A Deterministic Approach to Shortest Path Restoration in Edge Faulty Graphs

Keerti Choudhary
Department of Computer Science and Engineering, IIT Delhi, India

Rishabh Dhiman
Department of Computer Science and Engineering, IIT Delhi, India

Abstract

Afek, Bremler-Barr, Kaplan, Cohen, and Merritt (PODC'01) in their seminal work on shortest path restorations demonstrated that after a single edge failure in a graph G, a *replacement shortest path* between any two vertices s and t, which avoids the failed edge, can be represented as the concatenation of two original shortest paths in G. They also showed that we cannot associate a canonical[1] shortest path between the vertex pairs in G that consistently allows for the replacement path (in the surviving graph) to be represented as a concatenation of these canonical paths. Recently, Bodwin and Parter (PODC'21) proposed a randomized tie-breaking scheme for selecting canonical paths for the "ordered" vertex pairs in graph G with the desired property of representing the replacement shortest path as a concatenation of canonical shortest-paths provided for ordered pairs.

An interesting open question is whether it is possible to provide a deterministic construction of canonical paths in an efficient manner. We address this question in our paper by presenting an $O(mn)$ time deterministic algorithm to compute a canonical path family $\mathcal{F} = \{P_{x,y}, Q_{x,y} \mid x, y \in V\}$ comprising of two paths per (unordered) vertex pair. Each replacement is either a PQ-path (of type $P_{x,y} \circ Q_{y,z}$), a QP-path, a QQ-path, or a PP-path. Our construction is fairly simple and is a straightforward application of independent spanning trees. We also present various applications of family $\mathcal{F}$ in computing fault-tolerant structures.

2012 ACM Subject Classification Theory of computation → Data structures design and analysis; Mathematics of computing → Graph algorithms

Keywords and phrases Fault-tolerant Data-structures, Shortest Path Restoration, Replacement path

Digital Object Identifier 10.4230/LIPIcs.STACS.2025.24

Funding *Keerti Choudhary*: The author is supported in part by Google India Algorithms Research grant 2021.

1 Introduction

In their seminal work on shortest path restorations, Afek, Bremler-Barr, Kaplan, Cohen, and Merritt [1] studied the question of how the structure of shortest paths in a graph changes when edges fail. They showed that, after failure of any single edge in a graph, a replacement shortest path between any two vertices s and t avoiding a failed edge can be represented as the concatenation of two original shortest paths in the graph.

▶ **Theorem 1** (Afek et al. [1]). *For any vertices s, t and any failing edge e in G, a replacement shortest path between s and t in $G \setminus e$ is a concatenation of two original shortest paths in G, namely, $\pi(s, x)$ and $\pi(x, t)$, where vertex x is a function of s, t, and e.*

Various works in the past have employed the restoration path theory to develop efficient solutions for problems such as distance sensitivity oracles [2, 7, 8, 9], replacement paths [6, 5], fault-tolerant distance preservers [4, 5], routing schemes [5], among others.

[1] In Afek et al. (1991), the term "set of base paths" is used to refer to what is termed as "canonical paths" here.

42nd International Symposium on Theoretical Aspects of Computer Science (STACS 2025).
Editors: Olaf Beyersdorff, Michał Pilipczuk, Elaine Pimentel, and Nguyễn Kim Thắng;
Article No. 24; pp. 24:1–24:10
Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

An f-fault-tolerant distance preserver for a graph G is a sparse subgraph H such that, after failure of at most f edges, the distances between a pre-specified set of vertex pairs in the surviving part of H still match the corresponding distances in the surviving part of G. They are formally defined as follows.

▶ **Definition 2** (Fault Tolerant Distance Preservers). *For a graph $G = (V, E)$ and demand pairs P, a subgraph $H = (V, E_H \subseteq E)$ of G is an f-fault tolerant distance preserver (f-preserver) for P if for any subset F of E of size at most f, we have*

$$\text{dist}(s, t, G \setminus F) = \text{dist}(s, t, H \setminus F), \quad \textit{for all } (s, t) \in P.$$

For a set $S \subseteq V$, we say that H is a ***source-wise distance preserver*** *for S if $P = S \times V$; and a* ***subset distance preserver*** *for S if $P = S \times S$.*

Parter and Peleg [15] showed that for any n-vertex unweighted undirected graph, there exists a 1-fault-tolerant $S \times V$ preserver with $O(|S|^{1/2}n^{3/2})$ edges. They also showed that this bound is existentially tight. Specifically, there exist n-vertex graphs and a set of sources S such that any $S \times V$ fault-tolerant preserver must have $\Omega\left(|S|^{1/2}n^{3/2}\right)$ edges. Later, Parter [14] extended this result by providing an upper bound of $O(|S|^{1/3}n^{5/3})$ for dual fault-tolerant $S \times V$ preservers in undirected graphs. Gupta and Khan [11] extended this to directed graphs. For general f, Bodwin, Grandoni, Parter, and Williams [4] presented a construction of an f-fault-tolerant $S \times V$ preserver with $\widetilde{O}(|S|^{1/2^f} f n^{2-1/2^f})$ edges.

For the problem of computing subset distance preservers, Bodwin, Choudhary, Parter and Shahar [3] presented a construction of 1-fault-tolerant $S \times S$ distance preserver with $\widetilde{O}(|S|n)$ edges. Recently, Bodwin and Parter [5] extended this work to general f failures by providing a construction of $(f + 1)$-fault-tolerant $S \times S$ preserver with $\widetilde{O}(fn^{2-1/2^f}|S|^{1/2^f})$ edges, for any $f \geqslant 0$. These constructions for $S \times S$ preservers are obtained by employing the work of Afek, Bremler-Barr, Kaplan, Cohen, and Merritt [1] (see Theorem 1) on the structure of shortest paths in fault-prone graphs.

To exploit Theorem 1, Bodwin and Parter [5] introduced a tie-breaking scheme for selecting shortest paths in graph $G \setminus F$. Specifically, they used the Isolation lemma [13] to show that if each edge in G is assigned a random integer weight from the range $[1, n^{f+c+2}]$, then with probability $1 - 1/n^c$, for any pair of vertices s and t, and for any set F of edges, where $|F| \leqslant f$, exactly one of the shortest paths from s to t in $G \setminus F$ will be the unique shortest path in the modified edge weighted graph $G \setminus F$.

The main drawback of this tie-breaking scheme is that it is randomized and requires an additional $n^{O(f)}$ time to derandomize. This raises the following natural question:

Question: Is it possible to bypass randomness and have an efficient deterministic construction of the canonical path family?

We address this question by presenting a deterministic algorithm that runs in $O(mn)$ time to compute a canonical path family $\mathcal{F} = \{P_{x,y}, Q_{x,y} \mid x, y \in V\}$ comprising of two paths for each (unordered) vertex pair in G. We show that each replacement path in G can take the form of a PQ-path (of type $P_{x,y} \circ Q_{y,z}$), a QP-path, a QQ-path, or a PP-path. Our construction is not only straightforward but also represents a simple yet effective application of independent spanning trees [10].

We present in this paper an efficient construction of the family $\mathcal{F}$ (see Theorem 10) and also explore various applications of these structures.

Sourcewise Distance Preservers

We introduce a stronger notion of fault-tolerant distance preserving subgraphs, which we call $(f,1)$-preservers.

▶ **Definition 3** ($(f,1)$-Preservers). *For a graph $G=(V,E)$ and a set of demand pairs P, a subgraph $H=(V,E_H\subseteq E)$ of G is an $(f,1)$-preserver if, for every pair $(s,t)\in P$, every set F of edges of size at most f, and every edge e in G that satisfies $\operatorname{dist}(s,t,G\setminus F)=\operatorname{dist}(s,t,G\setminus F\cup e)$, we have*

$$\operatorname{dist}(s,t,H\setminus(F\cup e))=\operatorname{dist}(s,t,G\setminus(F\cup e)).$$

Note that for any $f\geqslant 0$, an $(f+1)$-preserver is also an $(f,1)$-preserver. However, an f-preserver is not necessarily an $(f,1)$-preserver.

Our first contribution is a construction of $(f,1)$-preservers in polynomial time that matches the size of the current best construction of f-preservers given by Bodwin, Grandoni, Parter, and Williams [4].

▶ **Theorem 4.** *Let $f\geqslant 1$ be a positive integer. For any undirected, unweighted n-vertex graph $G=(V,E)$ with a set of source vertices $S\subseteq V$, we can compute a $(f,1)$-source-wise preserver for G with $\widetilde{O}(fn^{2-1/2^f}|S|^{1/2^f})$ edges in $O(fmn)$ time.*

Further, for $f=0$, we can compute a $(0,1)$-source-wise preserver for G with $O(|S|n)$ edges in $O(m+n)$ time.

Subset Distance Preservers

We provide the following relation between $(f,1)$-preservers and $(f+1)$-fault-tolerant *subset* distance preservers.

▶ **Theorem 5.** *In any undirected graph $G=(V,E)$, a source-wise $(f,1)$-preserver H with respect to a set S is an $(f+1)$-fault-tolerant subset distance preserver for pairs in $S\times S$.*

Combined with the construction of $(f,1)$-preservers, this gives us a deterministic algorithm to compute subset distance preservers in polynomial time.

▶ **Theorem 6.** *For any n-vertex graph $G=(V,E)$, a set of source vertices $S\subseteq V$, and a fixed non-negative integer f, there is an $(f+1)$-fault-tolerant distance preserver of G for pairs in $S\times S$ with $\widetilde{O}((f+1)n^{2-1/2^f}|S|^{1/2^f})$ edges which can be computed in $O(fmn)$ time when $f\geqslant 1$, and $O(|S|m)$ time when $f=0$.*

Prior to our work, Bodwin et al. [5] gave construction for such preservers that takes $O(n^{2-1/2^f}|S|^{1/2^f})$ space and requires $O(mn)$ computation time in the randomized setting. However, the deterministic variant of their algorithm had a time complexity of $n^{O(f)}$, which is polynomial only for constant values of f. Our work improves upon this by providing a deterministic algorithm with a time complexity of $O((f+1)mn)$, almost matching the efficiency of the randomized version of [5].

Fault-tolerant Distance Labeling Scheme

A distance labeling scheme is an assignment of bit-string labels to each node of G such that we can recover $\operatorname{dist}(s,t,G)$ by looking at only the labels at s and t. An f-fault-tolerant distance labeling scheme allows us to compute the distances even when a set F of edges of size at most f fail.

While it is possible to store the entire graph within the labels, our objective is to find a labeling of subquadratic size. Bodwin et al. [5] gave an f-fault-tolerant distance labeling scheme that requires $O(n^{2-1/2^f} \log n)$ bits per vertex and can be computed in $O(mn)$ time in the randomized setting, and in $n^{O(f)}$ time deterministically. We offer an improvement to this by employing our $(f, 1)$-preservers, which allows for the computation of the labels deterministically in polynomial time.

▶ **Theorem 7.** *For any fixed nonnegative integer $f \geqslant 0$, and n-vertex unweighted undirected graph, there is an $(f+1)$-fault-tolerant distance labeling scheme that assigns each vertex a label of $O((f+1)n^{2-1/2^f} \log n)$ bits that can be computed in $O(fmn)$ time each when $f \geqslant 1$, and $O(m)$ time when $f = 0$.*

2 Preliminaries

Given a directed graph $G = (V, E)$ on $n = |V|$ vertices and $m = |E|$ edges, the following notations and definitions will be used throughout the paper. We omit the term G when graph is clear from context.

- $\text{dist}(x, y, G)$: Distance of node y from x in the graph G.
- $G \setminus F$: The graph obtained by removing the edges that lie in F from the graph G.
- $G \cup H$: The graph obtained by taking a union of the vertices and edges of G and H.
- $E(P)$: The edges lying in path P.
- $T[x, y]$: The path from vertex x to vertex y in tree T.
- $P[x, y]$: The subpath of path P lying between vertices x, y, assuming x precedes y on P.
- $\textsc{In-Edges}(v, G)$: The set of all edges incoming to v in G.
- (s, v)-cut-edge: An edge which lies on all (s, v)-paths and hence whose removal disconnects v from s.
- (s, v)-distance-cut edge: An edge that lies on all the (s, v) shortest paths and hence whose removal increases the distance from s to v.
- $\textsc{LastE}(P)$: the last edge lying on the path P.

All graphs in this paper are undirected and unweighted, unless stated otherwise.

3 Fault Tolerant Preservers

3.1 Preservers for Source-wise setting

▶ **Theorem 8.** *Any undirected or directed graph $G = (V, E)$ with a positive integer $f \geqslant 1$ and a set $S \subseteq V$ of sources has a source-wise $(f, 1)$-preserver H with $\widetilde{O}(f|S|^{1/2^f} n^{2-1/2^f})$ edges, which can be computed in $O(fmn)$ time.*

We also improve the running time for $f = 0$.

▶ **Theorem 9.** *Any undirected or directed graph $G = (V, E)$ and a set $S \subseteq V$ of sources has a source-wise $(0, 1)$-preserver H with $\widetilde{O}(|S|n)$ edges which can be computed in $O(|S|m)$ time.*

We first establish the following theorem.

▶ **Theorem 10.** *Any directed/undirected graph $G = (V, E)$ with a destination vertex $t \in V$ can be processed in $O(m + n)$ time to implicitly compute, for each $s \in V$, a pair of (s, t)-shortest paths, denoted as $P_{s,t}$ and $Q_{s,t}$, which intersect solely at the (s, t)-distance-cut edges.*

Furthermore, if G is undirected then for any $s, t \in V$ and any $e \in E$, there exists a vertex $w \in V$ such that at least one of the four paths obtained by concatenating paths from the family $\{P_{s,w}, Q_{s,w}\} \times \{P_{w,t}, Q_{w,t}\}$ forms a replacement shortest path between s and t in $G \setminus e$.

In order to prove Theorem 10, we use the concept of independent trees. Given a directed graph G and a designated source r, a pair of trees T_1, T_2 rooted at r are said to be *independent trees*, if for each $v \neq r$, the paths from v to r in T_1 and T_2 intersect only at the (v, r)-cut-edges.

▶ **Theorem 11** (Georgiadis and Tarjan [10])**.** *Given a directed graph G and a designated source r, a pair of independent trees T_1, T_2 rooted at r are computable in $O(m + n)$ time.*

We are now ready to prove Theorem 10.

Proof of Theorem 10. We begin by proving the first claim. Consider a directed acyclic graph (DAG) $D = (V, E_D \subseteq E)$ containing only those edges $e = (x, y) \in E$ for which $\text{dist}(x, t, G) = \text{dist}(y, t, G) + 1$. Let T_1 and T_2 be a pair of independent trees rooted at t in D. Then, for any vertex $v \in V$, the paths from v to t in T_1 and T_2 intersect only at the (v, t)-cut edges in D. By Theorem 11, the time required to compute T_1 and T_2 is $O(m + n)$.

For each $s \in V$, we define $P_{s,t}$ and $Q_{s,t}$ to be the paths $T_1[s, t]$ and $T_2[s, t]$, respectively. Note that there is a one-to-one correspondence between (v, t)-cut edges in D and (v, t)-distance-cut edges in G. Therefore, $P_{s,t}$ and $Q_{s,t}$ are the shortest (s, t)-paths in G, and they intersect only at the edges whose failure increases the (s, t)-distance in G.

Next, we prove the second claim. By Theorem 1, for any pair $s, t \in V$ and any failing edge e, there exists a vertex w such that:

1. $\text{dist}(s, w, G \setminus e) = \text{dist}(s, w, G)$,
2. $\text{dist}(w, t, G \setminus e) = \text{dist}(w, t, G)$,
3. $\text{dist}(s, t, G \setminus e) = \text{dist}(s, w, G) + \text{dist}(w, t, G)$.

This implies that e is not a distance cut-edge for the pairs (s, w) and (w, t). Therefore, at least one of the paths from $P_{s,w}, Q_{s,w}$ must avoid e, and similarly, at least one of the paths from $P_{w,t}, Q_{w,t}$ must avoid e.

Thus, we conclude that at least one of the four paths obtained by concatenating paths from the family $\{P_{s,w}, Q_{s,w}\} \times \{P_{w,t}, Q_{w,t}\}$ forms a replacement shortest path between s and t in $G \setminus e$. ◀

Proof of Theorem 9. For a source $s \in S$, let T_1^s and T_2^s be the independent trees constructed in the proof of Theorem 10. Then the graph $\bigcup_{s \in S}(T_1^s \cup T_2^s)$ is a $(0, 1)$-preserver. The correctness is a corollary of the earlier proof. ◀

In order to compute H in Theorem 8 we associate a set E_t of edges incident to each node $t \in V$, so that the graph $H = \bigcup_{t \in V} E_t$ obtained by taking the union of edges lying in E_t is an $(f, 1)$-preserver.

Our construction is inspired by FT-BFS algorithm of Bodwin et al. [4], and the running time matches the time taken by their construction of f-FT-BFS. Let S be the source set. For each $s \in S$, we compute a pair of (s, t)-shortest-paths, say $P_{s,t}$ and $Q_{s,t}$, using Theorem 10.

Now we set $L = \sqrt{8f\,|S|\,n \log n}$, and compute a uniformly random subset R of $V \setminus \{t\}$ of size L. Further, for $s \in S$, let

$$W_s = \{u \in V \mid 1 \leqslant \text{dist}(u, t, P_{s,t} \cup Q_{s,t}) \leqslant 8nf \log n / L\}.$$

Finally, with respect to $(\cup_{s \in S} W_s) \cup R$ as the source, we compute an $(f - 1, 1)$-preserver for the graph $G_0 = \left(V, E \setminus \bigcup_{s \in S} E(P_{s,t}) \cup E(Q_{s,t})\right)$ and designate E_t as the set of in-edges of t lying in this $(f - 1, 1)$-preserver. To compute a $(0, 1)$-preserver, we simply take E_t to be the last edges of $E(P_{s,t})$ and $E(Q_{s,t})$. This completes the description of our algorithm. For pseudocode see Algorithm 1.

Algorithm 1 COMPUTE-INCIDENT-EDGES(G, S, t, f).

if $f = 0$ **then**
  Return $\{\text{LASTE}(P_{s,t}), \text{LASTE}(Q_{s,t}) \mid s \in S\}$;
end
$L \leftarrow \sqrt{8f\,|S|\,n \log n}$;
$R \leftarrow$ uniformly random subset of $V \setminus \{t\}$ of size L;
for $s \in S$ **do**
  $(P_{s,t}, Q_{s,t}) \leftarrow$ Pair of (s,t)-shortest-paths computed using Theorem 10;
  $W_s \leftarrow \{u \in V \mid 1 \leqslant \text{dist}(u, t, P_{s,t} \cup Q_{s,t}) \leqslant 8nf \log n/L\}$
end
$G_0 \leftarrow \big(V, E \setminus \bigcup_{s \in S} E(P_{s,t}) \cup E(Q_{s,t})\big)$;
Return COMPUTE-INCIDENT-EDGES$(G_0, (\cup_{s\in S} W_s) \cup R,\ t,\ f-1)$;

▶ **Lemma 12** (Correctness). *The graph $H = \big(V, \bigcup_{t\in V} E_t\big)$, where E_t are sets computed using Algorithm 1, is an $(f,1)$-preserver of G with respect to the source S.*

Proof. We prove the correctness using induction on the value of f. Consider a source node $s \in S$, a set F of failing edges of size t most f, and an edge $e \in E \setminus F$ satisfying $\text{dist}(s, t, G \setminus F) = \text{dist}(s, t, G \setminus F \cup e)$. For $f = 0$, note that $F = \varnothing$, and on the failure of e, at least one of $P_{s,t}$ or $Q_{s,t}$ must remain intact. Therefore, we focus on the case where $|F| \geqslant 1$.

Let $P_{s,t,F}$ and $Q_{s,t,F}$ denote an *arbitrary* pair of (s,t)-shortest paths in $G \setminus F$ intersecting only at (s,v)-distance-cut edges. Further, let x and y be respectively the last vertices in $P_{s,t,F}$ and $Q_{s,t,F}$ lying in the structure $\bigcup_{r\in S}(P_{r,t} \cup Q_{r,t}) \setminus t$, where, $P_{r,t}$ and $Q_{r,t}$ are (r,t)-shortest-paths computed using Theorem 10. Observe that we can assume $(P_{s,t} \cup Q_{s,t})$ contains at least one edge from the set F, as otherwise, it suffices to keep the last edges of $P_{s,t}$, $Q_{s,t}$ in the set E_t.

Next, we compute a vertex $\bar{x}$ lying in $P_{s,t,F}$ as follows. If x lies in W_s, then simply set $\bar{x} = x$. If x does not lie in W_s, then $|P_{s,t,F}[x,t]| = \text{dist}(x, t, G \setminus F) \geqslant \text{dist}(x, t, G) > L$, implying that with a high probability R contains a vertex of path $P_{s,t,F}[x,t]$. So, in this case, $\bar{x}$ is set as an arbitrary vertex of R lying in $P_{s,t,F}[x,t]$. In a similar manner, $\bar{y}$ lying in $Q_{s,t,F}$ can be computed. It must be noted that the internal vertices of suffixes $P_{s,t,F}[\bar{x},t]$ and $Q_{s,t,F}[\bar{y},t]$ are disjoint from the structure $\bigcup_{r\in S}(P_{r,t} \cup Q_{r,t}) \setminus t$.

Observe that on the failure of e in the graph $G \setminus F$, at least one of the paths $P_{s,t,F}$ or $Q_{s,t,F}$ must be intact. Without loss of generality assume that $P_{s,t,F}$ does not contain e. Due to sub-structure property of shortest paths, we have $\text{dist}(\bar{x}, t, G \setminus F) = \text{dist}(\bar{x}, t, G \setminus (F \cup e))$. Thus, $P_{s,t,F}[s,\bar{x}]$ concatenated with an (x,t)-shortest-path in $G \setminus F$ that is disjoint from e gives us an (s,t)-shortest-path in $G \setminus F \cup e$. Such an $(\bar{x},t)$-shortest-path lies in $\bigcup_{t\in V} E_t$ as it contains incoming edges of t in an $(f-1,1)$ preserver of $\big(V, E \setminus \bigcup_{r\in S} E(P_{r,t}) \cup E(Q_{r,t})\big)$ with respect to $\bar{x} \in (\cup_{s\in S} W_s) \cup R$. ◀

In order to bound the size of E_t, we establish a recurrence relation on the size of E_t in the following lemma.

▶ **Lemma 13.** *Let $\mathcal{M}(f,z)$ denote an upper bound on the in-degree of a vertex t in an $(f,1)$-preserver of an n-vertex graph with respect to a source set comprising z vertices. Then,*

$$\mathcal{M}(f,z) \leqslant \mathcal{M}(f-1, 3\sqrt{8fzn\log n})\ .$$

Proof. In Algorithm 1, the size of the set $|W_s|$ is at most $2(8nf \log n/L)$ by Theorem 10, and $|R|$ is exactly L by definition. Since $\sum_{s\in S} |W_s| = 2L$, we have $|(\cup_{s\in S} W_s) \cup R| \leqslant 3L$. By correctness of the algorithm, we get the required recurrence. ◀

▶ **Lemma 14** (Size Analysis). *$\mathcal{M}(f,|S|) = \widetilde{O}(f|S|^{1/2^f} n^{1-1/2^f})$, and therefore, the number of edges in the $(f,1)$-preserver is $\widetilde{O}(f|S|^{1/2^f} n^{2-1/2^f})$.*

Proof. Let $\alpha_f = 3\sqrt{8f\log n}$, and $|S|$ be z. By using $\alpha_f \geqslant \alpha_{f-1}$ and that $\mathcal{M}(f,x)$ is an increasing function in x, we can resolve the recurrence in Lemma 13 as follows,

$$\mathcal{M}(f,z) \leqslant \mathcal{M}(0, \alpha_f^{2-1/2^{f-1}} z^{1/2^f} n^{1-1/2^f}).$$

By the algorithm's description $\mathcal{M}(0,z) \leqslant 2z$, therefore,

$$\mathcal{M}(f,z) \leqslant 2\alpha_f^{2-1/2^{f-1}} z^{1/2^f} n^{1-1/2^f} \leqslant 2\alpha_f^2 z^{1/2^f} n^{1-1/2^f} = \widetilde{O}(f|S|^{1/2^f} n^{1-1/2^f}).$$ ◀

▶ **Lemma 15** (Running Time). *Algorithm 1 can be made to run in $O(fmn)$ time.*

Proof. Rather than explicitly computing $(P_{s,t}, Q_{s,t})$ in the algorithm, we compute the two trees T_1, T_2 once using Theorem 10 in $O(m+n)$. We discard all edges from T_1 and T_2 which don't lie on an s-t path, $\cup_{s\in S} W_s$ can then be computed as $\{u \in V \mid 1 \leqslant \text{dist}(u,t,T_1 \cup T_2) \leqslant 8nf\log n/L\}$ which takes $O(n)$ time. In the final iteration, we simply compute a $(0,1)$-preserver which takes $O(m+n)$ time. We require $f+1$ iterations of this and run it for each node t, hence, the algorithm takes $O(fmn)$ time. ◀

3.2 Subset Distance Preservers

In this subsection, we will provide an efficient construction of subset distance preserver. In particular, we will prove the following result.

▶ **Theorem 16.** *Given an n-vertex graph $G=(V,E)$, a set of source vertices $S \subseteq V$, and a nonnegative integer f, there is an $(f+1)$-fault-tolerant distance preserver of G for all pairs in $S \times S$ on $\widetilde{O}((f+1)n^{2-1/2^f}|S|^{1/2^f})$ edges which can be computed in $O((f+1)mn)$ time.*

Further, the computation time can be reduced to $O(|S|m)$ when $f=0$.

We make use of the shortest path restoration theory by Afek et al. [1] to first prove the following.

▶ **Lemma 17.** *For any undirected graph $G=(V,E)$ and any nonnegative integer f, a source-wise $(f,1)$-preserver H of G with respect to a source set S is an $(f+1)$-fault-tolerant subset distance preserver for pairs in $S \times S$.*

Proof. Let (F,e) be a pair such that $F \subseteq E$, has at most f edges and e is an edge in $E \setminus F$. Applying Theorem 1 on $G \setminus F$, for any $s,t \in S$ and failing edge e there exists a vertex w such that, (i) $\text{dist}(s,w,G\setminus(F\cup e)) = \text{dist}(s,w,G\setminus F)$, (ii) $\text{dist}(w,t,G\setminus(F\cup e)) = \text{dist}(w,t,G\setminus F)$, and (iii) $\text{dist}(s,t,G\setminus(F\cup e)) = \text{dist}(s,w,G\setminus F) + \text{dist}(w,t,G\setminus F)$.

For any $s,t \in S$ and the corresponding node w, $\text{dist}(s,w,H\setminus(F\cup e)) = \text{dist}(s,w,G\setminus F)$ and $\text{dist}(w,t,H\setminus(F\cup e)) = \text{dist}(w,t,G\setminus F)$ by Definition 3, since H is an $(f,1)$-preserver with respect to source S. Therefore,

$$\begin{aligned}\text{dist}(s,t,H\setminus(F\cup e)) &\leqslant \text{dist}(s,w,H\setminus(F\cup e)) + \text{dist}(w,t,H\setminus(F\cup e))\\ &= \text{dist}(s,w,G\setminus F) + \text{dist}(w,t,G\setminus F)\\ &= \text{dist}(s,t,G\setminus(F\cup e)).\end{aligned}$$

However, since H is a subgraph of G, $\text{dist}(s,t,H\setminus(F\cup e)) \geqslant \text{dist}(s,t,G\setminus(F\cup e))$ as well, which implies $\text{dist}(s,t,H\setminus(F\cup e)) = \text{dist}(s,t,G\setminus(F\cup e))$. ◀

As a corollary of Theorem 8, Theorem 9, and Lemma 17, we get the construction in Theorem 16.

4 Other Applications

4.1 Distance Labeling Schemes

In this section, we make use of our $(f,1)$-preserver to provide an alternate construction for distance labels of sub-quadratic size.

▶ **Theorem 18.** *For any fixed nonnegative integer $f \geqslant 0$, and n-vertex unweighted undirected graph, there is an $(f+1)$-fault-tolerant distance labeling scheme that assigns each vertex a label of $O((f+1)n^{2-1/2^f}\log n)$ bits that can be computed in $O(fmn)$ time each when $f \geqslant 1$, and $O(m)$ time when $f = 0$.*

Proof. Let H_s be an $(f,1)$-preserver with respect to source $\{s\}$. At each node s, store H_s as the label. Since the number of edges in H_s is $O((f+1)n^{2-1/2^f})$ and it takes $O(\log n)$ bits to describe an edge, each label takes $O((f+1)n^{2-1/2^f}\log n)$ bits.

To compute $\mathrm{dist}(s,t,G\setminus F)$ for any $|F| \leqslant f+1$, we read the labels of s and t to determine H_s and H_t, then union them together to get $H_{st} = H_s \cup H_t$. We then simply find the distance in $H_{st}\setminus F$.

H_{st} is a $(f,1)$-preserver with respect to the source $\{s,t\}$. By Lemma 17, H_{st} is an $(f+1)$-fault-tolerant distance-preserver for pairs in $\{s,t\}\times\{s,t\}$. Thus, $\mathrm{dist}(s,t,H_{st}\setminus F) = \mathrm{dist}(s,t,G\setminus F)$ as desired. ◀

4.2 Subset Replacement Path Algorithm

In the Subset Replacement Path problem (Subset-RP), the input is a graph $G=(V,E)$ and a set of source vertices S, and for every pair of vertices $s,t\in S$ and failing edge $e\in E$, report $\mathrm{dist}(s,t,G\setminus e)$.

We modify the algorithm by Bodwin et al. [5] to use our $(0,1)$-preserver from Theorem 9 to solve the Subset-RP problem.

▶ **Theorem 19.** *Given a graph $G=(V,E)$ and a set of source vertices S, we can solve the Subset-RP problem in $O(|S|m)+\widetilde{O}(|S|^2 n)$ time in the word-RAM model.*

To this end, we will make use of the following result whose proof we will omit as we only make a black box use of it.

▶ **Theorem 20** (Hershberger and Suri [12]). *When $|S|=2$, there is an algorithm that solve Subset-RP(G,S) in time $\widetilde{O}(m+n)$.*

We now propose Algorithm 2 to solve the general problem.

Algorithm 2 Algorithm for Subset-RP(G,S).

for $s\in S$ **do**
  Compute $H_s \leftarrow (0,1)$-preserver of G with source s.
end
for $s,t\in S$ **do**
  Solve Subset-RP$(H_s\cup H_t,\{s,t\})$ using Theorem 20 to get the result for (s,t).
end

▶ **Lemma 21** (Correctness). *Algorithm 2 solves the Subset-RP(G,S) problem.*

Proof. $H_s \cup H_t$ is a $(0,1)$-preserver with respect to the source set $\{s,t\}$. By Lemma 17, $H_s \cup H_t$ is a 1-fault-tolerant distance-preserver for pairs in $\{s,t\} \times \{s,t\}$. Thus for any edge e, $\mathrm{dist}(s,t,(H_s \cup H_t) \setminus e) = \mathrm{dist}(s,t,G \setminus e)$. Thus, applying Theorem 20 on $H_s \cup H_t$ correctly computes $\mathrm{dist}(s,t,G \setminus e)$ for all edges e. ◀

▶ **Lemma 22** (Running Time). *Algorithm 2 runs in* $O(|S|m) + \widetilde{O}(|S|^2 n)$ *time.*

Proof. By Theorem 10, we can compute H_s in $O(m+n)$ time. Since H_s has $O(n)$ size, Subset-RP$(H_s \cup H_t, \{s,t\})$ can be solved in $\widetilde{O}(n)$ time by Theorem 20. Therefore, the total time taken is $O(|S|m) + \widetilde{O}(|S|^2 n)$. ◀

References

1 Yehuda Afek, Anat Bremler-Barr, Haim Kaplan, Edith Cohen, and Michael Merritt. Restoration by path concatenation: fast recovery of MPLS paths. *Distributed Computing*, 15(4):273–283, 2002. doi:10.1007/S00446-002-0080-6.

2 Davide Bilò, Shiri Chechik, Keerti Choudhary, Sarel Cohen, Tobias Friedrich, Simon Krogmann, and Martin Schirneck. Approximate Distance Sensitivity Oracles in Subquadratic Space. In *Proceedings of the 55th Symposium on Theory of Computing (STOC)*, pages 1396–1409, 2023. doi:10.1145/3564246.3585251.

3 Greg Bodwin, Keerti Choudhary, Merav Parter, and Noa Shahar. New fault tolerant subset preservers. In Artur Czumaj, Anuj Dawar, and Emanuela Merelli, editors, *47th International Colloquium on Automata, Languages, and Programming, ICALP 2020, July 8-11, 2020, Saarbrücken, Germany (Virtual Conference)*, volume 168 of *LIPIcs*, pages 15:1–15:19. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2020. doi:10.4230/LIPICS.ICALP.2020.15.

4 Greg Bodwin, Fabrizio Grandoni, Merav Parter, and Virginia Vassilevska Williams. Preserving distances in very faulty graphs. In *44th International Colloquium on Automata, Languages, and Programming, ICALP 2017, July 10-14, 2017, Warsaw, Poland*, pages 73:1–73:14, 2017. doi:10.4230/LIPICS.ICALP.2017.73.

5 Greg Bodwin and Merav Parter. Restorable shortest path tiebreaking for edge-faulty graphs. *J. ACM*, 70(5), October 2023. doi:10.1145/3603542.

6 Shiri Chechik and Sarel Cohen. Near optimal algorithms for the single source replacement paths problem. In Timothy M. Chan, editor, *Proceedings of the Thirtieth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2019, San Diego, California, USA, January 6-9, 2019*, pages 2090–2109. SIAM, 2019. doi:10.1137/1.9781611975482.126.

7 Shiri Chechik, Sarel Cohen, Amos Fiat, and Haim Kaplan. $(1+\epsilon)$-approximate f-sensitive distance oracles. In Philip N. Klein, editor, *Proceedings of the Twenty-Eighth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2017, Barcelona, Spain, Hotel Porta Fira, January 16-19*, pages 1479–1496. SIAM, 2017.

8 Dipan Dey and Manoj Gupta. Nearly optimal fault tolerant distance oracle. In Bojan Mohar, Igor Shinkar, and Ryan O'Donnell, editors, *Proceedings of the 56th Annual ACM Symposium on Theory of Computing, STOC 2024, Vancouver, BC, Canada, June 24-28, 2024*, pages 944–955. ACM, 2024. doi:10.1145/3618260.3649697.

9 Ran Duan and Hanlin Ren. Maintaining exact distances under multiple edge failures. In Stefano Leonardi and Anupam Gupta, editors, *STOC '22: 54th Annual ACM SIGACT Symposium on Theory of Computing, Rome, Italy, June 20–24, 2022*, pages 1093–1101. ACM, 2022. doi:10.1145/3519935.3520002.

10 Loukas Georgiadis and Robert Endre Tarjan. Dominators, directed bipolar orders, and independent spanning trees. In *Automata, Languages, and Programming – 39th International Colloquium, ICALP 2012, Warwick, UK, July 9-13, 2012, Proceedings, Part I*, pages 375–386, 2012. doi:10.1007/978-3-642-31594-7_32.

11 Manoj Gupta and Shahbaz Khan. Multiple source dual fault tolerant BFS trees. In *44th International Colloquium on Automata, Languages, and Programming, ICALP 2017, July 10-14, 2017, Warsaw, Poland*, pages 127:1–127:15, 2017. doi:10.4230/LIPIcs.ICALP.2017.127.

12 J. Hershberger and S. Suri. Vickrey prices and shortest paths: what is an edge worth? In *Proceedings 42nd IEEE Symposium on Foundations of Computer Science*, pages 252–259, 2001. doi:10.1109/SFCS.2001.959899.

13 Ketan Mulmuley, Umesh V. Vazirani, and Vijay V. Vazirani. Matching is as easy as matrix inversion. In Alfred V. Aho, editor, *Proceedings of the 19th Annual ACM Symposium on Theory of Computing, 1987, New York, New York, USA*, pages 345–354. ACM, 1987. doi:10.1145/28395.383347.

14 Merav Parter. Dual failure resilient BFS structure. *arXiv*, 2015. arXiv:1505.00692.

15 Merav Parter and David Peleg. Sparse fault-tolerant BFS trees. In *Algorithms – ESA 2013 – 21st Annual European Symposium, Sophia Antipolis, France, September 2-4, 2013. Proceedings*, pages 779–790, 2013. doi:10.1007/978-3-642-40450-4_66.