Faster Algorithms for Dual-Failure Replacement Paths

Shiri Chechik
Tel Aviv University, Israel

Tianyi Zhang
Tel Aviv University, Israel

Abstract

Given a simple weighted directed graph $G = (V, E, \omega)$ on n vertices as well as two designated terminals $s, t \in V$, our goal is to compute the shortest path from s to t avoiding any pair of presumably failed edges $f_1, f_2 \in E$, which is a natural generalization of the classical replacement path problem which considers single edge failures only.

This dual failure replacement paths problem was recently studied by Vassilevska Williams, Woldeghebriel and Xu [FOCS 2022] who designed a cubic time algorithm for general weighted digraphs which is conditionally optimal; in the same paper, for unweighted graphs where $\omega \equiv 1$, the authors presented an algebraic algorithm with runtime $\tilde{O}(n^{2.9146})$, as well as a conditional lower bound of $n^{8/3-o(1)}$ against combinatorial algorithms. However, it was unknown in their work whether fast matrix multiplication is necessary for a subcubic runtime in unweighted digraphs.

As our primary result, we present the first truly subcubic combinatorial algorithm for dual failure replacement paths in unweighted digraphs. Our runtime is $\tilde{O}(n^{3-1/18})$. Besides, we also study algebraic algorithms for digraphs with small integer edge weights from $\{-M, -M+1, \cdots, M-1, M\}$. As our secondary result, we obtained a runtime of $\tilde{O}(Mn^{2.8716})$, which is faster than the previous bound of $\tilde{O}(M^{2/3}n^{2.9144} + Mn^{2.8716})$ from [Vassilevska Williams, Woldeghebriela and Xu, 2022].

2012 ACM Subject Classification Theory of computation → Shortest paths

Keywords and phrases graph algorithms, shortest paths, replacement paths

Digital Object Identifier 10.4230/LIPIcs.ICALP.2024.41

Category Track A: Algorithms, Complexity and Games

Related Version *Full Version*: `https://arxiv.org/abs/2404.13907`

Funding This work is part of a project that has received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (grant agreement No 803118 UncertainENV).

1 Introduction

In the replacement path problem, we want to understand shortest paths in a directed graph that avoid presumably failed edges. More specifically, let $G = (V, E, \omega)$ be an edge-weighted simple digraph on n vertices and m edges. Fix a pair of source and terminal vertices $s, t \in V$, we want to compute the shortest path from s to t that avoids any designated set $F \subseteq E$ of failed edges.

The most classical setting is when the number of failures is at most one; namely, we want to compute all the values of $\mathsf{dist}(s, t, G \setminus \{f\})$ when f ranges over all edges on the shortest path from s to t in G. The complexity of single-failure replacement path is now well-understood. On the hardness side, it was proved that computing all single-failure replacement paths in weighted graphs requires at least $n^{3-o(1)}$ time [19] assuming the APSP conjecture. To breach the cubic barrier, we need to assume the input digraph has small integer edge weights, or allow approximation errors in the algorithm output. When the edge

51st International Colloquium on Automata, Languages, and Programming (ICALP 2024).
Editors: Karl Bringmann, Martin Grohe, Gabriele Puppis, and Ola Svensson;
Article No. 41; pp. 41:1–41:20
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

weights are integers in the range $\{-M, -M+1, \cdots, M-1, M\}$, there is an algorithm with runtime $\tilde{O}(Mn^{\omega})$[1] [7, 17]. For the special case when the input digraph is unweighted ($\omega \equiv 1$), there is a combinatorial algorithm (algorithms not using fast matrix multiplication) with runtime $\tilde{O}(mn^{1/2})$ [16], which is optimal under the hardness of combinatorial boolean matrix multiplication [19].

A natural extension is to study replacement paths when there are two edge failures. We are interested in fast algorithms that compute for all pairs of edges $f_1, f_2 \in E$ the value of $\mathsf{dist}(s, t, G \setminus \{f_1, f_2\})$. This problem was first studied in [4] and recently revisited in [20]. For general weighted digraphs, the authors of [20] designed an algorithm with runtime $\tilde{O}(n^3)$, which is the same as the easier single failure replacement paths problem. When the graph has small edge weights from range $\{-M, -M+1, \cdots, M-1, M\}$, in the same paper the authors have shown subcubic runtime upper bound of $\tilde{O}(M^{2/3}n^{2.9144} + Mn^{2.8716})$ using fast matrix multiplication. Finally, as complementary to their algorithms, the authors showed a conditional lower bound of $n^{8/3-o(1)}$ against combinatorial algorithms for unweighted digraphs assuming the hardness of boolean matrix multiplication.

According to the results in [20], there is a gap in their understanding about dual-failure replacement paths in unweighted graphs. On the one hand, their algebraic algorithm computes dual-failure replacement paths in $\tilde{O}(n^{2.9144})$ by setting $M = 1$; on the other hand, their conditional lower bound against combinatorial algorithms is also subcubic. So, it is not clear whether combinatorial algorithm can achieve subcubic runtime as well, or the conditional lower bound can be improved to cubic.

1.1 Our results

In this paper, we first study fast combinatorial algorithms for unweighted digraphs and show that subcubic runtime can indeed be achieved without using fast matrix multiplication.

► **Theorem 1.** *Given a simple unweighted directed graph $G = (V, E)$ on n vertices, and fix any pair of vertices $s, t \in V$, the values of all dual-failure replacement path distances $\mathsf{dist}(s, t, G \setminus \{f_1, f_2\}), \forall f_1, f_2 \in E$ can be computed in $\tilde{O}(n^{3-1/18})$ time with high probability; most importantly, the algorithm does not use fast matrix multiplication.*

Here, a digraph is simple if it does not contain two edges between the same pair of vertices with the same direction. Secondly, we also study fast algebraic algorithms for dual-failure replacement paths when the edge weights are from the set $\{-M, -M+1, \cdots, M-1, M\}$. The proof of the following statement is presented in the full version.

► **Theorem 2.** *Given a simple weighted directed graph $G = (V, E, \omega)$ on n vertices along with integer edge weights $\omega : E \to \{-M, -M+1, \cdots, M-1, M\}$ without negative cycles, and fix any pair of vertices $s, t \in V$, the values of all dual-failure replacement path distances $\mathsf{dist}(s, t, G \setminus \{f_1, f_2\}), \forall f_1, f_2 \in E$ can be computed in time $\tilde{O}(Mn^{2.8716})$.*

1.2 Other related works

The replacement paths problem has also been studied in other settings, including the single-source setting [13, 6, 12, 11] and the approximation setting [8, 2, 15].

[1] $\omega \in [2, 2.371552]$ is the fast matrix multiplication exponent [21, 10, 1, 14, 18].

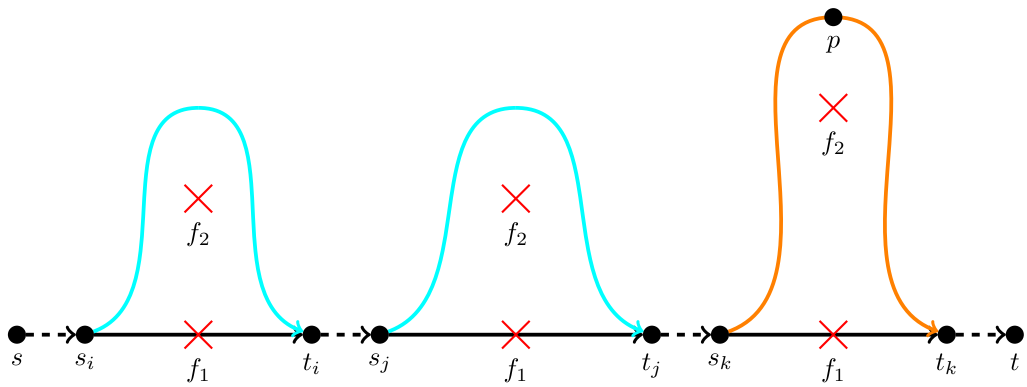


Figure 1 For simplicity, let us assume that when f_1 falls on the sub-paths $\pi[s_i, t_i]$, the dual-failure replacement path also passes through s_i, t_i. In this picture, the two cyan dual-failure replacement paths have length less than L, and we will show that they are vertex-disjoint. The orange dual-failure replacement path has length at least L, and so it hits a vertex $p \in U$ with high probability; in this case, we will compute single-source single-failure replacement paths to and from p in graph $G \setminus E(\pi)$ to help us compute dual-failure replacement paths.

1.3 Subcubic combinatorial algorithm for unweighted digraphs

1.3.1 One failure on a long st-path

Let us first consider the case where one edge failure f_1 lies on the shortest st-path π, while the other one f_2 does not. This case would be easy when we use fast matrix multiplication as did by [20], but it becomes complicated when we are restricted to purely combinatorial algorithms.

As a preliminary step, we first show how to deal with the case where $|\pi| > L$ for some parameter L slightly larger than $n^{0.5}$, say $L = n^{0.55}$. Partition the st-path π into sub-paths of length exactly $5L$ as $\pi = \gamma_1 \circ \gamma_2 \circ \cdots \circ \gamma_h, h \leq \lceil n/5L \rceil$, and let s_i, t_i be the endpoints of sub-path γ_i. Assume only one edge failure f_1 falls on sub-path γ_i, and let ρ be the optimal replacement path from s to t avoiding $\{f_1, f_2\}$.

Take a random set of vertices U of size $O(n \log n/L)$. If $|\rho \setminus \pi| > L$, then with high probability $\rho \setminus \pi$ would contain a vertex p from U. Then, we can compute single-source single-failure replacement paths to and from p in graph $G \setminus E(\pi)$ that takes runtime $\tilde{O}(n^{3.5}/L)$ [6], so that for each vertex $z \in V$, we know the shortest path between z, p in graph $G \setminus (E(\pi) \cup \{f_2\})$. Using this information, we will be able to recover ρ.

Now suppose $|\rho \setminus \pi| < L$. Then in this case, we will show that the detour parts of different dual-failure replacement paths ρ are vertex-disjoint from each other when the first edge failure f_1 comes from different choice of sub-paths γ_i. Then, we can partition G into vertex-disjoint subgraphs $G_1, G_2, \cdots, G_h$, such that the dual-failure replacement paths for failures on γ_i belong to subgragh G_i, and solve the dual-failure replacement paths problem for source-terminal pair (s_i, t_i) in graph G_i. See Figure 1 for an illustration.

1.3.2 One failure on a short st-path

By the previous subsection, we have reduced to the case that the st-path has length at most L. So, for the rest, let us rename the problem instance and assume that $|\pi| \leq L$. For the i-th edge e_i on π $(0 \leq i < L)$, we can compute the optimal replacement path from s to t avoiding e_i and let α_i be the corresponding detour whose endpoints are a_i and b_i. Again, through

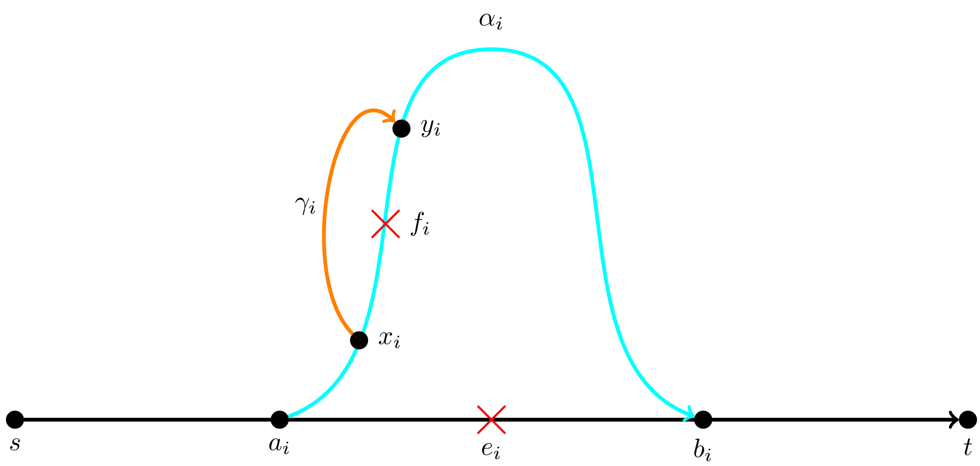


Figure 2 The cyan path is the detour α_i that avoids e_i, and the orange path is the detour γ_i that avoids f_i which lies on α_i. Via some case analysis, we will show the difficult case is that $|\alpha_i| < L$.

some case analysis, we can assume that the span of each detour α_i which is $|\pi[a_i, b_i]|$ is at most g for some parameter g slightly larger than $n^{1/3}$ (say $g = n^{0.35}$). Consider any edge failure $f_i \in E(\alpha_i)$, as a simplification let us assume that the optimal replacement path from s to t avoiding $\{e_i, f_i\}$ is a concatenation:

$$\rho = \pi[s, a_i] \circ \alpha_i[a_i, x_i] \circ \gamma_i \circ \alpha_i[y_i, b_i] \circ \pi[b_i, t]$$

where γ_i is a detour with respect to α_i in $G \setminus E(\pi)$. Via some case analysis, we can assume that $|\alpha_i| < L$. See Figure 2 for an illustration.

Let us first consider the case when $|\gamma_i| < g$. A wishful thought is that for two different choice of dual failures $\{e_i, f_i\}$ and $\{e_j, f_j\}$ where e_i and e_j are well-separated on the shortest path π ($|\pi(e_i, e_j)| \geq 10g$), we are guaranteed that the two dual-failure detours γ_i and γ_j are vertex-disjoint; if this is the case, then we can partition the graph G into $O(L/g)$ vertex-disjoint subgraphs and compute dual-failure replacement paths separately. However, this is generally not true. The key observation is that when f_i and f_j are roughly at the same height on the detour, namely $|\alpha_i[a_i, f_i)| \approx |\alpha_j[a_j, f_j)|$ up to an additive error of at most g, such a disjointness condition indeed holds. Therefore, our algorithm will further partition each detour α_i into sub-paths of length g as $\alpha_i = \beta_1^i \circ \beta_2^i \circ \cdots \circ \beta_l^i$. Then, fix any height index $1 \leq h \leq l$, we will deal with all the dual failures $\{e_i, f_i\}, \forall 1 \leq i \leq L/g, \forall f_i \in E(\beta_h^i)$ at the same time. See Figure 3 for an illustration.

Now, what happens if $|\gamma_i| \geq g$? We can take a random sample U of pivot vertices of size $O(n \log n/g)$. Then, with high probability, γ_i contains a vertex in U. For simplicity, assume both endpoints of the detour γ_i are lying within the sub-path β_h^i which contains the second edge failure f_i, then we could compute single-source shortest paths to and from each vertex $p \in U$ in the subgraph $G \setminus \left(E(\pi) \cup E(\beta_h^i)\right)$ which takes time $\tilde{O}(n^3/g)$, then we can compute each detour γ_i for each choice of f_i on β_h^i in time $\tilde{O}(g^2)$ by guessing the positions of x_i, y_i.

Unfortunately, there are L^2/g different choices for the subgraph $G \setminus \left(E(\pi) \cup E(\beta_h^i)\right)$, and thus we do not have enough time to compute single-source shortest paths for each vertex in U in all these subgraphs. In fact, we can only afford to compute single-source shortest paths for vertices in U in the subgraph $G_h = G \setminus \left(E(\pi) \cup \bigcup_{i=0}^{L-1} E(\beta_h^i)\right)$; that is, for each index h, remove all sub-paths $\beta_h^i, \forall 0 \leq i < L$ from $G \setminus E(\pi)$ simultaneously (which becomes G_h), and compute multi-source shortest paths to and from U in G_h. The key observation is that

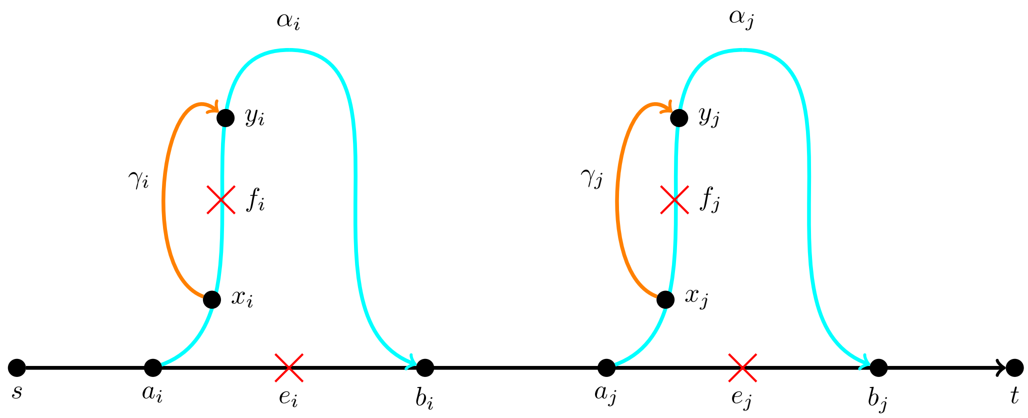


Figure 3 For two well-separated edges e_i, e_j such that $|\pi(e_i, e_j)| \geq 10g$, if both $|\gamma_i|, |\gamma_j|$ are less than g, and f_i, f_j are roughly at the same height (i.e., $|\alpha_i[a_i, f_i)| \approx |\alpha_j[a_j, f_j)|$), then we can show that the two dual-failure detours γ_i, γ_j are vertex-disjoint.

the detour γ_i cannot touch vertices on β_h^j if $|i - j| > 10g$. Therefore, to compute γ_i, we can build a small shortcut graph consisting of all vertices in $U \cup \bigcup_{j=i-10g}^{i+10g} V(\beta_h^j)$ which contains all edges in $\bigcup_{j=i-10g}^{i+10g} E(\beta_h^j) \setminus E(\beta_h^i)$ and all shortcut edges to and from $p \in U$ weighted by the single-source distances we have computed in G_h. See Figure 4 for an illustration.

1.3.3 Both failures on the st-path

Now let us assume both edge failures lie on π. As the same in [20], the main difficulty of dual-failure replacement path for this case comes from the backward paths. More specifically, given two edge failures $\{f_1, f_2\}$, in general the optimal dual replacement path ρ has three parts.

- A prefix of ρ that diverges from π before the first edge failure f_1 on $\pi[s, f_1)$, and then meets somewhere in the middle $y \in V(\pi(f_1, f_2))$ using edges in $G \setminus E(\pi)$.
- A sub-path of ρ that travels from y to another vertex $x \in V(\pi(f_1, y])$ using edges in $(G \setminus E(\pi)) \cup E(\pi(f_1, f_2))$; this sub-path is the so-called *backward* path of ρ which may converge and diverge multiple times with $\pi(f_1, f_2)$.
- A suffix of ρ that converges with π after the second edge failure f_2 on $\pi(f_2, t]$ starting from x using edges in $G \setminus E(\pi)$, and then reach t using the rest of π.

To compute the backward path, let us divide π into sub-paths of length L for some parameter L as $\pi = \gamma_1 \circ \gamma_2 \circ \cdots \circ \gamma_{n/L}$. Take a random pivot vertex set U of size $O(\frac{n \log n}{L})$. The main observation is that if f_1 and f_2 are in different sub-paths $\gamma_i, \gamma_j, j - i > 1$, plus that $x \in V(\gamma_i), y \in V(\gamma_j)$, then the prefix $\rho[s, y]$ must have length at least L and thus contain a pivot $p \in U$ with high probability. Therefore, if we compute single-source replacement paths [6] from p in graph $G \setminus E(\gamma_i)$, then it would provide useful information about the backward path of ρ from y to x. See Figure 5 for an illustration.

1.4 Faster algebraic algorithm for weighted digraphs

Let us divide the shortest path π from s to t into sub-paths of hops at most L; that is, $\pi = \gamma_1 \circ \gamma_2 \circ \cdots \circ \gamma_{\lceil n/L \rceil}$. Given a pair of edge failures $\{f_1, f_2\}$, suppose f_1 and f_2 belong to γ_{l_1} and γ_{l_2} respectively. In the previous paper [20], the difficult case is when f_1, f_2 come from different sub-paths, say $l_1 < l_2$. To compute $\mathsf{dist}(s, t, G \setminus \{f_1, f_2\})$ efficiently, their approach

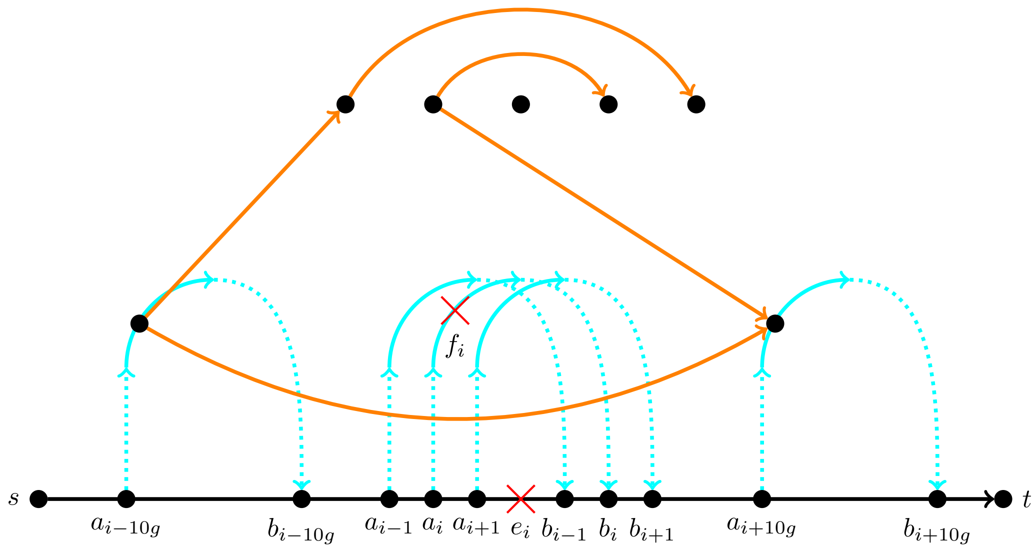


Figure 4 A shortcut graph that helps computing the dual-failure detour γ_i avoiding $\{f_i, g_i\}$, which consists of vertices in $U \cup \bigcup_{j=i-10g}^{i+10g} V(\beta_h^j)$. This shortcut graph includes all edges in $\bigcup_{j=i-10g}^{i+10g} E(\beta_h^j) \setminus E(\beta_h^i)$, and some shortcut edges representing distances to and from U in G_h; actually, it will also contain some shortcut edges between vertices in $\bigcup_{j=i-10g}^{i+10g} V(\beta_h^j)$ which we have not discussed in the overview.

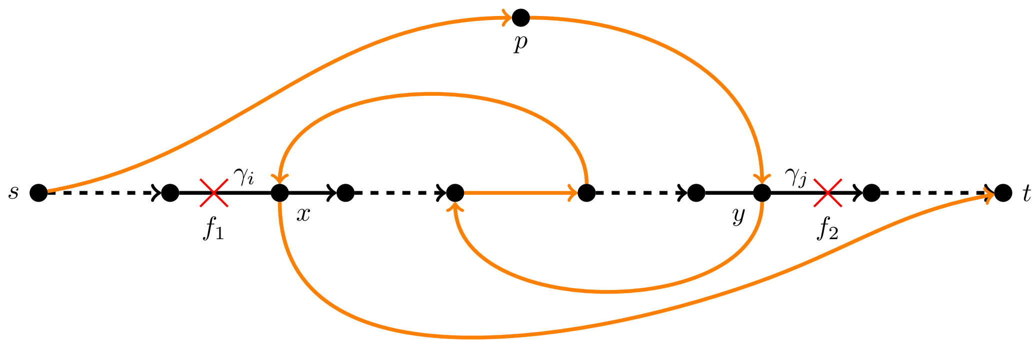


Figure 5 When f_1, f_2 lie in different sub-paths γ_i, γ_j such that $j - i > 1$, we can show that ρ must contain a vertex in U with high probability. Then we can apply the algorithm from [6] to compute single-source replacement path from p in graph $G \setminus E(\gamma_i)$ to learn about the sub-path $\rho[p, x]$ which contains the backward path in the middle.

was to build a sketch graph H_{f_1,f_2} on vertices $\{s,t\} \cup V(\gamma_{l_1}) \cup V(\gamma_{l_2})$ which encodes an optimal replacement path, and then run s-t shortest path in H_{f_1,f_2} which takes $\tilde{O}(L^2)$ time for each $\{f_1, f_2\}$, leading to a total runtime overhead of $\tilde{O}(n^2L^2)$.

Divide each sub-path γ_l into segments $\gamma_l = \alpha_1^l \circ \alpha_2^l \circ \cdots \circ \alpha_{\lceil L/g \rceil}^l$ of hops at most g (for some parameter $g < L$). Assume f_1 lies on $\alpha_{h_1}^{l_1}$ and f_2 lies on $\alpha_{h_2}^{l_2}$, and let ρ be the optimal replacement path from s to t avoiding $\{f_1, f_2\}$. Our main observation is that, if ρ intersects both $\alpha_{h_1}^{l_1}, \alpha_{h_2}^{l_2}$, then we can build a smaller sketch graph H_{f_1,f_2} on $\{s,t\} \cup V\left(\alpha_{h_1}^{l_1}\right) \cup V\left(\alpha_{h_2}^{l_2}\right)$ which still encodes ρ, and so the runtime would be reduced to $\tilde{O}(g^2)$. Otherwise, if ρ skips over $\alpha_{h_2}^{l_2}$ entirely, we will build a sketch graph $H_{f_1,(l_2,h_2)}$ on the vertex set $\{s,t\} \cup V(\gamma_{l_1}) \cup V(\gamma_{l_2})$ which only depends on f_1 and $\alpha_{h_2}^{l_2}$ and not on f_2. As the number of such graphs $H_{f_1,(l_2,h_2)}$ is at most $O(n^2/g)$, the runtime can be bounded by $\tilde{O}(n^2L^2/g)$. Overall, the runtime would be $\tilde{O}(n^2g^2 + n^2L^2/g)$ which is always better than the previous bound of $\tilde{O}(n^2L^2)$.

2 Preliminaries

Throughout the paper, logarithm $\log(*)$ will have base 2, and we assume the number of vertices n in the input graph is an integral power of 2. In any weighted digraph $G = (V, E, \omega)$, for any vertex $u \in V$, let $\deg(u, G)$ be its vertex degree (counting both in-edges and out-edges); for any pair of vertices $u, v \in V$, let $\mathsf{dist}(u, v, G)$ be the weighted length of the shortest path from u to v in G. Throughout the algorithm, we will maintain a distance estimation function $\mathsf{est}(*, *, *)$, such that the value $\mathsf{est}(u, v, G) \geq \mathsf{dist}(u, v, G)$ is always an upper bound. All values of $\mathsf{est}(*, *, *)$ are initially infinity and are non-increasing throughout the algorithms. When we update the value of an estimation $\mathsf{est}(u, v, G)$ with a distance value D, we mean $\mathsf{est}(u, v, G) \leftarrow \{D, \mathsf{est}(u, v, G)\}$.

Given any directed path or walk ρ in G, let $|\rho|$ be the number of edges on ρ, and let $\omega(\rho)$ be its total edge weight. For any two vertices $u, v \in V(\rho)$ where u comes before v on ρ, let $\rho[u, v] \subseteq \rho$ be the sub-path of ρ from u to v. In addition, let $\rho[*, v]$ and $\rho[u, *]$ be the prefix and suffix sub-path of ρ; this notation will be useful when we don't have variable names for the endpoints of path ρ.

For any edge $f \in E(\rho)$ and vertex $u \in V(\rho)$ which comes before edge f, let $\rho[u, f)$ be the sub-path from u to f (excluding edge f); similarly we can define notations $\rho(f, v]$ and $\rho(f_1, f_2)$ in the natural way.

Borrowing a terminology from [20], let us define the notion of canonical paths.

▶ **Definition 3** ([20]). *Let $s, t \in V$ be two vertices, and let π be a shortest path from s to t. For any edge set $F \subseteq E$, a path ρ from s to t in $G \setminus F$ is called* canonical *with respect to π and F, if for any $u, v \in V(\pi) \cap V(\rho)$ such that u appears before v in both π, ρ and $E(\pi[u, v]) \cap F = \emptyset$, then $\rho[u, v]$ is the same as $\pi[u, v]$.*

2.1 Unweighted digraphs

For tie-breaking among shortest paths, we can randomly perturb all unit edge weights slightly so that all replacement shortest paths for at most two edge failures are unique under the perturbed weights. We can show that, under the weight perturbation, all replacement shortest paths are canonical. Next, let us state a basic property regarding shortest replacement paths for one edge failure.

▶ **Lemma 4.** *Consider any edge $f \in E(\pi)$, any canonical shortest path ρ from s to t avoiding f can be decomposed as $\rho = \pi[s, a] \circ \alpha \circ \pi[b, t]$, where $a, b \in V(\pi)$ and α is a shortest path from a to b in $G \setminus E(\pi)$. For convenience, α will be called the* detour *of the replacement path, and a, b are called the* divergence *and* convergence *vertex, respectively.*

It is known that shortest replacement paths for one failure can be computed efficiently.

▶ **Lemma 5** ([16]). *Given an unweighted digraph $G = (V, E)$ on n vertices and m edges. Fixing any source-terminal pair $s, t \in V$, we can compute all canonical shortest replacement paths from s to t avoiding f with high probability in time $\tilde{O}(mn^{1/2} + n^2)$, where f ranges over all edges in E.*

We also need a basic property about replacement paths for dual edge failures where only one edge falls on the st-path π.

▶ **Lemma 6.** *Consider any edge $f_1 \in E(\pi)$ and let α be the detour of the shortest replacement path that avoids f_1. Consider any edge $f_2 \in E(\alpha)$. Then, there is a shortest replacement path ρ avoiding $\{f_1, f_2\}$ such that:*

- *ρ is a canonical shortest path avoiding f_1 in graph $G \setminus \{f_1\}$.*
- *ρ diverges and converges with π for once.*

We will be applying the following algorithm for single-source replacement paths algorithm from [6] as a black-box.

▶ **Lemma 7** ([6]). *Given an unweighted digraph $G = (V, E)$ on n vertices and m edges. Fixing any vertex $s \in V$, we can compute all shortest replacement paths from s to t avoiding f with high probability in time $\tilde{O}(mn^{1/2} + n^2)$, where t ranges over all vertices in $V \setminus \{s\}$ and f ranges over all edges in E.*

To be consistent with our perturbation-based tie-braking, we should also impose the same edge weight perturbation on the graph on which Lemma 7 is applied; although Lemma 7 is only stated for unweighted digraphs, it also works with edge weight perturbation (for example, all hitting set arguments still work, since perturbation only changes path lengths negligibly).

In the original paper [6], they only claim to compute all the length of shortest replacement paths, but here we need the actual shortest paths tree in each graph $G \setminus \{f\}$. To achieve such an augmentation, during the execution of the algorithm in [6], we can keep track of the last edge of each replacement path, and by uniqueness of shortest paths under weight perturbation, the set of these last edges form the shortest paths tree.

Finally, one of our basic tools is a *truncated* version of Dijkstra's algorithm, which is stated below.

▶ **Lemma 8** ([9]). *Given an unweighted digraph $G = (V, E)$ on n vertices. For any vertex $s \in V$ and any threshold L, let $U = \{u \mid \mathsf{dist}(s, u, G) \leq L\}$. Then we can compute shortest paths from s to all vertices in U in time $O(\sum_{u \in U} \deg(u, G) + n \log n)$.*

2.2 Weighted digraphs

When the input graph contains negative edge weights, we assume it does not contain any negative cycles. For weighted graphs, since fast matrix multiplication algorithms only work with small integer edge weights, we will not assume unique shortest paths by perturbing the edge weights.

Our algorithm relies on fast algorithms which computes shortest paths in digraphs with negative edge weights.

▶ **Lemma 9** ([3]). *Given a weighted digraph $G = (V, E, \omega)$ with edge weights $\omega : E \to \{-M, -M+1, \cdots, M-1, M\}$ without negative cycles, and fix any source vertex $s \in V$. Then, single-source shortest path from s can be computed in time $\tilde{O}(m \log M)$ with high probability.*

▶ **Lemma 10** ([22]). *Given a weighted digraph $G = (V, E, \omega)$ with edge weights $\omega : E \to \{-M, -M+1, \cdots, M-1, M\}$ without negative cycles, all-pairs shortest paths in G can be computed in time $\tilde{O}(M^{\frac{1}{4-\omega}} n^{2+\frac{1}{4-\omega}})$ with high probability.*

We will apply single-source multi-terminal replacement paths algorithms as a black-box; the currently best known upper bound is stated below.

▶ **Lemma 11** ([13]). *Given a weighted digraph $G = (V, E, \omega)$ with edge weights $\omega : E \to \{-M, -M+1, \cdots, M-1, M\}$ without negative cycles, and fix any source vertex $s \in V$ and a terminal set $T \subseteq V$. Then, the value of all $\mathsf{dist}(s, t, G \setminus \{f\}), \forall t \in T, f \in E$ can be computed in time $\tilde{O}(Mn^{\omega} + M^{\frac{1}{4-\omega}} n^{1+\frac{1}{4-\omega}} \cdot |T|)$ with high probability.*

We will use the following fast algorithm for distance sensitivity oracles in weighted digraphs. In a weighted digraph $G = (V, E, \omega)$ with edge weights $\omega : E \to \{-M, -M+1, \cdots, M-1, M\}$, a distance sensitivity oracle is an efficient data structure that answers $\mathsf{dist}(u, v, G \setminus \{f\})$ for any $u, v \in V, f \in E$.

▶ **Lemma 12** ([5]). *Given a weighted digraph $G = (V, E, \omega)$ with edge weights $\omega : E \to \{-M, -M+1, \cdots, M-1, M\}$ without negative cycles, a distance sensitivity oracle with $\tilde{O}(1)$ query time can be constructed in time $\tilde{O}(Mn^{2.8719})$.*

3 One failure on a short st-path

We use two parameters g and L which will be determined in the end such that $g < n^{1/2} < L$. In this section, we study the case where only one edge failure lies on a short st-path, plus that the input graph is sparse and $\mathsf{dist}(s, t, G) \le L$. More specifically, let $G = (V, E)$ be an unweighted digraph with n vertices and m edges, and consider a pair of vertices s, t with a shortest st-path $E(\pi)$ of length at most L. The task is to compute for any pairs of edges f_1, f_2 the value of $\mathsf{dist}(s, t, G \setminus \{f_1, f_2\})$, where $f_1 \in E(\pi), f_2 \notin E(\pi)$. The purpose of this section is to prove the following lemma.

▶ **Lemma 13.** *Let $G = (V, E)$ be an unweighted digraph with n vertices and m edges. Fix a pair of source and terminal $s, t \in V$ such that $\mathsf{dist}(s, t, G) \le L$. Then, all values of $\mathsf{dist}(s, t, G \setminus \{f_1, f_2\})$ can be computed with high probability in time:*

$$\tilde{O}\left(\frac{mn^{1.5} + n^3}{L} + mn^{1/2}L/g + n^2L/g + mL^2/g + mnL^2/g^3 + mLg + L^2g^4 + n^2L^2/g^2\right)$$

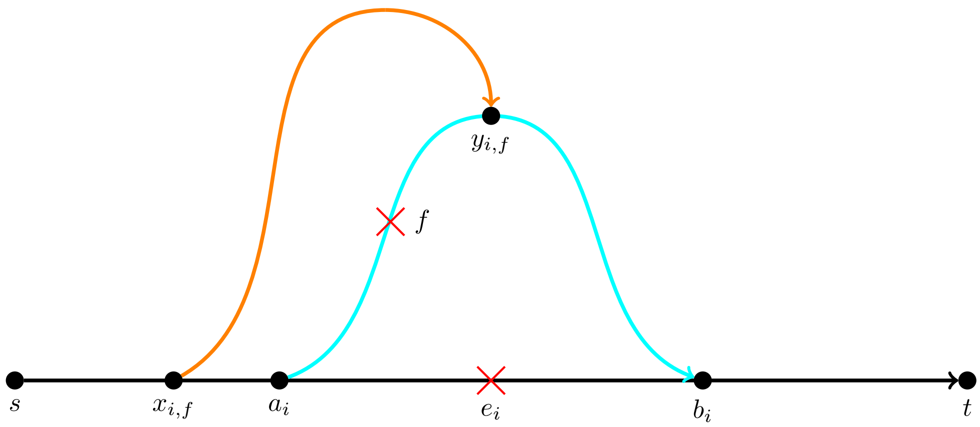
when $f_1 \in E(\pi)$ while $f_2 \notin E(\pi)$; here g is an arbitrary parameter to be determined later.

Let $\pi = \langle s = u_0, u_1, \cdots, u_{|\pi|} = t\rangle$. First, we use Lemma 5 to compute for each $(u_i, u_{i+1}) \in E(\pi)$ the replacement path from s to t that avoids $e_i = (u_i, u_{i+1})$, and let α_i be the corresponding detour avoiding e_i which starts at a_i and ends at b_i.

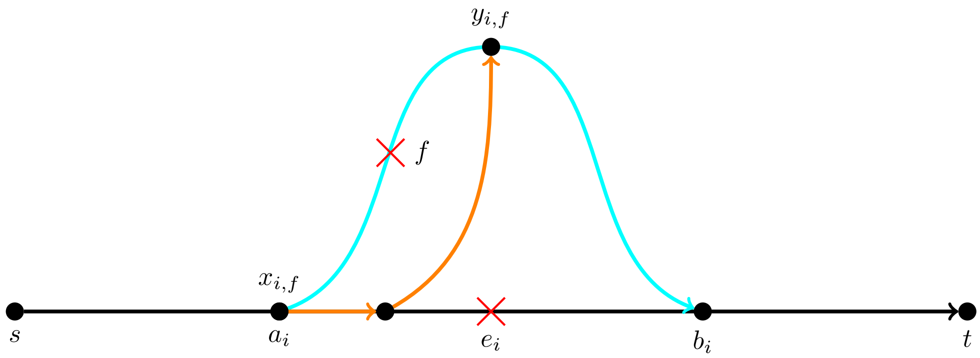
For each detour α_i, divide α_i into sub-paths of length g (except for the last sub-path) and list them as $\beta^i_1, \cdots, \beta^i_{l_i}$, and assume $f \in E(\beta^i_l)$; later on, when we refer to β^i_l, if $l \le 0$ or $l > l_i$, then β^i_l would simply be an empty path.

Throughout the algorithm, for each $0 \le i < |\pi|$ and every edge $f \in E(\alpha_i)$, we will maintain a distance estimation $\mathsf{est}(s, t, G \setminus \{e_i, f\}) \ge \mathsf{dist}(s, t, G \setminus \{e_i, f\})$, and in the end it will be guaranteed that $\mathsf{est}(s, t, G \setminus \{e_i, f\}) = \mathsf{dist}(s, t, G \setminus \{e_i, f\})$.

Let $\rho_{i,f}$ be a canonical shortest replacement path for $\{e_i, f\}$. Suppose $\rho_{i,f}$ departs from $\pi[s, a_i] \circ \alpha_i \circ \pi[b_i, t]$ at vertex $x_{i,f}$ and converges with $\pi[s, a_i] \circ \alpha_i \circ \pi[b_i, t]$ at vertex $y_{i,f}$. For the rest, we address different cases depending on the properties regarding $\alpha_i, \rho_{i,f}$; note that our algorithm does not need to know which case it is in advance.



(a) In this case, $x_{i,f} \neq a_i$ and $\rho_{i,f}[x_{i,f}, *]$ diverges from π immediately.



(b) In this case, $x_{i,f} = a_i$ and $\rho_{i,f}[x_{i,f}, *]$ uses some edges on $\pi[a_i, b_i]$ before it diverges.

Figure 6 In this picture, the cyan path is detour α_i, and the orange path is $\rho_{i,f}[x_{i,f}, y_{i,f}]$.

Case 1: $x_{i,f} \in V(\pi[s, a_i])$ or $y_{i,f} \in V(\pi[b_i, t])$

This is an easy case of our algorithm, and the runtime of this part can be bounded by $O(mL)$. See Figure 6 for an illustration. Check the full version for more details.

Case 2: $x_{i,f}, y_{i,f} \in V(\alpha_i)$ and $|\alpha_i| \geq L$

This is an easy case of our algorithm, and the runtime of this part can be bounded by $\tilde{O}(\frac{mn^{1.5}+n^3}{L})$. See Figure 7 for an illustration. Check the full version for more details.

Case 3: $x_{i,f}, y_{i,f} \in V(\alpha_i)$, plus that $|\pi[a_i, u_i]| > g$ or $|\pi[u_{i+1}, b_i]| > g$

This is an easy case of our algorithm, and the runtime of this part can be bounded by

$$\tilde{O}\left(m\sqrt{n}L/g + n^2L/g + L^2\right)$$

See Figure 8 for an illustration. Check the full version for more details.

Case 4: $x_{i,f}, y_{i,f} \in V(\alpha_i)$, plus that $x_{i,f} \notin V(\beta_l^i)$ or $y_{i,f} \notin V(\beta_l^i)$, and $|\alpha_i| < L$

This is an easy case of our algorithm, and the runtime of this part can be bounded by $\tilde{O}(mL^2/g)$. See Figure 9 for an illustration. Check the full version for more details.

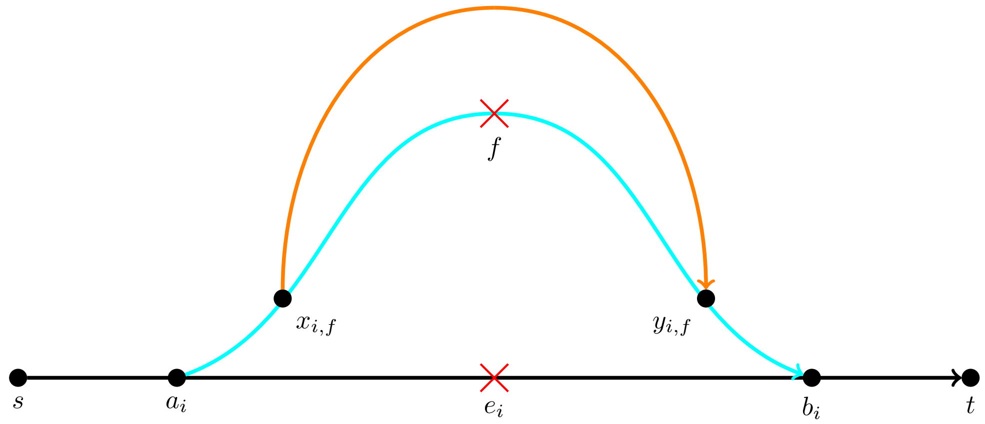


Figure 7 The sub-path $\rho_{i,f}[a_i, b_i]$ has length greater than L.

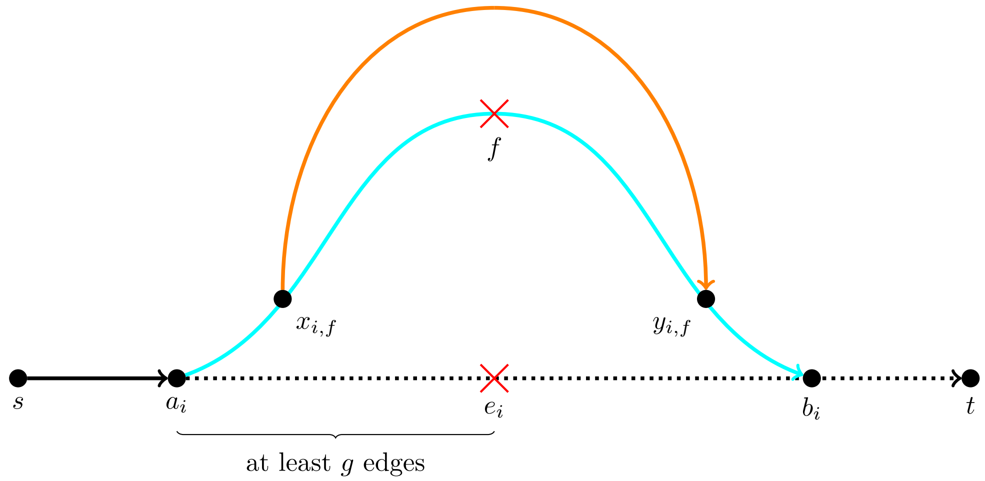


Figure 8 In this picture, $x_{i,f}, y_{i,f} \in V(\alpha_i)$ plus that $|\pi[a_i, u_i]| > g$.

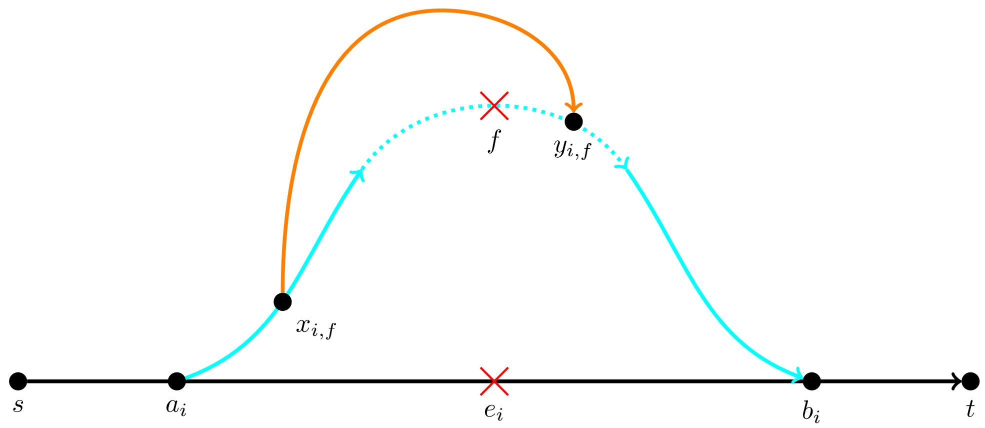


Figure 9 The dotted cyan path is β_l^i.

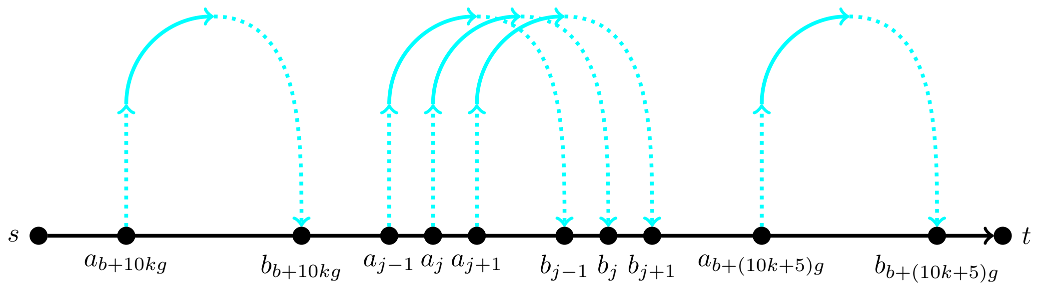


Figure 10 Graph X_l^{b+10kg} excludes all edges in $E(\pi) \cup \bigcup_{j=b+10kg, j\in I}^{b+(10k+5)g} E(\beta_l^j)$ which are drawn and black and cyan solid paths.

Main case: $|\pi[a_i, u_i]|, |\pi[u_{i+1}, b_i]| \leq g$, and $x_{i,f}, y_{i,f} \in V(\beta_l^i)$, and $|\alpha_i| < L$

Algorithm Main case.

Let $I \subseteq [L]$ be the set of all indices such that $|\pi[a_i, u_i]|, |\pi[u_{i+1}, b_i]| \leq g$ and $|\alpha_i| < L$. For each pair of indices $(p, q) \in [L/g] \times [L/g]$, define the set of sub-paths:

$$\mathcal{P}_{p,q} = \{\beta_l^i \mid (l, l_i - l + 1) = (p, q), i \in I\}$$

(1) Let $U \subseteq V$ be the uniformly random subset of size $\frac{10n \log n}{g}$. Then, for each pair of indices $p, q \in [L/g]$, define the graph:

$$G_{p,q} = G \setminus \left(E(\pi) \cup \bigcup_{\beta \in \mathcal{P}_{p,q}} E(\beta)\right)$$

Then, for each vertex $u \in U$, compute single-source shortest paths to and from u in $G_{p,q}$.

(2) For any index $1 \leq l \leq \lceil L/g \rceil$, and for any offset $1 \leq b \leq 10g$, initialize two sets of vertices $A_{b,l}, B_{b,l} \leftarrow V$. Then, go over the sequence of all sub-paths $\beta_l^b, \beta_l^{b+10g}, \cdots, \beta_l^{b+10hg}$, where $h \leq \lceil \frac{L}{10g} \rceil$.

Next, for each sub-path β_l^{b+10kg}, define the following two graphs

$$X_l^{b+10kg} = G \setminus \left(E(\pi) \cup \bigcup_{j \in [b+10kg, b+(10k+5)g] \cap I} E(\beta_l^j)\right)$$

$$Y_l^{b+10kg} = G \setminus \left(E(\pi) \cup \bigcup_{j \in [b+(10k-5)g, b+10kg] \cap I} E(\beta_l^j)\right)$$

See Figure 10 for an illustration.

(a) Then, for each vertex $v \in V\left(\beta_l^{b+10kg}\right)$, perform a truncated Dijkstra at v in the induced subgraph $X_l^{b+10kg}[A_{b,l}]$ up to depth g. For each $z \in A_{b,l}$, record the distance value

$$\mu_X(v, z) \leftarrow \mathsf{dist}\left(v, z, X_l^{b+10kg}[A_{b,l}]\right)$$

if $\mathsf{dist}\left(v, z, X_l^{b+10kg}[A_{b,l}]\right) \leq g$.

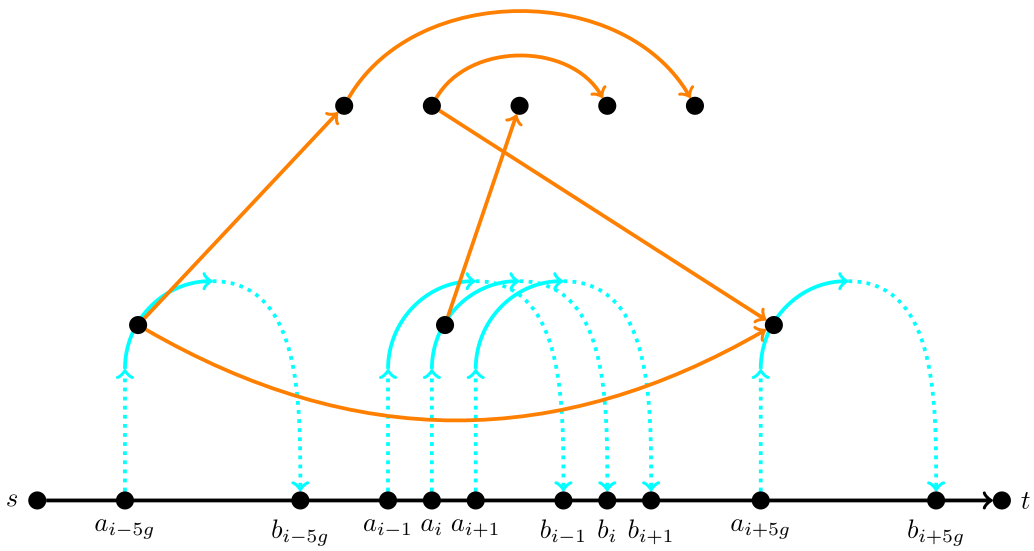


Figure 11 Vertices on the topmost are in U, and orange edges represent shortcut edges in H_l^i.

After we have visited all vertices $v \in V\left(\beta_l^{b+10kg}\right)$, let $P_{b,l}^k \subseteq A_{b,l}$ be the set of all vertices searched by truncated Dijkstra of any $v \in V\left(\beta_l^{b+10kg}\right)$. Then, prune the vertex set

$$A_{b,l} \leftarrow A_{b,l} \setminus P_{b,l}^k$$

(b) Symmetrically, for each vertex $v \in V\left(\beta_l^{b+10kg}\right)$, perform a truncated Dijkstra at v in the induced subgraph $Y_l^{b+10kg}[B_{b,l}]$ up to depth g. For each $z \in B_{b,l}$, record the distance value

$$\mu_Y(v, z) \leftarrow \mathsf{dist}\left(v, z, Y_l^{b+10kg}[B_{b,l}]\right)$$

if $\mathsf{dist}\left(v, z, Y_l^{b+10kg}[B_{b,l}]\right) \leq g$.

After we have visited all vertices $v \in V\left(\beta_l^{b+10kg}\right)$, let $Q_{b,l}^k \subseteq B_{b,l}$ be the set of all vertices searched by truncated Dijkstra of any $v \in V\left(\beta_l^{b+10kg}\right)$. Then, prune the vertex set

$$B_{b,l} \leftarrow B_{b,l} \setminus Q_{b,l}^k$$

(3) For any index $i \in I$ and index $1 \leq l \leq l_i/g$, let us build a shortcut digraph H_l^i with edge weight function ω in the following manner. See Figure 11 for an illustration.

- **Vertices.** Add all vertices in sub-paths $V(\beta_l^j), \forall j \in [i-5g, i+5g] \cap I$, as well as all pivot vertices in U to H_l^i.
- **Edges.** First, add to $E(H_l^i)$ all the sub-paths:

$$\left(\bigcup_{j\in[i-5g,i+5g]\cap I} E\left(\beta_l^j\right)\right) \setminus E\left(\beta_l^i\right)$$

Then, add the following three types of weighted edges.

(a) For any $u \in U$ and any vertex $v \in V(H_l^i)$, add an edge (u, v) with edge weight:

$$\omega(u, v) = \mathsf{dist}(u, v, G_{l,l_i-l+1})$$

and edge (v, u) with edge weight:

$$\omega(v, u) = \mathsf{dist}(v, u, G_{l,l_i-l+1})$$

(b) For any pair of vertices $u, v \in V(H_l^i)$ where $u \in V(\beta_l^j), j \in [i - 5g, i]$, add an edge (u, v), and assign a weight:

$$\omega(u, v) = \mu_X(u, v)$$

If $\mu_X(u, v)$ was not assigned in Step (2), then it is infinity by default.

(c) For any pair of vertices $u, v \in V(H_l^i)$ where $u \in V(\beta_l^j), j \in [i, i + 5g]$, add an edge (u, v), and assign a weight:

$$\omega(u, v) = \mu_Y(u, v)$$

If $\mu_Y(u, v)$ was not assigned in Step (2), then it is infinity by default.

After that, for each vertex $z \in V\left(\beta_l^i\right)$, apply single-source shortest path on z in H_l^i. In this way, for every pair of vertices $x, y \in V\left(\beta_l^i\right)$, we have computed a distance $\mathsf{dist}\left(x, y, H_l^i\right)$.
Finally, for every edge $f \in E\left(\beta_l^i\right)$, update the estimation $\mathsf{est}(s, t, G \setminus \{e_i, f\})$ with:

$$\min_{x \in V(\beta_l^i[*,f)), y \in V(\beta_l^i(f,*])} \left\{\mathsf{dist}(s, x, G \setminus \{e_i\}) + \mathsf{dist}\left(x, y, H_l^i\right) + \mathsf{dist}(y, t, G \setminus \{e_i\})\right\}$$

Runtime. The runtime of Step (1) is bounded by $\tilde{O}(|U| \cdot mL^2/g^2) = \tilde{O}(\frac{mnL^2}{g^3})$. As for the runtime of Step (2), consider any offset $1 \le b \le 10g$. We argue that the Dijkstra procedure for all vertices on sub-paths $\beta_l^b, \beta_l^{b+10g}, \cdots, \beta_l^{b+10hg}$ has runtime at most $O(mg)$; this is because every vertex in V is visited by at most $O(g)$ instances of Dijkstra rooted at vertices from some sub-paths β_l^{b+10kg}. Therefore, the overall runtime of Step (2) is bounded by $O(mLg)$ summing over all $1 \le b \le 10g$ and $1 \le l \le L/g$.

Finally, let us estimate the runtime of Step (3). For each index $i \in I$, there are at most L/g different sub-paths β_l^i as $|\alpha_i| \le L$. By definition, the shortcut digraph H_i contains at most $O(g^2 + \frac{n \log n}{g})$ vertices, and so the number of edges within is bounded by $\tilde{O}(g^4 + \frac{n^2}{g^2})$, and hence the runtime of multi-source shortest paths computation for all vertices in $V(\beta_l^i)$ in H_l^i is $\tilde{O}(g^5 + \frac{n^2}{g})$. Then, for each $f \in E(\beta_l^i)$, the time of calculating the formula

$$\min_{x \in V(\beta_l^i[*,f)), y \in V(\beta_l^i(f,*])} \{\mathsf{dist}(s, x, G \setminus \{e_i\}) + \mathsf{dist}(x, y, H_i) + \mathsf{dist}(y, t, G \setminus \{e_i\})\}$$

is bounded by $O(g^2)$. Summing over all i, l and $f \in E(\beta_l^i)$, the runtime of this part is bounded by $\tilde{O}(L^2 g^4 + \frac{n^2 L^2}{g^2})$.

Taking a summation, the overall runtime for this case is at most:

$$\tilde{O}\left(mnL^2/g^3 + mLg + L^2 g^4 + n^2 L^2/g^2\right)$$

Correctness. To prove that our algorithm computes the correct value for $|\rho_{i,f}|$, it suffices to prove that $\mathsf{dist}\left(x_{i,f}, y_{i,f}, H_l^i\right) = |\rho_{i,f}[x_{i,f}, y_{i,f}]|$. First we argue that $\mathsf{dist}\left(x_{i,f}, y_{i,f}, H_l^i\right) \geq |\rho_{i,f}[x_{i,f}, y_{i,f}]|$ due to the following reason.

▷ Claim 14. Any weighted edge (u, v) in H_l^i corresponds to a path from u to v in G that does not contain any edge in $E(\pi) \cup E(\beta_l^i)$.

Proof. If the weighted edge (u, v) was defined on Step (3)(a), then it corresponds to a shortest path in G_{l,l_i-l+1} which excludes all edges in $E(\pi) \cup E(\beta_l^i)$.

If the weighted edge (u, v) was defined on Step (3)(b), suppose $u \in V(\beta_l^j)$ where $j \in [i - 5g, i]$, then by definition of $\omega(u, v) = \mu_X(u, v)$, which is equal to the length of a path in graph X_l^j which excludes all edges in $E(\pi) \cup E(\beta_l^i)$.

Symmetrically, if the weighted edge (u, v) was defined on Step (3)(c), suppose $u \in V(\beta_l^j)$ where $j \in [i, i + 5g]$, then by definition of $\omega(u, v) = \mu_Y(u, v)$, it is equal to the length of a path in graph Y_l^j which excludes all edges in $E(\pi) \cup E(\beta_l^i)$. ◁

For the rest, let us prove that $\mathsf{dist}\left(x_{i,f}, y_{i,f}, H_l^i\right) \leq |\rho_{i,f}[x_{i,f}, y_{i,f}]|$. To do this, we will find a path in H_l^i from $x_{i,f}$ to $y_{i,f}$ with weighted length at most $|\rho_{i,f}[x_{i,f}, y_{i,f}]|$.

▷ Claim 15. $\rho_{i,f}[x_{i,f}, y_{i,f}]$ does not contain any vertices in the following vertex subset:

$$\bigcup_{j=0}^{i-5g-1} V\left(\beta_l^j\right) \cup \bigcup_{j=i+5g+1}^{L} V\left(\beta_{l_j-l_i+l}^j\right)$$

Proof. Suppose otherwise that the sub-path $\rho_{i,f}[x_{i,f}, y_{i,f}]$ contains a vertex $v \in \bigcup_{j=0}^{i-5g-1} V(\beta_l^j) \cup \bigcup_{j=i+5g+1}^{L} V(\beta_{l_j-l_i+l}^j)$. Let us assume that $v \in V(\beta_l^j)$ for some index $0 \leq j < i - 5g$; if $v \in V(\beta_{l_j-l_i+l}^j)$ for some $j > i + 5g$, the proof will be similar.

By the assumption that $i \in I$, we know that $|\pi[a_i, u_i]| \leq g$. Since $j < i - 5g$, we know that vertex u_j lies between s, a_i, and consequently $|\pi[s, a_j]| \leq |\pi[s, u_j]| < |\pi[s, a_i]| - 4g$.

Consider the path $\pi[s, a_j] \circ \alpha_j[a_j, v]$. We first argue that this path does not contain edges from $\{e_i, f\}$. Clearly, $\pi[s, a_j] \circ \alpha_j[a_j, v]$ does not contain the edge e_i, since a_j lies between s and u_i, and $E(\alpha_j[a_j, v]) \cap E(\pi) = \emptyset$. As for the position of f, if $f \in E(\alpha_j[a_j, v])$, then there must be a vertex $z \in V(\alpha_j[a_j, v]) \cap V(\beta_l^i)$. Then $\pi[s, a_j] \circ \alpha_j[a_j, z] \circ \alpha_i[z, b_i] \circ \alpha_i[b_i, t]$ is a replacement path that avoids edge e_i, with length at most:

$$\begin{aligned}
&|\pi[s, a_j]| + |\alpha_j[a_j, z]| + |\alpha_i[z, b_i]| + |\alpha_i[b_i, t]| \\
&< (|\pi[s, a_i]| - 4g) + |\alpha_j[a_j, v]| + |\alpha_i[z, b_i]| + |\alpha_i[b_i, t]| \\
&\leq (|\pi[s, a_i]| - 4g) + l \cdot g + |\alpha_i[z, b_i]| + |\alpha_i[b_i, t]| \\
&\leq (|\pi[s, a_i]| - 4g) + (|\alpha_i[a_i, z]| + g) + |\alpha_i[z, b_i]| + |\alpha_i[b_i, t]| \\
&\leq |\pi[s, a_i] + |\alpha_i[a_i, z]| + |\alpha_i[z, b_i]| + |\alpha_i[b_i, t]| - 3g = |\pi[s, a_i] \circ \alpha_i \circ \pi[b_i, t]| - 3g
\end{aligned}$$

which is a contradiction that $\pi[s, a_i] \circ \alpha_i \circ \pi[b_i, t]$ is a shortest replacement path avoiding e_i.

Next, we argue that $|\pi[s, a_j] \circ \alpha_j[a_j, v]| < |\rho_{i,f}[s, x_{i,f}]| < |\rho_{i,f}[s, v]|$. In fact, by $|\pi[s, a_j]| < |\pi[s, a_i]| - 4g$, we have:

$$\begin{aligned}
&|\pi[s, a_j]| + |\alpha_j[a_j, v]| < (|\pi[s, a_i]| - 4g) + l \cdot g \\
&\leq (|\pi[s, a_i]| - 4g) + (|\alpha_i[a_i, x_{i,f}]| + g) = |\rho_{i,f}[s, x_{i,f}]| - 3g
\end{aligned}$$

As we have proved, $\pi[s, a_j] \circ \alpha_j[a_j, v]$ is a path avoiding $\{e_i, f\}$ of length less than $|\rho_{i,f}[s, v]|$. So, if we replace the prefix $\rho_{i,f}[s, v]$ with $\pi[s, a_j] \circ \alpha_j[a_j, v]$ and consider a new path:

$$\rho' = \pi[s, a_j] \circ \alpha_j[a_j, v] \circ \rho_{i,f}[v, t]$$

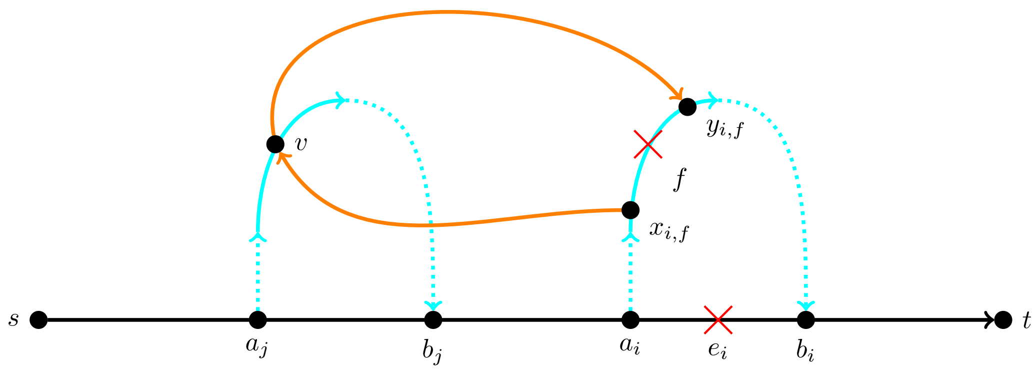


Figure 12 The sub-path $\rho_{i,f}[x_{i,f}, y_{i,f}]$ is drawn as the orange curve. If $\rho_{i,f}[x_{i,f}, y_{i,f}]$ contains a vertex $v \in V\left(\beta_l^j\right)$, then the alternate path $\pi[s, a_j] \circ \alpha_j[*, v] \circ \rho_{i,f}[v, y_{i,f}] \circ \alpha_i[y_{i,f}, b_i] \circ \pi[b_i, t]$ be a shorter replacement path than $\rho_{i,f}$ avoiding $\{e_i, f\}$.

then we have a new replacement path ρ' avoiding $\{e_i, f\}$ from s to t with a strictly better distance, a contradiction. See Figure 12 for an illustration.

$\triangleleft$

Decompose the path $\rho_{i,f}[x_{i,f}, y_{i,f}]$ into minimal sub-paths whose endpoints are belonging to $V(H_l^i)$; this is achievable because both $x_{i,f}, y_{i,f} \in V(H_l^i)$. To prove the upper bound:

$$\mathsf{dist}(x_{i,f}, y_{i,f}, H_l^i) \leq |\rho_{i,f}[x_{i,f}, y_{i,f}]|$$

it suffices to show that for any two consecutive vertices $u, v \in V(H_l^i)$ on path $\rho_{i,f}[x_{i,f}, y_{i,f}]$, we have $\omega(u, v) \leq |\rho_{i,f}[u, v]|$. Consider several cases below.

- One of u, v belongs to the pivot set U.
  Without loss of generality, assume that $u \in U$. It suffices to show that the shortest path from u to v in graph G_{l,l_i-l+1} is has the same length as $\rho_{i,f}[u, v]$.
  Since v is the next vertex in $V(H_l^i)$ after u on the path $\rho_{i,f}$, the sub-path $\rho_{i,f}[u, v]$ does not contain any vertices from $V(H_l^i)$ except for its endpoints; in other words, $\rho_{i,f}[u, v]$ does not contain vertices from $U \cup \bigcup_{j=i-5g}^{i+5g} V(\beta_l^j)$. Additionally, according to Claim 15, $\rho_{i,f}[u, v]$ does not contain (internally) any vertices in

  $$\bigcup_{j=0}^{i-5g-1} V\left(\beta_l^j\right) \cup \bigcup_{j=i+5g+1}^{L} V\left(\beta_{l_j-l_i+l}^j\right)$$

  Therefore, $\rho_{i,f}[u, v]$ does not contain (internally) any vertices from the set:

  $$U \cup \bigcup_{j\in[i-5g,i+5g]\cap I} V\left(\beta_l^j\right) \cup \bigcup_{j=0}^{i-5g-1} V\left(\beta_l^j\right) \cup \bigcup_{j=i+5g+1}^{L} V\left(\beta_{l_j-l_i+l}^j\right) \supseteq U \cup \bigcup_{\beta\in\mathcal{P}_{l,l_i-l+1}} V(\beta)$$

  By definition of graph G_{l,l_i-l+1}, we know that $\mathsf{dist}(u, v, G_{l,l_i-l+1}) \leq |\rho_{i,f}[u, v]|$.
- Both of u, v are not in U.
  Assume that $u \in V(\beta_l^c)$ for some $c \in [i - 5g, i + 5g]$, and $v \in V(\beta_l^d)$ for some $d \in [i - 5g, i + 5g]$. Without loss of generality, assume that $c \leq i$; if $c \geq i$, a symmetric argument would work.

Since v is the next vertex in $V(H_l^i)$ after u on the path $\rho_{i,f}$ and that U is a uniformly random vertex set of size $\frac{10n \log n}{g}$, with high probability over the randomness of U, we must have $|\rho_{i,f}[u,v]| \leq g$.
Similar to the previous case, we know that $\rho_{i,f}[u,v]$ does not contain (internally) vertices from $\bigcup_{j=i-5g}^{i+5g} V\left(\beta_l^j\right)$. Therefore, by definition of X_l^c, we know that $\rho_{i,f}[u,v] \subseteq X_l^c$; namely, the sub-path $\rho_{i,f}[u,v]$ belongs to graph X_l^c. Therefore, to prove that $\omega(u,v) = \mu_X(u,v) \leq |\rho_{i,f}[u,v]|$, it suffices to show that the truncated Dijkstra procedure correctly computes the value $\mu_X(u,v) = \mathsf{dist}(u,v,X_l^c)$.
Decompose the integer $c = b + 10kg$, where $1 \leq b \leq 10g$, $k \geq 0$. To prove that the truncated Dijkstra procedure correctly computes the value $\mu_X(u,v) = \mathsf{dist}(u,v,X_l^c)$, it suffices to show that the vertex set $A_{b,l}$ contains all vertices on $\rho_{i,f}[u,v]$ when u is performing a truncated Dijkstra in the induced subgraph $X_l^c[A_{b,l}]$; in other words, we need to show that any vertex on $\rho_{i,f}[u,v]$ has not been pruned from $A_{b,l}$ by truncated Dijkstraes from vertices on previous sub-paths $\beta_l^b, \beta_l^{b+10g}, \cdots, \beta_l^{b+10(k-1)g}$.
Assume otherwise there is a vertex $z \in V(\rho_{i,f}[u,v])$ which was also visited by the truncated Dijkstra of some vertices $w \in V(\beta_l^{b+10jg})$ for some $j < k$. As all Dijkstra searches are truncated up to depth g, we know that there is a path γ from w to z in X_l^{b+10jg} of length at most g. Consider the path

$$\theta = \pi[s, a_{b+10jg}] \circ \alpha_{b+10jg}[*, w] \circ \gamma \circ \rho_{i,f}[z,v] \circ \alpha_d[v, b_d] \circ \pi[b_d, t]$$

and claim two properties of it.

▷ Claim 16. θ is a path from s to t that avoids the edge e_d.

Proof. It is clear that path θ departs from π at vertex a_{b+10jg} and converges with π at vertex b_d. So it suffices to show that a_{b+10jg} lies between s and vertex u_d. This is straightforward since a_{b+10jg} lies on path $\pi[s, u_{b+10jg}]$ which is strictly a prefix of $\pi[s, u_d]$, as $d \geq i - 5g \geq b + (10k-5)g > b + 10jg$. ◁

To reach a contradiction, we show that $|\theta|$ is a strictly better replacement path from s to t that avoids e_d than path $\pi[s, a_d] \circ \alpha_d \circ \pi[b_d, t]$. In fact, on the one hand, since $d \in I$, we know that $|\pi[a_d, u_d]| \leq g$. Therefore,

$$|\pi[s, a_{b+10jg}]| \leq |\pi[s, u_{b+10jg}]| = b+10jg \leq 10(k-1)g \leq i-10g \leq d-5g \leq |\pi[s, a_d]|-4g$$

On the other hand, since $v \in V(\beta_l^d)$ and $w \in V(\beta_l^{b+10jg})$, we know that

$$|\alpha_d[*, v]| \geq (l-1)g + 1 > |\alpha_{b+10jg}[*, w]| - g$$

Finally, as $|\gamma|, |\rho_{i,f}[z,v]| \leq g$, we have:

$$\begin{aligned}
|\theta| &\leq (|\pi[s,a_d]| - 4g) + (|\alpha_d[*,v]| + g) + |\gamma| + |\rho_{i,f}[z,v]| + |\alpha_d[v,b_d]| + |\pi[b_d,t]| \\
&\leq |\pi[s,a_d]| + |\alpha_d[*,v]| + |\alpha_d[v,b_d]| + |\pi[b_d,t]| - g \\
&= |\pi[s,a_d] \circ \alpha_d \circ \pi[b_d,t]| - g
\end{aligned}$$

which contradicts the fact that $\pi[s, a_d] \circ \alpha_d \circ \pi[b_d, t]$ is the shortest replacement path avoiding e_d. See Figure 13 for an illustration.

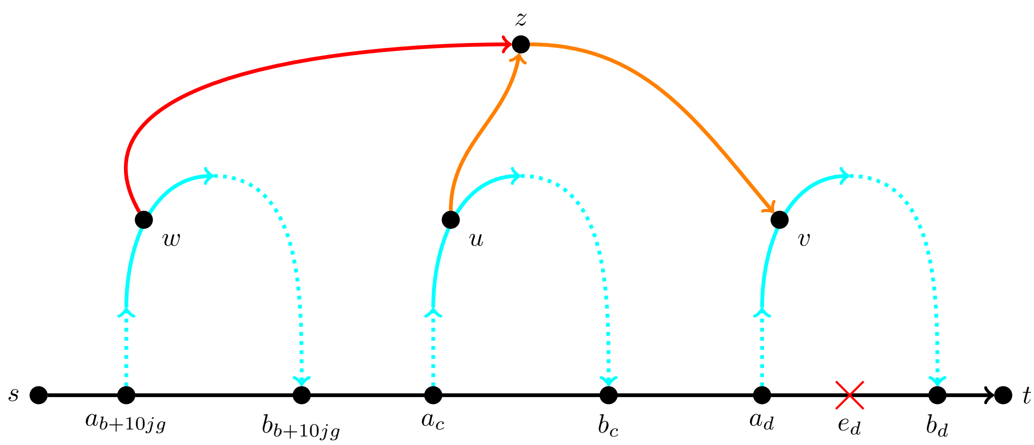


Figure 13 If $\mu_X(u, v)$ does not capture the orange path $\rho_{i,f}[u, v]$, then a previous Dijkstra search from vertex w must have intercepted $\rho_{i,f}[u, v]$ at a vertex z through a path γ drawn as the red curve. In this case, $\alpha_{b+10jg}[*, w] \circ \gamma \circ \rho_{i,f}[z, v] \circ \alpha_d[v, b_d]$ would be a better detour than α_d for avoiding e_d.

Proof of Lemma 13

Summarizing the total runtime of all five cases, the overall runtime is bounded by as following:

$$\tilde{O}\left(\frac{mn^{1.5} + n^3}{L} + mn^{1/2}L/g + n^2L/g + mL^2/g + mnL^2/g^3 + mLg + L^2g^4 + n^2L^2/g^2\right)$$

4 One failure on a long st-path

In this section, we study the case where only one edge failure lies on the shortest path, plus that the input graph is dense and $\mathsf{dist}(s, t, G)$ could be as large as $O(n)$. Let $G = (V, E)$ be a digraph with n vertices, and consider a pair of vertices s, t as well as an st-shortest path $\pi = \langle s = u_0, u_1, u_2, \cdots, u_{|\pi|} = t\rangle$. The task is to compute for any pairs of edges f_1, f_2, the value of $\mathsf{dist}(s, t, G \setminus \{f_1, f_2\})$ where $f_1 \in E(\pi), f_2 \notin E(\pi)$. The following lemma is proved in the full version.

▶ **Lemma 17.** *All values of* $\mathsf{dist}(s, t, G \setminus \{f_1, f_2\})$ *can be computed with high probability in time* $\tilde{O}\left(n^{3-1/18}\right)$ *when* $f_1 \in E(\pi)$ *while* $f_2 \notin E(\pi)$.

5 Both failures on the st-path

In this section, we study the case where both edge failures f_1, f_2 are lying the shortest path. Let $G = (V, E)$ be a digraph with n vertices, and consider a pair of vertices s, t as well as an st-shortest path $\pi = \langle s = u_0, u_1, u_2, \cdots, u_{|\pi|} = t\rangle$. For convenience, define $H = G \setminus E(\pi)$. The task is to compute for any pairs of edges $f_1, f_2 \in V(\pi)$, the value of $\mathsf{dist}(s, t, G \setminus \{f_1, f_2\})$. The following lemma is proved in the full version.

▶ **Lemma 18.** *All values of* $\mathsf{dist}(s, t, G \setminus \{f_1, f_2\})$ *can be computed in time* $\tilde{O}(n^{3-1/7})$ *when both edges* f_1, f_2 *are on* π.

Proof of Theorem 1. This is a direct combination of Lemma 17 and Lemma 18. ◀

References

1 Josh Alman and Virginia Vassilevska Williams. A refined laser method and faster matrix multiplication. In *Proceedings of the 2021 ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 522–539. SIAM, 2021.

2 Aaron Bernstein. A nearly optimal algorithm for approximating replacement paths and k shortest simple paths in general graphs. In *Proceedings of the twenty-first annual ACM-SIAM symposium on Discrete Algorithms*, pages 742–755. SIAM, 2010.

3 Aaron Bernstein, Danupon Nanongkai, and Christian Wulff-Nilsen. Negative-weight single-source shortest paths in near-linear time. In *2022 IEEE 63rd Annual Symposium on Foundations of Computer Science (FOCS)*, pages 600–611. IEEE, 2022.

4 Amit M Bhosle and Teofilo F Gonzalez. Replacement paths for pairs of shortest path edges in directed graphs. In *Proceedings of the 16th IASTED International Conference on Parallel and Distributed Computing and Systems*. Citeseer, 2004.

5 Shiri Chechik and Sarel Cohen. Distance sensitivity oracles with subcubic preprocessing time and fast query time. In *Proceedings of the 52nd Annual ACM SIGACT Symposium on Theory of Computing*, pages 1375–1388, 2020.

6 Shiri Chechik and Ofer Magen. Near optimal algorithm for the directed single source replacement paths problem. In *47th International Colloquium on Automata, Languages, and Programming (ICALP 2020)*. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2020.

7 Shiri Chechik and Moran Nechushtan. Simplifying and unifying replacement paths algorithms in weighted directed graphs. In *47th International Colloquium on Automata, Languages, and Programming (ICALP 2020)*. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2020.

8 Shiri Chechik and Tianyi Zhang. Nearly optimal approximate dual-failure replacement paths. In *Proceedings of the 2024 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 2568–2596. SIAM, 2024.

9 Edsger W Dijkstra. A note on two problems in connexion with graphs. In *Edsger Wybe Dijkstra: His Life, Work, and Legacy*, pages 287–290. 2022.

10 Ran Duan, Hongxun Wu, and Renfei Zhou. Faster matrix multiplication via asymmetric hashing. In *2023 IEEE 64th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 2129–2138. IEEE, 2023.

11 Fabrizio Grandoni and Virginia Vassilevska Williams. Improved distance sensitivity oracles via fast single-source replacement paths. In *2012 IEEE 53rd Annual Symposium on Foundations of Computer Science*, pages 748–757. IEEE, 2012.

12 Fabrizio Grandoni and Virginia Vassilevska Williams. Faster replacement paths and distance sensitivity oracles. *ACM Transactions on Algorithms (TALG)*, 16(1):1–25, 2019.

13 Yuzhou Gu, Adam Polak, Virginia Vassilevska Williams, and Yinzhan Xu. Faster monotone min-plus product, range mode, and single source replacement paths. In *48th International Colloquium on Automata, Languages, and Programming (ICALP 2021)*. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2021.

14 François Le Gall. Powers of tensors and fast matrix multiplication. In *Proceedings of the 39th international symposium on symbolic and algebraic computation*, pages 296–303, 2014.

15 Liam Roditty. On the k-simple shortest paths problem in weighted directed graphs. In *Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms*, pages 920–928. Citeseer, 2007.

16 Liam Roditty and Uri Zwick. Replacement paths and k simple shortest paths in unweighted directed graphs. In *Automata, Languages and Programming: 32nd International Colloquium, ICALP 2005, Lisbon, Portugal, July 11-15, 2005. Proceedings 32*, pages 249–260. Springer, 2005.

17 Virginia Vassilevska Williams. Faster replacement paths. In *Proceedings of the twenty-second annual ACM-SIAM symposium on Discrete Algorithms*, pages 1337–1346. SIAM, 2011.

18 Virginia Vassilevska Williams. Multiplying matrices faster than coppersmith-winograd. In *Proceedings of the forty-fourth annual ACM symposium on Theory of computing*, pages 887–898, 2012.

19 Virginia Vassilevska Williams and Ryan Williams. Subcubic equivalences between path, matrix and triangle problems. In *2010 IEEE 51st Annual Symposium on Foundations of Computer Science*, pages 645–654. IEEE, 2010.

20 Virginia Vassilevska Williams, Eyob Woldeghebriel, and Yinzhan Xu. Algorithms and lower bounds for replacement paths under multiple edge failure. In *2022 IEEE 63rd Annual Symposium on Foundations of Computer Science (FOCS)*, pages 907–918. IEEE, 2022.

21 Virginia Vassilevska Williams, Yinzhan Xu, Zixuan Xu, and Renfei Zhou. New bounds for matrix multiplication: from alpha to omega. In *Proceedings of the 2024 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 3792–3835. SIAM, 2024.

22 Uri Zwick. All pairs shortest paths using bridging sets and rectangular matrix multiplication. *Journal of the ACM (JACM)*, 49(3):289–317, 2002.