

Light Edge Fault Tolerant Graph Spanners

Greg Bodwin 

University of Michigan, Ann Arbor, MI, USA

Michael Dinitz 

Johns Hopkins University, Baltimore, MD, USA

Ama Koranteng 

Johns Hopkins University, Baltimore, MD, USA

Lily Wang 

University of Michigan, Ann Arbor, MI, USA

Abstract

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 metric settings, 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 if we use the traditional definition (normalizing by the MST). We then argue that a natural definition of fault-tolerant lightness 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 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: $f' = 2f$. In particular, we show that the 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 if one normalizes by the min-weight $2f$ -EFT connectivity preserver, then we show that the lightness is bounded by just $O(f^{1/2})$ times the non-fault tolerant lightness (roughly $n^{1/k}$ for a $(1 + \varepsilon)(2k - 1)$ -spanner).

2012 ACM Subject Classification Theory of computation \rightarrow Sparsification and spanners

Keywords and phrases Fault Tolerant Spanners, Light Spanners

Digital Object Identifier 10.4230/LIPIcs.ICALP.2025.32

Category Track A: Algorithms, Complexity and Games

Related Version Full Version: <https://arxiv.org/abs/2502.10890> [6]

Funding Greg Bodwin: Supported by NSF:AF 2153680.

Michael Dinitz: Supported in part by NSF award 2228995.

Ama Koranteng: Supported in part by an NSF Graduate Research Fellowship.

Lily Wang: Supported by NSF:AF 2153680.

1 Introduction

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



© Greg Bodwin, Michael Dinitz, Ama Koranteng, and Lily Wang;
licensed under Creative Commons License CC-BY 4.0

52nd International Colloquium on Automata, Languages, and Programming (ICALP 2025).

Editors: Keren Censor-Hillel, Fabrizio Grandoni, Joël Ouaknine, and Gabriele Puppis

Article No. 32; pp. 32:1–32:20



Leibniz International Proceedings in Informatics

LIPICs Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany



► **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 $\text{dist}_H(u, v) \leq k \cdot \text{dist}_G(u, v)$ for all $u, v \in V$.*

The value k is called the *stretch* of the spanner. Spanners were introduced by Peleg and Ullman [50] and Peleg and Schäffer [49] in the context of distributed computing. Since then, they have found numerous applications, ranging from traditional distributed systems and networking settings [51, 3, 2, 48], to efficient data structures like distance oracles [54], to preconditioning of linear systems [30], 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. [1]:

► **Theorem 2** ([1]). *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 [32]).*

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 (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*

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

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 [1, 19, 31, 21, 39, 12, 5], and reasonably tight bounds are known:

► **Theorem 4** ([39, 5]). *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 [32] (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 Levkopoulos,

Narasimhan, and Smid [41] and then (more generally) the setting of doubling metrics; see, e.g., [41, 42, 24, 45, 16, 53, 15, 17, 18, 40] and references within. In particular, the tradeoffs between the stretch and both the sparsity [41, 42, 24, 17, 18, 40] and the lightness [24, 53, 18, 40] for geometric spanners, including in the fault tolerant setting, have been the subject of significant study and interest, and optimal bounds are now known [24, 40].

The study of fault-tolerant spanners in general graphs was initiated in a seminal paper by Chechik, Langberg, Peleg, and Roditty [20]. They introduced the following definition.

► **Definition 5** (Edge Fault Tolerant Spanners [20]). *A subgraph H is an f -edge fault tolerant (f -EFT) k -spanner of $G = (V, E)$ if*

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

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. [20] were able to obtain the following strong upper bound.

► **Theorem 6** ([20]). *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 4). 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 [20, 28, 9, 14, 29, 10, 11, 47, 13, 52], which gradually improved on the pioneering f -dependencies of [20] in both settings. This has culminated in optimal sparsity bounds for VFT spanners [14, 10], near-optimal sparsity bounds for EFT spanners [11], and efficient construction algorithms [29, 10, 47].

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 roughly¹

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

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 [38].

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

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.

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 on three nodes u, v, w , with $w(u, v) = w(u, w) = 1$ and $w(v, w) = W$. Then $\text{mst}(G)$ 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 $\frac{w(H)}{\text{mst}(G)} \geq \Omega(W)$, where W may be selected as large as we like, demonstrating that this notion of lightness is unbounded.

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 [43, 44, 26, 27]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 $\ell_f(H) := \ell_f(H \mid H)$.

² An f -EFT connectivity preserver is also sometimes called an $f + 1$ -connectivity certificate [43, 44]. This other terminology is more commonly used when the focus is on sparsity or computation time, rather than weights.

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

(See the full version [6] 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.

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 graph⁴ G has an f -EFT $(1 + \varepsilon)(2k - 1)$ -spanner H with $2f$ -competitive lightness*

$$\ell_{2f}(H \mid G) \leq \text{poly}(f, \varepsilon) \cdot n^{1/k}.$$

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

$$\ell_{2f-1}(H \mid G) \geq \text{poly}(f, k) \cdot n.$$

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 [46], 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 [20]: what is the right dependence on f in the $2f$ -competitive lightness? We know from Theorem 10 that $\text{poly}(f)$ is an upper bound, but can we be more precise?

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

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

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 [31]. 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 [31] 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*

$$\begin{aligned} \ell_{2f}(H \mid G) &\leq f^{1/2} \cdot O(\lambda(n, (1 + \varepsilon)2k)) \\ &\leq f^{1/2} \cdot O_\varepsilon(n^{1/k}). \end{aligned} \quad (\text{by } \lambda \text{ upper bounds from [39, 5]})$$

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

$$\begin{aligned} \ell_{2f}(H \mid G) &\geq \Omega\left(\lambda\left(\frac{n}{(2f)^{1/2}}, (1 + \varepsilon)2k\right)\right) \\ &\geq f^{-\frac{1}{2k}} \cdot \Omega(n^{1/k}) \end{aligned} \quad (\text{assuming the girth conjecture [32].})$$

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*

$$\begin{aligned} \ell_{(2+\eta)f}(H \mid G) &\leq O_\eta(1) \cdot \lambda(n, (1 + \varepsilon)2k) \\ &\leq O(1) \cdot O_{\varepsilon, \eta}(n^{1/k}). \end{aligned} \quad (\text{by } \lambda \text{ upper bounds from [39, 5]})$$

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

$$\begin{aligned} \ell_{(2+\eta)f}(H \mid G) &\geq \Omega\left(\lambda\left(\frac{n}{((2 + \eta)f)^{1/2}}, (1 + \varepsilon)2k\right)\right) \\ &\geq f^{-\frac{1}{2k}} \cdot \Omega(n^{1/k}) \end{aligned} \quad (\text{assuming the girth conjecture [32].})$$

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 $\text{poly}(f)$ 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 [9, 14] (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 $\text{dist}_{H \setminus F}(u, v) > k \cdot w(u, v)$.

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 [23, 34, 33]). Thanks to previous work, it is straightforward to handle this issue: 2-approximations for min-weight f -EFT connectivity preserver have recently been developed [26, 27], building off of the seminal work on Survivable Network Design by Jain [35]. 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 [4]. In prior work on spanner *sparsity*, there have been two strategies to address this: a simple one with a minor penalty to sparsity [29], or a complex one with no penalty to sparsity [10]. 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 the full version [6] 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*

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

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), as well as our lower bounds, can be found in the full version [6].

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) [1]. 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 [9, 14, 11], and to the lightness of *non-fault tolerant* spanners [31]. 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 $\text{dist}_{H \setminus F_e}(u, v) > k \cdot \text{dist}_{G \setminus F_e}(u, v)$. 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 [31]. Standard arguments (first introduced by [14]) 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 [31], 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 [31], 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 [46], 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 $f/16$ [37]; improving the constant in Steiner forest packing is a fascinating open problem, and the correct bound is not even known for Steiner trees [36, 25].

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 [22] 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 [22] 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 [26, 27]. 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 [9], and eventually two different methods were developed to design polytime versions of the algorithm:

1. The first approach, introduced by [29] 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, [29] showed that this incurs only a relatively small extra loss in the sparsity. This approach has since been used extensively to turn variants of the fault tolerant greedy algorithm into polynomial-time algorithms, albeit with a small extra loss [11, 7, 8, 13, 52].
2. In order to avoid this extra loss, [10] 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. 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 [29].

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 [29] 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 [10] *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 [22] (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 [10] 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 [31] introduced a weighted analog of girth, for use in studying the lightness of (non-faulty) spanners:

► **Definition 14** (Normalized Weight and Weighted Girth [31]). *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** ([31]). *For all positive integers n and all k , every n -node graph G has a k -spanner H of lightness $\ell(H | G) \leq \lambda(n, k + 1)$, 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** ([39, 5, 12]). *For all positive integers n, k and all $\varepsilon > 0$, we have $\lambda(n, (1 + \varepsilon)2k) \leq O(\varepsilon^{-1}n^{1/k})$. When k is a constant and $\varepsilon = \Theta(n^{-\frac{1}{2k-1}})$, we have $\lambda(n, (1 + \varepsilon)2k) \geq \Omega(\varepsilon^{-1/k}n^{1/k})$.*

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 [9] 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.

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 $\ell(H | G) > \lambda(n, k + 1) - \varepsilon$.

■ **Algorithm 1** Light Fault-Tolerant Greedy Spanner Algorithm.

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* $\text{dist}_{H \setminus F}(u, v) > k \cdot w(u, v)$ **then**

 add (u, v) to H

Return H

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

Proof. As is well known in the spanners literature (e.g. [1]), it suffices to verify that for any set F of $|F| \leq f$ edge faults and any edge $(u, v) \in E(G) \setminus (E(H) \cup F)$, we have $\text{dist}_{H \setminus F}(u, v) > k \cdot w(u, v)$. To show this, we first observe that for any such edge we have $(u, v) \notin Q$ (since we add all of Q to H), and so we consider (u, v) at some point in the main for-loop of the algorithm. We choose not to add (u, v) to H , meaning that for all possible fault sets F we have $\text{dist}_{H \setminus F}(u, v) \leq k \cdot w(u, v)$. For the rest of the algorithm we only add edges to F , and since this property is monotonic in H , it will still hold in the final spanner. ◀

Now we turn to analyzing lightness. When $f = 0$, this algorithm produces a spanner H of weighted girth $> k + 1$ [31]. 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., [14, 29, 10, 11, 7, 8, 52, 13]).

► **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 $(e, e') \in B$.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 [14] and [31].

► **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 $w^*(C) \leq k + 1$ 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. In the main part of the construction, whenever we add an edge $(u, v) \in E(G) \setminus Q$ to the spanner H , we do so because there exists a fault set $F_{(u,v)}$ that satisfies the conditional (there might be many possible choices of $F_{(u,v)}$, in which case we fix one arbitrarily). Let

$$B := \{(e, e') \mid e \in E(H) \setminus Q, e' \in F_e\}.$$

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.

Since each $|F_e| \leq f$, we have that B is f -capped. To argue its cycle-blocking properties, let C be a cycle as in the lemma statement, with a heaviest edge $(u, v) \notin Q$. When we consider (u, v) in the main loop of the algorithm, just before (u, v) is added to H , there exists a $u \rightsquigarrow v$ path through the other edges of the cycle, and so

$$\text{dist}_H(u, v) \leq w(C) - w(u, v) \leq (k+1) \cdot w(u, v) - w(u, v) = k \cdot w(u, v).$$

However, in order to satisfy the conditional and include (u, v) in H , we must have that $\text{dist}_{H \setminus F_{(u,v)}}(u, v) > k \cdot w(u, v)$. Thus at least one edge $e \in C \setminus \{(u, v)\}$ must be included in $F_{(u,v)}$. So we have $\{(u, v), e\} \in B$, which blocks C , completing the proof. ◀

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

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 [46]).** *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. Replace every edge of G with two parallel edges, so that it is $2f$ -connected. Then consider any vertex partition \mathcal{P} , and note that (by connectivity) every part must have at least $2f$ outgoing edges. Thus there are at least $f \cdot |\mathcal{P}|$ edges between parts. This is even a bit stronger than the condition needed to apply the Nash-Williams theorem, and so by Theorem 22 we can partition G into f edge-disjoint spanning trees. Since we initially doubled the edges of G , this means that every original edge participates in at most two spanning trees. ◀

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** ([22], 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 $\text{poly}(n)$ 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. Identical to the previous corollary. Note that when we double edges, every node has even degree, so we guarantee that the graph is Eulerian and so can apply Theorem 24.⁷ ◀

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 $T \setminus \text{mst}(H''[T])$.

The important properties of these subgraphs are:

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

Proof. Let C be any cycle in $H[T]$ of normalized weight $\leq k + 1$. We will argue that C does not survive in $H'''[T]$. There are two cases, depending on whether C has a heaviest edge $(u, v) \notin T$, or whether all heaviest edges in C are in T .

In the first case, suppose that there is a heaviest edge $(u, v) \in C, (u, v) \notin T$. By the properties of the blocking set B from Lemma 20, there is $(e, e') \in B$ with $e, e' \in C$ and $e \notin T$. It is not possible for both edges e, e' to survive in $H''[T]$, since if they are both selected to remain in $H'[T]$ then we will choose to delete e when we construct $H''[T]$. Thus C does not survive in $H''[T]$.

⁷ Technically this edge-doubling step makes G a multigraph, which is not explicitly mentioned in the Chekuri-Shepherd theorem [22]. However, their proof extends straightforwardly to multigraphs.

In the second case, suppose that all heaviest edges in C are also in Q . Suppose that all edges in C survive in $H''[T]$. Then at least one of the heaviest edges $(u, v) \in C$ will not be in $\text{mst}(H''[T])$ (since otherwise we could exchange it for a lighter edge on C). So we will remove (u, v) when we move from $H''[T]$ to $H'''[T]$. \blacktriangleleft

► **Lemma 27.** *Every graph $H'''[T]$ has expected weight $\mathbb{E}[w(H'''[T])] \geq \Omega\left(\frac{w(H[T])}{f}\right) - w(T)$.*

Proof. Every edge in T is included in $H''[T]$ deterministically. Every edge $e \in E(H[T]) \setminus T$ survives in $H''[T]$ iff the following two independent events both occur:

- e is selected to be included in $H'[T]$, which happens with probability $1/f$, and
- For all pairs $(e, e') \in B$ with e first, the other edge e' is *not* selected to be included in $H'[T]$. Since B is f -capped, there are at most f such edges e' to consider. Additionally, since T hosts e , none of these edges e' are in T itself. Thus we select them each to be included in $H'[T]$ independently with probability $1/f$. Since we have f independent events that each occur with probability $1 - 1/f$, the probability that none of them occur is at least $1/4$.

Thus each edge $e \in E(H[T])$ survives in $H''[T]$ with probability $\Omega(1/f)$, implying that $\mathbb{E}[w(H''[T])] \geq \Omega\left(\frac{w(H[T])}{f}\right)$. Finally, when we move from $H''[T]$ to $H'''[T]$, we can only possibly delete edges in T and so we delete at most $w(T)$ weight, implying the lemma. \blacktriangleleft

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 $w(H[T]) \leq O(f) \cdot (w(H'''[T]) + w(T))$. Using this, we observe that for all trees $T \in \mathcal{T}$, we have

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

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:

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

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. Follow the previous analysis, but note that our guarantee is now that we have a set of trees \mathcal{T} such that (1) every edge is in at most two trees, and (2) for all $(u, v) \in E(G) \setminus Q$, there are $(2 + \eta)f + 1$ trees in \mathcal{T} that span the $(2 + \eta)f + 1$ -connected component that contains u, v . Thus, letting F be a fault set that caused us to add (u, v) to the spanner, there are $\Theta(\eta f)$ of these trees that are disjoint from F . We let **each** of these trees host the edge (u, v) . In particular, this means that (u, v) will appear in $\Theta(\eta f)$ many $H[T]$ subgraphs. From there the analysis continues as before, but our final calculation is:

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

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. Let us say that an edge $e \in E(H) \setminus Q$ is Q -heavy if there are at least $f - f^{1/2}$ different edges $e' \in Q$ with $(e, e') \in B$. Otherwise, we say that e is Q -light. Let $H_{\text{heavy}} \subseteq H$ be the subgraph that includes Q and all edges in $E(H) \setminus Q$ are Q -heavy, and similarly let $H_{\text{light}} \subseteq H$ be the subgraph that includes Q and all edges in $E(H) \setminus Q$ that are Q -light. Notice that

$$\begin{aligned} \ell_{2f}(H) &= \frac{w(H \setminus Q) + w(Q)}{w(Q)} = \frac{w(H_{\text{heavy}} \setminus Q) + w(H_{\text{light}} \setminus Q) + w(Q)}{w(Q)} \\ &= \frac{w(H_{\text{heavy}})}{w(Q)} + \frac{w(H_{\text{light}})}{w(Q)} - 1 < \ell_{2f}(H_{\text{heavy}}) + \ell_{2f}(H_{\text{light}}). \end{aligned}$$

Thus it suffices to individually bound $\ell_{2f}(H_{\text{heavy}}), \ell_{2f}(H_{\text{light}})$.

Bounding $\ell_{2f}(H_{\text{heavy}})$. Follow the same argument as before, but observe that any Q -heavy edge e can be blocked with at most $f^{1/2}$ other edges e' in its relevant subgraph $H[T]$. Thus we may generate $H'[T]$ from $H[T]$ by keeping all edges in T deterministically (as before),

but by sampling the edges in $H[T] \setminus T$ with probability only $1/f^{1/2}$ instead of $1/f$. This leads to a new bound of

$$\mathbb{E}[w(H''[T])] \geq \Omega\left(\frac{w(H[T])}{f^{1/2}}\right),$$

and following the same analysis from there, this improved factor of $f^{1/2}$ propagates into the final lightness bound.

Bounding $\ell_{2f}(H_{light})$. Follow the same argument as before, but observe that for any Q -light edge e , there are at least

$$2f + 1 - 2(f - f^{1/2}) \geq f^{1/2} + 1$$

different trees that span the $(2f + 1)$ -connected component containing the endpoints of e , and which do not contain edges e' with $(e, e') \in B$. We let e be hosted by **all** such trees. We then continue the analysis as before, with a slight change only in the final calculation, much like the previous theorem. The calculation is now:

$$\begin{aligned} \ell_{2f}(H_{light}) &= \frac{w(H_{light})}{w(Q)} \leq O\left(\frac{f^{-1/2} \cdot \sum_{T \in \mathcal{T}} w(H[T])}{\sum_{T \in \mathcal{T}} w(T)}\right) \\ &\quad \text{(each edge in at least } f^{1/2} \text{ subgraphs } H[T]) \\ &\leq O\left(\frac{f^{-1/2} \sum_{T \in \mathcal{T}} f \cdot \lambda(n, k + 1) \cdot w(T)}{\sum_{T \in \mathcal{T}} w(T)}\right) = O\left(f^{1/2} \cdot \lambda(n, k + 1)\right). \quad \blacktriangleleft \end{aligned}$$

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

Proof of Theorems 11 and 12, Upper Bounds. Theorem 11 is directly implied by Theorem 18 (which shows that H is indeed an f -EFT k -spanner) and Theorem 29 (which gives the claimed lightness bound). For Theorem 12, it is easy to see that the change we made in the algorithm (making Q a $(2 + \eta)f$ -EFT connectivity preserver rather than $2f$) does not affect the proof of Theorem 18 that H is feasible. Hence Theorem 12 is implied by Theorem 18 and Theorem 28. \blacktriangleleft

4 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 30.** *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 $\ell_{\mathcal{C}(f)}(H \mid G) \leq O(\text{poly}(f) \cdot \lambda(n, k + 1))$?*

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. For a discussion of some leads on this problem, as well as the various technical difficulties in bringing them to fruition, see the full version of this paper [6].

References

- 1 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. doi:10.1007/BF02189308.
- 2 Baruch Awerbuch, Alan Baratz, and David Peleg. Efficient broadcast and light-weight spanners. *Unpublished manuscript*, November, 1991.
- 3 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. doi:10.1109/FSCS.1990.89572.
- 4 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. doi:10.1145/1868237.1868241.
- 5 Greg Bodwin. An alternate proof of near-optimal light spanners. *TheoretCS*, 4, 2025. doi:10.46298/THEORETICS.25.2.
- 6 Greg Bodwin, Michael Dinitz, Ama Koranteng, and Lily Wang. Light edge fault tolerant graph spanners, 2025. doi:10.48550/arXiv.2502.10890.
- 7 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. doi:10.4230/LIPICS.ITCS.2022.25.
- 8 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. doi:10.4230/LIPICS.ITCS.2023.20.
- 9 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. doi:10.1137/1.9781611975031.123.
- 10 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. doi:10.1137/1.9781611976465.174.
- 11 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. doi:10.1137/1.9781611977073.129.
- 12 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. doi:10.1137/1.9781611978322.146.
- 13 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. doi:10.1137/1.9781611977912.93.
- 14 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. doi:10.1145/3293611.3331588.
- 15 Prosenjit Bose, Vida Dujmovic, Pat Morin, and Michiel Smid. Robust geometric spanners. *SIAM Journal on Computing*, 42(4):1720–1736, 2013. doi:10.1137/120874473.

- 16 Kevin Buchin, Sarel Har-Peled, and Dániel Oláh. A spanner for the day after. *Discrete & Computational Geometry*, 64(4):1167–1191, 2020. doi:10.1007/S00454-020-00228-6.
- 17 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. doi:10.1007/S00453-013-9779-Y.
- 18 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. doi:10.1137/130930984.
- 19 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. doi:10.1145/142675.142717.
- 20 Shiri Chechik, Michael Langberg, David Peleg, and Liam Roditty. Fault tolerant spanners for general graphs. *SIAM Journal on Computing*, 39(7):3403–3423, 2010. doi:10.1137/090758039.
- 21 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. doi:10.1137/1.9781611974331.CH63.
- 22 Chandra Chekuri and F Bruce Shepherd. Approximate integer decompositions for undirected network design problems. *SIAM Journal on Discrete Mathematics*, 23(1):163–177, 2009. doi:10.1137/040617339.
- 23 Joseph Cheriyan and Ramakrishna Thurimella. Approximating minimum-size k -connected spanning subgraphs via matching. *SIAM Journal on Computing*, 30(2):528–560, 2000. doi:10.1137/S009753979833920X.
- 24 Artur Czumaj and Hairong Zhao. Fault-tolerant geometric spanners. *Discrete & Computational Geometry*, 32(2):207–230, 2004. doi:10.1007/S00454-004-1121-7.
- 25 Matt DeVos, Jessica McDonald, and Irene Pivotto. Packing steiner trees. *Journal of Combinatorial Theory, Series B*, 119:178–213, 2016. doi:10.1016/j.jctb.2016.02.002.
- 26 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. doi:10.4230/LIPICS.APPROX/RANDOM.2022.41.
- 27 Michael Dinitz, Ama Koranteng, Guy Kortsarz, and Zeev Nutov. Improved approximations for relative survivable network design. *CoRR*, abs/2304.06656, 2023. doi:10.48550/arXiv.2304.06656.
- 28 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. doi:10.1145/1993806.1993830.
- 29 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. doi:10.1145/3382734.3405735.
- 30 Michael Elkin, Yuval Emek, Daniel A Spielman, and Shang-Hua Teng. Lower-stretch spanning trees. *SIAM Journal on Computing*, 38(2):608–628, 2008. doi:10.1137/050641661.
- 31 Michael Elkin, Ofer Neiman, and Shay Solomon. Light spanners. In *International Colloquium on Automata, Languages, and Programming*, pages 442–452. Springer, 2014. doi:10.1007/978-3-662-43948-7_37.
- 32 Paul Erdős. Extremal problems in graph theory. In *Proceedings of the Symposium on Theory of Graphs and its Applications*, page 2936, 1963.
- 33 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. doi:10.1137/080732572.
- 34 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. doi:10.1002/NET.20289.

- 35 Kamal Jain. A factor 2 approximation algorithm for the generalized steiner network problem. *Combinatorica*, 21(1):39–60, 2001. doi:10.1007/s004930170004.
- 36 Matthias Kriesell. Edge-disjoint trees containing some given vertices in a graph. *Journal of Combinatorial Theory, Series B*, 88(1):53–65, 2003. doi:10.1016/S0095-8956(02)00013-8.
- 37 Lap Chi Lau. Packing steiner forests. In *International Conference on Integer Programming and Combinatorial Optimization*, pages 362–376. Springer, 2005. doi:10.1007/11496915_27.
- 38 Hung Le. Recent Progress on Euclidean (and Related) Spanners, 2024. Workshop on Sublinear Graph Simplification, Simons Institute for the Theory of Computing. URL: <https://simons.berkeley.edu/talks/hung-le-university-massachusetts-amherst-2024-08-01>.
- 39 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.
- 40 Hung Le, Shay Solomon, and Cuong Than. Optimal fault-tolerant spanners in euclidean and doubling metrics: Breaking the $\Omega(\log n)$ 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. doi:10.1109/FOCS57990.2023.00013.
- 41 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. doi:10.1145/276698.276734.
- 42 Tamás Lukovszki. New results on fault tolerant geometric spanners. In *Workshop on Algorithms and Data Structures*, pages 193–204. Springer, 1999. doi:10.1007/3-540-48447-7_20.
- 43 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. doi:10.1007/BF01758778.
- 44 Hiroshi Nagamochi and Toshihide Ibaraki. *Algorithmic Aspects of Graph Connectivity*, volume 123 of *Encyclopedia of Mathematics and its Applications*. Cambridge University Press, 2008. doi:10.1017/CB09780511721649.
- 45 Giri Narasimhan and Michiel Smid. *Geometric Spanner Networks*. Cambridge University Press, 2007. doi:10.1017/CB09780511546884.
- 46 C. St.J. A. Nash-Williams. Edge-Disjoint Spanning Trees of Finite Graphs. *Journal of the London Mathematical Society*, s1-36(1):445–450, January 1961. doi:10.1112/jlms/s1-36.1.445.
- 47 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.
- 48 David Peleg. *Distributed computing: a locality-sensitive approach*. SIAM, 2000.
- 49 David Peleg and Alejandro A. Schäffer. Graph spanners. *J. Graph Theory*, 13(1):99–116, 1989. doi:10.1002/JGT.3190130114.
- 50 David Peleg and Jeffrey Ullman. An optimal synchronizer for the hypercube. *SIAM Journal on Computing (SICOMP)*, 18(4):740–747, 1989. doi:10.1137/0218050.
- 51 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. doi:10.1145/65950.65953.
- 52 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. doi:10.4230/LIPICS.ITCS.2024.88.
- 53 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. doi:10.1145/2591796.2591864.
- 54 Mikkel Thorup and Uri Zwick. Approximate distance oracles. *Journal of the ACM (JACM)*, 52(1):1–24, 2005. doi:10.1145/1044731.1044732.