Improved Shortest Path Restoration Lemmas for Multiple Edge Failures: Trade-offs Between Fault-tolerance and Subpaths

Greg Bodwin and Lily Wang
University of Michigan EECS
{bodwin, lilyxy}@umich.edu

Abstract

The *restoration lemma* is a classic result by Afek, Bremler-Barr, Kaplan, Cohen, and Merritt [PODC '01], which describes how the structure of shortest paths in a graph can change when some edges in the graph fail. Their work shows that, after one edge failure, any replacement shortest path avoiding this failing edge can be partitioned into two pre-failure shortest paths. More generally, this implies an *additive* tradeoff between fault tolerance and subpath count: for any f, k, we can partition any f-edge-failure replacement shortest path into $k+1$ subpaths which are each an $(f-k)$-edge-failure replacement shortest path. This generalized version of the result has found applications in routing, graph algorithms, fault tolerant network design, and more.

Our main result improves this to a *multiplicative* tradeoff between fault tolerance and subpath count. We show that for all f, k, any f-edge-failure replacement path can be partitioned into $O(k)$ subpaths that are each an (f/k)-edge-failure replacement path. We also show an asymptotically matching lower bound. In particular, our results imply that the original restoration lemma is exactly tight in the case $k = 1$, but can be significantly improved for larger k. We also show an extension of this result to weighted input graphs, and we give an efficient algorithms that computes path decompositions satisfying our improved restoration lemmas, which runs in near-linear time for fixed f.

1 Introduction

Suppose we want to route information, traffic, goods, or anything else along shortest paths in a distributed network. In practice, network edges can be prone to *failures*, in which a link is temporarily unusable as it awaits repair. It is therefore desirable for a system to be able to adapt to these failures, efficiently rerouting paths on the fly into new replacement paths that avoid the currently-failing edges. An algorithm that repairs a shortest path routing table following one or more edge failures is called a *restoration algorithm* [ABBK$^+$02]; the design of effective restoration algorithms forms a large and active body of work in both theory and practice, see e.g., [LYXsS20, ZGK$^+$21, WGY$^+$15, ABBK$^+$02, BP21] and references within. The focus of this paper will specifically be on restoration algorithms that recover *exact* shortest paths in the post-failure graph.

An ideal restoration algorithm will avoid recomputing shortest paths from scratch after each new failure event, instead leveraging its knowledge of the pre-failure shortest paths to speed up computation. Therefore, when designing restoration algorithms, it is often helpful to understand exactly how shortest paths in a graph can evolve following edge failures. A *restoration lemma* is the general name for a structural result relating the form of pre-failure shortest paths to post-failure shortest paths in a graph, named for their applications in restoration algorithms.

The original restoration lemma was made explicit in a classic paper by Afek, Bremler-Barr, Kaplan, Cohen, and Meritt [ABBK$^+$02], and it was implicit in work before that, such as [KIM82]. All graphs in this discussion are undirected and unweighted, until otherwise indicated.

Definition 1 (Replacement Paths)**.** A path π in a graph $G = (V, E)$ is an *f-fault replacement path* if there exists a set of edges $F \subseteq E, |F| \leq f$ such that π is a shortest path in the graph $G \setminus F$.

Theorem 2 (Original Restoration Lemma [ABBK$^+$02])**.** *In any graph G, every f-fault replacement path can be partitioned into $f+1$ subpaths that are each a shortest path in G.*

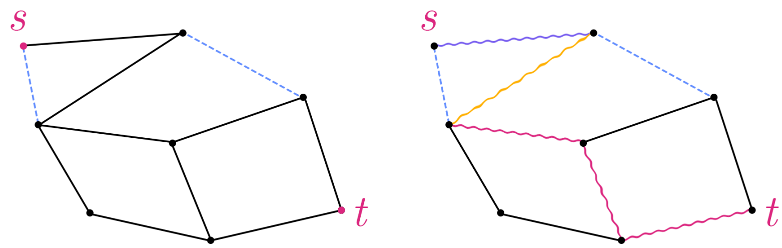

Figure 1: **(Left)** Suppose that two blue dashed edges in the graph fail. **(Right)** The restoration lemma guarantees that any shortest $s \rightsquigarrow t$ path (indicated with wavy edges) in the post-failure network can be partitioned into three subpaths, which are each a shortest path in the pre-failure network.

This restoration lemma suggests a natural approach for restoration algorithms: when f edges fail and an $s \rightsquigarrow t$ shortest path is no longer usable, we can find a replacement $s \rightsquigarrow t$ shortest path by searching only over $s \rightsquigarrow t$ paths that can be formed by concatenating $f+1$ shortest paths that we have already computed in the current routing table. Up to some subtleties involving shortest path tiebreaking [BP21], this approach works, and has been experimentally validated as an efficient restoration strategy [ABBK$^+$02]. It has also found widespread theoretical application, e.g., in pricing algorithms [HS01], replacement path algorithms [MR22, BP21, BCG$^+$18, MMG89, CC19, GJM20], fault-tolerant variants of spanner problems [BP21, BGPW17, CCFK17] and more. Most recently, in [DR22], Duan and Ren use the restoration lemma to guarantee low recursion depth in their multi-fault-tolerant exact distance oracle, the first to have query time dependent only on the number of failures rather than graph size.

1.1 The Fault Tolerance/Subpath Count Tradeoff and First Main Result

The main directions for future work left by Afek et al. concern whether one can obtain improved restoration lemmas, by allowing one to restore using larger or more carefully-chosen *base sets* of paths, rather than just shortest paths in the input graph (c.f. [ABBK$^+$02], p.277). Perhaps the most natural candidate would be to construct f-fault replacement paths by concatenating several ($f' < f$) replacement paths. Indeed, most applications of the restoration lemma in the previous literature are actually based on a concatenation method of this type. They use the following generalization of Theorem 2, which does not appear explicitly in [ABBK$^+$02] but which follows easily from the main result:

Corollary 3 ([ABBK$^+$02])**.** *For any graph G and any $1 \leq k \leq f$, every f-fault replacement path can be partitioned into at most $k+1$ subpaths that are each $(f-k)$-fault replacement paths in G.*

Proof of Corollary 3, given Theorem 2. Let π be a replacement shortest path in a graph G avoiding a set of edge failures $F = \{e_1, \dots, e_f\}$. Consider the graph $G' := G \setminus \{e_1, \dots, e_{f-k}\}$. In G', π is a k-fault replacement shortest path avoiding the remaining edge failures $\{e_{f-k+1}, \dots, e_f\}$. Thus, applying Theorem 2 in G', we can partition π into $k+1$ subpaths that are each a shortest path in G'. Each of these subpaths is an $(f-k)$-replacement shortest path, avoiding $\{e_1, \dots, e_{f-k}\}$, in the original graph G. □

In other words, this corollary states that if we take our base set to be all $(f-k)$-fault replacement paths, then we can improve the number of subpaths needed for the decomposition to $k+1$. However, at a technical level this only requires a very minor extension of the original restoration lemma, and it is not clear whether this tradeoff is optimal. The main contribution of this work is to show new asymptotically-matching upper and lower bounds, establishing that the correct tradeoff is actually **multiplicative**:

Theorem 4 (Main Result)**.** *The following hold for all positive integers k, f with $k \leq f$:*

- ***(Upper Bound)*** *In any graph G, every f-fault replacement path can be partitioned into at most $O(k)$ subpaths that are each a replacement path in G avoiding at most f/k faults.*
- ***(Lower Bound)*** *There are graphs G and f-fault replacement paths π that cannot be partitioned into $2k$ subpaths that are each an $(\lfloor f/k \rfloor - 2)$-fault replacement path in G.*

The specific upper bound we show is $8k+1$ subpaths, although in this paper we do not focus on optimizing the leading constant. We view Theorem 4 as a mixed bag, containing both good news and bad news for the area. The good news is that the restoration lemma tradeoff has been substantially improved, and this potentially opens up new avenues for restoration algorithms for routing table recovery and its applications. The bad news is that, in the important special case $k = 1$ (i.e., decomposing into only two subpaths), our new lower bound shows that the previous restoration lemma was tight: there are examples in which one cannot decompose an f-fault replacement path into two replacement paths avoiding $f - 2$ faults each (previously, a tradeoff of roughly $f/2$ was conceivable). This case $k = 1$ is particularly important in applications, especially to spanner and preserver problems [BP21, BGPW17], and so this lower bound may close a promising avenue for progress on these applications.

1.2 Weighted Restoration Lemmas and Second Main Result

The original paper by Afek et al. [ABBK$^+$02] also proved a *weighted* restoration lemma, which gives a weaker decomposition, but which holds also for weighted input graphs:

Theorem 5 (Weighted Restoration Lemma [ABBK$^+$02])**.** *For any* ***weighted*** *graph G and any $1 \leq k \leq f$, every f-fault replacement path π can be partitioned into $k + 1$ subpaths and k individual edges, where each subpath in the partition is an $(f - k)$-fault replacement path in G.*

More specifically, this theorem promises that the subpaths and individual edges occur in an alternating pattern (although some of these subpaths in this pattern may be empty). One can again ask whether this additive tradeoff between subpath count and fault tolerance per subpath is optimal. We show that it is not, and that it can be improved to a multiplicative tradeoff, similar to Theorem 4.

Theorem 6 (Main Result, Weighted Setting)**.** *For any* ***weighted*** *graph G and any $1 \leq k \leq f$, every f-fault replacement path π can be partitioned into $O(k)$ subpaths and $O(k)$ individual edges, where each subpath in the partition is an (f/k)-fault replacement path in G.*

For many graphs of interest, this theorem can be simplified. For example, suppose we consider the setting of graphs that represent *metrics*, in which we require that every edge is the shortest path between its endpoints. Then we can consider the $O(k)$ individual edges in the decomposition to each be a 0-fault replacement path, and so we could correctly state that π can be partitioned into $O(k)$ subpaths that are each at most (f/k)-fault replacement paths. However, we also note that our weighted main result cannot be simplified *in general*: one can easily construct weighted graphs containing edges (u, v) that are f-fault but not $(f - 1)$-fault replacement paths between their endpoints, and therefore any weighted restoration lemma will need to include some exceptional edges in its hypotheses, as in [ABBK$^+$02] and Theorem 6.

1.3 Algorithmic Considerations

The proofs of our new restoration lemmas (both weighted and unweighted) are given using a simple greedy decomposition strategy to find subpaths; essentially, we repeatedly peel off the longest possible prefix from the input path π that is an f/k-fault replacement path, and then argue that this process will repeat at most $O(k)$ times. Although this leads to a simpler proof, a downside of this greedy procedure is that it requires exponential time in the number of faults f. That is, given a subpath $\pi_i \subseteq \pi$, it is not clear how to test whether π_i is an f/k-fault replacement path, besides via brute force search over every subset of faults $F' \subseteq F, |F'| \leq f/k$ which takes $\text{poly}(n) \cdot \exp(f)$ time. To obtain algorithmic results, we thus need to change the decomposition strategy in Section 5 entirely. We show a more involved algorithm that implements our restoration lemmas in $\text{poly}(n, f)$ time. That is:

Theorem 7 (Unweighted Algorithmic Restoration Lemma)**.** *There is an algorithm that take on input a graph G, a set F of $|F| = f$ edge faults, a shortest path π in $G \setminus F$, and a parameter k, and which returns:*

- *A partition* $\pi = \pi_0 \circ \pi_1 \circ \cdots \circ \pi_q$ *into* $q = O(k)$ *subpaths, and*
- *Fault sets* $F_0, \dots, F_q \subseteq F$ *with each* $|F_i| \leq f/k$*, such that each path* π_i *in the decomposition is a shortest path in* $G \setminus F_i$

(hence the algorithm implements Theorem 4). This algorithm runs in $O(m \cdot \mathit{poly}(\log n, f))$ *time, where* $m := |E(G)|$*.*

The core of of our new decomposition approach is a reduction to the algorithmic version of Hall's theorem; this is somewhat involved, and so we overview it in more depth in the next part of this introduction. Using roughly the same algorithm, we also show the algorithmic restoration lemma in the weighted setting.

Theorem 8 (Weighted Algorithmic Restoration Lemma)**.** *There is an algorithm that take on input a weighted graph* G*, a set* F *of* $|F| = f$ *edge faults, a shortest path* π *in* $G \setminus F$*, and a parameter* k*, and which returns:*

- *A partition* $\pi = \pi_0 \circ e_0 \circ \cdots \circ \pi_{q-1} \circ e_{q-1} \circ \pi_q$*, where each* π_i *is a (possibly empty) subpath, each* e_i *is a single edge, and* $q = O(k)$*, and*
- *Fault sets* $F_0, \dots, F_q \subseteq F$ *with each* $|F_i| \leq f/k$*, such that each path* π_i *in the decomposition is a shortest path in* $G \setminus F_i$

(hence the algorithm implements Theorem 6). This algorithm runs in $O(m \cdot \mathit{poly}(\log n, f))$ *time, where* $m := |E(G)|$*.*

We remark that f is often considered to be at most polylog n in fault-tolerant exact preserver problems, since the bounds given in most prior work become trivial for larger f; in this regime our algorithms are near-linear time.

1.4 Technical Overview of Upper Bounds

The more involved parts of the paper are the upper bounds in Theorem 4 and 6. We will overview the proof in the unweighted setting (Theorem 4) here; the weighted setting carries only a few additional technical details.

Let π be an f-fault replacement path with endpoints (s, t) in an input graph G. In particular, let F be a set of $|F| \leq f$ edge faults, and suppose that π is a shortest $s \rightsquigarrow t$ path in the graph $G \setminus F$. We are also given a parameter $f' < f$, and our goal is to partition π into as few subpaths as possible, subject to the constraint that each subpath is a replacement path avoiding at most f' faults.

The Partition of π**.** We use a simple greedy process to determine the partition of π. We will determine a sequence of nodes $(s = x_0, x_1, \dots, x_k, x_{k+1} = t)$ along π, which form the boundaries between subpaths in the decomposition. Start with $s =: x_0$, and given node x_i, define x_{i+1} to be the furthest node following x_i such that the subpath $\pi[x_i, x_{i+1}]$ is an f'-fault replacement path. We will denote the subpath $\pi[x_i, x_{i+1}]$ as π_i, and so the decomposition is

$$\pi = \pi_0 \circ \cdots \circ \pi_k.$$

For each subpath we let $F_i \subseteq F, |F_i| \leq f'$ be an edge set of minimum size such that π_i is a shortest $x_i \rightsquigarrow x_{i+1}$ path in the graph $G \setminus F_i$. (There may be several choices for F_i, in which case we fix one arbitrarily.) Our goal is now to show that the parameter k, defined as (one fewer than) the number of subpaths that arise from the greedy decomposition, satisfies $kf' \leq O(f)$.

Proof Under Simplifying Assumptions. Our proof strategy will be to prove that an arbitrary faulty edge $e \in F$ can appear in only a constant number of subpath fault sets F_i, which implies that $kf' \leq O(f)$ by straightforward counting. To build intuition, let us see how the proof works under two rather strong simplifying assumptions:

- **(Equal Subpath Assumption)** We will assume that all subpaths in the decomposition have equal length: $|\pi_0| = \cdots = |\pi_k|$.

- **(First Fault Assumption)** Let us say that a *shortcut* for a subpath π_i is an alternate $x_i \rightsquigarrow x_{i+1}$ path in the original graph G that is strictly shorter than π_i. Every shortcut must contain at least one fault in F_i, and conversely, every fault in F_i lies on at least one shortcut (or else it may be dropped from F_i). Our second simplifying assumption is that for each $e \in F_i$, there exists a shortcut σ for π_i such that e is the *first* fault in F_i on σ.

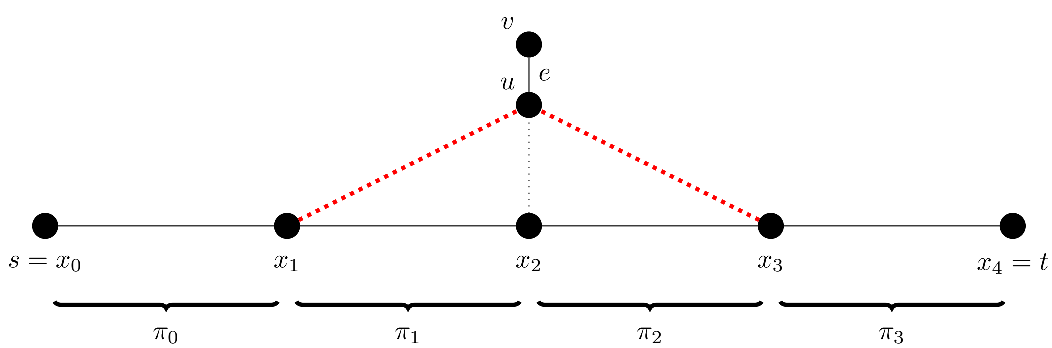

Figure 2: Under the equal subpath and first fault assumptions, we can reach contradiction if we assume that there are three different subpaths that all have shortcuts that use e as their first fault.

With these two assumptions in hand, we are ready to prove that each faulty edge e appears in only $O(1)$ many fault sets F_i. Suppose for contradiction that there are three separate subpaths that all have shortcuts that use e as their first edge, and moreover that these shortcuts use e with the same orientation (in Figure 2, the subpaths are π_1, π_2, π_3, and the shortcut prefixes are represented by the three dotted lines from x_1, x_2, x_3 to u). Consider the first and last of these shortcut prefixes, which we will denote as $q(x_1, u)$ and $q(x_3, u)$ (in Figure 2, $q(x_1, u), q(x_3, u)$ are colored red). Notice that $q(x_1, u) \cup q(x_3, u)$ forms an alternate $x_1 \rightsquigarrow x_3$ path. Since e is assumed to be the first fault on these shortcuts, this alternate $x_1 \rightsquigarrow x_3$ path avoids all faults in F. Additionally, by definition of shortcuts we have

$$|q(x_1, u)| + |q(x_3, u)| < |\pi_1| + |\pi_3|.$$

Since we have assumed that all subpaths have the same length, we can amend this to

$$|q(x_1, u)| + |q(x_3, u)| < |\pi_1| + |\pi_2|.$$

But this implies that $q(x_1, u) \cup q(x_3, u)$ forms an $x_1 \rightsquigarrow x_3$ path that is strictly shorter than the one used by π, which contradicts that π is a shortest path in $G \setminus F$. This completes the simplified proof, but the challenge is now to relax our two simplifying assumptions, which are currently doing a lot of work in the argument.

Relaxing the Equal-Subpath-Length Assumption. The equal-subpath-length assumption is the easier of the two to relax. It is only used in one place in the previous proof: to replace $|\pi_3|$ with $|\pi_2|$ on the right-hand side of the inequality. From this we see that $|\pi_2| \geq |\pi_3|$ is a good case for the argument, since this substitution remains valid. The bad case is when $|\pi_2| < |\pi_3|$.

To handle this bad case, we follow a proof strategy from [ABDS$^+$20]. Let us say that a subpath is *pre-light* if it is no longer than the preceding subpath, or *post-light* if it is no longer than the following subpath. In the above example, π_2 is pre-light if we have $|\pi_2| \leq |\pi_1|$, and it is post-light if $|\pi_2| \leq |\pi_3|$. It is possible for a particular subpath to be both pre- and post-light, or for a particular subpath to be neither. A simple counting argument shows that either a constant fraction (nearly half) of the subpaths are pre-light, or a constant fraction are post-light. We will specifically assume in the following discussion that a constant fraction of the subpaths are post-light; the other case is symmetric.

We can now restrict the previous counting argument to the post-light subpaths only. That is, we can argue that for each fault $e = (u, v)$ considered with orientation, there are only constantly many post-light subpaths for which it appears as the first fault of a shortcut. The same counting argument then implies an

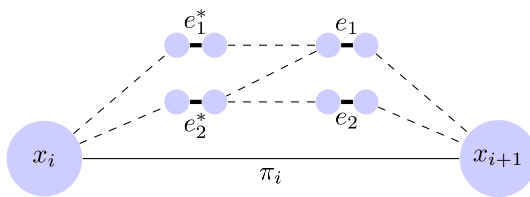

Figure 3: In order to relax the first fault assumption, instead of counting (e_1, π_i) and (e_2, π_i) as pairs, we can map these to distinct FS pairs $(e_1^*, \pi_i), (e_2^*, \pi_i)$. Our main technical step is to show that this distinct mapping is always possible.

upper bound of $|F_i| \leq O(f/k)$ for the fault sets F_i associated to post-light subpaths π_i, which completes the proof.

This still uses the first-fault assumption, and we next explain how this can be relaxed. We consider the machinery used to relax the first-fault assumption to be the main technical contribution of this paper.

Relaxing the First-Fault Assumption. Let us now consider the case where there is a fault $e \in F_i$ that is *not* the first fault of any shortcut for π_i. We can still assume that there exists at least one shortcut σ for π_i with $e \in \sigma$ (otherwise, we can safely drop e from F_i). Let e^* be the first fault along that shortcut σ. We will shift the focus of our counting argument. Previously, we considered each $(e \in F_i, \pi_i)$ as a pair, and our goal was to argue that faults e can only be paired with a constant number of subpaths π_i. Now, our strategy is to map the pair $(e \in F_i, \pi_i)$ to the different pair (e^*, π_i), and our goal is to argue that each fault e^* can only be paired with a constant number of subpaths π_i. We call these new pairs (e^*, π_i) *Fault-Subpath (FS) Pairs*, and we formally describe their generation in Section 3.2. (We note that, for a technical reason, we actually generate FS pairs using *augmented subpaths* that attach one additional node to π_i - but to communicate intuition about our proof, we will ignore this detail for now.)

Although we can bound the number of FS pairs (e^*, π_i) as before, this only implies our desired bound on the size of the fault sets $|F_i|$ if we can *injectively* map each pair $(e \in F_i, \pi_i)$ to a *distinct* FS pair (e^*, π_i). The main technical step in this part of the proof is to show that this injective mapping is always possible. Let Γ_i be a bipartite graph between vertex sets F_i and F. Put an edge between nodes $e \in F_i, e^* \in F$ if and only if there exists a shortcut σ for π_i, in which $e \in \sigma$ and e^* is the first fault in σ. An injective mapping to FS pairs corresponds to a matching in Γ_i of size $|F_i|$, i.e., a matching of maximum possible size (given that one of the sides of the bipartition has only $|F_i|$ nodes). The purpose of this graph construction is to enable the following application of Hall's theorem:

Lemma 9 (Hall's Theorem)**.** *The following are equivalent:*

- *The graph Γ_i has a matching of size $|F_i|$. (Equivalently, one can associate each pair $(e \in F_i, \pi_i)$ to a **unique** FS pair.)*
- *There does not exist a subset of faults $F_i' \subseteq F_i$ whose neighborhood in Γ_i is strictly smaller than F_i' itself (that is, $|N(F_i')| < |F_i'|$).*

We show that the latter property is implied by minimality of F_i, which means the former property is true as well. See Lemma 16 and surrounding discussion for details.

2 Preliminaries

Definition 10. Relative to a value of f, we call the pair (q, r) **restorable** if in every graph G, any f-fault replacement path can be partitioned into q subpaths which are each r-fault replacement subpaths in G.

Remark 11. (Monotonicity of Restorability) If (q, r) is restorable, then both $(q + 1, r)$ and $(q, r + 1)$ are restorable. Equivalently, if (q, r) is not restorable, neither $(q - 1, r)$ nor $(q, r - 1)$ are restorable.

For notation, we will commonly write F for a set of f failing edges, and s, t for the endpoint nodes of the replacement path under consideration, which is then written $\pi(s, t \mid F)$. To denote a subpath between intermeidate nodes u and v, we write $\pi(s, t \mid F)[u, v]$. We will denote the q r-fault replacement subpaths as $\pi(x_i, x_{i+1} \mid F_{i+1})$ with $x_0 := s$ and $x_q := t$, and each fault subset $|F_i| \leq r$. Then a decomposition of $\pi(s, t \mid F)$ satisfying restorability can be written

$$\pi(s, t \mid F) = \pi(x_0, x_1 \mid F_1) \circ \pi(x_1, x_2 \mid F_2) \circ \ldots \circ \pi(x_{q-1}, x_q \mid F_q).$$

Equivalently, for each i,

$$\pi(s, t \mid F)[x_i, x_{i+1}] = \pi(x_i, x_{i+1} \mid F_{i+1}).$$

3 Upper Bound

We now prove Theorem 4. Fix any s, t and replacement path $\pi(s, t \mid F)$, where $|F| =: f$. Recall that, to prove Theorem 4, our goal is to show that for any $k \in \mathbb{R}$, we can partition $\pi(s, t \mid F)$ into $O(k)$ (f/k)-fault replacement subpaths.

3.1 Subpath Generation

We generate $\{x_i\}$, the vertices used to split $\pi(s, t \mid F)$, by traversing along $\pi(s, t \mid F)$ and adding vertices greedily to the current subpath until adding one more vertex would make that subpath no longer an f/k-fault replacement path. More precisely, we set $x_0 := s$, and then we pick each x_{i+1} to maximize $|\pi(s, t \mid F)[x_i, x_{i+1}]|$ under the constraint that $\pi(s, t \mid F)[x_i, x_{i+1}]$ is an f/k-fault replacement subpath. We define x_i' to be the vertex immediately after x_i along $\pi(s, t \mid F)$, i.e., the vertex satisfying

$$|\pi(s, t \mid F)[x_i, x_i']| = 1.$$

Let q be the number of subpaths generated by this process, and so we have $\{x_1, \ldots, x_q\}$. We will denote the q corresponding subpaths of $\pi(s, t \mid F)$ by

$$\pi_i = \pi(s, t \mid F)[x_i, x_{i+1}] \quad \forall 0 \leq i \leq q-1.$$

Here we remark that the last subpath π_{q-1} differs from the others as it is bounded by the end of $\pi(s, t \mid F)$ and is not generated greedily.

Definition 12 (Augmented Subpaths)**.** For $0 \leq i \leq q-2$, We define **augmented subpaths** π_i' as

$$\pi_i' := \pi(s, t \mid F)[x_i, x_{i+1}'].$$

Note that no augmented subpath π_i' can be an f/k-fault replacement subpath in G by greedy choice of x_i.

For each i, fix $F_i \subseteq F$ to be any minimum size fault set such that π_i' is a shortest path in $G \setminus F_i$. That is, π_i' is an $|F_i|$-fault replacement path but not a c-fault replacement path for any $c < |F_i|$. By our choice of x_i, we must have

$$|F_i| > f/k.$$

Following the notation of [ABDS$^+$20], we will denote π_i' as **pre-light** if its length is less than or equal to the length of π_{i-1}', and **post-light** if its length is less than or equal to the length of π_{i+1}'. From Lemma 3 of [ABDS$^+$20], at least half of the π_i' are pre-light, or at least half are post-light. We will assume without loss of generality that for at least $\frac{q-1}{2}$ i, π_i' is post-light. The other case, where at least half of the π_i' are pre-light, follows from a symmetric argument.[1]

[1] In particular, in the case where at least half of the π_i' are pre-light, one can use the following argument but substitute "left ends" for "right ends", and "(left) FS-pairs" for "(right) FS-pairs".

3.2 FS-pair Generation

To bound the number of subpaths, we will use a counting argument tracking pairs of augmented subpaths and faults which are "close" to each other. In particular, any subpath-fault pair has a fixed distance to each other within the fault-free graph, and only a limited number of subpaths can have small distance to a specified fault without creating a shorter path than $\pi(s, t \mid F)$. We will formalize these ideas using FS-pairs in this section.

Definition 13. For any (u, v)-path p', we say a (u, v)-path p is a **shortcut** of p' if $\text{len}(p) < \text{len}(p')$.[2]

Definition 14. A (right) FS-pair is a pair (e^*, π'_i) with $e^* \in F$, π'_i a post-light augmented path, and the property that there exists a fault-free path from x'_{i+1} to e^* which is contained in a shortcut of π'_i.

We assume above that most subpaths are post-light. In the other case when most subpaths are pre-light, we would instead define left FS-pairs, where we require π'_i to be pre-light and the fault-free path from x_i to e^*, and the argument would be handled symmetrically.

We will now describe the generation of right FS-pairs, starting by setting up some notation. Fix any post-light augmented subpath π'_i. For a given $e \in F_i$ which we refer to as the **generating fault**, let S_e be the set of (x'_{i+1}, x_i)-shortcuts of π'_i[3] which contain e.[4] For a shortcut $p \in S_e$, we define the **base fault** $b(p) \in F$ of p as the first fault in p traversing from x'_{i+1} to x_i. More precisely, we define

$$b(p) := e_{\min\{j : e_j \in F\}} \text{ where } p = x'_{i+1} e_1 v_1 e_2 \dots e_m x_i.$$

For each $e \in F_i$, we define its set of base faults for π'_i as

$$B(e) := \{b(p) : p \in S_e\}$$

Finally, we define the family of base faults for π'_i as

$$\mathcal{B}(\pi'_i) = \{B(e) : e \in F_i\}$$

Our next goal will be to choose a distinct base fault from each $B(e) \in \mathcal{B}(\pi'_i)$ in order to construct at least $|F_i|$ FS-pairs with π'_i.

Explicitly, we define an auxiliary bipartite graph Γ_i where one side of the bipartition is F_i and the other is the set of faults F. For $e^* \in F$ and $e \in F_i$, we have $(e, e^*) \in E(\Gamma_i)$ if and only if $e^* \in B(e)$. In this set up, choosing a distinct base fault for each $B(e)$ is equivalent to finding a matching of Γ_i which saturates F_i. Hall's Theorem gives us a condition for this:

Lemma 15 (Hall's Condition)**.** *If for every $A \subseteq F_i$ we have $|N(A)| \geq |A|$, where $N(A)$ is the neighborhood of A in Γ_i, then Γ_i contains a matching that saturates F_i (and therefore it is possible to choose a distinct base fault for each $B(e)$).*

We therefore only need to verify the premises of Hall's Condition. The following lemma will be helpful.

Lemma 16. *For any $A \subseteq F_i$, the fault set $F'_i := (F_i \setminus A) \cup N(A)$ is also a valid fault set for π'_i. (That is, π'_i is a shortest path in $G \setminus F'_i$.)*

Proof. First, we observe that

$$N(A) = \left| \bigcup_{e \in A} B(e) \right|.$$

This holds because the neighbours of each generating fault e is the set of base faults $B(e)$ it generates.

We now need to prove that no shortcuts for π'_i survive in $G \setminus F'_i$. Let p be an arbitrary shortcut for π'_i in G. Then it must contain some fault $e' \in F_i$, since F_i is a valid fault set. There are two cases:

[2]We include the possibility of non-simple shortcuts, which may repeat nodes and be walks. Our existential upper bound proof would work equally well if we restricted attention to *simple* shortcuts, but this expanded definition will be more convenient for algorithmic reasons outlined in Section 5.

[3]We consider π_i a (x'_{i+1}, x_i)-path in this section.

[4]Note that S_e depends on the choice of subpath π'_i, although we do not include this parameter in the notation.

- If $e' \in F'_i$, then the shortcut p does not survive in $G \setminus F'_i$.
- Otherwise, suppose that $e' \notin F'_i$, and so in particular $e' \in A$. In this case, p's base fault $b(p)$ is in $B(e') \subseteq F'_i$, and thus not in $G \setminus F'_i$.

Therefore there are no surviving shortcuts for π'_i in $G \setminus F'_i$. □

Notice that Lemma 15 follows from Lemma 16: since we assume that F_i is a *minimal* fault set, we must have that $|N(A)| \geq A$ for all $A \subseteq F_i$, since otherwise we would have $|F'_i| < |F_i|$. Since Hall's condition holds, over any augmented subpath π'_i, we can assign a unique base fault to every generating fault. Accordingly, we can define an injective function $\phi_i : F_i \to F$ where $\phi_i(e) \in B(e)$.

We will construct our FS-pairs for π'_i as $\{(\phi_i(e), \pi'_i) \mid e \in F_i\}$, and since ϕ_i is injective, we get $|F_i|$ distinct FS-pairs from π'_i. We repeat this process for every post-light augmented subpath. It follows that we will generate at least $(q-1)f/(2k)$ FS-pairs, since we have $(q-1)/2$ post-light augmented subpaths which have corresponding fault sets F_i each with at least f/k faults.

3.3 Analysis of FS-pairs

Lemma 17. *Each fault in F will be in at most 4 FS-pairs.*

Proof. Recall that only post-light π'_i will be in FS-pairs. Suppose, for a contradiction, that there is some fault $e = (u, v)$ associated with 5 π'_i in FS-pairs. By pigeonhole, at least 3 of the π'_i have fault-free paths (as subsets of some shortcut) from their right ends x'_{i+1} to u, or at least 3 of the π'_i have fault-free paths from their right ends to v. Without loss of generality assume this is u. Let these subpaths be π'_a, π'_b, and π'_c, with $a < b < c$. We will also label the fault-free paths as p_a, p_b, and p_c. We have

$$|p_a| \leq |\pi'_a| - 2 \quad \text{and} \quad |p_c| \leq |\pi'_c| - 2$$

since the shortcut of π'_a which p_a is on has length at least $|p_a| + 1$ when we include e, and same with p_c.

Since π'_a is post-light, we have

$$|\pi'_{a+1}| \geq |\pi'_a|.$$

With each π'_i being extended from π_i by one vertex, we have also

$$|\pi_{a+1}| \geq |\pi_a|.$$

Moreover, since $a < b < c$, $a + 1 \neq c$. Note that the distance from x'_{a+1} to x'_{c+1} in $G \setminus F$ is equal to their distance along $\pi(s, t|F)$, a shortest path they're both on, which gives us a lower bound of

$$\begin{aligned} d_{G\setminus F}(x'_{a+1}, x'_{c+1}) &= \pi(s, t \mid F)[x'_{a+1}, x'_{c+1}] \\ &= \sum_{i=a+1}^{c} |\pi_i| \\ &\geq |\pi_{a+1}| + |\pi_c| \\ &\geq |\pi_a| + |\pi_c|. \end{aligned}$$

However, p_a and p_c give a fault-free path from x'_{a+1} to x'_{c+1} in $G \setminus F$ also, which upper bounds their distance as

$$\begin{aligned} d_{G\setminus F}(x'_{a+1}, x'_{c+1}) &\leq d_{G\setminus F}(x'_{a+1}, u) + d_{G\setminus F}(u, x'_{c+1}) \\ &\leq |p_a| + |p_c| \\ &\leq |\pi'_a| + |\pi'_c| - 4 \\ &= |\pi_a| + |\pi_c| - 2. \end{aligned}$$

Which contradicts $\pi(s, t \mid F)$ being a shortest path. Therefore, each fault in F is associated with at most 4 π'_i over all FS-pairs. □

We are now ready to finish the proof of Theorem 4. We can generate at least $\frac{(q-1)f}{2k}$ FS-pairs, but each fault is in at most 4 FS-pairs, and there are only f faults, so we have

$$4f \geq \frac{(q-1)f}{2k}.$$

Rearranging, we can upper bound q, the number of subpaths as

$$q \leq 8k + 1.$$

Corollary 18. *For any partition of $\pi(s,t \mid F)$ into subpaths π_i, there are at most $4f$ right FS-pairs containing post-light augmented subpaths $\{\pi'_i\}$.*

Again, in the other case where most subpaths are pre-light, the relevant corollary is that there are at most $4f$ left FS-pairs containing pre-light augmented subpaths $\{\pi'_i\}$. The proof is essentially identical.

4 Weighted Upper Bound

We next prove Theorem 6. Recall that the goal is to prove that in any weighted graph G, every f-fault replacement path π can be partitioned into

$$\pi = \pi_0 \circ e_0 \circ \pi_1 \circ e_1 \circ \cdots \circ e_{q-2} \circ \pi_{q-1}$$

where each e_i is an edge and each π_i is a (possibly empty) subpath of π that is an (f/k)-fault replacement path in G, with $q = O(k)$.

Our proof strategy will be similar to the previous argument with some minor changes: we still choose π_i greedily as the longest subpath which is an (f/k)-fault replacement path, and we will take the next edge in the subpath as the e_i to interweave. Let q be the number of subpaths resulting from this decomposition; our goal is to upper bound q to be linear in k.

We will define π'_i as π_i augmented with e_i (again π'_q is undefined). We define x_i as the vertex at the end of π'_{i-1} and at the beginning of π_i, so that for any i,

$$\pi(s,t \mid F)[x_i, x_{i+1}] = \pi'_i = \pi_i \circ e_i.$$

Unlike in the unweighted setting, we no longer have overlaps in the π'_i. We will assess whether subpaths π'_i are pre-light or post-light based on their weighted length, and proceed supposing that at least half of the subpaths are post-light. We generate FS-pairs with post-light subpaths as before, using the property that by maximality of π_i, each π'_i necessarily fails to be an (f/k)-fault replacement path. Using the same argument based on Hall's Theorem as before, this guarantees that we get at least $\frac{(q-1)f}{2k}$ distinct FS-pairs.

Now we can complete the proof of Theorem 6 by the following lemma to limit the number of FS-pairs each fault is in, which is analogous to Lemma 17 and has a similar proof. Theorem 6 follows as we can again upper bound $\frac{(q-1)f}{2k}$ by $4f$ to bound q.

Lemma 19. *Each (weighted) fault in F will be in at most 4 FS-pairs.*

Proof. Similarly to Lemma 17 we will prove the lemma by showing that no fault can be in 5 FS-pairs. Suppose, for a contradiction, that we have fault $e = (u,v)$ in 5 FS-pairs. Without loss of generality at least 3 subpaths π'_a, π'_b, and π'_c have fault-free paths which are contained in shortcuts from their right ends x_{a+1}, x_{b+1}, and x_{c+1} to u. Let these paths be p_a, p_b, and p_c. Since each path is contained in a shortcut using e, we have

$$w(p_a) < w(\pi'_a) - w(e) \quad \text{and} \quad w(p_c) < w(\pi'_c) - w(e).$$

Since π'_a is post-light, we have

$$w(\pi'_{a+1}) \geq w(\pi'_a).$$

Again we can use that $a+1 < c$ and that $\pi(s,t \mid F)$ is a shortest path to lower bound the weighted distance of x_{a+1} to x_{c+1} in $G \setminus F$ as

$$\begin{aligned} d_{G\setminus F}(x_{a+1}, x_{c+1}) &= w(\pi(s,t \mid F)[x_{a+1}, x_{c+1}]) \\ &= \sum_{i=a+1}^{c} w(\pi'_i) \\ &\geq w(\pi'_{a+1}) + w(\pi'_c) \\ &\geq w(\pi'_a) + w(\pi'_c). \end{aligned}$$

However we can use the fault free paths of p_a and p_c to upper bound the distance from x_{a+1} to x_{c+1} in $G \setminus F$ to get a contradiction with the previous lower bound:

$$\begin{aligned} d_{G\setminus F}(x_{a+1}, x_{c+1}) &\leq d_{G\setminus F}(x_{a+1}, u) + d_{G\setminus F}(u, x_{c+1}) \\ &\leq w(p_a) + w(p_c) \\ &< w(\pi'_a) + w(\pi'_c) - 2w(e). \end{aligned}$$

□

In the case that at least half of the subpaths are pre-light, we will generate FS-pairs with pre-light subpaths by defining base faults relative to the left ends x_i of subpaths π'_i. In the analysis, we replace x_{a+1}, x_{b+1}, and x_{c+1} with x_a, x_b and x_c. Our analysis of p_a, p_b, and p_c are unchanged. Comparing subpaths, we instead use the pre-light property of π'_c to get

$$w(\pi'_{c-1}) \geq w(\pi'_c).$$

Then the analysis on the distance is a lower bound of

$$\begin{aligned} d_{G\setminus F}(x_a, x_c) &= w(\pi(s,t \mid F)[x_a, x_c]) \\ &= \sum_{i=a}^{c-1} w(\pi'_i) \\ &\geq w(\pi'_a) + w(\pi'_{c-1}) \\ &\geq w(\pi'_a) + w(\pi'_c), \end{aligned}$$

and an upper bound of

$$\begin{aligned} d_{G\setminus F}(x_a, x_c) &\leq d_{G\setminus F}(x_a, u) + d_{G\setminus F}(u, x_c) \\ &\leq w(p_a) + w(p_c) \\ &< w(\pi'_a) + w(\pi'_c) - 2w(e). \end{aligned}$$

5 Algorithmic Path Decomposition

We will next prove Theorem 7, which holds for unweighted input graphs, and then afterwards describe the (minor) changes needed to adapt the algorithm to the weighted setting. As a reminder of our goal: we are given a graph G, a fault set F, a replacement path $\pi(s,t \mid F)$, and a parameter k on input. Our goal is to find nodes $\{x_i\}$ and fault sets F_i, which partitions $\pi(s,t \mid F)$ into $q = O(k)$ replacement paths avoiding f/k faults each, as

$$\pi(s,t \mid F) = \pi(x_0, x_1 \mid F_1) \circ \pi(x_1, x_2 \mid F_2) \circ \ldots \circ \pi(x_{q-1}, x_q \mid F_q).$$

5.1 Fault Set Reducing Subroutine

Before describing our main algorithm, we will start with a useful subroutine, driven by an observation about the matching step in FS-pair generation. In our upper bound proof, we used a process for generating FS-pairs to bound the number of subpaths in the decomposition. We used *minimum size* of the fault set F_i associated to each augmented subpath π'_i to argue that we could generate $|F_i|$ distinct FS-pairs.

The observation is that, letting F_i be *any* (not necessarily minimum) valid fault set for π'_i (that is, π'_i is a shortest path in $G \setminus F_i$), if we can produce an FS-pair for every fault in F_i then our previous argument works. On the other hand, if we cannot produce an FS-pair for every fault in F_i, then our previous argument gives us a process by which we can find a strictly smaller fault set F'_i that is also valid for π'_i, by replacing the subset of F_i with the reduced set of their base faults.

The subroutine FAULTREDUCE runs this process iteratively, in order to find a fault set F_i for the input subpath π_i that can be used to generate $|F_i|$ FS-pairs (from both the left and right). We note the subtlety that F_i is not necessarily a minimum valid fault set for π_i: as in Figure 4, there may exist a smaller valid fault set, but the algorithm will halt nonetheless if it can certify that the appropriate number of FS-pairs can be generated.

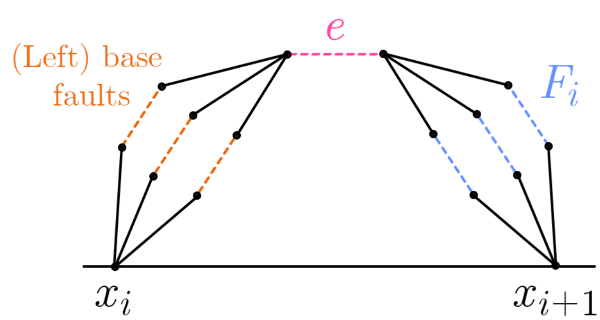

Figure 4: This subpath and fault set F_i produces multiple FS-pairs via a saturated matching using the faults on the left as base faults, but its minimum fault set is only one edge $\{e\}$.

Algorithm 1 FAULTREDUCE (π_i)

$F_i \leftarrow F$
Construct Γ_L.
Construct Γ_R.
loop
  Compute max matching M_L for F_i in Γ_L.
  Compute max matching M_R for F_i in Γ_R.
  if $|M_L| < |F_i|$ or $|M_R| < |F_i|$ **then**
    Reduce F_i
  else return F_i

The essential properties of Algorithm 1 are captured by the following lemma.

Lemma 20. *Relative to a graph G and fault set F, there is a subroutine (Algorithm 1 - FaultReduce) that runs in polynomial time with the following behavior:*

- *The input is a path π_i that is a shortest path in $G \setminus F$.*
- *The output is a fault set $F_i \subseteq F$, such that:*
  - *π_i is a shortest path in $G \setminus F_i$, and*
  - *one can generate $|F_i|$ left- and $|F_i|$ right-FS-pairs of π_i from F_i.*

We will next provide additional details on some of the steps in Algorithm 1, and then prove Lemma 20.

Construction of Γ_L and Γ_R. As in our previous proof, the graphs Γ_L, Γ_R are the graphs representing the association between faults in F_i and left or right (respectively) base faults in F. More specifically:

- Both Γ_L and Γ_R are bipartite graphs with vertex set $F_i \cup F$, where F_i is the current fault set, and F is all initial faults.

  Thus faults in F_i are represented by two vertices, one on each side of the bipartition.

- In Γ_L, we place an edge from $e \in F_i$ to $e_b \in F$ if and only if e_b is a left base fault for e. The edges of Γ_R are defined similarly, with respect to right base faults.

These graph constructions require us to efficiently check whether or not a particular fault $e_b \in F$ acts as a (left or right) base fault for some $e \in F_i$. We next describe this process:

Lemma 21. *Given a subpath π_i, a valid fault set F_i, and faults $e \in F_i, e_b \in F$, we can check whether or not e_b is a left and/or right base fault of e in polynomial time.*

Proof. First, the following notation will be helpful. Let x_i, x_{i+1} be the endpoints of the input subpath π_i. We will write $d(x_i, x_{i+1} \mid e_b \rightsquigarrow e)$ for the length of the shortest (possibly non-simple) (x_i, x_{i+1})-path that contains both e_b and e, and which specifically uses e_b as the first fault in F along the path. We define $d(x_{i+1}, x_i \mid e_b \rightsquigarrow e)$ similarly. Note that e_b is a left base fault for e if and only if

$$d(x_i, x_{i+1} \mid e_b \rightsquigarrow e) < |\pi_i|$$

and that e_b is a right base fault for e if and only if

$$d(x_{i+1}, x_i \mid e_b \rightsquigarrow e) < |\pi_i|.$$

Thus, it suffices to compute the values of the left-hand side of these two inequalities. We will next describe computation of $d(x_i, x_{i+1} \mid e_b \rightsquigarrow e)$; the other computation is symmetric. There are two cases, depending on whether or not $e_b = e$. Let $e = (u, v)$, $e_b = (u_b, v_b)$. When $e_b \neq e$, the formula is:

$$\begin{aligned} d(x_i, x_{i+1} \mid e_b \rightsquigarrow e) = \min\{ & d_{G \setminus F}(x_i, u_b) + d_G(v_b, u) + d_G(v, x_{i+1}) + 2, \\ & d_{G \setminus F}(x_i, u_b) + d_G(v_b, v) + d_G(u, x_{i+1}) + 2, \\ & d_{G \setminus F}(x_i, v_b) + d_G(u_b, u) + d_G(v, x_{i+1}) + 2, \\ & d_{G \setminus F}(x_i, v_b) + d_G(u_b, v) + d_G(u, x_{i+1}) + 2\}. \end{aligned}$$

The four parts are needed since we consider paths that use e, e_b with either orientation, and the $+2$ term arises to count the contribution of the edges e, e_b themselves. In the case where $e_b = e$, the formula is

$$d(x_i, x_{i+1} \mid e \rightsquigarrow e) = \min\{d_{G \setminus F}(x_i, u) + d_G(v, x_{i+1}) + 1, d_{G \setminus F}(x_i, v) + d_G(u, x_{i+1}) + 1\}.$$

□

Reducing F_i. Next, we provide more detail on the step of reducing the fault set F_i. This uses Hall's condition, in an analogous way to our previous proof. When we compute max matchings M_L, M_R for Γ_L, Γ_R, if we successfully find matchings of size $|M_L| \geq |F_i|$ or $|M_R| \geq |F_i|$, then we have certified the ability to generate $|F_i|$ left and right FS-pairs as in Section 3.2, and so the algorithm can return F_i and halt. Otherwise, suppose without loss of generality that $|M_L| < |F_i|$. By Hall's condition, that means there exists a fault subset $A \subseteq F_i$ such that the set of base faults $B \subseteq F$ used by faults in A is strictly smaller than A itself. For the reduction step, we set $F_i \leftarrow F_i \cup B \setminus A$, which reduces the size of $|F_i|$. By Lemma 16, this maintains the invariant that F_i is a valid fault set for the input path π_i.

In order to efficiently find the non-expanding fault subset $A \subseteq F_i$, we may compute the max matching in Γ_L (or Γ_R) using a primal-dual algorithm that returns both a max matching and a certificate of maximality of this form. For example, the Hungarian algorithm will do [CLRS09].

5.2 Main Algorithm

ComputeSubpaths, described in Algorithm 2, performs a greedy search for subpath boundaries. In each round, we set the next subpath boundary node x_{i+1} to be the furthest node from the previous subpath boundary node x_i, such that the corresponding subpath is certified by the algorithm FaultReduce to have size at most f/k. Thus, considering the augmented subpath that we get by adding an additional node to π_i, we can generate at more than f/k left and right FS-pairs from this subpath.

We next state the algorithm; for ease of notation we label the vertices of the input path $\pi(s, t \mid F)$ as $s = v_0, v_1, \ldots, v_\ell = t$.

Algorithm 2 ComputeSubpaths $(\pi(s,t \mid F), F, k)$

$x_0 \leftarrow s$.
$i \leftarrow 0$.
while $x_i \neq t$ **do**
  Binary search for largest y such that the fault set returned by FaultReduce$(\pi(s,t)[x_i, v_y])$ has size $\leq f/k$.
  $i \leftarrow i+1$.
  $x_i \leftarrow v_y$.
return $\{x_j\}_{j=0}^{i}$

Theorem 22. *Algorithm 2 is correct and runs in polynomial time.*

Proof. In Corollary 18 from our upper bound section, we showed that there exist only $O(f)$ total right FS-pairs using post-light subpaths (and, symmetrically, there exist only $O(f)$ left FS-pairs using pre-light subpaths). Since at least half of the augmented subpaths are pre-light or half are post-light, and by Lemma 20 every augmented subpath can generate at least f/k left and right FS-pairs, altogether we will have at most $O(k)$ subpaths.

For runtime, we always generate a linear number of subpaths, and locating the endpoint of each requires calling the subroutine $\log n$ times. Thus the entire algorithm runs in polynomial time, specifically $O(m \cdot poly(\log(n), f))$ time: each reduction of the subroutine takes $O(m)$ time for distance computations to build Γ_L and Γ_R and $poly(f)$ time for max matchings. □

A similar approach works in the weighted setting, since the method of counting FS-pairs extends to the structure in Theorem 6 and upper bounds the number of interweaved subpaths and edges. The construction of auxiliary graphs Γ_L and Γ_R requires checking the weighted distance, but the matching and FS-pair generation is the same. We change the algorithm to add the next edge into the decomposition of $\pi(s,t \mid F)$ after finding a maximal subpath with fault set at most f/k. The upper bound for the number of subpaths based on enough FS-pairs being generated follows from the analysis in Theorem 6.

6 Lower Bounds

As a warmup, we begin by showing our lower bound against decomposition into 2 subpaths. This graph will form a building block which we will sequentially compose into graphs where the number of subpaths required is amplified, to attain our main lower bound for $2k$ subpaths.

Our 2 subpath lower bound consists of a long shortest path after the failure of many additional 'shortcutting' edges. The additional edges are constructed so any two of them will always provide a shortcut for both halves of the long path, and at least one subpath entirely contains one of these halves. This necessitates the inclusion of almost every shortcutting edge into a replacement path's fault set and requires that subpath to be a $(f-1)$-replacement path. This is illustrated in Figure 5 and described in the following proof.

Proposition 23. *For all $f \geq 2$, $(2, f-2)$ is not restorable.*

Proof. We will first assume for convenience that f is even, and return to the case where f is odd at the end. Let $g = f/2$ and let G_f be the graph as illustrated in Figure 5. Formally: the vertices of G_f are $1, 2, \ldots, N := 2^{g+1} - 1$ (labeled clockwise in Figure 5), and its edge set is $E_1 \cup E_2 \cup E_3$, where

$$E_1 := \{(2^k, 2^{g+1} - 2^{k+2}), 0 \leq k \leq g-3\}$$

$$E_2 := \{(2^{k+2}, 2^{g+1} - 2^k), 0 \leq k \leq g-3\}$$

$$E_3 := \{(i, i+1), 1 \leq i \leq N-1\}$$

In Figure 5, the edges in E_1 are drawn in blue and slope upwards to the right, and the edges in E_2 are drawn in yellow and slope upwards to the left. E_3 is in black and forms the outer curve. Let $F := E_1 \cup E_2$, and

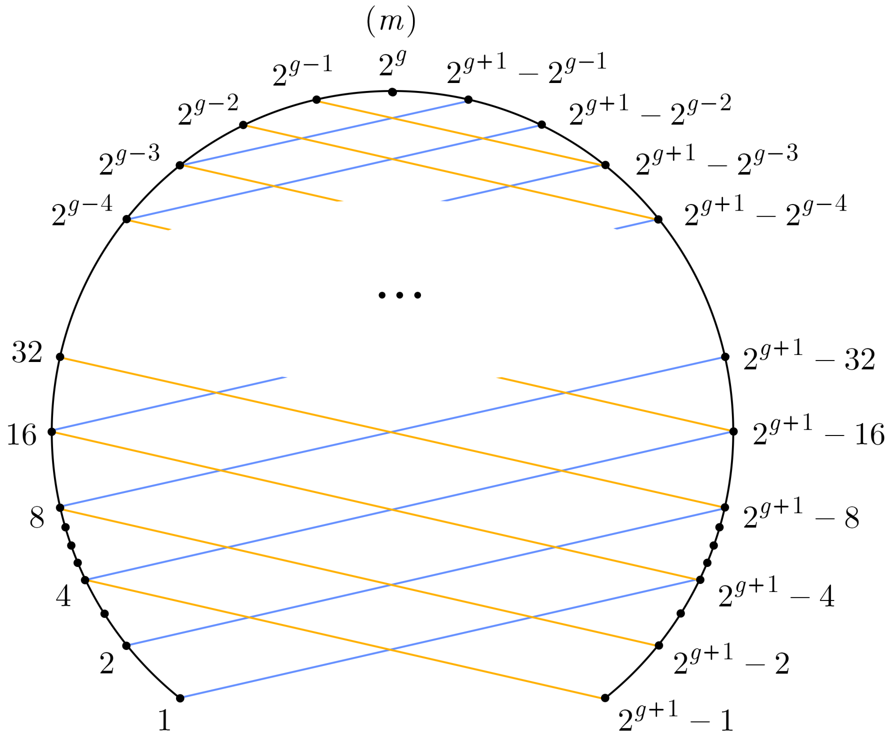

Figure 5: If the blue and yellow edges fail (i.e. all straight-line edges on the inside of the outer semicircle), then we can't partition the remaining shortest path (black edges along the outer semicircle) into two subpaths that are both $(f-2)$-fault replacement paths. For clarity, only power of two vertices are drawn here, but the outer cycle contains $2^{g+1}-1$ vertices.

notice there is a unique replacement path $\pi(1, N \mid F)$ which consists of E_3, the outer curve. Note that G_f is symmetric about the vertex $m = 2^g$, which is also the midpoint of $\pi(1, N \mid F)$. Define the "half-arcs" of this graph as $\pi(1, m \mid F)$ and $\pi(m, N \mid F)$, the two subpaths partitioning $\pi(1, N \mid F)$ into equal parts divided at midpoint m. (We note that this partitioning will be used again in Lemma 25 as well.)

Let $x \in \pi(1, N \mid F)$ be an arbitrary vertex, which splits the path into a prefix from 1 to x and a suffix from x to N. Let F_1, F_2 respectively be minimum-size fault sets such that the prefix and suffix are replacement paths avoiding F_1, F_2. We will write the prefix and suffix as

$$\pi(1, x \mid F_1) \quad \text{and} \quad \pi(x, N \mid F_2).$$

By symmetry of the construction we may assume without loss of generality that $x \geq m$, and so

$$\pi(1, m \mid F) \subseteq \pi(1, x \mid F_1).$$

We will now proceed to show that F_1 must contain every edge in E_1 and all but one edge in E_2, and hence $|F_1| \geq f - 1$.

First Part (Proof of $E_1 \subseteq F_1$). Consider $\pi(1, x \mid F_1)$; suppose for a contradiction that there is an edge $(2^c, 2^{g+1} - 2^{c+2}) \in E_1 \setminus F_1$, with $c \leq g - 3$. Then we can construct a path p, a shortcut from 1 to x by traversing through $(2^c, 2^{g+1} - 2^{c+2})$ and then using edges in E_3 to get to x. Explicitly, this path is:

$$p = \begin{cases} (1, 2, \ldots 2^c) \circ (2^c, 2^{g+1} - 2^{c+2}) \circ (2^{g+1} - 2^{c+2}, \ldots x+1, x) & \text{if } x \leq 2^{g+1} - 2^{c+2} \\ (1, 2, \ldots 2^c) \circ (2^c, 2^{g+1} - 2^{c+2}) \circ (2^{g+1} - 2^{c+2}, \ldots x-1, x) & \text{if } x > 2^{g+1} - 2^{c+2}. \end{cases}$$

In the first case, the length of p is

$$|p| = 2^c + 2^{g+1} - 2^{c+2} - x.$$

Since by assumption $x \geq m = 2^g$, we have $2^{g+1} - x \leq 2^g \leq x$, so we can upper bound

$$|p| \leq 2^c - 2^{c+2} + x.$$

In the second case, the length of the path is

$$|p| = 2^c + 2^{c+2} - 2^{g+1} + x.$$

Since $c \leq g - 3$, we can directly upper bound the path length as

$$|p| \leq 2^{g-3} + 2^{g-1} - 2^{g+1} + x.$$

In either case, the length of p is strictly less than $x-1$, the length of shortest path $\pi(1, x \mid F)$, a contradiction. We must therefore have $E_1 \subseteq F_1$.

Second Part (Proof of $|E_2 \setminus F_1| \leq 1$): Suppose for a contradiction that there are two edges of E_2 which F_1 does not contain: $(2^{a+2}, 2^{g+1} - 2^a)$ and $(2^{b+2}, 2^{g+1} - 2^b)$ with $a < b$. Then in $G \setminus F_1$ we have a walk w from 1 to x defined as

$$\begin{aligned} w = (1, 2, \ldots 2^{a+2}) \circ (2^{a+2}, 2^{g+1} - 2^a) \circ (2^{g+1} - 2^a, 2^{g+1} - 2^a - 1, \ldots 2^{g+1} - 2^b) \\ \circ (2^{g+1} - 2^b, 2^{b+2}) \circ (2^{b+2}, 2^{b+2} + 1, \ldots, x - 1, x), \end{aligned}$$

of length

$$\begin{aligned} |w| &= 2^{a+2} - 2^a + 2^b + 1 + x - 2^{b+2} \\ &= x - 3(2^b - 2^a) + 1 \end{aligned}$$

Then the length of w will be strictly less than $x-1$, the length of $\pi(1, x \mid F)$, a contradiction. Thus we must include all of F in F_1 except at most one edge from E_2.

Finally, in the case that f is odd, we instead construct G_f with $g = \lceil f/2 \rceil$, and take any edge out of E_1 or E_2, which does not change the analysis. ☐

Our lower bound with two subpaths generalises to our main lower bound result, which we rewrite below:

Proposition 24. *For any $k \in \mathbb{N}$, $(2k, \lfloor f/k \rfloor - 2)$ is not restorable.*

Proof. Assume for convenience that k divides f. We will glue k copies of the graph with f/k faults in the previous proposition together, and then show that for any division of a particular f-fault replacement path into subpaths, one subpath must contain one of the half-arcs as defined before, and its fault set will have to include $f/k - 1$ faults.

We take k copies of $G_{f/k}$ as defined in Proposition 23, denoted by $G_{1,f/k}, G_{2,f/k}, \ldots G_{k,f/k}$, labeling the vertices of $G_{i,f/k}$ as (i, j) where j is the label of the corresponding vertex in $G_{f/k}$. We identify each $(i, 2^{g+1} - 1)$ with $(i+1, 1)$. The edges in this graph are the union of all edges of the $G_{i,f/k}$ (see Figure 6), and we define F as the union of the fault sets of each $G_{f/k}$ as defined in the proof of Proposition 23. Let $E_{j,i}$ denote the E_j for $G_{i,f/k}$ with $j \in \{1, 2\}$, so that formally

$$F := \bigcup_{i=1}^{k} \left(E_{1,i} \cup E_{2,i}\right).$$

Let $s := (1, 1)$, $t := (k, 2^{g+1} - 1)$. Consider $\pi(s, t \mid F)$. This f-fault replacement path is precisely the non-fault edges in G, or the union of $E_{3,i}$ over each of the $G_{i,f/k}$.

We now bring in the previous half-arc structure from the case with two subpaths. This graph contains all the half-arcs of each $G_{i,f/k}$, and the half-arcs can be expressed either as $\pi(s,t)[(i,1),(i,m)]$ or $\pi(s,t)[(i,m),(i,2^{g+1} - 1)]$. From Proposition 23, we have the following:

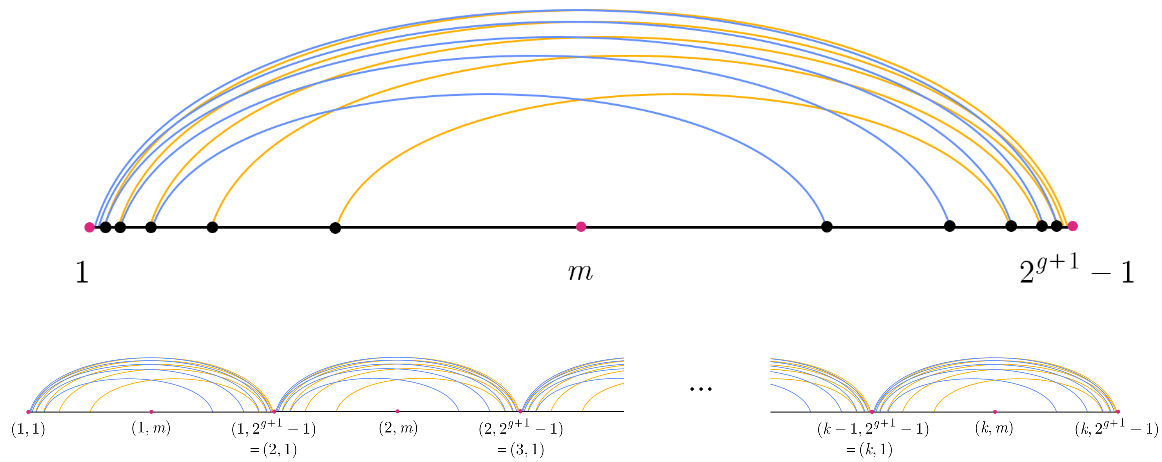

Figure 6: The top figure depicts one copy of $G_{f/k}$, and the bottom depicts all the copies combined together.

Lemma 25. *A path containing a half-arc of any $G_{f/k}$ subgraph cannot be a $(f/k-2)$-fault replacement path.*

Proof. Following from the argument of Proposition 23, a fault replacement path containing a half arc of $G_{i,2f/k}$ must have its fault set contain at least every edge in $E_{1,i} \cup E_{2,i}$ except possibly one. Thus any fault set of that path has size at least $f/k - 1$. □

We will show that any division of $\pi(s,t \mid F)$ into $2k$ subpaths will result in one subpath containing a half-arc, and thus failing to be a $(f/k-2)$-fault replacement path. Suppose we have some choice of boundary vertices $x_1, x_2, \ldots x_{2k-1}$ and corresponding fault subsets $F_1, F_2, \ldots F_{2k}$, so that each $\pi(s,t)[x_{i-1}, x_i]$ is a shortest path in $G \setminus F_i$.

Let the *interior vertices* of a path denote all its vertices except its first and last. Note that $\pi(s,t \mid F)$ contains $2k$ half-arcs, and any half-arc which does not have any x_i in its interior vertices will be completely contained in some $\pi(s,t)[x_{j-1}, x_j]$. The interior vertices of all $2k$ half-arcs are disjoint, and we only have $2k-1$ x_i which can be in the interior of half arcs. Therefore some subpath $\pi(s,t)[x_{i-1}, x_i]$ must contain a half-arc, and its fault set $|F_i|$ must have size at least $f/k - 1$. Thus we will always get that one of the subpaths cannot be a $(f/k-2)$-fault replacement subpath, proving the lower bound.

In the case when k does not divide f, we choose graphs which are as even as possible to combine; Let a be the remainder of f divided by k. We glue a copies of $G_{\lfloor f/k \rfloor + 1}$ to $(k-a)$ copies of $G_{\lfloor f/k \rfloor}$. In this case the subpath which contains a half-arc might contain a half arc of $G_{\lfloor f/k \rfloor}$, and will enforce a fault set of size only $\lfloor f/k \rfloor - 1$. □

If we want a similar result using this method for the case for an odd number of subpaths, say $2k-1$, we still need to construct k copies of $G_{f/k}$, since half-arcs come in pairs, and we get the same bound on fault sets. Alternatively, we can also use monotonicity to directly get:

Corollary 26. *For any $k \in \mathbb{N}$, $(2k-1, \lfloor f/k \rfloor - 2)$ is not restorable.*

Acknowledgments

We are grateful to Vijaya Ramachandran for helpful references to prior work, to Dmitry Paramonov for insightful writing suggestions, and to an anonymous reviewer for exceptionally helpful and thorough comments.

References

[ABBK+02] Yehuda Afek, Anat Bremler-Barr, Haim Kaplan, Edith Cohen, and Michael Merritt. Restoration by path concatenation: fast recovery of mpls paths. *Distributed Computing*, 15:273–283, 2002.

[ABDS+20] Reyan Ahmed, Greg Bodwin, Faryad Darabi Sahneh, Stephen Kobourov, and Richard Spence. Weighted additive spanners. In Isolde Adler and Haiko Müller, editors, *Graph-Theoretic Concepts in Computer Science*, pages 401–413, Cham, 2020. Springer International Publishing.

[BCG+18] Davide Bilò, Keerti Choudhary, Luciano Gualà, Stefano Leucci, Merav Parter, and Guido Proietti. Efficient oracles and routing schemes for replacement paths. In *35th Symposium on Theoretical Aspects of Computer Science (STACS 2018)*. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2018.

[BGPW17] Greg Bodwin, Fabrizio Grandoni, Merav Parter, and Virginia Vassilevska Williams. Preserving Distances in Very Faulty Graphs. In Ioannis Chatzigiannakis, Piotr Indyk, Fabian Kuhn, and Anca Muscholl, editors, *44th International Colloquium on Automata, Languages, and Programming (ICALP 2017)*, volume 80 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 73:1–73:14, Dagstuhl, Germany, 2017. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik.

[BP21] Greg Bodwin and Merav Parter. Restorable shortest path tiebreaking for edge-faulty graphs. In *Proceedings of the 2021 ACM Symposium on Principles of Distributed Computing*, PODC'21, page 435–443, New York, NY, USA, 2021. Association for Computing Machinery.

[CC19] Shiri Chechik and Sarel Cohen. Near optimal algorithms for the single source replacement paths problem. In *Proceedings of the Thirtieth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 2090–2109. SIAM, 2019.

[CCFK17] Shiri Chechik, Sarel Cohen, Amos Fiat, and Haim Kaplan. $(1+\varepsilon)$-approximate f-sensitive distance oracles. In *Proceedings of the Twenty-Eighth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 1479–1496. SIAM, 2017.

[CLRS09] Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein. *Introduction to Algorithms, 3rd Edition*. MIT Press, 2009.

[DR22] Ran Duan and Hanlin Ren. Maintaining exact distances under multiple edge failures. In *Proceedings of the 54th Annual ACM SIGACT Symposium on Theory of Computing*, STOC 2022, page 1093–1101, New York, NY, USA, 2022. Association for Computing Machinery.

[GJM20] Manoj Gupta, Rahul Jain, and Nitiksha Modi. Multiple source replacement path problem. In *Proceedings of the 39th Symposium on Principles of Distributed Computing*, pages 339–348, 2020.

[HS01] John Hershberger and Subhash Suri. Vickrey prices and shortest paths: What is an edge worth? In *Proceedings 42nd IEEE symposium on foundations of computer science*, pages 252–259. IEEE, 2001.

[KIM82] Naoki Katoh, Toshihide Ibaraki, and Hisashi Mine. An efficient algorithm for k shortest simple paths. *Networks*, 12(4):411–427, 1982.

[LYXsS20] Bao Ju Liu, Peng Yu, Qiu Xue-song, and Lei Shi. Survivability-aware routing restoration mechanism for smart grid communication network in large-scale failures. *EURASIP journal on wireless communications and networking*, 2020:1–21, 2020.

[MMG89] Kavindra Malik, Ashok K Mittal, and Santosh K Gupta. The k most vital arcs in the shortest path problem. *Operations Research Letters*, 8(4):223–227, 1989.

[MR22] Vignesh Manoharan and Vijaya Ramachandran. Near optimal bounds for replacement paths and related problems in the congest model. *arXiv preprint arXiv:2205.14797*, 2022.

[WGY+15] Rui Wang, Suixiang Gao, Wenguo Yang, Zhipeng Jiang, et al. Restorable energy aware routing with backup sharing in software defined networks. *J. Commun.*, 10(8):551–561, 2015.

[ZGK+21] Zhizhen Zhong, Manya Ghobadi, Alaa Khaddaj, Jonathan Leach, Yiting Xia, and Ying Zhang. Arrow: restoration-aware traffic engineering. In *Proceedings of the 2021 ACM SIGCOMM 2021 Conference*, pages 560–579, 2021.