Improved Purely Additive Fault-Tolerant Spanners[*]

Davide Bilò[1], Fabrizio Grandoni[2], Luciano Gualà[3],
Stefano Leucci[4], and Guido Proietti[4,5]

[1] Dipartimento di Scienze Umanistiche e Sociali, Università di Sassari, Italy
[2] IDSIA, University of Lugano, Switzerland
[3] Dipartimento di Ingegneria dell'Impresa, Università di Roma "Tor Vergata", Italy
[4] DISIM, Università degli Studi dell'Aquila, Italy
[5] Istituto di Analisi dei Sistemi ed Informatica, CNR, Roma, Italy
E-mail: davide.bilo@uniss.it; fabrizio@idsia.ch; guala@mat.uniroma2.it;
stefano.leucci@univaq.it; guido.proietti@univaq.it

Abstract. Let G be an unweighted n-node undirected graph. A β-*additive spanner* of G is a spanning subgraph H of G such that distances in H are stretched at most by an additive term β w.r.t. the corresponding distances in G. A natural research goal related with spanners is that of designing *sparse* spanners with *low* stretch.

In this paper, we focus on *fault-tolerant* additive spanners, namely additive spanners which are able to preserve their additive stretch even when one edge fails. We are able to improve all known such spanners, in terms of either sparsity or stretch. In particular, we consider the sparsest known spanners with stretch 6, 28, and 38, and reduce the stretch to 4, 10, and 14, respectively (while keeping the same sparsity).

Our results are based on two different constructions. On one hand, we show how to augment (by adding a *small* number of edges) a fault-tolerant additive *sourcewise spanner* (that approximately preserves distances only from a given set of source nodes) into one such spanner that preserves all pairwise distances. On the other hand, we show how to augment some known fault-tolerant additive spanners, based on clustering techniques. This way we decrease the additive stretch without any asymptotic increase in their size. We also obtain improved fault-tolerant additive spanners for the case of one vertex failure, and for the case of f edge failures.

1 Introduction

We are given an unweighted, undirected n-node graph $G = (V(G), E(G))$. Let $d_G(s,t)$ denote the shortest path distance between nodes s and t in G. A *spanner* H of G is a spanning subgraph such that $d_H(s,t) \leq \varphi(d_G(s,t))$ for all $s, t \in V(G)$, where φ is the so-called *stretch* or *distortion* function of the spanner. In particular,

[*] This work was partially supported by the Research Grant PRIN 2010 "ARS TechnoMedia", funded by the Italian Ministry of Education, University, and Research, and by the ERC Starting Grant "New Approaches to Network Design".

when $\varphi(x) = \alpha x + \beta$, for constants α, β, the spanner is named an (α,β) spanner. If $\alpha = 1$, the spanner is called *(purely) additive* or also β-*additive*. If $\beta = 0$, the spanner is called α-*multiplicative*.

Finding *sparse* (i.e., with a small number of edges) spanners is a key task in many network applications, since they allow for a small-size infrastructure onto which an efficient (in terms of paths' length) point-to-point communication can be performed. Due to this important feature, spanners were the subject of an intensive research effort, aiming at designing increasingly sparser spanners with lower stretch.

However, as any sparse structure, a spanner is very sensitive to possible failures of *components* (i.e., edges or nodes), which may drastically affect its performances, or even disconnect it! Thus, to deal with this drawback, a more robust concept of *fault-tolerant* spanner is naturally conceivable, in which the distortion must be guaranteed even after a subset of components of G fails.

More formally, for a subset F of edges (resp., vertices) of G, let $G - F$ be the graph obtained by removing from G the edges (resp., vertices and incident edges) in F. When $F = \{x\}$, we will simply write $G - x$. Then, an f*-edge fault-tolerant* (f-EFT) spanner with distortion (α, β), is a subgraph H of G such that, for every set $F \subseteq E(G)$ of at most f failed edges, we have[6]

$$d_{H-F}(s,t) \leq \alpha \cdot d_{G-F}(s,t) + \beta \quad \forall s,t \in V(G).$$

We define similarly an f*-vertex fault-tolerant* (f-VFT) spanner. For $f = 1$, we simply call the spanner edge/vertex fault-tolerant (EFT/VFT).

Chechik et al. [10] show how to construct a $(2k-1)$-multiplicative f-EFT spanner of size $O(f \cdot n^{1+1/k})$, for any integer $k \geq 1$. Their approach also works for weighted graphs and for vertex-failures, returning a $(2k-1)$-multiplicative f-VFT spanner of size $\widetilde{O}(f^2 \cdot k^{f+1} \cdot n^{1+1/k})$.[7] This latter result has been finally improved through a randomized construction in [14], where the expected size was reduced to $\widetilde{O}(f^{2-1/k} \cdot n^{1+1/k})$. For a comparison, the sparsest known $(2k-1)$-multiplicative *standard* (non fault-tolerant) spanners have size $O(n^{1+\frac{1}{k}})$ [2], and this is believed to be asymptotically tight due to the girth conjecture of Erdős [15].

Additive fault-tolerant spanners can be constructed with the following approach by Braunshvig et al [8]. Let M be an α-multiplicative f-EFT spanner, and A be a β-additive standard spanner. Then $H = M \cup A$ is a $(2f(2\beta + \alpha - 1) + \beta)$-additive f-EFT spanner. One can exploit this approach to construct *concrete* EFT spanners as follows. We know how to construct 6-additive spanners of size $O(n^{4/3})$ [4], randomized spanners that, w.h.p., have size $\widetilde{O}(n^{7/5})$ and additive distortion 4 [9], and 2-additive spanners of size $O(n^{3/2})$ [1]. By setting $f = 1$ and choosing k properly, this leads to EFT spanners of size $O(n^{4/3})$ with additive distortion 38, size $\widetilde{O}(n^{7/5})$ with additive distortion 28 (w.h.p.), and size $O(n^{3/2})$

[6] Note that in this definition we allow $d_{G-F}(s,t)$ to become infinite (if the removal of F disconnects s from t). In that case we assume the inequality to be trivially satisfied.

[7] The $\widetilde{O}$ notation hides poly-logarithmic factors in n.

Table 1. State of the art and new results on additive EFT spanners. Distortions and sizes marked with "*" hold w.h.p.

State of the art		Our results	
Size	Additive distortion	Size	Additive distortion
$\widetilde{O}(n^{5/3})$	2 [17]	$O(n^{5/3})$	2
$\widetilde{O}(n^{3/2})$	6 [17]	$O(n^{3/2})$	4
$\widetilde{O}(n^{\frac{7}{5}})$*	28* [8,9]	$\widetilde{O}(n^{\frac{7}{5}})$*	10*
$O(n^{\frac{4}{3}})$	38 [4,8]	$O(n^{\frac{4}{3}})$	14

with additive distortion 14. Finally, using a different approach, Parter [17] recently presented 2- and 6-additive EFT/VFT spanners of size $\widetilde{O}(n^{5/3})$ and $\widetilde{O}(n^{3/2})$, respectively.

1.1 Our Results.

In this paper, we focus on additive EFT spanners, and we improve all the known such spanners in terms of sparsity or stretch (see Table 1). We also present some better results for additive VFT and f-EFT spanners.

In more detail, our improved EFT spanners exploit the following two novel approaches. Our first technique (see Section 2), assumes that we are given an additive *sourcewise* fault-tolerant spanner A_S, i.e., a fault-tolerant spanner that guarantees low distortion only for the distances from a given set S of source nodes. We show that, by carefully choosing S and by augmenting A_S with a conveniently selected *small* subset of edges, it is possible to construct a fault-tolerant spanner (approximately preserving *all* pairwise distances) with a moderate increase of the stretch. This, combined with the sourcewise EFT spanners in [7,18], leads to the first two results in the table. In particular, we reduce the additive stretch of the best-known spanner of size $\tilde{O}(n^{3/2})$ from 6 [17] to 4 (actually, we also save a polylogarithmic factor in the size here). For the case of stretch 2, we slightly decrease the size from $\widetilde{O}(n^{5/3})$ [17] to $O(n^{5/3})$. This technique also applies to VFT spanners. In particular, we achieve a 2-additive VFT spanner of size $O(n^{5/3})$ rather than $\widetilde{O}(n^{5/3})$ [17], and a 4-additive VFT spanner of size $O(n^{3/2}\sqrt{\log n})$, improving on the 6-additive VFT spanner of size $\widetilde{O}(n^{3/2})$ in [17].

Our second technique (see Section 3) relies on some properties of known additive spanners. We observe that some known additive spanners are based on *clustering techniques* that construct a small-enough number of clusters. Furthermore, the worst-case stretch of these spanners is achieved only in some specific cases. We exploit these facts to augment the spanner $H = M \cup A$ based on the already mentioned construction of [8] with a small number of inter and intra-cluster edges. This allows us to reduce the additive stretch without any asymptotic increase in the number of edges.

Finally, for the case of multiple edge failures, we are able to prove that the construction in [8] has in fact an additive stretch of only $2f(\beta+\alpha-1)+\beta$ (rather than $2f(2\beta+\alpha-1)+\beta$).

Theorem 1. *Let A be a β-additive spanner of G, and let M be an α-multiplicative f-EFT spanner of G. The graph $H=(V(G), E(A) \cup E(M))$ is a $(2f(\beta+\alpha-1)+\beta)$-additive f-EFT spanner of G. In the special case $f=1$, the additive stretch is at most $2\beta+\alpha-1$.*

We see this as an interesting result since, to the best of our knowledge, the construction of [8] is the only known approach for building additive spanners withstanding more than a single edge fault. At a high-level, in [8] the shortest path in $G-F$ between two vertices is decomposed into (roughly $2f$) subpaths called *blocks*. The authors then show that it is possible to build a *bypass* (i.e., a fault-free path) in H between the endpoints of each block such that the additive error incurred by using this path is at most $\beta+\alpha-1$. Actually, in addition to those intra-block bypasses, the spanner H contains some inter-block *shortcuts*, that are exploited in order to prove a better distortion. Due to space limitations, the proof of this result will be given in the full version of the paper.

1.2 Related Work

A notion closely relate to fault-tolerant spanners is the one of *Distance Sensitivity Oracles* (DSO). The goal here is to compute, with a *low* preprocessing time, a *compact* data structure which is able to *quickly* answer distance queries following some component failures (possibly in an approximate way). For recent achievements on DSO, we refer the reader to [5, 6, 11, 16].

Another setting which is very close in spirit to fault-tolerant spanners is the recent work on fault-tolerant *approximate shortest-path trees*, both for unweighted [19] and for weighted [5,7] graphs. In [3] it was introduced the resembling concept of *resilient spanners*, i.e., spanners that approximately preserve the *relative* increase of distances due to an edge failure.

There was also some research (see for example [12, 13]) on spanners approximately preserving the distance from a given set of nodes (*sourcewise* spanners), among a given set of nodes (*subsetwise* spanners), or between given pairs of nodes (*pairwise* spanners). In this framework a spanner is called a *preserver* if distances are preserved (in other words, the stretch function is the identity function). In particular, in one of our constructions we exploit a fault-tolerant version of a sourcewise preserver.

1.3 Notation

Given an unweighted, undirected graph G, let us denote by $\pi_G(u, v)$ a shortest path) between u and v in G. When the graph G is clear from the context we might omit the subscript. Given a simple path π in G and two vertices $s, t \in V(\pi)$, we define $\pi[s, t]$ to be the subpath of π connecting s and t. Moreover, we denote

Algorithm 1: Algorithm for computing a fault-tolerant additive spanner of G from a β-additive f-EFT/VFT sourcewise spanner. The parameter p affects the size of the returned spanner and it will be suitably chosen.

```
$\texttt{color}(v) \leftarrow \texttt{white} \quad \forall v \in V$; $\texttt{counter}(v) \leftarrow f+1 \quad \forall v \in V$
$S \leftarrow \emptyset$; $E' \leftarrow \emptyset$

while $\exists s \in V \setminus S : \delta_{white}(s) \geq p$ do
    $S \leftarrow S \cup \{s\}$                                /* Add a new source $s$ */
    $\texttt{color}(s) \leftarrow \texttt{red}$
    foreach $u \in N_{white}(s)$ do
        $\texttt{counter}(u) \leftarrow \texttt{counter}(u) - 1$
        $E' \leftarrow E' \cup \{(s,u)\}$
        if $counter(u) = 0$ then
            $\texttt{color}(u) \leftarrow \texttt{black}$

$E' \leftarrow E' \cup \{(u,v) \in E : \texttt{color}(u) = \texttt{white}\}$
$A_S \leftarrow \beta$-additive f-EFT/VFT sourcewise spanner w.r.t. sources in $S$
return $H \leftarrow (V(G), E' \cup E(A_S))$
```

by $|\pi|$ the *length* of π, i.e., the number of its edges. When dealing with one or multiple *failed edges*, we say that a path π is *fault-free* if it does not contain any of such edges. Finally, if two paths π and π' are such that the last vertex of π coincides with the first vertex of π', we will denote by $\pi \circ \pi'$ the path obtained by concatenating π with π'.

2 Augmenting Sourcewise Fault-Tolerant Spanners

We next describe a general procedure (see Algorithm 1) to derive purely additive fault-tolerant spanners from sourcewise spanners of the same type.

The main idea of the algorithm is to select a small subset S of *source* vertices of G, which we call `red`. These vertices are used to build a fault-tolerant sourcewise spanner of the graph. The remaining vertices are either `black` or `white`. The former ones are always adjacent to a source, even in the event of f edge/vertex failures. Finally, edges incident to `white` vertices are added to the sought spanner, as their overall number is provably small.

During the execution of the algorithm, we let $\texttt{color}(u) \in \{\texttt{white}, \texttt{black}, \texttt{red}\}$ denote the current color of vertex u. We define $N_{\texttt{white}}(u)$ to be the set of neighbors of u which are colored `white`, and we let $\delta_{\texttt{white}}(u) = |N_{\texttt{white}}(u)|$. We will also assign a non-negative counter $\texttt{counter}(u)$ to each vertex. Initially all these counters will be positive, and then they will only be decremented. A vertex u is colored `black` only when $\texttt{counter}(u)$ reaches 0, and once a `white` vertex is colored either `black` or `red` it will never be recolored `white` again. Therefore, we have that $\texttt{color}(u) = \texttt{black}$ implies $\texttt{counter}(u) = 0$.

We first bound the size of the spanner H.

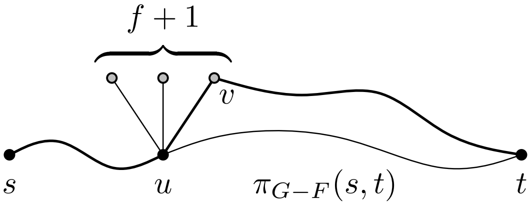

Fig. 1. A case of the proof of Theorem 2. Bold edges are in $H - F$. The black vertex u is adjacent to at least $f + 1$ vertices in S.

Lemma 1. *Algorithm 1 computes a spanner of size* $O\left(np + nf + \gamma\left(n, \left\lfloor \frac{(f+1)n}{p} \right\rfloor\right)\right)$, *where* $\gamma(n, \ell)$ *is the size of the spanner* A_S *for* $|S| = \ell$.

Proof. The edges added to E' by line 8 are at most $(f + 1)n$, since each time an edge (s, u) is added to H the counter $\texttt{counter}(u)$ is decremented, and at most $(f + 1)n$ counter decrements can occur.

To bound the edges added to E' by line 11, observe that all the edges in $\{(u, v) \in E : \texttt{color}(u) = \texttt{white}\}$ which are incident to a $\texttt{red}$ vertex, have already been added to H by line 8, hence we only consider vertices v which are either $\texttt{white}$ or $\texttt{black}$. Let v be such a vertex and notice that, before line 11 is executed, we must have $\delta_{\texttt{white}}(v) < p$, as otherwise v would have been selected as a source and colored $\texttt{red}$. This immediately implies that line 11 causes the addition of at most np edges to H.

It remains to bound the size of A_S. It is sufficient to show that $|S| \leq \left\lfloor \frac{(f+1)n}{p} \right\rfloor$ at the end of the algorithm. Each time a source s is selected, s has at least p white neighbors in G, hence the quantity $\sum_{u \in V(G)} \texttt{counter}(u)$ decreases by at least p. The claim follows by noticing that $\sum_{u \in V(G)} \texttt{counter}(u) = n(f + 1)$ at the beginning of the algorithm and, when the algorithm terminates, it must be non-negative. □

We next bound the distortion of H.

Theorem 2. *Algorithm 1 computes a* $(\beta + 2)$*-additive* f*-EFT/VFT spanner.*

Proof. Consider two vertices $s, t \in V(G)$ and a set F of at most f failed edges/vertices of G, we will show that $d_{H-F}(s, t) \leq d_{G-F}(s, t) + \beta + 2$. We assume, w.l.o.g., that s and t are connected in $G - F$, as otherwise the claim trivially holds.

If all the vertices in $\pi = \pi_{G-F}(s, t)$ are $\texttt{white}$, then all their incident edges have been added to H (see line 11 of Algorithm 1), hence $d_{H-F}(s, t) = d_{G-F}(s, t)$.

Otherwise, let $u \in V(\pi)$ be the closest vertex to s such that $\texttt{color}(u) \neq \texttt{white}$. Notice that, by the choice of u, $H - F$ contains all the edges of $\pi[s, u]$. If $\texttt{color}(u) = \texttt{red}$ then:

$$\begin{aligned} d_{H-F}(s,t) &\leq d_{H-F}(s,u) + d_{H-F}(u,t) \leq d_{G-F}(s,u) + d_{A_S-F}(u,t) \\ &\leq d_{G-F}(s,u) + d_{G-F}(u,t) + \beta = d_{G-F}(s,t) + \beta \end{aligned}$$

where we used the fact that $u \in \pi_{G-F}(s,t)$.

Finally, if $\texttt{color}(u) = \texttt{black}$ then $\texttt{counter}(u) = 0$, hence u has at least $f+1$ `red` neighbors in H (see Figure 1). As a consequence, there is at least one `red` vertex v such that $(u,v) \in H - F$ (and hence $(u,v) \in G - F$), therefore:

$$\begin{aligned} d_{H-F}(s,t) &\leq d_{H-F}(s,u) + d_{H-F}(u,v) + d_{H-F}(v,t) \\ &\leq d_{G-F}(s,u) + 1 + d_{A_S-F}(v,t) \leq d_{G-F}(s,u) + 1 + d_{G-F}(v,t) + \beta \\ &\leq d_{G-F}(s,u) + 1 + d_{G-F}(v,u) + d_{G-F}(u,t) + \beta \\ &= d_{G-F}(s,t) + \beta + 2. \end{aligned}$$

□

Let $S' \subset V(G)$ be a set of *sources*. In [18] it is shown that a sourcewise EFT/VFT *preserver* (i.e, a $(1,0)$ EFT/VFT sourcewise spanner) of G having size $\gamma(n, |S'|) = O(n\sqrt{n|S'|})$ can be built in polynomial time. Combining this preserver and Algorithm 1 with $p = n^{\frac{2}{3}}$, we obtain the following:

Corollary 1. *There exists a polynomial time algorithm to compute a 2-additive EFT/VFT spanner of size $O(n^{\frac{5}{3}})$.*

Furthermore, we can exploit the following result in [7].[8]

Lemma 2 ([7]). *Given an (α,β)-spanner A and a subset S' of vertices, it is possible to compute in polynomial time a subset of $O(|S'| \cdot n)$ edges E', so that $A \cup E'$ is an (α,β) EFT sourcewise spanner w.r.t. S'. The same result holds for VFT spanners, with E' of size $O(|S'| \cdot n \log n)$.*

Combining the above result with the 2-additive spanner of size $O(n^{3/2})$ in [1], we obtain 2-additive EFT and VFT sourcewise spanners of size $\gamma(n, |S'|) = O(n\sqrt{n} + |S'| \cdot n)$ and $\gamma(n, |S'|) = O(n\sqrt{n} + |S'| \cdot n \log n)$, respectively. By using these spanners in Algorithm 1, with $p = \sqrt{n}$ and $p = \sqrt{n \log n}$ respectively, we obtain the following result.

Corollary 2. *There exists a polynomial time algorithm to compute a 4-additive EFT spanner of size $O(n^{3/2})$, and a 4-additive VFT spanner of size $O(n^{3/2}\sqrt{\log n})$.*

3 Augmenting Clustering-Based Additive Spanners

Most additive spanners in the literature are based on a *clustering* technique. A subset of the vertices of the graph G is partitioned into *clusters*, each containing a special *center* vertex along with some of its neighbors. The distances between these clusters is then reduced by adding a suitable set of edges to the spanner. This technique is used, for example, in [4,9]. We now describe a general technique which can be used to augment such spanners in order to obtain a fault-tolerant additive spanner.

[8] Actually, the result in [7] is claimed for the single source case only, but it immediately extends to multiple sources.

Algorithm 2: Algorithm for computing a fault-tolerant additive spanner from multiplicative and clustering-based additive spanners. Here $\{\mathcal{C}, \mathrm{cnt}(\cdot)\}$ denotes the clustering of G while $\delta(C, C')$ is the set of the edges in $E(G)$ with one endpoint in C and the other in C'.

```
$E' \leftarrow \emptyset$; $M \leftarrow (\alpha, 0)$ EFT spanner; $A \leftarrow (1, \beta)$ clustering-based spanner
foreach $C \in \mathcal{C}$ do
    foreach $v \in C$ do
        if $\exists (v, x) \in E(G) : x \in C \setminus \{cnt(v)\}$ then
            $E' \leftarrow E' \cup \{(v, x)\}$.
foreach $C, C' \in \mathcal{C} : C \neq C'$ do
    if $\exists e, e' \in \delta(C, C') : e$ and $e'$ are vertex-disjoint then
        $E' \leftarrow E' \cup \{e, e'\}$.
    else if $\exists e, e' \in \delta(C, C') : e \neq e'$ then
        $E' \leftarrow E' \cup \{e, e'\}$.
    else
        $E' \leftarrow E' \cup \delta(C, C')$          /* $\delta(C, C')$ contains at most one edge */
return $H \leftarrow (V(G), E' \cup E(M) \cup E(A))$
```

More formally, a *clustering* of G is a partition $\mathcal{C}$ of a subset of $V(G)$. We call each element $C \in \mathcal{C}$ a *cluster*. We say that a vertex v is *clustered* if it belongs to a cluster, and *unclustered* otherwise. Each cluster $C \in \mathcal{C}$ is associated with a vertex $u \in C$ which is the *center* of C. For each *clustered* vertex v, we denote by $\mathrm{cnt}(v)$ the center of the cluster containing v.

We say that a β-additive spanner A is *clustering-based* if there exists a clustering $\mathcal{C}$ of G such that: (i) A contains all the edges incident to unclustered vertices, (ii) A contains all the edges between every clustered vertex v and $\mathrm{cnt}(v)$, and (iii) the following property holds:

Property 1. For every $u, v \in V(G)$ such that v is a clustered vertex, there exists a path $\tilde{\pi}(u, v)$ in A such that one of the following conditions holds:

(P1) $|\tilde{\pi}(u, v)| \leq d_G(u, v) + \beta - 2$;
(P2) $|\tilde{\pi}(u, v)| = d_G(u, v) + \beta - 1$ and either (i) $v = \mathrm{cnt}(v)$, or (ii) the last edge of $\tilde{\pi}(u, v)$ is $(\mathrm{cnt}(v), v)$.
(P3) $|\tilde{\pi}(u, v)| = d_G(u, v) + \beta$, $v \neq \mathrm{cnt}(v)$, and the last edge of $\tilde{\pi}(u, v)$ is $(\mathrm{cnt}(v), v)$.

Our algorithm works as follows (see Algorithm 2). We add to our spanner H a β-additive clustering-based spanner A, and a α-multiplicative EFT spanner M. Note that so far our construction is the same as in [8], with the extra constraint that A is clustering-based. We then augment H by adding a carefully chosen subset E' of inter and intra-cluster edges.

Let $\{\mathcal{C}, \mathrm{cnt}(\cdot)\}$ be the clustering of A. It is easy to see that E' contains at most $O(n + |\mathcal{C}|^2)$ edges and hence $|E(H)| = O(|E(A)| + |E(M)| + |\mathcal{C}|^2)$. We now prove an useful lemma which is then used to upper-bound the distortion of the spanner H.

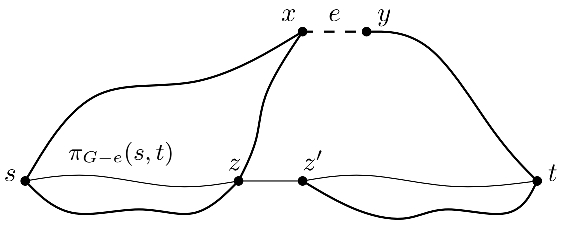

Fig. 2. Decomposition of $\pi_{G-e}(s,t)$ so that all shortest paths from s to z (resp. from z' to t) in A are fault-free. Bold lines denote shortest paths in A.

Lemma 3. *Let A be a spanning subgraph of G, let $e \in E(G)$ be a failed edge and $s, t \in V(G)$ be two vertices satisfying $d_A(s,t) < d_{A-e}(s,t) \neq \infty$. There exist two consecutive vertices z, z' in $V(\pi_{G-e}(s,t))$, with $d_{G-e}(s,z) < d_{G-e}(s,z')$, such that every shortest path in A between s and z (resp. t and z') is fault-free.*

Proof. First of all, notice that $e = (x,y)$ belongs to every shortest path between s and t in A, therefore let $\pi_A(s,t) = \langle s, \dots, x, y, \dots, t \rangle$. Consider the vertices of $\pi_{G-e}(s,t)$ from s to t, let $z \in V(\pi)$ be the last vertex such that there exists a shortest path π between z and t in A that contains e (z can possibly coincide with s), and call z' the vertex following z in π (see Figure 2). By the choice of z, we have that no shortest path between z' and t in A can contain e. Moreover, π must traverse e in the same direction as $\pi_A(s,t)$, i.e., $\pi = \langle z, \dots, x, y, \dots, t \rangle$. This is true since otherwise we would have $\pi = \langle z, \dots, y, x, \dots, t \rangle$ and hence $\pi_A(s,t)[s,x] \circ \pi[x,t]$ would be a fault-free shortest path between s and t in A, a contradiction.

It remains to show that no shortest path between s and z in A can contain e. Suppose this is not the case, then:

$$d_A(s,z) = d_A(s,y) + d_A(y,z) = d_A(s,x) + 1 + 1 + d_A(z,x) = d_A(s,z) + 2$$

which is again a contradiction. □

We are now ready to prove the main theorem of this section.

Theorem 3. *Algorithm 2 computes a $(2\beta + \max\{2, \alpha - 3\})$-additive EFT-spanner.*

Proof. Choose any two vertices $s, t \in V(G)$ and a failed edge $e \in E(G)$. Suppose that s, t are connected in $G - e$, and that every shortest path between s and t in A contains e (as otherwise the claim trivially holds). We partition $\pi_{G-e}(s,t)$ by finding $z, z' \in V(\pi_{G-e}(s,t))$ as shown by Lemma 3.

The edge (z,z') is in $\pi_{G-e}(s,t)$ and hence cannot coincide with e. Moreover, we suppose $(z,z') \notin E(H)$, as otherwise we would immediately have:

$$\begin{aligned} d_{H-e}(s,t) &\leq d_A(s,z) + d_{H-e}(z,z') + d_A(z',t) \\ &\leq d_{G-e}(s,z) + \beta + 1 + d_{G-e}(z',t) + \beta + (d_{G-e}(z,z') - 1) \\ &= d_{G-e}(s,t) + 2\beta - 1. \end{aligned}$$

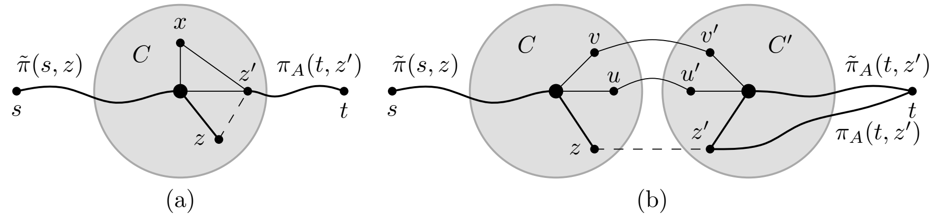

Fig. 3. Cases considered in the proof of Theorem 3 to build a fault-free path between s and t with small additive distortion. Bold lines represent shortest paths/edges in A. Solid lines represent paths/edges in H while the dashed edge (z,z') does not belong to $E(H)$ and cannot coincide with e.

This means that both z and z' must be clustered. Let C (resp. C') be the (unique) cluster that contains z (resp. z').[9]

Let $\gamma := d_A(s,z) - d_{G-e}(s,z)$ and $\gamma' := d_A(t,z') - d_{G-e}(z',t)$. Clearly $0 \le \gamma, \gamma' \le \beta$. If $\gamma + \gamma' \le 2\beta - 2$ then we are done as:

$$\begin{aligned} d_{H-e}(s,t) &\le d_A(s,z) + d_{B-e}(z,z') + d_A(z',t) \\ &\le d_{G-e}(s,z) + d_{G-e}(z',t) + 2\beta - 2 + \alpha d_{G-e}(z,z') + (d_{G-e}(z,z') - 1) \\ &\le d_{G-e}(s,t) + 2\beta + \alpha - 3. \end{aligned}$$

Next we assume that $\gamma + \gamma' \ge 2\beta - 1$. This means that either (i) γ and γ' are both equal to β, or (ii) exactly one of them is β while the other equals $\beta - 1$. Assume w.l.o.g. that $\gamma = \beta$. This implies that $d_A(s,z) = |\tilde{\pi}(s,z)|$, hence $\tilde{\pi}(s,z)$ is a shortest path between s and z in A, and by Lemma 3, it is fault-free.

In the rest of the proof we separately consider the cases $C = C'$ and $C \neq C'$. In the former case, since $(z,z') \notin E(H)$, we know that, during the execution of the loop in line 2 of Algorithm 2, an edge (z',x) such that $x \in C \setminus \{z, \mathrm{cnt}(z)\}$ has been added to E (see Figure 3 (a)). Since the paths $\langle \mathrm{cnt}(z), z' \rangle$ and $\langle \mathrm{cnt}(z), x, z' \rangle$ are edge-disjoint, at least one of them is fault free, hence: $d_{H-e}(\mathrm{cnt}(z), z') \le 2 = d_{G-e}(z,z') + 1$. Thus, by (P3) of Property 1:

$$\begin{aligned} d_{H-e}(s,t) &\le d_A(s,\mathrm{cnt}(z)) + d_{H-e}(\mathrm{cnt}(z),z') + d_A(z',t) \\ &\le d_{G-e}(s,z) + \beta - 1 + d_{G-e}(z,z') + 1 + d_{G-e}(z',t) + \beta \\ &\le d_{G-e}(s,t) + 2\beta. \end{aligned}$$

We now consider the remaining case, namely $C \neq C'$. We have that $(z,z') \notin E(H)$, therefore during the execution of the loop in line 6 of Algoritm 2, two distinct edges $(u,u'), (v,v')$ so that $u,v \in C$ and $u',v' \in C'$ must have been added to E (see Figure 3 (b)).

Notice that u' and v' might coincide, but this would imply that $u \neq v$ and hence $u' = v' = z'$. This, in turn, implies the existence of two edge-disjoint paths

[9] Notice that C and C' may coincide.

of length 2 between $\mathrm{cnt}(z)$ and z' in H, namely $\langle \mathrm{cnt}(z), u, z' \rangle$ and $\langle \mathrm{cnt}(z), v, z' \rangle$. As at least one of them must be fault-free. Therefore:

$$\begin{aligned} d_{H-e}(s,t) &\leq d_A(s, \mathrm{cnt}(z)) + d_{H-e}(\mathrm{cnt}(z), z') + d_A(z', t) \\ &\leq d_{G-e}(s,z) + \beta - 1 + d_{G-e}(z,z') + 1 + d_{G-e}(z',t) + \beta \\ &= d_{G-e}(s,t) + 2\beta. \end{aligned}$$

On the other hand, if $u' \neq v'$, we consider the two paths $\pi' = \langle \mathrm{cnt}(z), u, u', \mathrm{cnt}(z') \rangle$ and $\pi'' = \langle \mathrm{cnt}(z), v, v', \mathrm{cnt}(z') \rangle$.[10] Notice that π' and π'' can share at most a single edge, namely $(\mathrm{cnt}(z), z)$ (when $u = v = z$), and that this edge cannot coincide with e as it belongs to $\tilde{\pi}(s,z)$ which is a fault-free shortest path between s and z in A. This implies that at least one of π' and π'' is fault-free and hence $d_{H-e}(\mathrm{cnt}(z), \mathrm{cnt}(z')) \leq 3 = d_{G-e}(z,z') + 2$. If $e = (\mathrm{cnt}(z'), z')$ then, since $|\tilde{\pi}(t,z')| \geq d_A(z',t) \geq d_{G-e}(z',t) + \beta - 1$, either (P2) or (P3) of Property 1 must hold, so we know that $\tilde{\pi}(t,z')[t, \mathrm{cnt}(z')]$ is fault-free and has a length of at most $d_{G-e}(z',t) + \beta - 1$. We have:

$$\begin{aligned} d_{H-e}(s,t) &\leq d_A(s, \mathrm{cnt}(z)) + d_{H-e}(\mathrm{cnt}(z), \mathrm{cnt}(z')) + d_A(\mathrm{cnt}(z'), t) \\ &\leq d_{G-e}(s,z) + \beta - 1 + d_{G-e}(z,z') + 2 + d_{G-e}(z',t) + \beta - 1 \\ &\leq d_{G-e}(s,t) + 2\beta. \end{aligned}$$

Finally, when $e \neq (\mathrm{cnt}(z'), z')$, we have:

$$\begin{aligned} d_{H-e}(s,t) &\leq d_A(s, \mathrm{cnt}(z)) + d_{H-e}(\mathrm{cnt}(z), \mathrm{cnt}(z')) + d_{H-e}(\mathrm{cnt}(z'), z') + d_A(z',t) \\ &\leq d_{G-e}(s,z) + \beta - 1 + d_{G-e}(z,z') + 2 + 1 + d_{G-e}(z',t) + \beta \\ &\leq d_G(s,t) + 2\beta + 2. \end{aligned}$$

This concludes the proof. □

This result can immediately be applied to the 6-additive spanner of size $O(n^{\frac{4}{3}})$ in [4], which is clustering-based and uses $O(n^{\frac{2}{3}})$ clusters. Using the 5-multiplicative EFT spanner M of size $O(n^{4/3})$ from [10], we obtain:

Corollary 3. *There exists a polynomial time algorithm to compute a* 14*-additive EFT spanner of size* $O(n^{\frac{4}{3}})$.

We can similarly exploit the clustering-based spanner of [9] which provides, w.h.p., an additive stretch of 4 and a size of $\widetilde{O}(n^{\frac{7}{5}})$ by using $O(n^{\frac{3}{5}})$ clusters.

Corollary 4. *There exists a polynomial time randomized algorithm that computes w.h.p. a* 10*-additive EFT spanner of* G *of size* $\widetilde{O}(n^{\frac{7}{5}})$.

[10] Some consecutive vertices of π' (resp. π'') might actually coincide. In this case, we ignore all but the first of such vertices and define π' (resp. π'') accordingly.

References

1. Donald Aingworth, Chandra Chekuri, Piotr Indyk, and Rajeev Motwani. Fast estimation of diameter and shortest paths (without matrix multiplication). *SIAM J. Comput.*, 28(4):1167–1181, 1999.
2. Ingo Althöfer, Gautam Das, David P. Dobkin, Deborah Joseph, and José Soares. On sparse spanners of weighted graphs. *Discrete & Computational Geometry*, 9:81–100, 1993.
3. Giorgio Ausiello, Paolo Giulio Franciosa, Giuseppe Francesco Italiano, and Andrea Ribichini. On resilient graph spanners. In *ESA*, pages 85–96, 2013.
4. Surender Baswana, Telikepalli Kavitha, Kurt Mehlhorn, and Seth Pettie. Additive spanners and (alpha, beta)-spanners. *ACM Transactions on Algorithms*, 7(1):5, 2010.
5. Surender Baswana and Neelesh Khanna. Approximate shortest paths avoiding a failed vertex: Near optimal data structures for undirected unweighted graphs. *Algorithmica*, 66(1):18–50, 2013.
6. Aaron Bernstein and David R. Karger. A nearly optimal oracle for avoiding failed vertices and edges. In *STOC*, pages 101–110, 2009.
7. Davide Bilò, Luciano Gualà, Stefano Leucci, and Guido Proietti. Fault-tolerant approximate shortest-path trees. In *ESA*, pages 137–148, 2014.
8. Gilad Braunschvig, Shiri Chechik, and David Peleg. Fault tolerant additive spanners. In *WG*, pages 206–214, 2012.
9. Shiri Chechik. New additive spanners. In *SODA*, pages 498–512, 2013.
10. Shiri Chechik, Michael Langberg, David Peleg, and Liam Roditty. Fault-tolerant spanners for general graphs. In *STOC*, pages 435–444, 2009.
11. Shiri Chechik, Michael Langberg, David Peleg, and Liam Roditty. f-sensitivity distance oracles and routing schemes. In *ESA*, pages 84–96, 2010.
12. Don Coppersmith and Michael Elkin. Sparse sourcewise and pairwise distance preservers. *SIAM J. Discrete Math.*, 20(2):463–501, 2006.
13. Marek Cygan, Fabrizio Grandoni, and Telikepalli Kavitha. On pairwise spanners. In *STACS*, pages 209–220, 2013.
14. Michael Dinitz and Robert Krauthgamer. Fault-tolerant spanners: better and simpler. In *PODC*, pages 169–178, 2011.
15. Paul Erdős. Extremal problems in graph theory. In *Theory of Graphs and its Applications*, pages 29–36, 1964.
16. Fabrizio Grandoni and Virginia Vassilevska Williams. Improved distance sensitivity oracles via fast single-source replacement paths. In *FOCS*, pages 748–757, 2012.
17. Merav Parter. Vertex fault tolerant additive spanners. In *DISC*, pages 167–181, 2014.
18. Merav Parter and David Peleg. Sparse fault-tolerant BFS trees. In *ESA*, pages 779–790, 2013.
19. Merav Parter and David Peleg. Fault tolerant approximate bfs structures. In *SODA*, pages 1073–1092, 2014.