Downloaded 02/10/16 to 171.64.66.62. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

Better Distance Preservers and Additive Spanners*

Greg Bodwin and Virginia Vassilevska Williams

{gbodwin, virgi}@cs.stanford.edu

Computer Science Department, Stanford University

Abstract

We make improvements to the upper bounds on several popular types of distance preserving graph sketches. The first part of our paper concerns *pairwise distance preservers*, which are sparse subgraphs that exactly preserve the pairwise distances for a set of given pairs of vertices. Our main result here is that all unweighted, undirected n-node graphs G and all pair sets P have distance preservers on $|H| = O(n^{2/3}|P|^{2/3} + n|P|^{1/3})$ edges. This improves the known bounds whenever $|P| = \omega(n^{3/4})$.

We then develop a new graph clustering technique, based on distance preservers, and we apply this technique to show new upper bounds for additive *(standard) spanners*, in which *all* pairwise distances must be preserved up to an additive error function, and for *subset spanners*, in which only distances within a given node subset must be preserved up to an error function. For both of these objects, we obtain the new best tradeoff between spanner sparsity and error allowance in the regime where the error is polynomial in the graph size.

We leave open a conjecture that $O(n^{2/3}|P|^{2/3} + n)$ pairwise distance preservers are possible for undirected unweighted graphs. Resolving this conjecture in the affirmative would improve and simplify our upper bounds for all the graph sketches mentioned above.

1 Introduction

How much can all graphs be compressed while keeping their distance information roughly intact? This question falls within the scope of both metric embeddings and graph theory and is fundamental to our understanding of the metric properties of graphs. When the compressed version of the graph must be a subgraph, it is called a *spanner*. Spanners have a multitude of applications, essentially everywhere shortest path information needs to be compressed while still allowing for graph algorithms to be run. The quality of a spanner is measured by the tradeoff between its sparsity and its accuracy in preserving the distances. There are many different versions of spanners, which we discuss below.

1.1 Pairwise Distance Preservers

One possible formalization of the spanner problem is that distances must be preserved exactly. Naturally, if *all* distances must be preserved exactly, then one cannot sparsify the graph at all (e.g. consider the complete graph). Hence, the most studied version in the exact distance setting is that only *some* of the pairwise distances are considered.

Definition 1. *Given a graph G and a pair set $P \subset V \times V$, we say that a subgraph $H \subset G$ is a* pairwise distance preserver *of G, P if $\delta_H(u, v) = \delta_G(u, v)$ for all $(u, v) \in P$.*

This definition was first posed by Bollobás, Coppersmith, and Elkin [BCE03], who described the pair set implicitly as $\{(u, v) \mid \delta_G(u, v) \geq D\}$ for some parameter D (such an object is simply called a D-preserver of G). The same authors showed that $|E(H)| = \Theta(n^2/D)$ edges are sufficient and sometimes necessary to construct a D-preserver. Coppersmith & Elkin [CE06] later generalized the definition to the above form. They constructions of pairwise distance preservers with $O(n|P|^{1/2})$ edges (which apply to possibly directed and weighted graphs) and $O(n + n^{1/2}|P|)$ edges (which apply only to undirected, but possibly weighted graphs).

*This work was supported by NSF Grants CCF-1417238, CCF-1528078 and CCF-1514339, and BSF Grant BSF:2012338.

Copyright © by SIAM.
Unauthorized reproduction of this article is prohibited.

Downloaded 02/10/16 to 171.64.66.62. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

They also proved a host of lower bounds; most notably that a superlinear ($\omega(n+|P|)$) number of edges are necessary for any distance preserver unless $|P| = O(n^{1/2})$ or $|P| = \Omega(n^2)$. This lower bound holds even for undirected and unweighted graphs. This implies that for distance preservers for $|P| = \Theta(\sqrt{n})$ pairs of nodes, $\Theta(n)$ edges is both an upper and lower bound.

Distance preservers are fundamental combinatorial objects with many applications. For example, they are commonly used as a tool in creating other types of graph spanners [CE06, BCE03, BW15] (we will discuss some of these shortly). Additionally, they were recently applied by Elkin & Pettie [EP15] to construct low-stretch path reporting distance oracles. Although they have been successfully applied to several other important problems, no progress on upper or lower bounds for distance preservers themselves has been reported since Coppersmith & Elkin's initial work ten years ago. This paper provides the first such progress.

Theorem 3. *All undirected unweighted graphs G and pair sets P have a distance preserver H with $|E(H)| = O(n^{2/3}|P|^{2/3} + n|P|^{1/3})$.*

Following this result, the best upper bounds for undirected unweighted graphs are:

1. $O(n^{2/3}|P|^{2/3})$ when $|P| = \Omega(n)$ (this paper)
2. $O(n|P|^{1/3})$ when $\Omega(n^{3/4}) = |P| = O(n)$ (this paper)
3. $O(n + n^{1/2}|P|)$ when $|P| = O(n^{3/4})$ ([CE06])

We consider it fairly unlikely that this piecewise behavior reflects the true upper bound for undirected unweighted pairwise distance preservers. Note that the upper bound $O(n + n^{2/3}|P|^{2/3})$ is proven for both $|P| = \Omega(n)$ and for $|P| = O(n^{1/2})$. We take this as compelling evidence that this bound is attainable in general.

Conjecture 1. *Every unweighted, undirected graph G and pair set P admits a pairwise distance preserver on $O(n + n^{2/3}|P|^{2/3})$ edges.*

1.2 Graph Clustering

In many applications it is useful to cover the graph with clusters of small radius and small overlap. To this end, a rich body of work has developed a variety of graph clustering techniques following the following principles: each cluster consists of a central "core" plus a surrounding shell of non-core nodes, every node belongs to the core of at least one cluster, and the average node belongs to very few clusters, and, typically, each cluster has approximately the same radius. Just a few of the clustering algorithms with this sort of behavior can be found in [AP92, Coh93, PR10].

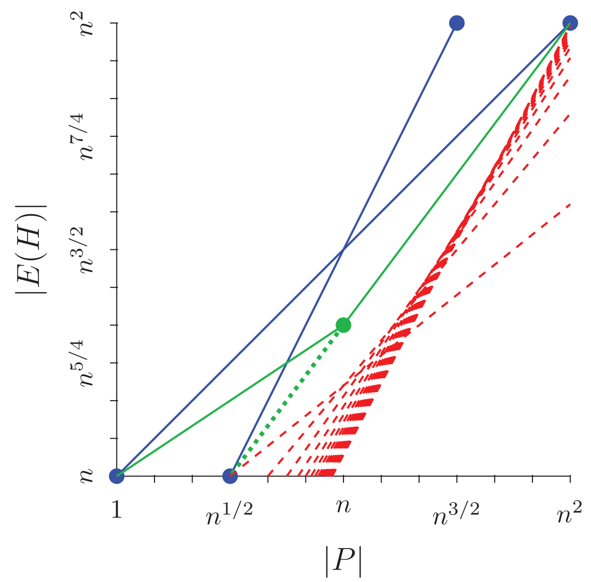


Figure 1: The state of the art after this paper for pairwise distance preservers on undirected unweighted graphs. Old upper bounds (due to Coppersmith & Elkin [CE06]) are in blue, new upper bounds in this paper are in solid green, and our conjectured upper bound is shown by the dotted green line. The dashed red lines are an infinite family of lower bounds due to Coppersmith & Elkin [CE06]; any tradeoff southeast of any of these lines is not possible in general.

What these algorithms commonly lack is a nontrivial bound on the size of each cluster. This makes them difficult to use for certain applications, particularly those related to spanners with *additive error* (called additive spanners).

Our contribution is a new clustering algorithm based on distance preservers, that allows us to have a handle of the number of clusters, and can be applied to constructing additive spanners. Our approach is roughly as follows. We threshold the size of each cluster. Clusters that are smaller than the threshold are called "small," and we use arguments based on distance preserver upper bounds to show that very few edges participate in shortest paths passing through the core of a small cluster. Clusters bigger than the threshold are called "large," and we can limit the total number of

Copyright © by SIAM.
Unauthorized reproduction of this article is prohibited.

Downloaded 02/10/16 to 171.64.66.62. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

large clusters due to the lower bound on the number of nodes each one contains.

Although our underlying clustering technique is similar to prior clustering techniques, our applications to additive spanners require additional properties that do not seem to hold in any prior clustering algorithm. In particular, we need that the "core" of each cluster is contained in a ball of radius r around a center node, and that the "fringe" (non-core nodes) of the cluster *contains* the ball of radius $2r$ around this center node. We devise a new algorithm with these properties.

1.3 Additive Spanners

The most popular definition of a spanner is that *all* pairwise distances must be preserved *up to an error function* which can be additive, multiplicative, or mixed.

Definition 2. *[Spanner] A subgraph H is an (α, β)-*spanner *of a graph G if*

$$\delta_H(u, v) \leq \alpha \cdot \delta_G(u, v) + \beta$$

for all $u, v \in V$.

Spanners are well-studied combinatorial objects. Some of their many applications include protocol synchronization in unsynchronized networks [PU89a], and the design of low-stretch routing algorithms which follow particularly compact routing tables [Cow01, CW04, PU89b, RTZ08, TZ01]. They have also been used to create low space distance oracles [TZ05, BS07, BK06, RTZ08] and almost-shortest path algorithms [EZ06, Elk05, Elk07, DHZ96]. Mild variations on graph spanners have appeared in broadcasting [FPZW04], solving diagonally dominant linear systems [ST04], and more.

Spanners were introduced by Peleg and Shäffer [PS89] in the *multiplicative* error setting ($\beta = 0$), and many results followed (e.g.[ADD+93, RZ04, RTZ05, BS07]). The more general mixed setting is studied for instance by [EP04]. In this paper we will focus on the case when the error is additive, i.e. when $\alpha = 1$.

Definition 3. *[Additive spanner] A subgraph H is a* $+\beta$ spanner *of a graph G if*

$$\delta_H(u, v) \leq \delta_G(u, v) + \beta$$

for all $u, v \in V$.

Additive spanners were first considered by Liestman and Shermer [LS91, LS93]. There are three known constructions of constant error additive spanners:

1. $+2$ spanners on $O(n^{3/2})$ edges [ACIM99]
2. $+4$ spanners on $\tilde{O}(n^{7/5})$ edges [Che13]
3. $+6$ spanners on $O(n^{4/3})$ edges [BKMP05]

Followup work has made various minor improvements to these base constructions, such as shaving log factors, improving the construction time, derandomizing, simplifying, etc. See [DHZ96, EP04, RTZ05, TZ06] for work on the $+2$ spanner or [Woo10, Knu14] for work on the $+6$ spanner.

Progress has mysteriously halted at this $n^{4/3}$ threshold: it is currently open whether or not there exist spanners on $O(n^{4/3-\delta})$ edges, even if the additive error function can be as large as $+n^{o(1)}$. Breaking this $n^{4/3}$ barrier is considered to be a major open question [DHZ96, BKMP05, BKMP10, Woo10, BW15, Knu14, Che13].

Meanwhile, current lower bounds for additive spanners allow plenty of room for improvement. Erdös' Girth Conjecture [Erd64] implies that $+(2k-1)$ spanners require $\Omega(n^{1+1/k})$ edges for any constant k; Woodruff [Woo06] has shown that this same lower bound holds independent of the Girth Conjecture. This implies that the $+2$ spanner is tight, but that the other spanners might be improvable; in particular, it is conceivable that there is a $+\beta_\epsilon$ spanner on $O(n^{1+\varepsilon})$ edges for all $\varepsilon > 0$.

Given the apparent robustness of the $n^{4/3}$ barrier to progress, researchers have sought spanners on $n^{4/3-\delta}$ edges with small polynomial amounts of error. This is where our work lies. The following sparsity bounds have been obtained for $+O(n^d)$ spanners, presented in chronological order: the first [BCE03] had $O(n^{3/2-d/2})$ edges, then $+O(n^{4/3-d/3})$ edges in [BKMP05], then $O(n^{9/14-8d/7})$ edges [Pet09], then $\tilde{O}(n^{1/3-2d/3})$ edges with the restriction $d \leq 4/17$ [Che13], then $\tilde{O}(n^{1-2d})$ edges [BW15], and finally $\tilde{O}(n^{2/5-3d/5})$ [BW15]. Jointly, these last three spanners form the current state of the art beneath the $n^{4/3}$ threshold. If the $O(n^{1/3-2d/3})$ spanner construction [Che13] worked for all d, it would beat all other known constructions. Obtaining this tradeoff for all d is considered an important open problem [Che13, BW15].

Our work *subsumes* this open problem, showing that the tradeoff $O(n^{1/3-2d/3})$ edges with $+O(n^d)$ error is not optimal for *any* d. Using our new graph clustering technique, we show:

Theorem 5. *Let a, b be constants such that all graphs and pair sets have a distance preserver on $O(n + n^a|P|^b)$ edges. Then for any constant $d > 0$, all graphs have $+O(n^d)$ spanners on $n^{1+o(1)+(a+2b-1)/(a+2b+1)-d(10b-a+1)/(3(a+2b+1))}$ edges.*

To understand how different known distance preservers translate into spanners, consider Table 1.

Copyright © by SIAM.
Unauthorized reproduction of this article is prohibited.

Using the preserver bound	The spanner has size
$O(n^{1/2}\|P\|+n)$ (Coppersmith & Elkin [CE06])	$\tilde{O}(n^{10/7-d})$
$O(n\|P\|^{1/3})$ if $\|P\|=O(n)$ (Theorem 3)	$\tilde{O}(n^{5/4-5d/12})$ if $d \geq 3/13$
$O(n^{2/3}\|P\|^{2/3})$ if $\|P\|=\Omega(n)$ (Theorem 3)	$\tilde{O}(n^{4/3-7d/9})$ if $d \leq 3/13$
$O(n^{2/3}\|P\|^{2/3}+n)$ (Conjecture 1)	$\tilde{O}(n^{4/3-7d/9})$

Table 1: Our new tradeoffs for standard spanners.

Our spanners are the sparsest known for all $d > 0$. In particular, our tradeoff is better than the hypothetical $n^{1/3-2d/3}$ tradeoff for all $d > 0$.

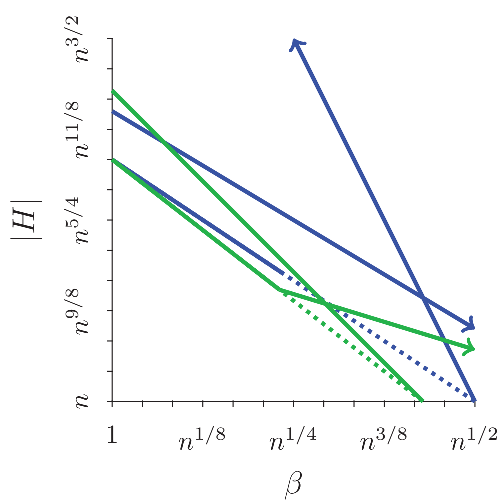


Figure 2: State of the art for $+\beta$ (polynomial error) additive spanners beneath the $n^{4/3}$ threshold. Old state-of-the-art upper bounds are in solid blue, and the (previously open) $n^{1/3-2d/3}$ bound discussed above is shown by the dotted blue line. Our new unconditional upper bounds are in solid green, and the upper bound obtained under our distance preserver conjecture is shown by the dotted green line.

1.4 Subset Spanners

A recent research trend has been to merge the previous two formalizations of the distance sparsification problem: only *some* pairwise distances must be preserved *up to an error function.*

Definition 4. *Let $G = (V, E)$ be an undirected unweighted graph, and let $P \subset V \times V$. We say that a subgraph $H = (V, E')$ is a $+\beta$* pairwise spanner *of G, P if*

$$\delta_H(u, v) = \delta_G(u, v)$$

for all $(u, v) \in P$.

A slight restriction of this concept is:

Definition 5. *A subgraph H is a $+\beta$* subset spanner *of a graph G and a node subset S if*

$$\delta_H(u, v) \leq \delta_G(u, v) + \beta$$

for all $u, v \in S$.

There are three known constructions for pairwise spanners in their most general form. These are: a $+2$ pairwise spanner on $\tilde{O}(n|P|^{1/3})$ edges due to Kavitha & Varma [KV13], a $+4$ pairwise spanner on $\tilde{O}(n|P|^{2/7})$ edges due to Kavitha [Kav15], and a $+6$ pairwise spanner on $O(n|P|^{1/4})$ edges also due to Kavitha [Kav15]. There is also a $+2$ subset spanner on $O(n|S|^{1/2})$ edges, originally due to Elkin (unpublished). Obtaining a constant error subset spanner on $O(n|S|^{1/2-\delta})$ edges (or, by extension, a constant error pairwise spanner on $O(n|P|^{1/4-\delta})$ edges) would be enough to break the $n^{4/3}$ threshold for standard spanners discussed above. As such, this task seems very difficult.

Like standard spanners, then, it seems important to achieve a good polynomial sparsity/error tradeoff below this bound. However, no progress on this task has yet been reported. The best construction we know is to naively ignore the given pair set and construct a sparse (standard) spanner with polynomial error. It is an important open question [CGK13, KV13, BW15] to construct a subset/pairwise spanner that benefits in a natural way from a polynomial error allowance.

That is exactly what we accomplish, for subset spanners. We prove:

Theorem 4. *For any constant $d > 0$, all graphs G and node subsets S have a $+O(n^d)$ subset spanner on $|E(H)| = \tilde{O}(n) + |S|^{(2b+a-1)/2} n^{1-d(1-a)+o(1)}$ edges.*

Downloaded 02/10/16 to 171.64.66.62. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

Copyright © by SIAM.
Unauthorized reproduction of this article is prohibited.

Using the preserver bound	The sub. spanner has size
$O(n^{1/2}\|P\|+n)$ (Coppersmith & Elkin [CE06])	$\|S\|^{3/4}n^{1-d/2+o(1)}$
$O(n\|P\|^{1/3})$ if $\|P\|=O(n)$ (Theorem 3)	$\|S\|^{1/3}n^{1+o(1)}$ if $\|S\|=O(n^{2d})$
$O(n^{2/3}\|P\|^{2/3})$ if $\|P\|=\Omega(n)$ (Theorem 3)	$\|S\|^{1/2}n^{1-d/3+o(1)}$ if $\|S\|=\Omega(n^{2d})$
$O(n^{2/3}\|P\|^{2/3}+n)$ (Conjecture 1)	$\|S\|^{1/2}n^{1-d/3+o(1)}$

Table 2: Our new tradeoffs for subset spanners.

2 Conventions

All graphs in this paper are undirected and unweighted. The variable n is reserved for the number of nodes in the graph G currently being discussed. If $G=(V,E)$ be a graph, then we say P is a *pair set* on G if $P \subset V \times V$. We use the notation $\delta_G(u,v)$ to refer to the shortest path distance between u and v in the graph G. For a node u in G, we denote by $B_{\le}(u,r)$ the set of nodes at distance r or less from u. Similarly, $B_{<}(u,r)$ is the set of nodes at distance strictly less than r from u, and $B_{=}(u,r)$ is the set of nodes at distance exactly r from u.

3 Pairwise Distance Preservers

Recall the following definition from the introduction:

Definition 1. *Given a graph G and a pair set $P \subset V \times V$, we say that a subgraph $H \subset G$ is a* pairwise distance preserver *of G,P if $\delta_H(u,v)=\delta_G(u,v)$ for all $(u,v)\in P$.*

Prior work has considered distance preservers on possibly directed or weighted G, but we will restrict our attention to the undirected and unweighted case.

One can imagine a pair set in which each pair $(u,v)\in P$ has a unique shortest path in G. In this case, there is no room for algorithmic cleverness in the construction of the preserver H; it is necessary that H is exactly the union of these shortest paths. The entire algorithmic component of the problem lies in path tiebreaking: if there is a pair (u,v) such that G contains several equally short paths between u and v, then we need to choose which one of these to include in our preserver. We formalize this as follows:

Definition 6. *A* path tiebreaking scheme *on a graph G is a function ρ_G that maps node pairs (u,v) to a shortest path in G from u to v.*

Given a graph G and a pair set P, one can construct a distance preserver by simply choosing a tiebreaking scheme ρ_G, and then setting $H=\bigcup_{p\in P}\rho_G(p)$. No generality is lost in this approach.

A major theme of this section is the difference in power between various tiebreaking schemes.

3.1 Old Tiebreaking Schemes

Coppersmith & Elkin's upper bound of $O(n\sqrt{|P|})$ is realized regardless of the tiebreaking scheme used. Their other upper bound of $O(n+\sqrt{n}|P|)$ is realized only by tiebreaking schemes with the following property:

Definition 7. *A tiebreaking scheme ρ_G is* consistent *if, whenever $w,x\in\rho_G(u,v)$, we have $\rho_G(w,x)\subset\rho_G(u,v)$.*

They also use a slight variant on the following definition:

Definition 8. *Let H be a directed graph. A* branching event *b is a pair of distinct edges that enter the same node.*

The metaphor at work here is that we imagine starting with an edgeless graph, and then build a distance preserver by adding directed paths to it one by one. Each branching event captures one instance of two paths intersecting (although "free" intersections, in which the intersecting edge was already in the preserver due to another path, are not counted).

The following lemma (also due to Coppersmith & Elkin) explains why this is a useful quantity to consider:

Lemma 1. *[CE06] A graph H with b branching events contains $O(n+(nb)^{1/2})$ edges.*

Proof. By a convexity argument, we have

$$b=\sum_{v\in V}\binom{\deg_{in} v}{2}\ge\sum_{v\in V}\binom{\lceil |H|/n\rceil}{2}$$

Assuming $\lceil |H|/n\rceil\ge 2$ (and so $|H|>n$), we have

$$\sum_{v\in V}\binom{\lceil |H|/n\rceil}{2}=\Theta(n(|H|/n)^2)=\Theta(|H|^2/n)$$

Copyright © by SIAM.
Unauthorized reproduction of this article is prohibited.

Downloaded 02/10/16 to 171.64.66.62. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

Therefore, if $|H| > n$, we have $\sqrt{bn} = \Omega(|H|)$. So $|H| = O(n + \sqrt{bn})$. $\square$

The proof of the $O(n + n^{1/2}|P|)$ upper bound is now straightforward. Let $H = \bigcup_{p\in P} \rho_G(p)$ be your distance preserver of G, P. If ρ_G is a consistent tiebreaking scheme, it is not too hard to see that any pair of paths $\rho_G(p_1)$ and $\rho_G(p_2)$ can contribute at most two branching events to H, and therefore H has only $O(|P|^2)$ branching events. The $O(n + n^{1/2}|P|)$ upper bound then follows from Lemma 1.

We now know that any consistent tiebreaking scheme implements the Coppersmith & Elkin upper bounds of $O(\min\{n + n^{1/2}|P|, n|P|^{1/2}\})$. Looking forward, how can these upper bounds be improved? There are two possible directions of research. Perhaps (1) there are stronger upper bounds that apply to *arbitrary consistent tiebreaking schemes*, and we just need to refine our proofs. Or maybe (2) we have exhausted the potential of the consistency definition, and in order to move forward, we will need to invent some new tiebreaking schemes that have other properties besides consistency. Our first result is that the answer is (2): the Coppersmith & Elkin bounds are tight for consistent tiebreaking schemes.

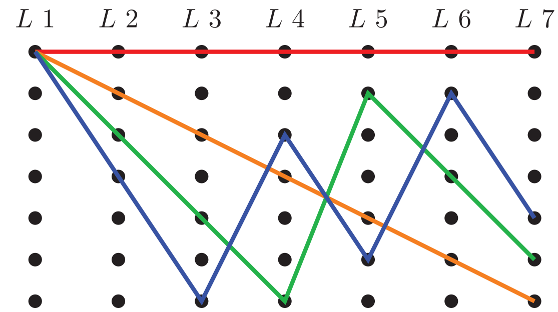


Figure 3: The graph described in Theorem 1, with $n^{1/2} = 7$ (not pictured: all possible edges between any two adjacent layers). We use $P = L1 \times L7$ (or any subset of this, if $c < 1$). The first four paths $\rho_G(p)$ that start at the first node in $L1$ have been drawn on the graph. Note that each pair intersects on only one node.

Theorem 1. *For infinitely many n and any parameter $\frac{1}{2} \le c \le 1$, there is an unweighted, undirected graph G on n nodes, a pair set P of size n^c, and a consistent tiebreaking scheme ρ_G such that*

$$\left|\bigcup_{p\in P} \rho_G(p)\right| = n^{1/2}|P|$$

Proof. Let $q = n^{1/2}$ be a prime. Let G be the complete graph on q layers; that is, it consists of q layers of q nodes, with edges placed such that a node in layer L is adjacent to exactly the set of nodes in layer $L-1$ (if $L \ne 1$) and $L+1$ (if $L \ne q$). Let P be any set of pairs (u, v) such that u is in layer 1 and v is in layer q. Number the nodes in each layer from 0 to $q-1$. Define ρ_G by the following rule: if u is the i^{th} node in the first layer, and v is the j^{th} node in the last layer, then $\rho_G(u, v)$ is the path that repeatedly travels from the k^{th} node in the L^{th} layer to the $(k + (i-j) \mod q)$ node in the $(L+1)^{th}$ layer.

We claim that no two paths $\rho_G(p_1), \rho_G(p_2)$ intersect on more than a single node. To see this: suppose that $\rho_G(w, x), \rho_G(u, v)$ share the a^{th} node in layer L and also the b^{th} node in layer $L' > L$. Then

$$a + (w-x)(L'-L) \equiv b \equiv a + (u-v)(L'-L) \mod q$$

(where integers a, b, u, v, w, x stands in for the numbering of the nodes a, b, u, v, w, x in their respective layer). Since q is prime we can reduce this equation to $w - x \equiv u - v$. We then have:

$$w + (w-x)L \equiv a \equiv u + (w-x)L \mod q$$

and so $w = u$. This implies that $(w, x) = (u, v)$, and so in fact these paths are identical.

Since each pair of paths intersects on only 1 or 0 nodes, it is clear that ρ_G is consistent. Additionally, this condition implies that no two paths share an edge. Since $\delta_G(p) = n^{1/2}$ for all $p \in P$, each path adds exactly $n^{1/2}$ edges to the preserver, and the claim follows. $\square$

Theorem 2. *or infinitely many n and any parameter $1 \le c \le 2$, there is an unweighted, undirected graph G on n nodes, a pair set P of size n^c, and a consistent tiebreaking scheme ρ_G such that*

$$\left|\bigcup_{p\in P} \rho_G(p)\right| = n|P|^{1/2}$$

Proof. Let $q = n^{c/2}$ be a prime. Construct the complete graph on n/q layers of q nodes each, and choose your pair set to be any appropriately-sized set of nodes such that each pair has one node in the first layer and the other node in the last layer. The proof is now identical to that of Theorem 1. $\square$

3.2 New Tiebreaking Schemes

We will next prove a new upper bound of $O(n^{2/3}|P|^{2/3} + n|P|^{1/3})$. By the theorems above, this improvement

Copyright © by SIAM.
Unauthorized reproduction of this article is prohibited.

Downloaded 02/10/16 to 171.64.66.62. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

Downloaded 02/10/16 to 171.64.66.62. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

will require a new tiebreaking scheme. This scheme is contained in the following lemma:

Lemma 2. *Let G be an unweighted undirected graph, and let S be a subset of nodes such that every pair of nodes in S is distance d or less apart. Let P be a pair set such that every pair in P has a shortest path incident on S. Then there is a tiebreaking scheme ρ_G such that*

$$\left| \bigcup_{p \in P} \rho_G(p) \right| = O(n + (n|P||S|d)^{1/2})$$

Proof. By Lemma 1, it suffices to prove that H has $O(|P||S|d)$ branching events. We will do exactly that. Let $H = (V, \emptyset)$ be a distance preserver that we will build iteratively. Assign each pair $p \in P$ to a node $u \in S$ such that p has a shortest path through u. Expand the pair set as follows: if (a, b) is in the pair set and is owned by node u, replace it with two pairs (u, a) and (u, b). We will add a shortest path to our preserver for each pair in this expanded pair set, and for purposes of counting branching events, we will direct each edge from the node closer to u to the node closer to a/b.

Fix an ordering of the nodes in S, and add all paths that belong to an earlier node before adding any paths that belong to a later node. For each node $u \in S$ in order, start adding its paths to H according to any consistent tiebreaking scheme. We will maintain the following invariant: for each previously added path p belonging to a node v that precedes u in the ordering, at most $2d + 1$ paths belonging to u branch with p. If we ever add a path belonging to u that violates this invariant, we will pause the algorithm and reroute one or more of these $2d + 2$ paths to restore the invariant.

Suppose that there are $2d + 2$ paths belonging to s that have each added a distinct edge entering some previously added path p, owned by node v. Let $v_1, \ldots, v_{2d+2}$ be distinct nodes in p on which a path owned by u adds an edge, ordered by distance from v (so $\delta_G(v, v_1) < \delta_G(v, v_2)$ and so on). By the triangle inequality, we have for all $1 \le j \le 2d + 2$:

$$\delta_G(u, v) \ge \delta_G(u, v_j) - \delta_G(v, v_j) \ge -\delta_G(u, v)$$

We also know $\delta_G(u, v) \le d$, so we can write

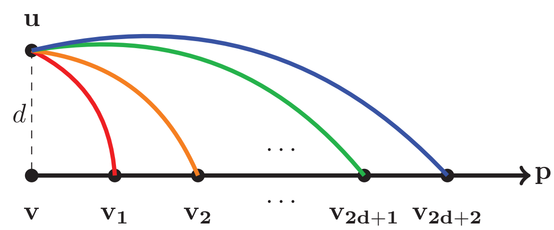
$$d \ge \delta_G(u, v_j) - \delta_G(v, v_j) \ge -d$$

By the pigeonhole principle, there exist values $1 \le j < k \le 2d + 2$ with

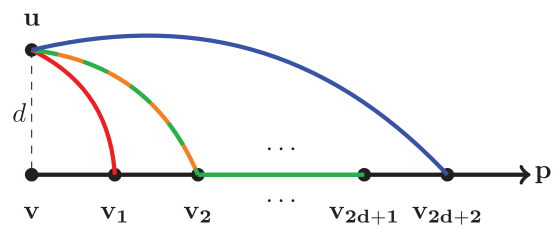
$$\delta_G(u, v_j) - \delta_G(v, v_j) = \delta_G(u, v_k) - \delta_G(v, v_k)$$

And so

$$\delta_G(u, v_j) + \delta_G(v, v_k) - \delta_G(v, v_j) = \delta_G(u, v_k)$$



(a) Suppose that $u, v \in S$, with v preceding u in the ordering, and let p be a path owned by v. If paths owned by u enter p at $2d + 2$ or more different points ...



(b) ... then we can reroute one of these paths, without stretching its length, so that it coincides with another path up until it reaches p (in this picture, we have rerouted green into orange).

Figure 4: A graphical depiction of the "rerouting" technique from Lemma 2.

$$\delta_G(u, v_j) + \delta_G(v_j, v_k) = \delta_G(u, v_k)$$

We may therefore replace the prefix $\rho_G(u, v_k)$ of all paths that first intersect p at the node v_k with the new prefix $\rho_G(u, v_j) \cup \rho_G(v_j, v_k)$, and this replacement will not stretch any of these paths. In doing so, we now have that no paths owned by u intersect p at the node v_k, and so the invariant is restored.

Note that when we perform this rerouting, we cannot introduce any new edges to the preserver; therefore, when we repair the invariant on the path p, we will not destroy the invariant on any other path. □

With this lemma in hand, we can now prove our new upper bound.

Theorem 3. *All undirected unweighted graphs G and pair sets P have a distance preserver H with $|E(H)| = O(n^{2/3}|P|^{2/3} + n|P|^{1/3})$.*

Proof. Let ϵ be a parameter. Start adding paths from P to the (initially empty) preserver in any order, according to any tiebreaking scheme. Suppose that at some point during this process, a node u gains the following property: there exists a set of at most n^ϵ nodes within distance 1 of u such that $n^{2\epsilon}$ distinct paths pass through

Copyright © by SIAM.
Unauthorized reproduction of this article is prohibited.

Downloaded 02/10/16 to 171.64.66.62. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

these nodes. We then remove exactly $n^{2\epsilon}$ of these paths from the preserver and create an auxiliary preserver that handles only these paths. We can now apply Lemma 2 to these paths with $d = 2, |S| \leq n^{\varepsilon}, |P| = n^{2\varepsilon}$. Therefore, the auxiliary preserver has $O(n + n^{1/2+3\varepsilon/2})$ edges.

At the end of this process, we have some number of auxiliary preservers, plus a "leftover" preserver full of paths that were never removed by the above process. We will next argue that the leftover preserver has only $O(n^{1+\varepsilon})$ edges. The leftover preserver has the property that, for all nodes v, there is no set of n^{ϵ} nodes within distance 1 of v such that at least $n^{2\epsilon}$ distinct paths pass through one of these nodes. Unmark all nodes and all edges. Repeat the following process until you can do so no longer:

1. Choose an unmarked node v.
2. If v has fewer than n^{ε} unmarked neighbors, then mark v and all its incident edges.
3. If v has more than n^{ε} unmarked neighbors, then choose n^{ε} of its neighbors, and mark all of these nodes and their incident edges.

Once we have marked all nodes, it is clear that we have also marked all edges. Each time we mark a single node, we mark at most n^{ϵ} edges along with it. Each time we mark a set of n^{ϵ} nodes, we mark at most $4n^{2\epsilon}$ edges along with it (the edges belonging to $n^{2\epsilon}$ paths incident on this set). Therefore the graph has $O(n^{\epsilon})$ times as many edges as it has nodes. So the leftover preserver has size $O(n^{1+\epsilon})$ edges.

We will next bound the size of the auxiliary preservers. First suppose that $\varepsilon \leq \frac{1}{3}$, and so the size of each auxiliary preserver is $O(n)$. We then set $n^{\varepsilon} = |P|^{1/3}$. The size of the leftover preserver is then $O(n|P|^{1/3})$. Additionally, each auxiliary preserver handles $|P|^{2/3}$ paths, and so at most $|P|^{1/3}$ of them exist, so (by a union bound) the total size of the auxiliary preservers is $O(n|P|^{1/3})$. The total size of the leftover plus auxiliary preservers is then $O(n|P|^{1/3})$.

Finally, suppose that $\varepsilon \geq \frac{1}{3}$, and so the size of each auxiliary preserver is $O(n^{1/2+3\varepsilon/2})$. We then set $n^{\varepsilon} = |P|^{2/3}/n^{1/3}$. The size of the leftover preserver is then $O(n^{2/3}|P|^{2/3})$. Additionally, each auxiliary preserver handles $|P|^{4/3}/n^{2/3}$ paths, and so we can have at most $n^{2/3}/|P|^{1/3}$ auxiliary preservers. Each one costs $O(|P|)$ edges, and so (by a union bound) the total size of the auxiliary preservers is $O(n^{2/3}|P|^{2/3})$. The total size of the leftover plus auxiliary preservers is then $O(n^{2/3}|P|^{2/3})$.

Regardless of the value of ε, then, the total size of the distance preserver can be expressed as $O(n^{2/3}|P|^{2/3} + n|P|^{1/3})$. □

The best known upper bounds are now $O(n^{1/2}|P|)$ when $\Omega(n^{1/2}) = |P| = O(n^{3/4})$, then $O(n|P|^{1/3})$ when $\Omega(n^{3/4}) = |P| = O(n)$, then $O(n^{2/3}|P|^{2/3})$ when $\Omega(n) = |P| = O(n^2)$. We consider it fairly unlikely that this piecewise behavior reflects the "true" distance preserver upper bound.

Conjecture 1. *Every unweighted, undirected graph G and pair set P admits a pairwise distance preserver on $O(n + n^{2/3}|P|^{2/3})$ edges.*

See Figure 1 in the introduction for a visualization of these bounds.

Throughout the rest of this paper, we will reserve a and b for the following purpose:

Definition 9. *We define a, b to be constants such that one can always construct distance preservers on $O(n + n^a|P|^b)$ edges.*

This allows us to prove general results in terms of a and b, and then substitute in any preserver upper bound at the end.

4 Graph Clustering with Preservers

4.1 Graph Clustering

We begin with the following clustering algorithm:

Lemma 3. *Let $G = (V, E)$ be an undirected unweighted graph, and let r be a parameter. In polynomial time, one can find a set of nodes $v_1, \ldots, v_k$ (called "cluster centers") and a set of integers $r_1, \ldots, r_k$, with $r \leq r_i \leq r \cdot n^{o(1)}$, such that the following properties hold:*

1. *For each node $v \in V$, there is an i such that $v \in B_{\leq}(v_i, r_i)$*
2. $\sum_{i=1}^{k} |B_{\leq}(v_i, 2r_i)| = \tilde{O}(n)$

The set $B_{\leq}(v_i, 2r_i)$ is called the "cluster" centered at v_i (also denoted X_i), and the set $B_{\leq}(v_i, r_i)$ is called the "core" of the cluster X_i (also denoted C_i).

This lemma is very similar to many previously known region-growing algorithms (see [AP92, Coh93, PR10] for example). The additional structure we need, which forces us to devise a new algorithm rather than recycling an old one, is that the core of each cluster is padded by non-core nodes for at least r_i distance in every direction.

Proof. First, for every node $v \in V$, we will compute a value r_v. Initialize $r_v \leftarrow r$. Check to see if

Copyright © by SIAM.
Unauthorized reproduction of this article is prohibited.

Downloaded 02/10/16 to 171.64.66.62. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

$|B_{\leq}(v,r_v)|\log n \geq |B_{\leq}(v,4r_v)|$. If so, fix r_v at its current value and move on to the next node $v \in V$. If not, set $r_v \leftarrow 4r_v$ and repeat. In each iteration of the process, we multiply r_v by 4 while we multiply $|B_{\leq}(v,r_v)|$ by at least $\log n$. Since $|B_{\leq}(v,r_v)| \leq n$ at all times, we iterate at most $\frac{\log n}{\log\log n}$ times, and so the final value of r_v is at most $r \cdot 4^{(\log n)/(\log\log n)} = r \cdot n^{o(1)}$.

Sort all nodes $v \in V$ descendingly by the value of r_v. Now, repeat the following process until you can do so no longer:

1. Remove the first remaining node v from the list, and add it to the set of cluster centers, calling it v_i and incrementing i (i starts at 1). Set its corresponding r_i value to be $2r_v$.

2. For each node u with $B_{\leq}(u,r_u) \cap B_{\leq}(v,r_v) \neq \emptyset$, delete u from the list.

We claim that we have generated a set of cluster centers with all desired properties. We have already shown that $r \leq r_i \leq r \cdot n^{o(1)}$ for all i. Next, we will show that for all $v \in V$, there is an i such that $v \in B_{\leq}(v_i, r_i)$. If v is a cluster center, then the claim is trivial. Otherwise, there must be some cluster center v_i that preceded v in the list with the property that $B_{\leq}(v_i, r_{v_i}) \cap B_{\leq}(v,r_v) \neq \emptyset$. By the triangle inequality, this implies that $\delta_G(v_i, v) \leq r_{v_i} + r_v \leq 2r_{v_i} = r_i$, which implies the claim.

Finally, we must show that $\sum_{i=1}^{k} |B_{\leq}(v_i, 2r_i)| = \tilde{O}(n)$. Note that the sets $B_{\leq}(v_i, r_i/2)$ (where v_i is a cluster center) are disjoint. We then have

$$\sum_{i=1}^{k} |B_{\leq}(v_i, 2r_i)| \leq \log n \cdot \sum_{i=1}^{k} |B_{\leq}(v_i, r_i/2)| \leq n \log n$$

implying the claim. ☐

We will add some machinery to this clustering algorithm to make it useful for spanner creation. We will make the following distinction in cluster size:

Definition 10. *A cluster X is* large *with respect to a parameter $\mathcal{E}$ if $|X| \geq r^{2b/(2b+a-1)}\mathcal{E}^{1/(2b+a-1)}$, or* small *otherwise.*

Our choice of exponents is designed to push through the following lemma:

Lemma 4. *For each small cluster X_i with center v_i, there is an integer $r_i < \bar{r_i} \leq 2r_i$ with*

$$|B_{\leq}(v_i, \bar{r_i})|^a (|(B_{=}(v_i, \bar{r_i})|^2)^b = O(|B_{<}(v_i, \bar{r_i})|\mathcal{E})$$

Proof. Suppose otherwise, towards a contradiction. Then we have

$$|B_{=}(v_i, \bar{r_i})| \geq c|B_{<}(v_i, \bar{r_i})|^{(1-a)/(2b)}\mathcal{E}^{1/(2b)}$$

for all $r_i < \bar{r_i} \leq 2r_i$ and constants c. We can interpret this expression as a recurrence relation on the size of $B_{<}(v_I, \bar{r_i})$ as $\bar{r_i}$ grows from r_i+1 to $2r_i$ (denoted $S_{\bar{r_i}}$).

$$S_{r_i+1} \geq 1 \qquad \text{and} \qquad S_{k+1} \geq S_k + cS_k^{(1-a)/(2b)}\mathcal{E}^{1/(2b)}$$

And so

$$\Delta_k \geq cS_k^{(1-a)/(2b)}\mathcal{E}^{1/(2b)}$$

where $\Delta_k = S_{k+1} - S_k$. This is a discrete approximation of the differential equation

$$\frac{d\,S_k}{d\,k} \geq c\mathcal{E}^{1/(2b)}S_k^{(1-a)/(2b)}$$

which has the standard form $y'(x) = \alpha y(x)^{\beta}$ (in this case, $\alpha = c\mathcal{E}^{1/(2b)}$, and $\beta = (1-a)/(2b)$), and so our discrete version enjoys the same asymptotics. The general solution to this differential equation is $y = c_1(\alpha x)^{1/(1-\beta)}$. Accordingly, for our discrete version, we gain:

$$S_{r_i+k} \geq c'(\mathcal{E}^{1/(2b)}k)^{1/(1-(1-a)/(2b))}$$

where c' is some new constant dependent on the old value of c. Algebraic manipulation now yields

$$S_{r_i+k} \geq c'(\mathcal{E}^{1/(2b)}k)^{2b/(2b+a-1)}$$
$$S_{r_i+k} \geq c'\mathcal{E}^{1/(2b+a-1)}k^{2b/(2b+a-1)}$$
$$S_{2r_i} \geq c'\mathcal{E}^{1/(2b+a-1)}r_i^{2b/(2b+a-1)}$$
$$S_{2r_i} \geq c'\mathcal{E}^{1/(2b+a-1)}r^{2b/(2b+a-1)}$$

If we choose c such that c' is sufficiently large, this contradicts the assumption that X_i is small. ☐

This lemma is the heart of our reduction from spanners to distance preservers, and it is the entire reason we have gone through the trouble to build our own clustering algorithm. The idea is that, for each cluster, one of the following two cases must happen: (1) each subsequent layer of nodes around the core represents a significant growth in the cluster size, or (2) one of these layers L is unusually small, and therefore it is "cheap" to make a distance preserver on the pair set $L \times L$.

Lemma 5. *Let X be a large cluster. Let Q be a set of node pairs contained in X. If $|Q| = O(r^{2(1-a)/(2b+a-1)}\mathcal{E}^{2/(2b+a-1)})$, then there is a tiebreaking scheme ρ_X such that*

$$\left|\bigcup_{q\in Q} \rho_X(q)\right| = O(|X|\mathcal{E})$$

Copyright © by SIAM.
Unauthorized reproduction of this article is prohibited.

Using the distance preserver bound	A large cluster has size
$O(n + n^{1/2}\|P\|)$ (Coppersmith & Elkin [CE06])	$\Omega(r^{4/3}\mathcal{E}^{2/3})$
$O(n\|P\|^{1/3})$ if $\|P\| = O(n)$ (Theorem 3)	$\Omega(r\mathcal{E}^{3/2})$ if $r = \Omega(\mathcal{E}^{3/2})$
$O(n^{2/3}\|P\|^{2/3})$ if $\|P\| = \Omega(n)$ (Theorem 3)	$\Omega(r^{4/3}\mathcal{E})$ if $r = O(\mathcal{E}^{3/2})$
$O(n + n^{2/3}\|P\|^{2/3})$ (Conjecture 1)	$\Omega(r^{4/3}\mathcal{E})$

Table 3: The threshold for a cluster being defined as "large," relative to the distance preserver upper bound being used.

Proof. Observe that

$$|Q| = (r^{2b/(2b+a-1)}\mathcal{E}^{1/(2b+a-1)})^{(1-a)/b}\mathcal{E}^{1/b}$$

Since X is large, we have $|X| \geq r^{2b/(2b+a-1)}\mathcal{E}^{1/(2b+a-1)}$. Therefore

$$|Q| = O(|X|^{(1-a)/b}\mathcal{E}^{1/b})$$

By definition of a and b, we can create a distance preserver for this pair set in the subgraph X paths on $O(|X|^a|Q|^b)$ edges. We then have

$$O(|X|^a|Q|^b) = O(|X|\mathcal{E})$$

as claimed. □

4.2 Path Decomposition

Before we proceed to our spanner algorithms, we will discuss a useful method for dividing paths into easy-to-analyze subpaths.

Lemma 6. *Let G be a graph and p be a shortest path in G. Let $\{x_i, v_i\}$ be a clustering of G as in Lemma 3. One can partition p into subpaths $\{p_1, \ldots, p_k\}$ such that every subpath p_i can be classified into one of two cases:*

1. *A* small subpath*, for which every edge in p_i is incident on some small cluster core C_i.*

2. *A* large subpath*, in which every node is in a large cluster X_i.*

Additionally, one can assign large clusters X to large subpaths p_i with $p_i \subset X$ such that no two subpaths correspond to the same large cluster.

Proof. Choose an i such that the first node of p is in C_i. If X_i is small, then let w be the first node in p that is not also in C_i. Otherwise, if X_i is large, then let w be the last node in X_i such that $\rho_G(x, w) \subset X_i$. In either case, add $\rho_G(u, w)$ to your list of subpaths, and then repeat the analysis on $\rho_G(w, v)$ (if this subpath is nonempty). Note that $w \neq u$ (because in either case $w \in C_i$ but $u \notin C_i$, and so this process will eventually terminate.

The only nontrivial detail to prove is that this process will never select the same large cluster X_i twice. Suppose towards a contradiction that a large cluster X_i is selected twice; then p must include a node $c \in C_i$, then a node $v \notin X_i$, then another node $c' \in C_i$ in that order. We know $\delta_G(c, v) > r_i$ and $\delta_G(c', v) > r_i$, because $c, c' \in B(v_i, r_i)$ but $v \notin B(v_i, 2r_i)$. This implies that $\delta_G(c, c') \geq 2r_i + 2$. However, we also have $\delta_G(c, v_i) \leq r_i$ and $\delta_G(c', v_i) \leq r_i$, which implies that $\delta_G(c, c') \leq 2r_i$. These statements are contradictory, so instead it must be the case that no large cluster is ever selected twice. □

We use this decomposition to classify the edges of each path as follows.

Definition 11. *Let $\rho_G(u, v)$ be a path that has been decomposed into subpaths $\{p_1, \ldots, p_k\}$ as in Lemma 6. Then we classify the subpaths as follows:*

1. *An* extreme subpath *is a subpath that belongs to a cluster X such that $u \in X$ or $v \in X$.*

2. *A* small subpath *is a non-extreme subpath that belongs to a small cluster X.*

3. *A* large subpath *is a non-extreme subpath that belongs to a large cluster X.*

5 Applications to Additive Spanners

5.1 Subset Spanners

Recall the following definitions from the introduction:

Definition 5. *A subgraph H is a $+\beta$* subset spanner *of a graph G and a node subset S if*

$$\delta_H(u, v) \leq \delta_G(u, v) + \beta$$

for all $u, v \in S$.

Copyright © by SIAM.
Unauthorized reproduction of this article is prohibited.

Downloaded 02/10/16 to 171.64.66.62. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

Downloaded 02/10/16 to 171.64.66.62. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

Algorithm 1: subspan$(G, S, d > 0)$

Initialize H to be a $\cdot \log n$ multiplicative spanner of G;
for *each pair* $s_1, s_2 \in S$ *(in some fixed order)* **do**
  if $\delta_H(s_1, s_2) > \delta_G(s_1, s_2) + n^d$ **then**
    Add all edges in $\rho_G(s_1, s_2)$ to H;
  end
end
return H;

We will use Algorithm 1 to generate our subset spanners. It is trivially true that the output of this algorithm is a $+n^d$ subset spanner of G, S; we omit this proof. We will now prove an upper bound on the number of edges in the graph H returned by this algorithm.

Overview of the Edge Bound. Take the set $S \times \{X_i\}$, where X_i are clusters in some clustering of G. Think of each element of this set as "unmarked." Whenever we add a shortest path to H with endpoint $s \in S$ that intersects a certain cluster X, we then "mark" the pair (s, X). Whenever we add a path $\rho_G(s_1, s_2)$ to H, each cluster that intersects $\rho_G(s_1, s_2)$ will be marked along with either s_1 or s_2, because otherwise we have already accurately spanned the pair (s_1, s_2).

We then argue that (1) not very many of the edges in H are added by extreme subpaths, (2) the total cost of the small subpaths can be bounded by our distance preserver reduction (see Lemma 4 or Figure 5), and (3) we only add $|S|$ large subpaths per large cluster, and so the total cost of the large subpaths can be bounded by Lemma 5.

We will now proceed with the proof.

Lemma 7. *Let $\{v_i, r_i\}$ be a clustering of G as in Lemma 3, with parameter r chosen such that $\max_i r_i \leq n^d/(8 \log n)$ (so $r = n^{d-o(1)}$). For each cluster X_i, Algorithm 1 will add at most $|S|$ paths to H that are incident on X_i.*

Proof. Consider each pair $s_1, s_2 \in S$ in turn. Let p be any shortest path between s_1 and s_2 in G, and let $\{p_1, \dots, p_k\}$ be a decomposition of p as in Lemma 6. First, suppose that for some cluster X_i, we have already added shortest paths to H with endpoints s_1 and s_2 that intersect X_i. In this case, we claim that we already have $\delta_H(s_1, s_2) \leq \delta_G(s_1, s_2) + n^d$, and therefore, we will skip adding $\rho_G(s_1, s_2)$ to H in the algorithm. To see this, let $x_1, x_2 \in X_i$ such that there is a shortest path between the pairs s_1, x_1 and s_2, x_2 already in H. By the triangle inequality, we have:

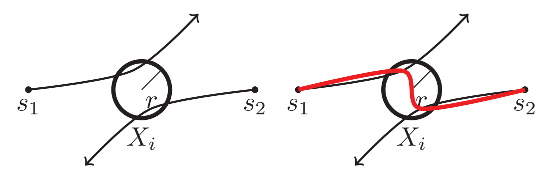
$$\delta_H(s_1, s_2) \leq \delta_H(s_1, x_1) + \delta_H(x_1, x_2) + \delta_H(x_2, s_2)$$

$$\delta_H(s_1, s_2) \leq \delta_G(s_1, x_1) + (n^d/2) + \delta_G(x_2, s_2)$$

Let x_3 be any node in X_i intersected by p. Then

$$\begin{aligned}\delta_H(s_1, s_2) \leq &(\delta_G(s_1, x_3) + n^d/(8 \log n)) + n^d/2 \\ &+ (\delta_G(x_3, s_2) + n^d/(48 \log n)) \\ \delta_H(s_1, s_2) \leq &\delta_G(s_1, s_2) + n^d\end{aligned}$$

Therefore, each time we add a path $\rho_G(s_1, s_2)$ to H, for each cluster X_i intersected by $\rho_G(s_1, s_2)$, we know that $\rho_G(s_1, s_2)$ is either (1) the first path with endpoint s_1 that intersects X_i added to H. The lemma follows. □



(a) Let X_i be a cluster intersected by $\rho_G(s_1, s_2)$. If there are already two shortest paths through X_i with endpoints at s_1 and s_2...

(b) ...then there is already a path between s_1 and s_2 in H with only $+n^d$ stretch, so Algorithm 1 will not choose to add $\rho_G(s_1, s_2)$ to H.

Figure 7: A graphical depiction of the proof of Lemma 7.

Theorem 4. *For any constant $d > 0$, all graphs G and node subsets S have a $+O(n^d)$ subset spanner on $|E(H)| = \tilde{O}(n) + |S|^{(2b+a-1)/2} n^{1-d(1-a)+o(1)}$ edges.*

Proof. It is well known that all n-node graphs have an $\cdot \log n$ multiplicative spanner with $\tilde{O}(n)$ edges.

The remaining edges in H are all the result of adding paths $\rho_G(u, v)$. One again let $\{v_i, r_i\}$ be a clustering of G with parameter r chosen such that $\max_i r_i \leq n^d/(8 \log n)$ (so $r = n^{d-o(1)}$). Each of our paths can be decomposed over this clustering. We will say that an edge $e \in H$ is *extreme, small, or large* depending on whether the decomposed subpath p_i that

Copyright © by SIAM.
Unauthorized reproduction of this article is prohibited.

Downloaded 02/10/16 to 171.64.66.62. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

first added e to H is classified as extreme, small, or large as in Definition 11.

We will now count the three types of edges separately.

Extreme Edges. Since there is a $\cdot \log n$ multiplicative spanner already in H, and every path p added to H is *not* spanned up to $+n^d$ accuracy at the time it is added, we know that p is missing at least $n^d/\log n$ edges in total. Each cluster has radius at most $n^d/(8 \log n)$, so jointly, the two clusters in which p begins and ends contribute at most $n^d/(2 \log n)$ of these missing edges. So at most half of the total edges in H fall into this category. It therefore suffices to prove the edge bound for the other two types of edges.

Small Edges. For each small edge e, we know that e was a part of a subpath p_i owned by a small cluster X_i, and that p_i was a part of a larger path $\rho_G(u,v)$ that did not start or end in X_i. Choose $\bar{r}_i$ as in Lemma 4; then there are nodes $x \neq x' \in B_=(v_i, \bar{r}_i) \cap p$ such that $x, x' \in \rho_G(u,v)$ and e is between x and x' in $\rho_G(u,v)$. Therefore, $e \subset \rho_{X_i}(x,x')$. We can then cover all small edges belonging to X_i using a single distance preserver on $B_=(v_i, \bar{r}_i)$ within the subgraph $B_\leq(v_i, \bar{r}_i)$. By Lemma 4, with the proper tiebreaking scheme, this requires $O(|B_\leq(v_i, \bar{r}_i)|\mathcal{E})$ edges. So the total number of small edges in the entire graph is

$$\begin{aligned}\sum_{i \mid X_i \text{ is small}} O(|B_\leq(v_i, \bar{r}_i)|\mathcal{E}) =& \mathcal{E} \sum_{X_i \text{ is small}} O(|X_i|) \\ =& \tilde{O}(n\mathcal{E})\end{aligned}$$

where again the last equality follows from Lemma 3.

Large Edges. For each path $\rho_G(s_1, s_2)$ added to H by Algorithm 1, when we decompose these paths as in Lemma 6, we know from Lemma 7 that a total of $|S|$ or fewer subpaths will be assigned to each large cluster. By Lemma 5, with the proper tiebreaking scheme, the total number of distinct edges contained in the paths belonging to a single large cluster X_i is only $O(|X_i|\mathcal{E})$, so long as

$$|S| = O(r^{2(1-a)/(2b+a-1)}\mathcal{E}^{2/(2b+a-1)})$$

Some algebraic manipulation gives:

$$|S|^{(2b+a-1)/2} = O(r^{1-a}\mathcal{E})$$

$$|S|^{(2b+a-1)/2} n r^{a-1} = O(n\mathcal{E})$$

Recall that $r = n^{d-o(1)}$, so

$$|S|^{(2b+a-1)/2} n^{1+o(1)-d(1-a)} = O(n\mathcal{E})$$

So if this condition holds, then the total number of large edges in H is:

$$\begin{aligned}\sum_{X_i \text{ is large}} O(|X_i|\mathcal{E}) &= \mathcal{E} \sum_{X_i \text{ is large}} O(|X_i|) \\ &= O(\sum_i |X_i|) \\ &= \tilde{O}(n\mathcal{E})\end{aligned}$$

where the last equality follows from Lemma 3.

Total. The total number of edges in H is then $2 \cdot (\tilde{O}(n\mathcal{E}) + \tilde{O}(n\mathcal{E})) = \tilde{O}(n\mathcal{E})$, *assuming* from the first case that

$$|S|^{(2b+a-1)/2} n^{1+o(1)-d(1-a)} = O(n\mathcal{E})$$

We conclude that the total number of edges in H is $|S|^{(2b+a-1)/2} n^{1+o(1)-d(1-a)}$. □

5.2 Standard Spanners

Recall the following definition from the introduction:

DEFINITION 3. *[Additive spanner] A subgraph H is a* $+\beta$ spanner *of a graph G if*

$$\delta_H(u,v) \leq \delta_G(u,v) + \beta$$

for all $u, v \in V$.

In other words, an additive spanner is a subset spanner with $S = V$.

We generate our spanners using Algorithm 2.

LEMMA 8. *The output of Algorithm 2 is a $+O(n^d)$ spanner of G.*

Proof. Consider each pair $u, v \in V$. If we decided *not* to add paths $\rho_G(u, x_u)$ and $\rho_G(v, x_v)$, then it must be the case that $\delta_H(u,v) \leq \delta_G(u,v) + n^d$. If we did add paths $\rho_G(u, x_u)$ and $\rho_G(v, x_v)$, then let s_u be the node in S within distance n^d of x_u, and let s_v be the same for x_v. From the triangle inequality, we have:

$$\begin{aligned}\delta_H(u,v) \leq& \delta_H(u, x_u) + \delta_H(x_u, s_u) + \delta_H(s_u, s_v) \\ &+ \delta_H(s_v, x_v) + \delta_H(x_v, v)\end{aligned}$$

We know that $\delta_G(x_u, s_u) \leq n^d/\log n$. We have a $\cdot \log n$ multiplicative spanner of G in H, so that gives $\delta_H(x_u, s_u) \leq n^d$. The same argument holds for $\delta_H(x_v, s_v)$. Additionally, due to our subset spanner, we have $\delta_H(s_u, s_v) \leq \delta_G(s_u, s_v) + n^d$. We then have:

$$\delta_H(u,v) \leq \delta_G(u, x_u) + n^d + \delta_G(s_u, s_v) + n^d + \delta_G(x_v, v)$$

By the triangle inequality, we have $\delta_G(s_u, s_v) \leq \delta_G(x_u, x_v) + O(n^d)$. Therefore,

$$\delta_H(u,v) \leq \delta_G(u, x_u) + \delta_G(x_u, x_v) + \delta_G(x_v, v) + O(n^d)$$

Copyright © by SIAM.
Unauthorized reproduction of this article is prohibited.

Downloaded 02/10/16 to 171.64.66.62. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

Algorithm 2: span(G, d)

```
Initialize H to be a · log n multiplicative
  spanner of G;
Let
  ℰ = n^{(a+2b−1)/(a+2b+1)−d(10b−a+1)/(3(a+2b+1))};
Let S be a random sample of Θ(log n ·
    n^{1−d(2b−a+1)/(2b+a−1)}/ℰ^{(3−2b−a)/(2b+a−1)})
    nodes in G // The size of the constant
    in the Θ determines the probability
    of the algorithm being correct
Add a +n^d subset spanner of G, S to H;
for each pair u, v ∈ V such that
  δ_H(u, v) > δ_G(u, v) + 8n^d do
    Let x_u be the first node in ρ_G(u, v) with the
    property that there exists s ∈ S with
    δ_G(s, x_u) ≤ n^d/ log n and let x_v be the last
    such node;
    Add ρ_G(u, x_u) and ρ_G(v, x_v) to H;
end
return H;
```

Since x_u, x_v lie on $\delta_G(u, v)$, this implies

$$\delta_H(u, v) \leq \delta_G(u, v) + O(n^d)$$

□

We now need to prove the edge bound.

Overview of the Edge Bound. For each of the paths $\rho_G(u, x_u)$ that we add to H, we can bound the cost of its extreme subpaths and its small subpaths exactly like we did in our subset spanner. The only challenging part of this proof is the bound on the cost of the large subpaths. Think about a specific large cluster X. If it contains only a few large subpaths, then we can upper bound its density using Lemma 5. If it contains many large subpaths, then we can argue that the average cost of one of these large subpaths is fairly small. We then make another distinction: a *heavy* subpath is one that contributes a lot of edges to X, and a *light* subpath is one that is fairly cheap to add to X. Heavy subpaths are rare, and so they don't contribute very many edges in total. Light subpaths mean that the path has lots of nodes in its neighborhood (all of X) for a relatively small number of missing edges; therefore, by the time the path is missing $\Theta(n^d)$ edges, its neighborhood is very large. That makes it likely that there is a node $s \in S$ in this neighborhood.

We will now start to prove the bound more formally. First, we make the following refinement of Definition 11:

Definition 12. *Let $H \subset G$. We say that a large subpath p, owned by large cluster X, is a* heavy subpath *if the number of edges in p but not H is at least*

$$|X|^{(b+a-1)/b}\mathcal{E}^{(b-1)/b}$$

Otherwise, p is a light subpath.

The purpose of this definition is:

Lemma 9. *There exists a tiebreaking scheme ρ_G such that the following statement is true:*

Let $H \subset G$. Let Q be a sequence of node pairs that are all contained in the same large cluster X. Suppose we add $\rho_X(q)$ to H in some order for all $q \in Q$. Then only $O(|X|\mathcal{E})$ edges will be added to H by a heavy path.

Proof. When you consider a certain pair $q \in Q$, if there exists a light shortest path between its endpoints, then add that particular path to H; this pair q then does not contribute any edges to the heavy path edge count.

We are left to bound the edges only of those pairs whose path is heavy; suppose there are h such pairs in total. We will next prove that $h = O(|X|^{(1-a)/b}\mathcal{E}^{1/b}$. Suppose otherwise, towards a contradiction (so $h = \omega(|X|^{(1-a)/b}\mathcal{E}^{1/b})$). Choose ρ_X to implement a distance preserver on $O(|X|^a h^b)$ edges on these pairs. The average number of edges contributed by each pair is $O(|X|^a/h^{1-b})$, which is

$$O(|X|^a/\omega((|X|^{(1-a)/b}\mathcal{E}^{1/b})^{1-b}))$$
$$o(|X|^a/(|X|^{(1-b)(1-a)/b}\mathcal{E}^{(1-b/b)}))$$
$$o(|X|^{(b+a-1)/b}\mathcal{E}^{(b-1)/b}$$

Note that this is smaller than the threshold for a path to be heavy. This implies that one of our "heavy" pairs is in fact light – a contradiction. Therefore, $h = O(|X|^{(1-a)/b}\mathcal{E}^{1/b})$.

Now, the cost of a distance preserver on this number of pairs is

$$O(|X|^a(|X|^{(1-a)/b}\mathcal{E}^{1/b})^b) = O(|X|\mathcal{E})$$

edges, which proves the lemma. □

We need one more technical lemma:

Lemma 10. *In Algorithm 2, whenever we add $\rho_G(u, x_u)$ and $\rho_G(v, x_v)$ to H for some pair $u, v \in V$, there are at least $n^2/\log n$ edges missing from H in $\rho_G(u, x_u) \cup \rho_G(v, x_v)$.*

Proof. Suppose towards a contradiction that $\rho_G(u, x_u) \cup \rho_G(v, x_v)$ are missing at *most* $n^d/\log n$ edges in H. By the triangle inequality, we have:

$$\begin{aligned}\delta_H(u, v) \leq{} & \delta_H(u, x_u) + \delta_H(x_u, s_u) + \delta_H(s_u, s_v) \\ & + \delta_H(s_v, x_v) + \delta_H(x_v, v)\end{aligned}$$

Copyright © by SIAM.
Unauthorized reproduction of this article is prohibited.

Downloaded 02/10/16 to 171.64.66.62. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

Since H contains a $\cdot \log n$ spanner of G, our hypothesis implies that $\delta_H(u, x_u) + \delta_H(v, x_v) \leq \delta_G(u, x_u) + \delta_G(v, x_v) + n^d$. Similarly, $\delta_H(x_u, s_u) \leq \delta_G(x_u, s_u) + n^d$, since the distance between x_u and s_u is at most $n^d/\log n$ (and similar for $\delta_H(x_v, s_v)$. Finally, we have $\delta_H(s_u, s_v) \leq \delta_G(s_u, s_v)$, because H contains a $+n^d$ subset spanner of S. We now have

$$\begin{aligned}\delta_H(u,v) \leq& (\delta_G(u,x_u) + \delta_G(v,x_v) + n^d) + (\delta_G(x_u,s_u) + n^d)\\ &+ (\delta_G(s_u,s_v) + n^d) + (\delta_G(s_v,x_v) + n^d)\\ \delta_H(u,v) \leq& \delta_G(u,x_u) + \delta_G(x_u,s_u) + \delta_G(s_u,s_v) + \delta_G(s_v,x_v)\\ &+ \delta_G(x_v,v) + 4n^d\end{aligned}$$

Another application of the triangle inequality gives that $\delta_G(x_u, s_u) + \delta_G(s_u, s_v) + \delta_G(s_v, x_v) \leq \delta_G(x_u, x_v) + n^d$. We then have

$$\begin{aligned}\delta_H(u,v) \leq& \delta_G(u,x_u) + \delta_G(x_u,s_u) + \delta_G(s_u,s_v)\\ &+ \delta_G(s_v,x_v) + \delta_G(x_v,v) + 5n^d\\ \delta_H(u,v) \leq& \delta_G(u,v) + 5n^d\end{aligned}$$

and so the pair u, v has already been spanned accurately enough, and so we will not add $\rho_G(u, x_u)$ or $\rho_G(v, x_v)$ to H. This is a contradiction, and so it must be the case that $\rho_G(u, x_u) \cup \rho_G(v, x_v)$ is missing more than $n^d/\log n$ edges in H. $\square$

We can now prove:

LEMMA 11. *For all G, there is a tiebreaking scheme ρ_G such that Algorithm 2 returns a graph on $n^{1+o(1)+(a+2b-1)/(a+2b+1)-d(10b-a+1)/(3(a+2b+1))}$ edges.*

Proof. Recall that

$$\mathcal{E} = n^{(a+2b-1)/(a+2b+1)-d(10b-a+1)/(3(a+2b+1))}$$

and so it suffices to prove that there are $n^{1+o(1)}\mathcal{E}$ edges in the graph returned by Algorithm 2.

Once again, the $\cdot \log n$ multiplicative spanner costs only $\tilde{O}(n)$ edges. The total cost of the subset spanner, implemented with Theorem 4, is

$$\begin{aligned}&n^{1-d/3}(\Omega(\log n \cdot n^{1-d(2b-a+1)/(2b+a-1)}/\\ &\qquad \mathcal{E}^{(3-2b-a)/(2b+a-1)}))^{1/2}\\ &\tilde{\Omega}(n^{1-d/3}(n^{1/2-d(2b-a+1)/(2(2b+a-1))}/\\ &\qquad \mathcal{E}^{(3-2b-a)/(2(2b+a-1))}))\end{aligned}$$

One can verify that

$$\begin{aligned}n\mathcal{E} =& n^{1-d/3}\cdot\\ &(n^{1/2-d(2b-a+1)/(2(2b+a-1))}/\mathcal{E}^{(3-2b-a)/(2(2b+a-1))})\end{aligned}$$

as follows:

$$\begin{aligned}&\mathcal{E}^{1+(3-2b-a)/(2(2b+a-1))} =\\ &\qquad n^{-d/3}(n^{1/2-d(2b-a+1)/(2(2b+a-1))})\\ &\mathcal{E}^{(1+2b+a)/(2(2b+a-1))} =\\ &\qquad n^{1/2-d(1/3+(2b-a+1)/(2(2b+a-1)))}\\ &\mathcal{E}^{(1+2b+a)} = n^{(2b+a-1)-d(2(2b+a-1)/3+(2b-a+1))}\\ &\mathcal{E}^{(1+2b+a)} = n^{(2b+a-1)-d\cdot(10b-a+1)/3}\end{aligned}$$

Substituting in

$$\mathcal{E} = n^{(a+2b-1)/(a+2b+1)-d(10b-a+1)/(3(a+2b+1))}$$

we get

$$n^{(a+2b-1)-d(10b-a+1)/3} = n^{(2b+a-1)-d\cdot(10b-a+1)/3}$$

which is true, and so the subset spanner fits within our edge budget. We now need to bound the edges added by paths $\rho_G(u, x_u)$ and $\rho_G(v, x_v)$. We will imagine a clustering $\{x_i, v_i\}$ of G with r chosen such that $\max_i r_i \leq n^d/(32 \log n)$. Once again, we will say that an edge is Extreme/Small/Large (and that a large edge is heavy or light) based on the classification of the subpath of $\rho_G(u, x_u)$ that first added this edge to H. We will again count each edge type separately.

Extreme Edges. There are at most $n^2/(2 \log n)$ extreme edges in $\rho_G(u, x_u) \cup \rho_G(v, x_v)$ (they belong to four clusters - at the beginning and end of $\rho_G(u, x_u)$ and $\rho_G(v, x_v)$ - and each cluster has diameter $n^d/(8 \log n)$). Further, from Lemma 10, we know that $\rho_G(u, x_u) \cup \rho_G(v, x_v)$ is missing at least $n^2/\log n$ edges.

We conclude that only a constant fraction of the total edges in H are extreme, and so it suffices to prove our edge bound for the remaining cases.

Small Edges. This case is identical to the Small Edges case in Theorem 4.

Large Edges. Large edges can be either heavy or light. By Lemma 9, each large cluster owns only $O(|X_i|\mathcal{E})$ heavy edges, and so the total number of heavy edges is

$$\sum_{X_i \text{ is large}} O(|X_i|\mathcal{E}) = \mathcal{E} \sum_{X_i \text{ is large}} |X_i| = \tilde{O}(n\mathcal{E})$$

To bound the number of light edges, we will argue that there are more heavy edges than there are light edges and so the same bound applies. To see this, assume towards a contradiction that there are more light edges than heavy edges. We know that at least $n^d/\log n$ edges are missing in $\rho_G(u, x_u) \cup \rho_G(v, x_v)$. Suppose at least half these edges are light, and let $\mathcal{L}$ be the set of large clusters that own a light subpath of $\rho_G(u, x_u)$ or

Copyright © by SIAM.
Unauthorized reproduction of this article is prohibited.

$\rho_G(v, x_v)$. Suppose that all the clusters in $\mathcal{L}$ have the minimum possible size for a large cluster; that is, for all $L \in \mathcal{L}$ we have $|L| = r^{2b/(2b+a-1)}\mathcal{E}^{1/(2b+a-1)}$ (we will later show that this is a worst-case assumption). Then we have:

$$|\mathcal{L}| \geq \frac{n^d}{2\log n}/(r^{2b/(2b+a-1)}\mathcal{E}^{1/(2b+a-1)})^{(b+a-1)/b}\mathcal{E}^{(b-1)/b})$$

$$|\mathcal{L}| \geq \frac{n^d}{2\log n}/(r^{2(b+a-1)/(2b+a-1)}\mathcal{E}^{(b+a-1)/(b(2b+a-1))}) \mathcal{E}^{(b-1)/b})$$

$$|\mathcal{L}| \geq \frac{n^d}{2\log n}/(r^{2(b+a-1)/(2b+a-1)}\mathcal{E}^{(2b+a-2)/(2b+a-1)})$$

And so

$$\sum_{L\in\mathcal{L}} |L| \geq \frac{n^d}{2\log n}/(r^{2(b+a-1)/(2b+a-1)}\mathcal{E}^{(2b+a-2)/(2b+a-1)}) \cdot r^{2b/(2b+a-1)}\mathcal{E}^{1/(2b+a-1)}$$

$$\sum_{L\in\mathcal{L}} |L| \geq \frac{n^d}{2\log n} \cdot r^{2(1-a)/(2b+a-1)}\mathcal{E}^{(3-2b-a)/(2b+a-1)}$$

We have $r = n^{d-o(1)}$, so

$$\sum_{L\in\mathcal{L}} |L| \geq n^{d(2b-a+1)/(2b+a-1)-o(1)}\mathcal{E}^{(3-2b-a)/(2b+a-1)}$$

Note that if our assumption fails – i.e. we have $|L| \geq r^{2b/(2b+a-1)}\mathcal{E}^{1/(2b+a-1)}$ – then by convexity, our lower bound on $\sum_{L\in\mathcal{L}} |L|$ can only become stronger and so this inequality will still hold.

Note, however, that the size of our random sample of S is

$$\Omega(n \log n/(n^{d(2b-a+1)/(2b+a-1)-o(1)}\mathcal{E}^{(3-2b-a)/(2b+a-1)}))$$

and therefore, with high probability, there is a node $s \in S$ in some cluster $L \in \mathcal{L}$. This implies that there is a node $s \in S$ within distance $< n^d/\log n$ of some node $w \in \rho_G(u, x_u) \cup \rho_G(v, x_v)$ – a contradiction. We then have that the number of light edges is strictly less than the number of heavy edges.

Total. This shows that the total number of edges in H is $n^{1+o(1)}\mathcal{E}$. By the previous discussion, we have set $\mathcal{E}$ such that this bound suffices to prove the lemma. $\square$

Jointly, Lemmas 8 and 11 imply:

THEOREM 5. *Let a, b be constants such that all graphs and pair sets have a distance preserver on $O(n + n^a|P|^b)$ edges. Then for any constant $d > 0$, all graphs have $+O(n^d)$ spanners on $n^{1+o(1)+(a+2b-1)/(a+2b+1)-d(10b-a+1)/(3(a+2b+1))}$ edges.*

References

[ACIM99] D. Aingworth, C. Chekuri, P. Indyk, and R. Motwani. Fast estimation of diameter and shortest paths (without matrix multiplication). *SIAM J. Comput.*, 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 of Discrete Math*, pages 5(2): 151–162, 1992.

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

[BK06] S. Baswana and T. Kavitha. Faster algorithms for approximate distance oracles and all-pairs small stretch paths. *Proc. 47th FOCS*, pages 591–602, 2006.

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

[BKMP10] Surender Baswana, Telikepalli Kavitha, Kurt Mehlhorn, and Seth Pettie. Additive spanners and (α, β)-spanners. *ACM Transactions on Algorithms*, 7(1):5, 2010.

[BS07] S. Baswana and S. Sen. A simple and linear time randomized algorithm for computing sparse spanners in weighted graphs. *Journal of Random Structures and Algorithms 30*, pages 4, 532–563, 2007.

[BW15] G. Bodwin and V. Vassilevska Williams. Very sparse additive spanners and emulators. *Innovations in Theoretical Computer Science (ITCS)*, 2015.

[CE06] D. Coppersmith and M. Elkin. Sparse sourcewise and pairwise distance preservers. *SIAM Journal on Discrete Mathematics*, pages 463–501, 2006.

[CGK13] M. Cygan, F. Grandoni, and T. Kavitha. On pairwise spanners. *Symposium on Theoretical Aspects of Computer Science (STACS)*, 2013.

[Che13] Shiri Chechik. New additive spanners. *SODA*, pages 498–512, 2013.

[Coh93] E. Cohen. Fast algorithms for constructing t-spanners and paths with stretch t. *Proc. 34th FOCS*, 1993.

[Cow01] L. J. Cowen. Compact routing with minimum stretch. *Journal of Algorithms*, pages 28, 170–183, 2001.

[CW04] L. J. Cowen and C. G. Wagner. Compact roundtrip routing in directed networks. *Journal of Algorithms*, pages 50, 1, 79–95, 2004.

[DHZ96] Dorit Dor, Shay Halperin, and Uri Zwick. All pairs almost shortest paths. *Proc. 37th FOCS*, pages 452–461, 1996.

[Elk05] M. Elkin. Computing almost shortest paths. *ACM Trans. Algorithms*, pages 1(2):283–323, 2005.

[Elk07] M. Elkin. A near-optimal distributed fully dynamic algorithm for maintaining sparse spanners. *Proc. 26th PODC*, pages 185–194, 2007.

[EP04] M. Elkin and D. Peleg. $(1 + \epsilon, \beta)$-spanner con-

Downloaded 02/10/16 to 171.64.66.62. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

Copyright © by SIAM.
Unauthorized reproduction of this article is prohibited.

structions for general graphs. *SIAM J. Comput., 33(3):608–631*, 2004.

[EP15] M. Elkin and S. Pettie. A linear size logarithmic stretch path-reporting distance oracle for general graphs. *Proc. SODA*, 2015.

[Erd64] P. Erdös. Extremal problems in graph theory. *Theory of graphs and its applications*, pages 29–36, 1964.

[EZ06] M. Elkin and J. Zhang. Efficient algorithms for constructing $(1+\epsilon,\beta)$-spanners in the distributed and streaming models. *Distributed Compting 18*, pages 5, 375–385, 2006.

[FPZW04] A. M. Farley, A. Proskurowski, D. Zappala, and K. Windisch. Spanners and message distribution in networks. *Discrete Applied Mathematics*, pages 137(2):159–171, 2004.

[Kav15] T. Kavitha. New pairwise spanners. *Proc. 42nd ICALP*, 2015.

[Knu14] Mathias Bæk Tejs Knudsen. Additive spanners: A simple construction. *Symposium and Workshop on Algorithm Theory (SWAT)*, pages 277–281, 2014.

[KV13] T. Kavitha and N. Varma. Small stretch pairwise spanners. *Proc. 40th ICALP*, 2013.

[LS91] Arthur L Liestman and Thomas C Shermer. Additive spanners for hypercubes. *Parallel Processing Letters*, 1(01):35–42, 1991.

[LS93] Arthur L Liestman and Thomas C Shermer. Additive graph spanners. *Networks*, 23(4):343–363, 1993.

[Pet09] Seth Pettie. Low distortion spanners. *ACM Transactions on Algorithms, 6(1)*, 2009.

[PR10] M. Pătraşcu and L. Roditty. Distance oracles beyond the thorup-zwick bound. *Proc. 51st FOCS*, 2010.

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

[PU89a] D. Peleg and J. D. Ullman. An optimal synchronizer for the hypercube. *SIAM Journal of Computing*, pages 18, 740–747, 1989.

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

[RTZ05] L. Roditty, M. Thorup, and U. Zwick. Deterministic constructions of approximate distance oracles and spanners. *Proc. 32nd ICALP*, pages 261–272, 2005.

[RTZ08] L. Roditty, M. Thorup, and U. Zwick. Roundtrip spanners and roundtrip routing in directed graphs. *ACM Trans. Algorithms*, page 3(4): Article 29, 2008.

[RZ04] L. Roditty and U. Zwick. On dynamic shortest paths problems. *Proc. 12th ESA*, pages 580–591, 2004.

[ST04] D. A. Spielman and S. H. Teng. Nearly-linear time algorithms for graph partitioning, graph sparsification, and solving linear systems. *STOC*, 2004.

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

[TZ05] M. Thorup and U. Zwick. Approximate distance oracles. *Journal of the ACM 52*, pages 1, 1–24, 2005.

[TZ06] M. Thorup and U. Zwick. Spanners and emulators with sublinear distance errors. *Proc. 17th SODA*, pages 802–809, 2006.

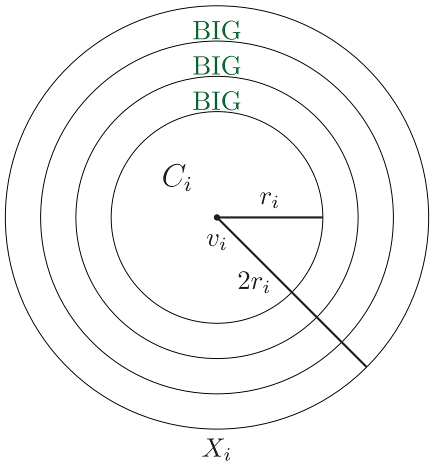
[Woo06] D. P. Woodruff. Lower bounds for additive spanners, emulators, and more. *Proc. 47th FOCS*, pages 389–398, 2006.

[Woo10] D. P. Woodruff. Additive spanners in nearly quadratic time. *Proc. 37th ICALP*, pages 463–474, 2010.

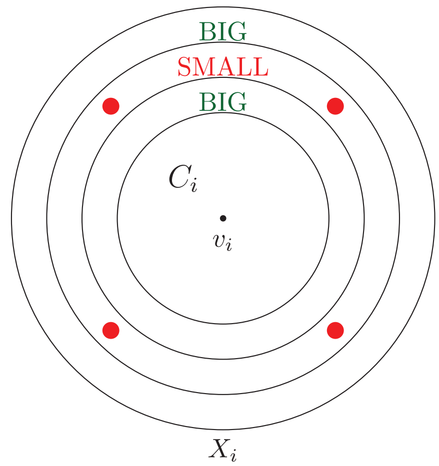
Downloaded 02/10/16 to 171.64.66.62. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

Copyright © by SIAM.
Unauthorized reproduction of this article is prohibited.

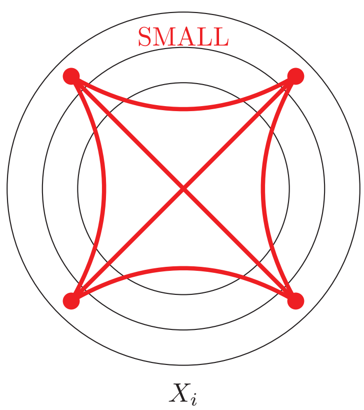
Downloaded 02/10/16 to 171.64.66.62. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php



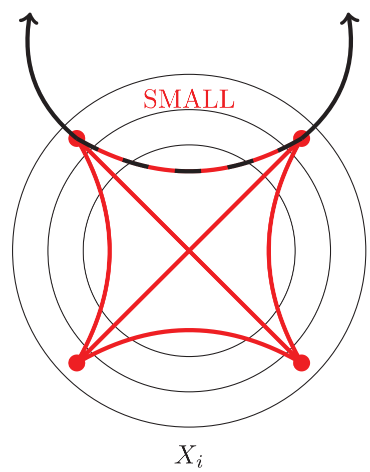
(a) Perhaps each subsequent ring around the core contains a lot of nodes. In this case, the size of the entire cluster must be fairly big, and so the cluster is classified as "large."



(b) Alternately, perhaps there exists a specific ring around the core that doesn't contain very many nodes. In this case ...



(c) ...we restrict attention to the subgraph of nodes contained in this small ring. Because the ring is small, it is not very expensive to add a distance preserver on all pairs of nodes in this ring.

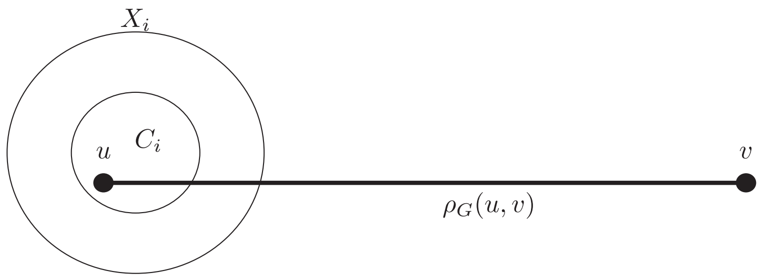


(d) Now, every time a shortest path enters and leaves the cluster, we have already handled all the edges of this path inside the small ring.

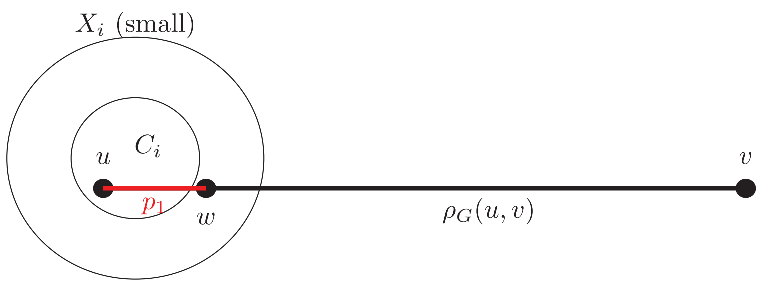
Figure 5: A graphical depiction of the reduction between distance preservers and graph clustering.

Copyright © by SIAM.
Unauthorized reproduction of this article is prohibited.

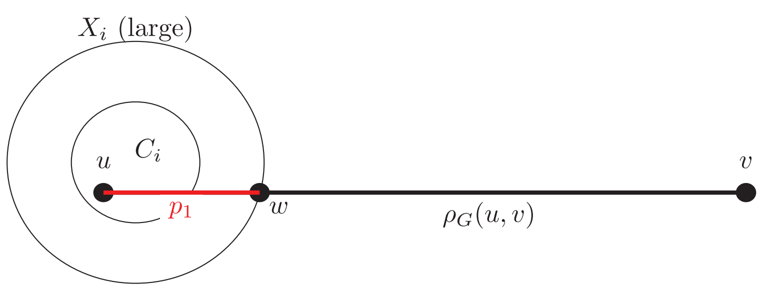
Downloaded 02/10/16 to 171.64.66.62. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php



(a) Look at the first node of your shortest path p. Find a cluster X_i that contains the first node of p in its core.



(b) First, suppose X_i is small. Then we partition p at the first node $w \notin C_i$, and repeat the analysis on $\rho_G(w, v)$.



(c) Otherwise, suppose that X_i is large. In this case, we let w be the *last* node such that $\rho_G(u, w) \subset X_i$, partition p over w, and repeat the analysis on $\rho_G(w, v)$. In this case a triangle inequality argument implies that $\rho_G(w, v)$ and C_i are disjoint, so we will never again choose the cluster X_i.

Figure 6: How to decompose a shortest path $\rho_G(u, v)$ over a graph clustering (Lemma 6).

Copyright © by SIAM.
Unauthorized reproduction of this article is prohibited.