# Additive Spanners in Nearly Quadratic Time

David P. Woodruff

IBM Almaden

**Abstract.** We consider the problem of efficiently finding an additive C-spanner of an undirected unweighted graph G, that is, a subgraph H so that for all pairs of vertices  $u, v, \, \delta_H(u, v) \leq \delta_G(u, v) + C$ , where  $\delta$  denotes shortest path distance. It is known that for every graph G, one can find an additive 6-spanner with  $O(n^{4/3})$  edges in  $O(mn^{2/3})$  time. It is unknown if there exists a constant C and an additive C-spanner with  $O(n^{4/3})$  edges. Moreover, for  $C \leq 5$  all known constructions require  $\Omega(n^{3/2})$  edges.

We give a significantly more efficient construction of an additive 6-spanner. The number of edges in our spanner is  $n^{4/3}$  polylog n, matching what was previously known up to a polylogarithmic factor, but we greatly improve the time for construction, from  $O(mn^{2/3})$  to  $n^2$  polylog n. Notice that  $mn^{2/3} \leq n^2$  only if  $m \leq n^{4/3}$ , but in this case G itself is a sparse spanner. We thus provide both the fastest and the sparsest (up to logarithmic factors) known construction of a spanner with constant additive distortion.

We give similar improvements in the construction time of additive spanners under the assumption that the input graph has large girth, or more generally, the input graph has few edges on short cycles.

### 1 Introduction

An additive C-spanner (often called a (1, C)-spanner) of an unweighted undirected graph G is a subgraph H with the property that for all vertices  $u, v, \delta_H(u, v) \leq \delta_G(u, v) + C$ . Here C is known as the distortion.

Spanners have a variety of applications. They are used in space-efficient routing tables that guarantee almost shortest routes [1], [10], [11], [17], [20], methods for simulating synchronized protocols in unsynchronized networks [16], parallel and distributed algorithms for computing almost shortest paths [7], [8], [14], and in labeling schemes and distance oracles for reporting approximate distances [4], [6], [19], [21].

There is a tradeoff between the distortion and the sparsity of the spanner. Studying what sparsity is possible for small constant C has received considerable attention. For C=2, Aingworth et al (see [2], and the followup work [5], [12], [15], [18], [22]) show that  $O(n^{3/2})$  edges are necessary and sufficient; that is, every graph G contains an additive 2-spanner with this many edges, and there exist graphs for which any additive 2-spanner requires this many edges. Later, Baswana et al [5] show that every graph G contains an additive 6-spanner with  $O(n^{4/3})$  edges. Nothing better than  $O(n^{4/3})$  is known even if the distortion is

allowed to be any constant, though lower bounds that degrade with the distortion are known [23].

Another important measure is the time complexity needed for constructing such a spanner. As many graph algorithms for distance-approximation [2], [5], [9], [12] have running time proportional to the number of edges, a key approach is to first find a spanner of the input graph, and then run existing algorithms on the spanner rather than the dense input. The value of this approach is diminished if the time to find the spanner is already too large.

The additive 2-spanner of Aingworth et~al~has construction time  $O(mn^{1/2})$ . This was improved by Dor, Halpern, and Zwick [12] to  $\tilde{O}(n^2)$  time<sup>1</sup> and  $\tilde{O}(n^{3/2})$  edges. The conference version of the additive 6-spanner construction of Baswana et~al~[5] had O(mn) time, but was improved to  $O(mn^{2/3})$  in the journal version [3], and the improvement is attributed to Elkin [13]. Notice that  $O(mn^{2/3})$  is much larger than  $\tilde{O}(n^2)$ . Indeed,  $m > n^{4/3}$ ; otherwise G itself serves as a sparse spanner. Hence,  $mn^{2/3} > n^2$ .

It should be noted that prior to the work of Baswana et al [5], Dor, Halpern, and Zwick [12] constructed what they called a 6-emulator of a graph with  $\tilde{O}(n^{4/3})$  edges and  $\tilde{O}(n^2)$  time. A 6-emulator H of a graph G=(V,E) is a weighted graph on the same vertex set V with an arbitrary edge set E', subject to the constraint that  $d_G(u,v) \leq d_H(u,v) \leq d_G(u,v) + 6$ . Emulators are often much easier to construct than spanners. For instance, it is known that every graph has a 4-emulator containing  $\tilde{O}(n^{4/3})$  edges [12], but the best known additive 4-spanner has size  $\Theta(n^{3/2})$ . It was not until the much later work of [5] that an additive 6-spanner with  $O(n^{4/3})$  edges was constructed, albeit with a much larger running time.

#### 1.1 Results

Our main contribution is a new construction of an additive 6-spanner with  $\tilde{O}(n^{4/3})$  edges in a significantly faster  $\tilde{O}(n^2)$  time. We thus provide both the fastest and the sparsest (up to logarithmic factors) known construction of a spanner with constant additive distortion.

Our techniques also solve the following problem: given a subset S of  $O(n^{2/3})$  vertices of a graph G, find a subgraph H of G containing  $\tilde{O}(n^{4/3})$  edges so that for all  $u, v \in S$ ,  $\delta_H(u, v) \leq \delta_G(u, v) + 2$ . Our method solves this problem in  $\tilde{O}(n^2)$  time. The best previous time was  $O(mn^{2/3})$ , using a technique of [3], attributed to Elkin.

In an attempt to achieve even sparser additive spanners, Baswana et al [5] study the construction of spanners by parameterizing the number of edges appearing on any short cycle. Let  $\Gamma_k(G)$  be the number of edges in G that lie on a cycle with length at most 2k. They show that for any integers  $k \geq 1$  and  $\ell \in [0, k]$ , there exists an additive  $(2k + 4\ell)$ -spanner with  $O(\Gamma_k(G) + n^{1 + \frac{1}{k + \ell + 1}})$  edges that can be constructed in  $O(mn^{1 - \frac{\ell}{k + \ell + 1}})$  time. We show that an additive

Let  $\tilde{O}(f(n))$  denote the class of functions that are bounded above by a function that grows as  $f(n) \cdot \text{polylog}(f(n))$ .

 $(2k+4\ell)$ -spanner with  $\tilde{O}(\Gamma_k(G)+n^{1+\frac{1}{k+\ell+1}})$  edges can be constructed in  $\tilde{O}(n^2)$  time. This gives a sizable improvement upon the previous time bound for a wide range of k and  $\ell$ .

As an example setting of parameters, if the input graph has girth greater than 4, we can construct an additive 4-spanner (resp. additive 8-spanner) with  $\tilde{O}(n^{4/3})$  edges (resp.  $\tilde{O}(n^{5/4})$  edges) in  $\tilde{O}(n^2)$  time, whereas the prior bound was O(mn) time (resp.  $\tilde{O}(mn^{3/4})$ ) time.

#### 1.2 Techniques

Our method is based on a new path-hitting framework. This approach is quite different than the path-purchasing methods used in [5] to construct an additive 6-spanner. Namely, our work is the first to look at hitting the neighborhood of a path in the absence of high-degree vertices, provided there are enough vertices of moderate degree along the path.

In more detail, as in [3], we initially include all edges incident on low-degree  $(< n^{1/3})$  vertices in the spanner. We then find a dominating set  $D_1$  of size  $\tilde{O}(n^{2/3})$  of all vertices of degree at least  $n^{1/3}$ . Each vertex in  $D_1$  can be thought of as a cluster center, as in [3], and for each vertex incident to at least one cluster center, we include an edge to one such center in our spanner. Now our algorithm deviates from that of [3]. That algorithm needs to compute a breadth-first search (BFS) tree around each cluster center. It then uses these trees to iteratively choose walks (i.e., paths with repeated vertices and edges) to include in the spanner. The main problem is the BFS tree computations, taking  $O(mn^{2/3})$  time, which is too costly. We now explain how to overcome this problem.

We first describe a new combinatorial way of looking at the problem. Suppose each shortest path P has x edges missing from the spanner. Then there are  $\Omega(n^{1/3}x)$  vertices which are at distance 1 from some vertex in P. This follows from the fact that the neighborhoods of two vertices that are at a distance larger than 2 on P cannot intersect, by the triangle inequality. Therefore, if we choose a random sample  $D_2$  of  $\tilde{O}(n^{2/3}/x)$  of the n vertices, we will hit the neighborhood of each shortest path. If  $u \in D_2$  is in the neighborhood of P, and  $v_1, v_r \in D_1$  are neighbors of the first and last missing edge in P, then there are almost shortest walks from u to  $v_1$  and u to  $v_r$  with a total of roughly x missing edges. If we include the missing edges on these walks to the spanner for all pairs of vertices in  $D_1$  and  $D_2$ , the total number of edges added is  $\tilde{O}((n^{2/3}/x) \cdot n^{2/3} \cdot x) = \tilde{O}(n^{4/3})$ . Since x is not the same for all shortest paths, we need to repeat this procedure for (a logarithmic number of) geometrically increasing intervals of x.

There are several technical hurdles we need to overcome to implement this in  $\tilde{O}(n^2)$  time. First, we need a procedure, which, given a subgraph of the input, finds almost shortest walks between pairs of vertices with the smallest number of missing edges efficiently. For this bicriteria problem, we design a procedure FHEASW-Solver which runs in  $\tilde{O}(m')$  time, where m' is the number of edges in the subgraph. This procedure is based on a BFS-like dynamic program. We can then run FHEASW-Solver from each vertex in  $D_2$ , provided for the given value of x and the shortest paths P with x edges missing, we can find a subgraph

which contains P and that has  $\tilde{O}(n^{4/3}x)$  edges. Indeed, in this case the time of the invocations of FHEASW-Solver will be  $\tilde{O}(|D_2|n^{4/3}x) = \tilde{O}(n^2)$ . If we choose the subgraph to be all edges with one endpoint of degree at most  $n^{1/3}x$ , then it has at most  $\tilde{O}(n^{4/3}x)$  edges, as we need. Unfortunately this subgraph may not contain P if P contains a vertex of degree larger than  $n^{1/3}x$ .

This leads us to our final case. If P contains a vertex of degree larger than  $n^{1/3}x$ , then we can compute another dominating set  $D_3$  of all vertices of degree at least d in G, where d is the largest degree of a vertex along P. This dominating set has size  $\tilde{O}(n/d)$ . We can then connect each vertex in this dominating set to each of the vertices in  $D_1$  via an almost shortest walk missing at most  $d/n^{1/3}$  edges. Indeed, by assumption  $d>n^{1/3}x$ , or equivalently, the number x of missing edges is at most  $d/n^{1/3}$ . The total number of edges added is therefore  $\tilde{O}(|D_1|(n/d)(d/n^{1/3})) = \tilde{O}(n^{4/3})$ . To find the edges to add in this step we need to run yet another invocation of FHEASW-Solver. We run it on a subgraph of O(nd) edges, once from each vertex in  $D_3$ , therefore taking  $\tilde{O}(|D_3|nd) = \tilde{O}(n^2)$  total time. Since d is not the same for all shortest paths, we need to vary it in geometrically increasing intervals as well. This idea of varying the degree and working in subgraphs is similar to ideas used for additive 2-spanners of Dor, Halpern, and Zwick [12].

An analysis of the union of the edgesets from the two cases shows that for any pair of vertices, there is a path in the spanner between them with distortion at most 6. It turns out that achieving this guarantee requires running FHEASW-Solver several times on the same subgraph with different parameters. We remark that our algorithm is Monte Carlo, that is, with high probability an additive-6 spanner is returned. We do not know how to derandomize the algorithm in  $\tilde{O}(n^2)$  time due to the fact that we need to hit the neighborhood of many paths, but can only implicitly represent the paths in the allotted  $\tilde{O}(n^2)$  time.

## 2 Preliminaries

The input is an unweighted undirected graph G = (V, E) on a set V of n vertices and a set E of m edges. W.l.o.g., n is a power of 8. Let  $\deg(u)$  be the degree of  $u \in V$ , and let N(u) be the set of neighbors of u. For a set S of vertices, let N(S) be  $\bigcup_{u \in S} N(u)$ .

Suppose E' is an arbitrary subset of E. For  $u \in V$ , let BFS(u, (V, E')) denote a shortest-path tree rooted at u in the graph (V, E'). The following theorem is standard and stated here for reference. See, e.g., the discussion in [9] or [12].

**Theorem 1.** (BFS) Given an unweighted undirected graph G = (V, E) and a vertex  $u \in V$ , there is an O(m+n)-time algorithm that finds distances, and a tree of shortest paths, from u to all the vertices of V.

All logarithms, if unspecified, are to the base 2. We will classify all edges of our input graph G according to their type: an edge  $e = \{u, v\}$  is of type i if  $\min(\deg(u), \deg(v)) \in [2^i, 2^{i+1})$ . An edge is  $\mathit{light}$  if it is of type i for an  $i \leq \log n^{1/3}$ . Otherwise, it is  $\mathit{heavy}$ . We assume the standard RAM model on  $O(\log n)$ -sized words in which arithmetic operations take O(1) time.

## 3 Constructing Additive 6-spanners

#### 3.1 A Subroutine

In our main algorithm, we need a subroutine which solves the following problem. Fix an unweighted undirected graph G = (V, E). Fix a root vertex  $u \in V$ . For each vertex  $v \in V$ , suppose we know  $\delta_G(u, v)$ . We would like to find a walk from u to v with the least number of heavy edges among all walks from u to v whose distance is at most  $\delta_G(u, v) + C$  for a constant C > 0. More precisely, for a walk  $P \mid \text{let } \phi(P)$  be the number of heavy edges along P. We sometimes abuse notation and let  $\phi\{u, v\} = \phi(\{u, v\})$ .

Fewest Heavy Edges with Almost Shortest Walks (FHEASW): Given G, a vertex u, and a constant C, output a data structure for which on input  $v \in V$  and  $i \in \{\delta_G(u,v), \delta_G(u,v) + 1, \ldots, \delta_G(u,v) + C\}$ , there is an algorithm that returns, in  $O(\phi(P))$  time, the heavy edges along a walk P from u to v in G with  $\delta_P(u,v) = i$  and  $\phi(P) = \min_{\text{walks } P' \mid \delta_{P'}(u,v) = i} \phi(P')$ .

Note that we allow P to be a walk rather than just a path, that is, it is a path that may contain repeated vertices and edges.

We can solve this problem as follows. For a  $v \in V$  and  $i \in \{\delta_G(u, v), \delta_G(u, v) + 1, \ldots, \delta_G(u, v) + C\}$ , let D(v, i) be the minimum number of heavy edges on a walk to u with length  $\delta_G(u, v) + i$ . If there is no such walk, we define  $D(v, i) = \infty$ . Let S(j) be the set of vertices w reachable by a walk from u of length exactly j, and for which  $j \leq \delta_G(u, w) + C$ . Then,

$$D(v,i) = \min_{w \in S(\delta_G(u,v)+i-1) \cap N(v)} D(w,i-1) + \phi(w,v)$$
(1)

We can build up the values D(v,i) as follows. We first obtain  $\delta_G(u,v)$  for all v by running the algorithm of Theorem 1. Then to create S(j) given S(j-1), we include each  $v \in N(S(j-1))$  for which  $j \leq \delta_G(u,v) + C$ . The time to do this is O(|N(S(j-1))|). When computing N(S(j-1)), we can also update the appropriate D(v,i) values by Equation (1). Hence, the total time is, up to a constant factor,  $\sum_j |N(S(j-1))|$ , where  $S(0) = \{u\}$ . The key point is that, since C is constant, any given vertex can occur in at most C+1 different sets S(j). It follows that the total time is O(Cm) = O(m).

While we have shown how to calculate the costs D(v, i), to solve the FHEASW problem we must also return all heavy edges along a walk from u to v with length  $\delta_G(u, v) + i$  and containing at most D(v, i) heavy edges. One can do this by keeping side information in the algorithm above. Namely, each time D(v, i) increases, we can append the heavy edge to a running list of heavy edges along the walk. This does not affect the overall time complexity by more than a constant factor. Due to space constraints, we defer further details to the full version of this paper.

# Spanner Construction(G = (V, E)):

- 1. Compute the type of each edge and store these values.
- 2. Initialize F to the set of light edges.
- 3. Repeat the following steps  $3 \log n$  times:
  - (a) Let R be a random sample of  $2n^{2/3}$  vertices. For each vertex in  $[n] \setminus R$ , if there is an edge to a vertex in R, add one such edge to F.
  - (b) For  $i = \log n$ ,  $(\log n) 1$ ,  $(\log n) 2$ , ...,  $(\log n^{1/3}) + 1$ ,
    - i. Let E' be the subset of E of edges of type at most i.
    - ii. For  $j = 0, 1, 2, \dots, \lceil \log 3n/2^i \rceil$ ,
      - A. Let  $S_{i,j}$  be a random subset of  $\lceil 3n/2^{i+j} \rceil$  vertices.
      - B. For  $u \in S_{i,j}$ ,  $D_u = \text{FHEASW-Solver}((V, E'), u, 4)$ .
      - C. For  $u \in S_{i,j}$ ,  $v \in R$ ,  $z \in \{\delta_{(V,E')}(u,v), \delta_{(V,E')}(u,v) + 1, \ldots, \delta_{(V,E')}(u,v) + 4\}$ , add to F the heavy edges on the walk from u to v given by  $D_u$  with query input z, provided the number of heavy edges on the walk is  $\leq 2^{i+j+1}/n^{1/3} + 2$ .
- 4. Output H = (V, F).

# **Theorem 2.** $|F| = O(n^{4/3} \log^3 n)$ .

*Proof.* The number of light edges added to F is  $O(n^{4/3})$ . The number of edges added in step 3a, over all  $3 \log n$  iterations of step 3, is  $O(n \log n)$ .

For each iteration i of step 3b, at most  $\sum_{j} |R| |S_{i,j}| (2^{i+j+1}/n^{1/3} + 2) = \sum_{j} O(n^{2/3}) \cdot O(n/2^{i+j}) \cdot (2^{i+j+1}/n^{1/3} + 2) = O(n^{4/3} \log n) + \sum_{j} O(n^{5/3})/2^{i+j} = O(n^{4/3} \log n)$  edges are added to F, where the last inequality follows from the fact that  $2^{i+j} \geq n^{1/3}$ . The number of iterations of step 3b is  $O(\log n)$ , and step 3 is invoked  $3 \log n$  times, resulting in  $|F| = O(n^{4/3} \log^3 n)$ .

**Theorem 3.** Spanner Construction can be implemented in  $O(n^2 \log^2 n)$  time.

*Proof.* In step 1 we classify each edge as light or heavy, and assign the corresponding weight. This can be done in O(m) time.

A single iteration of step 3a takes at most O(m) time. Hence, over all  $O(\log n)$  iterations, step 3a takes  $O(m \log n) = O(n^2 \log n)$  time.

For a single iteration of step 3biiB, for a fixed value of i,j, and  $u \in S_{i,j}$ , FHEASW-Solver takes time  $O(n2^i)$ , since there are at most  $O(n2^i)$  edges in the subgraph. As there are  $O(n/2^{i+j})$  different  $u \in S_{i,j}$ , it follows that step 3biiB takes  $O(n^2/2^j)$  time. Hence, summing over all iterations and i and j, step 3biiB takes  $O(n^2\log^2 n)$  time.

A single iteration of step 3biiC takes at most  $O(|S_{i,j}||R|(2^{i+j+1}/n^{1/3}+2)) = O(n/2^{i+j}) \cdot O(n^{2/3})(2^{i+j+1}/n^{1/3}+2) = O(n^{4/3})$  time. Summing over iterations, step 3biiC takes  $\tilde{O}(n^{4/3})$  time.

Hence, the total time of the algorithm is  $O(n^2 \log^2 n)$ .

It remains to argue the algorithm's correctness. We start with a lemma. If  $P = (v_1, \ldots, v_s)$  is a shortest path from vertex  $v_1$  to  $v_s$ , we define  $N(P) = \bigcup_{i=1}^s N(v_i)$ .

**Lemma 1.** In the graph (V, E') for any  $E' \subseteq E$ , if there is a shortest path P from u to v containing  $\ell$  vertices of degree at least  $2n^{1/3}$ , then  $|N(P)| \ge 2\ell n^{1/3}/3$ .

Proof. Let  $w_1, w_2, \ldots, w_\ell$  be the sequence of vertices of degree at least  $2n^{1/3}$  along P (possibly with other vertices in between). Observe that for each j,  $N(w_j)$  must be disjoint from  $\bigcup_{j' \geq j+3} N(w_{j'})$ . Otherwise one could go from  $w_j$  to a vertex  $x \in N(w_j) \cap (\bigcup_{j' \geq j+3} N(w_{j'}))$ , then to a vertex  $w_{j'}$  for some  $j' \geq j+3$ , in two steps. As  $\delta_P(w_j, w_{j'}) \geq 3$ , this contradicts P being a shortest path, since each of its sub-paths must be shortest. As  $|N(w_j)| \geq 2n^{1/3}$  for all j, it follows that  $|\bigcup_{j=1}^{\ell} N(w_j)| \geq 2\ell n^{1/3}/3$ .

**Theorem 4.** With probability at least 1 - 1/n, H is an additive 6-spanner.

*Proof.* Fix a pair  $\{a,b\}$  of vertices in G, and fix any shortest path P from a to b in G with the fewest heavy edges. We assume that there is at least one heavy edge on P, as otherwise the path P will be added to F in step 2. So there are at least two vertices of degree at least  $2n^{1/3}$  on P. Let  $w_1, w_2, \ldots, w_r$  be the ordered sequence of vertices on P of degree at least  $2n^{1/3}$ , where  $w_1$  is closest to a and a is closest to a.

Consider one iteration of step 3. We show that with probability at least 3/8, using only the edges added to F in step 2 and the current iteration of step 3, there is a path of length at most  $\delta_G(a,b) + 6$  from a to b. It will follow that the probability that there is some iteration for which there is a path of length at most  $\delta_G(a,b) + 6$  is at least  $1 - (3/8)^{3 \log n} = 1 - 1/n^3$ . By a union bound, it will follow that for every pair of vertices, there is such a path with probability at least 1 - 1/n, that is, H is an additive 6-spanner.

Let  $i^*$  be such that all edges along P have type at most  $i^*$ , and there is at least one edge  $e^*$  of type  $i^*$ . Observe that  $i^* \in \{\log n, (\log n) - 1, (\log n) - 2, \dots, (\log n^{1/3}) + 1\}$ . We shall only consider the  $i^*$ -th iteration of step 3b. Notice that the entire path P is included in the edgeset E'. We do a case analysis.

Case 1: The path P contains at most  $2^{i^*}/n^{1/3}$  heavy edges.

In this case we shall only consider the iteration in which j=0 of step 3bii. Now,  $e^*$  is of type  $i^*$ , which means that at least one of its endpoints y has degree in the range  $[2^{i^*}, 2^{i^*+1})$ . It follows that all edges incident to y have type at most  $i^*$ , and in particular, are included in the graph (V, E'). Consider the following event  $\mathcal{E}: S_{i^*,0} \cap N(y) \neq \emptyset$ . Then,

$$\Pr[\mathcal{E}] \ge 1 - \left(1 - \frac{|N(y)|}{n}\right)^{|S_{i^*,0}|} \ge 1 - \left(1 - \frac{2^{i^*}}{n}\right)^{3n/2^{i^*}} \ge 1 - \frac{1}{e}.$$

Conditioned on  $\mathcal{E}$ , let  $u \in S_{i^*,0} \cap N(y)$ . Let  $E^*$  be the set of edges added in step 3a, and consider the event  $\mathcal{F}$ :  $\exists v_1 \in R$  for which  $\{w_1, v_1\} \in E^*$  and  $\exists v_r \in R$  for which  $\{w_r, v_r\} \in E^*$ . Since  $\operatorname{degree}(w_1), \operatorname{degree}(w_r) \geq 2n^{1/3}$ ,  $\Pr[\mathcal{F}] \geq 1 - 2\left(1 - \frac{2n^{1/3}}{n}\right)^{2n^{2/3}} \geq 1 - 2e^{-4}$ . By a union bound,  $\Pr[\mathcal{E} \wedge \mathcal{F}] \geq 1 - \frac{1}{e} - \frac{2}{e^4} > \frac{3}{8}$ .

Conditioned on  $\mathcal{E} \wedge \mathcal{F}$ , let  $v_1$  be a vertex in R for which  $\{w_1, v_1\} \in E^*$ , and  $v_r$  be a vertex in R for which  $\{w_r, v_r\} \in E^*$ . Consider the walk Q from u to  $v_1$  which first traverses edge  $\{u, y\}$ , then agrees with path P from y to  $w_1$ , then traverses edge  $\{w_1, v_1\}$ . Observe that the number of heavy edges along Q is at most  $2^{i^*}/n^{1/3} + 2$ , since P contains at most  $2^{i^*}/n^{1/3}$  heavy edges. Moreover, by the triangle inequality, the walk Q is of length at most  $\delta_G(u, v_1) + 4 \leq \delta_{(V, E')}(u, v_1) + 4$ . It follows that in step 3biiC, there will be a walk Q' added to F from  $v_1$  to  $v_2$  of length at most  $\delta_{(V, E')}(w_1, y) + 2$ . Similarly, there will be a walk Q'' added to P from P0 to P1 to P2.

Hence, the walk P' from a to b which agrees with P from a until  $w_1$ , then goes to  $v_1$ , then takes the walk Q' to u, then the walk Q'' from u to  $v_r$ , then goes to  $w_r$ , then agrees with P from  $w_r$  to b, is of length at most  $|P'| \leq \delta_G(a, w_1) + 1 + |Q'| + |Q''| + 1 + \delta_G(w_r, b)$ , which is at most  $\delta_G(a, w_1) + 1 + \delta_{(V,E')}(w_1, y) + 2 + \delta_{(V,E')}(y, w_r) + 2 + 1 + \delta_G(w_r, b)$ , which is at most  $\delta_G(a, b) + 6$ .

Case 2: The path P contains more than  $2^{i^*}/n^{1/3}$  heavy edges. Let  $j^*$  be such that the number of heavy edges on P is in the interval  $[2^{i^*+j^*}/n^{1/3}, 2^{i^*+j^*+1}/n^{1/3})$  Each heavy edge on P is of type at most  $i^*$ , and so one of the two endpoints must have degree in the range  $[2n^{1/3}, 2^{i^*+1})$ . It follows that all of the edges incident to this endpoint in G are included in the graph (V, E'). It follows that there are at least  $2^{i^*+j^*-1}/n^{1/3}$  vertices of degree at least  $2n^{1/3}$  on P in the graph (V, E'). By Lemma 1, |N(P)| is therefore at least  $2^{i^*+j^*}/3$ . Notice that this is also at most n, and therefore  $2^{j^*} \leq 3n/2^{i^*}$ . Hence there is an iteration of step 3bii for which  $j=j^*$ . We only consider this iteration.

By Lemma 1,  $|N(P)| \geq 2^{i^*+j^*}/3$ . Consider the following event  $\mathcal{E}: S_{i^*,j^*} \cap N(P) \neq \emptyset$ . Then,  $\Pr[\mathcal{E}] \geq 1 - \left(1 - \frac{|N(P)|}{n}\right)^{|S_{i^*,j^*}|} \geq 1 - \left(1 - \frac{2^{i^*+j^*}}{3n}\right)^{3n/2^{i^*+j^*}} \geq 1 - \frac{1}{e}$ . Conditioned on  $\mathcal{E}$ , let  $u \in S_{i^*j^*} \cap N(P)$  and let  $y \in P$  be such that  $\{u,y\} \in E'$ . As in case 1, letting  $E^*$  be the set of edges added in step 3a, we have that with probablity at least 3/8 event  $\mathcal{E}$  occurs and there is a vertex  $v_1 \in R$  for which  $\{w_1,v_1\} \in E^*$ , and a vertex  $v_r \in R$  for which  $\{w_r,v_r\} \in E^*$ .

As in case 1, consider the walk Q from u to  $v_1$ , traversing edge  $\{u,y\}$ , agreeing with P from y to  $w_1$ , then traversing edge  $\{w_1,v_1\}$ . This walk contains at most  $2^{i^*+j^*+1}/n^{1/3}+2$  heavy edges and has length at most  $\delta_{(V,E')}(u,v_1)+4$ , so in step 3biiC there will be a walk Q' added to F from  $v_1$  to u of length at most  $\delta_{(V,E')}(w_1,y)+2$ . And as before, there will be a walk Q'' added to F from u to  $v_r$  of length at most  $\delta_{(V,E')}(y,w_r)+2$ . It follows that as in case 1 that there is a walk P' from a to b in the spanner with length at most  $\delta_G(a,b)+6$ . This completes the proof.

## 4 Construction for Inputs with Large Girth

Given a graph G with m edges and n vertices, let  $\Gamma_k(G)$  be the number of edges in G that lie on a cycle with length at most 2k. Choose  $\varepsilon$  so that  $n^{\varepsilon} = (k^2 n)^{1/(k+\ell+1)}$ . We need the following lemma of Baswana *et al.* 

**Lemma 2.** ([5]) There are clusterings  $C_{\ell}$  and  $C_k$  of vertices of G with the following properties: (1) for  $i \in \{\ell, k\}$ , each cluster  $C \in C_i$  is a rooted spanning tree with radius at most i, and (2) the number of clusters in  $C_i$  is  $n^{1-i\varepsilon}$ . The subgraph  $H_{k,\ell}$  containing all such spanning trees as well as every edge incident to a vertex that does not appear in both clusterings  $C_{\ell}$  and  $C_k$  can be constructed in O(m) time and satisfies  $\mathbf{E}[|H_{k,\ell}|] = O(\Gamma_k(G) + n^{1+\varepsilon})$ .

Now we define a heavy edge to be an edge that does not appear in  $H_{k,\ell}$ . It is easy to see that Theorem FHEASW-Solver continues to work with this new definition of a heavy edge, that is, FHEASW-Solver solves the FHEASW problem (with respect to this new definition of heavy) in O(m+n) time, for constant k and  $\ell$ .

We let Base-Edges denote the algorithm guaranteed by Lemma 2. The following is our main algorithm.

# InterClusterSpanner (G = (V, E)):

- 1. Initialize F to the output of  $\mathsf{Base\text{-}Edges}(G)$ . For  $i \in \{k, \ell\}$ , let  $V_i$  denote the set of centers of vertices in clusters in  $\mathcal{C}_i$ .
- 2. Repeat the following steps  $3 \log n$  times:
  - (a) For  $i = \log n$ ,  $(\log n) 1$ ,  $(\log n) 2$ , ..., 1,
    - i. Let  $S_i$  be a random sample of  $2n/2^i$  vertices. For each vertex in  $[n] \setminus S_i$ , if there is an edge to a vertex in  $S_i$ , add one edge to F.
    - ii. Put  $q = \lfloor \log n^{-k\varepsilon} 2^i/(2k) \rfloor$ . For  $j = q, q+1, \ldots, \lceil \log n^{1-k\varepsilon} \rceil$ , sample a set  $S_{i,j} \subseteq V_k$  of  $\lceil n^{1-k\varepsilon} 2^{-j} \rceil$  vertices.
    - iii. Let  $E' \subseteq E$  be the set of edges of type at most i.
    - iv. For each  $u \in S_i \cup (\cup_{\text{all } j} S_{i,j})$ ,
      - A. Compute  $D_u = \text{FHEASW-Solver}((V, E'), u, \ell + k)$ .
      - B. For  $v \in V_{\ell}$ ,  $z \in \{\delta_{(V,E')}(u,v), \delta_{(V,E')}(u,v) + 1, \ldots, \delta_{(V,E')}(u,v) + 2\ell + 2k\}$ , add to F the heavy edges along the walk from u to v given by  $D_u$  with query input z, provided it has at most  $2^i n^{-k\varepsilon} + \ell + 1$  heavy edges if  $u \in S_i$  and  $2^i \geq n^{k\varepsilon}$ , or at most  $4k2^j + \ell + k$  heavy edges if  $u \in S_{i,j}$  for some j.
- 3. Output H = (V, F).

Our analysis will assume that the output of Base-Edges in step 1 has  $O(\Gamma_k(G) + n^{1+\varepsilon})$  edges. This can be done with probability  $1-1/n^3$  by running the algorithm of Lemma 2  $O(\log n)$  times and taking the output with the least number of edges.

**Theorem 5.** 
$$|F| = O(\Gamma_k(G) + n^{1 + \frac{1}{k + \ell + 1}} \log^3 n)$$
.

*Proof.* The number of edges added in step 1 is  $O(\Gamma_k(G) + n^{1 + \frac{1}{k + \ell + 1}})$  by Lemma 2 and the definition of  $\varepsilon$ .

The number of edges added in step 2(a)i is O(n) per iteration, and thus  $O(n \log^2 n)$  over all  $3 \log n$  iterations of step 2 and all choices of i.

The number of edges added in step 2(a)ivB due to  $u \in S_i$  is at most  $|S_i| \cdot |V_\ell| \cdot (2^i n^{-k\varepsilon} + \ell + 1) = O(n2^{-i})n^{1-\ell\varepsilon}(2^i n^{-k\varepsilon} + \ell + 1) = O(n^{2-(k+\ell)\varepsilon} + n^{2-\ell\varepsilon}2^{-i})$ . Notice that paths are only added if  $2^i \ge n^{k\varepsilon}$ , and so this expression is bounded

by  $O(n^{2-(k+\ell)\varepsilon})$ . Hence, over all  $3 \log n$  iterations of step 2 and all choices of i, this is bounded by  $O(n^{2-(k+\ell)\varepsilon}\log^2 n)$ .

The number of edges added in step 2(a)ivB due to  $u \in S_{i,j}$  for a fixed j is at most  $|S_{i,j}| \cdot |V_{\ell}| \cdot (4k2^j + \ell + k) = O(n^{1-k\varepsilon}2^{-j}) \cdot n^{1-\ell\varepsilon} \cdot (4k2^j + \ell + k) = O(n^{2-(k+\ell)\varepsilon})$ , and so as we range over all j it is bounded by  $O(n^{2-(k+\ell)\varepsilon}\log n)$ . Since step 2(a)ivB is invoked  $O(\log^2 n)$  times, using the definition of  $\varepsilon$  we see that the total number of edges in H meets the claimed bound.

# **Theorem 6.** InterClusterSpanner can be implemented in $O(n^2 \log^2 n)$ time.

Proof. Step 1 can be implemented in O(m) time by Lemma 2. Fix some iteration of step 2a. Then steps 2ai, 2aii, and 2aiii can be implemented in O(n+m) time. Step 2aivA runs in time  $O((|S_i|+|\cup_{\text{all }j}S_{i,j}|)n2^i)=O(n^2)+O(|S_{i,q}|n2^i)$ . Here we used that (1)  $n^{1-k\varepsilon}2^{-j}$  is geometrically decreasing in j, (2)  $|S_{i,j}|=\lceil n^{1-k\varepsilon}2^{-j}\rceil$ , and (3) j is bounded above by  $\lceil \log n^{1-k\varepsilon}\rceil$ . Therefore,  $|\cup_{\text{all }j}S_{i,j}|=O(|S_{i,q}|)$ . Now,  $|S_{i,q}|=O(n^{1-k\varepsilon}/(n^{-k\varepsilon}2^i))=O(n/2^i)$ . Hence, step 2a(iv)A runs in time  $O(n^2)$ . Over all iterations of step 2 and choices of i in step 2a, this gives  $O(n^2\log^2 n)$  time.

By definition of FHEASW, step 2aivB runs in time  $O(|V_{\ell}|(|S_i|2^in^{-k\varepsilon} + \sum_j |S_{i,j}|(4k2^j))) = \tilde{O}(n^{1-\ell\varepsilon}(n2^{-i}\cdot 2^in^{-k\varepsilon} + \sum_j n^{1-k\varepsilon}2^{-j}(4k2^j)))$ , which equals  $O(n^{2-(k+\ell)\varepsilon}\log n)$ . Here we used the relation  $2^in^{-k\varepsilon} + \ell + 1 = \Theta(2^in^{-k\varepsilon})$ , which does not hold in general, but if we add the heavy edges along a path for  $u \in S_i$  in step 2aivB we require that  $2^i \geq n^{k\varepsilon}$ , so in this case the relation holds. Over all invocations of step 2 and choices of i in step 2a, this results in  $O(n^{2-(k+\ell)\varepsilon}\log^3 n)$  time, and thus is dominated by the time for iterations of step 2aivA.

### **Theorem 7.** With probability $\geq 1 - 1/n$ , H is an additive $(2k + 4\ell)$ -spanner.

*Proof.* Fix a pair  $\{a,b\}$  of vertices in G and a shortest path P between them. We assume there is at least one heavy edge along P (recall that we have modified the definition of heavy), otherwise P will be added to F in step 1. Let  $w_1, \ldots, w_r$  be the ordered sequence of vertices in P ( $w_1$  is closest to a, and  $w_r$  is closest to b) that appear in both  $\mathcal{C}_\ell$  and  $\mathcal{C}_k$  (note that  $r \geq 2$  since there is a heavy edge along P). Let  $v_1, \ldots, v_r$  be the centers of the clusters in  $\mathcal{C}_\ell$  containing  $w_1, \ldots, w_r$ , respectively.

Consider one iteration of step 2. We show that with probability at least 3/8, using only the edges added to F in step 1 and the current iteration of step 2, there is a path of length at most  $\delta_G(a,b) + 2k + 4\ell$  from a to b. Hence the probability that there is some iteration for which there is a path of length at most  $\delta_G(a,b) + 2k + 4\ell$  is at least  $1 - (3/8)^{3\log n} \ge 1 - 1/n^3$ . By a union bound, it will follow that for every pair of vertices, there is such a path with probability at least 1 - 1/n, that is, H is an additive  $(2k + 4\ell)$ -spanner.

Recall that an edge  $\{u,v\}$  is of type i if  $\min(\deg(u),\deg(v)) \in [2^i,2^{i+1})$ . We let  $i^*$  be such that all edges along P have type at most  $i^*$ , and there is at least one edge  $e^*$  of type  $i^*$ . We shall only consider the  $i^*$ -th iteration of step 2a. Notice that the entire path P is included in the edgeset E'.

Case 1: P contains at most  $n^{-k\varepsilon}2^{i^*}$  heavy edges. Since  $e^*$  is of type  $i^*$ , one of its endpoints y has degree in the range  $[2^{i^*}, 2^{i^*+1})$ . Hence, all edges incident to y have type at most  $i^*$ , and are included in the graph (V, E'). Since  $S_{i^*}$  is a randomly chosen set of size  $2n/2^{i^*}$ , with probability at least  $1-\left(1-\frac{2^*}{n}\right)^{2n/2^{i^*}} \ge 3/8$ , there is a vertex  $u \in S_{i^*} \cap N(y)$ . We condition on this event.

Consider the walk Q from u to  $v_1$  which first traverses edge  $\{u,y\}$ , then agrees with path P from y to  $w_1$ , then traverses at most  $\ell$  edges along the spanning tree from  $w_1$  to  $v_1$  in the cluster centered at  $v_1$ . Observe that the number of heavy edges along this walk is at most  $n^{-k\varepsilon}2^{i^*}+\ell+1$ . Also, since P contains at most  $n^{-k\varepsilon}2^{i^*}$  heavy edges and at least 1 heavy edge,  $2^{i^*} \geq n^{-k\varepsilon}$ , as needed by step 2aivB. Moreover, by the triangle inequality, the walk Q is of length at most  $\delta_G(w_1,y)+\ell+1\leq \delta_G(u,y)+2\ell+2\leq \delta_G(u,y)+2\ell+2k$ . It follows that in step 2aivB, there will be a walk Q' added to F from  $v_1$  to  $v_2$  of length at most  $\delta_{(V,E')}(w_1,y)+\ell+1$ . Similarly, there will be a walk Q'' added to F from  $v_2$  to  $v_3$  to  $v_4$  from  $v_4$  to  $v_4$  then goes to  $v_4$ , then takes the walk  $v_2$  to  $v_3$ , then the walk  $v_3$  from  $v_4$  to  $v_5$ , then goes to  $v_4$ , then agrees with  $v_4$  from  $v_4$  to  $v_5$ , is of length at most  $v_4$ , then goes to  $v_4$ , then agrees with  $v_5$  from  $v_6$  to  $v_7$ , then goes to  $v_7$ , then agrees with  $v_7$  from  $v_7$  to  $v_7$  then agrees with  $v_7$  from  $v_7$  to  $v_7$  then agrees with  $v_7$  from  $v_7$  to  $v_7$  to  $v_7$  to  $v_7$  then agrees with  $v_7$  from  $v_7$  to  $v_7$  to  $v_7$  to  $v_7$  then goes to  $v_7$ , then agrees with  $v_7$  from  $v_7$  to  $v_7$  then goes to  $v_7$ , then agrees with  $v_7$  from  $v_7$  to  $v_7$ 

Case 2: P contains more than  $n^{-k\varepsilon}2^{i^*}$  heavy edges. Let h denote the number of heavy edges along P. Since P is a shortest path, all vertices along P are distinct, and so there are at least h of them that appear in a cluster in  $\mathcal{C}_k$ . We need the slightly stronger property that there is a set T of  $t \geq \lceil h/(2k) \rceil$  distinct vertices  $v \in V_k$  for which some vertex in the cluster centered at v appears along P. Indeed, otherwise there would exist two vertices along P in the same cluster in  $\mathcal{C}_k$  at a distance larger than 2k from each other, contradicting that P is a shortest path. Observe that  $t \leq h+1$ , the number of heavy edges. Let  $j^* \geq 0$  be such that  $t \in [2^{j^*}, 2^{j^*+1})$ . Since  $2^j$  ranges from  $n^{-k\varepsilon}2^{i^*}/(2k)$  to at least  $n^{1-k\varepsilon}$ , we can consider  $j=j^*$  in step 2(a)ii.

Consider the following event  $\mathcal{E}: S_{i^*,j^*} \cap T \neq \emptyset$ . Then,  $\Pr[\mathcal{E}]$  is at least  $1 - \left(1 - \frac{t}{n^{1-k\varepsilon}}\right)^{|S_{i^*,j^*}|} \geq 1 - \left(1 - \frac{2^{j^*}}{n^{1-k\varepsilon}}\right)^{\lceil n^{1-k\varepsilon}2^{-j^*}\rceil} > \frac{3}{8}$ . Conditioned on  $\mathcal{E}$ , let  $v \in S_{i^*,j^*} \cap T$ , and let u be the vertex in the cluster centered at v in  $\mathcal{C}_k$  that lies along P. Consider the walk Q which first traverses the at most k edges on the spanning tree from v to u, then agrees with path P from u to  $w_1$ , then traverses the at most  $\ell$  edges along the spanning tree from  $w_1$  to  $v_1$  in the cluster centered at  $v_1$ . The number of heavy edges along this walk is at most  $h + \ell + k \leq 2kt + \ell + k \leq 4k2^{j^*} + \ell + k$ . By the triangle inequality, the walk Q is of length at most  $\delta_G(u, w_1) + k + \ell \leq \delta_G(v, v_1) + 2k + 2\ell$ . It follows that in step 4aiiC, there will be a walk Q' added to F from  $v_1$  to  $v_2$  of length at most  $\delta_{(V,E')}(u,w_1) + \ell + k$ . Similarly, there will be a walk Q'' added to F from  $v_2$  to  $v_3$  of length at most  $v_4$  in the proof of length at most  $v_4$  in the walk  $v_4$  in the proof of length at most  $v_4$  in the proof of  $v_4$  in the proof of length at most  $v_4$  in the proof of length at length  $v_4$  in the proof of length  $v_4$  in the proof of length  $v_4$  in the proof

Acknowledgment: The author would like to thank Michael Elkin, Vitaly Feldman, Mihai Pătrașcu, Seth Pettie, and Anastasios Sidiropoulos.

#### References

- Ittai Abraham, Cyril Gavoille, and Dahlia Malkhi. On space-stretch trade-offs: upper bounds. In SPAA, pages 217–224, 2006.
- Donald Aingworth, Chandra Chekuri, Piotr Indyk, and Rajeev Motwani. Fast estimation of diameter and shortest paths (without matrix multiplication). SIAM J. Comput., 28(4):1167–1181, 1999.
- 3. S. Baswana, T. Kavitha, K. Mehlhorn, and S. Pettie. Additive spanners and  $(\alpha, \beta)$ spanners. *ACM Transactions on Algorithms*, 2009.
- 4. Surender Baswana and Telikepalli Kavitha. Faster algorithms for approximate distance oracles and all-pairs small stretch paths. In FOCS, pages 591–602, 2006.
- Surender Baswana, Telikepalli Kavitha, Kurt Mehlhorn, and Seth Pettie. New constructions of (alpha, beta)-spanners and purely additive spanners. In SODA, pages 672–681, 2005.
- Surender Baswana and Sandeep Sen. Approximate distance oracles for unweighted graphs in õ(n<sup>2</sup>) time. In SODA, pages 271–280, 2004.
- Edith Cohen. Fast algorithms for constructing t-spanners and paths with stretch t. SIAM J. Comput., 28(1):210–236, 1998.
- Edith Cohen. Polylog-time and near-linear work approximation scheme for undirected shortest paths. J. ACM, 47(1):132–166, 2000.
- Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein. Introduction to Algorithms, Second Edition. The MIT Press and McGraw-Hill Book Company, 2001.
- Lenore Cowen. Compact routing with minimum stretch. J. Algorithms, 38(1):170– 183, 2001.
- 11. Lenore Cowen and Christopher G. Wagner. Compact roundtrip routing in directed networks. J. Algorithms, 50(1):79–95, 2004.
- Dorit Dor, Shay Halperin, and Uri Zwick. All-pairs almost shortest paths. SIAM J. Comput., 29(5):1740–1759, 2000.
- 13. M. Elkin. Personal communication, 2009.
- 14. Michael Elkin. Computing almost shortest paths. ACM Transactions on Algorithms, 1(2):283–323, 2005.
- 15. Michael Elkin and David Peleg. (1+epsilon, beta)-spanner constructions for general graphs. SIAM J. Comput., 33(3):608–631, 2004.
- David Peleg and Jeffrey D. Ullman. An optimal synchronizer for the hypercube. SIAM J. Comput., 18(4):740–747, 1989.
- 17. David Peleg and Eli Upfal. A trade-off between space and efficiency for routing tables. J. ACM, 36(3):510–530, 1989.
- 18. Seth Pettie. Low distortion spanners. In ICALP, pages 78–89, 2007.
- Liam Roditty, Mikkel Thorup, and Uri Zwick. Deterministic constructions of approximate distance oracles and spanners. In ICALP, pages 261–272, 2005.
- 20. Mikkel Thorup and Uri Zwick. Compact routing schemes. In SPAA, 2001.
- Mikkel Thorup and Uri Zwick. Approximate distance oracles. J. ACM, 52(1):1–24, 2005.
- Mikkel Thorup and Uri Zwick. Spanners and emulators with sublinear distance errors. In SODA, pages 802–809, 2006.
- David P. Woodruff. Lower bounds for additive spanners, emulators, and more. In FOCS, pages 389–398, 2006.