Fault-Tolerant ST-Diameter Oracles

Davide Bilò

Department of Information Engineering, Computer Science and Mathematics, University of L'Aquila, Italy

Keerti Choudhary

Department of Computer Science and Engineering, Indian Institute of Technology Delhi, India

Sarel Cohen

School of Computer Science, Tel-Aviv-Yaffo Academic College, Israel

Tobias Friedrich

Hasso Plattner Institute, Universität Potsdam, Germany

Simon Krogmann

Hasso Plattner Institute, Universität Potsdam, Germany

Martin Schirneck

Faculty of Computer Science, Universität Wien, Austria

Abstract

We study the problem of estimating the ST-diameter of a graph that is subject to a bounded number of edge failures. An *f-edge fault-tolerant ST-diameter oracle* (f-FDO-ST) is a data structure that preprocesses a given graph G, two sets of vertices S, T, and positive integer f. When queried with a set F of at most f edges, the oracle returns an estimate $\widehat{D}$ of the ST-diameter $\operatorname{diam}(G-F, S, T)$, the maximum distance between vertices in S and T in $G-F$. The oracle has stretch $\sigma \geqslant 1$ if $\operatorname{diam}(G-F, S, T) \leqslant \widehat{D} \leqslant \sigma \operatorname{diam}(G-F, S, T)$. If S and T both contain all vertices, the data structure is called an *f-edge fault-tolerant diameter oracle* (f-FDO). An *f-edge fault-tolerant distance sensitivity oracles* (f-DSO) estimates the pairwise graph distances under up to f failures.

We design new f-FDOs and f-FDO-STs by reducing their construction to that of *all-pairs* and *single-source* f-DSOs. We obtain several new tradeoffs between the size of the data structure, stretch guarantee, query and preprocessing times for diameter oracles by combining our black-box reductions with known results from the literature.

We also provide an information-theoretic lower bound on the space requirement of approximate f-FDOs. We show that there exists a family of graphs for which any f-FDO with sensitivity $f \geqslant 2$ and stretch less than 5/3 requires $\Omega(n^{3/2})$ bits of space, regardless of the query time.

2012 ACM Subject Classification Theory of computation → Shortest paths; Theory of computation → Data structures design and analysis; Theory of computation → Cell probe models and lower bounds

Keywords and phrases diameter oracles, distance sensitivity oracles, space lower bounds, fault-tolerant data structures

Digital Object Identifier 10.4230/LIPIcs.ICALP.2023.24

Category Track A: Algorithms, Complexity and Games

Related Version *Full Version*: https://arxiv.org/abs/2305.03697

1 Introduction

The diameter, i.e., the largest distance between any two vertices, is one of the most fundamental graph parameters for it measures how fast information can spread in a network. The problem of approximating the diameter of a given graph in a time-efficient manner has been extensively studied [1, 3, 4, 25, 26, 27, 43, 44, 45]. Here, we investigate the diameter computation problem in the fault-tolerant model. The interest in this setting stems from the fact that most real-world networks are prone to errors. These failures, though unpredictable, are transient due to some simultaneous repair process that is undertaken in these applications. This has motivated the research on designing fault-tolerant oracles for various graph problems in the past decade. An *f-edge/vertex fault-tolerant oracle* is a compact data structure that can quickly report the desired solution or graph property of the network on occurrence of up to f edge/vertex failures. The parameter f that describes the degree of robustness against errors is known as the *sensitivity* of the oracle. A lot of work has been done in designing fault-tolerant structures for various problems like connectivity [20, 32, 33], finding shortest paths [2, 12, 36], and distance sensitivity oracles [5, 7, 14, 24, 30, 31, 34, 47].

While the fault-tolerant model has been studied a lot for distances, the landscape of fault-tolerant diameter oracles is far less explored. For a given graph $G = (V, E)$ and two sets $S, T \subseteq V$ of vertices, an *f-edge fault-tolerant diameter oracle* (f-FDO-ST) is a data structure that stores information about G after a preprocessing step. When queried with a set F of at most f edges, the oracle returns an upper bound of the ST-diameter $\operatorname{diam}(G - F, S, T) = \max_{s \in S, t \in T} d_{G-F}(s, t)$ of $G - F$. This is the maximum among all s-t-distances for $s \in S$ and $t \in T$ under the condition that none of the shortest paths can use an edge in the query set F. We say that the oracle has a stretch of $\sigma \geqslant 1$ if the value $\widehat{D}$ returned upon query F satisfies $\operatorname{diam}(G - F, S, T) \leqslant \widehat{D} \leqslant \sigma \operatorname{diam}(G - F, S, T)$. When $S = T = V$, the data structure is called an *f-edge fault-tolerant diameter oracle* (f-FDO).

The problem of designing f-FDOs was originally raised by Henzinger, Lincoln, Neumann, and Vassilevska Williams [40] and has recently been studied by Bilò, Cohen, Friedrich, and Schirneck [17] and the same authors together with Choudhary [15], see also Section 1.1.

The problem of designing f-FDO-ST can be seen as a generalisation of the Bichromatic Diameter, a problem in which the two sets S and T form a partition of V. The latter problem is motivated by several related, well-studied problems in computational geometry, e.g., Bichromatic Diameter on point sets (commonly known as Bichromatic Farthest Pair), where one seeks to determine the farthest pair of points in a given set of points in space. The problem of Bichromatic Diameter was studied by Dalirrooyfard, Vassilevska Williams, Vyas, and Wein [28].

Given the plethora of work on distance oracles and the close connection between the distance and the diameter problem, a natural question is if we can convert the results on distance computation under failures into analogous oracles for the diameter without sacrificing much on our performance parameters.

► **Question.** *Are there black-box reductions from fault-tolerant diameter oracles to fault-tolerant distance oracles without considerable overhead in stretch, query time, and space?*

In this work, we present several such reductions and, from them, conclude trade-offs between the space, stretch, preprocessing, and query time for diameter oracles. In more detail, our techniques for obtaining upper bounds is by presenting reductions to the problem of constructing *f-edge fault-tolerant distance sensitivity oracles* (f-DSOs) in two widely studied categories. The *all-pairs* variant can be queried with any pair of vertices $s, t \in V$ and set $F \subseteq E$ of f failures and reports (an estimate) of the distance $d_{G-F}(s, t)$ between s and t in $G - F$. In the *single-source* variant, the source s is fixed and the set of allowed queries consists of the target vertices t together with a set F of failures.

For the regular diameter ($S = T = V$), we provide two theorems showing that both all-pairs and single-source f-DSOs can be used to construct f-FDOs.

Table 1 Properties of the f-FDOs obtained via Theorem 1 using all-pairs f-DSOs from the literature. The applicable graph class (un-/directed, un-/weighted) is determined by the f-DSO. W denotes the maximum edge weight for graphs with arbitrary positive weights, M is the maximum edge weight for integer weighted graphs. The parameter $k \geqslant 1$ is a positive integer, $\varepsilon > 0$ a positive real, $\alpha \in [0,1]$ is a real number in the unit interval, and $\omega < 2.37286$ denotes the matrix multiplication exponent.

Sensitivity	Stretch	Space	Query time	Preprocessing Time	Ref.
1	2	$\widetilde{O}(n^2)$	$O(1)$	$\widetilde{O}(mn)$	[10, 11]
1	2	$\widetilde{O}(n^2)$	$O(1)$	$\widetilde{O}(n^{2.5794}M + mn)$	[37]
1	$1 + (2k-1)(1+\varepsilon)$	$\widetilde{O}(k^5 n^{1+1/k} \varepsilon^{-4})$	$O(k)$	$O(kmn^{1+1/k})$	[8]
2	2	$\widetilde{O}(n^2)$	$\widetilde{O}(1)$	$\mathsf{poly}(n)$	[32]
$f = o(\frac{\log n}{\log\log n})$	2	$\widetilde{O}(n^{3-\alpha})$	$\widetilde{O}(f^2 n^{2-(1-\alpha)/f})$	$O(n^{\omega+1-\alpha}M)$	[47]
$f = o(\frac{\log n}{\log\log n})$	$2+\varepsilon$	$O(fn^{2+o(1)}(\log W)\varepsilon^{-f})$	$\widetilde{O}(f^7 \log\log W)$	$O(fn^{5+o(1)}(\log W)\varepsilon^{-f})$	[23]
$f \geqslant 1$	2	$O(fn^4)$	$f^{O(f)}$	$n^{O(f)}$	[34]
$f \geqslant 1$	2	$O(n^{2+\alpha}M)$	$\widetilde{O}(f^4 n^{2-\alpha}M + f^{2+\omega}nM)$	$\widetilde{O}(n^{\omega+(3-\omega)\alpha}M + mn)$	[18]
$f \geqslant 1$	$1 + (8k-2)(f+1)$	$O\left(fkn^{1+1/k} \log(nW)\right)$	$\widetilde{O}(f^3)$	$\mathsf{poly}(n)$	[24]

▶ **Theorem 1.** *Let G be a (undirected or directed) graph with n vertices, m edges, and possibly positive edge weights. Given access to an f-DSO $\mathcal{D}$ for G with stretch $\sigma \geqslant 1$, preprocessing time $\mathtt{P}$, space $\mathtt{S}$, and query time $\mathtt{Q}$, one can construct an f-FDO for G with stretch $1+\sigma$, preprocessing time $O(\mathtt{P} + mn\log n)$, space $O(\mathtt{S})$, and query time $O(f^2\mathtt{Q})$.*

In Section 1.2, we review existing all-pairs f-DSOs. By applying the reduction stated in Theorem 1 we obtain new f-FDOs as listed in Table 1.

The following theorem shows how we can use the single-source variant of distance sensitivity oracles to construct f-FDOs.

▶ **Theorem 2.** *Let G be a (undirected or directed) graph with n vertices, m edges, and possibly positive edge weights. Given access to a single-source f-DSO $\mathcal{D}$ for G with stretch $\sigma \geqslant 1$, preprocessing time $\mathtt{P}$, space $\mathtt{S}$, and query time $\mathtt{Q}$, one can construct an f-FDO for G with stretch $2(1+\sigma)$, preprocessing time $O(\mathtt{P})$, space $O(\mathtt{S})$, and query time $O(f\mathtt{Q})$.*

Section 1.3 discusses single-source f-DSOs from the literature. Together with Theorem 2 they give new f-FDOs, summarized in Table 2.

The main technical contribution of this work, however, is a novel fault-tolerant data structure for the more general ST-diameter problem that was introduced and studied in recent years. For example, Backurs, Roditty, Segal, Vassilevska Williams, and Wein [4] proved that for any undirected graph one can compute a 3-approximation of the ST-diameter in $O(mn)$ time. They also provided a randomized algorithm that computes a 2-approximation of the ST-diameter in $\widetilde{O}(m\sqrt{n})$ time.[1] Dalirrooyfard, Vassilevska Williams, Vyas, and Wein [28] studied the problem of computing the bi-chromatic ST-diameter, the special case of ST-diameter problem where the sets S and T form a partition of V. Similar to f-FDOs, we explore the problem of designing compact oracles that report the ST-diameter of a graph after occurrences of up to f failures. We present reductions between f-DSOs and f-FDO-STs, as stated in the following theorem. To the best of our knowledge, our paper is the first work that provides some results on f-FDO-STs, for general values of f.

[1] For a non-negative function $g(n,m,f)$, we write $\widetilde{O}(g)$ for $O(g \cdot \mathsf{polylog}(n))$.

Table 2 Properties of the f-FDOs obtained via Theorem 2 using single-source f-DSOs from the literature. The applicable graph class (un-/directed, un-/weighted) is determined by the single-source f-DSO. W denotes the maximum edge weight for graphs with arbitrary positive weights, M is the maximum edge weight for integer weighted graphs. The parameter $\varepsilon > 0$ is a positive real and $\omega < 2.37286$ denotes the matrix multiplication exponent.

Sensitivity	Stretch	Space	Query time	Preprocessing Time	Ref.
1	4	$\widetilde{O}(n^{3/2})$	$\widetilde{O}(1)$	$\widetilde{O}(mn^{1/2} + n^2)$	[16, 38]
1	4	$\widetilde{O}(n^{3/2}M^{1/2})$	$\widetilde{O}(1)$	$\widetilde{O}(n^{\omega}M)$	[16]
1	$4+\varepsilon$	$\widetilde{O}(n\,(\log W)\,\varepsilon^{-1})$	$O(\log\log_{1+\varepsilon}(nW))$	$\mathsf{poly}(n)$	[5, 8, 13]
1	6	$O(n)$	$O(1)$	$\widetilde{O}(mn)$	[13]
$f \geqslant 1$	$4f+4$	$\widetilde{O}(fn)$	$\widetilde{O}(f^3)$	$\widetilde{O}(fm)$	[14]

▶ **Theorem 3.** *Let $G = (V, E)$ be an undirected graph with n vertices, m edges, and possibly positive edge weights. Let $S, T \subseteq V$ be two non-empty sets. Given access to an f-DSO for G with stretch $\sigma \geqslant 1$, preprocessing time $\mathtt{P}$, space $\mathtt{S}$, and query time $\mathtt{Q}$, one can compute an f-FDO-ST for G with preprocessing time $\mathtt{P} + \widetilde{O}(mn + n|S||T|)$ and stretch $1 + 3\sigma$. Additionally, the f-FDO-ST has the following properties.*

- *If the sensitivity is $f = o(\log n)$, the oracle requires $\mathtt{S} + O(n^{3/2}\,(2^f + \log n))$ space and has a query time of $O(f^2\,(2^f + \mathtt{Q}))$.*
- *If $f = \Omega(\log n)$, the oracle requires $\mathtt{S} + O(n^2)$ space and has a query time of $O(f^2(f + \mathtt{Q}))$.*

Some more remarks on the preprocessing time stated in Theorem 3 may be in order. The reduction itself takes time $\mathtt{P} + O(mn + n^2 \log n + n|S||T|)$ to compute but requires that the shortest paths in G are unique. The total preprocessing time depends on how this condition is achieved. It is always possible to guarantee unique shortest paths either by randomly perturbing the edge weights with sufficiently small values, see [41], or by using a more complex deterministic method, also known as *lexicographic perturbation* [19, 21, 39]. While the first method increases the preprocessing only by a constant factor, it makes the preprocessing procedure randomized. Lexicographic perturbation, in turn, increases the time by an additive $O(mn + n^2 \log^2 n)$ term [19]. By applying the reduction stated in Theorem 3 to existing all-pairs f-DSOs we obtain the f-FDOs listed in Table 3.

In addition, we present improved constructions of f-FDO-STs for the important case of a single source or target, i.e., when $|S| = 1$ or $|T| = 1$, or when one is only given access to single-source f-DSOs. In the following, for the sake of readability, when $S = \{s\}$, we will use "sT-diameter" instead of "ST-diameter" or "$\{s\}T$-diameter", same for the oracles.

▶ **Theorem 4.** *Let $G = (V, E)$ be an undirected graph with n vertices, m edges, and possibly positive edge weights. Let $s \in V$ be a vertex and $T \subseteq V$ a non-empty set. Given a single-source f-DSO for G with preprocessing time $\mathtt{P}$, space $\mathtt{S}$, query time $\mathtt{Q}$, and stretch σ, one can compute an f-FDO-sT for G with preprocessing time $\mathtt{P} + O(m + n\log n)$, space $\mathtt{S} + O(n)$, query time $O(f^2 + f\mathtt{Q})$, and stretch $1 + 2\sigma$. For unweighted graphs, the preprocessing time can be improved to $\mathtt{P} + O(m)$.*

Table 4 shows the f-fault-tolerant sT-diameter-oracle obtained from Theorem 4.

Table 3 Properties of the f-FDO-ST for undirected graphs obtained via Theorem 3 using all-pairs f-DSOs from the literature. The preprocessing time is omitted due to space reasons. W denotes the maximum edge weight for graphs with arbitrary positive weights, M is the maximum edge weight for integer weighted graphs. The parameter $k \geqslant 1$ is a positive integer, $\varepsilon > 0$ a positive real, $\alpha \in [0,1]$ is a real number in the unit interval, and $\omega < 2.37286$ denotes the matrix multiplication exponent.

Sensitivity	Stretch	Space	Query time	Ref.
1	4	$\widetilde{O}(n^2)$	$O(1)$	[10, 11, 37]
1	$1+(6k-3)(1+\varepsilon)$	$\widetilde{O}(n^{3/2}+k^5 n^{1+1/k}\varepsilon^{-4})$	$O(1)$	[8]
2	4	$\widetilde{O}(n^2)$	$\widetilde{O}(1)$	[32]
$f=o(\frac{\log n}{\log\log n})$	4	$\widetilde{O}(n^{3-\alpha})$	$\widetilde{O}(f^2 n^{2-(1-\alpha)/f})$	[47]
$f=o(\frac{\log n}{\log\log n})$	$4+\varepsilon$	$O(fn^{2+o(1)}(\log W)\varepsilon^{-f})$	$\widetilde{O}(f^2 2^f + f^7 \log\log W)$	[23]
$f=o(\log n)$	4	$O(n^{2+\alpha}M)$	$\widetilde{O}(f^4 n^{2-\alpha}M + f^{2+\omega}nM)$	[18]
$f=o(\log n)$	4	$O(fn^4)$	$f^{O(f)}$	[34]
$f=o(\log n)$	$1+(24k-6)(f+1)$	$O(n^{3/2+o(1)}+fkn^{1+1/k}\log(nW))$	$\widetilde{O}(f^2 2^f)$	[24]

Table 4 Properties of the f-FDO-sT for undirected graphs obtained via Theorem 4 using single-source f-DSOs from the literature.

Sensitivity	Stretch	Space	Query time	Preprocessing Time	Ref.
1	3	$\widetilde{O}(n^{3/2})$	$\widetilde{O}(1)$	$\widetilde{O}(mn^{1/2}+n^2)$	[16, 38]
1	3	$\widetilde{O}(n^{3/2}M^{1/2})$	$\widetilde{O}(1)$	$\widetilde{O}(n^\omega M)$	[16]
1	$3+\varepsilon$	$\widetilde{O}(n(\log W)\varepsilon^{-1})$	$O(\log\log_{1+\varepsilon}(nW))$	$\mathsf{poly}(n)$	[5, 8, 13]
1	5	$O(n)$	$O(1)$	$\widetilde{O}(mn)$	[13]
$f \geqslant 1$	$4f+3$	$\widetilde{O}(fn)$	$\widetilde{O}(f^3)$	$\widetilde{O}(fm)$	[14]

▶ **Theorem 5.** *Let $G=(V,E)$ be an undirected graph with n vertices, m edges, and possibly positive edge weights. Let S,T be two non-empty subsets of V. Given a single-source f-DSO for G with preprocessing time $\mathtt{P}$, space $\mathtt{S}$, query time $\mathtt{Q}$, and stretch σ, one can compute an f-FDO-ST for G with preprocessing time $O(\mathtt{P}+m+n\log n)$, space $O(\mathtt{S}+n)$, query time $O(f^2+f\mathtt{Q})$, and stretch $2+5\sigma$. For unweighted graphs, the preprocessing time can be improved to $O(\mathtt{P}+m)$*

Table 5 corresponds to the oracles obtained via Theorem 5.

We also prove an information-theoretic lower bound on the space requirement of approximate f-FDOs that support $f \geqslant 2$ edge failures. Note that the lower bound in Theorem 6 holds independently of the query time. It is known from work of Bilò, Cohen, Friedrich, and Schirneck [17] that f-FDOs with stretch $\sigma < 1.5$ require $\Omega(n^2)$ bits of space, and in our work we complement this result by proving that f-FDOs with stretch $\sigma < 5/3$ require $\Omega(n^{1.5})$ bits of space. Obtaining $\Omega(n^2)$ lower bound for f-FDOs with stretch $\sigma < 2$ for undirected unweighted graphs is an interesting open problem.

▶ **Theorem 6.** *Let n be a positive integer. Any f-FDO or f-FDO-ST for n-vertex graphs with sensitivity $f \geqslant 2$ and stretch $\frac{5}{3}-\varepsilon$ for any $\varepsilon > 0$ requires $\Omega(n^{3/2})$ bits of space.*

Table 5 Properties of the fault-tolerant ST-diameter oracles (f-FDO-ST) obtained via the reduction in Theorem 5 using single-source distance sensitivity oracles (f-DSOs) from the literature. W denotes the maximum edge weight for graphs with arbitrary positive weights, M is the maximum edge weight for integer weighted graphs. The parameter $\varepsilon > 0$ is a positive real and $\omega < 2.37286$ denotes the matrix multiplication exponent.

Sensitivity	Stretch	Space	Query time	Preprocessing Time	Ref.
1	7	$\widetilde{O}(n^{3/2})$	$\widetilde{O}(1)$	$\widetilde{O}(mn^{1/2} + n^2)$	[16, 38]
1	7	$\widetilde{O}(n^{3/2}M^{1/2})$	$\widetilde{O}(1)$	$\widetilde{O}(n^\omega M)$	[16]
1	$7 + \varepsilon$	$\widetilde{O}(n(\log W)\varepsilon^{-1})$	$O(\log\log_{1+\varepsilon}(nW))$	$\mathsf{poly}(n)$	[5, 8, 13]
1	12	$O(n)$	$O(1)$	$\widetilde{O}(mn)$	[13]
$f \geqslant 1$	$10f + 7$	$\widetilde{O}(fn)$	$\widetilde{O}(f^3)$	$\widetilde{O}(fm)$	[14]

Outline. This work is structured as follows. In the remainder of this section, we review the literature focusing on diameter oracles and distance sensitivity oracles. We then fix our notations and some preliminaries in Section 2. Section 3 presents our constructions of f-FDO-ST, for the general case of $S, T \subseteq V$. In Section 4 we consider the special case of a single source, that is, of f-FDO-sT. In Section 5 we prove the space lower bound. The proofs of Theorems 1 and 2 follow from similar ideas as discussed in Section 3 and are deferred to the full version of the paper.

1.1 Related Work on Fault-Tolerant Diameter Oracles

Fault-tolerant diameter oracles were introduced by Henzinger, Lincoln, Neumann, and Vassilevska Williams [40]. They showed that for a single failure in unweighted directed graphs, one can compute in time $\widetilde{O}(mn + n^{1.5}\sqrt{Dm/\varepsilon})$, where $\varepsilon \in (0, 1]$ and D is the diameter of the graph, a 1-FDO with $1 + \varepsilon$ stretch that has $O(m)$ space, constant query time. Bilò, Cohen, Friedrich, and Schirneck [17] showed that one can improve the preprocessing time to $\widetilde{O}(mn + n^2/\varepsilon)$, which is nearly optimal under certain conditional hardness assumptions for combinatorial algorithms (see [40]). They also showed that fast matrix multiplication reduces the preprocessing time for dense graphs to $\widetilde{O}(n^{2.5794} + n^2/\varepsilon)$.

Bilò, Choudhary, Cohen, Friedrich, and Schirneck [15] addressed the problem of computing 1-FDOs with $o(m)$ space. They showed that for unweighted directed graphs with diameter $D = \omega(n^{5/6})$, there is a 1-FDO with $\widetilde{O}(n)$ space, $1 + \frac{n^{5/6}}{D} = 1 + o(1)$ stretch, and $O(1)$ query time. It has a preprocessing time of $O(mn)$. In the same work it was also shown that for graphs with diameter $D = \omega((n^{4/3} \log n)/(\varepsilon\sqrt{m}))$ and any $\varepsilon > 0$, there is a $(1 + \varepsilon)$-stretch 1-FDO, with preprocessing time $O(mn)$, space $o(m)$, and constant query time.

For *undirected* graphs the space requirement can be reduced. There is a folklore construction that combines the DSO by Bernstein and Karger [11] with the observation that in undirected graphs the eccentricity of an arbitrary vertex is a 2-approximation of the diameter. This results in an 1-FDO with stretch 2 and constant query time that takes only $O(n)$ space, details can be found in [17, 40].

For $f > 1$ edge failures in undirected graphs with non-negative edge weights, Bilò et al. [17] presented an f-FDO with $(f + 2)$ stretch, $O(f^2 \log^2 n)$ query time, $\widetilde{O}(fn)$ space, and $\widetilde{O}(fm)$ preprocessing time. A lower bound in that work showed that f-FDO with finite stretch must have $\Omega(fn)$ space, nearly matching their construction.

Table 6 Existing f-sensitive all-pairs distance oracles for undirected graphs. The parameter $k \geqslant 1$ is a positive integer, $\varepsilon > 0$ a positive real, $\alpha \in [0,1]$ is a real number in the unit interval, and $\omega < 2.37286$ denotes the matrix multiplication exponent.

Sensitivity	Stretch	Space	Query time	Preprocessing Time	Ref.
1	1	$\widetilde{O}(n^2)$	$O(1)$	$\widetilde{O}(mn)$	[10, 11]
1	1	$\widetilde{O}(n^2)$	$O(1)$	$\widetilde{O}(n^{2.5794}M)$	[37]
1	$(2k-1)(1+\varepsilon)$	$\widetilde{O}(k^5 n^{1+1/k} \varepsilon^{-4})$	$O(k)$	$O(kmn^{1+1/k})$	[8]
2	1	$\widetilde{O}(n^2)$	$\widetilde{O}(1)$	$\mathsf{poly}(n)$	[32]
$f = o(\frac{\log n}{\log\log n})$	1	$\widetilde{O}(n^{3-\alpha})$	$\widetilde{O}(n^{2-(1-\alpha)/f})$	$O(n^{\omega+1-\alpha}M)$	[47]
$f = o(\frac{\log n}{\log\log n})$	$1+\varepsilon$	$O(fn^{2+o(1)}(\log W)\varepsilon^{-f})$	$\widetilde{O}(f^5 \log\log W)$	$O(fn^{5+o(1)}(\log W)\varepsilon^{-f})$	[23]
$f \geqslant 1$	1	$O(fn^4)$	$f^{O(f)}$	$n^{O(f)}$	[34]
$f \geqslant 1$	1	$O(n^{2+\alpha}M)$	$\widetilde{O}(f^2 n^{2-\alpha}M + f^\omega nM)$	$\widetilde{O}(n^{\omega+(3-\omega)\alpha}M)$	[18]
$f \geqslant 1$	$(8k-2)(f+1)$	$O\left(fkn^{1+1/k}\log(nW)\right)$	$\widetilde{O}(f^3)$	$\mathsf{poly}(n)$	[24]

We are not aware of any $O(n)$-sized, constant-stretch FDOs for *directed* graphs with arbitrary diameter in the literature prior to this work, not even for sensitivity ($f = 1$). Also, no non-trivial f-FDOs with $o(f)$-stretch were known. To the best of our knowledge, we are the first to study the problem of general f-FDO-STs with $S, T \neq V$.

We now discuss the known information-theoretic lower bounds for FDOs. Bilò, Cohen, Friedrich, Schirneck [17] showed that any FDO with stretch $\sigma < 3/2$ for undirected unweighted graphs requires $\Omega(m)$ bits of space, even for $f = 1$. They also extended the same lower bound of $\Omega(m)$ bits to edge-weighted graphs and $\sigma < 2$. Bilò, Choudhary, Cohen, Friedrich, and Schirneck [15] extended this result to directed graphs. In particular, they showed that for directed unweighted graphs with diameter $D = O(\sqrt{n}/m)$, any FDO with stretch better than $\left(\frac{3}{2} - \frac{1}{D}\right)$ requires $\Omega(m)$ bits of space. They further proved that for directed graphs any f-FDO requires $\Omega(2^{f/2}n)$ bits of space, as long as $2^{f/2} = O(n)$.

1.2 All-Pairs Distance Sensitivity Oracles

The first distance-sensitive oracle was in the context of directed graphs [29]. It maintained exact distances and was capable of handling a single edge failure. The space requirement of this oracle is $O(n^2 \log n)$ and its query time is $O(\log n)$. This was later generalized to handle a single vertex or edge failure in [30]. Demetrescu, Thorup, Chowdhury, and Ramachandran [30] presented an exact 1-sensitive distance oracle of size $O(n^2 \log n)$, $O(1)$ query time and $\widetilde{O}(mn^2)$ preprocessing time. Later, in two consecutive papers, Bernstein and Karger improved the preprocessing time (while keeping the space and query time unchanged), first to $O(n^2\sqrt{m})$ in [10] and then to $\widetilde{O}(mn)$ in [11]. Baswana and Khanna [8] considered approximate 1-DSOs for unweighted graphs. More precisely, they presented a data structure of size $O(k^5 n^{1+1/k} \frac{\log^3 n}{\varepsilon^4})$, $(2k-1)(1+\varepsilon)$ stretch and $O(k)$ query time. Duan and Pettie [32] considered the case of two failures (vertices or edges) with exact distances. The size of their oracle is $O(n^2 \log^3 n)$, the query time is $O(\log n)$ and the construction time is polynomial.

Using fast matrix multiplication, Weimann and Yuster [47] presented, for any parameter $\alpha \in [0,1]$, a DSO that can handle up to $O(\log n / \log\log n)$ edges or vertices failures with $\widetilde{O}(n^{2-(1-\alpha)/f})$ query time and $O(Mn^{\omega+1-\alpha})$ preprocessing time for directed graphs with integer weights in the range $[-M, M]$, where $\omega < 2.373$ is the matrix multiplication exponent. In [35], Grandoni and Vassilevska Williams presented a distance sensitivity oracle with

Table 7 Existing f-sensitive single-source distance oracles for undirected graphs W denotes the maximum edge weight for graphs with arbitrary positive weights, M is the maximum edge weight for integer weighted graphs. The parameter $\varepsilon > 0$ is a positive real and $\omega < 2.37286$ denotes the matrix multiplication exponent.

Sensitivity	Stretch	Space	Query time	Preprocessing Time	Ref.
1	1	$\widetilde{O}(n^{3/2})$	$\widetilde{O}(1)$	$\widetilde{O}(mn^{1/2} + n^2)$	[16, 38]
1	1	$\widetilde{O}(n^{3/2}M^{1/2})$	$\widetilde{O}(1)$	$\widetilde{O}(n^{\omega}M)$	[16]
1	$1+\varepsilon$	$\widetilde{O}(n\,(\log W)\,\varepsilon^{-1})$	$O(\log\log_{1+\varepsilon}(nW))$	$\mathsf{poly}(n)$	[5, 8, 13]
1	2	$O(n)$	$O(1)$	$\widetilde{O}(mn)$	[13]
$f \geqslant 1$	$2f+1$	$\widetilde{O}(fn)$	$\widetilde{O}(f^2)$	$\widetilde{O}(fm)$	[14]

subcubic $\widetilde{O}(Mn^{\omega+1/2} + Mn^{\omega+\alpha(4-\omega)})$ preprocessing time and sublinear $\widetilde{O}(n^{1-\alpha})$ query time. Van den Brand and Saranurak [18] presented a distance-sensitive oracle that can handle $f \geqslant \log n$ updates (where an update is an edge insertion or deletion), with $\widetilde{O}(Mn^{\omega+(3-\omega)\mu})$ preprocessing time, $\widetilde{O}(Mn^{2-\mu}f^2 + Mnf^{\omega})$ update time, and $\widetilde{O}(Mn^{2-\mu}f + Mnf^2)$ query time, where the parameter $\mu \in [0, 1]$ can be chosen. Chechik and Cohen [22] presented a 1-DSO with with subcubic $\widetilde{O}(Mn^{2.873})$ preprocessing time and $\widetilde{O}(1)$ query time. This was improved by Ren [42] and later by Gu and Ren [37], who obtained a 1-DSO with $\widetilde{O}(Mn^{2.5794})$ preprocessing time and constant query time. Recently Duan and Ren [34] presented an exact f-DSO with $O(fn^4)$ space, $f^{O(f)}$ query time, and $n^{O(f)}$ preprocessing time.

In Table 6 we summarize several of the above f-DSOs for undirected graphs.

1.3 Related Work on Single-Source Distance Sensitivity Oracles

First, we discuss undirected graphs. Baswana and Khanna [8] showed that unweighted undirected graphs can be preprocessed in $O(m\sqrt{n/\varepsilon})$ time to compute a $(1+\varepsilon)$-stretch single-source edge/vertex fault-tolerant distance-oracle of size $O(n \log n + n/\varepsilon^3)$ and constant query time. For weighted graphs, they showed the construction of an $O(n \log n)$ size oracle which can report 3-approximate distances on single failure in $O(1)$ time. Bilò, Gualà, Leucci, and Proietti [13] showed that for a single edge failure in weighted graphs we can compute an $O(n)$-size oracle with stretch 2 and constant query time. Also, a construction is provided that has $1+\varepsilon$ stretch, with $O\big(\varepsilon^{-1} n \log(1/\varepsilon)\big)$ size and $O\big(\varepsilon^{-1} \log n \log(1/\varepsilon)\big)$ query time. All the results stated till now are for a single edge or vertex failure only. For multiple failures, Bilò, Gualà, Leucci, and Proietti [14] gave a construction of size $O(fn \log^2 n)$, computable in $\widetilde{O}(mf)$ time that reports $(2f+1)$-stretched distances in $O(f^2 \log^2 n)$ time.

Bilò, Cohen, Friedrich, and Schirneck [16] presented several additional single-source DSOs. For undirected unweighted graphs, they presented a single-source DSO that has size $O(n^{3/2})$, query time $\widetilde{O}(1)$ and $\widetilde{O}(m\sqrt{n} + n^2)$ preprocessing time. For graphs with integer edge weights in the range $[1, M]$ and using fast matrix multiplication, they presented a single-source DSO with $O(M^{1/2}n^{3/2})$ space, $\widetilde{O}(1)$ query time and $\widetilde{O}(Mn^{\omega})$ preprocessing time. For sparse graphs with $m = O(M^{3/7}n^{7/4})$ they presented a single-source DSO with the same size, $\widetilde{O}(1)$ query time, and subquadratic $\widetilde{O}(M^{7/8}m^{1/2}n^{11/8})$ preprocessing time.

For directed graphs, Baswana, Choudhary, Hussain, and Roditty [5] showed that we can preprocess directed weighted graphs with edge weights in range $[1, W]$ to compute an oracle of $\widetilde{O}(\varepsilon^{-1} n \log W)$ size that reports $(1+\varepsilon)$-approximate distances on single edge/vertex failure in $\widetilde{O}(\log\log_{1+\varepsilon}(nW))$ time. Gupta and Singh [38] designed exact distance oracles of $\widetilde{O}(n^{3/2})$ size that on single edge/vertex failure in directed/undirected unweighted graphs reports distances in $\widetilde{O}(1)$ time. In Table 7 we summarize several of the above f-DSOs for undirected graphs.

2 Preliminaries

For a given graph $G = (V, E)$, possibly with positive edge weights, we denote by $d_G(u, v)$ the distance in G from vertex $u \in V$ to vertex $v \in V$. Given two non-empty subsets $S, T \subseteq V$, the *ST-diameter* of G is defined as $\mathrm{diam}(G, S, T) = \max_{s\in S,t\in T} d_G(s, t)$. With a little abuse of notation, when $S = \{s\}$ (resp., $T = \{t\}$), we also use $\mathrm{diam}(G, s, T)$ (resp., $\mathrm{diam}(G, S, t)$) as a shorthand of $\mathrm{diam}(G, \{s\}, T)$ (resp., $\mathrm{diam}(G, S, \{t\})$) for the sT-diameter (resp., St-diameter). Moreover, if $S = T = V$, we use $\mathrm{diam}(G)$ instead of $\mathrm{diam}(G, V, V)$.

For a given set $F \subseteq E$, we denote by $G-F$ the graph obtained from G by removing all the edges of F. If H is a subgraph of G, we use $V(H)$ and $E(H)$ for the vertices and edges of H, respectively. An *f-edge fault-tolerant distance sensitivity oracle* (f-DSO) with *stretch* $\sigma \geqslant 1$ is a data structure that answers queries (u, v, F) with $u, v \in V$ and $F \subseteq E$ with $|F| \leqslant f$. It returns an estimate $\widehat{d}_{G-F}(u, v)$ of the distance from u to v in $G - F$ such that $d_{G-F}(u, v) \leqslant \widehat{d}_{G-F}(u, v) \leqslant \sigma \cdot d_{G-F}(u, v)$. An *$f$-edge fault-tolerant ST-diameter oracle* (f-FDO-ST) with stretch σ returns, upon query $F \subseteq E$ with $|F| \leqslant f$, an estimate $\widehat{D} = \widehat{D}(F, S, T)$ of the ST-diameter of $G - F$ such that $\mathrm{diam}(G-F, S, T) \leqslant \widehat{D} \leqslant \sigma \cdot \mathrm{diam}(G-F, S, T)$. If $S = \{s\}$ is a singleton or $S = T = V$ are both the whole vertex set, we abbreviate such oracles for as f-FDO-sT and f-FDO, respectively.

3 ST-Diameter Oracles

We start by showing how to use distance sensitivity oracles to design data structures for the fault-tolerant ST-diameter, i.e., the ST-diameter of $G - F$ after a set of edges $F \subseteq E$ failed. The maximum number f of supported failures is called the sensitivity of the data structure. The result is formally stated in Theorem 3.

In the following, we assume that the shortest paths in G are made unique. This way, we can identify a shortest path with its endpoints, which enabled saving both in the time-efficiency of the preprocessing and the space-efficiency of the resulting data structure. In particular, it allows for a subquadratic (in n) space overhead over the underlying f-DSO. However, the precise way how to make the paths unique influences the nature of the preprocessing. As discussed in Section 1, one can ensure a unique shortest path in a random fashion by slightly perturbing the edge weights. Alternatively, lexicographic perturbation [19, 21, 39] provides a deterministic procedure but adds an $O(mn + n^2 \log^2 n)$ term to the running time.

Let $\pi_{u,v}$ denote the (unique) shortest path in G from u to v. Fix a set $F \subseteq E$ of at most f edges and recall that we use $V(F)$ to denote the set of endpoints of edges in F. Our f-DSO-ST uses a data structure to map S and T into two suitable subsets S' and T' of $V(F)$, respectively. A vertex $v \in V(F)$ belongs to S' (resp., T') if there exists a shortest path $\pi_{s,t}$ from some $s \in S$ to some $t \in T$ such that v is a vertex on $\pi_{s,t}$ and the subpath $\pi_{s,v}$ (resp., $\pi_{v,t}$) of $\pi_{s,t}$ from s to v (resp., from t to v) contains no vertex of $V(F)$ other than v. Note that $\pi_{s,v}$ (resp., $\pi_{v,t}$) is completely contained in $G - F$, whence $d_{G-F}(s, v) = d_G(s, v)$ (analogously for $d_{G-F}(v, t)$). The sizes of $S', T' \subseteq V(F)$ are in $O(f)$.

3.1 Query Algorithm

Before describing the data structure, we present the query algorithm. Let $\mathcal{D}$ denote the f-DSO with stretch $\sigma \geqslant 1$ that is assumed in Theorem 3. Given the query F, our diameter oracle computes the two sets S' and T'. Next, for every two vertices u and v such that $u \in S'$ and $v \in T'$, it queries $\mathcal{D}$ with the triple (u, v, F) to obtain a σ-approximation of $d_{G-F}(u, v)$. The f-FDO-ST returns the value $\widehat{D} = \operatorname{diam}(G, S, T) + \max_{(u,v)\in S'\times T'} \mathcal{D}(u, v, F)$.

Given S' and T', the time needed to compute $\widehat{D}$ is $O(f^2\mathtt{Q})$, where $\mathtt{Q}$ is the query time of the f-DSO $\mathcal{D}$. The value $\operatorname{diam}(G, S, T)$ can be precomputed.

▶ **Lemma 7.** *The f-FDO-ST has a stretch of $1 + 3\sigma$.*

Proof. Let $s \in S$ and $t \in T$ be two arbitrary vertices. We first show that $d_{G-F}(s, t) \leqslant \widehat{D}$, that is, the returned value never underestimates the ST-diameter of $G - F$. We only need to prove the case in which some of the failing edges in F belong to $\pi_{s,t}$ as otherwise $d_{G-F}(s, t) = d_G(s, t) \leqslant \operatorname{diam}(G, S, T) \leqslant \widehat{D}$. Thus, let x_s (resp., x_t) be the vertex of $V(F)$ that is closest to s (resp., t) in $\pi_{s,t}$. By definition of S', T', we have $x_s \in S'$ and $x_t \in T'$ and thus $d_{G-F}(s, x_s) = d_G(s, x_s)$ and $d_{G-F}(x_t, t) = d_G(x_t, t)$. Moreover, it holds that $d_{G-F}(s, x_s) + d_{G-F}(x_t, t) = d_G(s, x_s) + d_G(x_t, t) \leqslant \operatorname{diam}(G, S, T)$ as π_{s,x_s} and $\pi_{x_t,t}$ are vertex-disjoint. Using the triangle inequality twice and the fact that $\max_{(u,v)\in S'\times T'} \mathcal{D}(u, v, F)$ never underestimates $\operatorname{diam}(G-F, S', T')$, we get

$$\begin{aligned} d_{G-F}(s,t) &\leqslant d_{G-F}(s, x_s) + d_{G-F}(x_s, x_t) + d_{G-F}(x_t, t) \\ &\leqslant \operatorname{diam}(G, S, T) + \operatorname{diam}(G-F, S', T') \leqslant \widehat{D}. \end{aligned}$$

We now prove that $\widehat{D} \leqslant (1 + 3\sigma) \cdot \operatorname{diam}(G-F, S, T)$. Let $u \in S'$ and $v \in T'$ be arbitrary. There are $s \in S$ and $t \in T$ such that $d_{G-F}(s, u), d_{G-F}(v, t) \leqslant \operatorname{diam}(G, S, T)$. We arrive at

$$\begin{aligned} \mathcal{D}(u, v, F) &\leqslant \sigma\, d_{G-F}(u, v) \leqslant \sigma\big(d_{G-F}(u, s) + d_{G-F}(s, t) + d_{G-F}(t, v)\big) \\ &\leqslant \sigma\big(\operatorname{diam}(G, S, T) + \operatorname{diam}(G-F, S, T) + \operatorname{diam}(G, S, T)\big) \\ &\leqslant 3\sigma \operatorname{diam}(G-F, S, T), \end{aligned}$$

thus $\widehat{D} = \operatorname{diam}(G, S, T) + \max_{u\in S', v\in T'} \mathcal{D}(u, v, F) \leqslant (1 + 3\sigma) \operatorname{diam}(G - F, S, T)$. ◀

3.2 Data Structure for the Sets S' and T' for Large Sensitivity

Recall that, given the failure set F, the set S' contains all $v \in V(F)$ such that there are $s \in T$ and $t \in T$ for which v is the closest vertex to s on $V(F) \cap E(\pi_{s,t})$, analogously for T'. We now describe the data structure that computes the sets S' and T', focusing on S' since the case of T' follows in the same fashion.

The construction algorithm depends on the sensitivity f. Suppose first that $f = \Omega(\log n)$. For each vertex $v \in V$, the data structure stores the shortest-path tree T_v of G rooted at v and mark some of its vertices. Namely, all $s \in S$ are marked for which there is a $t \in T$ such that v lies on the path $\pi_{s,t}$. For every two vertices $s \in S$ and $t \in T$, $\pi_{s,t}$ contains v if and only if $d_G(s, t) = d_G(s, v) + d_G(v, t)$. We used here that the paths are unique. It suffices to compute the all-pairs distances in G in time $O(mn + n^2 \log n)$ time[2] and use them to mark the vertices of T_v for *all* v with the obvious $O(n|S||T|)$-time algorithm.

[2] The time needed for this step reduces to $O(mn)$ in case G is unweighted or has only small integer or even floating point weights (in exponent-mantissa representation) using Thorup's algorithm [46].

Additionally, each vertex u of T_v is annotated with the value $count_v(u)$, the number of marked vertices in the subtree $(T_v)_u$ rooted at u. For a fixed tree T_v, all values $count_v(u)$ are computable in $O(n)$ time in a bottom-up fashion. Finally, we store, for each T_v, a data structure that supports least common ancestor (LCA) queries in constant time. Such structures can be built in time and space that is linear in the size of the tree [9]. The time needed to construct the data structure is $O(mn + n^2 \log n + n|S||T|)$ and the space is $O(n^2)$.

To answer a query F, the algorithm scans all the vertices $v \in V(F)$ and decides which of them to include in S'. The graph $T_v - F$ is a collection of rooted trees. (Possibly some of the trees degenerated to isolated vertices.) We observe that $v \in S'$ if and only if $T_v - F$ contains a marked vertex that is still reachable from v. To check this condition, the algorithm computes the set F_0 of all the edges $\{u, w\} \in F$ that are contained in T_v. This is the case if and only if the LCA of u and w in T_v is either u or w.

Next, we define a notion of domination for edges in F_0. We say that an edge $\{u, w\} \in F_0$, where u is the parent of w in T_v, is *dominated* by another edge $\{a, b\} \in F_0$, where a is the parent of b in T_v, if $\{u, w\}$ is in the subtree of T_v rooted at b. This is equivalent to b being the LCA of b and u. The query algorithm removes all dominated edges from F_0, which can be done in $O(|F_0|^2) = O(f^2)$ time.

Recall that $count_v(v)$ is the overall number of marked vertices in T_v. Evidently, some vertex in $T_v - F$ is reachable from v iff they are in the same connected component. Thus, there is a marked vertex reachable from v if and only if $count_v(v)$ is strictly larger than the number of marked vertices contained in those components of $T_v - F$ that do *not* contain v. Indeed, the difference between those two values is exactly the *number* of marked vertices reachable from v. Each connected component of $T_v - F$ that does not contain v is a tree T' rooted at some vertex $w \in V(F_0)\backslash\{v\}$. Let u be the parent of w in T_v. Compared to the full subtree $(T_v)_u$ rooted at u, T' is missing those subtrees "further down" that are rooted at some other vertex b whose parent a is a vertex of T'. Those are exactly the edges $\{a, b\} \in F_0$ that are dominated by $\{u, w\}$. Accordingly, the value $count_v(u)$ counts the marked vertices in T' and additionally those in the subtrees rooted at the vertices b. By removing all dominated edges from F_0, we avoid any double counting and ensure that $count_v(v) - \sum_{u \in V(F_0)} count_v(u)$ is indeed the quantity we are interested in. It can be computed in time $O(f)$ for each v.

3.3 Small Sensitivity

We now modify the data structure in the case where the sensitivity $f = o(\log n)$ is sublogarithmic. If so, the information of all the trees T_v can be stored in a more compact way. For every vertex $v \in V$, we define a new representation $\mathcal{T}_v$ of the tree T_v by first removing unnecessary parts and then replacing long paths with single edges. This corresponds to the two steps of the compression described below. For the first one, we need the following definition. We say a subtree $\mathcal{T}_v$ of T_v preserves the *source-to-leaf reachability* if, for every set $F \subseteq E$ of up to f failing edges, there is a marked vertex of T_v that is reachable from the source v in $T_v - F$ if and only if there is a leaf of $\mathcal{T}_v$ that is reachable from v in $\mathcal{T}_v - F$.

The first compression step. We first describe how to preserve the source-to-leaf reachability. We select a set $\mathcal{L}_v \subseteq S$ of at most 2^f marked vertices and set $\mathcal{T}_v$ as the smallest subtree of T_v that contains v and $\mathcal{L}_v$. We say that a marked vertex s of T_v is *relevant* if there is no marked vertex $s' \neq s$ that is contained in the path from v to s in T_v.

We compute $\mathcal{L}_v$ as follows. We construct a DAG G_v that is obtained from a copy of T_v in which each edge (u, u'), with u being the parent of u' in T_v, is directed from u to u'. The DAG is augmented with a dummy sink vertex x that contains an incoming directed edge from each relevant vertex s of T_v. We then run the algorithm of Baswana, Choudhary, and Roditty [6] to compute a subgraph H_v of G_v such that (i) the in-degree of each vertex of H_v is at most 2^f and (ii) for every possible set F of at most f edge failures, each vertex u is reachable from v in the graph $G_v - F$ iff u is reachable from v in $H_v - F$.

The set $\mathcal{L}_v$ of marked vertices corresponds to the tails of the edges in H_v that enter the sink x. As x has in-degree of at most 2^f in H_v, the size of $\mathcal{L}_v$ is $O(2^f)$. Moreover, $\mathcal{L}_v$ is the set of leaves of $\mathcal{T}_v$. The following lemma proves the correctness of our selection algorithm.

▶ **Lemma 8.** *For every $F \subseteq E(G)$, with $|F| \leqslant f$, there is a marked vertex of T_v that is reachable from v in $T_v - F$ iff there is a vertex of $\mathcal{L}_v$ that is reachable from v in $\mathcal{T}_v - F$.*

Proof. Fix a set F of at most f failing edges of G. As $\mathcal{T}_v$ is a subtree of T_v, if there is a vertex in $\mathcal{L}_v$ that is reachable from v in $\mathcal{T}_v - F$, then the same marked vertex is reachable from v in $T_v - F$. To prove the other direction, let X be the set of all marked vertices that are reachable from v in $T_v - F$. We prove that $X \cap \mathcal{L}_v \neq \emptyset$. Let $s \in X$ be a marked vertex that is reachable from v in $T_v - F$. Let $s^* \in S$ be the vertex closest to v in the path from v to s in T_v (possibly, $s^* = s$). We have that s^* is relevant and is reachable from v in $T_v - F$. This implies that the sink x is reachable from v in $G_v - F$ via the path that goes through s^*. As a consequence, x is also reachable in $H_v - F$. Hence, there is a vertex in $\mathcal{L}_v$ that is also reachable from v in $\mathcal{T}_v - F$. Therefore, $X \cap \mathcal{L}_v \neq \emptyset$. ◀

The second compression step. After the first compression step, the tree $\mathcal{T}_v$ contains at most 2^f leaves. However, it might still be the case that the number of vertices of $\mathcal{T}_v$ is large due to the presence of very long paths connecting two consecutive *branch vertices*, i.e., vertices of $\mathcal{T}_v$ with two or more children. The second step of compressing $\mathcal{T}_v$ allows us to represent long paths between consecutive branch vertices in a more compact way.

Let x and y be two consecutive branch vertices in $\mathcal{T}_v$, i.e., x is an ancestor of y in $\mathcal{T}_v$ and the internal vertices of the path P from x to y are not branch vertices. We say that P is *long* if it contains at least $\sqrt{n}$ edges. If the path P is long, we substitute the path P in $\mathcal{T}_v$ with a *representative* edge between x and y (so we also remove all the internal vertices of P from the tree) and we add the path P to the set $\mathcal{P}$ of long paths. So, in every tree $\mathcal{T}_v$, we replace every long path between two consecutive branch vertices with a representative edge. We observe that $\mathcal{P}$ can be computed in $O(n^2)$ time. Moreover, we observe that $\mathcal{P}$ contains $O(n^{3/2})$ paths as each tree T_v contributes with at most $\sqrt{n}$ long paths.

Next, we use the algorithm given in [2] to hit all the long paths in $\mathcal{P}$ with a set Z of $O(\sqrt{n}\log n)$ *pivot* vertices in $O(|\mathcal{P}|\sqrt{n}) = O(n^2)$ time, where a path is *hit* if we select a pivot vertex that belongs to the path. For each pivot $z \in Z$, we store the shortest-path tree T_z of G rooted at z. By construction, each long path $P \in \mathcal{P}$ between two consecutive branch vertices x and y of a tree $\mathcal{T}_v$ is contained in T_z, for some $z \in Z$ that hits P; moreover, a vertex $z \in Z$ that hits P is also the least-common-ancestor of x and y in T_z.

The representative edge (x, y) in $\mathcal{T}_v$ stores a pointer to the tree T_z of any pivot z that hits P (ties can be arbitrarily broken). Clearly, after the second compression step, each tree $\mathcal{T}_v$ contains $O(2^f\sqrt{n})$ vertices. Therefore, the overall size needed to store all the trees $\mathcal{T}_v$ is $O(2^f n^{3/2})$. Moreover, storing the trees T_z for all the pivots in Z requires $O(n)$ space per tree, for a total of $O(n^{3/2}\log n)$ space. Hence, the overall size of our data structure is $O(n^{3/2}(2^f + \log n))$.

Now, given a set F of at most f failing edges, we describe how the query algorithm computes the set S' in $O(f^2 2^f)$ time. As before, for every $v \in V(F)$, we need to understand whether v must be added to S' or not. In the following, we fix $v \in V(F)$ and explain how to check whether $v \in S'$ or not in $O(f2^f)$ time. We recall that v must be added to S' iff there is a marked vertex in $T_v - F$ that is still reachable from v. By Lemma 8, this is equivalent to having a leaf of $\mathcal{L}_v$ that is reachable from v in $\mathcal{T}_v - F$.

We visit the tree $\mathcal{T}_v$ and we remove from $\mathcal{T}_v$ all edges that correspond to edges in F. This can be easily done in $O(f)$ time for each non-representative edge using least-common-ancestor queries. For the representative edges we proceed as follows. We consider all the representative edges in $\mathcal{T}_v$. Let (x, y) be a representative edge of $\mathcal{T}_v$ and let z be the pivot of the tree T_z that is associated with the edge (x, y) in $\mathcal{T}_v$. We remove (x, y) from $\mathcal{T}_v$ iff there is a failing edge in F that is contained in the path P in T_z from x to y. We check whether P contains some edges of F in $O(f)$ time as follows. We look at all the failing edges in F and, for each failing edge $(u, u') \in F$, we check whether (u, u') is an edge of P using a constant number of *least-common-ancestor* queries in the tree T_z.[3] As each tree $\mathcal{T}_v$ contains $O(2^f)$ representative edges and we need $O(f)$ time to understand if a representative edge can be removed or not from the tree, we need $O(f2^f)$ to understand which are the representative edges that need to be removed from $\mathcal{T}_v$, for a fixed $v \in V(F)$.

Once all edges that represent F have been removed from $\mathcal{T}_v$, it is enough to check whether there is a vertex of $\mathcal{L}_v$ that is still reachable from v. This can be clearly done in $O(f^2)$ time per tree $\mathcal{T}_v$ using the values k_u, as already discussed for the case in which $f = \Omega(\log n)$. In particular, for every vertex u in $\mathcal{T}_v$, the value k_u is equal to the number of vertices of $\mathcal{L}_v$ that are contained in the subtree of $\mathcal{T}_v$ rooted at u.

4 Single-Source sT-Diameter Oracles

In the following theorem, we address the question of computing an sT-diameter oracle using a single-source DSO with source s. We restate the relevant theorem below. Its proof uses similar ideas as those shown in Section 3, but the single-source setting allows for a better preprocessing time, space, and stretch.

▶ **Theorem 4.** *Let $G = (V, E)$ be an undirected graph with n vertices, m edges, and possibly positive edge weights. Let $s \in V$ be a vertex and $T \subseteq V$ a non-empty set. Given a single-source f-DSO for G with preprocessing time $\mathtt{P}$, space $\mathtt{S}$, query time $\mathtt{Q}$, and stretch σ, one can compute an f-FDO-sT for G with preprocessing time $\mathtt{P} + O(m + n \log n)$, space $\mathtt{S} + O(n)$, query time $O(f^2 + f\mathtt{Q})$, and stretch $1 + 2\sigma$. For unweighted graphs, the preprocessing time can be improved to $\mathtt{P} + O(m)$.*

Proof. Let $\mathcal{D}$ denote the single-source f-DSO. The preprocessing algorithm for the f-FDO-sT first constructs $\mathcal{D}$ with source s. It also computes a shortest path tree T_s of G rooted at s. Each node $v \in V(T_s) = V$ is annotated with a pointer to its parent node and its respective number in the pre-order and post-order traversal of T_s. Similarly as above, the algorithm also computes the value $count(v)$ for every v, which is the number of descendants of v (including v itself) that are in T. Finally, it stores the maximum distance $C = \max_{t \in T} d_G(s, t)$ from the root among the vertices in the set T. The preprocessing takes total time $\mathtt{P} + O(m + n \log n)$ in general weighted graphs and, again, can be reduced to $\mathtt{P} + O(m)$ for certain classes of weights [46]. Storing the oracle and the tree takes $\mathtt{S} + O(n)$ space.

[3] We observe that (u, u') is on the path P iff one of the following two conditions hold: (i) the least-common-ancestor of u and x in T_z is u and the least-common-ancestor of u' and x in T_z is u'; (ii) the least-common-ancestor of u and y in T_z is u and the least-common-ancestor of u' and y in T_z is u'.

For the query, consider a set $F \subseteq E$ of up to f failing edges and let $F_0 = F \cap E(T_s)$ be those failures that are in the tree. Consider the collection of rooted (sub-)trees $T_s - F_0$. Define X_F to be the set of roots of those trees that contain some vertex from T. For some $v \in V$, let $\mathcal{D}(v, F)$ be the σ-approximation of the replacement distance $d_{G-F}(s, v)$ computed by the DSO $\mathcal{D}$. Our sT-diameter oracle answers the query F by reporting the value

$$\widehat{D} = C + \max_{x \in X_F} \mathcal{D}(x, F).$$

Regarding the correctness of that answer, consider a vertex $t \in T$. Let $x \in X_F$ be the root of the subtree of T_s that contains t. There is a path from s to t in $G - F$ of length at most $d_{G-F}(s,x) + d_G(x,t) \leqslant d_{G-F}(s,x) + d_G(s,t) \leqslant \mathcal{D}(x,F) + C$. Hence, we have $d_{G-F}(s,t) \leqslant C + \max_{x \in X_F} \mathcal{D}(x,F)$, that is, $\mathrm{diam}(G{-}F, s, T) \leqslant \widetilde{D}$. We next prove $\widehat{D} \leqslant (1+2\sigma) \cdot \mathrm{diam}(G{-}F, s, T)$. Let $x_0 \in X_F$ be the maximizer of $\mathcal{D}(x,F)$, and $t \in T$ be in the tree in $T_s - F_0$ that is rooted in x_0. Then, we have $d_{G-F}(s,x_0) \leqslant d_{G-F}(s,t) + d_G(t,x_0) \leqslant d_{G-F}(s,t) + d_G(t,s) \leqslant 2 \cdot d_{G-F}(s,t)$. We used here that G is undirected so that we can go "up" the tree from t to x_0. From this, we get

$$\widehat{D} = C + \mathcal{D}(x_0, F) \leqslant C + \sigma \cdot d_{G-F}(s, x_0) \leqslant C + 2\sigma \cdot d_{G-F}(s,t) \leqslant (1+2\sigma) \cdot \mathrm{diam}(G{-}F, s, T).$$

Given X_F, computing $\widehat{D}$ takes time $O(f\mathtt{Q})$. It remains to show how to compute X_F from F in $O(f^2)$ time. Recall that we know the parent of every non-root node in T_s. We use it to first obtain F_0 from F in time $O(f)$ as an edge $\{a, b\}$ is in T_s iff a is parent of b or vice versa.

For each edge $e \in F_0$, let $b(e)$ be the endpoint of e that is farther from the source s. Next, define $B_0 = \{b(e) \,|\, e \in F_0\} \cup \{s\}$. Every root in X_F is either the source s or the "lower" endpoint of a failing edge, i.e., $X_F \subseteq B_0$. For each $b \in B_0$, let $B_0(b)$ be the closest proper descendants of b in B_0, if any. That is, on the paths in T_s between b and any $b' \in B_0(b)$ there is no other vertex from B_0. We can compute the sets $B_0(b)$ for all $b \in B_0$ simultaneously in total time $O(|B_0|^2) = O(f^2)$ as follows. A vertex is a proper ancestor of b' iff its pre-order number is strictly smaller than that of b' and its post-order number is strictly larger. So finding those takes time $O(|B_0|)$ for each $b' \in B_0$. Then, b' is in the set $B_0(b)$ for the proper ancestor b with the highest pre-order number.

Finally, observe that a vertex $b \in B_0$ lies in X_F if and only if there is at least one vertex of T that falls into the subtree of T_s rooted at v but not in any of the subtrees rooted at (proper) descendants of b in B_0. To check this condition via the counts, we only need to consider the immediate descendants in $B_0(b)$. If the element of T is in some lower subtree, then it is also accounted for by an immediate descendant. In summary, some $b \in B_0$ is in X_F iff $\mathit{count}(b) - \sum_{b' \in B_0(b)} \mathit{count}(b') > 0$. This proves that X_F is computable in time $O(f^2)$. ◀

We now handle multiple sources, that is, we build an f-FDO-ST for a general set S. The next result is a straightforward reduction to the sT-case. As it turns out, it is enough to construct the sT-diameter oracle for two arbitrary vertices $s \in S$ and $t \in T$. Due to lack of space, the proof of Lemma 9 is deferred to the full version of the paper.

▶ **Lemma 9.** *Let $G = (V, E)$ be an undirected graph with n vertices, m edges, and possibly positive edge weights. Let $S, T \subseteq V$ be non-empty sets of vertices, and $s \in S$ and $t \in T$ be two vertices. Suppose one is given access to an f-FDO-sT and an f-FDO-tS for G with respective preprocessing times $\mathtt{P}_{sT}$ and $\mathtt{P}_{tS}$, space requirements $\mathtt{S}_{sT}$ and $\mathtt{S}_{tT}$, query times $\mathtt{Q}_{sT}$ and $\mathtt{Q}_{sT}$, and stretches σ_{sT} and σ_{tS}. Then, one can compute an f-FDO-ST for G with preprocessing time $\mathtt{P}_{sT} + \mathtt{P}_{tS}$, space $\mathtt{S}_{sT} + \mathtt{S}_{tS}$, query time $\mathtt{Q}_{sT} + \mathtt{Q}_{tT}$, and stretch $\sigma_{sT} + \sigma_{tS} + \min(\sigma_{sT}, \sigma_{tS})$.*

Table 8 Conditions for the presence of edges between the vertex sets of graph G in Section 5. The symbol $\oplus$ stands for the exclusive or. All conditions are symmetric with respect to the index pairs (i,x), (j,y), and (k,z), whence H is undirected.

Set Pair	Vertex Pair	Edge Condition
$A \times A$		independent set
$B \times B$	$b[i,j,k],\ b[x,y,z]$	$(i=x) \oplus (j=y)$
$C \times C$	$c[i,j,k],\ c[x,y,z]$	$(i=x) \oplus (j=y)$
$D \times D$		independent set
$A \times B$	$a[i,j],\ b[x,y,z]$	$(i=x) \wedge (z=0)$
$B \times C$	$b[i,j,k],\ c[x,y,z]$	$(i=x) \wedge (j=y)$
$C \times D$	$c[i,j,k],\ d[x,y]$	$(j=y) \wedge (k=0)$

Combining Theorem 4 and Lemma 9 gives a reduction from f-FDO-ST to single-source f-DSOs. However, it results in a data structure with a stretch of $3+6\sigma$, where σ is the original stretch of the f-DSO We can improve this by not treating Lemma 9 as a black box.

▶ **Theorem 5.** *Let $G=(V,E)$ be an undirected graph with n vertices, m edges, and possibly positive edge weights. Let S,T be two non-empty subsets of V. Given a single-source f-DSO for G with preprocessing time $\mathtt{P}$, space $\mathtt{S}$, query time $\mathtt{Q}$, and stretch σ, one can compute an f-FDO-ST for G with preprocessing time $O(\mathtt{P}+m+n\log n)$, space $O(\mathtt{S}+n)$, query time $O(f^2+f\mathtt{Q})$, and stretch $2+5\sigma$. For unweighted graphs, the preprocessing time can be improved to $O(\mathtt{P}+m)$*

Proof. Let $s \in S$ and $t \in T$ be arbitrary. The preprocessing algorithm of the f-FDO-ST uses the single-source f-DSO twice, once for source s and once for t, to construct an f-FDO-sT $\mathcal{D}_{sT}$ and an f-FDO-tS $\mathcal{D}_{tS}$ both with stretch $1+2\sigma$, as described in Theorem 4.

For a set $F \subseteq E$ of at most f edge failures, let $\mathcal{D}_{sT}(F)$ and $\mathcal{D}_{tS}(F)$ be the respective $(1+2\sigma)$-approximations of $\operatorname{diam}(G-F,s,T)$ and $\operatorname{diam}(G-F,t,S)$. Further, let $\mathcal{D}_{st}(F)$ be a σ-approximation of $d_{G-F}(s,t)$, obtained from the DSO with source s. The query algorithm outputs $\widehat{D} = \mathcal{D}_{tS}(F) + \mathcal{D}_{st}(F) + \mathcal{D}_{sT}(F)$. Let $(s_0,t_0) \in S \times T$. We have

$$\begin{aligned} d_{G-F}(s_0,t_0) &\leqslant d_{G-F}(s_0,t) + d_{G-F}(t,s) + d_{G-F}(s,t_0) \\ &\leqslant \mathcal{D}_{tS}(F) + \mathcal{D}_{st}(F) + \mathcal{D}_{sT}(F) \leqslant (2+5\sigma) \cdot \operatorname{diam}(G-F,S,T). \end{aligned}$$ ◀

5 Space Lower Bound

Recall that Theorem 6 states a space lower bound for f-FDOs and f-FDO-STs with sensitivity $f \geqslant 2$ in that if they have stretch better than $5/3$, they must take $\Omega(n^{3/2})$ space. The theorem is implied by the following lemma, which we prove in this section.

▶ **Lemma 10.** *For infinitely many n, there is a graph $G=(V,E)$ with n vertices (and two sets $S,T \subseteq V$) such that any data structure that decides for any pair of edges $e,e' \in E$, whether $G-\{e,e'\}$ has diameter (resp., ST-diameter) 3 or 5 requires $\Omega(n^{3/2})$ bits of space.*

We first construct an auxiliary graph H. Let $n=6N$ for some N which is a perfect square. In the following, indices i,j range over the set $[\sqrt{N}]$ and k ranges over $\{0,1\}$ Define four pairwise disjoint sets of vertices $A=\{a[i,j]\}_{i,j}$, $B=\{b[i,j,k]\}_{i,j,k}$, $C=\{c[i,j,k]\}_{i,j,k}$, $D=\{d[i,j]\}_{i,j}$ with respective cardinalities $N, 2N, 2N$, and N. The vertex set of H is $V(H)=A\cup B\cup C\cup D$. The edges in H are shown in Table 8 and are defined

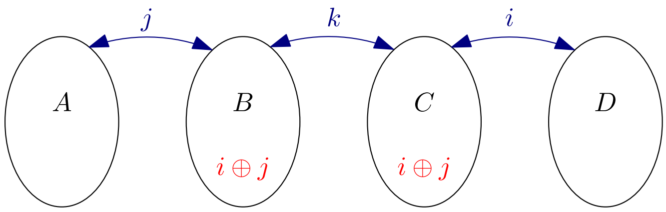

Figure 1 Visual representation of the graph H. Each vertex corresponds to a tuple $[i, j]$ or $[i, j, k]$ and belongs to one of the sets A, B, C, or D. To move from vertex $a[i, j]$ to $b[i', j', k']$ it must be the case that $i = i'$ and $k' = 0$, but one can jump from any j to any j'. This is marked by the blue edge labeled j between sets A and B, analogously for the other pairs of sets. When moving inside the sets B or C *either* the first index $i \neq i'$ *or* the second one $j \neq j'$ changes, marked by the red labels. Sets A and D have no internal edges.

depending on the relations among the indices of the participating vertices. For example, some edge $\{b[i, j, k],\ b[x, y, z]\}$ between elements of B and C exists if and only if *either* i and x are equal *or* j and y are equal, while $k, z \in \{0, 1\}$ can be arbitrary. Note that the number of edges in E is $\Theta(N^{3/2}) = \Theta(n^{3/2})$.

▶ **Lemma 11.** *The diameter of H is at most* 3.

Proof. To verify that the diameter of H is at most 3, we give explicit paths of length at most 3 between all possible vertex pairs from the sets A, B, C, and D. Note that all paths below are reversible as the edges are undirected. The symbol $\overline{x}$ stands for any index from $[\sqrt{N}]$ except x, analogously for $\overline{y}$.

- For vertices $a[i, j], a[x, y] \in A$, we distinguish two cases depending on whether the first indices $i \neq x$ are different or not. In the first case, the vertices are joined by the path $(a[i, j], b[i, y, 0], b[x, y, 0], a[x, y])$. In the second case, the middle two vertices are the same, thus the path shortens to $(a[i, j], b[x, y, 0], a[x, y])$.
- Symmetrically, for vertices $d[i, j], d[x, y] \in D$, the cases are defined with respect two the second indices, i.e., whether $j \neq y$. The paths are $(d[i, j], c[x, j, 0], c[x, y, 0], d[x, y])$ and $(d[i, j], c[x, y, 0], d[x, y])$, respectively.
- For vertices $b[i, j, k], b[x, y, z] \in B$, the generic path is $(b[i, j, k], b[x, j, k], b[x, y, z])$. If $i = x$, then the first two vertices are the same; if $j = y$, the last two are. The argument for vertices $c[i, j, k], c[x, y, z] \in C$ is the same.
- For the vertex pair $(a[i, j], b[x, y, z]) \in A \times B$, the ley point is that any edge inside of B changes exactly one of the first two indices. If $i \neq x$, the path is $(a[i, j], b[i, y, 0], b[x, y, z])$, otherwise it is $(a[i, j], b[x, \overline{y}, 0], b[x, y, z])$.
- The pair $(d[i, j], c[x, y, z]) \in D \times C$ is handled symmetrically. If $j \neq y$, the path is $(d[i, j], c[x, j, 0], c[x, y, z])$, otherwise it is $(d[i, j], b[\overline{x}, y, 0], b[x, y, z])$.
- Vertex pair $(a[i, j], c[x, y, z]) \in A \times C$: path $(a[i, j], b[i, y, 0], c[i, y, z], c[x, y, z])$. Note that if $i = x$ the last two vertices are the same. Vertex pair $(d[i, j], b[x, y, z]) \in D \times B$: path $(d[i, j], c[x, j, 0], b[x, j, z], b[x, y, z])$.
- Vertex pair $(a[i, j], d[x, y]) \in A \times D$: path $(a[i, j], b[i, y, 0], c[i, y, 0], d[x, y])$.
- Vertex pair $(b[i, j, k], c[x, y, z]) \in B \times C$: the path $(b[i, j, k], b[x, j, k], c[x, j, k], c[x, y, z])$ possibly shortens if consecutive vertices are the same. ◀

Consider an arbitrary binary $\sqrt{N} \times \sqrt{N} \times \sqrt{N}$ matrix (tensor) M. We build a supergraph $G \supseteq H$ embedding the information about the entries of M in the fault-tolerant diameter of G under dual failures, i.e., $\mathrm{diam}(G - F)$ with $|F| = 2$. The number of possible matrices M will then imply the space lower bounds for diameter oracles for G.

The graph G contains all vertices and edges of H and the following additional edges.

- For all $i, j, y \in [\sqrt{N}]$, if $M[i, j, y] = 1$, then add $\{a[i, j], b[i, y, 1]\}$ as an edge of G.
- For all $i, x, y \in [\sqrt{N}]$, if $M[i, x, y] = 1$, then add $\{c[i, y, 1], d[x, y]\}$.

Note that the diameter of G remains at most 3.

Consider any four indices $i, j, x, y \in [\sqrt{N}]$ such that $i \neq x$ and $j \neq y$. We define two sets F, F' both containing pairs of vertices in $V = V(H)$. First, let $F \subseteq E(H) \subseteq E$ contain $e_1 = \{a[i, j], b[i, y, 0]\}$ and $e_2 = \{c[i, y, 0], d[x, y]\}$. Secondly, let F' be the set comprising the two pairs $e'_1 = \{a[i, j], b[i, y, 1]\}$ and $e'_2 = \{c[i, y, 1], d[x, y]\}$. Note that the elements of F' are only edges of G if the entries $M[i, j, y]$ and $M[i, x, y]$ are 1.

▶ **Lemma 12.** *For any four indices $i, j, x, y \in [\sqrt{N}]$ such that $i \neq x$ and $j \neq y$, the diameter of $G - (F \cup F')$ is at least 5.*

Proof. We show that the distance between $a[i, j]$ and $d[x, y]$ in $G - (F \cup F')$ is at least 5. Contrarily, assume that $P = (a[i, j], w_1, w_2, w_3, d[x, y])$ is a path of length at most 4. P must pass across sets $A \to B$, $B \to C$, and $C \to D$ and change the indices from (i, j) to (x, y).

The neighborhood of $a[i, j]$ in $G - (F \cup F')$ is the set

$$\Big\{ b[i, \overline{y}, 0] \mid \overline{y} \in [\sqrt{N}] \setminus \{y\} \Big\} \cup \Big\{ b[i, \overline{y}, 1] \mid \overline{y} \in [\sqrt{N}] \setminus \{y\} \wedge M[i, j, \overline{y}]\} = 1 \Big\}.$$

The index i cannot change on the first edge $\{a[i, j], w_1\}$ of P and, since the edges $e_1 = \{a[i, j], b[i, y, 0]\} \in F$ and $e'_1 = \{a[i, j], b[i, y, 1]\} \in F'$ are missing, the second index of w_1 must differ from y. Symmetrically, the change of j cannot take place on the last edge $\{w_3, d[x, y]\}$ and the first index of w_3 must differ from x. At least one of the edges $\{w_1, w_2\}$ or $\{w_2, w_3\}$ passes from B to C, w.l.o.g. let this be $\{w_1, w_2\}$. This edge (already present in H) cannot change any of the indices. We are left with $\{w_2, w_3\}$. If P has strictly less than 4 edges, then $w_2 = w_3$. Otherwise, either both endpoints w_2 and w_3 are in B, both are in C or there is exactly one in either. None of those cases allows one to make the *two* necessary changes to the indices simultaneously. ◀

▶ **Lemma 13.** *The diameter of $(G - F) \cup F'$ is 3.*

Proof. The proof is very similar to that of Lemma 11, only that every time the edge $e_1 = \{a[i, j], b[i, y, 0]\} \in F$ (respectively, $e_2 = \{c[i, y, 0], d[x, y]\}$) has been used, it is replaced by $e'_1 = \{a[i, j], b[i, y, 1]\} \in F'$ (respectively, by $e'_2 = \{c[i, y, 1], d[x, y]\}$). ◀

▶ **Lemma 14.** *The diameter of $G - F$ is at most 3 if $M[i, j, y] = M[i, x, y] = 1$, and at least 5 if $M[i, j, y] = M[i, x, y] = 0$.*

Proof. The diameter of graph $G - F$ is at least 5 if neither vertex pair in F' is an edge of G by Lemma 12. This is only true if $M[i, j, y] = M[i, x, y] = 0$. Conversely, by Lemma 13, the diameter is at most 3 if both edges in F' lie in G, i.e., if $M[i, j, y] = M[i, x, y] = 1$. ◀

We now finish the proof of Lemma 10. Suppose there exists a data structure that distinguishes whether after any two edges fail the diameter of the resulting graph is bounded by 3 or at least 5. We can use it to infer the entry $M[i, j, y]$ for any triple $(i, j, y) \in [\sqrt{N}]^3$

of indices such that i and j differ from each other, and j and y differ. We compute the edges in F with respect to the indices $i \neq x = j \neq y$ and apply Lemma 14 to check whether $M[i,j,y] = M[i,x,y] = 1$ or $M[i,j,y] = M[i,x,y] = 0$. For the assertion in Lemma 10 about the ST-diameter, we choose $S = A$ and $T = D$. Since there are $2^{\sqrt{N}(\sqrt{N}-1)^2} = 2^{\Omega(n^{3/2})}$ collections of possible answers, the oracle must take $\Omega(n^{3/2})$ bits of space.

References

1 Donald Aingworth, Chandra Chekuri, Piotr Indyk, and Rajeev Motwani. Fast Estimation of Diameter and Shortest Paths (Without Matrix Multiplication). *SIAM Journal on Computing*, 28:1167–1181, 1999. doi:10.1137/S0097539796303421.

2 Noga Alon, Shiri Chechik, and Sarel Cohen. Deterministic Combinatorial Replacement Paths and Distance Sensitivity Oracles. In *Proceedings of the 46th International Colloquium on Automata, Languages, and Programming, (ICALP)*, pages 12:1–12:14, 2019. doi:10.4230/LIPIcs.ICALP.2019.12.

3 Bertie Ancona, Monika Henzinger, Liam Roditty, Virginia Vassilevska Williams, and Nicole Wein. Algorithms and Hardness for Diameter in Dynamic Graphs. In *Proceedings of the 46th International Colloquium on Automata, Languages, and Programming (ICALP)*, pages 13:1–13:14, 2019. doi:10.4230/LIPIcs.ICALP.2019.13.

4 Arturs Backurs, Liam Roditty, Gilad Segal, Virginia Vassilevska Williams, and Nicole Wein. Toward Tight Approximation Bounds for Graph Diameter and Eccentricities. *SIAM Journal on Computing*, 50:1155–1199, 2021. doi:10.1137/18M1226737.

5 Surender Baswana, Keerti Choudhary, Moazzam Hussain, and Liam Roditty. Approximate Single-Source Fault Tolerant Shortest Path. *ACM Transactions on Algorithms*, 16:44:1–44:22, 2020. doi:10.1145/3397532.

6 Surender Baswana, Keerti Choudhary, and Liam Roditty. Fault-Tolerant Subgraph for Single-Source Reachability: General and Optimal. *SIAM Journal on Computing*, 47:80–95, 2018. doi:10.1137/16M1087643.

7 Surender Baswana and Telikepalli Kavitha. Faster Algorithms for Approximate Distance Oracles and All-Pairs Small Stretch Paths. In *Proceedings of the 47th Symposium on Foundations of Computer Science (FOCS)*, pages 591–602, 2006. doi:10.1109/FOCS.2006.29.

8 Surender Baswana and Neelesh Khanna. Approximate Shortest Paths Avoiding a Failed Vertex: Near Optimal Data Structures for Undirected Unweighted Graphs. *Algorithmica*, 66:18–50, 2013. doi:10.1007/s00453-012-9621-y.

9 Michael A. Bender and Martin Farach-Colton. The LCA problem revisited. In Gaston H. Gonnet, Daniel Panario, and Alfredo Viola, editors, *LATIN 2000: Theoretical Informatics, 4th Latin American Symposium, Punta del Este, Uruguay, April 10-14, 2000, Proceedings*, volume 1776 of *Lecture Notes in Computer Science*, pages 88–94. Springer, 2000. doi:10.1007/10719839_9.

10 Aaron Bernstein and David R. Karger. Improved Distance Sensitivity Oracles via Random Sampling. In *Proceedings of the 19th Symposium on Discrete Algorithms (SODA)*, pages 34–43, 2008. URL: https://dl.acm.org/citation.cfm?id=1347082.1347087.

11 Aaron Bernstein and David R. Karger. A Nearly Optimal Oracle for Avoiding Failed Vertices and Edges. In *Proceedings of the 41st Symposium on Theory of Computing (STOC)*, pages 101–110, 2009. doi:10.1145/1536414.1536431.

12 Davide Bilò, Keerti Choudhary, Luciano Gualà, Stefano Leucci, Merav Parter, and Guido Proietti. Efficient Oracles and Routing Schemes for Replacement Paths. In *Proceedings of the 35th Symposium on Theoretical Aspects of Computer Science (STACS)*, pages 13:1–13:15, 2018. doi:10.4230/LIPIcs.STACS.2018.13.

13 Davide Bilò, Luciano Gualà, Stefano Leucci, and Guido Proietti. Compact and Fast Sensitivity Oracles for Single-Source Distances. In Piotr Sankowski and Christos D. Zaroliagis, editors, *Proceedings of the 24th European Symposium on Algorithms (ESA)*, pages 13:1–13:14, 2016. doi:10.4230/LIPIcs.ESA.2016.13.

14 Davide Bilò, Luciano Gualà, Stefano Leucci, and Guido Proietti. Multiple-Edge-Fault-Tolerant Approximate Shortest-Path Trees. *Algorithmica*, 84:37–59, 2022. doi:10.1007/s00453-021-00879-8.

15 Davide Bilò, Keerti Choudhary, Sarel Cohen, Tobias Friedrich, and Martin Schirneck. Deterministic Sensitivity Oracles for Diameter, Eccentricities and All Pairs Distances. In *Proceedings of the 49th International Colloquium on Automata, Languages, and Programming (ICALP)*, pages 22:1–22:19, 2022. doi:10.4230/LIPIcs.ICALP.2022.22.

16 Davide Bilò, Sarel Cohen, Tobias Friedrich, and Martin Schirneck. Near-Optimal Deterministic Single-Source Distance Sensitivity Oracles. In *Proceedings of the 29th European Symposium on Algorithms (ESA)*, pages 18:1–18:17, 2021. doi:10.4230/LIPIcs.ESA.2021.18.

17 Davide Bilò, Sarel Cohen, Tobias Friedrich, and Martin Schirneck. Space-Efficient Fault-Tolerant Diameter Oracles. In *Proceedings of the 46th International Symposium on Mathematical Foundations of Computer Science (MFCS)*, pages 18:1–18:16, 2021. doi:10.4230/LIPIcs.MFCS.2021.18.

18 Jan van den Brand and Thatchaphol Saranurak. Sensitive Distance and Reachability Oracles for Large Batch Updates. In *Proceedings of the 60th Symposium on Foundations of Computer Science (FOCS)*, pages 424–435, 2019. doi:10.1109/FOCS.2019.00034.

19 Sergio Cabello, Erin W. Chambers, and Jeff Erickson. Multiple-Source Shortest Paths in Embedded Graphs. *SIAM J. Comput.*, 42:1542–1571, 2013. doi:10.1137/120864271.

20 Diptarka Chakraborty and Keerti Choudhary. New Extremal Bounds for Reachability and Strong-Connectivity Preservers Under Failures. In *Proceedings of the 47th International Colloquium on Automata, Languages, and Programming (ICALP)*, pages 25:1–25:20, 2020. doi:10.4230/LIPIcs.ICALP.2020.25.

21 Abraham Charnes. Optimality and Degeneracy in Linear Programming. *Econometrica*, 20:160–170, 1952.

22 Shiri Chechik and Sarel Cohen. Distance Sensitivity Oracles with Subcubic Preprocessing Time and Fast Query Time. In *Proccedings of the 52nd Symposium on Theory of Computing (STOC)*, pages 1375–1388, 2020. doi:10.1145/3357713.3384253.

23 Shiri Chechik, Sarel Cohen, Amos Fiat, and Haim Kaplan. $(1+\varepsilon)$-Approximate f-Sensitive Distance Oracles. In *Proceedings of the 28th Symposium on Discrete Algorithms (SODA)*, pages 1479–1496, 2017. doi:10.1137/1.9781611974782.96.

24 Shiri Chechik, Michael Langberg, David Peleg, and Liam Roditty. f-Sensitivity Distance Oracles and Routing Schemes. *Algorithmica*, 63:861–882, 2012. doi:10.1007/s00453-011-9543-0.

25 Shiri Chechik, Daniel H. Larkin, Liam Roditty, Grant Schoenebeck, Robert E. Tarjan, and Virginia Vassilevska Williams. Better Approximation Algorithms for the Graph Diameter. In *Proceedings of the 25th Symposium on Discrete Algorithms (SODA)*, pages 1041–1052, 2014. doi:10.1137/1.9781611973402.78.

26 Keerti Choudhary and Omer Gold. Extremal Distances in Directed Graphs: Tight Spanners and Near-Optimal Approximation Algorithms. In *Proceedings of the 31st Symposium on Discrete Algorithms (SODA)*, pages 495–514, 2020. doi:10.1137/1.9781611975994.30.

27 Pierluigi Crescenzi, Roberto Grossi, Leonardo Lanzi, and Andrea Marino. On Computing the Diameter of Real-World Directed (Weighted) Graphs. In Ralf Klasing, editor, *Proceedings of the 11th Symposium on Experimental Algorithms (SEA)*, pages 99–110, 2012. doi:10.1007/978-3-642-30850-5_10.

28 Mina Dalirrooyfard, Virginia Vassilevska Williams, Nikhil Vyas, and Nicole Wein. Tight Approximation Algorithms for Bichromatic Graph Diameter and Related Problems. In *Proceedings of the 46th International Colloquium on Automata, Languages, and Programming (ICALP)*, pages 47:1–47:15, 2019. doi:10.4230/LIPIcs.ICALP.2019.47.

29 Camil Demetrescu and Mikkel Thorup. Oracles for Distances Avoiding a Link-Failure. In *Proceedings of the 13th Symposium on Discrete Algorithms (SODA)*, pages 838–843, 2002. URL: https://dl.acm.org/citation.cfm?id=545381.545490.

30 Camil Demetrescu, Mikkel Thorup, Rezaul A. Chowdhury, and Vijaya Ramachandran. Oracles for Distances Avoiding a Failed Node or Link. *SIAM Journal on Computing*, 37:1299–1318, 2008. doi:10.1137/S0097539705429847.

31 Ran Duan, Yong Gu, and Hanlin Ren. Approximate Distance Oracles Subject to Multiple Vertex Failures. In *Proceedings of the 32nd Symposium on Discrete Algorithms (SODA)*, pages 2497–2516, 2021. doi:10.1137/1.9781611976465.148.

32 Ran Duan and Seth Pettie. Dual-Failure Distance and Connectivity Oracles. In *Proceedings of the 20th Symposium on Discrete Algorithms (SODA)*, pages 506–515, 2009. URL: http://dl.acm.org/citation.cfm?id=1496770.1496826.

33 Ran Duan and Seth Pettie. Connectivity Oracles for Failure Prone Graphs. In Leonard J. Schulman, editor, *Proceedings of the 42nd Symposium on Theory of Computing (STOC)*, pages 465–474, 2010. doi:10.1145/1806689.1806754.

34 Ran Duan and Hanlin Ren. Maintaining Exact Distances under Multiple Edge Failures. In *Proceedings of the 54th Symposium on Theory of Computing (STOC)*, pages 1093–1101, 2022. doi:10.1145/3519935.3520002.

35 Fabrizio Grandoni and Virginia Vassilevska Williams. Improved Distance Sensitivity Oracles via Fast Single-Source Replacement Paths. In *Proceedings of the 53rd Symposium on Foundations of Computer Science (FOCS)*, pages 748–757, 2012. doi:10.1109/FOCS.2012.17.

36 Fabrizio Grandoni and Virginia Vassilevska Williams. Faster Replacement Paths and Distance Sensitivity Oracles. *ACM Transaction on Algorithms*, 16:15:1–15:25, 2020. doi:10.1145/3365835.

37 Yong Gu and Hanlin Ren. Constructing a Distance Sensitivity Oracle in $O(n^{2.5794}M)$ Time. In *Proceedings of the 48th International Colloquium on Automata, Languages, and Programming (ICALP)*, pages 76:1–76:20, 2021. doi:10.4230/LIPIcs.ICALP.2021.76.

38 Manoj Gupta and Aditi Singh. Generic Single Edge Fault Tolerant Exact Distance Oracle. In *Proceedings of the 45th International Colloquium on Automata, Languages, and Programming, (ICALP)*, pages 72:1–72:15, 2018. doi:10.4230/LIPIcs.ICALP.2018.72.

39 David Hartvigsen and Russell Mardon. The All-Pairs Min Cut Problem and the Minimum Cycle Basis Problem on Planar Graphs. *SIAM J. Discret. Math.*, 7:403–418, 1994. doi:10.1137/S0895480190177042.

40 Monika Henzinger, Andrea Lincoln, Stefan Neumann, and Virginia Vassilevska Williams. Conditional Hardness for Sensitivity Problems. In *Proceedings of the 8th Conference on Innovations in Theoretical Computer Science (ITCS)*, pages 26:1–26:31, 2017. doi:10.4230/LIPIcs.ITCS.2017.26.

41 Ketan Mulmuley, Umesh V. Vazirani, and Vijay V. Vazirani. Matching Is as Easy as Matrix Inversion. *Comb.*, 7:105–113, 1987. doi:10.1007/BF02579206.

42 Hanlin Ren. Improved Distance Sensitivity Oracles with Subcubic Preprocessing Time. *Journal of Computer and System Sciences*, 123:159–170, 2022. doi:10.1016/j.jcss.2021.08.005.

43 Liam Roditty. Approximating the Diameter. In Ming-Yang Kao, editor, *Encyclopedia of Algorithms*, pages 116–117. Springer, New York City, NY, USA, 2016. doi:10.1007/978-1-4939-2864-4_566.

44 Liam Roditty and Virginia Vassilevska Williams. Fast Approximation Algorithms for the Diameter and Radius of Sparse Graphs. In *Proceedings of the 45th Symposium on Theory of Computing (STOC)*, pages 515–524, 2013. doi:10.1145/2488608.2488673.

45 Frank W. Takes and Walter A. Kosters. Determining the Diameter of Small World Networks. In Craig Macdonald, Iadh Ounis, and Ian Ruthven, editors, *Proceedings of the 20th Conference on Information and Knowledge Management (CIKM)*, pages 1191–1196, 2011. doi:10.1145/2063576.2063748.

46 Mikkel Thorup. Undirected Single-Source Shortest Paths with Positive Integer Weights in Linear Time. *Journal of the ACM*, 46:362–394, 1999. doi:10.1145/316542.316548.

47 Oren Weimann and Raphael Yuster. Replacement Paths and Distance Sensitivity Oracles via Fast Matrix Multiplication. *ACM Transactions on Algorithms*, 9:14:1–14:13, 2013. doi:10.1145/2438645.2438646.