Spanners and emulators with sublinear distance errors

Mikkel Thorup * *Uri Zwick* †

Abstract

Let $k \geq 2$ be an integer. We show that any undirected and unweighted graph $G = (V, E)$ on n vertices has a subgraph $G' = (V, E')$ with $O(kn^{1+1/k})$ edges such that for any two vertices $u, v \in V$, if $\delta_G(u, v) = d$, then $\delta_{G'}(u, v) = d + O(d^{1-\frac{1}{k-1}})$. Furthermore, we show that such subgraphs can be constructed in $O(mn^{1/k})$ time, where m and n are the number of edges and vertices in the original graph. We also show that it is possible to construct a *weighted* graph $G^* = (V, E^*)$ with $O(kn^{1+1/(2^k-1)})$ edges such that for every $u, v \in V$, if $\delta_G(u, v) = d$, then $d \leq \delta_{G^*}(u, v) = d + O(d^{1-\frac{1}{k-1}})$. These are the first such results with additive error terms of the form $o(d)$, i.e., additive error terms that are sublinear in the distance being approximated.

1 Introduction

Let $G = (V, E)$ be an undirected graph. For two vertices $u, v \in V$, we let $\delta_G(u, v)$ denote the distance between u and v in G. A subgraph $G' = (V, E')$ of G is said to be a *t-spanner* of G if and only if for every $u, v \in V$ we have $\delta_{G'}(u, v) \leq t \cdot \delta_G(u, v)$. The parameter t is said to be the *stretch* factor of the spanner. The notion of spanners is implicit in [Awe85]. It was formally defined in [PS89]. It is known that for any integer $k \geq 1$, any undirected, possibly weighted, graph on n vertices has a $(2k-1)$-spanner with $O(n^{1+1/k})$ edges, and that this is essentially the optimal tradeoff between size and stretch (see [PS89], [ADD+93]). Baswana and Sen [BS03], improving a result of [TZ05], obtained a randomized linear time algorithm for constructing $(2k-1)$-spanner with $O(kn^{1+1/k})$ edges. Finally, a deterministic linear time algorithm for constructing such spanners appears in [RTZ05].

Spanners are useful in solving various distributed computing problems, such a network synchronization [Awe85], and routing [PU89, AP92, EGP98, Cow01, TZ01], for computing approximate shortest paths [ABCP99, Coh99] and constructing approximate distance oracles [TZ05]. They are also an appealing mathematical concept in their own right.

Spanners, as defined above, approximate distances with a *multiplicative* error. Is it possible to obtain similar results with an *additive* error? Perhaps surprisingly, the answer for *unweighted* graphs is yes! It is shown in [DHZ00], using ideas of [ACIM99], that any graph $G = (V, E)$ on n vertices has a subgraph $G' = (V, E')$ with $\tilde{O}(n^{3/2})$ edges such that $\delta_{G'}(u, v) \leq \delta_G(u, v) + 2$, for every $u, v \in V$. Recently, Baswana *et al.* [BKMP05] have shown that any graph $G = (V, E)$ on n vertices has a subgraph $G' = (V, E')$ with $O(n^{4/3})$ edges such that $\delta_{G'}(u, v) \leq \delta_G(u, v) + 6$, for every $u, v \in V$. It is a major open problem whether there exist sparser additive spanners.

It is also shown in [DHZ00] that for any unweighted graph $G = (V, E)$ on n vertices it is possible to construct a *weighted* graph $G^* = (V, E^*)$ with $\tilde{O}(n^{4/3})$ edges such that $\delta_G(u, v) \leq \delta_{G^*}(u, v) \leq \delta_G(u, v) + 4$, for every $u, v \in V$. Note that G^* is a weighted graphs and that it usually contains edges not present in the

*AT&T Labs - Research, 180 Park Avenue, Florham Park, NJ 07932, USA. E-mail address: `mthorup@research.att.com`.
†School of Computer Science, Tel Aviv University, Tel Aviv 69978, Israel. E-mail: `zwick@cs.tau.ac.il`.

original graph. The graph G^* is said to be a *4-emulator* of G. It is again an open problem whether there exist sparser emulators.

Another notion of spanners is introduced in [EP04]. They say that a subgraph $G' = (V, E')$ is an (a, b)-spanner of an unweighted undirected graph $G = (V, E)$ if and only if $\delta_{G'}(u, v) \le a \cdot \delta_G(u, v) + b$, for every $u, v \in V$. Thus, a conventional t-spanner is a $(t, 0)$-spanner, while an additive t-spanner is a $(1, t)$-spanner. Elkin and Peleg [EP04] show that for every $\epsilon, \delta > 0$, there exists $\beta = \beta(\epsilon, \delta)$ such any n-vertex graph has a $(1+\epsilon, \beta)$-spanner with $O(\beta n^{1+\delta})$ edges. (More specifically, $\beta(\epsilon, \delta) = 2^{(\log\frac{1}{\delta}-1)(\log\log\frac{1}{\delta}+\log\frac{1}{\epsilon})}$.) Elkin [Elk01] obtains a faster construction of spanners with slightly worse parameters. The constructions presented in [EP04, Elk01] are fairly complicated.

Bollobás *et al.* [BCE03], introduce the notion of d-preservers. A subgraph $G' = (V, E')$ of $G = (V, E)$ is said to be a *d-preserver* of G if and only if $\delta_{G'}(u, v) = \delta_G(u, v)$, for every $u, v \in V$ such that $\delta_G(u, v) \ge d$. They show that any n-vertex graph has a d-preserver with $O(n^2/d)$ edges. Combining this result with the constructions of [EP04, Elk01], they get that for any integer $k \ge 1$, any graph n-vertex graph $G = (V, E)$ has a subgraph $G' = (V, E')$ such that $\delta_{G'}(u, v) \le \delta_G(u, v) + O(2^k n^{1-2/k})$, for every $u, v \in V$. Finally Coppersmith and Elkin [CE05] show that for every set $P \subseteq V \times V$ of vertex pairs of an n-vertex undirected graph $G = (V, E)$ there exists a subgraph $G' = (V, E')$ with $O(\min\{n^{1/2}p, np^{1/2}\})$ edges, where $p = |P|$, that preserves all the distances between the pairs of vertices in P.

We obtain for the first time arbitrarily sparse subgraphs with an additive error terms that sublinear in the *distance* being approximated. Our constructions are very simple. They also provide alternative derivations of the results of Elkin and Peleg [EP04] and Elkin [Elk01], as well as some other results.

The rest of the paper is organized as follows. In the next two Sections we describe two different constructions that show that for any integer $k \ge 2$, any unweighted undirected graph $G = (V, E)$ on n vertices has a subgraph $G' = (V, E')$ with $O(kn^{1+1/k})$ edges such that for every two vertices $u, v \in V$, if $\delta_G(u, v) = d$, then $\delta_{G'}(u, v) = d + O(d^{1-\frac{1}{k-1}})$. The first construction, presented in Section 2, is new. It provides the simplest possible setting for presenting our new ideas. A disadvantage of this first construction, however, is that the construction of the subgraph $G' = (V, E')$ with the required properties, at least using a naive implementation, requires $O(mn)$ time. A second construction, which is as simple as the first construction, but with a slightly more complicated analysis, is presented in Section 3. The advantage of this second construction is that it can be carried out in $O(mn^{1/k})$ time, where $m = |E|$ is the number of edges in the original graphs. The second construction is actually not new. We show that multiplicative spanner construction algorithm of [TZ05], when applied to *unweighted* graphs, actually constructs subgraphs with the required properties. A new analysis is needed, however, to show that.

In Section 4 we show that for any integer $k \ge 2$, and any unweighted undirected graph $G = (V, E)$ on n vertices there exists a *weighted* graph $G = (V, E^*)$ with $O(kn^{1+1/(2^k-1)})$ edges such that for every two vertices $u, v \in V$, if $\delta_G(u, v) = d$, then $d \le \delta_{G'}(u, v) = d + O(d^{1-\frac{1}{k-1}})$. Note that by allowing the use of weighted edges that are not present in the original graph the number of edges needed to obtain an error term of the form $O(d^{1-\frac{1}{k-1}})$ is drastically reduced from $O(kn^{1+1/k})$ to $O(kn^{1+1/(2^k-1)})$. We end in Section 5 with some concluding remarks and open problems.

2 First spanner construction

We start by choosing a sequence $A_0 \supseteq A_1 \supseteq \cdots \supseteq A_{k-1} \supseteq A_k$ of vertex sets as follows. We let $A_0 = V$. For $1 \le i < k$, we obtain A_i by picking each vertex of A_{i-1}, independently, with probability $n^{-1/k}$. We let $A_k = \phi$. If $v \in V$ is a vertex, then the distance $\delta(v, A_i)$ from v to A_i is naturally defined as

$$\delta(v, A_i) \;=\; \min\{\, \delta(v, w) \mid w \in A_i \} \;.$$

```
SPAN_1(G, k):

A_0 = V; A_k = φ
for i ← 1 to k − 1 do
   // Get A_i by picking each vertex of A_{i−1},
   // independently, with probability n^{−1/k}.
   A_i ← SAMPLE(A_{i−1}, n^{−1/k})

for i ← 1 to k − 1 do
   for each v ∈ A_i − A_{i+1} do
      // Construct a BFS search tree rooted at v.
      // Stop at the first level containing vertices from A_{i+1}.
      // Out of the A_{i+1} vertices reached, choose the one with
      // the smallest index and add it to the tree.
      E(v) ← BFS(G, v, A_{i+1})

return (V, ∪_{v∈V} E(v))
```

Figure 1: Constructing a sparse subgraph that approximates all distances well.

As $A_k = \phi$, we let $\delta(v, A_k) = \infty$. For $v \in V$ and $0 \leq i < k$, we let $p_i(v)$ be a vertex of A_i that satisfies $\delta(v, p_i(v)) = \delta(v, A_i)$. (If there are several such vertices, we take the one with the smallest serial number.) We now define the open and closed *balls*, *Ball*(v) and *Ball*$[v]$, respectively, of a vertex $v \in A_i - A_{i+1}$, where $0 \leq i < k$, as follows:

$$\begin{aligned} Ball(v) &= \{\, w \in V \mid \delta(v, w) < \delta(v, A_{i+1}) \,\} \quad , \quad \text{if } v \in A_i - A_{i+1} \; , \\ Ball[v] &= Ball(v) \cup \{p_{i+1}(v)\} \quad , \quad \text{if } v \in A_i - A_{i+1} \; . \end{aligned}$$

Note that if $v \in A_{k-1}$, then $Ball(v) = V$. (This follows as $A_k = \phi$ and $\delta(v, A_k) = \infty$.) We also define then $Ball[v] = V$. It is easy to see that if $v \in A_i - A_{i+1}$, where $0 \leq i < k-1$, then $Ball(v) \cap A_{i+1} = \{p_{i+1}(v)\}$. It is also easy to check that if $u \in Ball[v]$ and p is a shortest path from u to v, then all vertices on the path p are also in $Ball[v]$. We let $Tree(v) = (Ball[v], E(v))$ be a tree of shortest paths from v to all other vertices of $Ball[v]$. Obviously, $E(v)$ contains only $|Ball[v]| - 1$ edges. The sparse subgraph of G that approximates well all distances in G is then $G' = (V, E')$, where $E' = \cup_{v \in V} E(v)$. A pseudo-code of the algorithm used to construct this subgraph is given in Figure 1.

Lemma 2.1 *The expected number of edges in the subgraph $G' = (V, E')$ returned by the call $SPAN_1(G, k)$ is $O(kn^{1+1/k})$.*

Proof: Consider a vertex $v \in A_i - A_{i+1}$. Each vertex $u \in V$ becomes an element of A_{i+1}, independently, with probability $n^{-(i+1)/k}$. It follows easily then that $E[|Ball(v)|] = n^{(i+1)/k}$. As $E[|A_i|] = n^{1-i/k}$, we get that the expected number of edges in the subgraph constructed is at most $O(kn^{1+k})$, as required. □

Theorem 2.2 *Let G' is a subgraph of G returned by the call $SPAN_1(G, k)$. Then, for every $u, v \in V$, if $\delta_G(u, v) = d$, then $\delta_{G'}(u, v) \leq d + 4(2 + \lceil d^{1/(k-1)} \rceil)^{k-2}$. In other words, $\delta_{G'}(u, v) = d + O(d^{1-\frac{1}{k-1}})$.*

Proof: For brevity, we use $\delta(u, v) = \delta_G(u, v)$ to denote distances in G, and $\delta'(u, v) = \delta_{G'}(u, v)$ to denote distances in G'. We show that for every two vertices $u, v \in V$ such that $\delta(u, v) = d$ and every $\epsilon > 0$, we have $\delta'(u, v) \leq (1 + \epsilon)d + 2\lceil 2 + \frac{2}{\epsilon} \rceil^{k-2}$. The choice $\epsilon = 2d^{-1/(k-1)}$ would then yield the claim of the theorem. To prove the bound $\delta'(u, v) \leq (1 + \epsilon)d + 2\lceil 2 + \frac{2}{\epsilon} \rceil^{k-2}$, we use a path-finding strategy described

```
FIND-PATH_1(u, v):
if u = v then return
i ← level(u)
w ← FAR(u, v)
if w = v or δ(u, w) ≥ x_i then
   WALK(u, w)
   FIND-PATH_1(w, v)
else
   u' ← p_{i+1}(u)
   WALK(u, u')
   FIND-PATH_1(u', v)
```

Figure 2: The path finding strategy used in the proof of Theorem 2.2.

below. Before describing the path-finding strategy, it is useful to define the notion of a *farthest reachable* vertex in G' on the way to of a given vertex:

Definition 2.3 ($WAY(u,v)$ **and** $FAR(u,v)$**)** *Let* $G=(V,E)$ *be a graph and let* $G'=(V,E')$ *be a subgraph of* G*. Let* $u,v \in V$*. Let* $WAY(u,v)=\{w \in V \mid \delta'(u,w)+\delta(w,v)=\delta(u,v)\}$ *be the set of vertices for which there is a shortest path from* u *to* v *in* G *that pass through* w *and for which* $\delta'(u,w)=\delta(u,w)$*. Informaly, we say that the vertices of* $WAY(u,v)$ *are* on the way *from* u *to* v*. Let* $FAR(u,v)$ *be a vertex* $w \in WAY(u,v)$ *which is farthest away from* u*, i.e., a vertex* $w \in WAY(u,v)$ *for which* $\delta'(u,w)=\delta(u,w)$ *is as large as possible. (Note that* $u \in WAY(u,v)$*, so* $FAR(u,v)$ *is well defined.)*

Note that the definitions of $WAY(u,v)$ and $FAR(u,v)$ depend on the G and the subgraph G' whose identity will always be clear from the context.

Lemma 2.4 *Let* $u \in A_i - A_{i+1}$*, where* $0 \le i < k-1$*,* $v \in V$ *and let* $w = FAR(u,v)$*. If* $w \ne v$*, then* $\delta'(u,p_{i+1}(u)) \le \delta(u,w)+1$*.*

Proof: Let p be a shortest path in G from u to v that passes through w. Let w' be the vertex that follows w on p. As $w' \notin WAY(u,v)$, we have $\delta'(u,w') > \delta(u,w')$. Thus, $w' \notin Ball(u)$. It follows, that $\delta'(u,p_{i+1}(u)) = \delta(u,p_{i+1}(u)) \le \delta(u,w') = \delta(u,w)+1$, as required. □

Our path-finding strategy finds a short path from u to v in the subgraph G' as follows. Suppose that $u \in A_i - A_{i+1}$, where $1 \le i < k$. Let $w = FAR(u,v)$. Let x_i be a parameter to be chosen later. The x_i's will grow exponentially in i. If $\delta'(u,w)=\delta(u,w) \ge x_i$, or if $w=v$, we use a shortest path of G' to walk from u and w. If $w=v$, we are done. If $w \ne v$, we apply the path-finding strategy recursively to find a short path from w to v. Otherwise, if $\delta'(u,w)=\delta(u,w) < x_i$, we use a shortest path of G' to walk from u to $u'=p_{i+1}(u)$. By Lemma 2.4, we have $\delta'(u,u') \le \delta(u,w)+1 \le x_i$. We then again use the path-finding strategy to find a short path from u' to v. The intuition behind this strategy is the following: if we cannot move far enough along a shortest path to v, we are better off heading to the nearest vertex of the next level, from which we will hopefully be able to make much more progress. Note that such 'diversions' can occur at most $k-2$ times, as when a vertex from A_{k-1} is reached, its ball contains all the vertices of the graph. An formal description of this path-finding strategy is given in Figure 2. For every vertex $u \in V$, we let *level*(u) be the index $0 \leq i < k$ such that $u \in A_i - A_{i+1}$.

The parameters x_i, for $0 \leq i < k$, used by the path-finding strategy are set as follows. We let $y_i = \lceil 2 + \frac{2}{\epsilon} \rceil^i$, for $0 \leq i \leq k-2$, and then $x_i = y_i - y_{i-1}$, for $1 \leq i \leq k-2$. We also let $x_0 = 1$ and $x_{k-1} = \infty$.

Lemma 2.5 *Let $u, v \in V$ and suppose that $\delta(u,v) = d$. Then, the path-finding strategy finds a path from u to v in G' whose length is at most $(1+\epsilon)d + 2\lceil 2 + \frac{2}{\epsilon} \rceil^{k-2}$.*

Proof: We prove the lemma by induction on d. The lemma clearly holds for $d = 0$. Assume the lemma holds for all values smaller than d. Consider the sequence $u = u_0, u_1, \ldots, u_i$, where $u_j \in A_j$, for $1 \leq j \leq i$, of centers reached before we could finally reach v, or make at least x_i steps in its direction. Note that $i \leq k-1$, as the spanner contains shortest paths trees rooted at all the vertices of A_{k-1}. By Lemma 2.4, we then have $\delta'(u_j, u_{j+1}) \leq x_j$, and therefore $\delta(u, u_i) \leq \sum_{j=0}^{i-1} \delta'(u_j, u_{j+1}) \leq x_0 + x_1 + \ldots + x_{i-1} = y_{i-1}$. We also have $\delta(u_i, v) \leq \delta(u_i, u_0) + \delta(u_0, v) \leq y_{i-1} + d$.

Suppose, at first, that the destination v is reached by taking at most x_i steps from u_i. Then the path in G' returned by the path-finding strategy is of length at most $d + 2y_{i-1} \leq d + 2y_{k-2} = d + 2\lceil 1 + \frac{2}{\epsilon} \rceil^{k-2}$, as required.

Consider now the more interesting case in which we have made $x'_i \geq x_i$ steps in the direction of v from u_i and reached a vertex $w \neq v$. The total length of the path followed from u to w is then at most $y_{i-1} + x'_i$, and the distance to v decreased by at least $x'_i - y_{i-1}$. To see that, note that as $w \in \mathit{WAY}(u_i, v)$, we have $\delta(w,v) = \delta(u_i, v) - x'_i$. As $\delta(u_i, v) \leq \delta(u_i, u) + \delta(u, v) \leq y_{i-1} + d$, we get that $\delta(w, v) \leq (y_{i-1} + d) - x'_i = d - (x'_i - y_{i-1})$, as claimed.

By the induction hypothesis, the subgraph G' contains a path from w to v of length at most $(1+\epsilon)(d - (x'_i - y_{i-1})) + 2\lceil 2 + \frac{2}{\epsilon} \rceil^{k-2}$. Together with the path of length at most $y_{i-1} + x'_i$ from u to w, we get a path of length at most $(1+\epsilon)(d - (x'_i - y_{i-1})) + (y_{i-1} + x'_i) + 2\lceil 2 + \frac{2}{\epsilon} \rceil^{k-2}$ from u to v.

All that remains, therefore, is to check that $(1+\epsilon)(d - (x'_i - y_{i-1})) + (y_{i-1} + x'_i) \leq (1+\epsilon)d$. This is equivalent to $y_{i-1} + x'_i \leq (1+\epsilon)(x'_i - y_{i-1})$ which is in turn equivalent to $(2+\epsilon)y_{i-1} \leq \epsilon x'_i$. As $x'_i \geq x_i$, it is enough to check this condition for $x'_i = x_i$. As $x_i = y_i - y_{i-1}$, the condition becomes $y_i \geq \frac{2(1+\epsilon)}{\epsilon} y_{i-1}$, which is exactly the property the sequence y_i was chosen to satisfy. □

This completes the proof of the Theorem. □

It is interesting to note that Theorem 2.2 implies the result obtained by Elkin and Peleg [EP04]. Namely, that for every $\epsilon, \delta > 0$, there exists $\beta = \beta(\epsilon, \delta)$ such any n-vertex graph has a $(1+\epsilon, \beta)$-spanner with $O(\beta n^{1+\delta})$ edges. To see that just use $k = \lceil 1/\delta \rceil$ and $\beta(\epsilon, \delta) = 2\lceil 2 + \frac{2}{\epsilon} \rceil^{k-2}$. Lemma 2.5 then says that G' is an $(1+\epsilon, \beta)$-spanner of G. Theorem 2.2 is, however, stronger than the result of [EP04], as the *same* graph is an $(1+\epsilon, \beta)$-spanner of G, for any $\epsilon > 0$. The parameter ϵ is not used in the construction of our subgraph G', only in the analysis. (See also the discussion after the proof of Theorem 4.1.)

A naive implementation of the above construction takes $O(mn)$ time. Can we get a similar construction with a faster running time? This question is answered in the next Section.

3 Second spanner construction

The construction of the previous section is very simple, but not as efficient as we would like it to be. Borrowing ideas from the analysis of the construction of the previous section, we can show that our 'good

$$SPAN_1(G,k): \quad E' \leftarrow \bigcup_{v\in V} \textit{SP-Tree}(\textit{Ball}[v], v)$$
$$SPAN_2(G,k): \quad E' \leftarrow \bigcup_{v\in V} \textit{SP-Tree}(\textit{Clust}(v), v)$$

Figure 3: A concise description and comparison of the spanner construction algorithm of Section 2, and the algorithm of [TZ05] used in Section 3.

old' multiplicative spanners from [TZ05], which we already know how to construct efficiently, give even slightly better sublinear distance errors when applied to unweighted graphs.

Let us briefly review the spanners constructed in [TZ05]. We start, as in the previous section, by defining a hierarchy of centers $A_0 \supseteq A_1 \supseteq \cdots \supseteq A_{k-1} \supseteq A_k$. We let $A_0 = V$. For $1 \le i < k$, we obtain A_i by picking each vertex of A_{i-1}, independently, with probability $n^{-1/k}$. We let $A_k = \phi$. Instead of defining balls, we now define *clusters*. The cluster, *Clust*(v), of a vertex $v \in A_i - A_{i+1}$ is defined as follows:

$$\textit{Clust}(v) \;=\; \{w \in V \mid \delta(v,w) < \delta(w, A_{i+1})\} \quad , \quad \text{if } v \in A_i - A_{i+1} \,.$$

The definition of clusters is very similar to the definition of balls. (They differ in fact by only one character!) Yet, balls and clusters have quite different properties. In particular, it is shown in [TZ05] that the clusters of all the vertices can be found in $O(mn^{1/k})$ time.

One property that clusters do have in common with balls is the following: If $w \in \textit{Clust}(v)$ and p is a shortest path in G from v to w, then all the vertices on p are also in *Clust*(v). That means that we can construct a tree of shortest paths for *Clust*(v) rooted at v. The spanners of [TZ05] are simply composed of a tree of shortest paths of *Clust*(v), rooted at v, for every $v \in G$. The constructions of the spanners of the previous Section, and the spanners of [TZ05] are compared in Figure 3. We use $\textit{SPAN}_2(G,k)$ to refer to the spanner construction algorithm of [TZ05] and use *SP-Tree*(S,v) to denote a tree of shortest paths of the set S rooted at v. Note that the only difference between the two constructions is that balls are used in the first, while clusters are used in the second. The following theorem summarizes some of the results of [TZ05]:

Theorem 3.1 ([TZ05]) *Let $G = (V,E)$ be a graph on $n = |V|$ vertices with $m = |E|$ edges, Let $G' = (V,E')$ be the subgraph returned by the call $\textit{SPAN}_2(G,k)$. Then, G' is a $(2k-1)$-spanner of G. The expected number of edges in G' is $O(kn^{1+1/k})$. Furthermore, algorithm $\textit{SPAN}_2(G,k)$ can be implemented to run in $O(mn^{1/k})$ time.*

Here we show that the subgraph $G' = (V,E')$ generated by $\textit{SPAN}_2(G,k)$ also gives sublinear distance errors:

Theorem 3.2 *Let $G = (V,E)$ be a graph on $n = |V|$ vertices and let $k \ge 2$ be an integer. Let $G' = (V,E')$ be the subgraph returned by the call $\textit{SPAN}_2(G,k)$. Then, for every $u,v \in V$, if $\delta_G(u,v) = d$, then $\delta_{G'}(u,v) \le d + 4(1 + \lceil d^{1/(k-1)} \rceil)^{k-2}$. In other words, $\delta_{G'}(u,v) = d + O(d^{1-\frac{1}{k-1}})$.*

Proof: The proof is similar to the proof of Theorem 2.2, with some subtle differences. As before, we use $\delta(u,v) = \delta_G(u,v)$ to denote distances in G, and $\delta'(u,v) = \delta_{G'}(u,v)$ to denote distances in G'. We show that for every two vertices $u,v \in V$ such that $\delta(u,v) = d$ and every $\epsilon > 0$, we have $\delta'(u,v) \le (1+\epsilon)d + 2\lceil 1 + \frac{2}{\epsilon} \rceil^{k-2}$. The choice $\epsilon = 2d^{-1/(k-1)}$ would then yield the claim of the theorem. To prove the bound $\delta'(u,v) \le (1+\epsilon)d + 2\lceil 1 + \frac{2}{\epsilon} \rceil^{k-2}$, we use the path-find strategy *FIND-PATH*$_2(u,v)$ given in Figure 4.

The path-finding strategy *FIND-PATH*$_2(u, v)$ again uses an exponentially growing sequence of integers x_i to be defined below. It finds a relatively short path in G' from u to v in the following way. Assume that *level*$(u) = i$, i.e., $u \in A_i - A_{i+1}$. Let $w = FAR(u, v)$. The path constructed by *FIND-PATH*$_2(u, v)$ starts by walking from u to w along a shortest path of G'. (Note that this is different from the behavior of *FIND-PATH*$_1(u, v)$ which walks to w only if $w = v$ or $\delta'(u, w) = \delta(u, w) \geq x_i$.) If $w = v$, we are done. If $w \neq v$ but $\delta'(u, w) \geq x_i$, we apply the path-finding strategy recursively from w. If $w \neq v$ and $\delta'(u, w) = \delta(u, w) < x_i$, then the path generated by *FIND-PATH*$_2(u, v)$ continues by taking a shortest path in G' from w to a vertex $u' = DIVERT(w, v) \in A_{i+1}$, to be defined below, and then using the path generated by a recursive call to the path-finding strategy from u'. In many cases, but not all, the vertex $u' = DIVERT(w, v)$ is $p_{i+1}(w)$. To define $DIVERT(w, v)$, we need the following Lemma which is somewhat analogous, though more complicated, than Lemma 2.4:

Lemma 3.3 *Let $u \in A_i - A_{i+1}$, where $0 \leq i < k - 1$, let $v \in V$ and let $w = FAR(u, v)$. If $w \neq v$, then there exists a vertex $u' \in A_{i+1}$, which we denote by $DIVERT(w, v)$, such that $\delta'(w, u') \leq \delta(u, w) + 2$ and either (i) $\delta(u', v) \leq \delta(u, v)$, or (ii) $u' \in A_{i+2}$ and $\delta(u', v) \leq \delta(u, v) + 2$.*

Proof: Let p be a shortest path from u to v in G that passes through w. Let w' be the vertex that follows w on p. As $w' \notin WAY(u, v)$, we have $\delta'(u, w') > \delta(u, w')$. It follows that $w' \notin Clust(u)$, and therefore, there exists a vertex $u_{i+1} \in A_{i+1}$ such that $\delta(w', u_{i+1}) \leq \delta(u, w')$. Thus $\delta(u_{i+1}, v) \leq \delta(u, v)$. Also $\delta(w, u_{i+1}) \leq 1 + \delta(w', u_{i+1}) \leq 1 + \delta(u, w') \leq \delta(u, w) + 2$. Thus, if $\delta'(w, u_{i+1}) = \delta(w, u_{i+1})$, then we can take $u' = u_{i+1}$ and condition (i) is satisfied. Otherwise, we have $w \notin Clust(u_{i+1})$. Thus, there exists a vertex $u_{i+2} \in A_{i+2}$ such that $\delta(w, u_{i+2}) \leq \delta(w, u_{i+1})$. (For simplicity, we assume that $u_{i+1} \in A_{i+1} - A_{i+2}$. The argument can be easily generalized.) Let u_{i+2} be a vertex with a highest level number satisfying the condition $\delta(w, u_{i+2}) \leq \delta(w, u_{i+1})$. It follows then that $w \in Clust(u_{i+2})$ and therefore $\delta'(w, u_{i+2}) = \delta(w, u_{i+2})$. We now have $\delta(u_{i+2}, v) \leq \delta(u_{i+2}, w) + \delta(w, v) \leq (\delta(u, w) + 2) + \delta(w, v) = \delta(u, v) + 2$. We can therefore take $u' = u_{i+2}$ and condition (ii) is satisfied. $\Box$

We return to the proof of Theorem 3.2. The parameters x_i, for $0 \leq i < k$, used by the path-finding strategy are set as follows. We let $y_i = \lceil 1 + \frac{2}{\epsilon} \rceil^i$, for $0 \leq i \leq k - 2$, and then $x_i = y_i - y_{i-1}$, for $1 \leq i \leq k - 2$. We also let $x_0 = 1$ and $x_{k-1} = \infty$. We are now ready to prove:

Lemma 3.4 *Let $u, v \in V$ and suppose that $\delta(u, v) = d$. Then FIND-PATH$_2(u, v)$ finds a path from u to v in G' whose length is at most $(1 + \epsilon)d + 2\lceil 1 + \frac{2}{\epsilon} \rceil^{k-2}$.*

Proof: We prove the lemma by induction on d. The lemma clearly holds for $d = 0$. Assume the lemma holds for all values smaller than d. Consider the sequence $u = u_0, u_1, \ldots, u_i$, where $u_j \in A_j$, for $1 \leq j \leq i$, of centers reached before we could finally reach v, or directly make at least x_i steps in its direction. Note that $i \leq k-1$, as the spanner contains shortest paths trees rooted at all the vertices of A_{k-1}. Let us assume that all the vertices $u_1, \ldots, u_i$ satisfy condition (i) of Lemma 3.3. (The other case is much more favorable for us, as it amounts to 'jumping' two levels of the hierarchy at almost no extra cost. The simple, but somewhat technical, verification of this claim is omitted from this extended abstract.) Let $w_j = FAR(u_j, v)$, for $0 \leq j \leq i$. Thus $\delta'(u_j, w_j) = \delta(u_j, w_j) \leq x_j - 1$, for $0 \leq j < i$, and $x'_i = \delta'(u_i, w_i) \geq x_i$, or $w_i = v$. We have $\delta'(w_j, u_{j+1}) \leq \delta(u_j, w_j) + 2 \leq x_j + 1$, for $0 \leq j < i$. Thus, $\delta'(u_j, u_{j+1}) \leq \delta(u_j, w_j) + \delta(w_j, u_{j+1}) \leq (x_j - 1) + (x_j + 1) = 2x_j$. We therefore have $\delta'(u, u_i) \leq \sum_{j=0}^{i-1} \delta'(u_j, u_{j+1} \leq 2(x_0 + x_1 + \ldots + x_{i-1}) = 2y_{i-1}$. By condition (i) we also have $\delta(u_i, v) \leq \delta(u, v) = d$.

If $w_i = v$, then the path found from u to v is of length at most $2y_{i-1} + \delta(u_i, v) \leq d + 2y_{k-2} = d + 2\lceil 1 + \frac{2}{\epsilon} \rceil^{k-2}$, as required. Suppose therefore that we have made $x'_i \geq x_i$ steps in the direction of v from u_i and reached a vertex $w_i \neq v$. The total length of the path used is at most $2y_{i-1} + x'_i$. Also $\delta(w_i, v) \leq \delta(u, v) - x'_i$. Thus,

> *FIND-PATH*$_2$**(u, v)**:
>
> if $u = v$ then return
>
> $i \leftarrow \mathit{level}(u)$
> $w \leftarrow \mathit{FAR}(u, v)$
> $\mathit{WALK}(u, w)$
>
> if $w = v$ then return
>
> if $\delta(u, w) \geq x_i$ then
>   *FIND-PATH*$_2(w, v)$
> else
> $u' \leftarrow \mathit{DIVERT}(w, v)$
> $\mathit{WALK}(w, u')$
>   *FIND-PATH*$_2(u', v)$

Figure 4: The path-finding strategy used in the proof of Theorem 3.2.

by the induction hypothesis, *FIND-PATH*$_2(w_i, v)$ finds a path of length at most $(1+\epsilon)(d-x'_i)+y_{k-2}$ in G' from w_i to v. Concatenating this path to the path found from u to w_i, we get a path of length at most $(1+\epsilon)(d-x'_i)+(2y_{i-1}+x'_i)+2y_{k-2}$ in G' from u to v.

All that remains, therefore, is to check that $(1+\epsilon)(d-x'_i)+(2y_{i-1}+x'_i) \leq (1+\epsilon)d$. This is easily seen to be equivalent to $2y_{i-1} \leq \epsilon x'_i$ which is indeed satisfied as $x'_i \geq x_i = y_i - y_{i-1}$ and $y_i \geq (1+\frac{2}{\epsilon})y_{i-1}$. □

This completes the proof of the Theorem. □

4 Emulator construction

In the previous two sections we constructed sparse subgraphs of an unweighted undirected graph $G = (V, E)$ that approximated all distances with a sublinear error term in the distance. In this section we show that even smaller error terms are possible, using the same number of edges, if instead of being restricted to choosing a subset of the edges of the graph, we are allowed to construct an arbitrary *weighted* graph $G^* = (V, E^*)$ on the vertices of the original graph G. Following [DHZ00], we call such graphs *emulators*.

An algorithm for constructing a sparse emulator of a given graph $G = (V, E)$ is given in Figure 5. The parameter k again determines the number of levels in the center hierarchy, and the tradeoff between the size and the accuracy. We again use a hierarchy $A_0 \supseteq A_1 \supseteq \cdots \supseteq A_{k-1}$ of k non-empty levels. We again let $A_k = \phi$. The hierarchy is, however, chosen in a different way this time. We let $\nu = 1/(2^k - 1)$ and start again with $A_0 = V$. Each element of A_i then, independently, becomes an element of A_{i+1} with probability $|A_i|/n^{1+\nu}$. Thus $E[|A_{i+1}|] = |A_i|^2/n^{1+\nu}$, for $0 \leq i < k$. It thus follows easily by induction that $E[|A_i|] = n^{1-(2^i-1)\nu}$, for $0 \leq i < k$. By the definition of ν, we then get that $E[|A_{k-1}|] = n^{(1+\nu)/2}$.

For every vertex $u \in A_i$, we let $p_{i+1}(u)$ be a vertex of A_{i+1} closest to u. (This is what meant by the command $p_{i+1}(u) \leftarrow \mathit{CLOSEST}(A_{i+1}, u)$.) We then define a *ball* $B_i(u)$ composed of all the vertices of A_i that are closer to u then $p_{i+1}(u)$. The essential difference with respect to the balls defined in Section 2 is that the ball of a vertex of A_i is now a subset of A_i. The closed ball $B_i[u]$ is obtained by adding $p_{i+1}(u)$ to $B_i(u)$. Direct weighted edges from u to all the other vertices of $B_i[u]$ are added to the emulator. (This is accomplished by the command $E^* \leftarrow E^* \cup (\{u\} \times B_i[u])$. The set E^* is the edge set of the emulator.)

$EMUL(G,k)$:
$\nu = 1/(2^k - 1)$
$A_0 \leftarrow V \; ; \; E^* \leftarrow \phi$
for $i \leftarrow 0$ to $k-2$
    $A_{i+1} \leftarrow SAMPLE(A_i, |A_i|/n^{1+\nu})$
    for every $u \in A_i$
        $p_{i+1}(u) \leftarrow CLOSEST(A_{i+1}, u)$
        $B_i(u) \leftarrow \{ v \in A_i \mid \delta(u,v) < \delta(u, A_{i+1}) \}$
        $B_i[u] = B_i(u) \cup \{ p_{i+1}(u) \}$
        $E^* \leftarrow E^* \cup (\{u\} \times B_i[u])$
    end-for
end-for
$E^* \leftarrow E^* \cup (A_{k-1} \times A_{k-1})$

Figure 5: Constructing a sparse emulator of G.

The weight of every edge $(u,v) \in E^*$ added to the spanner is always $\delta(u,v)$, the distance from u to v in G. Finally direct edges are added between any pair of vertices of A_{k-1}. We now claim:

Theorem 4.1 *Let $G = (V,E)$ be an unweighted graph, and $k \geq 2$ be an integer. Let $G^* = (V,E^*)$ be the graph returned by a call to EMUL(G,k). Then, the expected number of edges in $G^* = (V,E)$ is $O(k\, n^{1+1/(2^k-1)})$. For every $u,v \in V$ with $\delta(u,v) = d$, we have $d \leq \delta^*(u,v) = d + O(kd^{1-1/(k-1)})$.*

Proof: The $O(k\, n^{1+1/(2^k-1)})$ bound on the expected number of edges in the graph $G^* = (V,E^*)$ is obtained using standard arguments. The proof is omitted from this extended abstract.

In the sequel, we let $\delta(u,v) = \delta_G(u,v)$ denote the distance from u to v in G, and $\delta^*(u,v) = \delta_{G^*}(u,v)$ denote the distance from u to v in the emulator G^*. As the weight of each edge $(u,v) \in E^*$ added to the emulator is $\delta(u,v)$, we clearly have $\delta^*(u,v) \geq \delta(u,v)$ for every $u,v \in V$. The interesting part of the proof, of course, is showing that for every $u,v \in V$ with $\delta(u,v) = d$, we have $\delta^*(u,v) = d + O(kd^{1-1/(k-1)})$. We start by proving the following lemma:

Lemma 4.2 *Let $\Delta \geq 1$ be an integer. Then, for every $0 \leq i < k$ and every $u,v \in V$ such that $\delta(u,v) \leq \Delta^i$, either*

$$\delta^*(u,v) \;\leq\; \delta(u,v) + (\Delta+4)^i - \Delta^i \tag{1}$$

or, there exists $u_{i+1} \in A_{i+1}$ such that

$$\delta^*(u,u_{i+1}) \;\leq\; (\Delta+4)^i \,. \tag{2}$$

Proof: The proof is by induction on i. For $i = 0$, we have to show that if $\delta(u,v) \leq 1$, i.e., if $(u,v) \in E$, then either $\delta^*(u,v) = \delta(u,v)$ or there exists $u_1 \in A_1$ such that $\delta^*(u,u_1) \leq 1$. Indeed, if $\delta^*(u,v) > \delta(u,v)$ then $(u,v) \notin E^*$ and therefore $v \notin B_0[u]$. (We assume that $u \in A_0 - A_1$, as otherwise the claim trivially holds by taking $u_1 = u$.) Let $u_1 = p_1(u)$. Then, $\delta^*(u,u_1) = \delta(u,u_1) \leq \delta(u,v) = 1$, as required.

We next show that if the claim holds for i, then it also holds for $i+1$. Let $u, v \in V$ be such that $\delta(u,v) \le \Delta^{i+1}$. Let p be a shortest path in G from u to v. Let us break p into at most Δ segments each of length at most Δ^i. (This can always be done as the graph is unweighted.)

If condition (1) holds for all these segments, then

$$\delta^*(u,v) \;\le\; \delta(u,v) + \Delta((\Delta+4)^i - \Delta^i) \;<\; \delta(u,v) + (\Delta+4)^{i+1} - \Delta^{i+1} \;,$$

as required.

Suppose, therefore, condition (1) does not hold for at least one of the segments of p. Let u_0 be the start vertex of the *first* segment of p for which condition (1) does not hold, and let v_0 be the end vertex of the *last* segment of p for which condition (1) does not hold. (Note that v_0 is not necessarily the last vertex on the segment beginning with u_0.) By the induction hypothesis, condition (2) holds for both these segments, so we get that there exist $u_{i+1}, v_{i+1} \in A_{i+1}$ such that both

$$\delta^*(u_0, u_{i+1}) \;\le\; (\Delta+4)^i \;,$$
$$\delta^*(v_0, v_{i+1}) \;\le\; (\Delta+4)^i \;.$$

By the triangle inequality

$$\delta(u_{i+1}, v_{i+1}) \;\le\; \delta(u_{i+1}, u_0) + \delta(u_0, v_0) + \delta(v_0, v_{i+1}) \;\le\; \delta(u_0, v_0) + 2(\Delta+4)^i \;.$$

There are now two cases. If $(u_{i+1}, v_{i+1}) \in E^*$, then $\delta^*(u_{i+1}, v_{i+1}) = \delta(u_{i+1}, v_{i+1})$, and then

$$\begin{aligned}
\delta^*(u,v) &\le \delta^*(u,u_0) + \delta^*(u_0,u_{i+1}) + \delta^*(u_{i+1},v_{i+1}) + \delta^*(v_{i+1},v_0) + \delta^*(v_0,v) \\
&\le \delta^*(u,u_0) + (\Delta+4)^i + (\delta(u_0,v_0) + 2(\Delta+4)^i) + (\Delta+4)^i + \delta^*(v_0,v) \\
&= \delta(u,v) + \Delta\left((\Delta+4)^i - \Delta^i\right) + 4(\Delta+4)^i \\
&= \delta(u,v) + (\Delta+4)^{i+1} - \Delta^{i+1} \;,
\end{aligned}$$

as required. To justify the transition from the second to the third line, note that $\Delta\left((\Delta+4)^i - \Delta^i\right)$ bounds the total error accumulated from (1) over the at most Δ segments connecting u to u_0 and v_0 and v.

Finally, suppose that $(u_{i+1}, v_{i+1}) \notin E^*$. Let $u_{i+2} = p_{i+2}(u_{i+1}) \in A_{i+2}$. As $v_{i+1} \notin B_{i+1}(u_{i+1})$, we get that $\delta(u_{i+1}, u_{i+2}) \le \delta(u_{i+1}, v_{i+1})$. Now,

$$\delta^*(u, u_{i+2}) \;\le\; \delta^*(u,u_0) + \delta^*(u_0,u_{i+1}) + \delta^*(u_{i+1},u_{i+2}) \;\le\; \delta^*(u,u_0) + \delta^*(u_0,u_{i+1}) + \delta(u_{i+1},v_{i+1}) \;.$$

This expression is smaller than the expression we bounded above. (We even had there the extra terms $\delta^*(v_{i+1}, v_0) + \delta^*(v_0, v)$.) Thus, it is also at most $\delta(u,v) + (\Delta+4)^{i+1} - \Delta^{i+1}$, which is at most $(\Delta+4)^{i+1}$ as $\delta(u,v) \le \Delta^{i+1}$. This completes the proof of the lemma. □

We now return to the proof of Theorem 4.1. We apply Lemma 4.2 with $i = k-1$ and $\Delta = \lceil d^{1/(k-1)} \rceil$. As $A_k = \phi$, condition (1) must hold, and therefore

$$\delta^*(u,v) \;\le\; d + (\Delta+4)^{k-1} - \Delta^{k-1} \;\le\; (d^{1/(k-1)} + 5)^{k-1} \;=\; d + O(k\, d^{1-1/(k-1)}) \;,$$

as desired □

The emulator construction of this Section can be used to obtain yet another proof of the result of Elkin and Peleg [EP04]. For any fixed $\epsilon > 0$, we can convert the emulator $G^* = (V, E^*)$ produced by a call to *EMUL*(G,k), where $k = \lceil 1/\delta \rceil$, into an $(1+\epsilon, \beta)$ spanner by replacing every edge of G^* of length at most $1/\epsilon$ by a path of at most $1/\epsilon$ edges. The graph obtained in this way does depend on ϵ, but the value obtained for $\beta = \beta(\epsilon, \delta)$ is essentially the same as that of [EP04]. (The value of $\beta(\epsilon,\delta)$ obtained using the argument given at the end of Section 2 is larger, but the result obtained there is stronger as the same subgraph was used for all values of ϵ.)

5 Concluding remarks and open problems

For every $k \geq 2$, we showed that any undirected unweighted graph $G = (V, E)$ on n vertices has a subgraph $G' = (V, E')$ with $O(kn^{1+1/k})$ edges such that for every $u, v \in V$, if $\delta_G(u, v) = d$, then $\delta_{G'}(u, v) = d + O(d^{1-\frac{1}{k-1}})$. We in fact showed that our multiplicative spanners from [TZ05], which can be constructed in $O(mn^{1/k})$ time, have these properties. (Here $m = |E|$ is the number of edges in the original graph.) We also showed that there always exists a weighted graph $G = (V, E^*)$ with $O(kn^{1+1/(2^k-1)})$ edges such that for every $u, v \in V$, if $\delta_G(u, v) = d$, then $d \leq \delta_{G'}(u, v) = d + O(d^{1-\frac{1}{k-1}})$. The construction of these graphs is also quite simple.

Our main objective in this paper was to present the simplest possible constructions with $o(d)$ additive error terms. Previous constructions with $o(n^{4/3})$ edges had additive terms that were either of the form ϵd, for some $\epsilon > 0$, or $o(n)$.

Seth Pettie (private communication) was recently able to combine our techniques with the techniques of [BKMP05] and [CE05] to obtain constructions with slightly smaller error terms. The constructions are more involved, however, and the error terms obtained are still of the form $d^{1-\Theta(1/k)}$. It seems that major new ideas will be needed to get significantly smaller, and in particular, constant, error terms.

Another intriguing open problem is whether it is possible to turn our constructions into approximate distance oracles with additive error terms.

References

[ABCP99] B. Awerbuch, B. Berger, L. Cowen, and D. Peleg. Near-linear time construction of sparse neighborhood covers. *SIAM Journal on Computing*, 28:263–277, 1999.

[ACIM99] D. Aingworth, C. Chekuri, P. Indyk, and R. Motwani. Fast estimation of diameter and shortest paths (without matrix multiplication). *SIAM Journal on Computing*, 28:1167–1181, 1999.

[ADD+93] I. Althöfer, G. Das, D. Dobkin, D. Joseph, and J. Soares. On sparse spanners of weighted graphs. *Discrete & Computational Geometry*, 9:81–100, 1993.

[AP92] B. Awerbuch and D. Peleg. Routing with polynomial communication-space trade-off. *SIAM Journal on Discrete Mathematics*, 5(2):151–162, 1992.

[Awe85] B. Awerbuch. Complexity of network synchronization. *Journal of the ACM*, 32:804–823, 1985.

[BCE03] B. Bollobás, D. Coppersmith, and M. Elkin. Sparse distance preservers and additive spanners. In *Proc. of 14th SODA*, pages 414–423, 2003.

[BKMP05] S. Baswana, T. Kavitha, K. Mehlhorn, and S. Pettie. New constructions of (α, β)-spanners and purely additive spanners. In *Proc. of 16th SODA*, pages 672–681, 2005.

[BS03] S. Baswana and S. Sen. A simple linear time algorithm for computing $(2k-1)$-spanner of $O(n^{1+1/k})$ size for weighted graphs. In *Proc. of 30th ICALP*, pages 384–296, 2003.

[CE05] D. Coppersmith and M. Elkin. Sparse source-wise and pair-wise distance preservers. In *Proc. of 16th SODA*, pages 660–669, 2005.

[Coh99] E. Cohen. Fast algorithms for constructing t-spanners and paths with stretch t. *SIAM Journal on Computing*, 28:210–236, 1999.

[Cow01] L.J. Cowen. Compact routing with minimum stretch. *Journal of Algorithms*, 38:170–183, 2001.

[DHZ00] D. Dor, S. Halperin, and U. Zwick. All pairs almost shortest paths. *SIAM Journal on Computing*, 29:1740–1759, 2000.

[EGP98] T. Eilam, C. Gavoille, and D. Peleg. Compact routing schemes with low stretch factor. In *Proc. of 17th PODC*, pages 11–20, 1998.

[Elk01] M. Elkin. Computing almost shortest paths. In *Proc. of 20th PODC*, pages 53–62, 2001.

[EP04] M.L. Elkin and D. Peleg. $(1+\epsilon, \beta)$-Spanner constructions for general graphs. *SIAM Journal on Computing*, 33(3):608–631, 2004. (Announced in STOC'01).

[PS89] D. Peleg and A.A. Schäffer. Graph spanners. *Journal of Graph Theory*, 13:99–116, 1989.

[PU89] D. Peleg and E. Upfal. A trade-off between space and efficiency for routing tables. *Journal of the ACM*, 36(3):510–530, 1989.

[RTZ05] L. Roditty, M. Thorup, and U. Zwick. Deterministic constructions of approximate distance oracles and spanners. In *Proc. of 32th ICALP*, 2005.

[TZ01] M. Thorup and U. Zwick. Compact routing schemes. In *Porceedings of the 13th SPAA*, pages 1–10, 2001.

[TZ05] M. Thorup and U. Zwick. Approximate distance oracles. *Journal of the ACM*, 52(1):1–24, 2005. (Announced in STOC'01).