Multiple Source Dual Fault Tolerant BFS Trees*

Manoj Gupta[1] and Shahbaz Khan[†2]

1 **IIT Gandhinagar, Gandhinagar, India**
gmanoj@iitgn.ac.in

2 **Department of CSE, IIT Kanpur, Kanpur, India**
shahbazk@cse.iitk.ac.in

Abstract

Let $G = (V, E)$ be a graph with n vertices and m edges, with a designated set of σ sources $S \subseteq V$. The *fault tolerant subgraph* for any graph problem maintains a sparse subgraph $H = (V, E')$ of G with $E' \subseteq E$, such that for any set F of k failures, the solution for the graph problem on $G \setminus F$ is maintained in its subgraph $H \setminus F$. We address the problem of maintaining a fault tolerant subgraph for computing *Breath First Search tree* (BFS) of the graph from a single source $s \in V$ (referred as k FT-BFS) or multiple sources $S \subseteq V$ (referred as k FT-MBFS). We simply refer to them as FT-BFS (or FT-MBFS) for $k = 1$, and dual FT-BFS (or dual FT-MBFS) for $k = 2$.

The problem of k FT-BFS was first studied by Parter and Peleg [ESA13]. They designed an algorithm to compute FT-BFS subgraph of size $O(n^{3/2})$. Further, they showed how their algorithm can be easily extended to FT-MBFS requiring $O(\sigma^{1/2}n^{3/2})$ space. They also presented matching lower bounds for these results. The result was later extended to solve dual FT-BFS by Parter [PODC15] requiring $O(n^{5/3})$ space, again with matching lower bounds. However, their result was limited to only edge failures in undirected graphs and involved very complex analysis. Moreover, their solution doesn't seems to be directly extendible for dual FT-MBFS problem.

We present a similar algorithm to solve dual FT-BFS problem with a much simpler analysis. Moreover, our algorithm also works for vertex failures and directed graphs, and can be easily extended to handle dual FT-MBFS problem, matching the lower bound of $O(\sigma^{1/3}n^{5/3})$ space described by Parter [PODC15]. The key difference in our approach is a much simpler classification of path interactions which formed the basis of the analysis by Parter [PODC15].

1998 ACM Subject Classification E.1 Graphs and Networks, G.2.2 Graph Algorithms, Network problems, Trees, G.4 Algorithm Design and Analysis, F.2.2 Computations on Discrete Structures

Keywords and phrases BFS, fault-tolerant, graph, algorithms, data-structures

Digital Object Identifier 10.4230/LIPIcs.ICALP.2017.127

1 Introduction

Graph networks are extensively used to study real world applications ranging from communication networks as internet and telephony, to supply chain networks, road networks etc. Every now and then, these networks are susceptible to failures of links and nodes, which drastically affects the performance of these applications. Hence, most algorithms developed for these applications are also studied in the *fault tolerant model*, which aims to provide solutions to the corresponding problem that are resilient to such failures. Since such failures of nodes or links in the network though unpredictable are rare and are often readily repaired, the applications generally address the scenarios expecting the number of simultaneous faults to

* The full version of the paper can be found in [11].

† This research work was supported by Google India under the Google India PhD Fellowship Award.

44th International Colloquium on Automata, Languages, and Programming (ICALP 2017).
Editors: Ioannis Chatzigiannakis, Piotr Indyk, Fabian Kuhn, and Anca Muscholl;
Article No. 127; pp. 127:1–127:15
Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

be much smaller than the size of the network. This aspect is often modeled by bounding such failures using some parameter k (typically $k << n$), and studying fault tolerant structures resilient to upto k failures.

Among the different approaches to develop fault tolerance in a structure, we use the approach of computing a *fault tolerant subgraph* described as follows. For a given graph $G = (V, E)$, the fault tolerant subgraph for any graph problem maintains a sparse subgraph $H = (V, E')$ of G having $E' \subseteq E$, such that for any set of edge (or vertex) failures $F \subseteq E$ (or $F \subseteq V$), the solution for the graph problem on $G' = (V, E \setminus F)$ (or $G' = (V \setminus F, E)$) is maintained in its subgraph $H' = (V, E \setminus F)$ (or $H' = (V \setminus F, E)$). We shall henceforth abuse the notation to denote the graphs after such a set of failures F as $G \setminus F$ and $H \setminus F$ respectively. A standard motivation for this approach is a communication network where each link corresponds to a communication channel [16], where the system designer is required to purchase or lease the channels to be used by the application. Hence, the aim is to acquire a minimal set of these channels (the subgraph H of G) for successfully performing the application with resilience of upto k faults. Fault tolerant subgraphs are also developed for other graph problems maintaining reachability [13, 2, 3], strong-connectivity [3] and approximate shortest paths from a single source [12, 17, 5] and all sources [7, 9, 6, 14, 4].

Breadth First Search (BFS) is a fundamental technique for graph traversal. From any given source $s \in V$, BFS produces a rooted spanning tree in $O(m + n)$ time. For an unweighted graph, the BFS tree from a source s is also the shortest path tree from s because it preserves the shortest path from s to every vertex $v \in V$ that is reachable from s. We are thus interested to maintain fault tolerant subgraphs for computing BFS trees from a single source (referred as k FT-BFS) and multiple sources k FT-MBFS described as follows.

▶ **Definition 1** (k FT-BFS)**.** Given a graph $G = (V, E)$ with a designated source $s \in V$, build a subgraph $H = (V, E')$ with $E' \subseteq E$, such that after any set F of k failures in G, the BFS tree from s in $H \setminus F$ is a valid BFS tree from s in $G \setminus F$.

▶ **Definition 2** (k FT-MBFS)**.** Given a graph $G = (V, E)$ with a designated set of sources $S \subseteq V$, build a subgraph $H = (V, E')$ with $E' \subseteq E$, such that after any set F of k failures in G, for each $s \in S$ the BFS tree from s in $H \setminus F$ is a valid BFS tree from s in $G \setminus F$.

For convenience of notation, for $k = 1$ and $k = 2$ we refer to these problems as FT-BFS (or FT-MBFS) and dual FT-BFS (or dual FT-MBFS). The problems of k FT-BFS (and k FT-MBFS) were first studied by Parter and Peleg [16] for a single failure. They designed an algorithm to compute FT-BFS requiring $O(n^{3/2})$ space. Further, they showed their result can be easily extended to FT-MBFS requiring $O(\sigma^{1/2} n^{3/2})$ space. Moreover, their upper bounds were complemented by matching lower bounds for both their results. This result was later extended to address dual FT-BFS by Parter [15] requiring $O(n^{5/3})$ space. However, the application of this result was limited to only edge failures in undirected graphs. Though the analysis of their result was significantly complex, it paved a way for developing the theory studying the interaction of replacement paths after a single edge failure, their classification and corresponding properties. Further, they also generalized the lower bound for k FT-MBFS to $\Omega(\sigma^{\frac{1}{k+1}} n^{2-\frac{1}{k+1}})$ which matches their solution for dual FT-BFS. They also stated extensions of their result to dual FT-MBFS (or k FT-BFS) as an open problem.

The difference in complexity of dual FT-BFS over FT-BFS also reinforces the idea that extending such results from one failure to two failures (and beyond) requires a significantly more advanced analysis. As described by Parter [15], for the problem of maintaining shortest paths *"a sharp qualitative and quantitative difference"* has been widely noted while handling a single failure and multiple failures. For the problem of maintaining fault tolerant distance oracles, despite a simple and elegant algorithm for a single edge failure [8], the solution for two edge failures [10] is significantly complex. In fact, the authors [10] themselves mention that extending their approach beyond 2 edge failure would be infeasible due to numerous case analysis involved, requiring a fundamentally different approach. This key difference is also visible when we compare other problems, as bi-connectivity with tri-connectivity, single fault tolerant reachability [13, 2] with dual fault tolerant reachability [3], etc. Hence, simplifying the analysis of dual FT-BFS (and hence dual FT-MBFS) structures seem to be an essential building block for further developments of the problem for multiple failures.

1.1 Our Contributions

We design optimal algorithms for constructing dual FT-BFS and dual FT-MBFS structures. In principle, the core algorithm of our construction for dual FT-BFS is same as the one given by Parter [15], with a much simpler and more powerful analysis. As a result, our algorithm also works for vertex failures and directed graphs. Also, our dual FT-BFS structure can also be easily extended to handle dual FT-MBFS (as in case of FT-BFS [16]), which matches the lower bound described by Parter [15]. Thus, we optimally solve two open problems (dual FT-BFS for directed graphs and dual FT-MBFS for any graphs) as follows.

▶ **Theorem 3** (Optimal dual FT-BFS). *Given any graph $G = (V, E)$ having n vertices and m edges, with a designated source $s \in V$, there is a polynomial time constructable dual FT-BFS subgraph H having $O(n^{5/3})$ edges.*

▶ **Theorem 4** (Optimal dual FT-MBFS). *Given any graph $G = (V, E)$ having n vertices and m edges, with a designated set of σ sources $S \subseteq V$, there is a polynomial time constructable dual FT-MBFS subgraph H having $O(\sigma^{1/3} n^{5/3})$ edges.*

Our analysis is performed using simple techniques based on *counting arguments*. We classify a set of shortest paths as *standard* paths and prove the properties of *disjointness* and *convergence* for a designated suffix of such paths. The extension to directed graphs additionally uses the notion of *segmentable* paths (similar notion of *regions* was used in [15]) for every set of *converging* shortest paths, and establishes several interesting properties for them. These properties and analysis techniques might be of independent interest in the theory of shortest paths.

1.2 Related Work

As described earlier BFS is strongly related to shortest paths. Demetrescu et al. [8] showed that there exist weighted directed graphs, for which a fault tolerant subgraph requires $\Theta(m)$ edges for maintaining shortest paths even from a single source after a vertex failure. Hence, they designed a data-structure of size $\tilde{O}(n^2)$ [1] that reports all pairs shortest distances after a vertex failure in $O(1)$ time. Duan and Pettie [10] extended this result to two failures requiring nearly same (upto *poly* $\log n$ factors) size and reporting time.

Other related problems include fault tolerant DFS and fault tolerant reachability. Baswana et al. [1] presented a $\tilde{O}(m)$ sized fault tolerant data structure that reports the DFS tree of an undirected graph after k faults in $\tilde{O}(nk)$ time. For single source reachability, Baswana et al. [3] presented an algorithm for computing fault tolerant reachability subgraphs for k faults using $O(2^k n)$ edges. This result was also shown to be optimal upto constant factors.

[1] $\tilde{O}(\cdot)$ notation hides poly-log(n) factors

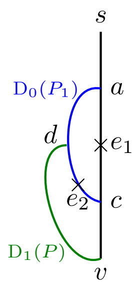

Figure 1 Showing P_0 (in black), $D_0(P_1)$ (in blue) and $D_1(P)$ (in green). Here $P_1 = P_0[s,a] \cup D_0(P_1) \cup P_0[c,v]$ and $P = P_0[s,a] \cup D_0(P_1)[a,d] \cup D_1(P)$.

1.3 Outline of the paper

We now present a brief outline of our paper. In Section 2, we present the basic notations that shall be used throughout the paper, which shall be followed by a brief overview of our approach and analysis in Section 3. In Section 4, we shall first begin with the description of our algorithm for dual FT-BFS and the properties of the shortest paths found using it, which shall be followed by the formal analysis. We then present our algorithm for dual FT-MBFS and its analysis, drawing similarities with solution of dual FT-BFS. Finally, we present the concluding remarks for our paper in Section 6. Due to page constraints some proofs have been omitted and deferred to the full paper [11]. For the sake of simplicity, we only describe our algorithm and analysis for edge failures. However, the same analysis can also be used to handle vertex failures.

2 Preliminaries

Given a graph $G = (V, E)$ with n vertices and m edges with a set of designated source $s \in S$. The following notations shall be used throughout the paper.

- $P, \mathcal{P}$: A path is denoted by P, where $Source(P)$ and $Dest(P)$ represents the source and destination of path P. In most parts of the paper, $Source(P) = s$ and $Dest(P) = v$. A set of paths is denoted by $\mathcal{P}$. Generally, we assume a path from s to v starts from the top (s) and ends at bottom (v). For two paths P', P'', we say P' leaves *earlier/higher* (or *later/lower*) than P'' from P, if P' leaves P closer to s (or closer to v) than P''.
- $F(P)$: For the shortest path P from $Source(P)$ to $Dest(P)$ after a set of edge failures, this set of failed edges is denoted by $F(P) = \{e_1, e_2, \ldots, e_k\}$ (say), where e_i denotes the i^{th} edge in the sequence. Similarly for some path P', e'_i denotes the i^{th} edge in the sequence.
- P_i: If $F(P) = \{e_1, e_2, \ldots, e_k\}$, then P_i is the shortest path avoiding the first i edge of $F(P)$, i.e., $F(P_i) = \{e_1, e_2, \ldots, e_i\}$, where $0 \le i < k$. Again, for most parts of the paper, P_0 denotes the shortest path from s to v in G.
- $D_i(P)$: If $|F(P)| = k$, the detour path of P from P_i, $D_i(P) = P \setminus \{\cup_{j=0}^{i} P_j\}$ [2], where $1 \le i < k-1$. For dual case, $D_0(P)$ is the detour of P from P_0, $D_1(P)$ is the detour of P from P_1, and $D_0(P_1)$ is the detour of P_1 from P_0 (See Figure 1).
- $LastE(P)$: The last edge of a path P.
- $P[x,y]$: The sub-path of P starting from x to y, where $x, y \in P$.

[2] This construction may give a set of disjoint subpaths of P instead of a single subpath. However, in most cases this path will be a single subpath, else we assume $D_i(P)$ to be the last such subpath on P.

We define the property of *convergence* of a set of paths $\mathcal{P}$ as follows. The paths in $\mathcal{P}$ are said to be *converging* if on intersection of any two paths $P, P' \in \mathcal{P}$, both P and P' merge and do not diverge till the end of the paths.

3 Overview

For analyzing the size of dual FT-BFS subgraph, i.e., the number of edges in shortest paths from the source s to each vertex $v \in V$ after any two failures, it suffices to count only the last edge of every such path P, for each $v \in V$ [16, 15]. The novelty of our approach is the classification of such paths based on interaction of corresponding P_1 and P_0, whereas Parter [15] studied the different interactions of P_1 and P_1', for two such paths P and P'.

We primarily use the disjointness of a designated suffix of such a path P (referred as *LastLeg*(P)) with *counting* arguments to bound the number of such paths. To achieve this, we classify some of these paths as *standard paths* based on the interactions of corresponding P_1 and P_0. The number of *non-standard* paths can be easily bound using simple *counting* arguments. The set of *standard* paths exhibit several interesting properties including convergence of corresponding paths $\mathrm{D}_0(P_1)$. We further classify the *standard* paths into *long standard* paths and *short standard* paths, each bounded separately using relatively harder techniques. For sake of easier presentation we first bound the number of *short standard* paths only for undirected graphs, with extension to directed graphs requiring an additional notion of *segmentable* paths. The only difference in the analysis of dual FT-MBFS is the definition of *standard* paths and dealing with interaction of P_1 with P_0' corresponding to other sources.

4 Dual FT-BFS

We shall now describe our algorithm to compute sparse dual FT-BFS subgraph H from a source $s \in V$. For every vertex $v \in V$, our algorithm computes the shortest paths from s to v avoiding upto two failures and adds the last edge of each such path to the adjacency list of vertex v. Note that repeating the procedure for each vertex on such a path adds the entire path to H [16, 15].

Our algorithm starts by adding the shortest path between s and v, i.e., P_0. It then processes single edge failures on P_0. We then find the replacement path P for all two edge failures $\{e_1, e_2\}$ such that $e_1 \in P_0$ and $e_2 \in P_1$. Further, in case $e_2 \in P_0 \cap P_1$ then e_1 is higher than e_2 on P_0.

However, we want to process all the failures in some particular order. This ordering plays a crucial role in the analysis. To this end, we define this ordering π as follows. The first failure in π is $F = \emptyset$, which adds P_0. The ordering π then contains single edge failures of type $F = \{e\}$ (where $e \in P_0$), ordered by their decreasing distance from s on P_0. Finally, we order the remaining failures as follows: for any two failures $F = \{e_1, e_2\}$ and $F' = \{e_1', e_2'\}$ (with corresponding replacement paths P and P'), $F \prec_\pi F'$ if either (1) e_1 is farther than e_1' from s on P_0, or (2) $e_1 = e_1'$ and e_2 is farther than e_2' from s on P_1 (note that $P_1 = P_1'$ in this case). If $F \prec_\pi F'$, F is said to be *lower* than F' in π.

For any failure of $F = \{e_1, \cdots, e_k\}$, we define the *preferred* shortest path avoiding F. Our preferred shortest path will be a path of shortest length avoiding F. However, there can be multiple such paths of same length. We use following rules to choose a unique preferred path.

Procedure Dual-FT-BFS(s, v, π): Augments the dual FT-BFS subgraph H, such that for BFS tree of G rooted at s after any two edge failures in G, the incoming edges to v are preserved in H.

foreach *Failure* F, *where* $0 \leq |F| \leq 2$, *ordered from lower to higher in* π **do**
    $P \leftarrow$ *Preferred* path from s to v in G avoiding F;
    if $LastE(P) \notin H$ **then**
        Assign P for failure of F;
        Add $LastE(P)$ to H;
    end
end

▶ **Definition 5.** Path P is **preferred** for failure of $\{e_1, \cdots, e_k\}$ where each $e_i \in P_{i-1}$, if

1. For each i, P leaves P_{i-1} before e_i exactly once.
2. For any other P' also avoiding $\{e_1, \cdots, e_i\}$, we have either (i) $|P| < |P'|$, (ii) $|P| = |P'|$, and for some $0 \leq i \leq k$, both P and P' leaves each of $P_0, ..., P_{i-1}$ at the same vertex, but P leaves P_i earlier than P', (iii) P is lexicographically smaller [3] than P'.

Intuitively, out of all the shortest paths avoiding F (say for $|F| = 2$), the preferred path leaves the path P_0 and/or P_1 as early as possible. In order to avoid the preferred path leaving P_0 (or P_1) multiple times just to achieve an earlier point of divergence from P_0 (or P_1), the first condition is imposed. The last condition in (2) is just to break ties between two paths that are of same length and leave P_0 and P_1 at the same vertex.

Finally, in order to add the preferred shortest path P avoiding a failure F, our algorithm simply adds $LastE(P)$ to H, which suffices to add the entire path as described earlier. Moreover, we also assign the corresponding P to the failure F if it was the first failure to add this edge in H. As a result, if P and P' are two preferred paths avoiding F and F' respectively where $LastE(P) = LastE(P')$, then if $F \prec_\pi F'$, only the path P shall be assigned to F. Refer to Procedure Dual-FT-BFS for the pseudocode of our algorithm.

In order to calculate the size of H, it is sufficient to analyze the number of different last edges added on each $v \in V$ in H. Let the set of all paths from s to v avoiding failures $F \subseteq E$ (where $|F| \leq 2$) be $\mathcal{P}_v$. We thus define the paths that will be counted for establishing the space bound as follows.

▶ **Definition 6.** The path $P \in \mathcal{P}_v$ is called *contributing* if while processing $\mathrm{F}(P)$, $LastE(P) \notin H$, i.e., P adds a new edge adjacent to v in H.

In order to count the number of contributing paths to a vertex v, we only need to consider its interactions with other contributing paths in $\mathcal{P}_v$. This is because, if any other path $P \in \mathcal{P}_x$ passes through v using some new edge, so does the corresponding $P' \in \mathcal{P}_v$ with $\mathrm{F}(P) = \mathrm{F}(P')$. Thus, to analyze the size of H, it suffices to look at last edges of the contributing paths in $\mathcal{P}_v$ for each vertex v separately.

[3] Let P and P' first diverge from each other to $x \in P$ and $x' \in P'$ respectively, i.e., $P[s, x] \setminus \{x\} = P'[s, x'] \setminus \{x'\}$. If the index of x is lower than that of x' then P is said to be *lexicographically smaller* than P'.

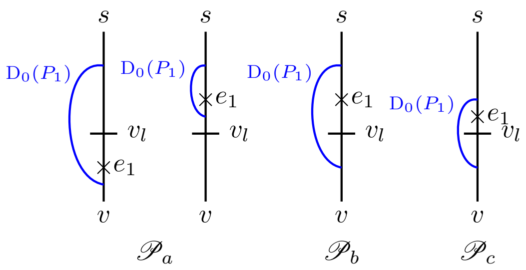

Figure 2 Classification of contributing paths: $\mathscr{P}_a$: Non-Standard Paths, $\mathscr{P}_b$: Long Standard Paths and $\mathscr{P}_c$: Short Standard Paths.

4.1 Properties of contributing paths

Parter [15] presented a simple proof bounding the number of contributing paths avoiding multiple failures on P_0 to $O(\sqrt{n})$ for each vertex v (an alternate proof using *counting arguments* is presented in the full paper [11]). Hence, excluding these paths, every contributing path satisfies the following properties (see full paper [11] for proofs).

▶ **Lemma 7.** *Excluding $O(\sqrt{n})$ paths, each contributing path P from s to v avoiding $\{e_1, e_2\}$ satisfies following properties*

$\mathbb{P}_1$: *$e_1 \in P_0$ and $e_2 \in \mathrm{D}_0(P_1)$.*

$\mathbb{P}_2$: *Except at v, $\mathrm{D}_0(P)$ does not intersect with P_0 and $\mathrm{D}_1(P)$ does not intersect with P_1, after diverging from P_0 and P_1 respectively.*

$\mathbb{P}_3$: *For any path P' which avoids $\{e_1, e_2\}$, P is the preferred path over P'.*

$\mathbb{P}_4$: *If P also avoids some failure F' where $F' \prec_\pi F$, then there exist another path P' which is the preferred path for F' over P, where P' does not avoid F.*

4.2 Space Analysis

As described earlier, in order to bound the size of dual FT-BFS subgraph to $O(n^{5/3})$, it suffices to bound the number of *contributing* paths from s to each vertex $v \in V$ avoiding two edge failures to $O(n^{2/3})$. Further, using $\mathbb{P}_1$ we are only concerned with a contributing path P if $e_1 \in P_0$ and $e_2 \in \mathrm{D}_0(P_1)$.

We first divide the path P_0 into two parts as follows. Let $v_l \in P_0$ be the vertex such that $|P_0[v_l, v]| = n^{1/3}$. We define $P_{high} = P_0[s, v_l]$ and $P_{low} = P_0[v_l, v]$. If $|P_0| < n^{1/3}$, we assume $v_l = s$ where $P_{high} = \phi$. We shall now define the *standard paths* as follows.

▶ **Definition 8** (Standard Paths)**.** A contributing path P is called a *standard path* if (a) $e_1 \in P_{high}$, and (b) $\mathrm{D}_0(P_1)$ merges with P_0 on P_{low}, i.e., $Dest(\mathrm{D}_0(P_1)) \in P_{low}$.

We can thus classify the contributing paths into following three types (see Figure 2):

$\mathscr{P}_a$: Non-standard paths.

$\mathscr{P}_b$: Long standard paths, i.e., standard paths with $|\mathrm{D}_0(P_1)| \geq n^{2/3}$.

$\mathscr{P}_c$: Short standard paths, i.e., standard paths with $|\mathrm{D}_0(P_1)| < n^{2/3}$.

Clearly, the sets $\mathscr{P}_a$, $\mathscr{P}_b$ and $\mathscr{P}_c$ are mutually disjoint and collectively exhaustive. Further, we define a set $\mathscr{P}_{1x}$ (for $x = a, b$ and c), where for each $P \in \mathscr{P}_x$, we add the corresponding P_1 to $\mathscr{P}_{1x}$. In addition, we identify the disjoint suffix of a path P as follows (see Figure 3).

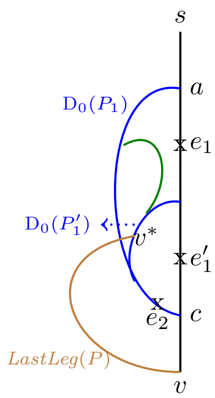

Figure 3 P avoids $\{e_1, e_2\}$. Its detour $D_1(P)$ (shown in blue) last intersects $LastPath(P) = P'_1$. P diverges from P'_1 at v^*, i.e., $LastLeg(P) = P[v^*, v]$ (shown in brown).

▶ **Definition 9.** For each $P \in \mathscr{P}_x$, for $x = a, b$ or c, we define the following

1. $LastPath(P)$: The path in $\mathscr{P}_{1x}$ that intersects last with P. If P diverges from P_0 and does not intersect any path in $\mathscr{P}_{1x}$, we set $LastPath(P) = P_0$.
2. $LastLeg(P)$: The part of P after diverging from $LastPath(P)$, i.e., $P[v^*, v]$, where v^* is the last vertex of P on $P \cap LastPath(P)$.

The suffix $LastLeg(P)$ of a contributing path P satisfies the following properties (see full paper [11] for proofs).

▶ **Lemma 10.** *For every set $\mathscr{P}_x$ (for $x = a, b$ or c), we have the following.*

(a) *For any $P, P' \in \mathscr{P}_x$, $LastLeg(P)$ and $LastLeg(P')$ are disjoint (except at v), i.e., $LastLeg(P) \cap LastLeg(P') = \{v\}$. Further, each P, P' starts from a distinct vertex on $\mathscr{P}_{1x}$.*

(b) *Number of paths $P \in \mathscr{P}_x$ with $|LastLeg(P)| > n^{1/3}$ or $LastPath(P) = P_0$, is $O(n^{2/3})$.*

Remark: Lemma 102 claims that $LastLeg(P)$ is disjoint from other $LastLeg(P')$, where $P \in \mathscr{P}_x$ and $P' \in \mathscr{P}_{x'}$ only when $x = x'$. However, in case $x \neq x'$ they can intersect and our proof does not require their disjointness.

Equipped with these properties we can easily analyze the number of *non-standard paths* ($\mathscr{P}_a$) and *standard paths* ($\mathscr{P}_b$ and $\mathscr{P}_c$) in the following sections.

4.2.1 Analyzing non-standard paths $\mathscr{P}_a$

Using Lemma 102, we know that the number of $P \in \mathscr{P}_a$ with $|LastLeg(P)| > n^{1/3}$ or $LastPath(P) = P_0$ is $O(n^{2/3})$. We now focus on the case when $|LastLeg(P)| \leq n^{1/3}$ and $LastPath(P) \in \mathscr{P}_{1a}$. For any path P, let $v^* = Source(LastLeg(P))$. Since $LastLeg(P)$ is a detour from $LastPath(P)[v^*, v]$ avoiding the entire P_0 (using $\mathbb{P}_2$), we have $|LastPath(P)[v^*, v]| \leq |LastLeg(P)| \leq n^{1/3}$. By definition, a contributing path P is *non-standard* if either (a) $e_1 \in P_{low}$, or (b) $D_0(P_1)$ merges with P_0 on P_{high}, i.e., $Dest(D_0(P_1)) \in P_{high}$. Hence, for every P, $LastPath(P)$ would correspond to one of the two cases (a) or (b). Case (b) is clearly not be applicable here because $|LastPath(P)[v^*, v]| \geq |P_{low}| = n^{1/3}$ (since $Dest(LastPath(P)) \in P_{high}$). For case (a), on each $LastPath(P) \in \mathscr{P}_{1a}$, v^* can be one of $n^{1/3}$ vertices of $LastPath(P)$ closest to v. Further, since $e_1 \in P_{low}$, there are only $n^{1/3}$ such paths in $\mathscr{P}_{1a}$ because each such path corresponds to failure of unique edge in P_{low}. Thus, there are only $n^{1/3} \times n^{1/3} = n^{2/3}$ different vertices v^* limiting the number of $P \in \mathscr{P}_a$ with $|LastLeg(P)| \leq n^{1/3}$ to $O(n^{2/3})$ (using Lemma 101).

Properties of standard paths ($\mathscr{P}_b$ or $\mathscr{P}_c$)

We shall now prove two important properties of standard paths (see full paper [11] for proofs). The first result states that if $\mathrm{D}_0(P_1)$ and $\mathrm{D}_0(P_1')$ intersect, where $P, P' \in \mathscr{P}_{1b} \cup \mathscr{P}_{1c}$, then they cannot diverge. The second result states that the length of paths in $\mathscr{P}_b \cup \mathscr{P}_c$ are different. A similar result was proved by Parter[15].

▶ **Lemma 11.** *For the set of contributing* standard *paths, we have the following properties.*

(a) *The set of paths* $\{\mathrm{D}_0(P_1) | P_1 \in \mathscr{P}_{1b} \cup \mathscr{P}_{1c}\}$*, is converging.*

(b) *(Parter [15]) For any two paths* $P, P' \in \mathscr{P}_b \cup \mathscr{P}_c$, $|P| \neq |P'|$.

4.2.2 Analyzing long standard paths $\mathscr{P}_b$

We first prove a generic technique to bound the number of contributing paths P if the set of corresponding paths P_1 is converging and each P_1 sufficiently long.

▶ **Theorem 12.** *Given a set* $\mathcal{P}$ *of converging paths satisfying Lemma 101, where for each* $P_1 \in \mathcal{P}$ *we have* $|P_1| \geq \alpha^2$ *(where* $\alpha \geq 1$*), the number of contributing paths* P *having* $P_1 \in \mathcal{P}$ *is* $O(n/\alpha)$*.*

Proof. Recall the definition of $LastPath(P)$, here we shall define $LastPath(P)$ (and hence $LastLeg(P)$) corresponding to paths in $\mathcal{P}$ (rather than $\mathscr{P}_{1x}$ in Definition 9). Using Lemma 101, if $|LastLeg(P)| \geq \alpha$, then P can be associated with α unique vertices of $LastLeg(P)$. This limits the total number of such paths to $O(n/\alpha)$. Hence, we assume that $LastLeg(P)) \leq \alpha$.

For each path $P_1 \in \mathcal{P}$, let $v_l = Dest(P_1)$. Similarly, for each such P, let the last intersection vertex of $LastLeg(P)$ and $LastPath(P)$ be v^*. Using Lemma 101, we know that for each such contributing path P, its corresponding $LastLeg(P)$ starts from a distinct vertex of $\mathcal{P}$. Since $LastLeg(P)$ is a detour from $LastPath(P)[v^*, v_l]$ avoiding the entire P_1 (using $\mathbb{P}_2$), we have $|LastLeg(P)| \geq |LastPath(P)[v^*, v_l]|$. Since $|LastLeg(P)| \leq \alpha$, v^* can be one of α vertices of $LastPath(P)$ closest to v_l.

We shall associate each such vertex v^* on $LastPath(P) \in \mathcal{P}$ uniquely with α vertices of $LastPath(P)$, for all $LastPath(P) \in \mathcal{P}$, as follows. Let the vertices of some $LastPath(P)$ be $v_1, ..., v_k$ where v_1 is the closest vertex to v_l. For each v_i, $i = 1, ..., \alpha$, we associate the vertices $v_{(i-1)\alpha}, ..., v_{i\alpha}$. Since $|LastPath(P)| \geq \alpha^2$ (by definition of $\mathcal{P}$) and $i \in [1, \alpha]$ such an association can be made. Now, in order to prove that such an association is unique, i.e., a vertex x is not associated with two different vertices v_1^*, v_2^* of $\mathcal{P}$, we exploit the convergence of $\mathcal{P}$ as follows. Clearly if $x \in P_1$ for a unique path $P_1 \in \mathcal{P}$, there is a unique $v_1^* \in \mathcal{P}$ to which it is associated. However, if $x \in P_1$ and $x \in P_1'$ for any two paths $P_1, P_1' \in \mathcal{P}$, then P_1 and P_1' will not diverge after intersection (by convergence of $\mathcal{P}$). This implies $P_1[x, v_l] = P_1'[x, v_l']$. Thus, the corresponding $v_1^* \in P_1$ and $v_2^* \in P_1'$ would also be same as by definition $v_1^* \in P_1[x, v_l]$. Hence, for every P emerging from v^* with $|LastPath(P)[v^*, v_l]| \leq \alpha$, the corresponding v^* can be uniquely associated with at least α vertices of $\mathcal{P}$. This limits the total number of such paths to $O(n/\alpha)$ proving the theorem. ◀

Using Lemma 111 and by definition of long standard paths $\mathscr{P}_b$, Theorem 12 is applicable for the set $\mathrm{D}_0(P_1)$ for $P_1 \in \mathscr{P}_{1b}$ and $\alpha = n^{1/3}$ limiting the number of paths in $\mathscr{P}_b$ to $O(n^{2/3})$.

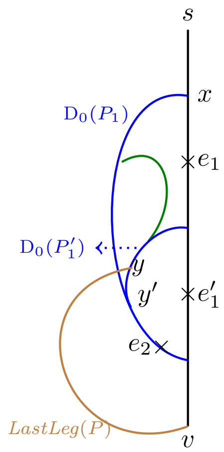

Figure 4 Let P_1' be $\mathit{LastPath}(P)$. Then the path $P_0[s,x] \cup P_1[x,y'] \cup P_1'[y',y] \cup \mathit{LastLeg}(P)$ is a valid path avoiding $\{e_1, e_2\}$.

4.2.3 Analyzing short standard paths $\mathscr{P}_c$

To highlight the simplicity of our approach, we only analyze the paths in $\mathscr{P}_c$ for undirected graphs here. For extension of this proof to handle directed graphs we use the theory of *segmentable paths* (refer to full paper [11] for details).

Using Lemma 102, we know that the number of $P \in \mathscr{P}_c$ with $|\mathit{LastLeg}(P)| > n^{1/3}$ or $\mathit{LastPath}(P) = P_0$ is $O(n^{2/3})$. We now focus on the case when $|\mathit{LastLeg}(P)| \leq n^{1/3}$ and $\mathit{LastPath}(P) \in \mathscr{P}_{1c}$. Any such contributing path P can be divided into two parts (see Figure 4), (a) $P[s,y]$, where $y = \mathit{Source}(\mathit{LastLeg}(P))$, and (b) $P[y,v] = \mathit{LastLeg}(P)$. We will now find an alternate path for $P[s,y]$, which will help us in bounding its length. Since P is a contributing path, it diverges from $\mathit{LastPath}(P)$ which requires either e_1 or e_2 to be on $\mathit{LastPath}(P)[y,v]$. By definition of *standard paths*, we have $\mathrm{D}_0(\mathit{LastPath}(P))$ terminates on P_0 only on P_{low}, whereas $e_1 \notin P_{low}$ ensuring that $e_1 \notin \mathit{LastPath}(P)$. Thus, $e_2 \in \mathit{LastPath}(P)[y,v]$ and hence it intersects with P_1 as $e_2 \in P_1$. Using Lemma 111, we can thus say that $\mathit{LastPath}(P)$ and P_1 merge at some vertex say y', where $e_2 \in \mathit{LastPath}(P)[y',v] = P_1[y',v]$ (see Figure 4). We have an alternate path for $P[s,y]$ avoiding $\mathrm{F}(P)$ formed by $P_1[s,y'] \cup \mathit{LastPath}(P)[y',y]$. Let $x = \mathit{Source}(\mathrm{D}_0(P_1))$. Since $P[s,v]$ is the shortest path avoiding $\mathrm{F}(P)$ we have

$$
\begin{aligned}
|P| &= |P[s,y]| + |P[y,v]| \\
&= |P_1[s,y]| + |P[y,v]| \\
&\leq |P_1[s,y']| + |\mathit{LastPath}(P)[y',y]|) + |\mathit{LastLeg}(P)[y,v]| \\
&= (|P_1[s,x]| + |P_1[x,y']|) + |\mathit{LastPath}(P)[y',y]| + |\mathit{LastLeg}(P)[y,v]| \\
&\leq |P_0| + |\mathrm{D}_0(P_1)| + |\mathrm{D}_0(\mathit{LastPath}(P))| + |\mathit{LastLeg}(P)| \\
&\leq |P_0| + n^{2/3} + n^{2/3} + n^{1/3} \qquad \text{(by definition of } \mathscr{P}_c\text{)}
\end{aligned}
$$

Now, using Lemma 112, we know that for any $P, P' \in \mathscr{P}_c$ we have $|P| \neq |P'|$. We thus arrange the paths in $\mathscr{P}_c$ (except the ones in Lemma 102) in the increasing order of sizes, where i^{th} such path has the length $\geq |P_0| + i$ (as all paths at least as long as P_0). Since for any such $P \in \mathscr{P}_c$ we also have $|P| \leq |P_0| + 3n^{2/3}$ (described above), clearly the number of paths in $\mathscr{P}_c$ are $O(n^{2/3})$ (for i upto $3n^{2/3}$).

Procedure Dual-FT-MBFS(S,v,π): Augments the dual FT-MBFS subgraph H, such that for BFS tree of G rooted at each $s \in S$ after any two edge failures in G, the incoming edges to v are preserved in H.

```
foreach s ∈ S do  Dual-FT-BFS(s,v,π_s(0)) ;
foreach s ∈ S do  Dual-FT-BFS(s,v,π_s(1)) ;
foreach s ∈ S do  Dual-FT-BFS(s,v,π_s(2)) ;
```

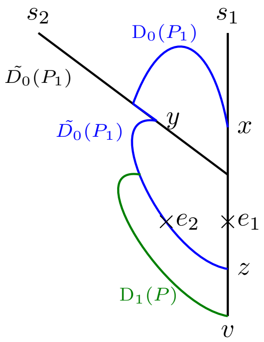

Figure 5 Shortest path avoiding $\{e_1, e_2\}$ is P. $D_1(P)$ last intersects $\tilde{P_0}(P_1) = P_0(s_2, v)$. P_1 diverges from $\tilde{P_0}(P_1)$ at y, i.e., $\tilde{D_0}(P_1) = P_1[y, z]$ (shown in blue). P also diverges from $\tilde{P_0}(P)$ at y, i.e., $\tilde{D_0}(P) = P[y, v]$.

This completes the proof of our dual FT-BFS result in Theorem 3.

5 Extension to dual FT-MBFS

In this section we shall extend our analysis of the previous section to handle σ sources using total $O(\sigma^{1/3} n^{5/3})$ space. We follow the approach similar to the case for single source. Let S be the set of sources, where $|S| = \sigma$. Given a source s, let $\pi_s \subseteq \pi$ denote the ordering of edge failure of size upto 2. Let $\pi_s(0), \pi_s(1)$ and $\pi_s(2)$ be the subset of π_s of size 0, 1 and 2 respectively. Our algorithm for finding dual FT-MBFS mimics the single source case.

The first for loop in the above procedure finds shortest path from each source to v. We shall refer to the set of the shortest paths from each source to v for different $s \in S$ as $\mathscr{P}_0$. We then move on to find the shortest path from each source to v avoiding one edge failure and two failures respectively.

In the previous section, for each contributing path P (that avoids ≥ 1 failure), we saw that it necessarily diverges from P_0. Since we have multiple paths in $\mathscr{P}_0$, we define some new notations (see Figure 5).

▶ **Definition 13** (Modified P_0 and $D_0(P)$).

1. For any path P (or its corresponding P_1), we define $\tilde{P_0}(P)$ (or $\tilde{P_0}(P_1)$) to be the last path from $\mathscr{P}_0$ which intersects with P (or P_1), say at vertex y, such that at least one of e_1 or e_2 is present in $\tilde{P_0}(P)[y, v]$ (or $e_1 \in \tilde{P_0}(P_1)[y, v]$).
2. For any path P (or P_1), we define $\tilde{D_0}(P) = P[v^*, v]$ (or $\tilde{D_0}(P_1) = P_1[v_0^*, v]$), where v^* is the last vertex of P (or P_1) on $P \cap \tilde{P_0}(P)$ (or $P_1 \cap \tilde{P_0}(P_1)$).

Note that in the single source case, both P and P_1 diverge from the same path P_0. However, in multiple source that path $\tilde{P_0}(P)$ and $\tilde{P_0}(P_1)$ may differ. This is one of the major changes from the single source case. In fact, the reader will see that all our lemmas in Section 4 extend here with P_0 changed to $\tilde{P_0}(P)$ and $\mathrm{D}_0(P)$ changed to $\tilde{D_0}(P)$. However, for completeness we have re-proven all lemmas.

5.1 Properties of Contributing paths

We now describe important properties of paths in $\mathscr{P}_0$ and contributing paths as follows (see full paper [11] for proofs).

▶ **Lemma 14.** *The set of paths $\mathscr{P}_0$ is converging.*

The number of contributing paths avoiding failures in $\mathscr{P}_0$ can easily bounded to $O(\sqrt{\sigma n})$ for each v (see full paper [11] for details). Excluding these paths, every contributing path satisfies the following properties.

▶ **Lemma 15.** *Excluding $O(\sqrt{\sigma n})$ paths, for any contributing path P from s to v avoiding $\{e_1, e_2\}$, the following properties holds true*

$\mathbb{P}_1$: *$e_1 \in P_0$ and $e_2 \in \mathrm{D}_0(P_1)$.*

$\mathbb{P}_2$: *$\tilde{D_0}(P)$ does not intersect with any path in $\mathscr{P}_0$. Also, if $\tilde{D_0}(P)$ diverges from P_1 it does not intersect it again.*

5.2 Space Analysis

As described earlier, in order to bound the size of dual FT-MBFS subgraph to $O(\sigma^{1/3}n^{5/3})$, it suffices to bound the number of *contributing* paths from $s \in S$ to each vertex $v \in V$ avoiding two edge failures to $O(\sigma^{1/3}n^{2/3})$. Further, using $\mathbb{P}_1$ we are only concerned with a contributing path P if $e_1 \in P_0$ and $e_2 \in \mathrm{D}_0(P_1)$. For the sake of highlighting similarity with single source case, we shall use $n_\sigma = n/\sigma$ throughout the section.

We first divide the paths in $\mathscr{P}_0$ into two parts as follows. For each $s \in S$, let $P_0(s,v)$ be the shortest path from s to v. Let v_{ls} be the vertex such that $|P_0(s,v)[v_{ls}, v]| = n_\sigma^{1/3}$. We define $\mathcal{P}_{low} = \{P_0(s,v)[v_{ls}, v] \mid s \in S\}$ and $\mathcal{P}_{high} = \{P_0(s,v)[s, v_{ls}] \mid s \in S\}$. This definition naturally extends the P_{low} and P_{high} defined in the single source case.

With this modified $\mathcal{P}_{low}$ and $\mathcal{P}_{high}$, we use the same definition of *standard paths* and hence $\mathscr{P}_a$ and $\mathscr{P}_{1a}$. However, the distinction of *long standard paths* ($\mathscr{P}_b$) from *short standard paths* ($\mathscr{P}_c$) would now be done by using $\tilde{D_0}(P_1)$ instead of $D_0(P_1)$. Hence, the *long standard paths* would be the standard paths with $|\tilde{D_0}(P_1)| \geq n_\sigma^{2/3}$. Finally, the definition of *LastPath*(P) and *LastLeg*(P) does not change, except in case *LastPath*$(P) = \phi$, we use *LastPath*$(P) = \tilde{P_0}(P)$ instead of *LastPath*$(P) = P_0$ (recall Definition 9). Moreover, the properties of *LastLeg*(P) also remain same except for Lemma 102 which is modified as follows.

▶ Lemma 10. For every set $\mathscr{P}_x$ (for $x = a, b$ or c), we have the following.

b^***.** Number of paths $P \in \mathscr{P}_x$ with $|LastLeg(P)| > n_\sigma^{\frac{1}{3}}$ or $LastPath(P) = \tilde{P_0}(P)$, is $O(\sigma^{\frac{1}{3}} n^{\frac{2}{3}})$.

Now, using the properties described in Lemma 10 (see full paper [11] for proof), we can analyze the number of *non-standard paths* ($\mathscr{P}_a$) using the same *counting arguments* as in case of single source, bounding the number of such paths to $O(\sigma^{1/3}n^{2/3})$ (see full paper [11] for details). Hence, we only focus on analyzing the *standard paths* ($\mathscr{P}_b$ and $\mathscr{P}_c$) as follows.

Properties of standard paths ($\mathscr{P}_b$ and $\mathscr{P}_c$)

Recall the properties of *standard paths* described in Lemma 11. For multiple sources, Lemma 111 does not hold, because for two paths P and P', their corresponding paths $\mathrm{D}_0(P_1)$

and $\mathrm{D}_0(P_1')$ can diverge after intersection, if they start from different sources say s_1, s_2 (for $s_1 = s_2$, Lemma 111 applies). For example (see Figure 5), P_1 avoids e_1 on $P_0[s_1, v]$. Also, $\mathrm{D}_0(P_1)$ passes through $P_0[s_2, v]$. Let P_1' be a path avoiding e_1' on $P_0[s_2, v] \cap P_1$, such that $\mathrm{D}_0(P_1')$ intersect $\mathrm{D}_0(P_1)$ before $\mathrm{D}_0(P_1)$ enters $P_0[s_2, v]$. Hence, $\mathrm{D}_0(P_1')$ has to diverge from $\mathrm{D}_0(P_1)$ as $\mathrm{D}_0(P_1)$ passes through e_1' after the intersection.

This is the primary reason for defining *modified detour* $\tilde{D}_0(P_1)$, for which a lemma equivalent to Lemma 111 holds. Thus, the analysis of *standard paths* for multiple sources, uses $\tilde{D}_0(P_1)$ instead of $D_0(P_1)$ satisfying the following properties.

▶ **Lemma 16.** *For the set of contributing* standard *paths, we have the following properties.*

(a) *The set of paths* $\{\tilde{D}_0(P_1) | P_1 \in \mathscr{P}_{1b} \cup \mathscr{P}_{1c}\}$*, is converging.*

(b) *The number of paths* $P \in \mathscr{P}_b \cup \mathscr{P}_c$*, which* $Source(LastLeg(P)) \notin \tilde{D}_0(P_1')$ *for some* $P_1' \in \mathscr{P}_{1b} \cup \mathscr{P}_{1c}$ *are* $O(\sigma^{1/3} n^{2/3})$.

Using Lemma 162, we only have to bound the number of *standard paths* whose $LastLeg(P)$ originates from some $\tilde{D}_0(P_1')$. Using Lemma $10b^*$and by definition of long standard paths $\mathscr{P}_b$, Theorem 12 is applicable for the set $\tilde{D}_0(P_1)$ for $P_1 \in \mathscr{P}_{1b}$ and $\alpha = n_\sigma^{1/3}$, bounding number of such paths in $\mathscr{P}_b$ to $O(\sigma^{1/3} n^{2/3})$. This leaves only the number of *short standard paths* that originate from some $\tilde{D}_0(P_1')$ described in the following section.

5.2.1 Analyzing short standard paths $\mathscr{P}_c$

Again, we only analyze the paths in $\mathscr{P}_c$ for undirected graphs here (see full paper [11] for directed graphs). Using Lemma $10b^*$and Lemma 162, we know that the number of $P \in \mathscr{P}_c$ with $|LastLeg(P)| > n_\sigma^{1/3}$ or $LastPath(P) = \tilde{P}_0(P)$ or $Source(LastLeg(P)) \notin \tilde{D}_0(P_1')$ (for some $P_1' \in \mathscr{P}_{1c}$) is $O(\sigma^{1/3} n^{2/3})$. We thus focus on the case when $|LastLeg(P)| \le n_\sigma^{1/3}$ and $LastPath(P) \in \mathscr{P}_{1c}$ with $Source(LastLeg(P)) \in \tilde{D}_0(LastPath(P))$. Any such path can be divided into three parts (not necessarily non-empty) including (a) $P[s, x] = P_1[s, x]$, where $x = Source(\tilde{D}_0(P_1))$, (b) $P[x, y]$ where $y = Source(LastLeg(P))$ and (c) $P[y, v] = LastLeg(P)$.

We find alternate paths for $P[s, x]$ and $P[x, y]$, which will help us in bounding their respective lengths (see Figure 6). By definition $\tilde{P}_0(P_1)$ intersects with P_0 and passes through e_1. Further, using Lemma 14 we know that P_0 and $\tilde{P}_0(P_1)$ will merge after the intersection at some point, say x', where $e_1 \in \tilde{P}_0(P_1)[x', v] = P_0[x', v]$. Hence, we have an alternate path for $P_1[s, x]$ avoiding e_1 and e_2 (since $e_2 \notin \mathscr{P}_0$ by $\mathbb{P}_2$) formed by $P_0[s, x'] \cup \tilde{P}_0(P_1)[x', x]$. Now, bounding $P[x, y]$ is exactly same as in the case of single source, using $\tilde{D}_0(P_1)$ instead of $\mathrm{D}_0(P_1)$, bounding $P[x, y]$ to $2n_\sigma^{2/3}$ as shown in Figure 6 (see full paper [11] for an exhaustive proof). Since $P[s, v]$ is the shortest path avoiding $\mathrm{F}(P)$ we have

$$\begin{aligned}
|P| &= |P_0[s,x]| + |P_1[x,y]| + |P[y,v]| && \text{(by definition of } x \text{ and } y\text{)}\\
&\le (|P_0[s,x']| + |\tilde{P}_0(P_1)[x',x]|) + 2n_\sigma^{2/3} + n_\sigma^{1/3} && \text{(Similar to dual FT-BFS)}\\
&\le |P_0[s,v]| + |\tilde{P}_0(P_1)[x,v]| + 2n_\sigma^{2/3} + n_\sigma^{1/3}\\
&\le |P_0[s,v]| + |\tilde{D}_0(P_1)[x,v]| + 2n_\sigma^{2/3} + n_\sigma^{1/3}\\
& \qquad (\tilde{D}_0(P_1) \text{ is a detour from } \tilde{P}_0(P_1)\text{, hence } |\tilde{D}_0(P_1)[x,v]| > |\tilde{P}_0(P_1)[x,v]|)\\
&\le |P_0[s,v]| + n_\sigma^{2/3} + 2n_\sigma^{2/3} + n_\sigma^{1/3} && \text{(by definition of } \mathscr{P}_c\text{)}
\end{aligned}$$

Now, for any $s \in S$, let $\mathscr{P}_c(s)$ be the set of all contributing paths in $\mathscr{P}_c$ that start from s. Using Lemma 112 (that holds for $P \in \mathscr{P}_c(s)$), we know that for any $P, P' \in \mathscr{P}_c(s)$ we have $|P| \neq |P'|$. We thus arrange the paths in $\mathscr{P}_c(s)$ (except the ones in Lemma $10b^*$and

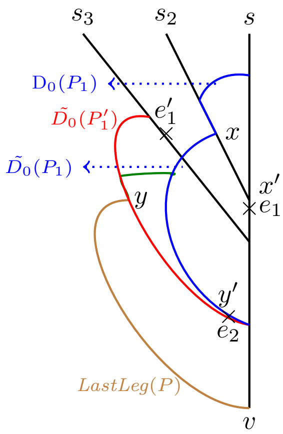

Figure 6 Let $\tilde{P}_0(P_1) = P_0(s_2, v)$, $LastPath(P) = P_1'$ and $\tilde{P}_0(P_1') = P_0(s_3, v)$. Then the path $P_0[s, x'] \cup \tilde{P}_0(P_1)[x', x] \cup \tilde{D}_0(P_1)[x, y'] \cup \tilde{D}_0(P_1')[y', y] \cup LastLeg(P)$ is a valid path avoiding $\{e_1, e_2\}$.

Lemma 162) in the increasing order of sizes, where i^{th} such path has the length $\geq |P_0(s, v)| + i$ (as all paths at least as long as $P_0(s, v)$). Since for any such $P \in \mathscr{P}_c(s)$ we also have $|P| \leq |P_0[s, v]| + 4n^{2/3}$ (described above), clearly the number of paths in $\mathscr{P}_c(s)$ are $O(n_\sigma^{2/3})$ (for i upto $4n^{2/3}$). Hence, overall the number of paths in $\mathscr{P}_c$ considering all sources $s \in S$ are $O(\sigma * n_\sigma^{2/3}) = O(\sigma^{1/3} n^{2/3})$.

This completes the proof of Theorem 4.

6 Conclusion

In this paper, we simplified the analysis in [15] for dual FT-BFS problem and extended it to dual FT-MBFS problem. Unfortunately, extending our result to k FT-MBFS (or even k FT-BFS) problem requires a lot of case analysis. Ideally, one would wish to design a simple data structure to handle multiple failures using some new insight with little or no case analysis. A natural step would be to completely understand these simple cases and derive significant inferences from them to develop new techniques. The simplicity of FT-BFS structure [16] enables a clear understanding of the basic technique used for its construction and analysis. Our work aims to be a significant step to achieve the same for dual FT-BFS by simplifying the result of [15] and generalizing it similar to [16].

References

1 Surender Baswana, Shreejit Ray Chaudhury, Keerti Choudhary, and Shahbaz Khan. Dynamic DFS in Undirected Graphs: breaking the O(m) barrier. In *Proceedings of the Twenty-Seventh Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2016, Arlington, VA, USA, January 10-12, 2016*, pages 730–739, 2016.

2 Surender Baswana, Keerti Choudhary, and Liam Roditty. Fault tolerant reachability for directed graphs. In *Distributed Computing - 29th International Symposium, DISC 2015, Tokyo, Japan, October 7-9, 2015, Proceedings*, pages 528–543, 2015.

3 Surender Baswana, Keerti Choudhary, and Liam Roditty. Fault tolerant subgraph for single source reachability: generic and optimal. In *Proceedings of the 48th Annual ACM SIGACT Symposium on Theory of Computing, STOC 2016, Cambridge, MA, USA, June 18-21, 2016*, pages 509–518, 2016.

4 Davide Bilò, Fabrizio Grandoni, Luciano Gualà, Stefano Leucci, and Guido Proietti. Improved purely additive fault-tolerant spanners. In *Algorithms - ESA 2015 - 23rd Annual European Symposium, Patras, Greece, September 14-16, 2015, Proceedings*, pages 167–178, 2015.

5 Davide Bilò, Luciano Gualà, Stefano Leucci, and Guido Proietti. Multiple-edge-fault-tolerant approximate shortest-path trees. In *33rd Symposium on Theoretical Aspects of Computer Science, STACS 2016, February 17-20, 2016, Orléans, France*, pages 18:1–18:14, 2016.

6 Gilad Braunschvig, Shiri Chechik, David Peleg, and Adam Sealfon. Fault tolerant additive and (μ, α)-spanners. *Theor. Comput. Sci.*, 580:94–100, 2015.

7 Shiri Chechik, Michael Langberg, David Peleg, and Liam Roditty. Fault Tolerant Spanners for General Graphs. *SIAM J. Comput.*, 39(7):3403–3423, 2010. doi:10.1137/090758039.

8 Camil Demetrescu, Mikkel Thorup, Rezaul Alam Chowdhury, and Vijaya Ramachandran. Oracles for distances avoiding a failed node or link. *SIAM J. Comput.*, 37(5):1299–1318, 2008.

9 Michael Dinitz and Robert Krauthgamer. Fault-tolerant spanners: better and simpler. In *Proceedings of the 30th Annual ACM Symposium on Principles of Distributed Computing, PODC 2011, San Jose, CA, USA, June 6-8, 2011*, pages 169–178, 2011. doi:10.1145/1993806.1993830.

10 Ran Duan and Seth Pettie. Dual-failure distance and connectivity oracles. In *Proceedings of the Twentieth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2009, New York, NY, USA, January 4-6, 2009*, pages 506–515, 2009.

11 Manoj Gupta and Shahbaz Khan. Multiple Source Dual Fault Tolerant BFS Trees. *CoRR*, abs/1704.06907, 2017.

12 Neelesh Khanna and Surender Baswana. Approximate shortest paths avoiding a failed vertex: Optimal size data structures for unweighted graphs. In *27th International Symposium on Theoretical Aspects of Computer Science, STACS 2010, March 4-6, 2010, Nancy, France*, pages 513–524, 2010.

13 Thomas Lengauer and Robert Endre Tarjan. A fast algorithm for finding dominators in a flowgraph. *ACM Trans. Program. Lang. Syst.*, 1(1):121–141, 1979.

14 Merav Parter. Vertex Fault Tolerant Additive Spanners. In *Distributed Computing - 28th International Symposium, DISC 2014, Austin, TX, USA, October 12-15, 2014. Proceedings*, pages 167–181, 2014.

15 Merav Parter. Dual failure resilient BFS structure. In *Proceedings of the 2015 ACM Symposium on Principles of Distributed Computing, PODC 2015, Donostia-San Sebastián, Spain, July 21 - 23, 2015*, pages 481–490, 2015.

16 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.

17 Merav Parter and David Peleg. Fault tolerant approximate BFS structures. In *Proceedings of the Twenty-Fifth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2014, Portland, Oregon, USA, January 5-7, 2014*, pages 1073–1092, 2014.