerant* structure for a network is required to continue functioning following the failure of some of the network's edges or vertices. This paper considers *breadth-first search (BFS)* spanning trees, and addresses the problem of designing a sparse *fault-tolerant* BFS tree, or *FT-BFS tree* for short, namely, a sparse subgraph T of the given network G such that subsequent to the failure of a single edge or vertex, the surviving part T' of T still contains a BFS spanning tree for (the surviving part of) G. For a source node s, a target node t and an edge $e \in G$, the shortest $s-t$ path $P_{s,t,e}$ that does not go through e is known as a *replacement path*. Thus, our `FT-BFS` tree contains the collection of all replacement paths $P_{s,t,e}$ for every $t \in V(G)$ and every failed edge $e \in E(G)$. Our main results are as follows. We present an algorithm that for every n-vertex graph G and source node s constructs a (single edge failure) `FT-BFS` tree rooted at s with $O(n \cdot \min\{\texttt{Depth}(s), \sqrt{n}\})$ edges, where $\texttt{Depth}(s)$ is the depth of the BFS tree rooted at s. This result is complemented by a matching lower bound, showing that there exist n-vertex graphs with a source node s for which any edge (or vertex) `FT-BFS` tree rooted at s has $\Omega(n^{3/2})$ edges. We then consider *fault-tolerant multi-source BFS trees*, or `FT-MBFS` *trees* for short, aiming to provide (following a failure) a BFS tree rooted at each source $s \in S$ for some subset of sources $S \subseteq V$. Again, tight bounds are provided, showing that there exists a poly-time algorithm that for every n-vertex graph and source set $S \subseteq V$ of size σ constructs a (single failure) `FT-MBFS` tree $T^*(S)$ from each source $s_i \in S$, with $O(\sqrt{\sigma} \cdot n^{3/2})$ edges, and on the other hand there exist n-vertex graphs with source sets $S \subseteq V$ of cardinality σ, on which any `FT-MBFS` tree from S has $\Omega(\sqrt{\sigma} \cdot n^{3/2})$ edges. Finally, we propose an $O(\log n)$ approximation algorithm for constructing `FT-BFS` and `FT-MBFS` structures. The latter is complemented by a hardness result stating that there exists no $\Omega(\log n)$ approximation algorithm for these problems under standard complexity assumptions. In comparison with previous constructions our algorithm is deterministic and may improve the number of edges by a factor of up to $\sqrt{n}$ for some instances. All our algorithms can be extended to deal with one *vertex* failure as well, with the same performance.

*The Weizmann Institute of Science, Rehovot, Israel. {`merav.parter,david.peleg`}`@ weizmann.ac.il`. Supported in part by the Israel Science Foundation (grant 894/09), the I-CORE program of the Israel PBC and ISF (grant 4/11), the United States-Israel Binational Science Foundation (grant 2008348), the Israel Ministry of Science and Technology (infrastructures grant), and the Citi Foundation.

†Recipient of the Google Europe Fellowship in distributed computing; research supported in part by this Google Fellowship.

‡A preliminary version of the paper has appeared in the proceedings of ESA2013.

1 Introduction

Background and motivation. Modern day communication networks support a variety of logical structures and services, and depend on their undisrupted operation. As the vertices and edges of the network may occasionally fail or malfunction, it is desirable to make those structures robust against failures. Indeed, the problem of designing fault-tolerant constructions for various network structures and services has received considerable attention over the years.

Fault-resilience can be introduced into the network in several different ways. This paper focuses on a notion of fault-tolerance whereby the structure at hand is augmented or "reinforced" (by adding to it various components) so that subsequent to the failure of some of the network's vertices or edges, the surviving part of the structure is still operational. As this reinforcement carries certain costs, it is desirable to minimize the number of added components.

To illustrate this type of fault tolerance, let us consider the structure of graph k-spanners (cf. [20, 22, 23]). A graph spanner H can be thought of as a skeleton structure that generalizes the concept of spanning trees and allows us to faithfully represent the underlying network using few edges, in the sense that for any two vertices of the network, the distance in the spanner is stretched by only a small factor. More formally, consider a weighted graph G and let $k \geq 1$ be an integer. Let $\text{dist}(u, v, G)$ denote the (weighted) distance between u and v in G. Then a k-spanner H satisfies that $\text{dist}(u, v, H) \leq k \cdot \text{dist}(u, v, G)$ for every $u, v \in V$.

Towards introducing fault tolerance, we say that a subgraph H is an *f-edge fault-tolerant k-spanner* of G if $\text{dist}(u, v, H \setminus F) \leq k \cdot \text{dist}(u, v, G \setminus F)$ for any set $F \subseteq E$ of size at most f, and any pair of vertices $u, v \in V$. A similar definition applies to f-vertex fault-tolerant k-spanners. Sparse fault-tolerant spanner constructions were presented in [7, 12].

This paper considers *breadth-first search (BFS)* spanning trees, and addresses the problem of designing *fault-tolerant* BFS trees, or `FT-BFS` trees for short. By this we mean a subgraph T of the given network G, such that subsequent to the failure of some of the vertices or edges, the surviving part T' of T still contains a BFS spanning tree for the surviving part of G. We also consider a generalized structure referred to as a *fault-tolerant multi-source BFS tree*, or `FT-MBFS` *tree* for short, aiming to provide a BFS tree rooted at each source $s \in S$ for some subset of sources $S \subseteq V$.

The notion of `FT-BFS` trees is closely related to the problem of constructing *replacement paths* and in particular to its *single source* variant, the *single-source replacement paths* problem, studied in [15]. That problem requires to compute the collection $\mathcal{P}_s$ of all $s - t$ replacement paths $P_{s,t,e}$ for every $t \in V$ and every failed edge e that appears on the $s - t$ shortest-path in G. The vast literature on *replacement paths* (cf. [5, 15, 26, 28, 30]) focuses on *time-efficient* computation of the these paths as well as their efficient maintenance in data structures (a.k.a *distance oracles*). In contrast, the main concern in the current paper is with optimizing the *size* of the resulting fault tolerant structure that contains the collection $\mathcal{P}_s$ of all replacement paths given a source node s. A typical motivation for such a setting is where the graph edges represent the channels of a communication network, and the system designer would like to purchase or lease a minimal collection of channels (i.e., a subgraph $G' \subseteq G$) that maintains its functionality as a "BFS tree" with respect to the source s upon any single edge or vertex failure in G. In such a context, the cost of computation at the preprocessing stage may often be negligible compared to the purchasing/leasing cost of the resulting structure. Hence, our key cost measure in this paper is the *size* of the fault tolerant structure, and our main goal is to achieve *sparse* (or *compact*) structures.

Most previous work on sparse / compact fault-tolerant structures and services concerned structures that are *distance-preserving* (i.e., dealing with distances, shortest paths or shortest routes), *global* (i.e., centered on "all-pairs" variants), and *approximate* (i.e., settling for near optimal distances), such as *spanners*, *distance oracles* and *compact routing schemes*. The problem considered here, namely, the construction of `FT-BFS` trees, still concerns a distance preserving structure. However, it deviates from tradition with respect to the two other features, namely, it concerns a "single source" variant, and it insists on exact shortest paths. Hence our problem is on the one hand easier, yet on the other hand harder, than previously studied ones. Noting that in previous studies, the "cost" of adding fault-tolerance (in the relevant complexity measure) was often low (e.g., merely polylogarithmic in the graph size n), one might be tempted to conjecture that a similar phenomenon may reveal itself in our problem as well. Perhaps surprisingly, it turns out that our insistence on exact distances plays a dominant role and makes the problem significantly harder, outweighing our willingness to settle for a "single source" solution.

Contributions. We obtain the following results. In Sec. 2, we define the *Minimum* `FT-BFS` and *Minimum* `FT-MBFS` problems, aiming at finding the minimum such structures tolerant against a single edge or vertex fault. Section 3 presents lower bound constructions for these problems. For the single source case, in Subsec. 3.1, we present a lower bound stating that for every n there exists an n-vertex graph and a source node $s \subseteq V$ for which any `FT-MBFS` tree from s requires $\Omega(n^{3/2})$ edges. In Subsec. 3.2, we then show that there exist n-vertex graphs with source sets $S \subseteq V$ of size σ, on which any `FT-MBFS` tree from the source set S has $\Omega(\sqrt{\sigma} \cdot n^{3/2})$ edges.

These results are complemented by matching upper bounds. In Subsec. 4.1, we present a simple algorithm that for every n-vertex graph G and source node s, constructs a (single edge failure) `FT-BFS` tree rooted at s with $O(n \cdot \min\{\texttt{Depth}(s), \sqrt{n}\})$ edges. A similar algorithm yields an `FT-BFS` tree tolerant to one vertex failure, with the same size bound. In addition, for the multi source case, in Subsec. 4.2, we show that there exists a polynomial time algorithm that for every n-vertex graph and source set $S \subseteq V$ of size $|S| = \sigma$ constructs a (single failure) `FT-MBFS` tree $T^*(S)$ from each source $s_i \in S$, with $O(\sqrt{\sigma} \cdot n^{3/2})$ edges.

In Sec. 5 we consider the minimum `FT-BFS` problem. In Subsec. 5.1, we show that the minimum `FT-BFS` problem is NP-hard and moreover, cannot be approximated (under standard complexity assumptions) to within a factor of $\Omega(\log n)$, where n is the number of vertices of the input graph G. Note that while the algorithms of Sec. 4 match the worst-case lower bounds, they might still be far from optimal for certain instances, as illustrated in Sec. 5. Consequently, in Subsec. 5.2, we complete the upper bound analysis by presenting an $O(\log n)$ approximation algorithm for the Minimum `FT-MBFS` problem. This approximation algorithm is superior in instances where the graph enjoys a sparse `FT-MBFS` tree, hence paying $O(n^{3/2})$ edges (as does the algorithm of Sec. 4) is wasteful. In light of the hardness result for these problems (of Sec.

5.1), the approximability result is tight (up to constants). All our results hold for directed graphs as well.

Related work. To the best of our knowledge, this paper is the first to study the sparsity of fault-tolerant BFS structures for graphs. The question of whether it is possible to construct a sparse fault tolerant *spanner* for an arbitrary undirected weighted graph, raised in [10], was answered in the affirmative in [7], presenting algorithms for constructing an f-vertex fault tolerant $(2k-1)$-spanner of size $O(f^2 k^{f+1} \cdot n^{1+1/k} \log^{1-1/k} n)$ and an f-edge fault tolerant $2k-1$ spanner of size $O(f \cdot n^{1+1/k})$ for a graph of size n. A randomized construction attaining an improved tradeoff for vertex fault-tolerant spanners was shortly afterwards presented in [12], yielding (with high probability) for every graph $G = (V, E)$, odd integer s and integer f, an f-vertex fault-tolerant s-spanner with $O\left(f^{2-\frac{2}{s+1}} n^{1+\frac{2}{s+1}} \log n\right)$ edges. This should be contrasted with the best stretch-size tradeoff currently known for non-fault-tolerant spanners [27], namely, $2k-1$ stretch with $\tilde{O}(n^{1+1/k})$ edges. Fault tolerant spanners for the d-dimensional Euclidean case were studied in [10, 18, 19].

An efficient algorithm that given a set V of n points in d-dimensional Euclidean space constructs an f-vertex fault tolerant geometric $(1+\epsilon)$-spanner for V, namely, a sparse graph H satisfying that $\text{dist}(u, v, H \setminus F) \leq (1+\epsilon)\text{dist}(u, v, G)$ for any set $F \subseteq V$ of size f and any pair of points $u, v \in V \setminus F$, was presented in [18]. A fault tolerant geometric spanner of improved size was later presented in [19]; finally, a fault tolerant geometric spanner of optimal maximum degree and total weight was presented in [10]. The distinction between the stronger type of fault-tolerance obtained for geometric graphs (termed *rigid* fault-tolerance) and the more flexible type required for handling general graphs (termed *competitive* fault-tolerance) is elaborated upon in [21].

A related network service is the *distance oracle* [4, 25, 28], which is a succinct data structure capable of supporting efficient responses to distance queries on a weighted graph G. A distance query (s, t) requires finding, for a given pair of vertices s and t in V, the distance (namely, the length of the shortest path) between u and v in G. The query protocol of an oracle $\mathcal{S}$ correctly answers distance queries on G. In a *fault tolerant distance oracle*, the query may include also a set F of failed edges or vertices (or both), and the oracle $\mathcal{S}$ must return, in response to a query (s, t, F), the distance between s and t in $G' = G \setminus F$. Such a structure is sometimes called an *F-sensitivity distance oracle*. The focus is on both fast preprocessing time, fast query time and low space. It has been shown in [11] that given a directed weighted graph G of size n, it is possible to construct in time $\tilde{O}(mn^2)$ a 1-sensitivity fault tolerant distance oracle of size $O(n^2 \log n)$ capable of answering distance queries in $O(1)$ time in the presence of a single failed edge or vertex. The preprocessing time was recently improved to $\tilde{O}(mn)$, with unchanged size and query time [5]. A 2-sensitivity fault tolerant distance oracle of size $O(n^2 \log^3 n)$, capable of answering 2-sensitivity queries in $O(\log n)$ time, was presented in [13].

Recently, distance sensitivity oracles have been considered for weighted and directed graphs in the *single source* setting [15]. Specifically, Grandoni and Williams considered the problem of *single-source replacement paths* where one aims to compute the collection of all replacement paths for a given source node s, and proposed an efficient randomized algorithm that does so in

$\widetilde{O}(APSP(n,M))$ where $APSP(n,M)$ is the time required to compute all-pairs-shortest-paths in a weighted graph with integer weights $[-M,M]$. Interestingly, although their algorithm does not aim explicitly at minimizing the total number of edges used by the resulting collection of replacement paths, one can show that the resulting construction yields a rather sparse path collection, with at most $O(n^{3/2}\log n)$ edges (although it may also be far from optimal in some instances).

Label-based fault-tolerant distance oracles for graphs of bounded clique-width are presented in [9]. The structure is composed of a label $L(v)$ assigned to each vertex v, and handles queries of the form $(L(s), L(t), F)$ for a set of failures F. For an n-vertex graph of tree-width or clique-width k, the constructed labels are of size $O(k^2\log^2 n)$.

A relaxed variant of distance oracles, in which distance queries are answered by *approximate* distance estimates instead of *exact* ones, was introduced in [28], where it was shown how to construct, for a given weighted undirected n-vertex graph G, an approximate distance oracle of size $O(n^{1+1/k})$ capable of answering distance queries in $O(k)$ time, where the *stretch* (multiplicative approximation factor) of the returned distances is at most $2k-1$.

An f-sensitivity approximate distance oracle $\mathcal{S}$ was presented in [6]. For an integer parameter $k \geq 1$, the size of $\mathcal{S}$ is $O(kn^{1+\frac{8(f+1)}{k+2(f+1)}}\log(nW))$, where W is the weight of the heaviest edge in G, the stretch of the returned distance is $2k-1$, and the query time is $O(|F|\cdot\log^2 n\cdot\log\log n\cdot\log\log d)$, where d is the distance between s and t in $G\setminus F$.

A fault-tolerant label-based $(1+\epsilon)$-approximate distance oracle for the family of graphs with doubling dimension bounded by α is presented in [3]. For an n-vertex graph $G(V,E)$ in this family, and for desired precision parameter $\epsilon > 0$, the distance oracle constructs and stores an $O(\log n/\epsilon^{2\alpha})$-bit label at each vertex. Given the labels of two end-vertices s and t and of collections F_V and F_E of failed (or "forbidden") vertices and edges, the oracle computes, in time polynomial in the length of the labels, an estimate for the distance between s and t in the surviving graph $G(V\setminus F_V, E\setminus F_E)$, which approximates the true distance by a factor of $1+\epsilon$. The case of planar graphs is handled in [2].

Our final example concerns fault tolerant routing schemes. A fault-tolerant routing protocol is a distributed algorithm that, for any set of failed edges F, enables any source vertex $\hat{s}$ to route a message to any destination vertex $\hat{d}$ along a shortest or near-shortest path in the surviving network $G\setminus F$ in an efficient manner (and without knowing F in advance).

In addition to route efficiency, it is often desirable to optimize also the amount of memory stored in the routing tables of the at the cost of lower route efficiency, giving rise to the problem of designing compact routing schemes (cf. [1, 8, 20, 24, 27]). Label-based fault-tolerant routing schemes for graphs of bounded clique-width are presented in [9]. To route from s to t, the source needs to specify the labels $L(s)$ and $L(t)$ and the set of failures F, and the scheme efficiently calculates the shortest path between s and t that avoids F. For an n-vertex graph of tree-width or clique-width k, the constructed labels are of size $O(k^2\log^2 n)$.

Fault-tolerant compact routing schemes are considered in [6], for up to two edge failures. Given a message M destined to t at a source vertex s, presence of a failed edge set F of size $|F| \leq 2$ the scheme presented therein routes M from s to t in a distributed manner, over a path of length at most the length of the optimal path (avoiding F). The total amount of information stored in vertices of G on average is bounded by $O(kn^{1+1/k})$. This should be compared with the best memory-stretch known for non-fault-tolerant compact routing [27], namely, $2k-1$ stretch with $\tilde{O}(n^{1+1/k})$ memory per

A compact routing scheme capable of handling multiple edge failures is presented in [8]. The scheme routes messages (provided their source s and destination t are still connected in the surviving graph $G \setminus F$) over a path whose length is proportional to the distance between s and t in $G \setminus F$, to $|F|^3$ and to some poly-log factor. The routing table required at a node v is of size proportional to v's degree and some poly-log factor.

A routing scheme with stretch $1+\epsilon$ for graphs of bounded doubling dimension is also presented in [3]. The scheme can be generalized also to the family of weighted graphs of bounded doubling dimension and bounded degree. In this case, the label size will also depend linearly on the maximum vertex degree Δ, and this is shown to be necessary.

2 Preliminaries

Notation. Given a graph $G = (V, E)$ and a source node s, let $T_0(s) \subseteq G$ be a shortest paths (or BFS) tree rooted at s. For a source node set $S \subseteq V$, let $T_0(S) = \bigcup_{s \in S} T_0(s)$ be a union of the single source BFS trees. Let $\pi(s, v, T)$ be the $s - v$ shortest-path in tree T, when the tree $T = T_0(s)$, we may omit it and simply write $\pi(s, v)$. Let $\Gamma(v, G)$ be the set of v neighbors in G. Let $E(v, G) = \{(u, v) \in E(G)\}$ be the set of edges incident to v in the graph G and let $\texttt{deg}(v, G) = |E(v, G)|$ denote the degree of node v in G. When the graph G is clear from the context, we may omit it and simply write $\texttt{deg}(v)$. Let $\texttt{depth}(s, v) = \text{dist}(s, v, G)$ denote the *depth* of v in the BFS tree $T_0(s)$. When the source node s is clear from the context, we may omit it and simply write $\texttt{depth}(v)$. Let $\texttt{Depth}(s) = \max_{u \in V}\{\texttt{depth}(s, u)\}$ be the *depth* of $T_0(s)$. For a subgraph $G' = (V', E') \subseteq G$ (where $V' \subseteq V$ and $E' \subseteq E$) and a pair of nodes $u, v \in V$, let $\text{dist}(u, v, G')$ denote the shortest-path distance in edges between u and v in G'. For a path $P = [v_1, \ldots, v_k]$, let $\texttt{LastE}(P)$ be the last edge of path P. Let $|P|$ denote the length of the path and $P[v_i, v_j]$ be the subpath of P from v_i to v_j. For paths P_1 and P_2, $P_1 \circ P_2$ denote the path obtained by concatenating P_2 to P_1. Assuming an edge weight function $W : E(G) \to \mathbb{R}^+$, let $SP(s, v_i, G, W)$ be the set of $s - v_i$ shortest-paths in G according to the edge weights of W. Throughout, the edges of these paths are considered to be directed away from the source node s. Given an $s - v$ path P and an edge $e = (x, y) \in P$, let $\text{dist}(s, e, P)$ be the distance (in edges) between s and e on P. In addition, for an edge $e = (x, y) \in T_0(s)$, define $\text{dist}(s, e) = i$ if $\texttt{depth}(x) = i - 1$ and $\texttt{depth}(y) = i$.

Definition 2.1 *A graph T^* is an edge (resp., vertex)* FT-BFS *tree for G with respect to a source node $s \in V$, iff for every edge $f \in E(G)$ (resp., vertex $f \in V$) and for every $v \in V$,* $\text{dist}(s, v, T^* \setminus \{f\}) = \text{dist}(s, v, G \setminus \{f\})$.

A graph T^* is an edge (resp., vertex) FT-MBFS *tree for G with respect to source set $S \subseteq V$, iff for every edge $f \in E(G)$ (resp., vertex $f \in V$) and for every $s \in S$ and $v \in V$,* $\text{dist}(s, v, T^* \setminus \{f\}) = \text{dist}(s, v, G \setminus \{f\})$.

For simplicity, we refer to edge FT-BFS (resp., edge FT-MBFS) trees simply by FT-BFS (resp., FT-MBFS) trees. Throughout, we focus on edge fault, yet the entire analysis extends trivially to the case of vertex fault as well.

Like other papers in this field [16, 5], throughout, we assume without loss of generality that the shortest paths are unique since we can always add small perturbations to break any ties. Let W be a weight assignment that captures these symbolic perturbations.

The Minimum FT-BFS problem. Denote the set of solutions for the instance (G, s) by $\mathcal{T}(s, G) = \{\widehat{T} \subseteq G \mid \widehat{T}$ is an FT-BFS tree w.r.t. $s\}$. Let $\texttt{Cost}^*(s, G) = \min\{|E(\widehat{T})| \mid \widehat{T} \in \mathcal{T}(s, G)\}$ be the minimum number of edges in any FT-BFS subgraph of G. These definitions naturally extend to the multi-source case where we are given a source set $S \subseteq V$ of size σ. Then

$$\mathcal{T}(S, G) = \{\widehat{T} \subseteq G \mid \widehat{T} \text{ is a FT-MBFS with respect to } S\}$$

and

$$\texttt{Cost}^*(S, G) = \min\{|E(\widehat{T})| \mid \widehat{T} \in \mathcal{T}(S, G)\}.$$

In the *Minimum* FT-BFS problem we are given a graph G and a source node s and the goal is to compute an FT-BFS $\widehat{T} \in \mathcal{T}(s, G)$ of minimum size, i.e., such that $|E(\widehat{T})| = \texttt{Cost}^*(s, G)$. Similarly, in the *Minimum* FT-MBFS problem we are given a graph G and a source node set S and the goal is to compute an FT-MBFS $\widehat{T} \in \mathcal{T}(S, G)$ of minimum size i.e., such that $|E(\widehat{T})| = \texttt{Cost}^*(S, G)$.

3 Lower Bounds

In this section we establish lower bounds on the size of the FT-BFS and FT-MBFS structures. In Subsec. 3.1 we consider the single source case and in Subsec. 3.2 we consider the case of multiple sources.

3.1 Single Source

We begin by presenting a lower bound for the case of a single source.

Theorem 3.1 *There exists an n-vertex graph $G(V, E)$ and a source node $s \in V$ such that any* FT-BFS *tree rooted at s has $\Omega(n^{3/2})$ edges, i.e.,* $\texttt{Cost}^*(s, G) = \Omega(n^{3/2})$.

Proof: Let us first describe the structure of $G = (V, E)$. Set $d = \lfloor \sqrt{n}/2 \rfloor$. The graph consists of four main components. The first is a path $\pi = [s = v_1, \ldots, v_{d+1} = v^*]$ of length d. The second component consists of a node set $Z = \{z_1, \ldots, z_d\}$ and a collection of d disjoint paths of deceasing length, $P_1, \ldots, P_d$, where $P_j = [v_j = p^j_1, \ldots, z_j = p^j_{t_j}]$ connects v_j with z_j and its length is $t_j = |P_j| = 6 + 2(d - j)$, for every $j \in 1, \cdots, d$. Altogether, the set of nodes in these paths, $Q = \bigcup_{j=1}^{d} V(P_j)$, is of size $|Q| = d^2 + 7d$. The third component is a set of nodes X of size $n - (d^2 + 7d)$, all connected to the terminal node v^*. The last component is a complete bipartite graph $B = (X, Z, \hat{E})$ connecting X to Z. Overall, $V = X \cup Q$ and $E = \hat{E} \cup E(\pi) \cup \bigcup_{j=1}^{d} E(P_j)$. Note that $n/4 \leq |Q| \leq n/2$ for sufficiently large n. Consequently, $|X| = n - |Q| \geq n/2$, and

$|\hat{E}| = |Q| \cdot |X| \geq n^{3/2}/4$. A BFS tree T_0 rooted at s for this G (illustrated by the solid edges in the figure) is given by

$$E(T_0) \;=\; \{(x_i, z_i) \mid i \in \{1, \ldots, d\}\} \cup \bigcup_{j=1}^{d} E(P_j) \setminus \{(p^j_{\ell_j}, p^j_{\ell_j - 1})\},$$

where $\ell_j = t_j - (d - j)$ for every $j \in \{1, \ldots, d\}$. For an illustration of the construction, see Fig. 1. We now show that every FT-BFS tree $T' \in \mathcal{T}(s, G)$ must contain all the edges of B,

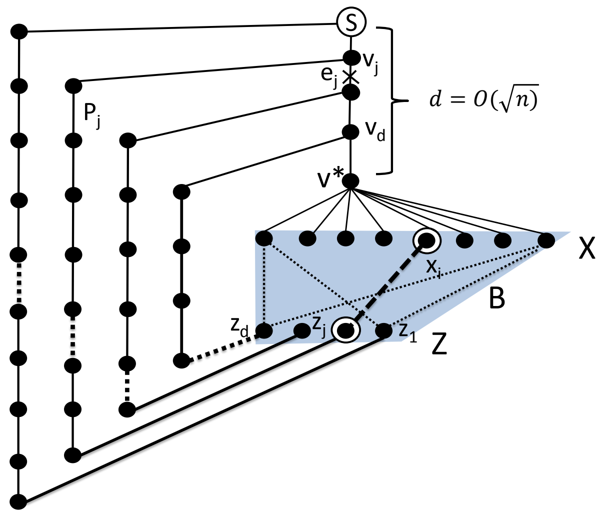

Figure 1: Lower bound construction for FT-BFS. The original BFS tree consists of the non-dashed edges. The dashed edges are the ones necessary to make it an FT-BFS tree. For example, the bold dashed edge (x_j, z_i) is required upon failure of the edge e_i.

namely, the edges $e_{i,j} = (x_i, z_j)$ for every $i \in \{1, \ldots, |X|\}$ and $j \in \{1, \ldots, d\}$ (the dashed edges in the figure). Assume, towards contradiction, that there exists a $T' \in \mathcal{T}(s, G)$ that does not contain $e_{i,j}$ (the bold dashed edge (x_i, z_j) in the figure). Note that upon the failure of the edge $e_j = (v_j, v_{j+1}) \in \pi$, the unique $s - x_i$ shortest path connecting s and x_i in $G \setminus \{e_j\}$ is $P'_j = \pi[v_1, v_j] \circ P_j \circ [z_j, x_i]$, and all other alternatives are strictly longer. Since $e_{i,j} \notin T'$, also $P'_j \nsubseteq T'$, and therefore $\mathrm{dist}(s, x_i, G \setminus \{e_j\}) < \mathrm{dist}(s, x_i, T' \setminus \{e_j\})$, in contradiction to the fact that T' is an FT-BFS tree. It follows that every FT-BFS tree T' must contain at least $|\hat{E}| = \Omega(n^{3/2})$ edges. The theorem follows. ∎

3.2 Multiple Sources

We next consider an intermediate setting where it is necessary to construct a fault-tolerant subgraph FT-MBFS containing several FT-BFS trees in parallel, one for each source $s \in S$, for some $S \subseteq V$. We establish the following.

Theorem 3.2 *There exists an n-vertex graph $G(V, E)$ and a source set $S \subseteq V$ of cardinality σ, such that any* FT-MBFS *tree from the source set S has $\Omega(\sqrt{\sigma} \cdot n^{3/2})$ edges, i.e.,* $\mathrm{Cost}^*(S, G) = \Omega(\sqrt{\sigma} \cdot n^{3/2})$.

Our construction is based on the graph $G(d) = (V_1, E_1)$, which consists of three components: (1) a set of vertices $U = \{u_1, \ldots, u_d\}$ connected by a path $P_1 = [u_1, \ldots, u_d]$, (2) a set of terminal vertices $Z = \{z_1, \ldots, z_d\}$ (viewed by convention as ordered from left to right), and (3) a collection of d vertex disjoint paths Q_i of length $|Q_i| = 6 + 2(d - i)$ connecting u_i and z_i for every $i \in \{1, \ldots, d\}$. The vertex $\mathtt{r}(G(d)) = u_d$ is fixed as the root of $G(d)$, hence the edges of the paths Q_i are viewed as directed away from u_i, and the terminal vertices of Z are viewed as the *leaves* of the graph, denoted $\mathtt{Leaf}(G(d)) = Z$. See Fig. 2 for illustration.

Overall, the vertex and edge sets of $G(d)$ are $V_1 = U \cup Z \cup \bigcup_{i=1}^{d} V(Q_i)$ and $E_1 = E(P_1) \cup \bigcup_{i=1}^{d} E(Q_i)$. Observe the following.

Observation 3.3 *(a)* $\mathtt{nLeaf}(G(d)) = d$.

(b) $|V_1| = c \cdot d^2$ *for some constant* c.

We are now ready to complete the proof of Thm. 3.2.

Proof: Take σ copies, $G'_1, \ldots, G'_\sigma$, of $G(d)$, where $d = O((n/\sigma)^{1/2})$. Note that Obs. 3.3, each copy G'_i consists of $O(n/\sigma)$ nodes. Let y_i be the node u_d and $s_i = \mathtt{r}(G'_i)$ in the ith copy G'_i. Add a node v^* connected to a set X of $\Omega(n)$ nodes and connect v^* to each of the nodes y_i, for $i \in \{1, \ldots, d\}$. Finally, connect the set X to the σ leaf sets $\mathtt{Leaf}(G'_1), \ldots, \mathtt{Leaf}(G'_\sigma)$ by a complete bipartite graph, adjusting the size of the set X in the construction so that $|V(G)| = n$. Since $\mathtt{nLeaf}(G'_i) = \Omega((n/\sigma)^{1/2})$ (see Obs. 3.3), overall $|E(G)| = \Omega(n \cdot \sigma \cdot \mathtt{nLeaf}(G_1(d))) = \Omega(n \cdot (\sigma n)^{1/2})$. Since the path from each source s_i to X cannot aid the nodes of G'_j for $j \neq i$, the analysis of the single-source case can be applied to show that each of the bipartite graph edges in necessary upon a certain edge fault. See Fig. 3 for an illustration. ∎

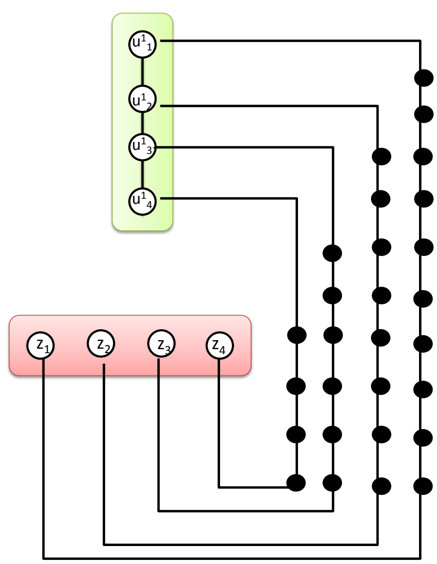

Figure 2: The graph $G_1(d)$.

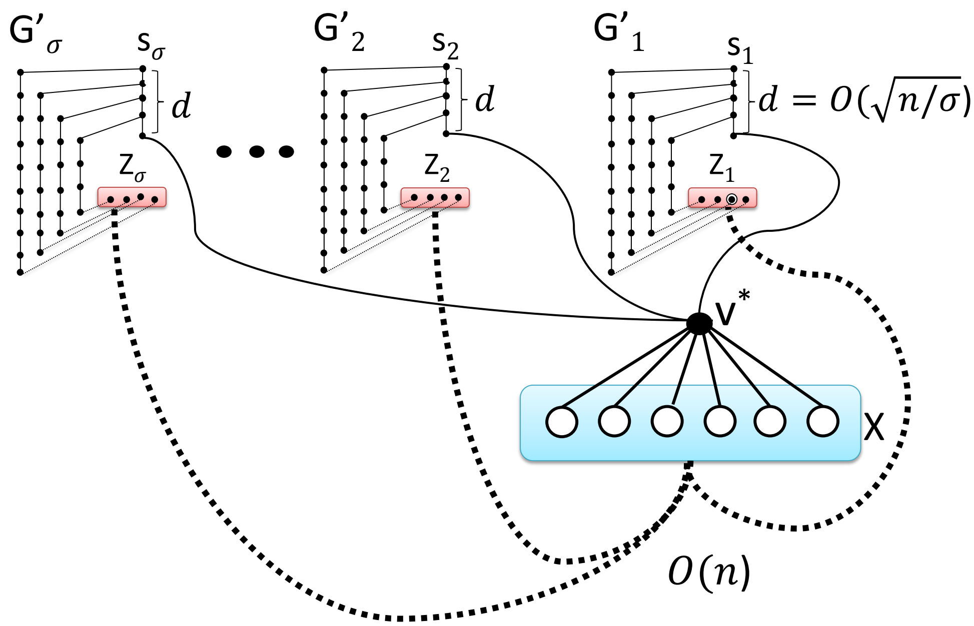

Figure 3: Illustration of the lower bound for the multi-source case.

4 Upper Bounds

In this section we provide tight matching upper bounds to the lower bounds presented in Sec. 3.

4.1 Single Source

For the case of FT-BFS trees, we establish the following.

Theorem 4.1 *There exists a polynomial time algorithm that for every n-vertex graph G and source node s constructs an* FT-BFS *tree rooted at s with $O(n \cdot \min\{\texttt{Depth}(s), \sqrt{n}\})$ edges.*

To prove the theorem, we first describe a simple algorithm for the problem and then prove its correctness and analyze the size of the resulting FT-BFS tree. Using the sparsity lemma of [26] and the tools of [15], one can provide a randomized construction for an FT-BFS tree with $O(n^{3/2} \log n)$ edges with high probability. In contrast, the simple algorithm presented here is *deterministic* and achieves an FT-BFS tree with $O(n^{3/2})$ edges, matching exactly the lower bound established in Sec. 3. We note that known time-efficient (and rather involved) algorithms for constructing replacement paths and distance sensitivity oracles (cf., [16, 26, 5, 30, 15]) can be modified to construct sparse FT-BFS and FT-MBFS trees by breaking shortest path ties properly and maintaining the successors of the computed replacement paths. Since our focus here is on the size of the resulting FT-BFS trees, and not on optimizing the running time, we introduce the construction using a simple but slow ($O(nm + n^2 \log n)$ round) algorithm. In the analysis section we then show that as long as the collection of the single-source replacement paths are computed in a way that breaks shortest path ties properly, the total number of edges in this collection is bounded by $O(n^{3/2})$.

The Algorithm. Recall that W is a weight assignment that guarantees the uniqueness of the shortest paths, by introducing some symbolic perturbation to the edge lengths. Let $T_0 = BFS(s, G)$ be the BFS tree rooted at s in G, computed according to the weight assignment W. For every $e_j \in T_0$, let $T_0(e_j)$ be the BFS tree rooted at s in $G \setminus \{e_j\}$. Then the final FT-BFS tree is given by

$$T^*(s) \;=\; T_0 \cup \bigcup_{e_j \in T_0} T_0(e_j).$$

The correctness is immediate by construction.

Observation 4.2 *$T^*(s)$ is an* FT-BFS *tree.*

Proof: Consider a vertex v and an edge e. If $e \notin \pi(s, v)$, then $\pi(s, v) \subseteq T^*(s) \setminus \{e\}$, hence $\text{dist}(s, v, T^*(s) \setminus \{e\}) = |\pi(s, v)| = \text{dist}(s, v, G \setminus \{e\})$. Otherwise, $e \in \pi(s, v) \subseteq T_0$. Then by construction, $T_0(e) \subseteq T^*(s)$. By definition, $\text{dist}(s, v, T^*(s) \setminus \{e\}) = \text{dist}(s, v, T_0(e)) = \text{dist}(s, v, G \setminus \{e\})$. The observation follows. ∎

Each of the $n - 1$ BFS trees $T_0(e_j)$ can be constructed in $\widetilde{O}(m)$ time, hence $\widetilde{O}(nm)$ rounds are required in total. It therefore remains to bound the size of $T^*(s)$.

Size Analysis. We first provide some notation. For a path P, let $\texttt{Cost}(P) = \sum_{e \in P} W(e)$ be the weighted cost of P, i.e., the sum of its edge weights. An edge $e \in G$ is defined as *new* if $e \notin E(T_0)$. For every $v_i \in V$ and $e_j \in T_0$, let $P^*_{i,j} = \pi(s, v_i, T_0(e_j)) \in SP(s, v_i, G \setminus \{e_j\}, W)$ be the optimal *replacement path* of s and v_i upon the failure of $e_j \in T_0$. Let $\texttt{New}(P) = E(P) \setminus E(T_0)$ and

$$\texttt{New}(v_i) \;=\; \{\texttt{LastE}(P^*_{i,j}) \mid e_j \in T_0\} \setminus E(T_0)$$

be the set of v_i new edges appearing as the last edge in the replacement paths $P^*_{i,j}$ of v_i and $e_j \in T_0$. It is convenient to view the edges of $T_0(e_j)$ as directed away from s. We then have that

$$T^*(s) = T_0 \cup \bigcup_{v_i \in V \setminus \{s\}} \texttt{New}(v_i).$$

I.e., the set of new edges that participate in the final FT-BFS tree $T^*(s)$ are those that appear as a last edge in some replacement path.

We now upper bound the size of the FT-BFS tree $T^*(s)$. Our goal is to prove that $\texttt{New}(v_i)$ contains at most $O(\sqrt{n})$ edges for every $v_i \in V$. The following observation is crucial in this context.

Observation 4.3 *If $\texttt{LastE}(P^*_{i,j}) \notin E(T_0)$, then $e_j \in \pi(s, v_i)$.*

Proof: Assume, towards contradiction, that $e_j \notin \pi(s, v_i)$ and let $P^*_{i,j} \subseteq T_0(e_j)$ be the $s - v_i$ replacement path in $G \setminus \{e_j\}$ according to the weight assignment W. Since $\texttt{LastE}(P^*_{i,j}) \notin E(T_0)$, we have two different $s - v_i$ shortest paths in $G \setminus \{e_j\}$, namely, $\pi(s, v_i)$ and $P^*_{i,j}$. By the optimality of $\pi(s, v_i)$ in G, i.e., $\pi(s, v_i) \in SP(s, v_i, G, W)$, it holds that $\texttt{Cost}(\pi(s, u)) < \texttt{Cost}(P^*_{i,j})$. On the other hand, by the optimality of $P^*_{i,j}$ in $G \setminus \{e_j\}$, i.e., $P^*_{i,j} \in SP(s, v_i, G \setminus \{e_j\}, W)$, we have that $\texttt{Cost}(\pi(s, u)) > \texttt{Cost}(P^*_{i,j})$. Contradiction. ∎

Obs. 4.3 also yields the following.

Corollary 4.4 *(1)* $\texttt{New}(v_i) = \{\texttt{LastE}(P^*_{i,j}) \mid e_j \in \pi(s, v_i)\} \setminus E(T_0)$ *and*
(2) $|\texttt{New}(v_i)| \leq \min\{\texttt{depth}(v_i), \texttt{deg}(v_i)\}$.

This holds since the edges of $\texttt{New}(v_i)$ are coming from at most $\texttt{depth}(v_i)$ replacement paths $P^*_{i,j}$ (one for every $e_j \in \pi(s, v_i)$), and each such path contributes at most one edge incident to v_i.

For the reminder of the analysis, let us focus on one specific node $u = v_i$ and let $\pi = \pi(s, u)$, $N = |\texttt{New}(u)|$. For every edge $e_k \in \texttt{New}(u)$, we define the following parameters. Let $f(e_k) \in \pi$ be the failed edge such that $e_k \in T_0(f(e_k))$ appears in the replacement path $P_k = \pi(s, u, T')$ for $T' = T_0(f(e_k))$. (Note that e_k might appear as the last edge on the path $\pi(s, u, T_0(e'))$ for several edges $e' \in \pi$; in this case, one such e' is chosen arbitrarily).

Let b_k be the *last* divergence point of P_k and π, i.e., the last vertex on the replacement path P_k that belongs to $V(\pi) \setminus \{u\}$. Since $\texttt{LastE}(P_k) \notin E(T_0)$, it holds that b_k is not the neighbor of u in P_k.

Let $\texttt{New}(u) = \{e_1, \ldots, e_N\}$ be sorted in non-decreasing order of the distance between b_k and u, $\text{dist}(b_k, u, \pi) = |\pi(b_k, u)|$. I.e.,

$$\text{dist}(b_1, u, \pi) \leq \text{dist}(b_2, u, \pi) \ldots \leq \text{dist}(b_N, u, \pi). \tag{1}$$

We consider the set of truncated paths $P'_k = P_k[b_k, u]$ and show that these paths are vertex-disjoint except for the last common endpoint u. We then use this fact to bound the number of these paths, hence bound the number N of new edges. The following observation follows immediately by the definition of b_k.

Observation 4.5 $(V(P'_k) \cap V(\pi)) \setminus \{b_k, u\} = \emptyset$.

Lemma 4.6 $\Big(V(P'_i) \cap V(P'_j)\Big) \setminus \{u\} = \emptyset$ *for every* $i, j \in \{1, \ldots, N\}$, $i \neq j$.

Proof: Assume towards contradiction that there exist $i \neq j$, and a node

$$u' \in \big(V(P'_i) \cap V(P'_j)\big) \setminus \{u\}$$

in the intersection. Since $\texttt{LastE}(P'_i) \neq \texttt{LastE}(P'_j)$, by Obs. 4.5 we have that $P'_i, P'_j \subseteq G \setminus E(\pi)$. The faulty edges $f(e_i), f(e_j)$ belong to $E(\pi)$. Hence there are two distinct $u' - u$ shortest paths in $G \setminus \{f(e_i), f(e_j)\}$. By the optimality of P'_i in $T_0(f(e_i))$, (i.e., $P_i \in SP(s, u, G \setminus \{f(e_i)\}, W)$), we have that $\texttt{Cost}(P'_i[u', u]) < \texttt{Cost}(P'_j[u', u])$. In addition, by the optimality of P'_j in $T_0(f(e_j))$, (i.e., $P_j \in SP(s, u, G \setminus \{f(e_j)\}, W)$), we have that $\texttt{Cost}(P'_j[u', u]) < \texttt{Cost}(P'_i[u', u])$. Contradiction. ∎

We are now ready to prove our key lemma.

Lemma 4.7 $|\texttt{New}(u)| = O(n^{1/2})$ *for every* $u \in V$.

Proof: Assume towards contradiction that $N = |\texttt{New}(u)| > \sqrt{2n}$. By Lemma 4.6, we have that $b_1, \ldots, b_N$ are distinct and by definition they all appear on the path π. Therefore, by the ordering of the P'_k, we have that the inequalities of Eq. (1) are strict, i.e., $\text{dist}(b_1, u, \pi) < \text{dist}(b_2, u, \pi) < \ldots < \text{dist}(b_N, u, \pi)$. Since $b_1 \neq u$ (by definition), we also have that $\text{dist}(b_1, u, \pi) \geq 1$. We Conclude that

$$\text{dist}(b_k, u, \pi) = |\pi(b_k, u)| \geq k \ . \tag{2}$$

Next, note that each P'_k is a replacement $b_k - u$ path and hence it cannot be shorter than $\pi(b_k, u)$, implying that $|P'_k| \geq |\pi(b_k, u)|$. Combining with Eq. (2), we have that

$$|P'_k| \geq k \ \text{ for every } \ k \in \{1, \ldots, N\} \ . \tag{3}$$

Since by Lemma 4.6, the paths P'_k are vertex disjoint (except for the common vertex u), we have that

$$\left| \bigcup_{k=1}^{N} (V(P'_k) \setminus \{u\}) \right| \ = \ \sum_{k=1}^{N} |V(P'_k) \setminus \{u\}| \ \geq \ \sum_{k=1}^{N} (k-1) \ > \ n,$$

where the first inequality follows by Eq. (3) and the last by the assumption that $N > \sqrt{2n}$. Since there are n nodes in G, we end with contradiction. ▮

Turning to the case of a single vertex failure, the entire proof goes through almost without change, yielding the following.

Theorem 4.8 *There exists a polynomial time algorithm that for every n-vertex graph and source node s constructs an* FT-BFS *tree from s tolerant to one vertex failure, with* $O(n \cdot \min\{\texttt{Depth}(s), \sqrt{n}\})$ *edges.*

4.2 Multiple Sources

For the case of multiple sources, we establish the following upper bound.

Theorem 4.9 *There exists a polynomial time algorithm that for every n-vertex graph $G = (V, E)$ and source set $S \subseteq V$ of size $|S| = \sigma$ constructs an* FT-MBFS *tree $T^*(S)$ from each source $s_i \in S$, tolerant to one edge or vertex failure, with a total number of* $n \cdot \min\{\sum_{s_i \in S} \texttt{depth}(s_i), O(\sqrt{\sigma n})\}$ *edges.*

The algorithm. As in the single source case, to avoid complications due to shortest paths of the same length, all shortest path distances in G are computed using a weight function W defined so as to ensure the uniqueness of a single $u - v$ shortest-path. For every $s_i \in S$ and every $e_j \in T_0(s_i)$, let $T(s_i, e_j)$ be the BFS tree rooted at s_i in $G \setminus \{e_j\}$. Let

$$T_0(S) = \bigcup_{s_i \in S} T_0(s_i)$$

be the joint structure containing all the BFS trees of S. Then by the previous section, the FT-BFS tree for s_i is $T^*(s_i) = T_0 \cup \bigcup_{e_j \in T_0(s_i)} T(s_i, e_j)$. Define the FT-MBFS for S as

$$T^*(S) = \bigcup_{s_i \in S} T^*(s_i) = \bigcup_{s_i \in S, e_j \in T_0(s_i)} T(s_i, e_j).$$

Analysis. The correctness follows immediately by the single source case. It remains to bound the number of edges of $T^*(S)$. An edge e is *new* if $e \notin T_0(S)$. For every $v_i \in V$, define its new edge set in the graph $T^*(S)$ by

$$\texttt{New}(S, v_i) = \{\texttt{LastE}(\pi(s, v_i, T(s_i, e_j))) \mid s_i \in S, e_j \in T_0(s_i)\} \setminus E(T_0(S)).$$

To bound the size of $T^*(S)$, we focus on node $u = v_i$, and bound its new edges $\texttt{New}(S, u) = \{e_1, \ldots, e_N\}$. Obs. 4.3 yields the following.

Corollary 4.10 $\texttt{New}(S, u) \leq \sum_{s_i \in S} \texttt{depth}(s_i)$.

Towards the end of this section, we prove that $\texttt{New}(S, u)$ contains at most $O(\sqrt{\sigma n})$ new edges. For ease of notation, let $\pi(s_i) = \pi(s_i, u)$ for every $i \in \{1, \ldots, \sigma\}$. For every edge $e_k \in \texttt{New}(S, u)$, we define the following parameters. Let $s(e_k) \in S$ and $f(e_k) \in T_0(s(e_k))$ be such that $e_k \in T(s(e_k), f(e_k))$. I.e., the edge e_k appears in the replacement $s(e_k) - u$ path $P_k = \pi(s, u, T')$, where $T' = T(s(e_k), f(e_k))$ is the BFS tree rooted at $s(e_k)$ in $G \setminus \{f(e_k)\}$. By Obs. 4.3, $f(e_k) \in \pi(s(e_k))$. (Note that for a given new edge e_k there might be several s' and e' such that $e_k = \texttt{LastE}(\pi(s', u, T(s', e')))$; in this case one such pair s', e' is chosen arbitrarily.) For every replacement path P_k (whose last edge is e_k), denote by b_k the *last* divergence point of P_k and the collection of shortest $s_i - u$ paths $\mathcal{P} = \bigcup_{s_i \in S} \pi(s_i, u) \setminus \{u\}$. I.e., b_k is the last point on P_k that belongs to $V(\mathcal{P}) \setminus \{u\}$. Let $P'_k = P_k[b_k, u]$ be the truncated path from the divergence point b_k to u. Note that since $e = (x, u) = \texttt{LastE}(P_k) \notin E(T_0(S))$ is a new edge, it holds that $x \notin V(\mathcal{P}) \setminus \{u\}$ and b_k is in $V \setminus \{u\}$. The following observation is useful.

Observation 4.11 $P'_k \subseteq G \setminus E(\mathcal{P})$ *for every* $k \in \{1, \ldots, N\}$.

We now show that the paths P'_k are vertex disjoint except for their endpoint u (this is regardless of their respective source $s(e_k)$).

Lemma 4.12 $\left(V(P'_i) \cap V(P'_j)\right) \setminus \{u\} = \emptyset$ *for every* $i \neq j \in \{1, \ldots, N\}$.

Proof: Assume towards contradiction that there exists $i \neq j$, and a node

$$u' \in \left(V(P'_i) \cap V(P'_j)\right) \setminus \{u\}$$

in the intersection. Since $\texttt{LastE}(P'_i) \neq \texttt{LastE}(P'_j)$ and by Obs. 4.11, $P'_i, P'_j \subseteq G \setminus E(\mathcal{P})$, and the faulty edges $f(e_i), f(e_j) \in \mathcal{P}$, we have two distinct $u' - u$ replacement paths in $G \setminus \{f(e_i), f(e_j)\}$. By the optimality of P'_i in $T(s(e_i), f(e_i))$, (i.e., $P_i \in SP(s(e_i), u, G \setminus \{f(e_i)\}, W)$), we have that $\texttt{Cost}(P'_i) < \texttt{Cost}(P'_j)$. Similarly, by the optimality of P'_j in $T(s(e_j), f(e_j))$, (i.e., $P_j \in SP(s(e_j), u, G \setminus \{f(e_j)\}, W)$), we have that $\texttt{Cost}(P'_j) < \texttt{Cost}(P'_i)$, contradiction. The lemma follows. ∎

We are now ready to state and prove our main lemma.

Lemma 4.13 $N = |\texttt{New}(S,u)| = O(\sqrt{\sigma n})$.

We begin by classifying the set of new edges $e_i \in \texttt{New}(S,u)$ into σ classes according to the position of the divergence point b_i. For every $e_i \in \texttt{New}(S,u)$, let $\widehat{s}(e_i) \in S$ be some source node such that the divergence point $b_i \in \pi(\widehat{s}(e_i),u)$ appears on its $\widehat{s}(e_i)-u$ shortest path in $T_0(\widehat{s}(e_i))$. If there are several such sources for the edge e_i, one is chosen arbitrarily.

For every $s_j \in S$, let

$$\texttt{New}(s_j) = \{e_i \in \texttt{New}(S,u) \mid \widehat{s}(e_i) = s_j\}$$

be the set of new edges in $\texttt{New}(S,u)$ that are mapped to $s_j \in S$. Then, $\texttt{New}(S,u) = \bigcup_{s_j \in S} \texttt{New}(s_j)$. Let $x_j = |\texttt{New}(s_j)|$.

We now focus on s_j. For every $e_{j_k} \in \texttt{New}(s_j)$, $k = \{1,\ldots,x_j\}$, let $P_{j_k} = \pi(s(e_{j_k}),u,T')$ for $T' = T(s(e_{j_k}), f(e_{j_k}))$ be the replacement path such that $\texttt{LastE}(P_{j_k}) = e_{j_k}$ and b_{j_k} be its corresponding (last) divergence point with $\pi(s_j,u)$ ($s_j = \widehat{s}(e_{j_k})$). In addition, the truncated path is given by $P'_{j_k} = P_{j_k}[b_{j_k},u]$. Note that $\texttt{LastE}(P_{j_k}) = e_{j_k}$.

Consider the set of divergence points $b_{j_1},\ldots,b_{j_{x_j}}$ sorted in non-decreasing order of the distance between b_{j_k} and s_j on the shortest $s_j - u$ path $\pi(s_j)$ i.e., $|\pi(b_{j_k},u,T_0(s_j))|$, where

$$|\pi(b_{j_1},u,T_0(s_j))| \le |\pi(b_{j_2},u,T_0(s_j))| \ldots \le |\pi(b_{j_{x_j}},u,T_0(s_j))| \ . \tag{4}$$

Note that by Lemma 4.12, $b_{j_\ell} \neq b_{j_{\ell'}}$ for every $\ell,\ell' \in \{1,\ldots,x_j\}$. In addition, since each $b_{j_\ell} \neq u$, $|\pi(b_{j_1},u,T_0(s_j))| \ge 1$. Hence, since $b_{j_1},\ldots,b_{j_{x_j}} \in \pi(s_j)$, combining with Eq. (4) we get that

$$1 \le |\pi(b_{j_1},u,T_0(s_j))| < |\pi(b_{j_2},u,T_0(s_j))| \ldots < |\pi(b_{j_{x_j}},u,T_0(s_j))| \ . \tag{5}$$

Since P'_{j_ℓ} is an alternative $b_{j_\ell} - u$ replacement path, we have that

$$|P'_{j_\ell}| \ge |\pi(b_{j_\ell},u,T_0(s_j))| \ge \ell. \tag{6}$$

where the last inequality follows by Eq. (4). Hence, since all P'_{j_ℓ} are vertex disjoint, except for the last node u, we get the total number of nodes $V(s_j) = \bigcup V(P'_{j_\ell}) \setminus \{u\}$ occupied by P'_{j_ℓ} paths is

$$\sum_{\ell=1}^{x_j} |V(P'_{j_\ell})| = |V(s_j)| = O(x_j^2).$$

Since the nodes of $V(s_{j_1})$ and $V(s_{j_2})$ are disjoint for every $s_{j_1}, s_{j_2} \in S$, by Lemma 4.12, it follows that $|\texttt{New}(S,u)| = \sum_{j=1}^{\sigma} x_j$ but $\sum_{j=1}^{\sigma} |V(s_j)| = O(x_j^2) \le n$. Therefore, $|\texttt{New}(S,u)| = \sum_{j=1}^{\sigma} x_j \le O(\sqrt{\sigma n})$. ∎

As there are n nodes, combining with Cor. 4.10, we get that the total number of edges in $T^*(S)$ is given by

$$E(T^*(S)) \le |E(T_0(S))| + \sum_{u \in V} |\texttt{New}(S,u)| \le \sigma n + n \cdot \min\{\sum_{s_i \in S} \texttt{depth}(s_i), O(\sqrt{\sigma n})\},$$

as required. Thm. 4.9 is established. The analysis for the case of vertex faults follows with almost no changes. ▮

We next note that our lower bounds extend trivially to the case of directed and edge weighted graphs. Moreover, the analysis of our upper bounds also extends naturally to the case of directed and edge weighted graphs with integer weights in the range $[1, M]$, paying an extra factor of $O(\sqrt{M})$ in the size of the resulting FT-MBFS trees[1].

5 The Minimum FT-MBFS Problem

In this section we consider the optimization formulation for our FT-MBFS structures, namely, the Minimum FT-MBFS problem. In Sec. 4, we presented an algorithm that for every graph G and source s constructs an FT-BFS tree $\widehat{T} \in \mathcal{T}(s, G)$ with $O(n^{3/2})$ edges. In Sec. 3, we showed that there exist graphs G and $s \in V(G)$ for which $\texttt{Cost}^*(s, G) = \Omega(n^{3/2})$, establishing tightness of our algorithm in the worst-case. Yet, there are also inputs (G', s') for which the algorithm of Sec. 4, as well as algorithms based on the analysis of [15] and [26], might still produce an FT-BFS $\widehat{T} \in \mathcal{T}(s', G')$ which is denser by a factor of $\Omega(\sqrt{n})$ than the size of the optimal FT-BFS tree, i.e., such that $|E(\widehat{T})| \geq \Omega(\sqrt{n}) \cdot \texttt{Cost}^*(s', G')$. For an illustration of such a case consider the graph $G' = (V, E)$ which is a modification of the graph G described in Sec. 3. The modifications are as follows. First, add a node z_0 to Z and connect it to every $x_i \in X$. Replace the last edge $e'_i = \texttt{LastE}(P_i)$ of the $v_i - z_i$ path P_i by a vertex r_i that is connected to the endpoints of the edge e'_i for every $i \in \{1, \ldots, d\}$. Let P'_i be the $s - z_i$ modified path where $\texttt{LastE}(P'_i) = (r_i, z_i)$. Finally, connect the node z_0 to all nodes r_i for every $i \in \{1, \ldots, d\}$. See Fig. 4 for illustration. Observe that whereas $\texttt{Cost}^*(s, G) = \Omega(n^{3/2})$, the modified G' has $\texttt{Cost}^*(s, G') = O(n)$, as the edges of the complete bipartite graph B that are required in every $\widehat{T} \in \mathcal{T}(s, G)$ are no longer required in every $T' \in \mathcal{T}(s, G')$; it is sufficient to connect the nodes of X to z_0 only, and by that "save" the $\Omega(n^{3/2})$ edges of B in T'. Nevertheless, as we show next, for certain weight assignments the algorithm of Sec. 4 constructs an FT-BFS tree $\widehat{T}$ of size $O(n^{3/2})$. Specifically, let W be such that each of the edges of

$$E' = \{(z_0, r_i) \ \mid \ i \in \{1, \ldots, d\}\} \cup \{(z_0, x_i) \ \mid \ x_i \in X\}$$

is assigned a weight which is strictly larger than the weights of the other edges. That is, $W(e_k) > W(e_\ell)$ for every $e_k \in E'$ and $e_\ell \in E(G') \setminus E'$. Note that for every edge $e_i = (v_i, v_{i+1}) \in \pi$, $i \in \{1, \ldots, d\}$, there are two alternative $s - x_j$ replacement paths of the same length, namely, $Q_{i,j} = \pi[s, v_i] \circ P'_i \circ (z_i, x_j)$ that goes through z_i and $\widehat{Q}_{i,j} = \pi[s, v_i] \circ P'_i[s, r_i] \circ (r_i, z_0) \circ (z_0, x_i)$ that goes through z_0. Although $|Q_{i,j}| = |\widehat{Q}_{i,j}|$, the weight assignment implies that $\texttt{Cost}(Q_{i,j}) < \texttt{Cost}(\widehat{Q}_{i,j})$ and hence $\widehat{Q}_{i,j} \notin SP(s, x_j, G \setminus \{e_i\}, W)$ for every $i \in \{1, \ldots, d\}$ and every $x_j \in X$. Therefore, $E(B) \subseteq \widehat{T}$, for every FT-BFS tree $\widehat{T}$ computed by the algorithm of Sec. 4 with the weight assignment W. Hence $|E(\widehat{T})| = \Theta(n^{3/2})$ while $\texttt{Cost}^*(s, G') = O(n)$.

We begin by showing that the minimum FT-BFS problem is NP-hard and moreover, cannot be approximated (under standard complexity assumptions) to within a factor of $\Omega(\log n)$ where

[1]Negative weights can be handled as well, albeit using a more complicated algorithm, not described herein.

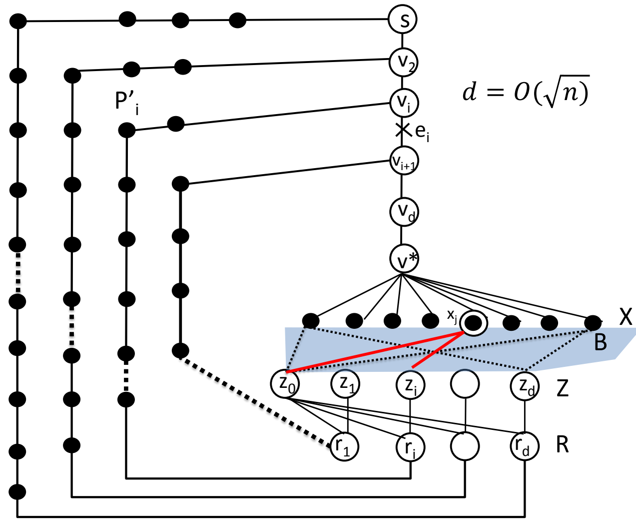

Figure 4: Bad example for the algorithm of Sec. 4. The weights of the z_0 edges are larger than those of the other edges. Thus, the entire complete bipartite graph $B(X, Z \setminus \{z_0\})$ of size $\Omega(n^{3/2})$ is included in the resulting FT-BFS tree $\widehat{T} \in \mathcal{T}(s, G)$ returned by the algorithm. However, an FT-BFS tree T^* of $O(n)$ edges can be given by including the edges of (z_0, x_i) for every $x_i \in X$. The red edges are two optional edges necessary upon the failure of e_i. Adding the edge (x_j, z_0) is better, yet the algorithm of Sec. 4 adds (x_j, z_i) to $\widehat{T}$ for every $x_j \in X$.

n is the number of vertices of the input graph G. Clearly, a universally optimal algorithm is unlikely given the hardness of approximation result of Subsec. 5.1. Yet in Subsec. 5.2, we show that the gap can be narrowed down and provide a $O(\log n)$ approximation for this problem.

5.1 Hardness of Approximation of the Minimum FT-BFS Problem

In this section we establish the following.

Theorem 5.1 *The Minimum* FT-BFS *problem is NP-complete and cannot be approximated to within a factor $c \log n$ for some constant $c > 0$ unless $\mathcal{NP} \subseteq \mathcal{TIME}(n^{poly \log(n)})$.*

We prove Theorem 5.1 by showing a gap preserving reduction from the Set-Cover problem to the Minimum FT-BFS problem. An instance $\langle U, \mathfrak{F} \rangle$ of the Set-Cover problem consists of a set of N elements $U = \{u_1, \ldots, u_N\}$ and a collection of M sets $\mathfrak{F} = \{S_1, \ldots, S_M\}$ such that $S_i \subseteq U$ and $\bigcup S_i = U$. The task is to choose the minimal number of sets in $\mathfrak{F}$ whose union covers all of U. Fiege [14] showed that the Set Cover problem cannot be approximated within a ratio of $(1 - o(1)) \ln n$ unless $\mathcal{NP} \subseteq \mathcal{TIME}(M^{poly \log(M)})$.

The Transformation. Given a Set-Cover instance $\langle U, \mathfrak{F} \rangle$, we construct a Minimum FT-BFS instance $\mathcal{I}(U, \mathfrak{F}) = (G, s)$ as follows. Let $X = \{x_1, \ldots, x_M\}$ (resp., $Z = \{z_1, \ldots, z_N\}$) be the vertex set corresponding to the collection of sets $\mathfrak{F}$ (resp., elements of U). Let $B_{XZ} = (X, Z, E_{XZ})$

be the bipartite graph corresponding to the input $\langle U, \mathfrak{F} \rangle$, where $E_{XZ} = \{(x_j, z_i) \mid u_i \in S_j,\ j \in \{1, \ldots, M\}$ and $i \in \{1, \ldots, N\}\}$. Embed the bipartite graph B_{XZ} in G in the following manner. Construct a length-$(N+1)$ path $P = [s = p_0, p_1 \ldots, p_N, p_{N+1}]$, connect a vertex v' to p_N and connect a set of vertices $Y = \{y_1, \ldots, y_R\}$ for $R = O((MN)^3)$ to the vertex p_{N+1} by the edges of $E_{pY} = \{(p_{N+1}, y_i) \mid i \in \{1, \ldots, R\}\}$. Connect these vertices to the bipartite graph B_{XZ} as follows. For every $i \in \{1, \ldots, N\}$, connect the node p_{i-1} of P to the node z_i of Z by a path $Q_i = [p_{i-1} = q_0^i, \ldots, q_{t_i}^i = z_i]$ where $t_i = |Q_i| = 6 + 2(N - i)$. Thus the paths Q_i are monotonely decreasing and vertex disjoint. In addition, connect the vertices v' and p_{N+1} to every vertex of X, adding the edge sets $E_{vX} = \{(v', x_i) \mid x_i \in X\}$ and $E_{pX} = \{(p_{N+1}, x_j) \mid x_j \in X\}$. Finally, construct a complete bipartite graph $B_{XY} = (X, Y, E_{XY})$ where $E_{XY} = \{(y_\ell, x_j) \mid x_j \in X, y_\ell \in Y\}$. This completes the description of G. For illustration, see Fig. 5. Overall,

$$V(G) = X \cup Z \cup V(P) \cup \bigcup_{i=1}^{N} V(Q_i) \cup \{v'\} \cup Y,$$

and

$$E(G) = E_{XZ} \cup E(P) \cup \bigcup_{i=1}^{N} E(Q_i) \cup \{(p_N, v')\} \cup E_{pY} \cup E_{vX} \cup E_{pX} \cup E_{XY}.$$

Note that $|V(G)| = O(R)$ and that $|E(G)| = O(|E_{XZ}| + N^2 + MR) = O(MR)$.

First, note the following.

Observation 5.2 *Upon the failure of the edge $e_i = (p_{i-1}, p_i)$, $i \in \{1, \ldots, N\}$, the following happen:*
(a) the unique $s - z_i$ shortest path in $G \setminus \{e_i\}$ is given by $\widetilde{P}_i = P[s, p_{i-1}] \circ Q_i$.
(b) the shortest-paths connecting s and the vertices of $\{p_N, p_{N+1}, v'\} \cup X \cup Y$ disconnect and hence the replacement paths in $G \setminus \{e_i\}$ must go through the Z nodes.

We begin by observing that all edges except those of B_{XY} are necessary in every `FT-BFS` tree $\widehat{T} \in \mathcal{T}(s, G)$. Let $\widetilde{E} = E(G) \setminus E_{XY}$.

Observation 5.3 *$\widetilde{E} \subseteq \widehat{T}$ for every $\widehat{T} \in \mathcal{T}(s, G)$.*

Proof: The edges of the paths P and the edges of $E_{pY} \cup \{(p_N, v')\}$ are trivially part of every `FT-BFS` tree. The edges of the path Q_i are necessary, by Obs. 5.2(a), upon the failure of e_i for every $i \in \{1, \ldots, N\}$. To see that the edges of E_{vX} are necessary, note that upon the failure of the edge (p_N, p_{N+1}) or the edge (p_{N+1}, x_j), the unique $s - x_j$ replacement path goes through v' for every $j \in \{1, \ldots, M\}$. Similarly, the edges E_{pX} are necessary upon the failure of (p_N, v') or (v', x_j).

It remains to consider the edges of E_{XZ}. Assume, towards contradiction, that there exists some $T' \in \mathcal{T}(s, G)$ that does not contain $e_{j,i} = (x_j, z_i) \in E_{XZ}$. Note that by Obs. 5.2(a), upon the failure of the edge $e_i = (p_{i-1}, p_i) \in P$, the unique $s - x_j$ shortest-path in $G \setminus \{e_i\}$ is $P_i' = \pi[p_0, p_{i-1}] \circ Q_i \circ [z_i, x_j]$, and all other alternatives are strictly longer. Since $e_{j,i} \notin T'$, also $P_i' \not\subseteq T'$, and therefore $\text{dist}(s, x_j, G \setminus \{e_i\}) < \text{dist}(s, x_j, T' \setminus \{e_i\})$, in contradiction to the fact that $T' \in \mathcal{T}(s, G)$. The observation follows. ∎

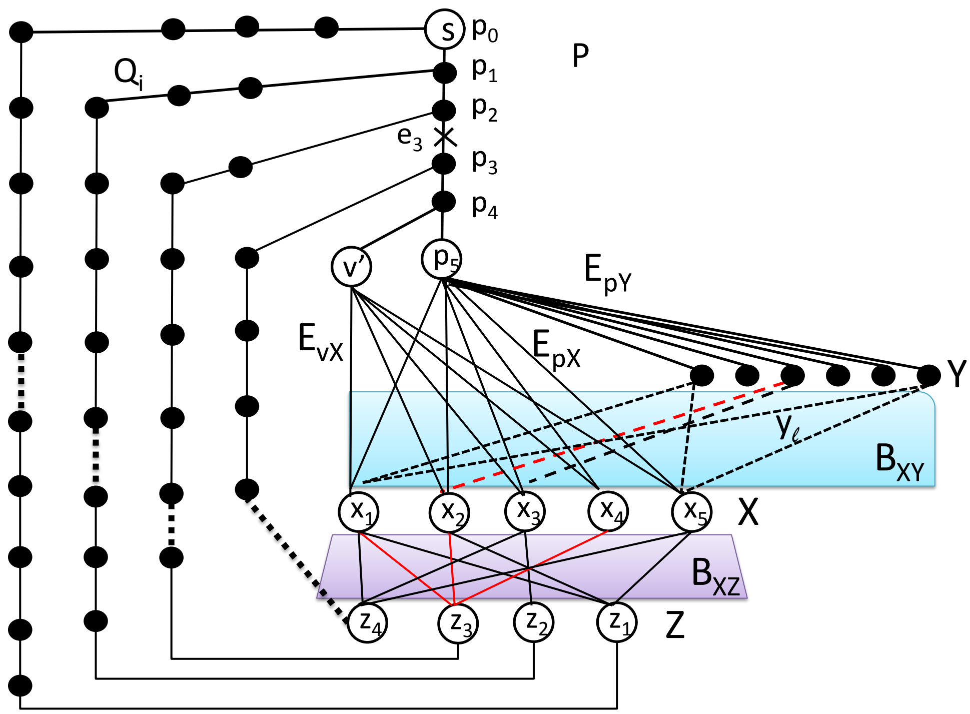

Figure 5: Schematic illustration of the reduction from Set-Cover to Minimum `FT-BFS`. In this example $\mathfrak{F} = \{S_1, S_2, \ldots, S_5\}$ where $S_1 = \{u_1, u_3, u_4\}$, $S_2 = \{u_1, u_3\}$, $S_3 = \{a_2, a_4\}$, $S_4 = \{a_3\}$ and $S_5 = \{a_1, a_4\}$. Thus, $N = 4$ and $M = 5$. The minimal vertex cover is given by S_2 and S_3. The vertex set Y is fully connected to X. In the optimal `FT-BFS` T^*, Y is required to be connected to the x_j nodes that corresponds to the sets appearing in the optimal cover. For example, y_ℓ is connected to x_2 and x_3 which "covers" the Z nodes. The red edges are necessary upon the fault of e_3. All edges described except for the (x_j, y_ℓ) edges are required in any `FT-BFS` tree.

We now prove the correctness of the reduction and then consider gap-preservation. Let $\widehat{T} \in \mathcal{T}(s,G)$ and define by $\Gamma(y_\ell, \widehat{T}) = \{x_j \mid (x_j, y_\ell) \in \widehat{T}\}$ the X nodes that are connected to y_ℓ in $\widehat{T}$, for every $y_\ell \in Y$. Let $\kappa(\widehat{T}) = \min\{|\Gamma(y_\ell, \widehat{T})| \mid y_\ell \in Y\}$. Note that since the edges of $\widetilde{E}$ are necessary in every $\widehat{T} \in \mathcal{T}(s,G)$ it follows that

$$|E(\widehat{T})| \geq |\widetilde{E}| + \kappa(\widehat{T}) \cdot R \ . \tag{7}$$

Lemma 5.4 *If $\widehat{T} \in \mathcal{T}(s,G)$ then there exists a Set-Cover for $\langle U, \mathfrak{F} \rangle$ of size at most $\kappa(\widehat{T})$.*

Proof: Consider $\widehat{T} \in \mathcal{T}(s,G)$ and let $y_\ell \in Y$ be such that $|\Gamma(y_\ell, \widehat{T})| = \kappa(\widehat{T})$. A cover $\mathfrak{F}'$ for U for size $\kappa(\widehat{T})$ is constructed as follows. Let $\mathfrak{F}' = \{S_j \mid x_j \in \Gamma(y_\ell, \widehat{T})\}$. By definition, $|\mathfrak{F}'| = \kappa(\widehat{T})$. We now claim that it is a cover for U. Assume, towards contradiction, that there exists some $u_i \in U$ not covered by $\mathfrak{F}'$. Consider the graph $G' = G \setminus \{e_i\}$ where $e_i = (p_{i-1}, p_i)$. Recall that by Obs. 5.2(a), $\widetilde{P}_k = P[s, p_{k-1}] \circ Q_k$ is the $s - z_k$ path in $G \setminus \{e_k\}$. Note that $\widetilde{P}_k \not\subseteq G'$ for every $k > i$ and $|\widetilde{P}_k| > |\widetilde{P}_i|$ for every $k < i$. Hence denoting the set of neighbors of z_i

in X by $\Gamma(z_i) = \{x_j \mid (z_i, x_j) \in E_{XZ}\}$, by Obs. 5.2(b), the unique $s - x_j$ shortest-path, for every $x_j \in \Gamma(z_i)$ such that $(z_i, x_j) \in E_{XY}$, is given by $P'_j = \widetilde{P}_i \circ (z_i, x_j)$. Therefore the $s - y_\ell$ shortest-paths in G' are all given by $P'_j \circ (x_j, y_\ell)$, for every $x_j \in \Gamma(z_i)$. But since $(x_j, y_\ell) \notin \widehat{T}$ for every $x_j \in \Gamma(z_i)$, we have that $\text{dist}(s, y_\ell, G') < \text{dist}(s, y_\ell, \widehat{T} \setminus \{e_i\})$, in contradiction to the fact that $\widehat{T} \in \mathcal{T}(s, G)$. ∎

Lemma 5.5 *If there exists a Set-Cover of size κ then* $\texttt{Cost}^*(s, G) \leq |\widetilde{E}| + \kappa \cdot R$.

Proof: Given a cover $\mathfrak{F}' \subseteq \mathfrak{F}$, $|\mathfrak{F}'| = \kappa$, construct a `FT-BFS` tree $\widehat{T} \in \mathcal{T}(s, G)$ with $|\widetilde{E}| + \kappa \cdot R$ edges as follows. Add $\widetilde{E}$ to $\widehat{T}$. In addition, for every $S_j \in \mathfrak{F}'$, add the edge (y_ℓ, x_j) to $\widehat{T}$ for every $y_\ell \in Y$. Clearly, $|E(\widehat{T})| = |\widetilde{E}| + \kappa \cdot R$. It remains to show that $\widehat{T} \in \mathcal{T}(s, G)$. Note that there is no $s - u$ replacement path that uses any $y_\ell \in Y$ as a relay, for any $u \in V(G)$ and $y_\ell \in Y$; this holds as X is connected by two alternative shortest-paths to both p_{N+1} and to v' and the path through y_ℓ is strictly longer. In addition, if the edge $e \in \{(p_N, p_{N+1}), (p_{N+1}, y_\ell)\}$ fails, then the $s - y_\ell$ shortest path in $G \setminus \{e\}$ goes through any neighbor x_j of y_ℓ. Since each y_ℓ has at least one X node neighbor in $\widehat{T}$, it holds that $\text{dist}(s, y_\ell, \widehat{T} \setminus \{e\}) = \text{dist}(s, y_\ell, G \setminus \{e\})$.

Since the only missing edges of $\widehat{T}$, namely, $E(G) \setminus E(\widehat{T})$, are the edges of E_{XY}, it follows that it remains to check the edges $e_i = (v_{i-1}, v_i)$ for every $i \in \{1, \ldots, N\}$. Let $S_j \in \mathfrak{F}'$ such that $u_i \in S_j$. Since $\mathfrak{F}'$ is a cover, such S_j exists. Hence, the optimal $s - y_\ell$ replacement path in $G \setminus \{e_i\}$, which is by Obs. 5.2(b), $P' = \widetilde{P}_i \circ (z_i, x_j) \circ (x_j, y_\ell)$, exists in $\widehat{T} \setminus \{e_i\}$ for every $y_\ell \in Y$. It follows that $\widehat{T} \in \mathcal{T}(s, G)$, hence $\texttt{Cost}^*(s, G) \leq |E(\widehat{T})| = |\widetilde{E}| + \kappa \cdot R$. The lemma follows. ∎

Let κ^* be the cost of the optimal Set-Cover for the instance $\langle U, \mathfrak{F} \rangle$. We have the following.

Corollary 5.6 $\texttt{Cost}^*(s, G) = |\widetilde{E}| + \kappa^* \cdot R$.

Proof: Let $T^* \in \mathcal{T}(s, G)$ be such that $|E(T^*)| = \texttt{Cost}^*(s, G)$. It then holds that

$$|\widetilde{E}| + \kappa(T^*) \cdot R \leq |E(T^*)| = \texttt{Cost}^*(s, G) \leq |\widetilde{E}| + \kappa^* \cdot R,$$

where the first inequality holds by Eq. (7) and the second inequality follows by Lemma 5.5. Hence, $\kappa(T^*) \leq \kappa^*$. Since by Lemma 5.4, there exists a cover of size $\kappa(T^*)$, we have that $\kappa^* \leq \kappa(T^*)$. It follows that $\kappa^* = \kappa(T^*)$ and $\texttt{Cost}^*(s, G) = |\widetilde{E}| + \kappa^* \cdot R$ as desired. ∎

We now show that the reduction is gap-preserving. Assume that there exists an α approximation algorithm $\mathcal{A}$ for the Minimum `FT-BFS` problem. Then applying our transformation to an instance $\mathcal{I}(U, \mathfrak{F}) = (G, s)$ would result in an `FT-BFS` tree $\widehat{T} \in \mathcal{T}(s, G)$ such that

$$|\widetilde{E}| + \kappa(\widehat{T}) \cdot R < |E(\widehat{T})| \leq \alpha(|\widetilde{E}| + \kappa^* \cdot R) \leq 3\alpha \cdot \kappa^* \cdot R \ ,$$

where the first inequality follows by Eq. (7), the second by the approximation guarantee of $\mathcal{A}$ and by Cor. 5.6, and the third inequality follows by the fact that $|\widetilde{E}| \leq 2R$. By Lemma 5.4, a cover of size $\kappa(\widehat{T}) \leq 3\alpha\kappa^*$ can be constructed given $\widehat{T}$, which results in a 3α approximation to the Set-Cover instance. As the Set-Cover problem is inapproximable within a factor of $(1 - o(1)) \ln n$, under an appropriate complexity assumption [14], we get that the Minimum `FT-BFS` problem is inapproximable within a factor of $c \cdot \log N$ for some constant $c > 0$. This complete the proof of Thm. 5.1.

5.2 $O(\log n)$-Approximation for FT-MBFS Trees

The goal of this section is to present an $O(\log n)$ approximation algorithm for the Minimum `FT-BFS` Problem (hence also to its special case, the Minimum `FT-BFS` Problem, where $|S| = 1$).

Theorem 5.7 *There exists a polynomial time algorithm that for every n-vertex graph G and source node set $S \subseteq V$ constructs an `FT-MBFS` tree $\widehat{T} \in \mathcal{T}(S, G)$ such that $|E(\widehat{T})| \leq O(\log n) \cdot$ $\mathtt{Cost}^*(S, G)$.*

To prove the theorem, we first describe the algorithm and then bound the number of edges. Let $\mathtt{ApproxSetCover}(\mathfrak{F}, U)$ be an $O(\log n)$ approximation algorithm for the Set-Cover problem, which given a collection of sets $\mathfrak{F} = \{S_1, \ldots, S_M\}$ that covers a universe $U = \{u_1, \ldots, u_N\}$ of size N, returns a cover $\mathfrak{F}' \subseteq \mathfrak{F}$ that is larger by at most $O(\log N)$ than any other $\mathfrak{F}'' \subseteq \mathfrak{F}$ that covers U (cf. [29]).

The Algorithm. Starting with $\widehat{T} = \emptyset$, the algorithm adds edges to $\widehat{T}$ until it becomes an `FT-MBFS` tree.

Set an arbitrary order on the vertices $V(G) = \{v_1, \ldots, v_n\}$ and on the edges $E^+ = E(G) \cup \{e_0\} = \{e_0, \ldots, e_m\}$ where e_0 is a new fictitious edge whose role will be explained later on. For every node $v_i \in V$, define

$$U_i = \{\langle s_k, e_j \rangle \ \mid s_k \in S \setminus \{v_i\}, e_j \in E^+\}.$$

The algorithm consists of n rounds, where in round i it considers v_i. Let $\Gamma(v_i, G) = \{u_1, \ldots, u_{d_i}\}$ be the set of neighbors of v_i in some arbitrary order, where $d_i = \mathtt{deg}(v_i, G)$. For every neighbor u_j, define a set $S_{i,j} \subseteq U_i$ containing certain source-edge pairs $\langle s_k, e_\ell \rangle \in U_i$. Informally, a set $S_{i,j}$ contains the pair $\langle s_k, e_\ell \rangle$ iff there exists an $s_k - v_i$ shortest path in $G \setminus \{e_\ell\}$ that goes through the neighbor u_j of v_i. Note that $S_{i,j}$ contains the pair $\langle s_k, e_0 \rangle$ iff there exists an $s_k - v_i$ shortest-path in $G \setminus \{e_0\} = G$ that goes through u_j. I.e., the fictitious edge e_0 is meant to capture the case where no fault occurs, and thus we take care of true shortest-paths in G. Formally, every pair $\langle s_k, e_\ell \rangle \in U_i$ is included in every set $S_{i,j}$ satisfying that

$$\mathrm{dist}(s_k, u_j, G \setminus \{e_\ell\}) = \mathrm{dist}(s_k, v_i, G \setminus \{e_\ell\}) - 1. \tag{8}$$

Let $\mathfrak{F}_i = \{S_{i,1}, \ldots, S_{i,d_i}\}$. The edges of v_i that are added to $\widehat{T}$ in round i are now selected by using algorithm `ApproxSetCover` to generate an approximate solution for the set cover problem on the collection $\mathfrak{F} = \{S_{i,j} \ \mid \ u_j \in \Gamma(v_i, G)\}$. Let $\mathfrak{F}'_i = \mathtt{ApproxSetCover}(\mathfrak{F}_i, U_i)$. For every $S_{i,j} \in \mathfrak{F}'_i$, add the edge (u_j, v_i) to $\widehat{T}$.

Analysis. We first show that algorithm constructs an `FT-MBFS` $\widehat{T} \in \mathcal{T}(S, G)$ and then bound its size.

Lemma 5.8 $\widehat{T} \in \mathcal{T}(S, G)$.

Proof: Assume, towards contradiction, that $\widehat{T} \notin \mathcal{T}(S, G)$. Let $s \in S$ be some source node such that $\widehat{T} \notin \mathcal{T}(s, G)$ is not an `FT-BFS` tree with respect to s. By the assumption, such s exists. Let

$$BP = \{(i, k) \mid v_i \in V, e_k \in E^+ \text{ and } \mathrm{dist}(s, v_i, \widehat{T} \setminus \{e_k\}) > \mathrm{dist}(s, v_i, G \setminus \{e_k\})\}$$

be the set of "bad pairs," namely, vertex-edge pairs (i,k) for which the $s-v_i$ shortest path distance in $\widehat{T}\setminus\{e_k\}$ is greater than that in $G\setminus\{e_k\}$. (By the assumption that $\widehat{T}\notin\mathcal{T}(s,G)$, it holds that $BP\neq\emptyset$.) For every vertex-edge pair (i,k), where $v_i\in V\setminus\{s\}$ and $e_k\in E^+$, define an $s-v_i$ shortest-path $P^*_{i,k}$ in $G\setminus\{e_k\}$ in the following manner. Let $u_j\in\Gamma(v_i,G)$ be such that the pair $\langle s,e_k\rangle\in S_{i,j}$ is covered by the set $S_{i,j}$ of u_j and $S_{i,j}\in\mathfrak{F}'_i$ is included in the cover returned by the algorithm `ApproxSetCover` in round i. Thus, $(u_j,v_i)\in\widehat{T}$ and $\text{dist}(s,u_j,G\setminus\{e_k\})=\text{dist}(s,v_i,G\setminus\{e_k\})-1$. Let $P'\in SP(s,u_j,G\setminus\{e_k\})$ and define

$$P^*_{i,k}=P'\circ(u_j,v_i).$$

By definition, $|P^*_{i,k}|=\text{dist}(s,v_i,G\setminus\{e_k\})$ and by construction, $\texttt{LastE}(P^*_{i,k})\in\widehat{T}$. Define $BE(i,k)=P^*_{i,k}\setminus E(\widehat{T})$ to be the set of "bad edges," namely, the set of $P^*_{i,k}$ edges that are missing in $\widehat{T}$. By definition, $BE(i,k)\neq\emptyset$ for every bad pair $(i,k)\in BP$. Let $d(i,k)=\max_{e\in BE(i,k)}\{\text{dist}(s,e,P^*_{i,k})\}$ be the maximal depth of a missing edge in $BE(i,k)$, and let $DM(i,k)$ denote that "deepest missing edge" for (i,k), i.e., the edge e on $P^*_{i,k}$ satisfying $d(i,k)=\text{dist}(s,e,P^*_{i,k})$. Finally, let $(i',k')\in BP$ be the pair that minimizes $d(i,k)$, and let $e_1=(v_{\ell_1},v_{i_1})\in BE(i',k')$ be the deepest missing edge on $P^*_{i',k'}$, namely, $e_1=DM(i',k')$. Note that e_1 is the *shallowest* "deepest missing edge" over all bad pairs $(i,k)\in BP$. Let $P_1=P^*_{i_1,k'}$, $P_2=P^*_{i',k'}[s,v_{i_1}]$ and $P_3=P^*_{i',k'}[v_{i_1},v_{i'}]$; see Fig. 6 for illustration. Note that since $(i',k')\in BP$, it follows that also $(i_1,k')\in BP$. (Otherwise, if $(i_1,k')\notin BP$, then any $s-v_{i_1}$ shortest-path $P'\in SP(s,v_{i_1},\widehat{T}\setminus\{e_{k'}\})$, where $|P'|=|P^*_{i_1,k'}|$, can be appended to P_3 resulting in $P''=P'\circ P_3$ such that (1) $P''\subseteq\widehat{T}\setminus\{e_{k'}\}$ and (2) $|P''|=|P'|+|P_3|=|P_2|+|P_3|=|P^*_{i',k'}|$, contradicting the fact that $(i',k')\in BP$.) Thus we conclude that $(i_1,k')\in BP$. Finally, note that $\texttt{LastE}(P_1)\in\widehat{T}$ by definition, and therefore the deepest missing edge of (i,k) must be shallower, i.e., $d(i_1,k')<d(i',k')$. However, this is in contradiction to our choice of the pair (i',k'). The lemma follows. ▮

Let $W:E(G)\to\mathbb{R}_{>0}$ be the weight assignment that guarantees the uniqueness of shortest-paths. Note that the algorithm did not use W in the computation of the shortest-paths. For every node v_i, let $\Gamma(v_i,G)=\{u_1,\ldots,u_{d_i}\}$ be its ordered neighbor set as considered by the algorithm. For every `FT-MBFS` tree $\widetilde{T}\in\mathcal{T}(S,G)$, $v_i\in V, e_\ell\in E^+$ and $s_k\in S$, let $\widetilde{P}_i(s_k,e_\ell)\in SP(s_k,v_i,\widetilde{T}\setminus\{e_\ell\},W)$ be an s_k-v_i shortest-path in $\widetilde{T}\setminus\{e_\ell\}$. Let

$$A_i(\widetilde{T})=\{\texttt{LastE}(\widetilde{P}_i(s_k,e_\ell))\ \mid\ e_\ell\in E^+,s_k\in S\setminus\{v_i\}\}$$

be the edges of v_i that appear as last edges in the shortest-paths and replacement paths from S to v_i in $\widetilde{T}$. Define

$$\mathfrak{F}_i(\widetilde{T})=\{S_{i,j}\ \mid\ (u_j,v_i)\in A_i(\widetilde{T})\}.$$

We then have that

$$|\mathfrak{F}_i(\widetilde{T})|=|A_i(\widetilde{T})|\ . \tag{9}$$

The correctness of the algorithm (see Lemma 5.8) established that if a subgraph $\widetilde{T}\subseteq G$ satisfies that $\mathfrak{F}_i(\widetilde{T})$ is a cover of U_i for every $v_i\in V$, then $\widetilde{T}\in\mathcal{T}(S,G)$. We now turn to show the reverse direction.

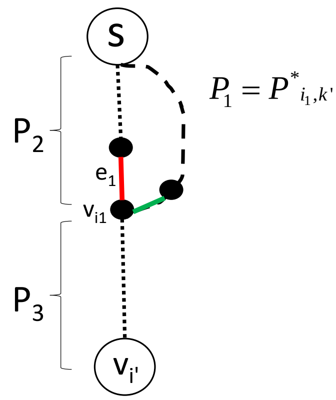

Figure 6: Red solid lines correspond to new edges. The "deepest missing edge" for (i', k'), edge e_1, is the shallowest such edge over all bad pairs in BP. Yet the pair (i_1, k') is bad too. As the last (green) edge of P_1 is included in the FT-MBFS tree, and since P_1 and P_2 are of the same length, it follows that P_1 has a shallower "deepest missing edge".

Lemma 5.9 *For every $\widetilde{T} \in \mathcal{T}(S,G)$, the collection $\mathfrak{F}_i(\widetilde{T})$ is a cover of U_i, namely, $\bigcup_{S_{i,j} \in \mathfrak{F}_i(\widetilde{T})} S_{i,j} = U_i$, for every $v_i \in V$.*

Proof: Assume, towards contradiction, that there exists an FT-MBFS tree $\widetilde{T} \in \mathcal{T}(S,G)$ and a vertex $v_i \in V$ whose corresponding collection of sets $\mathfrak{F}_i(\widetilde{T})$ does not cover U_i. Hence there exists at least one uncovered pair $\langle s_k, e_\ell \rangle \in U_i$, i.e.,

$$\langle s_k, e_\ell \rangle \in U_i \setminus \bigcup_{S_{i,j} \in \mathfrak{F}_i(\widetilde{T})} S_{i,j} \ . \tag{10}$$

By definition $s_k \neq v_i$. We next claim that $\widetilde{T}$ does not contain an optimal $s_k - v_i$ path when the edge e_ℓ fails, contradicting the fact that $\widetilde{T} \in \mathcal{T}(S,G)$. That is, we show that

$$\text{dist}(s_k, v_i, \widetilde{T} \setminus \{e_\ell\}) > \text{dist}(s_k, v_i, G \setminus \{e_\ell\}).$$

Towards contradiction, assume otherwise, and let $(u_j, v_i) = \texttt{LastE}(P^*_{i,\ell})$ where $P^*_{i,\ell} \in SP(s_k, v_i, \widetilde{T} \setminus \{e_\ell\}, W)$, hence $(u_j, v_i) \in A_i(\widetilde{T})$ and $S_{i,j} \in \mathfrak{F}_i(\widetilde{T})$. By the contradictory assumption, $|P^*_{i,\ell}| = \text{dist}(s_k, v_i, G \setminus \{e_\ell\})$ and hence $\text{dist}(s_k, u_j, G \setminus \{e_\ell\}) = \text{dist}(s_k, v_i, G \setminus \{e_\ell\}) - 1$. This implies that $\langle s_k, e_\ell \rangle \in S_{i,j} \in \mathfrak{F}_i(\widetilde{T})$, in contradiction to Eq. (10), stating that $\langle s_k, e_\ell \rangle$ is not covered by $\mathfrak{F}_i(\widetilde{T})$. The lemma follows. ▮

We now turn to bound that number of edges in $\widehat{T}$.

Lemma 5.10 $|E(\widehat{T})| \leq O(\log n) \cdot \texttt{Cost}^*(S,G)$.

Proof: Let $\delta = c \log n$ be the approximation ratio guarantee of ApproxSetCover. For ease of notation, let $O_i = A_i(T^*)$ for every $v_i \in V$. Let $\mathfrak{F}_i = \{S_{i,1}, \ldots, S_{i,d_i}\}$ be the collection of v_i sets considered at round i where $S_{i,j} \subseteq U_i$ is the set of the neighbor $u_j \in \Gamma(v_i, G)$ computed according to Eq. (8).

Let $\mathfrak{F}'_i = \texttt{ApproxSetCover}(\mathcal{S}_i, U_i)$ be the cover returned by the algorithm and define $A_i = \{(u_j, v_i) \mid S_{i,j} \in \mathfrak{F}'_i\}$ as the collection of edges whose corresponding sets are included in $\mathcal{S}'_i$. Thus, by Eq. (9), $|O_i| = |\mathfrak{F}_i(T^*)|$ and $|A_i| = |\mathfrak{F}'_i|$ for every $v_i \in V$.

Observation 5.11 *$|A_i| \le \delta |O_i|$ for every $v_i \in V \setminus \{s\}$.*

Proof: Assume, towards contradiction, that there exists some i such that $|A_i| > \delta |O_i|$. Then by Eq. (9) and by the approximation guarantee of `ApproxSetCover` where in particular $|\mathfrak{F}_i(\widetilde{T})| \le \delta |\mathfrak{F}''_i|$ for every $\mathfrak{F}''_i \subseteq \mathfrak{F}_i$ that covers U_i, it follows that $\mathfrak{F}_i(T^*)$ is not a cover of U_i. Consequently, it follows by Lemma 5.9 that $T^* \notin \mathcal{T}(S, G)$, contradiction. The observation follows. ∎

Since $\bigcup A_i$ contains precisely the edges that are added by the algorithm to the constructed `FT-MBFS` tree $\widehat{T}$, we have that

$$|E(\widehat{T})| \le \sum_i |A_i| \le \delta \sum_i |O_i| \le 2\delta \cdot \texttt{Cost}^*(S, G) ,$$

where the second inequality follows by Obs. 5.11 and the third by the fact that $|E(T^*)| \ge \sum_i |O_i|/2$ (as every edge in $\bigcup_{v_i \in V} O_i$ can be counted at most twice, by both its endpoints). The lemma follows. ∎

Thm. 5.7 is established.

Acknowledgment. We are grateful to Gilad Braunschvig, Alon Brutzkus, Adam Sealfon, Oren Weimann and the anonymous reviewers for helpful comments.

References

[1] B. Awerbuch, A. Bar-Noy, N. Linial, and D. Peleg. Compact distributed data structures for adaptive network routing. In *Proc. 21st ACM Symp. on Theory of Computing*, 230–240, 1989.

[2] I. Abraham, S. Chechik, and C. Gavoille. Fully dynamic approximate distance oracles for planar graphs via forbidden-set distance labels. In *Proc 44th Symp. on Theory of Computing*, 1199–1218, New York, 2012.

[3] I. Abraham, S. Chechik, C. Gavoille and D. Peleg. Forbidden-Set Distance Labels for Graphs of Bounded Doubling Dimension. In *Proc. 29th ACM Symp. on Principles of Distributed Computing*, 2010, 192–200.

[4] S. Baswana and S. Sen. Approximate distance oracles for unweighted graphs in expected $O(n^2)$ time. *ACM Trans. Algorithms*, 2(4):557–577, 2006.

[5] A. Bernstein and D. Karger. A nearly optimal oracle for avoiding failed vertices and edges. In *Proc. 41st ACM Symp. on Theory of Computing*, 101–110, 2009.

[6] S. Chechik, M. Langberg, D. Peleg, and L. Roditty. f-sensitivity distance oracles and routing schemes. *Algorithmica*, 861–882, 2012.

[7] S. Chechik, M. Langberg, D. Peleg, and L. Roditty. Fault-tolerant spanners for general graphs. In *Proc. 41st ACM Symp. on Theory of computing*, 435–444, 2009.

[8] S. Chechik. Fault-Tolerant Compact Routing Schemes for General Graphs. *Inf. Comput.* **(222)**, 2013, 36–44.

[9] B. Courcelle and A. Twigg. Compact forbidden-set routing. In *Proc. 24th Symp. on Theoretical Aspects of Computer Science*, 37–48, 2007.

[10] A. Czumaj and H. Zhao. Fault-tolerant geometric spanners. *Discrete & Computational Geometry*, 32, 2003.

[11] C. Demetrescu, M. Thorup, R. Chowdhury, and V. Ramachandran. Oracles for distances avoiding a failed node or link. *SIAM J. Computing*, 37:1299–1318, 2008.

[12] M. Dinitz and R. Krauthgamer. Fault-tolerant spanners: better and simpler. In *Proc. ACM Symp. on Principles of Distributed Computing*, 2011, 169-178.

[13] R. Duan and S. Pettie. Dual-failure distance and connectivity oracles. In *Proc. 20th ACM-SIAM Symp. on Discrete Algorithms*, 2009.

[14] U. Feige. A Threshold of ln n for Approximating Set Cover. *J. ACM*, 634–652, 1998.

[15] F. Grandoni and V.V Williams. Improved Distance Sensitivity Oracles via Fast Single-Source Replacement Paths. In *Proc. IEEE Symp. on Foundations of Computer Science*, 2012.

[16] J. Hershberger and S. Subhash. Vickrey prices and shortest paths: What is an edge worth?. In *Proc. IEEE Symp. on Foundations of Computer Science*, 2001.

[17] J. Hershberger and S. Subhash and A. Bhosle. On the difficulty of some shortest path problems. In *Proc. STACS*, 343–354, 2003.

[18] C. Levcopoulos, G. Narasimhan, and M. Smid. Efficient algorithms for constructing fault-tolerant geometric spanners. In *Proc. 30th ACM Symp. on Theory of computing*, 186–195, 1998.

[19] T. Lukovszki. New results of fault tolerant geometric spanners. In *Proc. 6th Workshop on Algorithms and Data Structures*, London 193–204, 1999.

[20] D. Peleg. *Distributed Computing: A Locality-Sensitive Approach.* SIAM, 2000.

[21] D. Peleg. As good as it gets: Competitive fault tolerance in network structures. In *Proc. 11th Symp. on Stabilization, Safety, and Security of Distributed Systems*, LNCS 5873, 2009, 35–46.

[22] D. Peleg and A.A. Schäffer. Graph spanners. *J. Graph Theory*, 13:99–116, 1989.

[23] D. Peleg and J.D. Ullman. An optimal synchronizer for the hypercube. *SIAM J. Computing*, 18(2):740–747, 1989.

[24] D. Peleg and E. Upfal. A trade-off between space and efficiency for routing tables. *J. ACM*, 36:510–530, 1989.

[25] L. Roditty, M. Thorup, and U. Zwick. Deterministic constructions of approximate distance oracles and spanners. In *Proc. 32nd Int. Colloq. on Automata, Languages & Prog.*, 261–272, 2005.

[26] L. Roditty and U. Zwick. Replacement paths and k simple shortest paths in unweighted directed graphs. *ACM Trans. Algorithms* ,2012.

[27] M. Thorup and U. Zwick. Compact routing schemes. In *Proc. 14th ACM Symp. on Parallel Algorithms and Architectures*, Hersonissos, Crete, 1–10, 2001.

[28] M. Thorup and U. Zwick. Approximate distance oracles. *J. ACM*, 52:1–24, 2005.

[29] V. Vazirani. Approximation Algorithms. *Georgia Inst. Tech.*, 1997.

[30] O. Weimann and R. Yuster. Replacement paths via fast matrix multiplication. *51th IEEE Symp. on Foundations of Computer Science*, 2010.