New Additive Spanners

Shiri Chechik *

Abstract

This paper considers additive and purely additive spanners. We present a new purely additive spanner of size $\tilde{O}(n^{7/5})$ with additive stretch 4. This construction fills in the gap between the two existing constructions for purely additive spanners, one for 2-additive spanner of size $O(n^{3/2})$ and the other for 6-additive spanner of size $O(n^{4/3})$, and thus answers a main open question in this area. In addition, we present a construction for additive spanners with $\tilde{O}(n^{1+\delta})$ edges and additive stretch of $\tilde{O}(n^{1/2-3\delta/2})$ for any $3/17 \leqslant \delta < 1/3$, improving the stretch of the existing constructions from $O(n^{1-3\delta})$ to $\tilde{O}(\sqrt{n^{1-3\delta}})$. Finally, we show that our $(1, n^{1/2-3\delta/2})$-spanner construction can be tweaked to give a sublinear additive spanner of size $\tilde{O}(n^{1+3/17})$ with additive stretch $O(\sqrt{distance})$.

1 Introduction

Graph spanners are sparse subgraphs that faithfully preserve the pairwise distances of a given graph. Formally, an (α, β)-spanner of a graph $G = (V, E)$ is a subgraph H such that for any pair of nodes s, t, $\texttt{dist}(s, t, H) \leqslant \alpha \cdot \texttt{dist}(s, t, G) + \beta$, where $\texttt{dist}(s, t, H')$ for a subgraph H' is the distance from s to t in H'. If $\alpha = 1$ we say that the spanner is *additive* and if in addition $\beta = O(1)$, we say that the spanner is *purely additive*. If $\beta = 0$ we say that the spanner is *multiplicative*, otherwise we say that the spanner is *mixed*.

Graph spanners were extensively studied since they were first introduced in [19, 20] in the late 80's. Many distributed applications use spanners as a key ingredient, e.g., synchronizers [20], compact routing schemes [21, 9, 26, 10, 25], distance oracles [3, 27], broadcasting [18], near-shortest path algorithms [12, 13, 16], etc.

Much of the work on spanners considers multiplicative spanners. It is well-known that one can efficiently construct a $(2k-1, 0)$-spanner with $O(n^{1+1/k})$ edges [2]. This size-stretch ratio is conjectured to be tight based on the girth conjecture of Erdős [17]. The girth conjecture has been proved for the specific cases of $k = 1, 2, 3,$ and 5 [29].

Although many papers considered additive spanners or mixed spanners, several key questions in this area remain open. The girth conjecture applies only to short distances. In particular, it does not contradict the existence of $(1, 2k-2)$-spanners of size $O(n^{1+1/k})$, or any (α, β)-spanners of size $O(n^{1+1/k})$ such that $\alpha + \beta = 2k - 1$ with $\alpha \geqslant 1$ and $\beta > 0$. The first construction for purely additive spanners was presented by Aingworth et al. [1]. They show how to efficiently construct a $(1, 2)$ spanner, or a 2-*additive* spanner for short, with $O(n^{3/2})$ edges (see [11, 15, 28, 24] for further follow-up). Later, an efficient construction for 6-additive spanners with $O(n^{4/3})$ edges was presented by Baswana et al. [4, 5]. Woodruff [31] later presented a different construction for 6-additive spanners with $\tilde{O}(n^{4/3})$ edges with better construction time. These are the only two purely additive spanners known so far. A major open problem in this field concerns the existence of purely additive spanners with $O(n^{1+\delta})$ edges for any fixed $\delta > 0$. Woodruff [30] showed a lower bound for additive spanners matching the girth conjecture bounds but independent of the correctness of the conjecture. More precisely, he showed the existence of graphs for which any spanner of size $O(k^{-1}n^{1+1/k})$ has an additive stretch of at least $2k - 1$.

In the absence of additional purely additive spanners or impossibility results, attempts were made to seek spanners with either non-constant additive stretch or a mix of both multiplicative and additive stretch (see, e.g., [15, 28, 23, 5]).

Bollobás *et al.* [6] presented efficient constructions for a spectrum of additive spanners with additive stretch that depends on n. More precisely, they show how to efficiently construct a $(1, n^{1-2\delta})$-spanner with $O(2^{1/\delta} n^{1+\delta})$ edges for any $\delta > 0$. This additive stretch was later improved to $(1, n^{1-3\delta})$ by Baswana *et al.* in [4, 5] and to $(1, n^{9/16-7\delta/8})$ by Pettie [22, 23] (the latter is smaller than the former for every $\delta < 7/34$). In addition, sublinear additive spanners, namely, additive spanners with stretch that is sublinear in the distances, were also considered. Thorup and Zwick [28] showed how to construct a spanner of size $O(kn^{1+1/k})$ such that for every pair of nodes s and t, the additive stretch is $O(d^{1-1/k} + 2^k)$, where $d = \texttt{dist}(s, t)$. Pettie [22, 23] later improved that result presenting an efficient span-

*Microsoft Research Silicon Valley, Mountain View CA, USA. Email: schechik@microsoft.com.

ner construction of size $O(kn^{1+\frac{(3/4)^{k-2}}{7-2(3/4)^{k-2}}})$ with additive stretch of $O(kd^{1-1/k}+k^k)$, where $d = \mathtt{dist}(s,t)$. Specifically, for $k = 2$, the size of the spanner is $O(n^{6/5})$ and the additive stretch is $O(\sqrt{d})$. For further results on mixed spanners see [14, 15, 4, 28, 22, 23, 5].

This paper considers additive and purely additive spanners. We make an additional step towards better undereating the picture of purely additive spanners, by presenting a new simple algorithm for $(1,4)$-additive spanners with $\tilde{O}(n^{7/5})$ edges. We thus answer one of the main open questions in this area of purely additive spanners, by filling in the gap between the two existing constructions. In addition, we present a construction for additive spanners with $\tilde{O}(n^{1+\delta})$ edges and additive stretch of $\tilde{O}(n^{1/2-3\delta/2})$ for any $3/17 \leqslant \delta < 1/3$. We thus decrease the stretch for this range to the root of the best known additive stretch so far. We note that it is possible to extend this range a little bit (to δ values smaller than 3/17) but the construction and analysis become much more complex. It would be interesting to see if this range can be extended all the way, to any $0 \leqslant \delta < 1/3$. Our construction for $(1, n^{1/2-3\delta/2})$-spanners with $\tilde{O}(n^{1+\delta})$ edges is quite involved and requires a number of new ideas. The construction consists of several procedures, where each procedure provides certain desired properties and may be of independent interest. Finally, we show that our $(1, n^{1/2-3\delta/2})$-spanner construction can be tweaked to slightly improve the size of the sublinear additive spanner of Pettie [22, 23] with additive stretch $O(\sqrt{d})$ from $O(n^{1+1/5})$ to $\tilde{O}(n^{1+3/17})$.

2 $\tilde{O}(n^{7/5})$ edge spanners with additive stretch 4

In this section we present a new construction for a $(1,4)$-spanner with $O(n^{7/5}\log^{1/5} n)$ edges. Here and throughout, $n = |V|$ and $m = |E|$. Let us introduce some preliminaries. Denote the vertex set and edge set of a subgraph H by $V(H)$ and $E(H)$, respectively. For nodes $x, y \in V$ and subgraph H, $\mathtt{dist}(x,y,H)$ is the distance between x and y in H. For a node $x \in V$, a set of nodes $S \subseteq V$ and subgraph H, $\mathtt{dist}(x,S,H)$ is the distance between x and the node $y \in S$ closest to x in H. For a node $x \in V$, an integer r, and a subgraph H, let $\Gamma(x,r,H)$ be the set of nodes at distance at most r from x in H, namely, $\Gamma(x,r,H) = \{v \in V \mid \mathtt{dist}(x,v,H) \leqslant r\}$, and let $\Gamma^*(v,r,H) = \{x \in V \mid \mathtt{dist}(v,x,H) = r\}$. Similarly, for a path P, an integer r and a subgraph H, denote the set of neighbors of P by $\Gamma(P,r,H) = \{v \in V \mid \mathtt{dist}(v,V(P),H) \leqslant r\}$. To simplify notation, when $H = G$ and/or when $r = 1$ we omit them. Let $|P|$ denote the number of edges in P.

Let $\deg(v)$ for a node v be its degree. We say that a node is *heavy* if its degree is at least $\mu = \lceil n^{2/5}\log^{1/5} n\rceil$, and *light* otherwise. For every pair of nodes s and t, select a shortest path $P(s,t)$ from s to t in G and let $\mathcal{P} = \{P(s,t) \mid s,t \in V\}$. Let $\mathcal{P}(V_1,V_2) = \{P(s,t) \mid s \in V_1, t \in V_2\}$ for subsets of nodes V_1 and V_2. The *heavy distance* between s and t, denoted $\mathtt{heavy\_dist}(s,t,G)$, is defined to be the number of heavy nodes on the path $P(s,t)$. Similarly, for a path P, denote by $\mathtt{heavy\_dist}(P,G)$ the number of heavy nodes on the path P.

We now turn to describe our $(1,4)$-spanner construction. Initially set H to be $(V,\varnothing)$. The construction consists of three stages. In the first stage, add to H all edges incident to light nodes. In the second stage, randomly select a set of nodes S_1 of expected size 9μ, by choosing every node from V independently at random with probability $9\mu/n$. For every node $x \in S_1$, construct a BFS tree $T(x)$ rooted at x spanning all vertices V, and add the edges of $T(x)$ to H.

In the third and final stage, choose a set S_2 of n/μ nodes in expectation, called hereafter center clusters. This can be done by choosing each node independently at random with probability $1/\mu$. Next, for each heavy node x such that none of the nodes in $\{x\}\cup\Gamma(x)$ were chosen to S_2, add all incident edges of x to H. For each node $x \in S_2$, create a cluster $C(x)$, initially set to $\{x\}$. For every heavy node v such that $v \notin S_2$ and $\Gamma(v)\cap S_2 \neq \varnothing$, arbitrarily choose one node x in $\Gamma(v)\cap S_2$, add v to x's cluster $C(x)$ and add the edge (v,x) to H. Finally, for each pair of nodes x_1 and x_2 in S_2 do the following. Consider all shortest paths $P(y_1,y_2)$ in $\mathcal{P}(C(x_1),C(x_2))$ such that $\mathtt{heavy\_dist}(y_1,y_2,G) \leqslant \mu^3/n$, namely, all shortest paths $P(y_1,y_2) \in \mathcal{P}$ such that $y_1 \in C(x_1)$, $y_2 \in C(x_2)$ and $\mathtt{heavy\_dist}(y_1,y_2,G) \leqslant \mu^3/n$. Choose the path $P(\hat{y_1},\hat{y_2})$ with minimal length $|P(\hat{y_1},\hat{y_2})|$, and add it to H.

This completes the description of our spanner construction. See Procedure 4-**Additive-Sanner** for the formal code.

We now bound the number of edges in the resulting spanner H.

LEMMA 2.1. *The expected number of edges in H is $O(n\mu) = \tilde{O}(n^{7/5})$.*

Proof: Let us bound the number of edges added to H in the three different stages. In the first stage, only edges adjacent to light nodes were added. Each such light node contributes at most μ edges, so at most $n\mu$ edges were added to H in this stage.

In the second stage, each node is added to S_1 with probability $9\mu/n$. Therefore, the expected number of nodes in S_1 is 9μ. For each node in S_1, a BFS tree of $n-1$ edges is added to H. Hence the expected number

```
Procedure 4-Additive-Sanner(G)

E' ← ∅.
Add to E' all edges incident to light nodes.
Select a set of nodes S_1 by independently sampling at random every node with probability 9μ/n.
For every node x ∈ S_1 do:
    Construct a BFS tree T(x) rooted at x spanning all vertices V.
    E' ← E' ∪ E(T(x)).
Select a set of nodes S_2 by independently sampling at random every node with probability 1/μ.
For each heavy node x such that ({x} ∪ Γ(x)) ∩ S_2 = ∅ do:
    Add all incident edges of x to H.
For each node x ∈ S_2 do:
    C(x) ← {x}.
For every heavy node v such that v ∉ S_2 and Γ(v) ∩ S_2 ≠ ∅ do:
    Arbitrarily choose one node x in Γ(v) ∩ S_2.
    C(x) ← C(x) ∪ {v}.
    E' ← E' ∪ {(v, x)}.
For each pair of nodes x_1 and x_2 in S_2 do:
     Let P̂ ← {P ∈ P(C(x_1), C(x_2)) | heavy_dist(P, G) ⩽ μ^3/n}
     Let P(ŷ_1, ŷ_2) be the path in P̂ with minimal |P(ŷ_1, ŷ_2)|.
     E' ← E' ∪ E(P(ŷ_1, ŷ_2)).
H ← (V, E')
Return H
```

of edges added in the second stage is $O(n\mu)$.

We now turn to analyze the expected number of edges added in the third stage. In the first part of the third stage, for every heavy node $v \notin S_2$ we either add v to one of the clusters $C(x)$ and then add the edge (v, x) to H, or (in case v remains unclustered, as $(\{v\} \cup \Gamma(v)) \cap S_2 = \varnothing$) we add to H all $\deg(v)$ edges adjacent to v. The probability that a node v will be unclustered, and thus all of its edges will be added, is $(1-1/\mu)^{\deg(v)}$. We get that the expected number of edges added for a node v is at most $1+\deg(v)(1-1/\mu)^{\deg(v)} < \mu$. Finally, the expected number of clusters is n/μ, therefore the number of cluster pairs is n^2/μ^2. For each such pair, we add a path $P = P(y_1, y_2) \in \mathcal{P}$ of heavy distance $\texttt{heavy\_dist}(y_1, y_2, G) \leqslant \mu^3/n$. Note that all edges of the path that are adjacent to light nodes were already added to H on the first stage, and as there are at most μ^3/n heavy nodes on P, at most μ^3/n edges are added for the path P on the third stage. We conclude that the number of edges added for all cluster pairs is $O((n^2/\mu^2) \cdot (\mu^3/n)) = O(n\mu) = \tilde{O}(n^{7/5})$. ∎

Next, we show that the additive stretch of the resulting spanner is indeed at most 4.

LEMMA 2.2. *For every two nodes s and t, $\texttt{dist}(s, t, H) \leqslant \texttt{dist}(s, t, G) + 4$ with probability at least $1 - 1/n^3$.*

Proof: Consider two nodes s and t. A node is said to be *covered* by the spanner H if all its adjacent edges are in H. Notice that it is enough to prove the lemma for pairs of nodes s and t that are both uncovered. To see this, let s' be the first uncovered node on the path $P(s, t)$ and let t' be the last such node. Note that all edges from s to s' on the path $P(s, t)$ and all edges from t' to t on $P(s, t)$ are in H. Therefore, if the lemma holds for s' and t', namely, with probability at least $1 - 1/n^3$, $\texttt{dist}(s, t, H) \leqslant \texttt{dist}(s, t, G) + 4$, then it holds for s, t with probability at least $1 - 1/n^3$.

So we assume now that s and t are uncovered.

We consider two cases and prove the claim separately for each case. The first case is when $\texttt{heavy\_dist}(s, t, G) > \mu^3/n$. Note that since $P(s, t)$ is a shortest path in G, necessarily every node $v \in V$ can have at most three neighbors in $P(s, t)$. Combining this with the fact that the number of heavy nodes in $P(s, t)$ is more than μ^3/n, and hence the sum of their degrees is more than μ^4/n, we get that $|\Gamma(P(s, t))| > \mu^4/(3n)$. We claim that the probability that $\Gamma(P(s, t)) \cap S_1 \neq \varnothing$

is at least $1 - 1/n^3$, as

$$\begin{aligned}&\mathbb{P}(\Gamma(P(s,t)) \cap S_1\\&\quad = \varnothing) \leqslant (1 - 9\mu/n)^{\mu^4/(3n)}\\&\quad \leqslant (1 - 9\log^{1/5} n/n^{3/5})^{(n^{3/5}/(9\log^{1/5} n))\cdot 3\log n}\\&\quad \approx 1/n^3.\end{aligned}$$

We now claim that if $\Gamma(P(s,t)) \cap S_1 \neq \varnothing$ then $\texttt{dist}(s,t,H) \leqslant \texttt{dist}(s,t,G) + 2$. To see this, let $x \in \Gamma(P(s,t)) \cap S_1$ and let z be x's neighbor in $P(s,t)$ (or x itself in case x is on $P(s,t)$). Recall that a BFS tree rooted at x is added to H in the second stage. Therefore, $\texttt{dist}(x,y,H) = \texttt{dist}(x,y,G)$ for every $y \in V$. We get that

$$\begin{aligned}\texttt{dist}(s,t,H) &\leqslant \texttt{dist}(s,x,H) + \texttt{dist}(x,t,H)\\&= \texttt{dist}(s,x,G) + \texttt{dist}(x,t,G)\\&\leqslant \texttt{dist}(s,z,G) + 1 + \texttt{dist}(z,t,G) + 1\\&= \texttt{dist}(s,t,G) + 2.\end{aligned}$$

We are left with the second case, where $\texttt{heavy\_dist}(s,t,G) \leqslant \mu^3/n$. In this case, the claim holds deterministically. Notice that there exists center clusters $x_1, x_2 \in S_2$ such that $s \in C(x_1)$ and $t \in C(x_2)$, as otherwise we would have added all their adjacent edges to H, making them covered. Let $C_1 = C(x_1)$ and $C_2 = C(x_2)$. In the third stage of the algorithm, the shortest path $P = P(\hat{y_1}, \hat{y_2})$, among all paths $P(y_1, y_2)$ such that $y_1 \in C_1$ and $y_2 \in C_2$ and $\texttt{heavy\_dist}(y_1, y_2, G) \leqslant \mu^3/n$, is added to H. Note that $|P| \leqslant |P(s,t)|$. Note also that as $\hat{y_1} \in C_1$ and $\hat{y_2} \in C_2$, we have that $\texttt{dist}(s, \hat{y_1}, H) \leqslant 2$ and $\texttt{dist}(\hat{y_2}, t, H) \leqslant 2$. We get that

$$\begin{aligned}&\texttt{dist}(s,t,H)\\&\quad \leqslant \texttt{dist}(s,\hat{y_1},H) + \texttt{dist}(\hat{y_1},\hat{y_2},H)\\&\qquad + \texttt{dist}(\hat{y_2},t,H)\\&\quad \leqslant 2 + |P| + 2 \leqslant 4 + |P(s,t)|\\&\quad = \texttt{dist}(s,t,G) + 4.\end{aligned}$$

The lemma follows. ∎

We note that the technique for handling pairs of nodes s and t such that $|\Gamma(P(s,t))| > \mu^4/(3n)$ (by selecting independently at random a set of nodes that with high probability contains a node in $\Gamma(P(s,t))$) is already used by Woodruff in [31].

By applying the union bound on all pairs of nodes, we get the following corollary.

COROLLARY 2.1. *With probability at least $1 - 1/n$, the constructed subgraph H is a 4-additive spanner for G.*

3 $\tilde{O}(n^{1+\delta})$ edge spanners with additive stretch $\tilde{O}(n^{1/2-3\delta/2})$

In this section we present a construction for a $(1, \tilde{O}(n^{1/2-3\delta/2}))$-spanner with $\tilde{O}(n^{1+\delta})$ edges for any $3/17 \leqslant \delta \leqslant 1/3$.

Throughout, let $\tilde{B}(v) = \Gamma(v,\mu)$ and $\mu = n^{1/2-3\delta/2}$. Let us partition the nodes into three sets. The set S_1 contains all nodes v such that $|\tilde{B}(v)| \leqslant \mu n^\delta$. The set S_2 contains all nodes v such that $\mu n^\delta < |\tilde{B}(v)| \leqslant n^{3\delta}$ (note that in the relevant range of $3/17 \leqslant \delta \leqslant 1/3$, $\mu n^\delta < n^{3\delta}$), and S_3 contains all nodes v such that $|\tilde{B}(v)| > n^{3\delta}$.

The algorithm consists of two main procedures, **Short-distances** and **Long-distances**. As their names imply, these procedures handle node pairs of short and long distances respectively, where we say that the path from s and t is *short*, or that s and t are *close*, if $|\Gamma(P(s,t), n^{1/2-3/2\delta})| \leqslant n^{1-2\delta}$, and *long* otherwise. Procedure **Short-distances** consists of three sub procedures: **Very-sparse**, **Sparse** and **Dense**.

Procedure **Short-distances** adds a set of edges E_{short} to the constructed spanner H such that the distance for every two close nodes in H is within $O(\mu \log n)$ additive stretch from their distance in G. As mentioned above, Procedure **Short-distances** consists of three sub procedures: **Very-sparse**, **Sparse** and **Dense**. Procedure **Very-sparse** handles very sparse areas, namely, nodes $v \in S_1$. The high level idea is that in very sparse areas, the algorithm adds a small set of edges E_{vs} such that for every node v in S_1, prefixes to all its shortest paths are contained in E_{vs}. Therefore, in some sense (and as will become clearer later on) these nodes are already "taken care of". Procedure **Sparse** handles sparse areas, namely, nodes $v \in S_2$. In this case by adding a set of edges E_{sparse} the algorithm ensures an additive stretch of at most $3\log n$ for node pairs in sparse areas at distance up to μ.

Loosely speaking, Procedure **Sparse** partitions all nodes of degree n^δ or higher into disjoint clusters. Each such cluster C is centered at some node v, and all nodes that belong to the cluster C are at distance 1 from v. For every cluster, the procedure adds edges between the center cluster to the other nodes in that cluster. The procedure then looks on balls $\tilde{B}(v)$ for every such center cluster v, and by a sophisticated BFS algorithm it ensures an additive stretch of at most $3\log n$ between v and every node in $\tilde{B}(v)$ (the main difference between the outcome of this algorithm and the standard BFS algorithm is that this algorithm adds a smaller number of edges at the price of approximated distances that are within $O(\log n)$ additive stretch from the exact distances, by exploiting the fact that the algorithm already added some edges inside the clusters).

Finally, Procedure **Dense** handles dense areas, namely, nodes $v \in S_3$. More precisely, it picks a set C_{rep} and adds a set of edges E_{dense} to H with the following properties. Every pair of nodes in C_{rep} has "small" additive stretch, in addition, all nodes in S_3 have a node in C_{rep} close to them.

The rough idea of the analysis of Procedure **Short-distances** is as follows. To handle close pair of nodes s and t, the general idea is as follows. We show that $E_{\mathrm{vs}} \cup E_{sparse}$ contains a path P_1 between s and some node $c_1 \in C_{rep}$ with the following properties. First, c_1 is "close" to some node on the path $P(s,t)$. Second, the path P_1 is within $O(\mu \log n)$ additive stretch from the distance between s and c_1 in G. Similarly, we show that $E_{\mathrm{vs}} \cup E_{sparse}$ contains a path P_3 between some node $c_2 \in C_{rep}$ and t with the following properties. First, c_3 is "close" to some node on the path $P(s,t)$. Second, the path P_3 is within $O(\mu \log n)$ additive stretch from the distance between c_2 and t in G. In addition, we show that the set of edges E_{dense} contains a path P_2 between c_1 and c_2 with a small additive stretch. Concatenating these three paths together, we get a path from s to t with a small additive stretch. This handles close pair of nodes.

To handle long pair of nodes Procedure **Long-distances** uses a similar technique. The procedure picks a set R_{long} and a set of edges E_{long} with the following properties. First, every pair of nodes in R_{long} is within additive stretch 2 from the distance in G. Second, for every pair of nodes s, t that is far away (i.e. not close), we show that there exist nodes $r_1, r_2 \in R_{long}$ such that r_1 and r_2 are "close" to some nodes on the path $P(s,t)$ and in addition r_1 is closed to s and r_2 is closed to t. As s and r_1 are closed, as explained above, procedure **Short-distances** guarantees that the constructed spanner H contains a path between s and r_1 within additive stretch $O(\mu \log n)$ from their distance in G. Similarly H contains a path from r_2 to t within additive stretch $O(\mu \log n)$ from their distance in G. Since r_1 and r_2 belong to R_{long}, as mentioned above E_{long} (and thus the constructed spanner H) contains a path between r_1 and r_2 that is within additive stretch 2 from their distance in G. Concatenating all these paths together we get a path from s to t that is within additive stretch $O(\mu \log n)$ from their distance in G.

Let us introduce some definitions. For a node v, the *sparse threshold* of v, denoted by $\mathtt{st}(v)$, is the smallest integer r such that $|\Gamma(v,r,G)| \leqslant r \cdot n^{\delta}$. For a subgraph P and set of edges E', let $\mathtt{cost}(P,E') = |E(P) \setminus E'|$.

For simplicity of presentation, assume the shortest path between any two nodes is unique and every subpath of a shortest path ia also a shortest path. (This is without loss of generality since one can enforce it by a perturbation of the edge weights.)

Very sparse areas. Procedure **Very-sparse** handles very sparse areas (i.e., nodes $v \in S_1$), by constructing an edge subset E_{vs} for these areas and adding it to the constructed spanner H. In this case the algorithm tries to add prefixes of exact shortest paths for node pairs of distance 2μ or higher. More precisely, the algorithm adds a set of edges E_{vs} to the constructed spanner H such that for every node $v \in S_1$ and for every node z such that $\mathtt{dist}(v,z,G) \geqslant 2\mu$, a nonempty prefix of the path $P(v,z)$ is contained in E_{vs}.

Roughly speaking, if a node v satisfies $|\tilde{B}(v)| = |\Gamma(v,\mu)| \leqslant \mu \cdot n^{\delta}$, then there must be a radius $r \leqslant \mu$ such that $|\Gamma^*(v,r)| \leqslant n^{\delta}$. We add a BFS tree from every node in $\Gamma^*(v,r)$ spanning the nodes in $\tilde{B}(v)$, and since $\Gamma^*(v,r)$ contains a "small" number of nodes this process requires adding a "small" number of edges. Moreover, for every node $z \in \tilde{B}(v)$ and for every node y at distance greater than 2μ from it, the path $P(z,y)$ must intersect with $\Gamma^*(v,r)$. We thus can show that a prefix of this path is added to the constructed spanner. In other words, for every node in a very sparse area, we add prefixes to all its shortest paths (that are of length greater than 2μ).

Formally, Procedure **Very-sparse** operates as follows. Initially all nodes are unmarked. While there is an unmarked node v with $1 < \mathtt{st}(v) \leqslant \mu$, choose v to be the unmarked node with maximal $\mathtt{st}(v)$. For every node x in $\Gamma^*(v,\mathtt{st}(v))$ construct a BFS tree $T(x)$ rooted at x in the induced graph $\Gamma(v,\mathtt{st}(v)-1) \cup \{x\}$, add the edges of $T(x)$ to E_{vs} and mark all nodes in $\Gamma(v,\mathtt{st}(v)-1,G)$. See Procedure **Very-sparse** for pseudocode.

LEMMA 3.1. *Procedure* **Very-sparse** *satisfies the following two properties.*
(a) for every node z_1 such that $|\Gamma(z_1,\mu)| \leqslant \mu \cdot n^{\delta}$, and for every node z_2 such that $\mathtt{dist}(z_1,z_2) > 2\mu$, there exists a node $x \neq z_1$ on $P(z_1,z_2)$ such that E_{vs} contains the path $P(z_1,x)$.
(b) $|E_{vs}| = O(n^{1+\delta})$.

Proof: Let S_{vs} be the set of nodes v that were chosen in the while loop of Procedure **Very-sparse** and for each $v \in S_{\mathrm{vs}}$ let $i(v)$ be the iteration of Procedure **Very-sparse** in which v was chosen. To prove (a), consider two nodes z_1 and z_2 as in the lemma. Since $|\Gamma(z_1,\mu)| \leqslant \mu \cdot n^{\delta}$, z_1 must be marked at the end of Procedure **Very-sparse**. Let $v \in S_{\mathrm{vs}}$ be the node such that $z_1 \in \Gamma(v,\mathtt{st}(v)-1,G)$ and z_1 is marked in iteration $i(v)$. Note that $z_2 \notin \Gamma(v,\mathtt{st}(v)-1,G)$ as $\mathtt{dist}(z_1,z_2) > 2\mu$. Clearly, $\Gamma^*(v,\mathtt{st}(v)) \cap P(z_1,z_2) \neq \varnothing$. Let z be the first node of $\Gamma^*(v,\mathtt{st}(v))$ on the path $P(z_1,z_2)$. Note that $P(z_1,z)$ is contained in the induced graph

```
Procedure Very-sparse(G(V, E))

E_vs ← ∅
Unmark all nodes v ∈ V
While there is an unmarked node v with 1 < st(v) ⩽ μ do:
   Choose v to be the unmarked node with maximal st(v)
   For every node x in Γ*(v, st(v)) do:
      Construct a BFS tree T(x) rooted at x in the induced graph
         Γ(v, st(v) − 1) ∪ {x}
      E_vs ← E_vs ∪ E(T(x))
   Mark all nodes of T(x), i.e., Γ(v, st(v) − 1, G)
Return E_vs
```

of $(\Gamma(v, \mathtt{st}(v)-1, G) \cup \{z\})$ and since a BFS tree $T(z)$ rooted at z in the induced graph $\Gamma(v, \mathtt{st}(v)-1, G) \cup \{z\}$ is added to E_{vs}, we get that $P(z_1, z) \subseteq E_{\text{vs}}$, as required.

To prove (b), we consider two types of nodes, the first type is nodes v such that $\mathtt{st}(v) \leqslant 2$, and the second type is nodes v such that $\mathtt{st}(v) > 2$. We show that the number of edges added to E_{vs} for each type separately is $O(n^{1+\delta})$.

Consider the first type and let $v \in S_{\text{vs}}$ such that $\mathtt{st}(v) \leqslant 2$. Note that in this case $O(n^{\delta})$ edges are added to E_{vs} for $T(v)$. Hence, for all nodes v' such that $\mathtt{st}(v') \leqslant 2$, $O(n^{1+\delta})$ are added to E_{vs}.

Consider now the second type and let $v \in S_{\text{vs}}$ such that $\mathtt{st}(v) > 2$. Let E^v_{vs} be the set of edges added to E_{vs} in iteration $i(v)$ of Procedure **Very-sparse**. We claim that $|\Gamma^*(v, \mathtt{st}(v))| \leqslant n^{\delta}$. To see this, note that by the definition of $\mathtt{st}(v)$, $|\Gamma(v, \mathtt{st}(v)-1, G)| \geqslant (\mathtt{st}(v)-1)n^{\delta}$ and that $|\Gamma(v, \mathtt{st}(v), G)| \leqslant \mathtt{st}(v) \cdot n^{\delta}$. It follows that $|E^v_{\text{vs}}| \leqslant n^{\delta} \cdot |\Gamma(v, \mathtt{st}(v)-1, G)|$. By definition of $\mathtt{st}(v)$ we also have $|\Gamma(v, \lfloor(\mathtt{st}(v)-1)/2\rfloor, G)| \geqslant \lfloor(\mathtt{st}(v)-1)/2\rfloor \cdot n^{\delta}$. It is not hard to verify that for every $\mathtt{st}(v) > 2$, $|\Gamma(v, \lfloor(\mathtt{st}(v)-1)/2\rfloor, G)| = \Theta(|\Gamma(v, \mathtt{st}(v)-1, G)|)$, and hence

$$\begin{aligned} |E_{\text{vs}}| &\leqslant \sum_{v \in S_{\text{vs}}} |E^v_{\text{vs}}| \leqslant n^{\delta} \sum_{v \in S_{\text{vs}}} |\Gamma(v, \mathtt{st}(v)-1, G)| \\ &\leqslant n^{\delta} \sum_{v \in S_{\text{vs}}} O(|\Gamma(v, \lfloor(\mathtt{st}(v)-1)/2\rfloor, G)|). \end{aligned}$$

Next, we show that the sets $\Gamma(v, \lfloor(\mathtt{st}(v)-1)/2\rfloor, G)$ are disjoint for $v \in S_{\text{vs}}$. Assume towards contradiction that $z \in \Gamma(v_1, \lfloor(\mathtt{st}(v_1)-1)/2\rfloor, G)$ and $z \in \Gamma(v_2, \lfloor(\mathtt{st}(v_2)-1)/2\rfloor, G)$ for some $v_1, v_2 \in S_{\text{vs}}$. Assume w.l.o.g. that v_1 is chosen first in Procedure **Very-sparse**. Then $\mathtt{dist}(z, v_1) \leqslant \lfloor(\mathtt{st}(v_1)-1)/2\rfloor$ and $\mathtt{dist}(z, v_2) \leqslant \lfloor(\mathtt{st}(v_2)-1)/2\rfloor$, so $\mathtt{dist}(v_1, v_2) \leqslant \lfloor(\mathtt{st}(v_1)-1)/2\rfloor + \lfloor(\mathtt{st}(v_2)-1)/2\rfloor \leqslant \mathtt{st}(v_1) - 1$. It follows that v_2 was marked at the end of iteration $i(v_1)$ of Procedure **Very-sparse**, contradiction. It follows that $\sum_{v \in S_{\text{vs}}} |\Gamma(v, \lfloor(\mathtt{st}(v)-1)/2\rfloor, G)| \leqslant n$, hence $|E_{\text{vs}}| = O(n^{1+\delta})$. ∎

Sparse areas. We now turn to describing Procedure **Sparse**, for handling sparse areas, namely, nodes v such that $\mu \cdot n^{\delta} \leqslant |\tilde{B}(v)| \leqslant n^{3\delta}$.

In this case the algorithm attempts to ensure an additive stretch of $3\log n$ for node pairs at distance up to μ. Specifically, the algorithm adds a set of edges E_{sparse} such that for every node $v \in S_2$, the distance from v to all nodes in $\tilde{B}(v)$ is within additive stretch $3\log n$ from the distance in G.

Procedure **Sparse** starts with sampling a set of center nodes C_{sparse} of expected size $n^{1-\delta}$, by selecting every node at random with probability $1/n^{\delta}$. For every node v none of whose neighbors was chosen to C_{sparse}, add all its incident edges to E^0_{sparse} (initially set to be empty). Otherwise, pick a neighbor $\mathtt{center}(v, C_{\text{sparse}}) \in C_{\text{sparse}}$ of v and add the edge $(v, \mathtt{center}(v, C_{\text{sparse}}))$ to the constructed spanner. This essentially attempts to partition all nodes of degree n^{δ} or higher into disjoint clusters.

Let us introduce some notation. For a path P, let $\mathtt{centers}(P, C_{\text{sparse}}) = \{c \in C_{\text{sparse}} \mid \exists z \in P, c = \mathtt{center}(z, C_{\text{sparse}})\}$. Consider a path P, a node $v \in V$, and a subgraph H. Let $\mathtt{BFS\text{-}Val}(P, v, C_{\text{sparse}}, H)$ be the number of center nodes $c \in \mathtt{centers}(P, C_{\text{sparse}})$ such that adding P to H will improve their distance to v, namely, such that $\mathtt{dist}(c, v, H \cup P) < \mathtt{dist}(c, v, H)$.

For a subgraph H, a node v, a center node $c \in C_{\text{sparse}}$ and a path $P = P(c, z)$ from c to some node z in $P(c, v)$, let $\mathtt{First\text{-}not\text{-}Help}(P, v, c, C_{\text{sparse}}, H)$ be the node v' on P closest to c such that adding P to H does not help the center c' of v' (in terms of its distance from v), or formally, such that $\mathtt{dist}(c', v, H \cup P) = \mathtt{dist}(c', v, H)$ where $c' = \mathtt{center}(v', C_{\text{sparse}})$.

Procedure **Sparse** employs a procedure **Approximate-BFS** that given a node v where

$|\tilde{B}(v)| \leqslant n^{3\delta}$, returns a set of edges E_{BFS} of size $\tilde{O}(n^{2\delta})$ such that the distance from v to all nodes in $\tilde{B}(v)$ in $E_{\text{BFS}} \cup E^0_{sparse}$ is within additive stretch $3\log n$ from the distance in G.

Procedure **Approximate-BFS** is invoked on every center $v \in C_{\text{sparse}}$ and we show that by adding $O(n^{2\delta}\log n)$ additional edges, the distance between v and every node in $\tilde{B}(v)$ is within $O(\log n)$ additive stretch from the distance in G. In particular, the procedure examines every other center $c \in C_{\text{sparse}}$ and adds some prefix of the path $P(c,v)$. It first tries to add the entire path $P(c,v)$, but would take the entire path only if sufficient many other centers benefit from it. Otherwise, the procedure will try to add a subpath of $P(c,v)$ of at most half it's length, again, only provided there are many centers who may benefit. This testing process continues until the procedure finds a prefix whose "benefit" is sufficiently large with respect to its length.

Formally, Procedure **Approximate-BFS** operates as follows. Let $(v, \tilde{C}(v), E^0_{sparse}, G)$ be its input, where $v \in C_{\text{sparse}}$ and $\tilde{C}(v) = \tilde{B}(v) \cap C_{\text{sparse}}$. Initially, set $E_{\text{BFS}} = \varnothing$. For every node $c \in \tilde{C}(v)$, v' is set to be v and the path $P(c,v')$ is examined and we add this path to the constructed spanner if $6 \cdot \texttt{BFS-Val}(P(c,v'), v, \tilde{C}(v), E^0_{sparse} \cup E_{\text{BFS}}) \geqslant \texttt{cost}(P(c,v'), E^0_{sparse} \cup E_{\text{BFS}})$. If not, we set $v' = \texttt{First-not-Help}(P(c,v'), v, c, \tilde{C}(v), E^0_{sparse} \cup E_{\text{BFS}})$. This process continues until $6 \cdot \texttt{BFS-Val}(P(c,v'), v, \tilde{C}(v), E^0_{sparse} \cup E_{\text{BFS}}) \geqslant \texttt{cost}(P(c,v'), E^0_{sparse} \cup E_{\text{BFS}})$. Notice that the process ends as the inequality holds for $v' = c$.

Lemma 3.2. *For every node $v \in C_{sparse}$, the set E_{BFS} returned by Procedure* **Approximate-BFS** *satisfies that* $\texttt{dist}(c, v, E^0_{sparse} \cup E_{BFS}) \leqslant \texttt{dist}(c, v, G) + 2\log n$ *for every node $c \in C_{sparse} \cap \tilde{B}(v)$.*

Proof. Consider a node $c \in \tilde{C}(v)$. Let j be the number of iterations in the while loop of Procedure **Approximate-BFS** for the node $c \in \tilde{C}(v)$. Let $v'(i)$ be the node v' in the end of iteration i for $i < j$ and let $c'(i) = \texttt{center}(v'(i), C_{\text{sparse}})$. See Figure 1 for illustration.

We claim that $\texttt{dist}(v'(i), v, E^0_{sparse} \cup E_{\text{BFS}}) \leqslant \texttt{dist}(v'(i), v, G) + 2i$. The proof of this claim is by induction on i. For $i = 0$, namely $v' = v$, the claim is trivial. Assume correctness for $i < k$ and consider $i = k$. Recall that the node $v'(k)$ is a node on the path $P = P(c, v'(k-1))$ such that $\texttt{dist}(c'(k), v, E^0_{sparse} \cup E_{\text{BFS}} \cup P) = \texttt{dist}(c'(k), v, E^0_{sparse} \cup E_{\text{BFS}})$. We get that

$$\begin{aligned}
&\texttt{dist}(v'(k), v, E^0_{sparse} \cup E_{\text{BFS}}) \\
&\quad \leqslant 1 + \texttt{dist}(c'(k), v, E^0_{sparse} \cup E_{\text{BFS}}) \\
&\quad = 1 + \texttt{dist}(c'(k), v, E^0_{sparse} \cup E_{\text{BFS}} \cup P) \\
&\quad \leqslant 2 + \texttt{dist}(v'(k), v, E^0_{sparse} \cup E_{\text{BFS}} \cup P) \\
&\quad \leqslant 2 \\
&\qquad + \texttt{dist}(v'(k), v'(k-1), E^0_{sparse} \cup E_{\text{BFS}} \cup P) \\
&\qquad + \texttt{dist}(v'(k-1), v, E^0_{sparse} \cup E_{\text{BFS}} \cup P) \\
&\quad \leqslant 2 + \texttt{dist}(v'(k), v'(k-1), G) \\
&\qquad + \texttt{dist}(v'(k-1), v, E^0_{sparse} \cup E_{\text{BFS}}) \\
&\quad \leqslant 2 + \texttt{dist}(v'(k), v'(k-1), G) \\
&\qquad + \texttt{dist}(v'(k-1), v, G) + 2(k-1) \\
&\quad = 2k + \texttt{dist}(v'(k), v, G),
\end{aligned}$$

where the last inequality follows from the inductive hypothesis. We next show that the number of centers adjacent to the considered path is at least halved in each iteration of the procedure, and hence $j \leqslant \log n$, which yields the lemma. Formally, we show that $|\texttt{centers}(P(c, v'(i)), C_{\text{sparse}})| \leqslant |\texttt{centers}(P(c, v'(i-1)), C_{\text{sparse}})|/2$. By definition of $v'(i)$, for every node $x \in \texttt{centers}(P(c, v'(i)), C_{\text{sparse}})$, $\texttt{dist}(x, v, E^0_{sparse} \cup E_{\text{BFS}}) > \texttt{dist}(x, v, E^0_{sparse} \cup E_{\text{BFS}} \cup P(c, v'(i-1)))$. We get that, $\texttt{BFS-Val}(P(c, v'(i-1)), v, \tilde{C}(v), E^0_{sparse} \cup E_{\text{BFS}}) \geqslant |\texttt{centers}(P(c, v'(i)), C_{\text{sparse}})|$. Recall that if the path was not chosen, then by the condition in the procedure, $6 \cdot \texttt{BFS-Val}(P(c, v'(i-1)), v, \tilde{C}(v), E^0_{sparse} \cup E_{\text{BFS}}) < \texttt{cost}(P(c, v'(i-1)), E^0_{sparse} \cup E_{\text{BFS}}) \leqslant 3|\texttt{centers}(P(c, v'(i-1)), C_{\text{sparse}})|$, where the last inequality follows from the fact that every $c \in \texttt{centers}(P(c, v'(i-1)), C_{\text{sparse}})$ can have at most three neighbors in $P(c, v'(i-1))$ since $P(c, v'(i-1))$ is a shortest path. We get that $|\texttt{centers}(P(c, v'(i)), C_{\text{sparse}})| \leqslant |\texttt{centers}(P(c, v'(i-1)), C_{\text{sparse}})|/2$, as required. ∎

Lemma 3.3. *For every $v \in C_{sparse}$, the set E_{BFS} returned by Procedure* **Approximate-BFS** *satisfies $|E_{BFS}| = O(|\tilde{C}(v)|\log n) = O(n^{2\delta}\log n)$.*

Proof: Let E^c_{BFS} be the set E_{BFS} at the beginning of c's iteration of Procedure **Approximate-BFS**. Let $P(c) = P(c, v')$ be the path that was added to E_{BFS} in c's iteration of Procedure **Approximate-BFS**, where $P(c, v')$ can also be empty if $v' = c$. We argue that the cost of adding $P(c)$ is roughly proportional to its benefit. Consider the set $\texttt{X}(c) = \{y \in \texttt{centers}(P(c), C_{\text{sparse}}) \mid \texttt{dist}(y, v, E^0_{sparse} \cup E^c_{\text{BFS}} \cup P(c)) < \texttt{dist}(y, v, E^0_{sparse} \cup E^c_{\text{BFS}})\}$. Note that $\texttt{BFS-Val}(P(c), v, \tilde{C}(v), E^0_{sparse} \cup E^c_{\text{BFS}}) = |\texttt{X}(c)|$. We claim that each node $z \in \tilde{C}(v)$

Procedure **Approximate-BFS**$(v, \tilde{C}(v), E^0_{sparse}, G)$

$E_{\text{BFS}} \leftarrow \varnothing$
For every node $c \in \tilde{C}(v)$ do:
  Set $v' \leftarrow v$ and $ind = false$.
  While $(ind = false)$ do:
    If $6 \cdot \texttt{BFS-Val}(P(c, v'), v, \tilde{C}(v), E^0_{sparse} \cup E_{\text{BFS}}) \geqslant \texttt{cost}(P(c, v'), E^0_{sparse} \cup E_{\text{BFS}})$ then:
      Set $E_{\text{BFS}} \leftarrow E_{\text{BFS}} \cup P(c, v')$
      Set $ind = true$
    Else set $v' \leftarrow \texttt{First-not-Help}(P(c, v'), v, c, \tilde{C}(v), E^0_{sparse} \cup E_{\text{BFS}})$
Return E_{BFS}

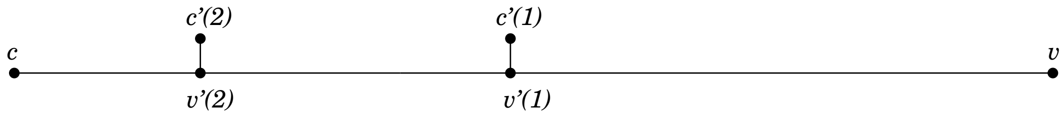

Figure 1: Illustration for Lemma 3.2.

may belong to at most $O(\log n)$ sets $\mathtt{X}(c)$. To see this let $c \in \tilde{C}(v)$ be the first node that was considered in Procedure **Approximate-BFS** and that $z \in \mathtt{X}(c)$. By the analysis of Lemma 3.2, after adding the path $P(c)$ to E_{BFS}, $\texttt{dist}(c, v, E^0_{sparse} \cup E_{\text{BFS}}) \leqslant \texttt{dist}(c, v, G) + 2\log n$. Thus the distance between c and v can improve at most $2\log n$ times. Since $|\tilde{B}(v)| \leqslant n^{3\delta}$, and C_{sparse} contains each node with probability $1/n^{\delta}$, we get that in expectation $|\tilde{C}(v)| \leqslant n^{2\delta}$. Hence, $|E_{\text{BFS}}| \leqslant \sum_{c \in \tilde{C}(v)} \texttt{cost}(P(c), E^0_{sparse} \cup E^c_{\text{BFS}}) \leqslant \sum_{c \in \tilde{C}(v)} O(|\mathtt{X}(c)|) = O(n^{2\delta} \log n)$. ∎

By Lemma 3.2 we have the following.

Corollary 3.1. *Let H' be a $\log n/3$-multiplicative spanner. The set E_{sparse} returned by Procedure* **Sparse** *satisfies the following. For every $v \in V$ such that $|\tilde{B}(v)| \leqslant n^{3\delta}$ and every $x \in \tilde{B}(v)$, $\texttt{dist}(v, x, E_{sparse} \cup H') \leqslant \texttt{dist}(v, x, G) + 3\log n$.*

Proof: Consider a node $v \in V$ such that $|\tilde{B}(v)| \leqslant n^{3\delta}$ and a node $x \in \tilde{B}(v)$. Let $c_v = \texttt{center}(v, C_{\text{sparse}})$ and $c_x = \texttt{center}(x, C_{\text{sparse}})$ (as explained above it is enough to consider the case where both v and x are uncovered and thus c_v and c_x are well defined). If $\texttt{dist}(v, x) \leqslant \mu - 2$ then it is not hard to verify that $c_x \in \tilde{B}(c_v)$. We thus have by Lemma 3.2 that $\texttt{dist}(c_v, c_x, E^0_{sparse} \cup E_{\text{BFS}}) \leqslant \texttt{dist}(c_v, c_x, E_{sparse}) \leqslant \texttt{dist}(c_v, c_x, G) + 2\log n$. Hence, $\texttt{dist}(x, v, E_{sparse}) \leqslant \texttt{dist}(x, c_x, E_{sparse}) + \texttt{dist}(c_x, c_v, E_{sparse}) + \texttt{dist}(c_v, v, E_{sparse}) \leqslant 2 + \texttt{dist}(c_x, c_v, G) + 2\log n \leqslant 4 + \texttt{dist}(x, v, G) + 2\log n \leqslant \texttt{dist}(x, v, G) + 3\log n$. Let us consider now the end case where $\texttt{dist}(v, x) \geqslant \mu - 1$. In this case it might be that $c_x \notin \tilde{B}(c_v)$. Let x' be the node on the path $P(x, v)$ at distance 2 from x. It is not hard to verify that $\texttt{center}(x', C_{\text{sparse}}) \in \tilde{B}(c_v)$ and thus, using similar analysis as above, $\texttt{dist}(x', v, E_{sparse}) \leqslant \texttt{dist}(x', v, G) + 2\log n + 4$. Since H' contains $\log n/3$-multiplicative spanner, we get $\texttt{dist}(x, v, E_{sparse} \cup H') \leqslant \texttt{dist}(x, x', H') + \texttt{dist}(x', v, E_{sparse}) \leqslant 2\log n/3 + \texttt{dist}(x', v, G) + 2\log n + 4 \leqslant \texttt{dist}(x, v, G) + 3\log n$. ∎

Lemma 3.4. *The expected number of edges in E_{sparse} is $O(n^{1+\delta}\log n)$.*

Proof: Consider a node $c \in C_{\text{sparse}} \cap (S_1 \cup S_2)$, i.e., such that $|\tilde{B}(c)| \leqslant n^{3\delta}$. By Lemma 3.3, the number of edges added to E_{sparse} in Procedure **Sparse** in c's iteration is $O(n^{2\delta}\log n)$. The expected number of nodes in C_{sparse} is $O(n^{1-\delta})$. Thus the expected number of edges added to E_{sparse} is $O(n^{1+\delta}\log n)$. ∎

Dense areas. An *r-separated r-dominating* set (or an $r - SD$ for short) for a set of nodes C' is a subset C'' of C' such that all nodes in C'' are at distance at least r from one another and every node in C' has a node at distance at most r from it in C''. Note that such a set always exists and can be constructed greedily, as one can simply consider the nodes in C' one by one and add each node c to the set C'' if none of the nodes already in C'' is at distance r from c. It is not hard to verify that C'' is an $r - SD$ set.

Procedure **Dense** handles dense areas. Procedure **Dense** picks a set of edges E_{dense} such that the set of edges $E' = E_{\text{vs}} \cup E_{sparse} \cup E_{dense}$ satisfies that for every two close nodes x_1 and x_2, their distance in E' is within additive stretch $O(\log n \cdot \mu)$ from the distance in G.

```
Procedure Sparse(G)

E_sparse, E^0_sparse ← ∅
Choose a set of nodes C_sparse by independently sampling at random every node with probability 1/n^δ
For every node v:
   If Γ(v) ∩ C_sparse = ∅ add all incident edges of v to E^0_sparse
   Else do:
      Select center(v, C_sparse) to be some neighbor of v in C_sparse
      Add the edge (v, center(v, C_sparse)) to E^0_sparse
For every node v ∈ C_sparse such that |B̃(v)| ⩽ n^{3δ} do:
   E_BFS ← Approximate-BFS(v, B̃(v) ∩ C_sparse, E^0_sparse, G)
   E_sparse = E_sparse ∪ E_BFS
Return E_sparse ∪ E^0_sparse
```

More precisely, it picks a maximal $(3\mu) - SD$ set C_{rep} for S_3 and a set of edges E_{dense} with the following properties. Every pair of nodes in C_{rep} has "small" additive stretch, in addition, all nodes in S_3 have a node in C_{rep} close to them.

For a center $c \in C_{rep}$, let $Cluster(c)$ be the set of all nodes $v \in V$ such that $\texttt{dist}(c, v) \leqslant 3\mu$ and c is closer to v than all nodes in C_{rep} (recall that we assume uniqueness of the shortest path). Note that all nodes x on the shortest path from $v \in Cluster(c)$ to c satisfy $x \in Cluster(c)$. Note that every node $x \in S_3$ satisfies $x \in Cluster(c)$ for some $c \in C_{rep}$.

For a path P, let $E_{important}(P)$ be the set of edges of P at distance at most 2μ from some node $v \in V(P) \cap S_3$.

For a path P, let $C_P = \{c \in C_{rep} \mid \exists v \in V(P), v \in Cluster(c)\}$.

Formally, Procedure **Dense** operates as follows. First pick a maximal $(3\mu) - SD$ set C_{rep} for S_3. For every node $c \in C_{rep}$ construct a BFS tree on $Cluster(c)$ and add the edges of the tree to the constructed spanner.

Next, the procedure goes over pairs of centers $(c_1, c_2) \in C_{rep}$, and considers adding their path $P = P(c_1, c_2)$ to the spanner. Adding this path will benefit certain sufficiently close pairs of centers from C_P, by reducing their distance. The procedure will refrain from adding the path $P(c_1, c_2)$ to the output spanner if there exists a center $c \in C_P$ such that both (c_1, c) and (c, c_2) have already benefitted from paths that were added to the spanner earlier on. Formally, unmark all pairs (c_1, c_2) such that $c_1, c_2 \in C_{rep}$. For every two centers $c_1, c_2 \in C_{rep}$ do the following. If there there is no node $c \in C_P$ such that both (c_1, c) and (c, c_2) are marked, then add $E_{important}(P(c_1, c_2))$ to the constructed spanner, and mark all pairs (c_1, c) and (c, c_2) such that $c \in C_P$.

Consider two nodes $c_1, c_2 \in C_{rep}$. We say that the path $P(c_1, c_2)$ is purchased by the algorithm if the set of edges $E_{important}(P(c_1, c_2))$ were added to the spanner.

LEMMA 3.5. *$|E_{dense}| \leqslant O(n^{1+\delta})$ for every $3/17 \leqslant \delta$.*

Proof: The sets $Cluster(c)$ for $c \in C_{rep}$ are disjoint. Moreover, since the nodes in C_{rep} are at distance at least 3μ from one another, we get that $\tilde{B}(c) \subseteq Cluster(c)$ for every $c \in C_{rep}$. Recall that $|\tilde{B}(c)| > n^{3\delta}$ for every $c \in C_{rep}$. We thus get that $|C_{rep}| \leqslant n^{1-3\delta}$. Therefore, there are at most $n^{2-6\delta}$ pairs of nodes in C_{rep}. Consider a path $P = P(c_1, c_2)$ that was purchased by Procedure **Dense**. Consider a center $c \in C_P$, let v_1 be the first node (closest to c_1) in P such that $v_1 \in Cluster(c) \cap S_3$ and let v_2 be the last node (closest to c_2) in P such that $v_2 \in Cluster(c) \cap S_3$. We claim that $|P(v_1, v_2)| \leqslant 6\mu$. To see this, note that $\texttt{dist}(v_1, c) \leqslant 3\mu$ and $\texttt{dist}(v_2, c) \leqslant 3\mu$. Therefore, the number of edges in P that are added to $E_{important}(P(c_1, c_2))$ for all nodes $v \in S_3 \cap Cluster(c)$ is $O(\mu)$ (the edges that are at distance 2μ from v_1 or v_2 plus the path $P(v_1, v_2)$). We get that the number of edges in $E_{important}(P)$ is at most $O(\mu|C_P|)$. Notice that the number of pairs in C_{rep} that are marked in Procedure **Dense** after purchasing the path P is at least $|C_P|$ and that every pair is marked once. Let $\mathcal{P}_{\text{dense}}$ be the set of paths that were purchased by Procedure **Dense**. We have

$$\begin{aligned} |E_{dense}| &\leqslant \sum_{P \in \mathcal{P}_{\text{dense}}} |E_{important}(P)| \\ &\leqslant \sum_{P \in \mathcal{P}_{\text{dense}}} O(\mu|C_P|) \leqslant O(\mu n^{2-6\delta}) \\ &\leqslant O(n^{1+\delta}), \end{aligned}$$

where the last inequality holds for every $3/17 \leqslant \delta$.

In addition, a BFS tree on $Cluster(c)$ is constructed and added to E_{dense} for every $c \in C_{rep}$. Note that each

```
Procedure Dense(G)

E_dense ← ∅
Let S_3 be the set of dense nodes, namely, v ∈ V such that |B̃(v)| > n^{3δ}
Let C_rep be a maximal (3μ) − SD set for S_3
For every node v, let ctr(v) be the center c ∈ C_rep closest to v
For every c ∈ C_rep, let Cluster(c) ← {v | ctr(v) = c}
For every node c ∈ C_rep construct a BFS tree on Cluster(c) and add the edges of the tree
to E_dense
Unmark all pairs (c_1, c_2) such that c_1, c_2 ∈ C_rep
For every two centers c_1, c_2 ∈ C_rep do:
   Let C_P = {c ∈ C_rep | ∃v ∈ P(c_1, c_2), v ∈ Cluster(c)}
   If ∄ a center c ∈ C_P such that both (c_1, c) and (c, c_2) are marked then:
      E_important(P(c_1, c_2)) ← {e ∈ P(c_1, c_2) | dist(e, S_3 ∩ P(c_1, c_2)) ⩽ 2μ}
      E_dense ← E_dense ∪ E_important(P(c_1, c_2))
      Mark all pairs (c_1, c) and (c, c_2) such that c ∈ C_P
Return E_dense
```

node belongs to only one cluster and thus at most $O(n)$ edges are added by this step. We thus conclude that the number of edges in E_{dense} is $O(n^{1+\delta})$. ∎

Short Distances. Procedure **Short-distances** handles short distances, namely, pair of nodes s and t such that $|\Gamma(P(s,t),\mu)| \leqslant n^{1-2\delta}$. Procedure **Short-distances** starts by constructing a $(\log n/3)$-multiplicative spanner and adding its edges to the constructed spanner. It then invokes Procedures **Very-sparse**(G), **Sparse**(G) and **Dense**(G) and adds the set of edges returned by these procedures to the constructed spanner. We show that the set of edges E_{short} returned by Procedure **Short-distances** satisfies that the distance for every two close nodes is within additive stretch $O(\mu \log n)$ from the distance in G.

We say that a path P is S_3*-tolerant* if the edges of P that are incident to nodes in S_3 belong to E_{short}. Towards proving the desired additive stretch on short distances, we first prove the following auxiliary lemma. The lemma bounds the additive stretch incurred by certain pairs of nodes x, y in E_{short} by the term

$$\Delta(x,y) = \lceil |\Gamma(P(x,y),\mu)|/(\mu \cdot n^{\delta}) \rceil \cdot 7\log n + \mu \cdot \log n.$$

LEMMA 3.6. *For every two close nodes x_1 and x_2 such that $P(x_1,x_2)$ is S_3-tolerant,*

$$\mathtt{dist}(x_1,x_2,E_{short}) \leqslant \mathtt{dist}(x_1,x_2,G) + \Delta(x_1,x_2).$$

Proof: The proof is by induction on $\mathtt{dist}(x_1,x_2,G)$. If $\mathtt{dist}(x_1,x_2,G) < 3\mu$ then the lemma follows by the fact that E_{short} contains a $(\log n/3)$-multiplicative spanner.

Assume the lemma holds for every two nodes x'_1 and x'_2 such that $\mathtt{dist}(x'_1,x'_2,G) < d$ and consider two nodes x_1 and x_2 such that $\mathtt{dist}(x_1,x_2,G) = d$.

We consider two cases, the first case is where $x_1 \in S_1 \cup S_3$ and the second case is where $x_1 \in S_2$.

First note that every node $x \in P(x_1,x_2) \cap (S_1 \cup S_3)$ such that $\mathtt{dist}(x,x_2) \geqslant 2\mu$ satisfies the following. There exists a node $y \in P(x,x_2)$ such that $y \neq x$ and $P(x,y) \subseteq E_{short}$. In case $x \in S_1$, the claim follows by Lemma 3.1. In case $x \in S_3$, the claim follows by the fact that $P(x_1,x_2)$ is S_3-tolerant.

In Particular, the above observation holds for $x = x_1$, so let y_1 be the node satisfying $y_1 \in P(x_1,x_2)$ and $P(x_1,y_1) \subseteq E_{short}$. Hence $\mathtt{dist}(x_1,y_1,E_{short}) = \mathtt{dist}(x_1,y_1,G)$. By the induction hypothesis we have, $\mathtt{dist}(y_1,x_2,E_{short}) \leqslant \mathtt{dist}(y_1,x_2,G) + \Delta(y_1,x_2)$. We thus get that, $\mathtt{dist}(x_1,x_2,E_{short}) \leqslant \mathtt{dist}(x_1,y_1,E_{short}) + \mathtt{dist}(y_1,x_2,E_{short}) \leqslant \mathtt{dist}(x_1,y_1,G) + \mathtt{dist}(y_1,x_2,G) + \Delta(y_1,x_2) \leqslant \mathtt{dist}(x_1,x_2,G) + \Delta(x_1,x_2)$.

We are left with the case where $x_1 \in S_2$. Let z_1 be the node at distance μ from x_1 on $P(x_1,x_2)$. Again, we handle separately the cases $z_1 \in S_1 \cup S_3$ and $z_1 \in S_2$. If $z_1 \in S_2$ then let z_2 be the node at distance $\mu + 1$ from z_1 on $P(z_1,x_2)$. Note that the additive distortion from x_1 to z_2 is at most $7\log n$. To see this, let y be the node at distance μ from z_1 on $P(z_1,x_2)$. Note that y and z_2 are neighbors. Since $x_1, z_1 \in S_2$, by Lemma 3.1 we have $\mathtt{dist}(x_1,z_1,E_{short}) \leqslant \mathtt{dist}(x_1,z_1,G) + 3\log n$ and $\mathtt{dist}(z_1,y,E_{short}) \leqslant \mathtt{dist}(z_1,y,G) + 3\log n$. In addition, since E_{short} contains a $(\log n/3)$ multiplicative spanner and since $\mathtt{dist}(y,z_2,G) = 1$, we have $\mathtt{dist}(y,z_2,E_{short}) \leqslant \log n/3$. We thus conclude $\mathtt{dist}(x_1,z_2,E_{short}) < \mathtt{dist}(x_1,z_2,G) + 7\log n$.

Otherwise, if $z_1 \in (S_1 \cup S_3)$ then note that the adjacent edge to z_1 in $P(z_1,x_2)$ belongs to E_{short}.

Procedure **Short-distances**(G)

$E_0 \leftarrow (\log n/3)$-multiplicative spanner
$E_1 \leftarrow$ **Very-sparse**(G)
$E_2 \leftarrow$ **Sparse**(G)
$E_3 \leftarrow$ **Dense**(G)
$E_{short} \leftarrow E_0 \cup E_1 \cup E_2 \cup E_3$
Return E_{short}

Let y be the first node on $P(z_1, x_2)$ such that the adjacent edge to y in $P(y, x_2)$ is not in E_{short} and that $\mathtt{dist}(z_1, y) \leqslant \mu$.

If no such node exists then set z_2 to be the node at distance $\mu + 1$ from z_1 on $P(z_1, x_2)$. Note that in this case $P(z_1, z_2) \subseteq E_{short}$. We thus get $\mathtt{dist}(x_1, z_2, E_{short}) \leqslant \mathtt{dist}(x_1, z_1, E_{short}) + \mathtt{dist}(z_1, z_2, E_{short}) \leqslant \mathtt{dist}(x_1, z_1, G) + 3\log n + \mathtt{dist}(z_1, z_2, G) \leqslant \mathtt{dist}(x_1, z_2, G) + 3\log n$. Otherwise, if there exists such node y, set z_2 to be the node at distance μ from y on $P(y, x_2)$. Note that $y \in S_2$. We get that

$$\begin{aligned}
&\mathtt{dist}(x_1, z_2, E_{short}) \\
&\quad \leqslant \mathtt{dist}(x_1, z_1, E_{short}) \\
&\qquad + \mathtt{dist}(z_1, y, E_{short}) + \mathtt{dist}(y, z_2, E_{short}) \\
&\quad \leqslant \mathtt{dist}(x_1, z_1, G) + 3\log n \\
&\qquad + \mathtt{dist}(z_1, y, G) + \mathtt{dist}(y, z_2, G) + 3\log n \\
&\quad = \mathtt{dist}(x_1, z_2, G) + 6 \cdot \log n.
\end{aligned}$$

Moreover, note that in all cases $\mathtt{dist}(x_1, z_2, G) \geqslant 2\mu + 1$. Using shortest path properties, it is not hard to verify that $\Gamma(x_1, \mu, G) \cap \Gamma(P(z_2, x_2), \mu) = \varnothing$. Recalling that $x_1 \in S_2$, we thus have $|\Gamma(P(z_2, x_2), \mu)| \leqslant |\Gamma(P(x_1, x_2), \mu)| - |\Gamma(x_1, \mu, G)| \leqslant |\Gamma(P(x_1, x_2), \mu)| - \mu \cdot n^\delta$. Hence by the induction hypothesis,
$\mathtt{dist}(x_1, x_2, E_{short}) \leqslant \mathtt{dist}(x_1, z_2, E_{short}) + \mathtt{dist}(z_2, x_2, E_{short}) \leqslant \mathtt{dist}(x_1, z_2, G) + 7\log n + \mathtt{dist}(z_2, x_2, G) + \Delta(z_2, x_2) \leqslant \mathtt{dist}(x_1, x_2, G) + \lceil (|\Gamma(P(x_1, x_2), \mu)| - \mu \cdot n^\delta)/(\mu \cdot n^\delta) \rceil \cdot 7\log n + 7\log n + \mu \cdot \log n = \mathtt{dist}(x_1, x_2, G) + \Delta(x_1, x_2)$. ∎

Consider two close nodes x_1 and x_2 on some path $P(c_1, c_2)$ that was purchased by Procedure **Dense**. It is not hard to verify that since $P(c_1, c_2)$ was purchased by Procedure **Dense** then the path $P(c_1, c_2)$ is S_3-tolerant. In addition, every subpath of an S_3-tolerant path is also S_3-tolerant. Hence $P(x_1, x_2)$ is S_3-tolerant and we have the following corollaries.

Corollary 3.2. *For every two close nodes x_1 and x_2 on some path $P(c_1, c_2)$ that was purchased by Procedure* **Dense**,

$$\mathtt{dist}(x_1, x_2, E_{short}) \leqslant \mathtt{dist}(x_1, x_2, G) + \Delta(x_1, x_2).$$

Corollary 3.3. *For every two close centers $c_1, c_2 \in C_{rep}$, if $P(c_1, c_2)$ was purchased by the algorithm, then* $\mathtt{dist}(c_1, c_2, E_{short}) \leqslant \mathtt{dist}(c_1, c_2, G) + 8\mu \cdot \log n$.

Lemma 3.7. *For every two close nodes x_1, x_2 such that $x_1, x_2 \in S_3$,* $\mathtt{dist}(x_1, x_2, E_{short}) \leqslant \mathtt{dist}(x_1, x_2, G) + 17\mu \log n$.

Proof: Consider two close nodes $x_1, x_2 \in S_3$ and let c_1 and c_2 be the centers in C_{rep} such that $x_1 \in Cluster(c_1)$ and $x_2 \in Cluster(c_2)$. Let $d = \mathtt{dist}(x_1, x_2, G)$. We consider two cases, the first case is when the pair (c_1, c_2) is marked by Procedure **Dense** and the second case is when it is not marked.

Consider the first case where the pair (c_1, c_2) is marked. The pair (c_1, c_2) is marked since there was some path $P(c_3, c_4)$ (could be that $P(c_3, c_4) = P(c_1, c_2)$) such that $P(c_3, c_4)$ was purchased by the algorithm and there are two nodes y_1 and y_2 on $P(c_3, c_4)$ such that $y_1 \in Cluster(c_1)$ and $y_2 \in Cluster(c_2)$.

By Corollary 3.2, $\mathtt{dist}(y_1, y_2, E_{short}) \leqslant \mathtt{dist}(y_1, y_2, E_{short}) + 8\mu \cdot \log n$. Since the distance from a node to its cluster center is at most 3μ, we get that

$$\begin{aligned}
&\mathtt{dist}(x_1, x_2, E_{short}) \\
&\quad \leqslant \mathtt{dist}(x_1, y_1, E_{short}) + \mathtt{dist}(y_1, y_2, E_{short}) \\
&\qquad + \mathtt{dist}(y_2, x_2, E_{short}) \\
&\quad \leqslant 6\mu + \mathtt{dist}(y_1, y_2, G) + 8\mu \cdot \log n + 6\mu \\
&\quad \leqslant 6\mu + d + 6\mu + 8\mu \cdot \log n \\
&\quad = d + 12\mu + 8\mu \cdot \log n \leqslant d + 9\mu \cdot \log n,
\end{aligned}$$

where the last inequality holds for every $\log n > 12$. Consider the second case where the pair (c_1, c_2) is not marked. The path $P(c_1, c_2)$ was not purchased by the algorithm since there are a node z on $P(c_1, c_2)$ such that $z \in Cluster(c_3)$ and (c_1, c_3) is marked and (c_2, c_3) is marked.

Using the same analysis as before, we get that $\mathtt{dist}(x_1, z, E_{short}) \leqslant \mathtt{dist}(x_1, z, G) + 12\mu + 8\mu \cdot \log n$ and

$\texttt{dist}(z, x_2, E_{short}) \leqslant \texttt{dist}(z, x_2, G) + 12\mu + 8\mu \cdot \log n$. Thus,

$$\begin{aligned} &\texttt{dist}(x_1, x_2, E_{short}) \\ &\quad \leqslant \texttt{dist}(x_1, z, E_{short}) + \texttt{dist}(z, x_2, E_{short}) \\ &\quad \leqslant \texttt{dist}(x_1, z, G) + 12\mu + 8\mu \cdot \log n \\ &\qquad + \texttt{dist}(z, x_2, G) + 12\mu + 8\mu \cdot \log n \\ &\quad \leqslant \texttt{dist}(x_1, x_2, G) + 17\mu \cdot \log n, \end{aligned}$$

where the last inequality holds for every $\log n > 24$. ∎

LEMMA 3.8. *For every two close nodes x_1, x_2, $\texttt{dist}(x_1, x_2, E_{short}) \leqslant \texttt{dist}(x_1, x_2, G) + 33\mu \log n$.*

Proof: By Lemma 3.6, we have that every shortest path $P(y_1, y_2)$ between two close nodes y_1 and y_2 such that all nodes on $P(y_1, y_2) \setminus \{y_2\}$ are not in S_3, satisfies $\texttt{dist}(y_1, y_2, E_{short}) \leqslant \texttt{dist}(y_1, y_2, G) + \Delta(y_1, y_2) \leqslant \texttt{dist}(y_1, y_2, G) + 8\mu \cdot \log n$.

If $P(x_1, x_2) \setminus \{x_2\} \cap S_3 = \varnothing$, we get $\texttt{dist}(x_1, x_2, E_{short}) \leqslant \texttt{dist}(x_1, x_2, G) + \Delta(x_1, x_2) \leqslant \texttt{dist}(x_1, x_2, G) + 8\mu \log n$. Otherwise, let z_1 (respectively, z_2) be the first (respectively, last) node of S_3 on the path $P(x_1, x_2, G)$ (it could be that $z_1 = z_2$).

By Lemma 3.7, $\texttt{dist}(z_1, z_2, E_{short}) \leqslant \texttt{dist}(z_1, z_2, G) + 17\mu \log n$.

We thus have, $\texttt{dist}(x_1, x_2, E_{short}) \leqslant \texttt{dist}(x_1, z_1, E_{short}) + \texttt{dist}(z_1, z_2, E_{short}) + \texttt{dist}(z_2, x_2, E_{short}) \leqslant \texttt{dist}(x_1, z_1, G) + 8\mu \log n + \texttt{dist}(z_1, z_2, G) + 17\mu \log n + \texttt{dist}(z_2, x_2, G) + 8\mu \log n = \texttt{dist}(x_1, x_2, G) + 33\mu \log n$. ∎

Long Distances. Procedure **Long-distances** handles long distances. More specifically, we show the following. Consider a randomly selected set of vertices R_{long} obtained by taking each node with probability $9 \log n / n^{1-2\delta}$. Procedure **Long-distances** finds a set of edges E_{long} with the following properties. First, the number of edges in E_{long} is $\tilde{O}(n^{1+\delta})$. Second, for every pair of nodes $u, v \in R_{long}$, $\texttt{dist}(u, v, E_{long}) \leqslant \texttt{dist}(u, v, G) + 2$.

Let $cater(P, R)$ for a path P and a set of nodes R be the caterpillar that is obtained by taking the path P and connecting all nodes in $\Gamma(P) \cap R \setminus P$ to the path P by a single edge. Let $\texttt{Gain}(P, R, E')$ denote the set of pairs $\{r_1, r_2\}$ such that $r_1, r_2 \in \Gamma(P) \cap R$ and adding the caterpillar $cater(P, R)$ to E' improves their distance, i.e. $\texttt{dist}(r_1, r_2, E' \cup cater(P, R)) < \texttt{dist}(r_1, r_2, E')$, and let $\texttt{value}(P, R, E') = |\texttt{Gain}(P, R, E')|$.

Formally, Procedure **Long-distances** operates as follows. For every node v with degree at most n^δ, i.e., $|\Gamma(v)| \leqslant n^\delta$, add to E_{long} all edges incident to v. Next, choose a set R_{long} by independently sampling at random every node with probability $9 \log n / n^{1-2\delta}$. For every pair of nodes $\{r_1, r_2\}$ such that $r_1, r_2 \in R_{long}$ do the following. Add $cater(P, R_{long})$ to E_{long} if $4 \cdot \texttt{value}(P, R_{long}, E_{long}) \cdot n^{1-3\delta} > \texttt{cost}(cater(P, R_{long}), E_{long})$, where $P = P(r_1, r_2)$.

```
Procedure Main(G)

H ← Short-distances(G)
      ∪ Long-distances(G)
Return H
```

Here, we say that a node v is heavy if its degree is at least n^δ, namely, $\Gamma(v) \geqslant n^\delta$ and light otherwise. Let $\texttt{heavy\_dist}(P)$ be the number of nodes in P with degree at least n^δ.

Denote by E^i_{long} the subgraph under construction after the i'th iteration of Procedure **Long-distances**, namely, the subgraph E_{long} after considering the first i paths $P(r_1, r_2)$. Note that E^0_{long} contains all incident edges to light nodes. Let P_i be the path considered during the i'th iteration.

Denote by $\mathcal{I}'$ the set of iterations in which Procedure **Long-distances** added the caterpillars considered to the constructed spanner E_{long} and by $\mathcal{P}'$ the set of paths that their caterpillars were added to E_{long} using this process.

LEMMA 3.9. *The expected number of edges added by Procedure* **Long-distances** *is $\tilde{O}(n^{1+\delta})$.*

Proof: In the first step of Procedure **Long-distances** all edges incident to light nodes are added to E_{long}. It is not hard to verify that $O(n^{1+\delta})$ edges are added for this step.

Note that each pair of nodes u and v can belong to some set $\texttt{Gain}(P_i, R_{long}, E_{long})$ that is added to E_{long} in at most 5 iterations. To see this, let $\hat{P}$ be the first path added to E_{long} such that the pair $\{u, v\}$ belongs to $\texttt{Gain}(\hat{P}, R_{long}, E_{long})$. Assume u and v are not in $\hat{P}$, and let u' and v' be the nodes in $\hat{P}$ connected to u and v respectively. Let the distance between u and v be d and the distance between u' and v' be d'. Note that $d' \leqslant d + 2$ and $d \leqslant d' + 2$, we get that the distance between u and v in $cater(\hat{P}, R_{long})$ is at most $d + 4$. We get that the distance between u and v can be improved at most 5 times. If both u and v are in P then the shortest path between u and v in $cater(\hat{P}, R_{long})$ is also the shortest path between them in G, hence the distance between u and v can not improve anymore and the pair $\{u, v\}$ belongs only to $\texttt{Gain}(\hat{P}, R_{long}, E_{long})$. We are left with the case where exactly one of u and v is in $\hat{P}$. In this case, $d' \leqslant d + 1$ and $d \leqslant d' + 1$, we get that the distance between u and v in $cater(\hat{P}, R_{long})$

Procedure **Long-distances**(G)

$E_{long} \leftarrow \varnothing$
For every node v such that $|\Gamma(v)| \leqslant n^{\delta}$, add to E_{long} all edges incident to v.
Choose a set R_{long} by independently sampling at random every node with probability $9 \log n / n^{1-2\delta}$
For every pair of nodes $r_1, r_2 \in R_{long}$ do:
  Let $P = P(r_1, r_2)$
    Add $cater(P, R_{long})$ to E_{long} if $(4 \cdot \texttt{value}(P, R_{long}, E_{long}) \cdot n^{1-3\delta}) \geqslant \texttt{cost}(cater(P, R_{long}), E_{long})$
Return E_{long}

is at most $d+2$, therefore the distance between u and v can improve at most 3 times. This implies that the sum of values in $\mathcal{P}'$ is $\tilde{O}(n^{4\delta})$ as the expected number of nodes in R_{long} is $\tilde{O}(n^{2\delta})$. By the rule used by Procedure **Long-distances** to add $cater(P_i, R_{long})$ to E_{long}, we thus have, $\sum_{i \in \mathcal{I}'} \texttt{cost}(cater(P_i, R_{long}), E^{i-1}_{long}) \leqslant 4 \cdot n^{1-3\delta} \cdot \sum_{i \in \mathcal{I}'} \texttt{value}(P_i, R_{long}, E^{i-1}_{long}) = \tilde{O}(n^{1+\delta})$. The lemma follows. ∎

For a path P, let $R_P = R_{long} \cap \Gamma(P)$.

Towards proving that every pair of nodes in R_{long} is within additive stretch 2 from the distance in G, we first prove the following auxiliary lemmas.

Lemma 3.10. *With high probability, $\Gamma(P(u,v)) \cap R_{long} \neq \varnothing$ for every two nodes u and v such that $|\Gamma(P(u,v))| \geqslant n^{1-2\delta}/3$.*

Proof: Consider two nodes u and v such that $|\Gamma(P(u,v))| \geqslant n^{1-2\delta}/3$. The probability that none of the nodes in $\Gamma(P(u,v))$ were chosen to R_{long} is

$$(1 - \frac{1}{3 \cdot n^{2\delta}})^{9n^{2\delta} \log n} \approx (1/e)^{3 \log n} = \frac{1}{n^3}.$$

By the union bound we get that the probability that there is a pair of nodes u, v such that $|\Gamma(P(u,v))| \geqslant n^{1-2\delta}/3$ and $\Gamma(P(u,v)) \cap R_{long} = \varnothing$ is at most $1/n$. ∎

Lemma 3.11. *Assume that $\Gamma(P(u,v)) \cap R_{long} \neq \varnothing$ for every two nodes u and v such that $|\Gamma(P(u,v))| \geqslant n^{1-2\delta}/3$. Then for every pair of nodes x and y, $\texttt{cost}(cater(P(x,y), R_{long}), E^0_{long}) \leqslant |R_{P(x,y)}| \cdot (n^{1-3\delta} + 3) + n^{1-3\delta}$.*

Proof: First note that $\texttt{cost}(cater(P(x,y), R_{long}), E^0_{long}) \leqslant \texttt{heavy\_dist}(x,y) + |R_{P(x,y)}|$. To see this, recall that E^0_{long} contains all incident edges to light nodes.

We thus need to show that $\texttt{heavy\_dist}(x,y) \leqslant |R_{P(x,y)}| \cdot (n^{1-3\delta} + 2) + n^{1-3\delta}$. We prove by induction on the heavy distance $\texttt{heavy\_dist}(x',y')$ that for every pair of nodes x' and y', $\texttt{heavy\_dist}(x',y') \leqslant |R_{P(x',y')}| \cdot (n^{1-3\delta} + 2) + n^{1-3\delta}$.

If $\texttt{heavy\_dist}(x',y') \leqslant n^{1-3\delta}$, the claim holds trivially.

Assume the claim holds for every pair of nodes $\{x'', y''\}$ such that $\texttt{heavy\_dist}(x'',y'') < d$ and consider pair of nodes $\{x', y'\}$ such that $\texttt{heavy\_dist}(x',y') = d$ for some $d > n^{1-3\delta}$. Let x_1 be the node in $P(x',y')$ such that $\texttt{heavy\_dist}(x',x_1) = n^{1-3\delta}$. Note that $|\Gamma(P(x',x_1))| \geqslant n^{1-2\delta}/3$ and thus $\Gamma(P(x',x_1)) \cap R_{long} \neq \varnothing$. Let x_2 be the node at distance 2 from x_1 on $P(x_1, y')$. By shortest path properties we get that $R_{P(x',x_1)} \cap \Gamma(P(x_2,y')) = \varnothing$. By the induction hypothesis, we have $\texttt{heavy\_dist}(x_2,y) \leqslant |R_{P(x_2,y')}|(n^{1-3\delta}+2) + n^{1-3\delta}$. We thus have, $\texttt{heavy\_dist}(x',y') \leqslant n^{1-3\delta} + 2 + |R_{P(x_2,y')}| \cdot (n^{1-3\delta}+2) + n^{1-3\delta} \leqslant n^{1-3\delta} + 2 + (|R_{P(x',y')}| - |R_{P(x',x_1)}|) \cdot (n^{1-3\delta} + 2) + n^{1-3\delta} \leqslant |R_{P(x',y')}| \cdot (n^{1-3\delta} + 2) + n^{1-3\delta}$. ∎

Corollary 3.4. *With high probability, for every pair of nodes x and y, $\texttt{cost}(cater(P, R_{long}), E^0_{long}) \leqslant |R_{P(x,y)}| \cdot (n^{1-3\delta} + 3) + n^{1-3\delta}$.*

Lemma 3.12. *With high probability, for every pair of nodes $u, v \in R_{long}$, $\texttt{dist}(u,v,E_{long}) \leqslant \texttt{dist}(u,v,G) + 2$.*

Proof: To show the lemma, we need to consider a pair of nodes u and v in R_{long} such that $\texttt{dist}(u,v,E_{long}) > \texttt{dist}(u,v,G)$, namely, that the shortest path $P_i = P(u,v)$ was not added to E_{long}. The path P_i was not added to E_{long} as $4 \cdot \texttt{value}(P_i, E^{i-1}_{long}) \cdot n^{1-3\delta} < \texttt{cost}(cater(P_i, R_{long}), E^{i-1}_{long})$. By Corollary 3.4, $\texttt{cost}(cater(P_i, R_{long}), E^0_{long}) \leqslant |R_{P_i}| \cdot (n^{1-3\delta} + 3) + n^{1-3\delta} \leqslant 2|R_{P_i}| \cdot n^{1-3\delta}$, where the last inequality follows from the fact that $|R_{P_i}| > 1$ (as $u, v \in R_{P_i}$) and straightforward calculations. We thus get $\texttt{value}(P_i) < |R_{P_i}|/2$. Consider all pairs: $A = \{\{s,t\} \mid s \in \{u,v\}, t \in R_{P_i} \text{ and } \texttt{dist}(s,t,P_i) < \texttt{dist}(s,t,E^{i-1}_{long})\}$. By definition $|A| \leqslant \texttt{value}(P_i)$, thus $|A| < \texttt{value}(P_i) < |R_{P_i}|/2$. This implies that there is a node $w \in R_{P_i}$ such that $\texttt{dist}(u,w,E^{i-1}_{long}) \leqslant \texttt{dist}(u,w,cater(P_i,R_{long}))$ and $\texttt{dist}(w,v,E^{i-1}_{long}) \leqslant \texttt{dist}(w,v,cater(P_i,R_{long}))$. Let w' be the node on the path P_i that has an edge to w in $cater(P_i,R_{long})$. Note that $\texttt{dist}(u,v,G) = \texttt{dist}(u,w',G) + \texttt{dist}(w',v,G)$ and that $\texttt{dist}(u,w,cater(P_i,R_{long})) = \texttt{dist}(u,w',G) + 1$ and $\texttt{dist}(w,v,cater(P_i,R_{long})) = \texttt{dist}(w',v,G) + 1$, therefore $\texttt{dist}(u,v,E^{i-1}_{long}) \leqslant \texttt{dist}(u,w,cater(P_i,R_{long})) + \texttt{dist}(w,v,cater(P_i,R_{long})) \leqslant \texttt{dist}(u,w',G) + 1 + \texttt{dist}(w',v,G) + 1 = \texttt{dist}(u,v,G) + 2$. ▮

Putting it all together. Finally, we prove the bound on the additive stretch of the spanner in the following lemma.

Lemma 3.13. *The stretch of the spanner is $O(\mu \log n)$.*

Proof: Consider two nodes s and t that are not close, namely, $|\Gamma(P(s,t),\mu)| > n^{1-2\delta}$. Consider the first (respectively, last) node y_1 (respectively, y_2) on $P(s,t)$ such that $|\Gamma(P(s,y_1),\mu)| \geqslant n^{1-2\delta}$ (respectively, $|\Gamma(P(y_2,t),\mu)| \geqslant n^{1-2\delta}$). By Lemma 3.10 we have, $\Gamma(P(s,y_1),\mu) \cap R_{long} \neq \varnothing$ and $\Gamma(P(y_2,t),\mu) \cap R_{long} \neq \varnothing$. Let z_1 ba a node on the path $P(s,y_1)$ such that there exists a node $r_1 \in R_{long}$ and $\texttt{dist}(z_1,r_1,G) \leqslant \mu$ and let z_2 be the last node on the path $P(y_2,t)$ such that there exists a node $r_2 \in R_{long}$ and $\texttt{dist}(z_2,r_2,G) \leqslant \mu$. Let z'_1 be the neighbor of z_1 on $P(s,z_1)$ and let z'_2 be the neighbor of z_2 on $P(z_2,t)$. We claim that $|\Gamma(P(s,z'_1),\mu)| \leqslant n^{1-2\delta}$ and $|\Gamma(P(z'_2,t),\mu)| \leqslant n^{1-2\delta}$. To see this, note that by definition of y_1, every node x in $P(s,y_1) \setminus \{y_1\}$ satisfies $|\Gamma(P(s,x),\mu)| < n^{1-2\delta}$. Since $z'_1 \in P(s,y_1) \setminus \{y_1\}$ it follows that $|\Gamma(P(s,z'_1),\mu)| \leqslant n^{1-2\delta}$. Similarly, we can show that $|\Gamma(P(z'_2,t),\mu)| \leqslant n^{1-2\delta}$.

By Lemma 3.8, $\texttt{dist}(s,z'_1,H) \leqslant \texttt{dist}(s,z'_1,G) + 33\mu \log n$, $\texttt{dist}(z'_2,t,H) \leqslant \texttt{dist}(z'_2,t,G) + 33\mu \log n$. By Lemma 3.12, $\texttt{dist}(r_1,r_2,H) \leqslant \texttt{dist}(r_1,r_2,G) + 2$. Note also that $\texttt{dist}(r_1,r_2,G) \leqslant \texttt{dist}(r_1,z_1,G) + \texttt{dist}(z_1,z_2,G) + \texttt{dist}(z_2,r_2,G) \leqslant 2\mu + \texttt{dist}(z_1,z_2,G)$. We thus have $\texttt{dist}(r_1,r_2,H) \leqslant \texttt{dist}(z_1,z_2,G) + 2\mu + 2$. In addition, since H contains a $\log n/3$ multiplicative spanner, we get $\texttt{dist}(z'_1,r_1,H) \leqslant \log n/3\texttt{dist}(z'_1,r_1,G) \leqslant \log n/3(\texttt{dist}(z'_1,z_1,G) + \texttt{dist}(z_1,r_1,G)) \leqslant \log n(1+\mu)/3$. Similarly, $\texttt{dist}(z'_2,r_2,H) \leqslant \log n(1+\mu)/3$. Hence $\texttt{dist}(s,t,H) \leqslant \texttt{dist}(s,z'_1,H) + \texttt{dist}(z'_1,r_1,H) + \texttt{dist}(r_1,r_2,H) + \texttt{dist}(r_2,z'_2,H) + \texttt{dist}(z'_2,t,H) \leqslant \texttt{dist}(s,t,G) + O(\mu \log n)$. ▮

3.1 New sublinear distance stretch spanners

We note that it possible to tweak our construction from Section 3 to give a sublinear distance stretch spanner. More precisely, we have the following.

Lemma 3.14. *One can efficiently construct a spanner $\hat{H}$ with $\tilde{O}(n^{1+3/17})$ edges such that for every pair of nodes s,t, $\texttt{dist}(s,t,\hat{H}) \leqslant \texttt{dist}(s,t,G) + \tilde{O}(\sqrt{\texttt{dist}(s,t,G)})$.*

We now sketch the construction (we omit the complete details from this version). The construction involves $\log n$ iterations, where each iteration i handles distances between 2^{i-1} to 2^i, (we stop once $2^{i-1} = n^{1-9/17}$). In Each iteration i invoke Procedure **Short-distances** from Section 3, but use $\sqrt{2^{i-1}}$ instead of μ (in every place that uses μ). Let H_i be the constructed spanner for iteration i. Add the edges of H_i to the constructed spanner $\hat{H}$. To handle distances greater than $n^{1-9/17}$, we simply add the spanner H from Section 3 to the constructed spanner $\hat{H}$.

Following the analysis of Section 3, one can show that for every pair of nodes s,t such that $\texttt{dist}(s,t,G) = O(2^i)$, the additive stretch for the pair s,t in H_i is within additive stretch $O(\sqrt{2^i})$. This handles pairs of nodes of distance at most $n^{1-9/17}$. It is not hard to see that pairs of nodes s,t of distance greater than $n^{1-9/17}$ are satisfied by the spanner H from Section 3.

4 Conclusions

In this paper we make an additional step towards better understanding the picture of purely additive spanners. We present a new simple algorithm for $(1,4)$-additive spanner with $\tilde{O}(n^{7/5})$ edges. In addition, we present a construction for additive spanners with $\tilde{O}(n^{1+\delta})$ edges and additive stretch of $\tilde{O}(n^{1/2-3\delta/2})$ for any $3/17 \leqslant \delta < 1/3$. It would be interesting to extend this result to any $0 < \delta < 1/3$. Our result for spanners of size $o(n^{4/3})$ gives the best additive stretch known so far (for the mentioned range). However, it is unclear that indeed a polynomial stretch is needed. Specifically, a major open problem in this area is the existence of a spanner of size $O(n^{4/3-\varepsilon})$ for some fixed ε with constant or even polylog additive stretch.

Acknowledgement I am very grateful to my advisor, David Peleg, for many helpful discussions and for reviewing this paper.

References

[1] D. Aingworth, C. Chekuri, P. Indyk, and R. Motwani. Fast estimation of diameter and shortest paths (without matrix multiplication). *SIAM J. Comput.*, 28(4):1167–1181, 1999.

[2] I. Althöfer, G. Das, D. Dobkin, D. Joseph, and J. Soares. On sparse spanners of weighted graphs. *Discrete & Computational Geometry*, 9:81–100, 1993.

[3] S. Baswana and T. Kavitha. Faster algorithms for approximate distance oracles and all-pairs small stretch paths. In *Proc. IEEE Symp. on Foundations of Computer Science (FOCS)*, 591–602, 2006.

[4] S. Baswana, T. Kavitha, , K. Mehlhorn, and S. Pettie. New constructions of (α, β)-spanners and purely additive spanners. In *Proc. 16th SODA*, 672-681, 2005.

[5] S. Baswana, T. Kavitha, , K. Mehlhorn, and S. Pettie. Additive spanners and (α, β)-spanners. *ACM Trans. Algo.* 7, A.5, 2010.

[6] B. Bollobás, D. Coppersmith, and M. Elkin. Sparse distance preservers and additive spanners. In *Proc. 14th ACM-SIAM Symp. on Discrete Algorithms (SODA)*, pages 414–423, 2003.

[7] E. Cohen. Fast algorithms for constructing t-spanners and paths with stretch t. *SIAM J. Comput.*, 28:210-236, 1998.

[8] E. Cohen. Polylog-time and near-linear work approximation scheme for undirected shortest-paths. *J.ACM*, 47:132-166, 2000.

[9] L. J. Cowen. Compact routing with minimum stretch. *J. Algor.*, 28, 170-183, 2001.

[10] L. J. Cowen and C. G. Wagner. Compact roundtrip routing in directed networks. *J. Algor. 50, 1*, 79-95, 2004.

[11] D. Dor, S. Halperin, and U. Zwick. All-pairs almost shortest paths. *SIAM J. Comput.*, 29(5):1740–1759, 2000.

[12] M. Elkin. Computing almost shortest paths. *ACM Trans. Algorithms*, 1(2):283–323, 2005.

[13] M. Elkin. A near-optimal distributed fully dynamic algorithm for maintaining sparse spanners. In *Proc. 26th ACM Symp. on Principles of Distributed Computing (PODC)*, pages 185–194, 2007.

[14] M. Elkin and D. Peleg. $(1+\epsilon, \beta)$-spanner constructions for general graphs. In*Proc. 33rd ACM Symp. on Theory of Computing (STOC)*, 173-182, 2001.

[15] M. Elkin and D. Peleg. $(1+\epsilon, \beta)$-spanner constructions for general graphs. *SIAM J. Comput.*, 33(3):608-631, 2004.

[16] M. Elkin and J. Zhang. Efficient algorithms for constructing $(1+\epsilon, \beta)$-spanners in the distributed and streaming models. In *Proc. 23rd ACM Symp. on Principles of Distributed Computing (PODC)*, pages 160–168, 2004.

[17] P. Erdős. Extremal problems in graph theory. In *Theory of graphs and its applications*, pages 29-36, 1964.

[18] A. M. Farley, A. Proskurowski, D. Zappala, and K. Windisch. Spanners and message distribution in networks. *Discrete Applied Mathematics*, 137(2):159–171, 2004.

[19] D. Peleg and A.A. Schäffer. Graph spanners. *J. Graph Theory* 13, 99–116, 1989.

[20] D. Peleg and J. D. Ullman. An optimal synchronizer for the hypercube. *SIAM J. Comput.*, 18(4):740–747, 1989.

[21] D. Peleg and E. Upfal. A trade-off between space and efficiency for routing tables. *J. ACM*, 36(3):510—530, 1989.

[22] Seth Pettie. Low distortion spanners. In *34th International Colloquium on Automata, Languages and Programming (ICALP)*, 78–89, 2007.

[23] S. Pettie. Low distortion spanners. *ACM Trans. on Algorithms* 6(1), 2009.

[24] L. Roditty, M. Thorup, and U. Zwick. Deterministic constructions of approximate distance oracles and spanners. In *Proc. 32nd Int. Colloq. on Automata, Languages & Prog.*, 261–272, 2005.

[25] L. Roditty, M. Thorup, and U. Zwick. Roundtrip spanners and roundtrip routing in directed graphs. *ACM Trans. Algorithms*, 3(4):Article 29, 2008.

[26] M. Thorup and U. Zwick. Compact routing schemes. In *Proc. 13th ACM Symp. on Parallel Algorithms and Architectures (SPAA)*, 1–10, 2001.

[27] M. Thorup and U. Zwick. Approximate distance oracles. *J. ACM*, 52(1):1–24, 2005.

[28] M. Thorup and U. Zwick. Spanners and emulators with sublinear distance errors. In *17th Symp. on Discrete Algorithms (SODA)*, 802–809. ACM-SIAM, 2006.

[29] R. Wenger. Extremal graphs with no C^4s, C^6s, or C^{10}s. *J. Combin. Theory Ser. B*, 52(1):113-116, 1991.

[30] D. P. Woodruff. Lower bounds for additive spanners, emulators, and more. In *Proc. 47th IEEE Symp. on Foundations of Computer Science (FOCS)*, pages 389–398, 2006.

[31] D. P. Woodruff. Additive Spanners in Nearly Quadratic Time. In *37th International Colloquium on Automata, Languages and Programming (ICALP)*, 463–474, 2010.