Light Edge Fault Tolerant Graph Spanners

Greg Bodwin[*] Michael Dinitz[†] Ama Koranteng[†] Lily Wang[*]

Abstract

Traditionally, the study of graph spanners has focused on the tradeoff between stretch and two different notions of size: the sparsity (number of edges) and the lightness (total weight, normalized by the weight of an MST). There has recently been significant interest in *fault tolerant* spanners, which are spanners that still maintain their stretch guarantees after some nodes or edges fail. This work has culminated in an almost complete understanding of the three-way tradeoff between stretch, sparsity, and number of faults tolerated. However, despite some progress in specific metric spaces (e.g., [Le-Solomon-Than FOCS '23]) there have been no results to date on the tradeoff in general graphs between stretch, *lightness*, and number of faults tolerated.

We initiate the study of light edge fault tolerant (EFT) graph spanners, obtaining the first such results:

- First, we observe that lightness can be unbounded in the fault-tolerant setting if we use the traditional definition (normalizing by the MST), even when tolerating just a single edge fault.
- We then argue that a natural definition of lightness in the fault-tolerant setting is to instead normalize by a min-weight fault tolerant connectivity preserver; essentially, a fault-tolerant version of the MST. However, even with this, we show a new lower bound establishing that it is still not generally possible to construct f-EFT spanners whose weight compares reasonably to the weight of a min-weight f-EFT connectivity preserver.
- In light of this lower bound, it is natural to then consider *bicriteria* notions of lightness, where we compare the weight of an f-EFT spanner to a min-weight $(f' > f)$-EFT connectivity preserver. The most interesting question is to determine the minimum value of f' that allows for reasonable lightness upper bounds. Our main result is a precise answer to this question: $f' = 2f$. More formally, we show that the bicriteria lightness can be untenably large (roughly n/k, for a k-spanner) if one normalizes by the min-weight $(2f-1)$-EFT connectivity preserver, but that it is bounded by just $O(f^{1/2})$ times the corresponding bound on non-fault tolerant lightness (roughly $n^{1/k}$, for a $(1+\varepsilon)(2k-1)$-spanner) if one normalizes by the min-weight $2f$-EFT connectivity preserver instead.

Additional results include lower bounds on the $2f$-bicriteria lightness, improved f-dependence for $(2+\eta)f$-bicriteria lightness (for arbitrary constant $\eta > 0$), and a way to trade a slightly worse f-dependence for the ability to construct these spanners in polynomial time.

[*] University of Michigan. Supported by NSF:AF 2153680.

[†] Johns Hopkins University. Supported in part by NSF award 2228995.

1 Introduction

We study *graph spanners*, a basic kind of graph sparsifier that preserves distances up to a small stretch factor:

Definition 1 (Spanners)**.** Let $G = (V, E, w)$ be a weighted graph and let $k \geq 1$. A *k-spanner* of G is an edge-subgraph $H = (V, E' \subseteq E, w)$ for which $\mathtt{dist}_H(u, v) \leq k \cdot \mathtt{dist}_G(u, v)$ for all $u, v \in V$.

The value k is called the *stretch* of the spanner. Spanners were introduced by Peleg and Ullman [PU89a] and Peleg and Schäffer [PS89] in the context of distributed computing. Since then, they have found numerous applications, ranging from traditional distributed systems and networking settings [PU89b, AP90, ABP91, Pel00], to efficient data structures like distance oracles [TZ05], to preconditioning of linear systems [EEST08], and many others.

A large fraction of all work on graph spanners focuses on the tradeoff between the stretch k and the "size" of the spanner. One way to measure size is by sparsity (total number of edges). The tradeoff between stretch and sparsity was essentially resolved in a classic theorem by Althöfer et al. [ADD+93]:

Theorem 2 ([ADD+93])**.** *For every positive integer $k \geq 1$, every n-node graph G admits a $(2k - 1)$-spanner H with $|E(H)| \leq O(n^{1+1/k})$, and this bound is tight (assuming the Erdős girth conjecture [Erd63]).*

Besides having a spanner with few edges, in some applications one wants a spanner with small total edge weight. However, nothing along the lines of Theorem 2 will be possible: that is, not all graphs admit $(2k-1)$-spanners of total weight $w(H) \leq O(n^{1+1/k})$, or any other function of n. This is because we can always scale the edge weights of G as high as we like, and the edge weights of the spanner must scale accordingly. So, in order to study existential results for low-weight spanners, we need to tweak the definition: the standard move is to study *lightness*, which normalizes spanner weight by the weight of a minimum spanning tree ($\mathtt{mst}$).

Definition 3 (Lightness)**.** Given a subgraph H of a graph G, we define the lightness of H (with respect to G) to be the quantity

$$\ell(H \mid G) = \frac{w(H)}{w(\mathtt{mst}(G))}.$$

We will also write $\ell(H) := \ell(H \mid H)$.

This fixes the scale-invariance issue, and it is the dominant notion of "weighted size" in the study of spanners. It has been the subject of intensive study [ADD+93, CDNS92, ENS14, CWN16, LS23, BF25, Bod25], and reasonably tight bounds are known:

Theorem 4 ([LS23, Bod25])**.** *For every positive integer $k \geq 1$ and every $\varepsilon > 0$, every n-node graph G admits a $(1 + \varepsilon)(2k - 1)$-spanner H of lightness $\ell(H \mid G) \leq O(\varepsilon^{-1} n^{1/k})$. This dependence on n is tight assuming the Erdős girth conjecture [Erd63] (but the dependence on ε might be improvable).*

1.1 Fault Tolerance

Another important aspect of spanners goes back to their origins in distributed systems and computer networking. An issue that affects real-life systems that is not captured by the standard notion of spanners is the possibility of *failure*. If some edges (e.g., communication links) or vertices (e.g., computer processors) fail, what remains of the spanner might not still approximate the distances in what remains of the original graph. This motivates the notion of *fault tolerant* spanners, originally studied in the geometric setting by Levcopoulos, Narasimhan, and Smid [LNS98] and then (more generally) the setting of doubling metrics; see, e.g., [LNS98, Luk99, CZ04, NS07, BHPO20, Sol14, BDMS13, CLN15, CLNS15, LST23] and references within. In particular, the tradeoffs between the stretch and both the sparsity [LNS98, Luk99, CZ04, CLN15, CLNS15, LST23] and the lightness [CZ04, Sol14, CLNS15, LST23] for geometric spanners, including in the fault tolerant setting, have been the subject of significant study and interest, and optimal bounds are now known [CZ04, LST23].

The study of fault-tolerant spanners in general graphs was initiated in a seminal paper by Chechik, Langberg, Peleg, and Roditty [CLPR10]. They introduced the following definition.

Definition 5 (Edge Fault Tolerant Spanners [CLPR10])**.** A subgraph H is an f-edge fault tolerant (f-EFT) k-spanner of $G = (V, E)$ if

$$\mathtt{dist}_{H \setminus F}(u, v) \leq k \cdot \mathtt{dist}_{G \setminus F}(u, v)$$

for all $u, v \in V$ and $F \subseteq E$ with $|F| \leq f$.

Note that faults in general graphs are significantly more complex than in geometric settings, since in a general graph the failure of an edge or node can affect many distances (any pair that uses the failed object in the shortest path), while in a geometric setting the distances between nodes are fixed in the underlying metric space regardless of failures. Nevertheless, Chechik et al. [CLPR10] were able to obtain the following strong upper bound.

Theorem 6 ([CLPR10])**.** *For all positive integers n, k, f, every n-node weighted graph has an f-EFT $(2k-1)$-spanner H on $|E(H)| \leq O(f \cdot n^{1+1/k})$ edges.*

They also showed a bound of $|E(H)| \leq \exp(f) \cdot n^{1+1/k}$ for an analogous notion of f-vertex fault tolerant (VFT) spanners (see Section 5). Crucially, the dependence on n in these theorems is identical to the dependence on n in the non-faulty setting ($n^{1+1/k}$, from Theorem 2). Hence further work on sparsity of fault-tolerant spanners has focused on improving and lower bounding the dependence on the fault tolerance parameter f in the spanner size. This has led to a long line of work [CLPR10, DK11, BDPW18, BP19, DR20, BDR21, BDR22, Par22, BHP24, PST24], which gradually improved on the pioneering f-dependencies of [CLPR10] in both settings. This has culminated in optimal sparsity bounds for VFT spanners [BP19, BDR21], near-optimal sparsity bounds for EFT spanners [BDR22], and efficient construction algorithms [DR20, BDR21, Par22].

But what about the *lightness* of fault-tolerant spanners in the general graph setting? An analogous upper bound would be that every n-node graph admits an f-EFT $(1+\epsilon)(2k-1)$-spanner H of lightness roughly[1]

$$\ell(H \mid G) \leq O_{\varepsilon, f}\left(n^{1/k}\right), \tag{1}$$

that is, matching the optimal dependence on n from Theorem 4. But to date, no such result has been achieved, despite the intensive study of both (non-faulty) spanner lightness and of fault-tolerant spanner sparsity. It is thus the next natural question for the area; indeed, the existence of light fault-tolerant graph spanners was explicitly raised as an open problem in a recent survey talk at the Simons Institute [Le24].

In this paper we initiate the study of light fault-tolerant spanners. Our contribution is threefold. First, we explain the lack of previous results by showing that the first two natural attempts to *define* lightness in the fault-tolerant setting both encounter strong lower bounds, showing that no upper bound of this form (i.e., the form of (1)) are possible. Second, we propose a bicriteria notion of lightness (which we call *competitive lightness*), and we pin down the exact threshold of bicriteria approximation at which analogous upper bounds for fault tolerant spanners become available. Third, we provide bounds on the required and achievable dependence on f in competitive lightness.

1.2 Main Result: Defining Fault-Tolerant Lightness

1.2.1 (Failed) Attempt 1: Normalize by an MST

The obvious first attempt to study light fault-tolerant spanners is to keep the same definition as before: lightness is the weight of the spanner divided by the weight of an MST. However, it turns out that this definition suffers from a scaling issue, similar to the one discussed in the original definition of lightness: this measure of lightness can be unbounded.

More specifically, consider a graph G (see Figure 1) on three nodes u, v, w, with $w(u, v) = w(u, w) = 1$ and $w(v, w) = W$. Then $\mathtt{mst}(G)$ has two edges $\{u, v\}, \{u, w\}$ and total weight 2, while any 1-EFT spanner H with finite stretch must include all three edges, and so will have weight $W + 2$. So we have

$$\frac{w(H)}{\mathtt{mst}(G)} \geq \Omega(W),$$

where W may be selected as large as we like, demonstrating that this notion of lightness is unbounded.

[1] Throughout the paper, we use O_x notation to hide factors that depend on the variable(s) x, i.e., they are constant when x is a constant.

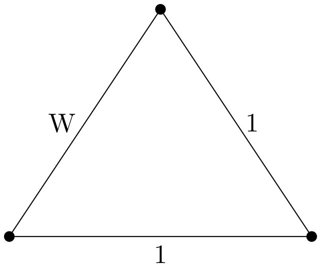

Figure 1: This graph G shows that the quantity $\frac{w(H)}{w(\texttt{mst}(G))}$ can be unbounded for 1-EFT spanners. Here $w(G) = W + 2$ for any W we like, and $w(\texttt{mst}(G)) = 2$, but we cannot remove any edges in a 1-EFT spanner. Thus a different definition of lightness is needed in the fault tolerant setting.

1.2.2 (Failed) Attempt 2: Normalize by a Fault-Tolerant Connectivity Preserver

Normalizing by the MST is not clearly the natural notion of fault tolerant lightness in the first place. In the non-faulty setting, the MST is the cheapest subgraph that preserves *connectivity*, and the lowest-weight spanner is the cheapest subgraph that preserves *approximate distances*. Thus non-faulty lightness can be interpreted as the "relative price of approximating distances," as compared to just connectivity.

So arguably a more natural definition of fault tolerant lightness would measure the price of *fault-tolerant distance approximation* vs. *fault-tolerant connectivity*. That is, instead of normalizing by an MST, our next attempt will be to normalize by the minimum weight *f-edge fault connectivity preserver* [NI92, NI08, DKK22, DKKN23][2].

Definition 7 (EFT Connectivity Preservers)**.** A subgraph $H \subseteq G$ is an *f-edge fault tolerant (EFT) connectivity preserver* if for any fault set $F \subseteq E$ with $|F| \leq f$, the connected components of $H \setminus F$ are identical to the connected components of $G \setminus F$. Or equivalently, for all pairs of nodes $u, v \in V$, they are connected in $H \setminus F$ if and only if they are connected in $G \setminus F$.

This leads to a natural definition of lightness:

Definition 8 (Competitive Lightness)**.** Given a graph G, let $T_f(G)$ denote the set of all f-EFT connectivity preservers of G. Then we define the f-competitive lightness of a subgraph H of G to be

$$\ell_f(H \mid G) := \frac{w(H)}{\min_{Q \in T_f(G)} w(Q)}.$$

We will also write $\ell_f(H) := \ell_f(H \mid H)$.

We refer to f in this definition as the "lightness competition parameter", or just the "competition parameter." We note that $\ell_0(H \mid G) = \ell(H \mid G)$,[3] and so we recover classical lightness as a special case. This definition successfully escapes the previous lower bound by setting $f > 0$, and in particular it is natural to hypothesize that all n-node graphs will admit an f-EFT $(1+\varepsilon)(2k-1)$-spanner H with competitive lightness $\ell_f(H \mid G) \leq O_{\varepsilon,f}(n^{1/k})$. Unfortunately, we refute this possibility with a new lower bound: the dependence on n for this notion of lightness must be essentially *linear*.

Theorem 9. *For any $f, k \geq 1$, there is a family of n-node weighted graphs G for which every f-EFT k-spanner H has competitive lightness*

$$\ell_f(H \mid G) \geq \Omega\left(\frac{n}{f^2 k}\right).$$

(See Section 4 for the proof.) So, in light of this lower bound, we need to again revisit our search for a definition of fault-tolerant lightness that admits reasonable existential upper bounds.

[2] An f-EFT connectivity preserver is also sometimes called an $f+1$-connectivity certificate [NI92, NI08]. This other terminology is more commonly used when the focus is on sparsity or computation time, rather than weights.

[3] In particular, if $\ell = 0$ then $T_f(G)$ consists of all connected subgraphs, and so $\min_{Q \in T_f(G)} w(Q) = w(\texttt{mst}(G))$.

1.2.3 (Successful) Attempt 3: Bicriteria Competitive Lightness

The fix we propose is to use the paradigm of *bicriteria* approximation. In our context, this means that we will escape the lower bound in Theorem 9 by comparing the quality of an f-EFT spanner to an f'-EFT connectivity preserver, for some $f' > f$. In other words, we will consider the f'-competitive lightness of the best f-EFT spanner.

How high do we need to push f' in order to enable the study of existential bounds for light EFT spanners? The main result of this paper is an exact answer to this question:

Theorem 10 (Main Result)**.** *For all positive integers f, k, n and all $\varepsilon > 0$:*

- ***(Upper Bound)*** *Every n-node graph*[4] *G has an f-EFT $(1+\varepsilon)(2k-1)$-spanner H with $2f$-competitive lightness*
$$\ell_{2f}(H \mid G) \leq \texttt{poly}(f, \varepsilon) \cdot n^{1/k}.$$
- ***(Lower Bound)*** *There are n-node graphs G for which any f-EFT $(1+\varepsilon)(2k-1)$-spanner H has $(2f-1)$-competitive lightness*
$$\ell_{2f-1}(H \mid G) \geq \texttt{poly}(f, k) \cdot n.$$

Thus, the right answer is exactly $f' = 2f$. From a technical standpoint, we remark that there is some precedent for this factor of 2 gap: it arises for a similar technical reason as in the Nash-Williams tree packing theorem [NW61], a classic structural result for fault-tolerant graph connectivity.

1.3 The Bicriteria Price of Fault Tolerance

Theorem 10 is the point of this paper: it initiates the study of $2f$-competitive spanner lightness, and it shows that this is essentially the strictest definition of fault-tolerant lightness that one can hope for. But now that we have settled the right setting of the lightness competition parameter, we are in a similar situation to sparsity bounds after [CLPR10]: what is the right dependence on f in the $2f$-competitive lightness? We know from Theorem 10 that $\texttt{poly}(f)$ is an upper bound, but can we be more precise?

We begin to answer this question, proving upper and lower polynomial bounds. Both our upper and lower bounds work by reducing to the setting of non-faulty light spanners. In turn, the bounds for non-faulty light spanners can be understood via the "weighted girth" framework of [ENS14]. We overview this formally in Section 3.1, but informally, we define the weighted girth of G to be the minimum over all cycles C of the total weight of the cycle divided by the max weight edge of the cycle. Let $\lambda(n, k)$ be the maximum (classical) lightness of any n-node graph with weighted girth greater than k. Elkin, Neiman, and Solomon [ENS14] proved that every n-node graph has a k-spanner of lightness at most $\lambda(n, k+1)$, and that this bound is tight. We prove the following bounds on the $2f$-competitive lightness in terms of λ, which we can then instantiate with the known upper and lower bounds on this function:

Theorem 11. *For all positive integers f, k, n and all $\varepsilon > 0$:*

- ***(Upper Bound)*** *Every n-node graph G has an f-EFT $(1+\varepsilon)(2k-1)$-spanner H with competitive lightness*
$$\begin{aligned}\ell_{2f}(H \mid G) &\leq f^{1/2} \cdot O\left(\lambda\left(n, (1+\varepsilon)2k\right)\right) \\ &\leq f^{1/2} \cdot O_\varepsilon\left(n^{1/k}\right). \qquad \text{(by } \lambda \text{ upper bounds from [LS23, Bod25])}\end{aligned}$$
- ***(Lower Bound)*** *There are n-node graphs G for which any f-EFT $(1+\varepsilon)(2k-1)$-spanner H has competitive lightness*
$$\begin{aligned}\ell_{2f}(H \mid G) &\geq \Omega\left(\lambda\left(\frac{n}{(2f)^{1/2}}, (1+\varepsilon)2k\right)\right) \\ &\geq f^{-\frac{1}{2k}} \cdot \Omega\left(n^{1/k}\right) \qquad \text{(assuming the girth conjecture [Erd63].)}\end{aligned}$$

[4] This and all of our other upper bounds extend to multigraphs as well.

Note that it remains open *whether $2f$-competitive lightness should correlate positively or negatively with f*. This is an important gap, which we hope can be closed by future work.

We provide results for one more setting. We know from Theorem 10 that $2f$ is the smallest possible competitive parameter, and Theorem 11 provides bound on the $2f$-competitive lightness, but what if we increase the competition parameter slightly? Do we get even better bounds on the competitive lightness? We show that better bounds are indeed possible if we tolerate a higher competition parameter of $(2+\eta)f$.

Theorem 12. *For all positive integers f, k, n and all $\varepsilon > 0, \eta > 0$:*

- ***(Upper Bound)*** *Every n-node graph G has an f-EFT $(1+\varepsilon)(2k-1)$-spanner H with competitive lightness*

$$\begin{aligned} \ell_{(2+\eta)f}(H \mid G) &\leq O_\eta(1) \cdot \lambda\left(n, (1+\varepsilon)2k\right) \\ &\leq O(1) \cdot O_{\varepsilon,\eta}\left(n^{1/k}\right). \qquad \text{(by } \lambda \text{ upper bounds from [LS23, Bod25])} \end{aligned}$$

- ***(Lower Bound)*** *There are n-node graphs G for which any f-EFT $(1+\varepsilon)(2k-1)$-spanner H has competitive lightness*

$$\begin{aligned} \ell_{(2+\eta)f}(H \mid G) &\geq \Omega\left(\lambda\left(\frac{n}{((2+\eta)f)^{1/2}}, (1+\varepsilon)2k\right)\right) \\ &\geq f^{-\frac{1}{2k}} \cdot \Omega\left(n^{1/k}\right) \qquad \text{(assuming the girth conjecture [Erd63].)} \end{aligned}$$

In other words: for slightly higher competition parameters we can completely remove the f-dependence from the upper bound, but our lower bound does not degrade at all. While there are still $\texttt{poly}(f)$ gaps in this setting, between $O(1)$ and $\Omega(f^{-1/k})$, this time we can at least say that competitive lightness is nonincreasing with f and that the gap disappears in the limit of large k.

1.4 Efficient Construction Algorithms

The spanners in our previous theorems are all achieved by a simple greedy algorithm, which is only a light variant on the greedy algorithm that is standard in prior work [BDPW18, BP19] (see Algorithm 1). That is:

1. Initialize the spanner H as a min-weight $2f$-EFT (or $(2+\eta)f$-EFT) connectivity preserver.
2. For each remaining edge (u, v) in the input graph order of increasing weight, add (u, v) to the spanner H iff there exists a set of edge failures F under which $\texttt{dist}_{H \setminus F}(u, v) > k \cdot w(u, v)$.

This algorithm is simple, and it trivially produces a correct spanner, but unfortunately it is not efficient. In fact, both steps encode NP-hard problems, and thus run in exponential time.

First: computing a min-weight connectivity preserver generalizes the $f+1$-Edge Connected Spanning Subgraph problem, which is NP-hard and which has received significant research attention [CT00, GGTW09, GG12]). Thanks to previous work, it is straightforward to handle this issue: 2-approximations for min-weight f-EFT connectivity preserver have recently been developed [DKK22, DKKN23], building off of the seminal work on Survivable Network Design by Jain [Jai01]. Using these approximation algorithms in step 1 will affect the lightness bounds only by a constant factor.

Second: testing whether there exists a fault set forcing us to add (u, v) to the spanner encodes the Length-Bounded Cut problem, which is also NP-hard [BEH+10]. In prior work on spanner *sparsity*, there have been two strategies to address this: a simple one with a minor penalty to sparsity [DR20], or a complex one with no penalty to sparsity [BDR21]. Unfortunately, the simpler of these approaches does not work at all for lightness. The more complex one works, but for technical reasons it loses an extra $f^{1/2}$ in the $2f$-competitive lightness setting (but is lossless in the $(2+\eta)f$-competitive setting). See Section 2.2 and Section 3.5 for more details. This gives the following dependencies for *efficiently constructable* light EFT spanners:

Theorem 13. *For all positive integers f, k, n and all $\varepsilon > 0, \eta > 0$, there is a randomized polynomial time algorithm that takes as input an n-node weighted graph G, and with high probability returns an f-EFT $(1+\varepsilon)(2k-1)$-spanner H of competitive lightness*

$$\begin{aligned}\ell_{2f}(H \mid G) &\leq O\left(f \cdot \lambda(n, (1+\varepsilon)2k)\right) \\ &\leq f \cdot O_\varepsilon\left(n^{1/k}\right)\end{aligned} \qquad \text{or} \qquad \begin{aligned}\ell_{(2+\eta)f}(H \mid G) &\leq O_\eta\left(\lambda(n, (1+\varepsilon)2k)\right) \\ &\leq O(1) \cdot O_{\varepsilon,\eta}\left(n^{1/k}\right).\end{aligned}$$

1.5 Paper Outline

We begin in Section 2 with a high-level overview of our approach. We then get into technical details in Section 3. In order to clearly show the main ideas, we warm up with an analysis that gives weaker bounds in Section 3.3. The proofs of our main upper bounds in Theorems 10, 11, and 12 are in Section 3.4. The proof of our polytime algorithm (Theorem 13) is in Section 3.5. The lower bound parts of these theorems appear in Section 4.

2 Technical Overview

The bulk of the technical work in this paper involves analyzing the (slightly modified) fault tolerant greedy algorithm to prove the upper bound part of our main results. These proofs are closely connected.

2.1 Overview of Upper Bounds

A highly successful strategy to analyze the *sparsity* of *non-fault tolerant* spanners is to construct them using a simple greedy algorithm, and show that it produces spanners of high girth (shortest cycle length) [ADD+93]. Then one can apply results from extremal graph theory stating that all high-girth graphs, including these spanners, must be sparse. Previous work has independently extended this method to analyze the *sparsity* of fault tolerant spanners [BDPW18, BP19, BDR22], and to the lightness of *non-fault tolerant* spanners [ENS14]. Since we deal with light fault tolerant spanners, our proof strategy can broadly be described as a composition of these two extensions, as well as some new technical tools that handle the issues that arise from their interaction.

Algorithm and Blocking Sets. As discussed, our algorithm for constructing a f-EFT k-spanner is the following. Let Q be the optimal $2f$-EFT connectivity preserver of G, i.e., the graph that we are competing with (the weight in the denominator of our lightness definition). We begin by adding Q to the spanner H. Then we use the standard greedy algorithm: we consider each other edge $e \in E(G) \setminus Q$ in nondecreasing weight order, and we add $e = \{u, v\}$ to H if there is some set $F_e \subseteq E(H)$ with $|F_e| \leq f$ such that $\texttt{dist}_{H \setminus F_e}(u, v) > k \cdot \texttt{dist}_{G \setminus F_e}(u, v)$. We say that e is in a *block* with each $e' \in F_e$, or equivalently that (e, e') form a block for each such e'. Note that e is the first edge of a block at most $|F_e| \leq f$ times. The collection of all blocks is called a blocking set.

Weighted Girth. Next, we explain the relationship to the weighted girth framework from [ENS14]. Standard arguments (first introduced by [BP19]) can be adapted to our modified algorithm to show that essentially every cycle in H with at most $k+1$ edges must contain two edges that form a block, i.e., the blocks cover all cycles with at most $k+1$ edges. Following a proof from [ENS14], this argument can be adapted to show that they also cover essentially all cycles with "normalized weight" at most $k+1$, i.e., cycles C where $w(C)/\max_{e \in C} w(e)$ is at most $k+1$. This is useful because if there were *no* cycles of normalized weight at most $k+1$ — or, in the language of [ENS14], H has weighted girth $> k+1$ — then by definition H has lightness at most $\lambda(n, k+1)$. So we have a spanner H where cycles of normalized weight at most $k+1$ exist but are blocked, and we know that if there were *no* such cycles then we have good lightness bounds. So our goal will be to turn H into some other graph H' without *any* cycles of normalized weight at most $k+1$, in a way that allows us to bound the lightness of H using the lightness of H'.

The Subsampling Method. In the sparsity setting, one of the now-standard ways of going from a graph H where short cycles exist but are blocked to a graph H' where there are no short cycles is by subsampling H to get H'. Informally, by choosing the correct probability, one can often show that no blocks survive in H', and thus there cannot be any short cycles and so there are not many edges in H'. On the other hand, since we obtained H' by subsampling H with a known probability, there are not too many more edges in H than in H'. Thus H cannot have many edges. A natural first attempt is to use the same idea for lightness, but it immediately runs into a serious problem. The connectivity preserver Q itself can have unblocked cycles of low normalized weight, and so if the subsampling does not remove any edges of Q then we cannot hope to get rid of all such cycles. However, if the subsampling *does* get rid of edges of Q, then the min-weight connectivity preserver of the subsampled graph H' might be very different from Q, making it hard to compare the lightnesses of H and H'. (Note that this problem does not show up when analyzing sparsity, since the number of edges is not normalized by anything and so it is relatively easy to compare $|E(H)|$ to $|E(H')|$).

To get around this, we first observe that if Q happened to be a tree that was disjoint from the blocking set then we would be in good shape: none of the low-normalized-weight cycles would use Q, so we could do standard subsampling of $E(H) \setminus Q$ and then include Q deterministically to get a subgraph with high weighted girth where Q still exists. Unfortunately, Q will not be a tree. But suppose that we could find at least $f + 1$ disjoint spanning trees in Q. Then since each edge $e \in E(H) \setminus Q$ is blocked by at most f other edges (the edges in F_e), we can find at least one of these trees which does not contain any edge blocked with e. We can then add e to this tree. Once we do this for every $e \in E(H) \setminus Q$, we have a partition of $E(H) \setminus Q$ into $f + 1$ disjoint subgraphs with the property that each subgraph contains a spanning tree of Q, and no edge in that subgraph includes a block with the spanning tree. So we're in the setting we want to be in, and we can show that each of these subgraphs has good lightness. Since Q and H are both partitioned across these subgraphs, this implies that we have good lightness overall.

Steiner Forest Packing. How can we find $f + 1$ disjoint spanning trees in Q? One very useful tool for finding disjoint spanning trees in graphs is the tree packing theorem of Nash-Williams [NW61], which implies that it suffices for Q to be $2f + 2$-connected. This is close to what we want, except for two issues:

1. First, Q is a $2f$-EFT connectivity preserver, which means that if G is $2f + 1$-connected (or more) then Q is $2f + 1$-connected. We desire $2f + 2$-connectivity, so we are off by one from our desired connectivity threshold.

2. Second, G need not be $2f + 1$-connected at all, and so Q need not even be $2f + 1$-connected.

Fortunately, it turns out that we can get around both of these issues via a "doubling trick" and by applying bounds on *Steiner Forest packing* rather than spanning tree packing. Since Q is a $2f$-EFT connectivity preserver, for any edge $(u, v) \in E(H) \setminus Q$ we know that the pair (u, v) is $2f + 1$-connected in G, or else we would have to include (u, v) in Q. So while Q is not $2f + 1$-connected, every pair u, v that are endpoints of an edge in $E(H) \setminus Q$ are $2f + 1$-connected in G and thus in Q. It turns out that this actually implies that there are $\Omega(f)$ disjoint *forests* in Q so that for each $\{u, v\} \in E(H) \setminus Q$, u and v are connected in all of the forests. In other words, we can find $\Omega(f)$ Steiner forests when the demand pairs are $e \in E(H) \setminus Q$. We would now be done if this $\Omega(f)$ was $f + 1$, but unfortunately it is more like $f/16$ [Lau05]; improving the constant in Steiner forest packing is a fascinating open problem, and the correct bound is not even known for Steiner trees [Kri03, DMP16].

So we add one more step: we first double every edge of Q to get a multigraph Q'. Now every $\{u, v\} \in E(H) \setminus Q$ is $4f + 2$-connected in Q' and, importantly, all degrees are even and thus Q' is Eulerian. This turns out to make Steiner Forest packing significantly easier, and optimal bounds are known: it was shown by Chekuri and Shepherd [CS09] that if all demand pairs u, v are $2k$-connected in an Eulerian graph G, then we can find k disjoint Steiner forests in G (and can even do so in polynomial time). This means that we can find $2f + 1$ disjoint Steiner forests in Q', and thus when interpreted in Q we get $2f + 1$ Steiner forests that are not disjoint, but where every edge of Q is in at most 2 of them.

But this is sufficient for us. Since each edge e is in a block with at most f edges of Q, and each edge of Q can appear at most two of $2f + 1$ Steiner forests, we get that there must be some Steiner forest which has no edge blocked with e, as desired.

To sum up: to bound the $2f$-competitive lightness of H, we double every edge of Q and use the Steiner Forest packing in Eulerian graphs result of [CS09] to find $2f+1$ edge-disjoint Steiner forests, which correspond to $2f+1$ Steiner forests in Q where every edge in Q appears at most twice. Then for every edge $e \in E(H) \setminus Q$ we add it to a Steiner forest which does not contain any of its blocks. Finally, for each of the resulting subgraphs we use subsampling techniques to bound their lightness.

This approach turns out to give a lightness bound of $f \cdot \lambda(n, k+1)$ (see Section 3.3). In order to improve the bound to $f^{1/2} \cdot \lambda(n, k+1)$, we need to be a bit more careful. At a very high level, we allow ourselves to add edges of $E(H) \setminus Q$ to *many* of the Steiner forests at once, where possible. If it is possible then we get an improved bound, roughly since our previous analysis overcounts multiply-added edges, letting us divide out an additional factor. For edges that cannot be added to many Steiner forests, it turns out the previous subsampling technique can be made more efficient, also giving an improved bound in this other case.

The upper bound in Theorem 12, with a higher competition parameter, has an almost identical proof to Theorem 11. The main difference is that, since we now have that Q is a $(2+\eta)f$-EFT connectivity preserver, if we repeat the above analysis then for any $e \in E(H) \setminus Q$ we can now find $\Omega_\eta(f)$ Steiner forests in the forest packing of Q that don't contain a block with e, rather than just 1. Thus if we add the weights of all of the subgraphs, we are overcounting every edge by a factor of $\Omega(f)$. When we plug this into the previous calculations, we end up exactly canceling out the f-dependence, giving the improved upper bound.

2.2 Running Time (Theorem 13)

One downside of the fault tolerant greedy algorithm is that it does not run in polynomial time. First, we need to seed it with Q, which as mentioned above requires solving an NP-hard problem. This is straightforward to get around, though, since we can just use a 2-approximation for the min-weight f-EFT connectivity preserver problem due to [DKK22, DKKN23]. The larger difficulty is that we need to check for every edge $e \in E(H) \setminus Q$ whether there is a fault set which forces us to include e. Doing this in the obvious way takes $\Omega(n^f)$ time. This drawback has been noticed since the fault tolerant greedy algorithm was introduced [BDPW18], and eventually two different methods were developed to design polytime versions of the algorithm:

1. The first approach, introduced by [DR20] for sparse fault-tolerant spanners, involves giving an approximation algorithm for the question of whether there exists a fault set forcing us to add e. This problem is NP-hard, but there are simple $O(k)$-approximations. By carefully analyzing the effect of approximating rather than solving this problem, [DR20] showed that this incurs only a relatively small extra loss in the sparsity. This approach has since been used extensively to to turn variants of the fault tolerant greedy algorithm into polynomial-time algorithms, albeit with a small extra loss [BDR22, BDN22, BDN23, BHP24, PST24].

2. In order to avoid this extra loss, [BDR21] designed a very different subsampling-based condition to use in the greedy spanner. Informally, the main idea of this technique is to move the subsampling from the analysis into the actual algorithm. So instead of just subsampling to analyze sparsity, we actually subsample to decide whether to include an edge. This method is conservative (it may add some edges even when there is no fault set forcing us to do so), but it can be proved that it does not add asymptotically more edges than the base fault tolerant greedy algorithm. Perhaps due to its relative complexity, this approach has not proved to be nearly as useful or popular as the approximation algorithm-based approach of [DR20].

It turns out that making our version of fault-tolerant greedy polynomial time without losing its lightness properties is significantly more complex than if we cared only about sparsity. Interestingly, the approximation algorithm approach of [DR20] does not work, as it fundamentally depends on being able to treat weighted edges as unweighted. This turns out to be OK for sparsity, but makes it impossible to use for lightness. However, the subsampling approach of [BDR21] *can* be made to work. Interestingly, this is the first setting (to the best of our knowledge) where the approximation algorithm approach does not work but the subsampling does; until now, it seemed as if the approximation approach was at least as flexible (and significantly easier to use) as the subsampling approach.

To get this approach to work we have to move the subsampling from the analysis to the algorithm. It turns out that we can do this by actually computing the Steiner Forest packing of [CS09] (fortunately, they showed that it could be computed efficiently). Then when we consider adding an edge e, for every Steiner forest in the packing we use the sampling idea from [BDR21] on the graph consisting of T and the already-added spanner edges that are not in Q. This basically suffices for Theorems 13. Unfortunately, the method we used before to improve the f dependence to $f^{1/2}$ for $2f$-competitive lightness (in Theorem 11) required subsampling different types of edges with different probabilities (depending on whether or not an edge could be added to many of the Steiner forests). Since the subsampling is now in the algorithm rather than the analysis, we cannot do this, and thus must fall back on a lightness bound of $f \cdot \lambda(n, k+1)$.

3 Constructions of Light Fault-Tolerant Spanners

3.1 The Weighted Girth Framework

Elkin, Neiman, and Solomon [ENS14] introduced a weighted analog of girth, for use in studying the lightness of (non-faulty) spanners:

Definition 14 (Normalized Weight and Weighted Girth [ENS14])**.** For a cycle C in a weighted graph G, the *normalized weight* of C is the quantity

$$w^*(C) := \frac{w(C)}{\max_{e \in C} w(e)}.$$

The *weighted girth* of G is the minimum normalized weight over its cycles.

They then proved an equivalence between light spanners and the maximum possible lightness of a graph of high weighted girth. In particular:

Definition 15 (Extremal Lightness of Weighted Girth)**.** We write $\lambda(n, k) := \sup \ell(G)$, where the sup is taken over n-node graphs G of weighted girth $> k$.

Theorem 16 ([ENS14])**.** *For all positive integers n and all k, every n-node graph G has a k-spanner H of lightness $\ell(H \mid G) \leq \lambda(n, k+1)$, and this is existentially tight.*[5]

Settling the asymptotic value of λ is a major open problem. Currently, the following bounds are known:

Theorem 17 ([LS23, Bod25, BF25])**.** *For all positive integers n, k and all $\varepsilon > 0$, we have $\lambda(n, (1+\varepsilon)2k) \leq O\left(\varepsilon^{-1} n^{1/k}\right)$. When k is a constant and $\varepsilon = \Theta(n^{-\frac{1}{2k-1}})$, we have $\lambda(n, (1+\varepsilon)2k) \geq \Omega\left(\varepsilon^{-1/k} n^{1/k}\right)$.*

3.2 The Greedy Algorithm and Blocking Sets

We will analyze spanners that arise from a variant of the fault-tolerant greedy algorithm, introduced in [BDPW18] and used in many recent papers on fault tolerance. The only difference algorithmically is that we seed the spanner with a min-weight connectivity preserver, whereas prior work immediately enters the main loop.

```
Input: Graph $G = (V, E, w)$, stretch $k$, fault tolerance $f$

Let $Q \leftarrow$ min-weight $2f$-FT connectivity preserver of $G$
Let $H \leftarrow Q$ be the initial spanner
foreach edge $(u, v) \in E(G) \setminus Q$ in order of nondecreasing weight do
    if there exists $F \subseteq E(H), |F| \leq f$ such that $\mathbf{dist}_{H \setminus F}(u, v) > k \cdot w(u, v)$ then
        add $(u, v)$ to $H$
Return $H$
```

Algorithm 1: Light Fault-Tolerant Greedy Spanner Algorithm

The proof of correctness of the algorithm is standard:

[5] More specifically, this means that for any $\varepsilon > 0$, there exists an n-node graph on which any k-spanner H has lightness $\ell(H \mid G) > \lambda(n, k+1) - \varepsilon$.

Theorem 18. *The output spanner H from Algorithm 1 is an f-EFT k-spanner of the input graph G.*

Proof. TOPROVE 0 □

Now we turn to analyzing lightness. When $f = 0$, this algorithm produces a spanner H of weighted girth $> k+1$ [ENS14]. For larger f, there is not much we can say about the weighted girth of the output spanner, which could be quite small. However, we might intuitively expect the output spanner to be "structurally close" to a graph of high weighted girth. That notion of closeness can be formalized by adapting the blocking set framework, which has been used in many recent papers on fault-tolerant spanners and related objects (see, e.g., [BP19, DR20, BDR21, BDR22, BDN22, BDN23, PST24, BHP24]).

Definition 19 (Edge-Blocking Sets)**.** For a graph H, an *edge-blocking set* is a set of ordered edge pairs $B \subseteq E(H) \times E(H)$. We say that B *blocks* a cycle C if there is a pair $(e_1, e_2) \in B$ with $e_1, e_2 \in C$. We say that B is *f-capped* if for all $e \in E(H)$, there are at most f edges e' with $(e, e') \in E(H)$.[6]

The following is a tweak on a standard lemma bounding the size of the blocking set of the output spanner; the proof is a straightforward mixture of related lemmas from [BP19] and [ENS14].

Lemma 20. *The output spanner H from the fault-tolerant greedy algorithm has an f-capped edge-blocking set B that blocks all cycles C with normalized weight $w^*(C) \le k+1$ and where a heaviest edge in C is not in Q. Additionally, for every pair $(e, e') \in B$, the first edge e is not in Q.*

Proof. TOPROVE 1 □

A subtlety here is that the output spanner H could still have unblocked cycles C of low weighted girth, in the case where the heaviest edge of C is in Q (and so it was added before the main greedy loop of the algorithm), and the edges in $C \setminus Q$ are much lighter. In general, min-weight f-EFT connectivity preservers are less well-behaved than min-weight spanning trees, and so situations like this are indeed possible even despite Q being min-weight. We will need to handle these cycles in our analysis in another way (see Lemma 26).

3.3 Warmup: Slow Suboptimal Competitive Lightness Upper Bounds

As a warmup, we will prove:

Theorem 21 (Warmup)**.** *For all input graphs G, the output spanner H from Algorithm 1 satisfies*

$$\ell_{2f}(H \mid G) \le O\left(f \cdot \lambda(n, k+1)\right).$$

We will later improve the dependence on f, and show a variant that runs in polynomial time. But it is easier to demonstrate our main ideas with this simpler result.

Extensions of Nash-Williams Tree Decompositions. To begin the proof, we recall the classic Nash-Williams tree packing theorem:

Theorem 22 (Nash-Williams [NW61])**.** *For all positive integers f, a multigraph G contains f edge disjoint spanning trees if and only if for every vertex partition $\mathcal{P}$, there are at least $f(|\mathcal{P}| - 1)$ edges between parts.*

The following is a simple corollary. The actual packing that we will use is a bit more complicated, but we give this simpler corollary to build intuition.

Corollary 23. *For all positive integers f, an f-connected graph G contains a collection of f spanning trees with the property that any edge of G is in at most two of the trees.*

Proof. TOPROVE 2 □

[6] There may be additional pairs of the form $(e', e) \in E(H)$ – this property only bounds the number of pairs that have e first.

If the input graph G is $(2f+1)$-connected (and therefore Q is also $(2f+1)$-connected), then this corollary would give the right technical tool. However, in order to avoid assuming any connectivity properties of the input graph, we instead need a generalization. We begin with the following theorem by Chekuri and Shepherd, which gives the appropriate analog in the special case of Eulerian graphs:

Theorem 24 ([CS09], c.f. Theorem 3.1)**.** *For all positive integers f and Eulerian graphs G, there exist edge-disjoint forests $\{F_1, \dots, F_f\}$ such that every $2f$-connected component of G is connected in all forests. Moreover, these forests can be constructed in poly(n) time.*

We will use the following corollary:

Corollary 25. *For all positive integers f and graphs G, there exists a collection of subtrees $\mathcal{T}$ such that (1) every edge in $E(G)$ is in at most two trees in $\mathcal{T}$, and (2) for every f-connected component C of G, there are at least f trees in $\mathcal{T}$ in which C is connected.*

Proof. TOPROVE 3 □

The first step in our analysis of Algorithm 1 is to apply this corollary to Q, with parameter $2f+1$, yielding a collection of subtrees $\mathcal{T}$.

Construction of $H[T]$ Subgraphs. Our next step is to construct a sequence of four subgraphs $H[T] \supseteq H'[T] \supseteq H''[T] \supseteq H'''[T]$ associated to each tree $T \in \mathcal{T}$. In the following let B be an f-capped blocking set for H as in Lemma 20.

- **(Construction of $H[T]$)** Consider the edges in $(u,v) \in E(H) \setminus Q$ one at a time. Notice that there cannot exist $2f$ edge faults in Q that disconnect the nodes u, v, since otherwise we would need to include the edge (u,v) in Q. Thus the node pair (u,v) is $(2f+1)$-connected in Q. So the endpoint nodes u, v lie in the same $(2f+1)$-connected component C of Q, and by Corollary 25 there exist $2f+1$ trees in $\mathcal{T}$ that all span C.

  Since B is f-capped, there are at most f edges e' with $(e, e') \in B$. Since each such edge e' can be in at most two trees (by Corollary 25), it follows that there exists at least one tree $T \in \mathcal{T}$ for which no such edge e' is in T (if there are multiple such trees, choose one arbitrarily). We choose such a tree T and say that it *hosts* e. Then, for all $T \in \mathcal{T}$, let $H[T]$ be the graph that contains T and all edges hosted by T.

- **(Construction of $H'[T]$)** For each tree T, we construct $H'[T]$ by keeping every edge in T deterministically, and keeping every edge in $H \setminus T$ independently with probability $1/f$.

- **(Construction of $H''[T]$)** To construct $H''[T]$ from $H'[T]$, for every pair $(e, e') \in B$ with $e, e' \in H'[T]$, we delete the first edge e, and let $H''[T]$ be the remaining subgraph.

- **(Construction of $H'''[T]$)** To construct $H'''[T]$ from $H''[T]$, we delete all edges in $T \setminus \texttt{mst}(H''[T])$.

The important properties of these subgraphs are:

Lemma 26. *Every graph $H'''[T]$ has weighted girth $> k+1$ (deterministically).*

Proof. TOPROVE 4 □

Lemma 27. *Every graph $H'''[T]$ has expected weight*

$$\mathbb{E}\left[w(H'''[T])\right] \geq \Omega\left(\frac{w(H[T])}{f}\right) - w(T).$$

Proof. TOPROVE 5 □

Analysis of Lightness. As in the probabilistic method, there exists a realization of the subgraphs $H'''[T]$ that satisfies the expected weight inequality in the previous lemma (for all T). Rearranging the bound from Lemma 27, we have:

$$\begin{aligned} w(H'''[T]) &\geq \Omega\left(\frac{w(H[T])}{f}\right) - w(T) \\ \implies O(f) \cdot (w(H'''[T]) + w(T)) &\geq w(H[T]). \end{aligned}$$

Using this, we observe that for all trees $T \in \mathcal{T}$, we have

$$\begin{aligned} \frac{w(H[T])}{w(T)} &\leq \frac{O\left(f\right) \cdot \left(w(H'''[T]) + w(T)\right)}{w(T)} \\ &\leq O(f) \cdot \frac{w(H'''[T])}{w(\mathtt{mst}(H'''[T]))} + O(f) \\ &= O(f) \cdot \ell(H'''[T]) + O(f) \\ &\leq O(f) \cdot \lambda(n, k+1) \qquad (2) \end{aligned}$$

where the last steps follow from the definition of λ, the fact that $\lambda(n, k+1) \geq 1$, and the previous lemma establishing that $H'''[T]$ has weighted girth $> k+1$. Thus, to wrap up the proof, we bound:

$$\begin{aligned} \ell_{2f}(H \mid G) = \frac{w(H)}{w(Q)} &\leq \frac{\sum_{T \in \mathcal{T}} w(H[T])}{w(Q)} && \text{(every } e \in E(H) \text{ is in at least one } H[T]) \\ &\leq 2 \cdot \frac{\sum_{T \in \mathcal{T}} w(H[T])}{\sum_{T \in \mathcal{T}} w(T)} && \text{(Corollary 23 and def of } \mathcal{T}) \\ &\leq O\left(\frac{\sum_{T \in \mathcal{T}} f \cdot \lambda(n, k+1) \cdot w(T)}{\sum_{T \in \mathcal{T}} w(T)}\right) && \text{(Eq. (2))} \\ &= O(f \cdot \lambda(n, k+1)). \end{aligned}$$

3.4 Improved Lightness Bounds via Multiple Host Trees

A way in which our previous analyses are suboptimal is that they do not acknowledge the possibility that an edge could be hosted by many trees, not just one. As a simple example to introduce the technique, let us observe that the lightness bound from Algorithm 1 improves by a factor of f if we are willing to take a higher lightness competition parameter:

Theorem 28. *Suppose we modify Algorithm 1 to instead construct Q as a min-weight $(2+\eta)f$-FT connectivity preservers, for some $\eta > 0$. Then the output f-EFT spanner H has $(2+\eta)f$-competitive lightness*

$$\ell_{(2+\eta)f}(H \mid G) \leq O\left(\eta^{-1}\lambda(n, k+1)\right).$$

Proof. TOPROVE 6 □

If we don't wish to harm our competition parameter, we can use a related method to improve the dependence to $f^{1/2}$:

Theorem 29. *The output spanner H from Algorithm 1 (with no modifications) satisfies*

$$\ell_{2f}(H \mid G) \leq O\left(f^{1/2} \cdot \lambda(n, k+1)\right).$$

Proof. TOPROVE 7 □

These two theorems now essentially directly imply the upper bound parts of Theorems 11 and 12:

Proof. TOPROVE 8 □

```
Input: Graph $G$, stretch $k$, fault tolerance $f$

Let $Q \leftarrow$ 2-approximate min-weight $2f$-FT connectivity preserver of $G$ // polytime by [DKK22]
Let $\mathcal{T} \leftarrow$ tree decomposition of $Q$ satisfying Corollary 25 w.r.t. connectivity parameter $2f+1$
Let $E \leftarrow \emptyset$ be the initial set of non-$Q$ spanner edges

// $c$ large enough constant
foreach edge $(u,v) \in E(G) \setminus Q$ in order of nondecreasing weight do
    foreach tree $T \in \mathcal{T}$ that spans the $(2f+1)$-connected component of $Q$ containing $u,v$ do
        sample $c \log n$ subgraphs by including $T$, and including each edge in $E$ with probability $1/f$
        let $\widehat{P}^T_{(u,v)}$ be the fraction of sampled subgraphs $H'$ in which $\texttt{dist}_{H'}(u,v) > k \cdot w(u,v)$
        if $\widehat{P}^T_{(u,v)} \geq 1/8$ then
            add $(u,v)$ to $E$
Return $H = Q \cup E$
```

Algorithm 2: FT spanners in polynomial time

3.5 Competitive Lightness Upper Bounds in Polynomial Time

We now prove Theorem 13, improving to polynomial runtime at the cost of a worse dependence on f (relative to the previous section). Recall the theorem:

Theorem 13. *For all positive integers f, k, n and all $\varepsilon > 0, \eta > 0$, there is a randomized polynomial time algorithm that takes as input an n-node weighted graph G, and with high probability returns an f-EFT $(1+\varepsilon)(2k-1)$-spanner H of competitive lightness*

$$\begin{aligned}\ell_{2f}(H \mid G) &\leq O\left(f \cdot \lambda(n,(1+\varepsilon)2k)\right) \\ &\leq f \cdot O_\varepsilon\left(n^{1/k}\right)\end{aligned} \qquad \text{or} \qquad \begin{aligned}\ell_{(2+\eta)f}(H \mid G) &\leq O_\eta\left(\lambda(n,(1+\varepsilon)2k)\right) \\ &\leq O(1) \cdot O_{\varepsilon,\eta}\left(n^{1/k}\right).\end{aligned}$$

The algorithm that we will analyze to prove this is Algorithm 2. Both the algorithm and its analysis are adaptations of the ideas of [BDR21]. To begin the analysis, let us set up some useful definitions. For an edge $e \in E(G) \setminus Q$, let us write E_e for the subset of non-Q spanner edges that were added strictly before the edge e was considered in the algorithm. We also write

$$H_e := Q \cup E_e \qquad \text{and} \qquad H^T_e := T \cup E_e$$

for trees $T \in \mathcal{T}$. For all $e \in E(G) \setminus Q, T \in \mathcal{T}$, we let H'^T_e be a random subgraph of H^T_e obtained by including the edges of T (deterministically), and then including each other edge from H^T_e independently with probability $1/f$. Then let

$$P^T_{e=(u,v)} := \Pr\left[\texttt{dist}_{H'^T_e}(u,v) > k \cdot w(u,v)\right]$$

where the probability is over the random definition of H'^T_e. We cannot compute $P^T_{e=(u,v)}$ exactly, but we can view Algorithm 2 as computing experimental estimates $\widehat{P}^T_{e=(u,v)}$ by repeatedly sampling subgraphs H'^T_e. The following lemma applies standard Chernoff bounds to show that our estimates are probably reasonably good. This argument directly follows one from [BDR21], but we recap it here from scratch.

Lemma 30 (c.f. [BDR21], Lemma 3.1)**.** *With high probability, for all $e \in E(G) \setminus Q$ and all $T \in \mathcal{T}$, we have $\widehat{P}^T_e \in P^T_e \pm \frac{1}{8}$.*

Proof. TOPROVE 9 □

The following lemmas will assume that the event from Lemma 30 holds, and hence these lemmas only hold with high probability, rather than deterministically. We next establish correctness:

Lemma 31. *With high probability, the output spanner H from Algorithm 2 is an f-EFT k-spanner of the input graph.*

Proof. TOPROVE 10 □

We now turn to bounding the number of edges in the final spanner. For an edge $(u, v) \in E$, we will say it is *hosted* by the first tree $T \in \mathcal{T}$ that caused the edge to be added to E in the main loop. We will write $H[T]$ for the subgraph of the final spanner that contains T and all of the edges that it hosts. Note that we are reusing this notation from the previous warmup; although the definition of $H[T]$ is slightly different here, it plays a directly analogous role.

Lemma 32. *With high probability, for all trees $T \in \mathcal{T}$, we have*

$$\frac{w(H[T])}{w(T)} \leq O\left(f \cdot \lambda(n, k+1)\right).$$

Proof. TOPROVE 11 □

We can now wrap up the proof:

Lemma 33. *With high probability, the output spanner H has $2f$-competitive lightness*

$$\ell_{2f}(H \mid G) \leq O\left(f \cdot \lambda(n, k+1)\right).$$

Proof. TOPROVE 12 □

The first part of Theorem 13 is now directly implied by Lemmas 31 and 33.

3.5.1 Improved Lightness Bounds in Polynomial Time via Multiple Host Trees

Of the two results from Section 3.4, only one of them extends readily to the polynomial time algorithm. The one that does *not* seem to extend is Theorem 29, which improves the f-dependence for $2f$-competitive lightness to $f^{1/2}$. The issue is roughly that the proof samples heavy and light edges with different probabilities, but in our polynomial time algorithm we commit to a sampling probability at runtime, meaning that we cannot easily apply different runtimes to different edge types. However, Theorem 28 extends fairly straightforwardly, in order to prove the corresponding part of Theorem 13.

To prove this we will use Algorithm 3. Most of our analysis of Algorithm 3 is the same as that for Algorithm 2, and we will not repeat it here. For example, Lemma 30 (asserting that $\widehat{P}^T_{(u,v)} \approx P^T_{(u,v)}$ for all $T, (u, v)$) still holds with exactly the same proof. Lemma 31 (asserting that the output spanner is correct with high probability) also still holds by essentially the same proof, but we note that for any edge e, there will be at least $(2 + \eta)f + 1 - 2f = \eta f + 1$ trees that are disjoint from the blocked edges e' paired with e. All such trees will vote for (u, v) (i.e. we will increment the "votes" variable when these trees are considered), and so (u, v) will indeed be added to the spanner if an appropriate fault set F exists.

The part that changes is the hosting of edges in trees: instead of an edge $(u, v) \in E$ being hosted by the *first* tree T that caused the edge to be added to the spanner, it is instead hosted by *all* $\Omega(\eta f)$ trees that voted for it. Under this new hosting strategy, the proof of Lemma 32 (controlling the weight of each host graph $H[T]$) still holds with no significant changes. But the final calculation in Lemma 33 now admits an optimization. We calculate:

$$\begin{aligned}
\ell_{2f}(H \mid G) &\leq 2 \cdot \frac{w(H)}{w(Q)} && (Q \text{ is 2-approx min weight preserver})\\
&\leq 2 \cdot \frac{(\eta f)^{-1} \sum\limits_{T \in \mathcal{T}} w(H[T])}{w(Q)} && (\text{every } e \in E(H) \text{ in at least } \Omega(\eta f) \text{ graphs } H[T])\\
&\leq 4 \cdot \frac{(\eta f)^{-1} \sum\limits_{T \in \mathcal{T}} w(H[T])}{\sum\limits_{T \in \mathcal{T}} w(T)} && (\text{Corollary 25 and def of } \mathcal{T})
\end{aligned}$$

```
Input: Graph $G$, stretch $k$, fault tolerance $f$, parameter $\eta > 0$

Let $Q \leftarrow$ 2-approximate min-weight $(2+\eta)f$-FT connectivity preserver of $G$ // polytime by
    [DKK22]
Let $\mathcal{T} \leftarrow$ tree decomposition of $Q$ satisfying Corollary 25 w.r.t. connectivity parameter $(2+\eta)f+1$
Let $E \leftarrow \emptyset$ be the initial set of non-$Q$ spanner edges

// $c$ large enough constant
foreach edge $(u,v) \in E(G) \setminus Q$ in order of nondecreasing weight do
    votes $\leftarrow 0$
    foreach tree $T \in \mathcal{T}$ that spans the $(2+\eta)f+1$-connected component of $Q$ containing $u,v$ do
        sample $c \log n$ subgraphs by including $T$, and including each edge in $E$ with probability $1/f$
        let $\widehat{P}^{T}_{(u,v)}$ be the fraction of sampled subgraphs $H'$ in which $\texttt{dist}_{H'}(u,v) > k \cdot w(u,v)$
        if $\widehat{P}^{T}_{(u,v)} \geq 3/8$ then
            votes $\leftarrow$ votes $+1$
    if votes $\geq \eta f + 1$ then
        add $(u,v)$ to $E$
Return $H = Q \cup E$
```

Algorithm 3: FT spanners in polynomial time with competition parameter $(2+\eta)f$

$$\leq O\left(\frac{(\eta f)^{-1} \sum\limits_{T \in \mathcal{T}} f \cdot \lambda(n,k+1) \cdot w(T)}{\sum\limits_{T \in \mathcal{T}} w(T)}\right) \qquad \text{(Lemma 32)}$$

$$= O\big((\eta^{-1} \cdot \lambda(n,k+1)\big).$$

4 Lower Bounds

In this section we prove our lower bounds. We begin with our main lower bound, showing that if we want lightness bounds similar to the non-fault tolerant setting (e.g., dependence of $n^{1/k}$ for a $(1+\varepsilon)\cdot(2k-1)$-spanner, or even anything below n/k) then we need the competition parameter to be at least $2f$. We then give lower bounds for $2f$-competitive lightness.

4.1 Competition Parameters at most $2f-1$

We now want to prove Theorem 9, which essentially rules out sublinear $2f-1$-competitive lightness (for small f,k):

Theorem 9. *For any $f,k \geq 1$, there is a family of n-node weighted graphs G for which every f-EFT k-spanner H has competitive lightness*

$$\ell_f(H \mid G) \geq \Omega\left(\frac{n}{f^2 k}\right).$$

Before proving our main lower bound, we start with a warmup in the setting of a single fault to show a lightness bound of $\Omega(n/k)$. Let G be a graph of $2n$ vertices v_0, v_1, $\ldots$, v_{2n-1} with unit weight edges from v_i to v_{i+1} (mod $2n$) and additional edges of weight $\frac{2n-2}{k} - \epsilon$ from v_{2j} to v_{2j+2}. Clearly $\texttt{mst}(G)$ is the cycle of unit weight edges (minus one arbitrary edge), and so has weight $2n-1$. On the other hand, we claim that any 1-EFT k-spanner H must include all of the additional edges. To see this, suppose that H does not include one of the additional edges, say $\{v_{2j}, v_{2j+2}\}$. Consider the fault set $F = \{\{v_{2j}, v_{2j+1}\}\}$. Then $\texttt{dist}_{G \setminus F}(v_{2j}, v_{2j+2}) = \frac{2n-2}{k} - \epsilon$, via the edge $\{v_{2j}, v_{2j+2}\}$. But in $H \setminus F$, the only way to get from v_{2j} to v_{2j+2} is to go all the way around the cycle, for a total distance of $2n-2$. So then the stretch is $(2n-2)/\left(\frac{2n-2}{k} - \epsilon\right) > k$, and thus H is not a 1-EFT k-spanner.

Thus H has all of the additional edges, and so has total weight at least $n \cdot \left(\frac{2n-2}{k} - \epsilon\right) = \Omega(n^2/k)$. Thus $\ell_1(H \mid G) \geq \Omega(n/k)$.

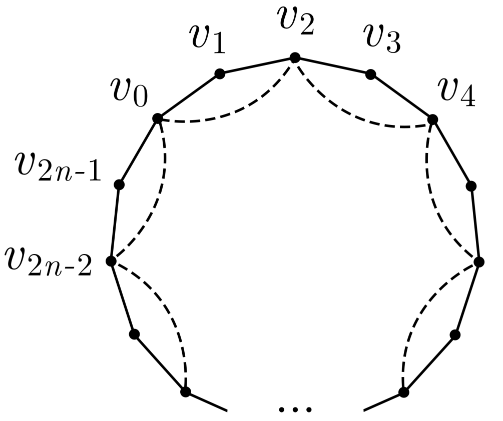

Figure 2: Outer (solid) cycle edges have unit weight, dotted edges have weight $\frac{n-2}{k} - \epsilon$

To extend this to larger faults (and thus prove Theorem 9). we modify the above argument by replacing each of the v_{2j+1} nodes (the odd indices) by a "cloud" of f nodes.

Proof. TOPROVE 13 □

4.2 $2f$- and $2(1+\eta)f$-Competitive Lightness

We now prove the lower bound parts of Theorem 11 and Theorem 12. We in fact prove a stronger statement which directly implies both of them.

Theorem 34. *For any constant $c \geq 2$, there is an infinite family of n-node graphs G for which every f-EFT k-spanner H has competitive lightness*

$$\ell_{cf}(H \mid G) \geq \Omega\left(\lambda\left(\frac{n}{(cf)^{1/2}}, k+1\right)\right).$$

Proof. TOPROVE 14 □

5 Future Directions: Light Vertex Fault Tolerant Spanners

In addition to the obvious open questions of a) determining the precise bound on the achievable $2f$-competitive lightness, and b) providing simultaneous bounds on both the lightness and the sparsity, an interesting open problem left by this paper is to determine the appropriate competition parameter for light *vertex* fault tolerant (VFT) spanners. We can define vertex competitive lightness analogously to Definition 8, but with respect to the set of f-VFT connectivity preservers rather than f-EFT connectivity preservers. We then ask:

Question 35. What is the smallest function $\mathcal{C}(f)$ such that every n-node weighted graph G has an f-VFT k-spanner with **vertex**-competitive lightness[7]

$$\ell_{\mathcal{C}(f)}(H \mid G) \leq O\left(\texttt{poly}(f) \cdot \lambda(n, k+1)\right)?$$

Theorem 10 proves that the right answer in the edge-fault setting is exactly $\mathcal{C}(f) = 2f$, and it is not hard to see that the lower bound extends to show that the answer is $\mathcal{C}(f) \geq 2f$ for vertex faults. But it is unclear whether a matching upper bound is possible. Below, we discuss a few leads on this problem, as well as the various technical difficulties in bringing them to fruition.

[7] It would also be interesting if, instead of a λ reduction, one could match the state-of-the art non-faulty size/stretch tradeoff up to a $\texttt{poly}(f)$ factor. That is, the goal would be to obtain stretch $(1+\varepsilon)(2k-1)$ and lightness $O\left(\texttt{poly}(f, \varepsilon^{-1}) \cdot n^{1+1/k}\right)$.

Connected Dominating Sets. Censor-Hillel, Ghaffari, and Kuhn [CHGK14] proposed that *connected dominating sets* are the natural analog of spanning trees when considering vertex connectivity rather than edge connectivity, and they proved the following analog of the Nash-Williams theorem:

Theorem 36 ([CHGK14])**.** *Every f-vertex-connected n-node graph contains a collection of $\Omega(f/\log n)$ vertex-disjoint CDSes.*

So a natural approach to getting vertex fault tolerant light spanners is to replace the use of spanning trees and the Nash-Williams Theorem in our EFT result with CDSes and Theorem 36. While there are some technical issues, this approach basically works, but with two very large caveats. First, we only get $\Theta(f \log n)$-competitive lightness, rather than $2f$-competitive lightness. Second, recall that our EFT approach actually needed an *extension* of the Nash-Williams Theorem due to Chekuri and Shepherd [CS09] which allowed for the underlying graph to be less than f connected. If we use Theorem 36 directly, rather than proving a similar extension, we would require that the underlying graph be at least $\Omega(f \log n)$-vertex connected. Since we do not even want to assume f-connectivity, much less $\Omega(f \log n)$-connectivity, this is an important caveat. However, while it is possible that one could prove an extension of Theorem 36 which does not require high underlying connectivity, even then we would be limited to $\Omega(f \log n)$-competitive lightness.

Independent Spanning Trees. To overcome the fact that the CDS approach can only give a fairly large competition parameter of $\Theta(f \log n)$, we must consider other objects that guarantee many paths that do not overlap too much. Another object that would suffice are *independent spanning trees*, defined as follows:

Definition 37. Let r be a vertex of graph $G = (V, E)$. A collection of spanning trees $T_1, \ldots, T_k$ of G are *independent with root* r if for each vertex $v \in V$, the paths from v to r in $T_1, \ldots, T_k$ are pairwise vertex disjoint.

It was conjectured by [IR88] that every k-vertex connected graph contains a collection of k independent spanning trees for every root r. They proved this conjecture for $k = 2$, and it has subsequently been proved for $k = 3$ [SS19], $k = 4$ [HT18], and $k = 5$ [AL23], but it is still open for all $k \geq 6$. While it is not nearly as simple as the CDS case (mostly due to the fact that the choice of root implies that paths actually *do* overlap at the root), it turns out that one can similarly show that these objects can be used inside of our framework. However, this suffers again from two major drawbacks. First, this is still a conjecture. It could very well be false, and even if true, it has been open for over 35 years with very little progress. Second, it would still require that the underlying graph be $2f$-vertex connected; it would provide no guarantees for the more general case (which we can do in the EFT setting).

References

[ABP91] Baruch Awerbuch, Alan Baratz, and David Peleg. Efficient broadcast and light-weight spanners. *Unpublished manuscript, November*, 1991.

[ADD$^+$93] Ingo Althöfer, Gautam Das, David Dobkin, Deborah Joseph, and José Soares. On sparse spanners of weighted graphs. *Discrete & Computational Geometry*, 9(1):81–100, 1993.

[AL23] Alonso Ali and Orlando Lee. Five edge-independent spanning trees. *Procedia Computer Science*, 223:223–230, 2023. XII Latin-American Algorithms, Graphs and Optimization Symposium (LAGOS 2023).

[AP90] Baruch Awerbuch and David Peleg. Network synchronization with polylogarithmic overhead. In *Foundations of Computer Science, 1990. Proceedings., 31st Annual Symposium on*, pages 514–522. IEEE, 1990.

[BDMS13] Prosenjit Bose, Vida Dujmovic, Pat Morin, and Michiel Smid. Robust geometric spanners. *SIAM Journal on Computing*, 42(4):1720–1736, 2013.

[BDN22] Greg Bodwin, Michael Dinitz, and Yasamin Nazari. Vertex Fault-Tolerant Emulators. In *13th Innovations in Theoretical Computer Science Conference (ITCS 2022)*, pages 25:1–25:22, 2022.

[BDN23] Greg Bodwin, Michael Dinitz, and Yasamin Nazari. Epic Fail: Emulators Can Tolerate Polynomially Many Edge Faults for Free. In *14th Innovations in Theoretical Computer Science Conference (ITCS 2023)*, volume 251, pages 20:1–20:22, 2023.

[BDPW18] Greg Bodwin, Michael Dinitz, Merav Parter, and Virginia Vassilevska Williams. Optimal vertex fault tolerant spanners (for fixed stretch). In *Proceedings of the 29th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 1884–1900. Society for Industrial and Applied Mathematics, 2018.

[BDR21] Greg Bodwin, Michael Dinitz, and Caleb Robelle. Optimal vertex fault-tolerant spanners in polynomial time. In *Proceedings of the 2021 ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 2924–2938. SIAM, 2021.

[BDR22] Greg Bodwin, Michael Dinitz, and Caleb Robelle. Partially optimal edge fault-tolerant spanners. In *Proceedings of the 2022 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 3272–3286. SIAM, 2022.

[BEH$^+$10] Georg Baier, Thomas Erlebach, Alexander Hall, Ekkehard Köhler, Petr Kolman, Ondřej Pangrác, Heiko Schilling, and Martin Skutella. Length-bounded cuts and flows. *ACM Transactions on Algorithms (TALG)*, 7(1):1–27, 2010.

[BF25] Greg Bodwin and Jeremy Flics. A lower bound for light spanners in general graphs. In *Proceedings of the 2025 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 4327–4337. SIAM, 2025.

[BHP24] Greg Bodwin, Bernhard Haeupler, and Merav Parter. Fault-tolerant spanners against bounded-degree edge failures: Linearly more faults, almost for free. In *Proceedings of the 2024 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 2609–2642. SIAM, 2024.

[BHPO20] Kevin Buchin, Sariel Har-Peled, and Dániel Oláh. A spanner for the day after. *Discrete & Computational Geometry*, 64(4):1167–1191, 2020.

[Bod25] Greg Bodwin. An alternate proof of near-optimal light spanners. *TheoretiCS*, 4, 2025.

[BP19] Greg Bodwin and Shyamal Patel. A trivial yet optimal solution to vertex fault tolerant spanners. In *Proceedings of the 2019 ACM Symposium on Principles of Distributed Computing*, pages 541–543, 2019.

[CDNS92] Barun Chandra, Gautam Das, Giri Narasimhan, and José Soares. New sparseness results on graph spanners. In *Proceedings of the eighth annual symposium on Computational geometry*, pages 192–201. ACM, 1992.

[CHGK14] Keren Censor-Hillel, Mohsen Ghaffari, and Fabian Kuhn. A new perspective on vertex connectivity. In *Proceedings of the twenty-fifth annual ACM-SIAM symposium on Discrete algorithms*, pages 546–561. SIAM, 2014.

[CLN15] T-H Hubert Chan, Mingfei Li, and Li Ning. Sparse fault-tolerant spanners for doubling metrics with bounded hop-diameter or degree. *Algorithmica*, 71(1):53–65, 2015.

[CLNS15] T-H Hubert Chan, Mingfei Li, Li Ning, and Shay Solomon. New doubling spanners: Better and simpler. *SIAM Journal on Computing*, 44(1):37–53, 2015.

[CLPR10] Shiri Chechik, Michael Langberg, David Peleg, and Liam Roditty. Fault tolerant spanners for general graphs. *SIAM Journal on Computing*, 39(7):3403–3423, 2010.

[CS09] Chandra Chekuri and F Bruce Shepherd. Approximate integer decompositions for undirected network design problems. *SIAM Journal on Discrete Mathematics*, 23(1):163–177, 2009.

[CT00] Joseph Cheriyan and Ramakrishna Thurimella. Approximating minimum-size k-connected spanning subgraphs via matching. *SIAM Journal on Computing*, 30(2):528–560, 2000.

[CWN16] Shiri Chechik and Christian Wulff-Nilsen. Near-optimal light spanners. In *Proceedings of the twenty-seventh annual ACM-SIAM symposium on Discrete algorithms*, pages 883–892. Society for Industrial and Applied Mathematics, 2016.

[CZ04] Artur Czumaj and Hairong Zhao. Fault-tolerant geometric spanners. *Discrete & Computational Geometry*, 32(2):207–230, 2004.

[DK11] Michael Dinitz and Robert Krauthgamer. Fault-tolerant spanners: better and simpler. In *Proceedings of the 30th annual ACM SIGACT-SIGOPS symposium on Principles of distributed computing*, pages 169–178. ACM, 2011.

[DKK22] Michael Dinitz, Ama Koranteng, and Guy Kortsarz. Relative Survivable Network Design. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques (APPROX/RANDOM 2022)*, pages 41:1–41:19, 2022.

[DKKN23] Michael Dinitz, Ama Koranteng, Guy Kortsarz, and Zeev Nutov. Improved approximations for relative survivable network design. *CoRR*, abs/2304.06656, 2023.

[DMP16] Matt DeVos, Jessica McDonald, and Irene Pivotto. Packing steiner trees. *Journal of Combinatorial Theory, Series B*, 119:178–213, 2016.

[DR20] Michael Dinitz and Caleb Robelle. Efficient and simple algorithms for fault-tolerant spanners. In *Proceedings of the 39th Symposium on Principles of Distributed Computing*, pages 493–500, 2020.

[EEST08] Michael Elkin, Yuval Emek, Daniel A Spielman, and Shang-Hua Teng. Lower-stretch spanning trees. *SIAM Journal on Computing*, 38(2):608–628, 2008.

[ENS14] Michael Elkin, Ofer Neiman, and Shay Solomon. Light spanners. In *International Colloquium on Automata, Languages, and Programming*, pages 442–452. Springer, 2014.

[Erd63] Paul Erdős. Extremal problems in graph theory. In *Proceedings of the Symposium on Theory of Graphs and its Applications*, page 2936, 1963.

[GG12] Harold N. Gabow and Suzanne R. Gallagher. Iterated rounding algorithms for the smallest k-edge connected spanning subgraph. *SIAM Journal on Computing*, 41(1):61–103, 2012.

[GGTW09] Harold N. Gabow, Michel X. Goemans, Éva Tardos, and David P. Williamson. Approximating the smallest k-edge connected spanning subgraph by lp-rounding. *Networks*, 53(4):345–357, 2009.

[HT18] Alexander Hoyer and Robin Thomas. Four edge-independent spanning trees. *SIAM Journal on Discrete Mathematics*, 32(1):233–248, 2018.

[IR88] Alon Itai and Michael Rodeh. The multi-tree approach to reliability in distributed networks. *Inf. Comput.*, 79(1):43–59, October 1988.

[Jai01] Kamal Jain. A factor 2 approximation algorithm for the generalized steiner network problem. *Combinatorica*, 21(1):39–60, 2001.

[Kri03] Matthias Kriesell. Edge-disjoint trees containing some given vertices in a graph. *Journal of Combinatorial Theory, Series B*, 88(1):53–65, 2003.

[Lau05] Lap Chi Lau. Packing steiner forests. In *International Conference on Integer Programming and Combinatorial Optimization*, pages 362–376. Springer, 2005.

[Le24] Hung Le. Recent Progress on Euclidean (and Related) Spanners , 2024. Workshop on Sublinear Graph Simplification, Simons Institute for the Theory of Computing.

[LNS98] Christos Levcopoulos, Giri Narasimhan, and Michiel Smid. Efficient algorithms for constructing fault-tolerant geometric spanners. In *Proceedings of the thirtieth annual ACM symposium on Theory of computing*, pages 186–195. ACM, 1998.

[LS23] Hung Le and Shay Solomon. A unified framework for light spanners. In *Proceedings of the 55th Annual ACM SIGACT Symposium on Theory of Computing (STOC)*. ACM, 2023.

[LST23] Hung Le, Shay Solomon, and Cuong Than. Optimal fault-tolerant spanners in euclidean and doubling metrics: Breaking the Ω (log n) lightness barrier. In *64th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2023, Santa Cruz, CA, USA, November 6-9, 2023*, pages 77–97. IEEE, 2023.

[Luk99] Tamás Lukovszki. New results on fault tolerant geometric spanners. In *Workshop on Algorithms and Data Structures*, pages 193–204. Springer, 1999.

[NI92] Hiroshi Nagamochi and Toshihide Ibaraki. A linear-time algorithm for finding a sparse k-connected spanning subgraph of a k-connected graph. *Algorithmica*, 7(5&6):583–596, 1992.

[NI08] Hiroshi Nagamochi and Toshihide Ibaraki. *Algorithmic Aspects of Graph Connectivity*, volume 123 of *Encyclopedia of Mathematics and its Applications*. Cambridge University Press, 2008.

[NS07] Giri Narasimhan and Michiel Smid. *Geometric Spanner Networks*. Cambridge University Press, 2007.

[NW61] C. St.J. A. Nash-Williams. Edge-Disjoint Spanning Trees of Finite Graphs. *Journal of the London Mathematical Society*, s1-36(1):445–450, 01 1961.

[Par22] Merav Parter. Nearly optimal vertex fault-tolerant spanners in optimal time: Sequential, distributed and parallel. In *Proceedings of the 54th Annual ACM SIGACT Symposium on Theory of Computing (STOC)*. ACM, 2022.

[Pel00] David Peleg. *Distributed computing: a locality-sensitive approach*. SIAM, 2000.

[PS89] David Peleg and Alejandro A. Schäffer. Graph spanners. *J. Graph Theory*, 13(1):99–116, 1989.

[PST24] Asaf Petruschka, Shay Sapir, and Elad Tzalik. Color Fault-Tolerant Spanners. In *15th Innovations in Theoretical Computer Science Conference (ITCS 2024)*, pages 88:1–88:17, 2024.

[PU89a] David Peleg and Jeffrey Ullman. An optimal synchronizer for the hypercube. *SIAM Journal on Computing (SICOMP)*, 18(4):740—-747, 1989.

[PU89b] David Peleg and Eli Upfal. A trade-off between space and efficiency for routing tables. *Journal of the ACM (JACM)*, 36(3):510–530, 1989.

[Sol14] Shay Solomon. From hierarchical partitions to hierarchical covers: Optimal fault-tolerant spanners for doubling metrics. In *Proceedings of the forty-sixth annual ACM symposium on Theory of computing*, pages 363–372. ACM, 2014.

[SS19] Lena Schlipf and Jens M Schmidt. Edge-orders. *Algorithmica*, 81:1881–1900, 2019.

[TZ05] Mikkel Thorup and Uri Zwick. Approximate distance oracles. *Journal of the ACM (JACM)*, 52(1):1–24, 2005.