THE EMERGENCE OF SPARSE SPANNERS AND WELL-SEPARATED PAIR DECOMPOSITION UNDER ANARCHY

Jie Gao[*] *and Dengpan Zhou*[†]

Abstract. A spanner graph on a set of points in $\mathbb{R}^d$ provides shortest paths between any pair of points with lengths at most a constant factor of their Euclidean distance. A spanner with a sparse set of edges is thus a good candidate for network backbones, as desired in many practical scenarios such as the transportation network and peer-to-peer network overlays. In this paper we investigate new models and aim to interpret why good spanners 'emerge' in reality, when they are clearly built in pieces by agents with their own interests and the construction is not coordinated. We show that the following algorithm of constructing an edge pq, if and only if there is no existing edge $p'q'$ with p' and q' at distances no more than $\frac{1}{4(1+1/\varepsilon)} \cdot |pq|$ from p, q respectively, generates a $(1+\varepsilon)$-spanner with a linear number of edges. The algorithm also implies a simple algorithm for constructing linear-size well-separated pair decompositions that may of interest on their own. This new spanner construction algorithm has applications in the construction of nice network topologies for peer-to-peer systems, when peers join and leave the network and has only limited information about the rest of the network.

1 Introduction

A geometric graph G defined on a set of points $P \subseteq \mathbb{R}^d$ with all edges as straight line segments of weight equal to their length is called a *Euclidean spanner*, if for any two points $p, q \in P$ the shortest path in G has length at most $t \cdot |pq|$ where $|pq|$ is the Euclidean distance. The factor t is called the *stretch factor* of G and the graph G is called an t-spanner. Spanners with a sparse set of edges provide good approximations to the pairwise Euclidean distances and are good candidates for network backbones. Thus, there has been a lot of work on the construction of sparse Euclidean spanners in both centralized [18, 39] and distributed settings [41].

In this paper we are interested in the emergence of good Euclidean spanners formed by uncoordinated agents. Many real-world networks, such as the transportation network and the Internet backbone network, are good spanners — one can typically drive from any city to any other city in the U.S. with the total travel distance at most a small constant times their straight line distance. The same thing happens with the Internet backbone graph. However, these large networks are not owned or built by any single authority. They are often assembled with pieces built by different governments or different ISPs, at different points in time. Nevertheless altogether they provide a convenient sparse spanner. The work

[*] *Stony Brook University, Computer Science Department*, jgao@cs.sunysb.edu
[†] *Stony Brook University, Computer Science Department*, dpzhou@cs.sunysb.edu

in this paper is motivated by this observation of the lack of coordination in reality and we would like to interpret why a good Euclidean spanner is able to 'emerge' from these agents incrementally.

Prior work that attempt to remove centralized coordination has been done, as in the network creation game [20, 14, 32, 4, 37], first introduced by Fabrikant *et al.* [20] to understand the evolution of network topologies maintained by selfish agents. A cost function is assigned to each agent, capturing the cost paid to build connections to others minus the benefit received due to the resulting network topology. The agents play a game by minimizing their individual costs and one is interested in the existence and the price of anarchy of Nash equilibria. Though being theoretically intriguing, there are two major open questions along this direction. First, the choice of cost functions is heuristic. Almost all past literatures use a unit cost for each edge and they deviate in how the benefit of 'being connected to others' is modeled. There is little understanding on what cost function best captures reality yet small variation in the cost function may result in big changes in the network topologies at Nash equilibria. There is also not much understanding of the topologies at Nash equilibria, some of them are simplistic topologies such as trees or complete graphs, that do not show up often in the real world. It remains open whether there is a natural cost model with which the Nash equilibrium is a sparse spanner.

The game theoretic model also has limitations capturing the reality: selfish agents may face deadlines and have to decide on building an edge or not immediately; once an edge is built, it probably does not make sense to remove it (as in the case of road networks); an agent may not have the strategies of all other agents making the evaluation of the cost function difficult. In this paper, we take a different approach and ask whether there is any simple rule, with which each agent can determine on its own, and collectively build and maintain a sparse spanner topology without any necessity of coordination or negotiation. The simple rule serves as a 'certificate' of the sparse spanner property that warrants easy spanner maintenance under edge dynamics and node insertion. We believe such models and good algorithms under these models are worth further exploration and this paper makes a first step along this line.

1.1 Our contribution

We consider in this paper the following model that abstracts the scenarios explained earlier. There are n points in the plane. Each point represents a separate agent and may consider to build edges from itself to other points. These decisions can happen at different points in time. When an agent p plans on an edge pq, p will only build it if there does not exist a 'nearby' edge $p'q'$ in the network, where $|pp'|$ and $|qq'|$ are within $\frac{1}{4(1+1/\varepsilon)} \cdot |p'q'|$ from p and q respectively. This strategy is very intuitive — if there is already a cross-country highway from Washington D.C. to San Francisco, it does not make economical sense to build a highway from New York to Los Angeles. We assume that each agent will eventually check on each possible edge from itself to all other points, but the order on who checks which edge can be *completely arbitrary*. With this strategy, the agents only make decisions with limited information and no agent has full control over how and what graph will be constructed. It is not obvious that this strategy will lead to a sparse spanner. It is not clear that the graph is even connected.

The main result in this paper is to prove that with the above strategy executed in *any* arbitrary order, the graph built at the end of the process is a sparse spanner:

- The stretch factor of the spanner is $1+\varepsilon$.
- The number of edges is $O(n)$.
- The total edge length of the spanner is $O(|\mathrm{MST}| \cdot \log \alpha)$, where α is the *aspect ratio*, i.e., the ratio of the distance between the furthest pair and the closest pair, and $|\mathrm{MST}|$ is the total edge length of the minimum spanning tree of the point set. The $\log \alpha$ factor could be improved to $O(\log n)$ by using a different technique.
- The degree of each point is $O(\log \alpha)$ in the worst case and $O(1)$ on average.

To explain how this result is proved, we first obtain as a side product the following *greedy* algorithm for computing a well-separated pair decomposition. A pair of two sets of points, (A, B), is called *s-well-separated* if the smallest distance between any two points in A, B is at least s times greater than the diameters of A and B. An *s-well-separated pair decomposition* (s-WSPD for short) for P is a collection of s-well-separated pairs $\mathcal{W} = \{(A_i, B_i)\}$ such that for any pair of points $p, q \in P$ there is a pair $(A, B) \in \mathcal{W}$ with $p \in A$ and $q \in B$. The size of an s-WSPD is the number of point set pairs in $\mathcal{W}$. Well-separated pair decomposition (WSPD) was first introduced by Callahan and Kosaraju [11] and they developed algorithms for computing an s-WSPD with linear size for points in $\mathbb{R}^d$. Since then WSPD has found many applications in computing k-nearest neighbors, n-body potential fields, geometric spanners and approximate minimum spanning trees [8, 9, 11, 10, 6, 5, 38, 35, 27, 19].

So far there are three algorithms for computing optimal size WSPD, in [11], [31] and in [24]. All three of them use a hierarchical organization of the points (e.g., the fair split tree in [11], the compressed quadtree in [31] and the discrete center hierarchy in [24]) and output the well-separated pairs in a recursive way. In this paper we show the following simple algorithm also outputs an s-WSPD with linear size. We take an *arbitrary* pair of points p, q that is not yet covered in any existing well-separated pair, and consider the pair of subsets $(B_r(p), B_r(q))$ with $r = |pq|/(2s+2)$ and $B_r(p)$ ($B_r(q)$) as the set of points of P within distance r from p (q). Clearly $(B_r(p), B_r(q))$ is an s-well-separated pair and all the pairs of points (p', q') with $p' \in B_r(p)$ and $q' \in B_r(q)$ are covered. The algorithm continues until all pairs of points are covered. We show that, no matter in which order the pairs are selected, the greedy algorithm will always output a linear number of well-separated pairs. Similarly, this algorithm can be executed in an environment when coordination is not present, while the previous algorithms (in [11, 24]) cannot.

WSPD is deeply connected to geometric spanners. Any WSPD will generate a spanner if one puts an edge between an arbitrary pair of points p, q from each well-separated pair $(A, B) \in \mathcal{W}$ [6, 5, 38, 35]. The number of edges in the spanner equals the size of $\mathcal{W}$. In the other direction, the deformable spanner proposed in [24] implies a WSPD of linear size. The connection is further witnessed in this paper: our spanner emergence algorithm implies a WSPD generated under anarchism. Thus the well-separated pairs and spanner edges are in one-to-one correspondence.

Last, this paper focuses on the Euclidean case when the points are distributed in the plane. The basic idea extends naturally to points in higher dimensions as well as metrics with constant doubling dimensions [28] (thus making the results introduced in previous section applicable in non-Euclidean settings), as the main technique involves essentially various forms of geometric packing arguments. Sparse spanners and WSPD exist for metrics with constant doubling dimension as shown in [31] and [24] in a centralized setting. Ours uses distributed construction.

1.2 Applications

The results can be applied in maintaining network overlay topologies with good properties for P2P file sharing applications [36]. Such P2P overlay networks are often constructed in a distributed manner without centralized control, to achieve robustness, reliability and scalability. One important issue is reducing routing delay by making the overlay topology aware of the underlying network topology [13, 43, 34, 49, 50]. But all these existing algorithms are heuristics without any guarantee. A spanner graph would be a good solution for the overlay construction, yet there is no centralized authority in the P2P network that supervises the spanner construction and the peers may join or leave the network frequently. The work in this paper initiates the study of the emergence of good spanners in the setting when there is little coordination between the peers and the users only need a modest amount of incomplete information of the current overlay topology.

It has been shown that the delay on the Internet has geometric growth properties [42, 40]. Thus we can apply our spanner construction for a P2P network under the model that the delay function has constant doubling dimension. We show that the spanner can be constructed under a proper model of the P2P network such that only $O(n \log \alpha)$ messages need to be delivered. The spanner topology is implicitly stored on the nodes with each node's storage cost bounded by $O(\log \alpha)$. With such partial information stored at each node, there is a local distributed algorithm that finds a $(1+\varepsilon)$-stretch path between any two nodes.

1.3 Related work

In the vast amount of prior literature on geometric spanners, there are three main ideas: Θ-graphs, the greedy spanners, and the WSPD-induced spanners [39]. Please refer to the book Geometric Spanner Networks for a nice survey [39]. We will review two spanner construction ideas that are most related to our approach. The first idea is the path-greedy spanner construction [12, 15, 16, 17]. All pairwise edges are ordered with non-decreasing lengths and checked in that order. An edge is included in the spanner if the shortest path in the current graph is longer than t times the Euclidean distance, and is discarded otherwise. Variants of this idea generate spanners with constant degree and total weight $O(|\mathrm{MST}|)$. This idea cannot be applied in our setting as edges constructed in practice may not be in non-decreasing order of their lengths. The second idea is to use the gap property [12] — the sources and sinks of any two edges in an edge set are separated by a distance at least proportional to the length of the shorter of the two edges and their directions differ no more than a given angle. The gap-greedy algorithm [7] considers pairs of points, again in order of non-decreasing distances, and includes an edge in the spanner if and only if it does not violate the gap property. The spanner generated this way has constant degree and total weight $O(|\mathrm{MST}|)$. Compared with our algorithm, our strategy is a relaxation of the gap property in the way that the edges in our spanner may have one of their endpoints arbitrarily close (or at the same points) and we have no restriction on the direction of the edges and the ordering of the edges to be considered. The proof for the gap greedy algorithm requires heavy plane geometry tools and our proof technique only uses packing argument and can be extended to the general metric setting as long as a similar packing argument holds. To get these benefits our algorithm has a slightly worse upper bounds on the spanner weight by a logarithmic factor.

Prior work on compact routing [46, 26, 3, 33, 2] usually implies a $(1+\varepsilon)$-spanner explicitly or implicitly. Again, these spanners are constructed in a coordinated setting.

An extended abstract of this paper was presented in 2008 and also published in [25]. During the review of this journal version, we became aware of the independent work at around the same time by Smid [47], which introduced the 'weak gap property' for spanner construction. The spanner constructed from our algorithm satisfies the weak gap property. Therefore, the properties resulting from weak gap property can be applied here directly. Specifically, it says any directed graph satisfying the weak gap property has $O(n)$ edges and total weight $O(\lg n \cdot |\mathrm{MST}|)$. This corresponds to our conclusion on the linear size of the spanner edges and improves our result on the weight of the constructed spanner from $O(\lg \alpha \cdot |\mathrm{MST}|)$ to $O(\lg n \cdot |\mathrm{MST}|)$. The proof techniques and methodologies in the two proofs are different however.

1.4 Organization

In the rest of the paper we first elaborate the spanner construction in an uncoordinated manner and then show the connection of the spanner with the greedy WSPD. We then show the good properties of both the greedy WSPD and our spanner. At the end, we describe how to apply the spanner in the P2P setting to support low-storage spanner representation and efficient local low-stretch routing.

2 Spanner construction under anarchism

Given n points in $\mathbb{R}^d$, each point represents an agent. As explained in the introduction, we consider the following algorithm for constructing a sparse spanner with stretch factor s in an uncoordinated way. For any point p, denote by $B_r(p)$ the collection of points that are within distance r from point p, i.e., inside the ball with radius r centered at p.

Spanner construction algorithm. Each point/agent p checks to see whether an edge from itself to another point q should be constructed or not. At this point there might be some edges already constructed by other agents. The order of which agent checks on which edge is completely *arbitrary*. Specifically, p performs the following operations:

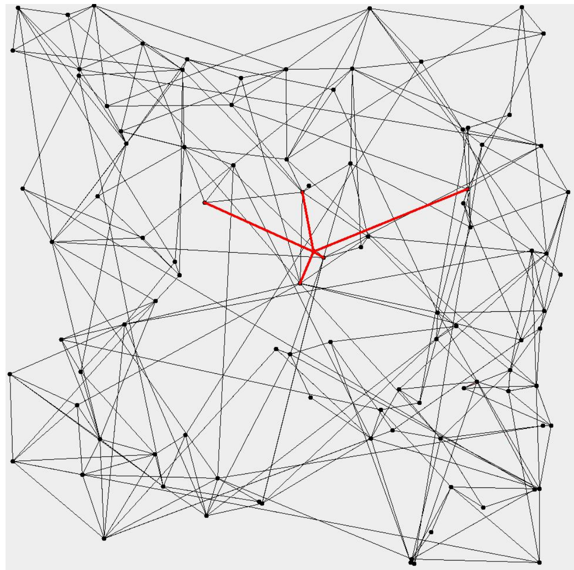

Figure 1. A greedy spanner example for 100 points with aspect ratio $\alpha = 223$, the average degree is 6.5, and the stretch is 3.4.

Check where there is already an edge $p'q'$ *such that* p *and* q *are within distance* $\frac{|p'q'|}{2(s+1)}$ *from* p', q' *respectively. If so,* p *does not build the edge to* q*. Otherwise,* p *will build an edge to* q*.*

This incremental construction of edges is executed by different agents in a completely uncoordinated manner. We assume that no two agents perform the above strategy at exactly the same time. Thus when any agent conducts the above process, the decision is based on the current network already constructed. The algorithm terminates when all agents finish checking the edges from themselves to all other points. In this paper we first study the properties of the constructed graph G by these uncoordinated behaviors. We will discuss later in Section 5 a proper complexity model for the uncoordinated construction in a distributed environment and also bound the computing cost of this spanner. An spanner example generated from the above algorithm is shown in Figure 1. Throughout the construction the following invariant is maintained by the graph G.

Lemma 2.1.

1. *For any edge* pq *that is not in* G*, there is another edge* $p'q'$ *in* G *such that* $|pp'| \leq |p'q'|/(2s+2)$, $|qq'| \leq |p'q'|/(2s+2)$.
2. *For any two edges* pq, $p'q'$ *in the constructed graph* G*, suppose that* pq *is built before* $p'q'$*, then one of the following is true:* $|pp'| > |pq|/(2s+2)$ *or* $|qq'| > |pq|/(2s+2)$.

To show that the algorithm eventually outputs a good spanner, we first show the connection of G with the notion *well-separated pair decomposition.*

Definition 2.2 (Well-separated pair). *Let* $s > 0$ *be a constant, and a pair of sets of points* A, B *are well-separate with respect to* s *(or* s*-separated), if* $d(A,B) \geq s \cdot \max(diam(A), diam(B))$*, where* $diam(A)$ *is the diameter of the point set* A*, and* $d(A,B) = \min_{p \in A, q \in B} |pq|$.

Definition 2.3 (Well-separated pair decomposition). *Let $t > 0$ be a constant,and P be a point set. A well-separated pair decomposition (WSPD) with respect to t of P is a set of pairs $\mathcal{W} = \{\{A_1, B_1\}, \ldots, \{A_m, B_m\}\}$, such that*

1. *$A_i, B_i \subseteq P$, and the pair sets A_i and B_i are s-separated for every i.*
2. *For any pair of points $p, q \in P$, there is at least one pair (A_i, B_i) such that $p \in A_i$ and $q \in B_i$.*

Here m is called the size of the WSPD.

In our construction of G, it is not so hard to see that a well-separated pair decomposition is actually implied.

Theorem 2.4. *From the uncoordinated construction of the graph G, we can build the following s-WSPD $\mathcal{W}$: for each edge pq in G, include in the well-separated pair decomposition the pair $(B_r(p), B_r(q))$, with $r = |pq|/(2s+2)$. The size of the WSPD is the number of edges in G.*

Proof: First each pair $(B_r(p), B_r(q))$ is an s-well-separated pair. Obviously, $d(B_r(p), B_r(q)) \geq |pq| - 2r$, and $\mathrm{diam}(B_r(p)), \mathrm{diam}(B_r(q)) \leq 2r$. One can then verify that $d(B_r(p), B_r(q)) \geq s \cdot \max(\mathrm{diam}(B_r(p)), \mathrm{diam}(B_r(q)))$.

We now show that any point p, q is included in one well-separated pair. If the edge pq is in the graph the claim is true obviously. Otherwise, there is an edge $p'q'$ in G such that $|pp'| \leq |p'q'|/(2s+2)$, $|qq'| \leq |p'q'|/(2s+2)$, by Lemma 2.1. This means that $p \in B_{r'}(p')$ and $q \in B_{r'}(q')$ with $r' = |p'q'|/(2s+2)$. This finishes the proof. □

Algorithm for well-separated pair decomposition. The above theorem shows the connection of the uncoordinated graph G with a WSPD $\mathcal{W}$. In fact, the way to compute the WSPD $\mathcal{W}$ via the construction of G is equivalent to the following algorithm that computes an s-WSPD, where s is the separation parameter of the well-separated pairs, $s > 1$.

1. Choose an arbitrary pair (p, q), not yet covered by existing well-separated pairs in $\mathcal{W}$.
2. Include the pair of point sets $B_r(p)$ and $B_r(q)$ in the WSPD $\mathcal{W}$, with $r = |pq|/(2+2s)$.
3. Label the point pair (p_i, q_i) with $p_i \in B_r(p)$ and $B_r(q)$ as being covered.
4. Repeat the above steps until every pair of points is covered.

With the s-WSPD $\mathcal{W}$, the uncoordinated construction of the graph G is in fact by taking an edge from each and every well-separated pair in $\mathcal{W}$ — the simple rule in Lemma 2.1 prevents two edges from the same well-separated pair in $\mathcal{W}$ to be constructed. It is already known that for any well-separated pair decomposition, if one edge is taken from each well-separated pair, then the edges will become a spanner on the original point set [6, 5, 38, 35]. For our specific greedy s-WSPD, we are able to get a slightly better stretch, since our well separated pairs are centered on two points and we are picking the edge between these two points as the spanner edge. The previous schemes choose an arbitrary edge in each well separated pair.

Theorem 2.5. *Graph G constructed from the greedy s-WSPD is a spanner with stretch factor $(s+1)/(s-1)$.*

Proof: Denote by $\pi(p,q)$ the shortest path length between p,q in the graph G. We show that $\pi(p,q) \le \beta \cdot |pq|$ for any $p,q \in P$, with $\beta = (s+1)/(s-1)$. We prove this claim by induction on the distance between two points p,q. Take p,q as the closest pair of P. Then any s-WSPD will have to use a singleton pair (p,q) to cover the pair (p,q) if $s \ge 1$. Otherwise, say (P,Q) is an s-well-separated pair that covers (p,q), and $|P| > 1$. Then $\text{diam}(P) > |pq|$, and $d(P,Q) = |pq|$. This contradicts the fact that $d(P,Q) \ge s \cdot \text{diam}(P)$. Therefore the edge pq is included in G for sure and $\pi(p,q) = |pq|$.

Now suppose that for all pairs of nodes x,y with Euclidean distance $|xy| \le \ell$, we have $\pi(x,y) \le \beta \cdot |xy|$. Now we consider the pair of nodes p,q with the smallest distance (among all remaining pairs) that is still greater than ℓ. (p,q) is covered by an s-well-separated pair $(P,Q) \in \mathcal{W}$, where $P = B_r(p')$ and $Q = B_r(q')$ with $r = |p'q'|/(2s+2)$ and $p'q'$ an edge in G. Now we argue that $|pp'| \le \ell$. Otherwise, $|pq| \ge d(P,Q) \ge s \cdot \text{diam}(P) \ge s \cdot |pp'| > |pp'|$. So we should have selected the pair (p,p') instead of (p,q). Similarly, $|qq'| \le \ell$. Thus by induction hypothesis $\pi(p,p') \le \beta \cdot |pp'|$, $\pi(q,q') \le \beta \cdot |qq'|$. By triangle inequality, we have $\pi(p,q) \le \pi(p,p') + |p'q'| + \pi(q,q') \le \beta \cdot (|pp'| + |qq'|) + |p'q'| \le 2\beta \cdot r + |p'q'|$. On the other hand, we know by triangle inequality that $|pq| \ge |p'q'| - 2r = \frac{s}{s+1} \cdot |p'q'|$. Combining everything we get that $\pi(p,q) \le (\frac{\beta}{s+1} + 1) \cdot \frac{s+1}{s} \cdot |pq| = \beta |pq|$, with $\beta = (s+1)/(s-1)$. This finishes the proof. □

The above theorem shows that the uncoordinated construction is indeed a spanner. To make the stretch factor as $1+\varepsilon$, we just take $s = 1 + 2/\varepsilon$ in our spanner construction. We also want to show that the spanner is sparse and has some other nice properties useful for our applications. For that we will first show that the greedy WSPD algorithm will output a linear number of well-separated pairs, in the next section.

3 A greedy algorithm for well-separated pair decomposition

To show that the WSPD by the greedy algorithm has a linear number of pairs, we actually show the connection of this WSPD with a specific WSPD constructed by the deformable spanner [24], in the way that at most a constant number of pairs in $\mathcal{W}$ is mapped to each well-separated pair constructed by the deformable spanner. To be consistent, in the following description, the greedy WSPD is denoted by $\mathcal{W}$ and the WSPD constructed by the deformable spanner is denoted by $\hat{\mathcal{W}}$.

3.1 Deformable spanner and WSPD

In this section, we review the basic definition of the deformable spanner and some related properties, which will be used in our own algorithm analysis in the next subsection.

Given a set of points P in the plane, a set of *discrete centers* with radii r is defined to be the maximal set $S \subseteq P$ that satisfies the *covering* property and the *separation* property:

any point $p \in P$ is within distance r to some point $p' \in S$; and every two points in S are of distance at least r away from each other. In other words, all the points in P can be covered by balls with radii r, whose centers are exactly those points in the discrete center set S. And these balls do not cover other discrete centers.

We now define a hierarchy of discrete centers in an recursive way. S_0 is the original point set P. S_i is the discrete center set of S_{i-1} with radii 2^i. Without loss of generality we assume that the closest pair has distance 1 (as we can scale the point set and do not change the combinatorial structure of the discrete center hierarchy). Thus the number of levels of the discrete center hierarchy is $\log \alpha$, where α is the aspect ratio of the point set P, defined as the ratio of the maximum pairwise distance to the minimum pairwise distance, that is, $\alpha = \max_{u,v \in P} |uv| / \min_{u,v \in P} |uv|$. Since a point p may stay in multiple consecutive levels and correspond to multiple nodes in the discrete center hierarchy, we denote by $p^{(i)}$ the existence of p at level i. For each point $p^{(i-1)} \in S_{i-1}$ on level i, it is within distance 2^i from at least one other point on level $i+1$. Thus we assign to $p^{(i-1)}$ a *parent* $q^{(i)}$ in S_i such that $|p^{(i-1)}q^{(i)}| \leq 2^i$. When there are multiple points in S_i that cover $p^{(i-1)}$, we choose one as its parent arbitrarily. We denote by $P(p^{(i-1)})$ the parent of $p^{(i-1)}$ on level i. We denote by $P^{(i)}(p) = P(P^{(i-1)}(p))$ the *ancestor* of p at level i.

The deformable spanner is based on the hierarchy, with all edges between two points u and v in S_i if $|uv| \leq c \cdot 2^i$, where c is a constant equal to $4 + 16/\varepsilon$.

Now we will restate some important properties of the deformable spanner that will be useful in our algorithm analysis.

Lemma 3.1 (Packing Lemma [24]). *In a point set $S \subseteq R^d$, if every two points are at least distance r away from each other, then there can be at most $(2R/r+1)^d$ points in S within any ball with radii R.*

Lemma 3.2 (Deformable spanner properties [24]). *For a set of n points in R^d with aspect ratio α,*

1. *For any point $p \in S_0$, its ancestor $P^{(i)}(p) \in S_i$ is of distance at most 2^{i+1} away from p.*
2. *Any point $p \in S_i$ has at most $(1+2c)^d - 1$ edges with other points of S_i.*
3. *The deformable spanner $\hat{G}$ is a $(1+\varepsilon)$-spanner G with $O(n/\varepsilon^d)$ edges.*
4. *$\hat{G}$ has total weight $O(|MST| \cdot \lg \alpha/\varepsilon^{d+1})$, where $|MST|$ is the weight of the minimal spanning tree of the point set S.*

As shown in [24], the deformable spanner implies a well-separated pair decomposition $\hat{\mathcal{W}}$ by taking all the 'cousin pairs'. Specifically, for a node $p^{(i)}$ on level i, we denote by P_i the collection of points that are descent of $p^{(i)}$ (including $p^{(i)}$ itself), called the *descendants*. Now we take the pair (P_i, Q_i), the sets of descendants of a *cousin pair* $p^{(i)}$ and $q^{(i)}$, i.e., $p^{(i)}$ and $q^{(i)}$ are *not* neighbors in level i but their parents are neighbors in level $i+1$. This collection of pairs constitutes a $\frac{4}{\varepsilon}$-well-separated pair decomposition. The size of $\hat{\mathcal{W}}$ is bounded by the number of cousin pairs and is shown to be $O(n/\varepsilon^d)$.

3.2 Greedy well-separated pair decomposition has linear size

With the WSPD $\hat{\mathcal{W}}$ constructed by the deformable spanner, we now prove that the greedy WSPD $\mathcal{W}$ has linear size as well. The basic idea is to map the pairs in $\mathcal{W}$ to the pairs in $\hat{\mathcal{W}}$ and show that at most a constant number of pairs in $\mathcal{W}$ map to the same pair in $\hat{\mathcal{W}}$.

Theorem 3.3. *The greedy s-WSPD $\mathcal{W}$ has size $O(ns^d)$.*

Proof: Suppose that we have constructed a deformable spanner DS with $c = 4(s+1)$ and obtained an s-well-separated pair decomposition (WSPD) of it, call it $\hat{\mathcal{W}}$, where $s = c/4-1$. The size of $\hat{\mathcal{W}}$ is $O(ns^d)$. The deformable spanner has stretch $1+4/s$. Now we will construct a map that takes each pair in $\mathcal{W}$ and map it to a pair in $\hat{\mathcal{W}}$.

Each pair $\{P, Q\}$ in $\mathcal{W}$ is created by considering the points inside the balls $B_r(p), B_r(q)$ with radii $r = |pq|/(2+2s)$ around p, q. Now we consider the ancestors of p, q in the spanner DS respectively. There is a unique level i such that the ancestor $u_i = P^{(i)}(p)$ and $v_i = P^{(i)}(q)$ do not have a spanner edge in between but the ancestor $u_{i+1} = P^{(i+1)}(p)$ and $v_{i+1} = P^{(i+1)}(q)$ have an edge in between. The pair u_i, v_i is a cousin pair by definition and thus the decedents of them correspond to an s-well-separated pair in $\hat{\mathcal{W}}$. We say that the pair $(B_r(p), B_r(q)) \in \mathcal{W}$ maps to the descendant pair $(P_i, Q_i) \in \hat{\mathcal{W}}$.

By the discrete center hierarchy (Lemma 3.2), we show that,

$$|pq| \geq |u_iv_i| - |pu_i| - |qv_i| \geq |u_iv_i| - 2 \cdot 2^{i+1} \geq (c-4) \cdot 2^i.$$

The last inequality follows from that fact that u_i, v_i do not have an edge in the spanner and $|u_iv_i| > c \cdot 2^i$. On the other hand,

$$|pq| \leq |pu_{i+1}| + |u_{i+1}v_{i+1}| + |qv_{i+1}| \leq 2 \cdot 2^{i+2} + c \cdot 2^{i+1} = 2(c+4) \cdot 2^i.$$

The last inequality follows from the fact that u_{i+1}, v_{i+1} have an edge in the spanner and $|u_{i+1}v_{i+1}| \leq c \cdot 2^{i+1}$. Similarly, we have

$$c \cdot 2^i < |u_iv_i| \leq |u_iu_{i+1}| + |u_{i+1}v_{i+1}| + |v_iv_{i+1}| \leq 2 \cdot 2^{i+1} + c \cdot 2^{i+1} = 2(c+2) \cdot 2^i.$$

Therefore the distance between p and q is $c' \cdot |u_iv_i|$, where $(c-4)/(2c+4) \leq c' \leq (2c+8)/c$.

Now suppose two pair $(B_{r_1}(p_1), B_{r_1}(q_1))$, $(B_{r_2}(p_2), B_{r_2}(q_2))$ in $\mathcal{W}$ map to the same pair u_i and v_i by the above process. Without loss of generality suppose that p_1, q_1 are selected before p_2, q_2 in our greedy algorithm. Here is the observation:

1. $|p_1q_1| = c'_1 \cdot |u_iv_i|$, $|p_2q_2| = c'_2 \cdot |u_iv_i|$, $r_1 = |p_1q_1|/(2+2s) = c'_1 \cdot |u_iv_i|/(2+2s)$, $r_2 = c'_2 \cdot |u_iv_i|/(2+2s)$, where $(c-4)/(2c+4) \leq c'_1, c'_2 \leq (2c+8)/c$, and r_1, r_2 are the radii of the balls for the two pairs respectively.
2. The reason that (p_2, q_2) can be selected in our greedy algorithm is that at least one of p_2 or q_2 is outside the balls $B(p_1), B(q_1)$, by Lemma 2.1. This says that at least one of p_2 or q_2 is of distance r_1 away from p_1, q_1.

Now we look at all the pairs (p_ℓ, q_ℓ) that are mapped to the same ancestor pair (u_i, v_i). The pairs are ordered in the same order as they are constructed, i.e., p_1, q_1 is the first pair selected in the greedy WSPD algorithm. Suppose r_{min} is the minimum among all radius r_i. $r_{min} \geq c/(2c+8) \cdot |u_i v_i|/(2+2s) = |u_i v_i|/(4s+8)$. We group these pairs in the following way. The first group H_1 contains (p_1, q_1) and all the pairs (p_ℓ, q_ℓ) that have p_ℓ within distance $r_{min}/2$ from p_1. We say that (p_1, q_1) is the representative pair in H_1 and the other pairs in H_1 are *close* to the pair (p_1, q_1). The second group H_2 contains, among all remaining pairs, the pair that was selected in the greedy algorithm the earliest, and all the pairs that are close to it. We repeat this process to group all the pairs into k groups, $H_1, H_2, \cdots, H_k$. For all the pairs in each group H_j, we have one representative pair, denoted by (p_j, q_j) and the rest of the pairs in this group are close to it.

We first bound the number of pairs belonging to each group by a constant with a packing argument. With our group criteria and the above observations, all p_ℓ in the group H_j are at most r_{min} away from each other. This means that the q_ℓ's must be far away — the q_ℓ's must be at least distance r_{min} away from each other, by Lemma 2.1. On the other hand, all the q_ℓ's are descendant of the node v_i, so $|v_i q_\ell| \leq 2^{i+1}$ by Theorem 3.2. That is, all the q_ℓ's are within a ball of radius 2^{i+1} centered at v_i. By the packing Lemma 3.1, the number of such q_ℓ's is at most $(2 \cdot 2^{i+1}/r_{min} + 1)^d \leq (2 \cdot 2^{i+1}(4s+8)/|u_i v_i| + 1)^d \leq (4(s+2)/(s+1)+1)^d$. This is also the bound on the number of pairs inside each group.

Now we bound the number of different groups, i.e., the value k. For the representative pairs of the k groups, $(p_1, q_1), (p_2, q_2), \cdots, (p_k, q_k)$, all the p_i's must be at least distance $r_{min}/2$ away from each other. Again these p_i's are all descendant of u_i and thus are within distance 2^{i+1} from u_i. By a similar packing argument, the number of such p_i's is bounded by $(4 \cdot 2^{i+1}/r_{min} + 1)^d \leq (8(s+2)/(s+1)+1)^d$. So the total number of pairs mapped to the same ancestor pair in $\hat{\mathcal{W}}$ will be at most $(4(s+2)/(s+1)+1)^d \cdot (8(s+2)/(s+1)+1)^d = (O(1+1/s))^d$. Thus the total number of pairs in W is at most $O(ns^d)$. This finishes the proof. □

4 Size, degree and weight of the uncoordinated spanner

With the result that the greedy WSPD has linear size in the previous section and the connection of the greedy WSPD with the uncoordinated spanner construction in Section 2, we immediately get the following theorem.

Theorem 4.1. *The uncoordinated spanner with parameter s is a spanner with stretch factor $(s+1)/(s-1)$ and has $O(ns^d)$ number of edges.*

Proof: The number of edges in the spanner is the same as the size of the greedy WSPD $\mathcal{W}$ with the same parameter s constructed by selecting the same set of edges in the same order. □

Theorem 4.2. *The uncoordinated spanner has a maximal degree of $O(\lg \alpha \cdot s^d)$ and average degree $O(s^d)$.*

Proof: With the same argument as in Theorem 3.3, each pair (p, q) built in the uncoordinated spanner construction maps to a pair of ancestors $(P^{(i)}(p), P^{(i)}(q))$ in the deformable spanner that is a cousin pair. Consider all the edges of p in G, (p, q_ℓ), that map to the same ancestor pair $(P^{(i)}(p), P^{(i)}(q))$. By a similar argument, all the q_ℓ's must be at least distance r_{min} away from each other (since all these pairs have p as the first element in the pair). Thus we have the number of such edges is bounded by $(4(s+2)/(s+1)+1)^d$. The cousin pairs associated with $P^{(i)}(p)$ is at most 5^d times the number of adjacent edges of $P^{(i+1)}(p)$, and is bounded by $5^d \cdot [(8s+9)^d - 1]$ (by Theorem 3.2). Since there are $\lfloor \lg \alpha \rfloor$ levels, the total number of edges associated with the node p is at most $\lfloor \lg \alpha \rfloor \cdot 5^d \cdot [(8s+9)^d - 1] \cdot (4(s+2)/(s+1)+1)^d$. Then the maximal degree of the spanner is $O(\lg \alpha \cdot s^d)$.

Since the spanner has total $O(ns^d)$ edges, the average degree is $O(s^d)$. □

Theorem 4.3. *The uncoordinated spanner has total weight $O(\lg \alpha \cdot |MST| \cdot s^{d+1})$.*

Proof: Again we use the mapping of the uncoordinated spanner edges to the cousin pairs in the deformable spanner DS, as in Theorem 3.3. We also use the same notation here. Consider all the edges (p_ℓ, q_ℓ) that map to the same ancestor cousin pair (u_i, v_i). We now map them to the edge between the parents of this cousin pairs, i.e., edge $u_{i+1}v_{i+1}$ in DS. The pair (u_{i+1}, v_{i+1}) has at most 5^{2d} number of cousin pairs. Thus at most $(4(s+2)/(s+1)+1)^d \cdot (8(s+2)/(s+1)+1)^d \cdot 5^{2d} = (O(1+1/s))^d$ edges in G are mapped to one edge in DS.

Now we will bound the length of an edge pq in G and the edge $u_{i+1}v_{i+1}$ in DS it maps to. From the proof of Theorem 3.3, we know that $(c-4) \cdot 2^i \leq |pq| \leq 2(c+4) \cdot 2^i$, where $c = 4(s+1)$. In addition, $|u_{i+1}v_{i+1}| \leq 2c \cdot 2^i$ as $u_{i+1}v_{i+1}$ is an edge in DS, and $|u_{i+1}v_{i+1}| \geq |u_i v_i| - |u_{i+1}u_i| - |v_{i+1}v_i| \geq c \cdot 2^i - 2 \cdot 2^{i+1} = (c-4) \cdot 2^i$. Thus, $(c-4)/(2c) \leq |pq|/|u_{i+1}v_{i+1}| \leq 2(c+4)/(c-4)$.

We now bound the total weight of the spanner G. We group all the edges by the spanner edge in DS that they map to. Thus we have the total weight of G is at most $2(c+4)/(c-4) \cdot (O(1+1/s))^d$ the weight of DS. By Lemma 3.2, the weight of DS is at most $O(\lg \alpha \cdot |\text{MST}| \cdot s^{d+1})$. Thus the weight of G is at most $O(\lg \alpha \cdot |\text{MST}| \cdot s^{d+1})$. □

We remark here that independently Smid shows that the weight of the spanner is bounded by $O(\log n \cdot |\text{MST}| \cdot s^{d+1})$ by using the weak gap property in [47].

5 Spanner construction and routing in P2P networks

The uncoordinated spanner construction can be applied for peer-to-peer system design, to allow users to maintain a spanner in a distributed manner. For that, we will first extend our spanner results to a metric with constant doubling dimension. The doubling dimension of a metric space (X, d) is the smallest value γ such that each ball of radius R can be covered by at most 2^γ balls of radius $R/2$ [28].

Theorem 5.1. *For n points and a metric space defined on them with constant doubling dimension γ, the uncoordinated spanner construction outputs a spanner G with stretch* *factor $(s+1)/(s-1)$, has total weight $O(\gamma^2 \cdot \lg \alpha \cdot |MST| \cdot s^{O(\gamma)})$ and has $O(\gamma^2 \cdot n \cdot s^{O(\gamma)})$ number of edges. Also it has a maximal degree of $O(\gamma \cdot \lg \alpha \cdot s^{O(\gamma)})$ and average degree $O(\gamma \cdot s^{O(\gamma)})$.*

Proof: The proof follows almost the same as those in the previous section. The deformable spanner can be applied for metrics with constant doubling dimension [24]. Whenever we use a geometric packing argument, we replace by the property of metrics of constant doubling dimension. □

5.1 Distributed construction.

Now we would like to discuss the model of computing for P2P overlay design as well as the construction cost of the uncoordinated spanner. We assume that there is already a mechanism maintained in the system such that any node x can obtain the distance to any node y in $O(1)$ time. For example, this can be done by a TraceRoute command executed by x to the node y. We also assume that there is a service answering near neighbor queries: given a node p and a distance r, return the neighbours within distance r from p. Such an oracle is often maintained in distributed file sharing systems. Various structured P2P system support such function with low cost [36]. Even in unstructured system such as BitTorrent, the Ono plugin is effective at locating nearby peers, with vanishingly small overheads [1].

The spanner edges are recorded in a distributed fashion so that no node has the entire picture of the spanner topology. After each edge pq is constructed, the peers p, q will inform their neighboring nodes (those in $B_r(p)$ and $B_r(q)$ with $r = |pq|/(2s+2)$) that such an edge pq exists so that they will not try to connect to one another. We assume that these messages are delivered immediately so that when any newly built edge is informed to nodes of relevance. The number of messages for this operation is bounded by $|B_r(p)| + |B_r(q)|$. The amount of storage at each node x is proportional to the number of well-separated pairs that include x. The following theorem bounds the total number of such messages during the execution of the algorithm and the amount of storage at each node.

Theorem 5.2. *For the uncoordinated spanner G and the corresponding greedy WSPD $\mathcal{W} = \{(P_i, Q_i)\}$ with size m, each node x is included in at most $O(s^d \lg \alpha)$ well-separated pairs in $\mathcal{W}$. Thus, $\sum_{i=1}^{m}(|P_i| + |Q_i|) = O(ns^d \cdot \lg \alpha)$.*

Proof: For each point x, each pair set it belongs to will be mapped to some ancestor pair in some level in the corresponding deformable spanner. Let's consider the set pair (P_k, Q_k) that are mapped to level i by selecting the point pair (p, q), where $x \in P_k$. Suppose the ancestor pair is $(p^{(i)}(p), p^{(i)}(q))$. If x is the descendent of $p^{(i)}(p)$, there are only a constant number of such pair sets mapped to $(p^{(i)}(p), p^{(i)}(q))$ according to our previous argument, which means x will be covered only in a constant number of pair sets corresponding to this level. If x is not the descendent of $p^{(i)}(p)$, then x may be mapped to different ancestor pairs $(p^{(i)}(p), p^{(i)}(q))$. The corresponding radius with (p, q) is $r = |pq|/(2+2s) \leq (1+4/c)2^{i+2}$. In this case, $2^{i+1} < |p^{(i)}(p)x| \leq 2^{i+1} + r \leq 2^{i+1} + (1+4/c)2^{i+2} = (1+2/c)2^{i+2}$. According to Lemma 3.2, different $p^{(i)}(p)$ in level i must be at least 2^i away from each other. With packing argument, there can be at most $(((1+2/c)2^{i+1})^d - (2^{i+1})^d)/(2^{i-1})^d = (4+8/c)^d = (2/(s+1)+4)^d - 2^{2d} = O(s^d)$ different such $p^{(i)}(p)$. So x can only be covered in a constant number of different pairs in level i which are not mapped to x's ancestor. Combining the above argument, there are only $O(s^d)$ pair sets that cover x in each level i. And there are only $\lg \alpha$ different levels. So each node x is included in at most $O(s^d \lg \alpha)$ well-separated pairs in $\mathcal{W}$.

As there are only linear number of pairs, the last statement follows immediately. □

5.2 Distributed routing.

Although the spanner topology is implicitly stored on the nodes with each node only knows some piece of it, we are actually able to do a distributed and local routing on the spanner with only information available at the nodes such that the path discovered has maximum stretch $(s+1)/(s-1)$. This can be useful when we would like to operate on the overlay topology only in a P2P application for security concerns.

In particular, for any node p who has a message to send to node q, it is guaranteed that (p,q) is covered by a well-separated pair $(B_r(p'), B_r(q'))$ with $p \in B_r(p')$ and $q \in B_r(q')$. By the construction algorithm, the edge $p'q'$, after constructed, is informed to all nodes in $B_r(p') \cup B_r(q')$, including p. Thus p includes in the packet a partial route with $\{p \rightsquigarrow p', p'q', q' \rightsquigarrow q\}$. The notation $p \rightsquigarrow p'$ means that p will need to first find out the low-stretch path from p to the node p' (inductively), from where the edge $p'q'$ can be taken, such that with another low-stretch path to be found out from q' to q, the message can be delivered to q. This way of routing with partial routing information stored with the packet is similar to the idea of source routing [48] except that we do not include the full routing path at the source node. By the same induction as used in the proof of spanner stretch (Theorem 2.5), the final path is going to have stretch at most $(s+1)/(s-1)$. The number of hops is bounded by $d^{1/(1+\lg s)}$.

Theorem 5.3. *For any two points p and q, $d = |pq|$, the distributed routing scheme gives a path with at most $h(d) = d^{1/(1+\lg s)}$ hops between p and q.*

Proof: For any pair (p,q), it will be covered by a pair set (P,Q), which is created by a pair (p',q'). Then there is an edge between p' and q'. The path $p \rightsquigarrow q$ between p and q can be found in this way: $p \rightsquigarrow p', p'q', q' \rightsquigarrow q$. Obviously $|pp'| \le r \le |pq|/(2s)$. We can get the recurrence $h(|pq|) = h(|pp'|) + 1 + h(|qq'|) \le h(|pq|/(2s)) + 1 + h(|pq|/(2s))$, that is, $h(d) = 2h(d/(2s)) + 1$. Solve this recurrence we get $h(d) = 2^{\lg_{2s} d + 1} = 2^{\lg d/\lg(2s)+1} = 2d^{1/(1+\lg s)}$. □

5.3 Nearest neighbor search.

By maintaining the spanner each node actually keeps its nearest neighbor automatically. Recall that each point x keeps all the pairs (p,q) that create a 'dumb-bell' pair set covering

x. Then we claim, among all these p, one of them must be the nearest neighbor of x. Otherwise, suppose y is the nearest neighbor of x, and y is not one of p. But in the WSPD, (x, y) will belong to one of the pair set (P_i, Q_i), which correspond to a pair (p', q'). Then there is a contradiction, as $|xp'| < |xy|$ implies that y is not the nearest neighbour of x. Thus one's nearest neighbor is locally stored at this node already. According to Theorem 5.2, x will belong to at most $O(s^d \lg \alpha)$ different pair sets. So the nearest neighbor is already locally stored and by searching in worst case $O(s^d \lg \alpha)$ time one can find the nearest neighbor without any communication.

6 Conclusion and Future Work

This paper aims to explain the emergence of good spanners from the behaviors of agents with their own interests. The results can be immediately applied to the construction of good network overlays by distributed peers with incomplete information. For our future work we would like to explore incentive-based overlay construction [21]. One problem faced in the current P2P system design is to reward peers that contribute to the network maintenance or service quality and punish the peers that try to take free rides [23, 29, 30, 22, 44, 45]. We would like to extend the results in this paper and come up with a spanner construction with different quality of service for different peers to achieve fairness — those who build more edges should have a smaller stretch to all other nodes and those who do not build many edges are punished accordingly by making the distances to others slightly longer.

References

[1] http://www.aqualab.cs.northwestern.edu/projects/Ono.html.

[2] I. Abraham, C. Gavoille, A. V. Goldberg, and D. Malkhi. Routing in networks with low doubling dimension. In *Proc. of the 26th International Conference on Distributed Computing Systems (ICDCS)*, July 2006.

[3] I. Abraham and D. Malkhi. Compact routing on euclidian metrics. In *PODC '04: Proceedings of the twenty-third annual ACM symposium on Principles of distributed computing*, pages 141–149, New York, NY, USA, 2004. ACM.

[4] S. Albers, S. Eilts, E. Even-Dar, Y. Mansour, and L. Roditty. On nash equilibria for a network creation game. In *SODA '06: Proceedings of the seventeenth annual ACM-SIAM symposium on Discrete algorithm*, pages 89–98, New York, NY, USA, 2006. ACM.

[5] S. Arya, G. Das, D. M. Mount, J. S. Salowe, and M. Smid. Euclidean spanners: short, thin, and lanky. In *Proc. 27th ACM Symposium on Theory Computing*, pages 489–498, 1995.

[6] S. Arya, D. M. Mount, and M. Smid. Randomized and deterministic algorithms for geometric spanners of small diameter. In *Proc. 35th IEEE Symposium on Foundations of Computer Science*, pages 703–712, 1994.

[7] S. Arya and M. Smid. Efficient construction of a bounded-degree spanner with low weight. *Algorithmica*, 17:33–54, 1997.

[8] Callahan and Kosaraju. Faster algorithms for some geometric graph problems in higher dimensions. In *Proc. 4th ACM-SIAM Symposium on Discrete Algorithms*, pages 291–300, 1993.

[9] P. B. Callahan. Optimal parallel all-nearest-neighbors using the well-separated pair decomposition. In *Proc. 34th IEEE Symposium on Foundations of Computer Science*, pages 332–340, 1993.

[10] P. B. Callahan and S. R. Kosaraju. Algorithms for dynamic closest-pair and n-body potential fields. In *Proc. 6th ACM-SIAM Symposium on Discrete Algorithms*, pages 263–272, 1995.

[11] P. B. Callahan and S. R. Kosaraju. A decomposition of multidimensional point sets with applications to k-nearest-neighbors and n-body potential fields. *J. ACM*, 42:67–90, 1995.

[12] B. Chandra, G. Das, G. Narasimhan, and J. Soares. New sparseness results on graph spanners. *Internat. J. Comput. Geom. Appl.*, 5:125–144, 1995.

[13] Y. Chu, S. Rao, S. Seshan, and H. Zhang. Enabling conferencing applications on the internet using an overlay muilticast architecture. *SIGCOMM Comput. Commun. Rev.*, 31(4):55–67, 2001.

[14] J. Corbo and D. Parkes. The price of selfish behavior in bilateral network formation. In *PODC '05: Proceedings of the twenty-fourth annual ACM symposium on Principles of distributed computing*, pages 99–107, New York, NY, USA, 2005. ACM.

[15] G. Das, P. Heffernan, and G. Narasimhan. Optimally sparse spanners in 3-dimensional Euclidean space. In *Proc. 9th Annu. ACM Sympos. Comput. Geom.*, pages 53–62, 1993.

[16] G. Das and G. Narasimhan. A fast algorithm for constructing sparse Euclidean spanners. *Internat. J. Comput. Geom. Appl.*, 7:297–315, 1997.

[17] G. Das, G. Narasimhan, and J. Salowe. A new way to weigh malnourished Euclidean graphs. In *Proc. 6th ACM-SIAM Sympos. Discrete Algorithms*, pages 215–222, 1995.

[18] D. Eppstein. Spanning trees and spanners. In J.-R. Sack and J. Urrutia, editors, *Handbook of Computational Geometry*, pages 425–461. Elsevier Science Publishers B.V. North-Holland, Amsterdam, 2000.

[19] J. Erickson. Dense point sets have sparse Delaunay triangulations. In *Proc. 13th ACM-SIAM Symposium on Discrete Algorithms*, pages 125–134, 2002.

[20] A. Fabrikant, A. Luthra, E. Maneva, C. H. Papadimitriou, and S. Shenker. On a network creation game. In *PODC '03: Proceedings of the twenty-second annual symposium on Principles of distributed computing*, pages 347–351, 2003.

[21] J. Feigenbaum and S. Shenker. Distributed algorithmic mechanism design: recent results and future directions. In *DIALM '02: Proceedings of the 6th international workshop on Discrete algorithms and methods for mobile computing and communications*, pages 1–13, New York, NY, USA, 2002. ACM.

[22] M. Feldman and J. Chuang. Overcoming free-riding behavior in peer-to-peer systems. *SIGecom Exch.*, 5(4):41–50, 2005.

[23] M. Feldman, K. Lai, I. Stoica, and J. Chuang. Robust incentive techniques for peer-to-peer networks. In *EC '04: Proceedings of the 5th ACM conference on Electronic commerce*, pages 102–111, New York, NY, USA, 2004. ACM.

[24] J. Gao, L. Guibas, and A. Nguyen. Deformable spanners and their applications. *Computational Geometry: Theory and Applications*, 35(1-2):2–19, 2006.

[25] J. Gao and D. Zhou. The emergence of sparse spanners and greedy well-separated pair decomposition. In *Proc. of the the 12th Scandinavian Symposium and Workshops on Algorithm Theory (SWAT'10)*, pages 50–61, June 2010.

[26] L.-A. Gottlieb and L. Roditty. Improved algorithms for fully dynamic geometric spanners and geometric routing. In *SODA '08: Proceedings of the nineteenth annual ACM-SIAM symposium on Discrete algorithms*, pages 591–600, Philadelphia, PA, USA, 2008. Society for Industrial and Applied Mathematics.

[27] J. Gudmundsson, C. Levcopoulos, G. Narasimhan, and M. Smid. Approximate distance oracles for geometric graphs. In *Proc. 13th ACM-SIAM Symposium on Discrete Algorithms*, pages 828–837, 2002.

[28] A. Gupta, R. Krauthgamer, and J. R. Lee. Bounded geometries, fractals, and low-distortion embeddings. In *FOCS '03: Proceedings of the 44th Annual IEEE Symposium on Foundations of Computer Science*, pages 534–543, 2003.

[29] A. Habib and J. Chuang. Incentive mechanism for peer-to-peer media streaming. In *Proc. of the 12th IEEE International Workshop on Quality of Service (IWQoS'04)*, June 2004.

[30] A. Habib and J. Chuang. Service differentiated peer selection: An incentive mechanism for peer-to-peer media streaming. *IEEE Transactions on Multimedia*, 8(3):610–621, June 2006.

[31] S. Har-Peled and M. Mendel. Fast construction of nets in low dimensional metrics, and their applications. *SIAM J. Comput.*, 35(5):1148–1184, 2006.

[32] T. Jansen and M. Theile. Stability in the self-organized evolution of networks. In *GECCO '07: Proceedings of the 9th annual conference on Genetic and evolutionary computation*, pages 931–938, New York, NY, USA, 2007. ACM.

[33] G. Konjevod, A. W. Richa, and D. Xia. Optimal-stretch name-independent compact routing in doubling metrics. In *PODC '06: Proceedings of the twenty-fifth annual ACM symposium on Principles of distributed computing*, pages 198–207, 2006.

[34] M. Kwon and S. Fahmy. Topology-aware overlay networks for group communication. In *NOSSDAV '02: Proceedings of the 12th international workshop on Network and operating systems support for digital audio and video*, pages 127–136, New York, NY, USA, 2002. ACM.

[35] C. Levcopoulos, G. Narasimhan, and M. H. M. Smid. Improved algorithms for constructing fault-tolerant spanners. *Algorithmica*, 32(1):144–156, 2002.

[36] K. Lua, J. Crowcroft, M. Pias, R. Sharma, and S. Lim. A survey and comparison of peer-to-peer overlay network schemes. *Communications Surveys & Tutorials, IEEE*, pages 72–93, 2005.

[37] T. Moscibroda, S. Schmid, and R. Wattenhofer. On the topologies formed by selfish peers. In *PODC '06: Proceedings of the twenty-fifth annual ACM symposium on Principles of distributed computing*, pages 133–142, New York, NY, USA, 2006. ACM.

[38] G. Narasimhan and M. Smid. Approximating the stretch factor of Euclidean graphs. *SIAM J. Comput.*, 30:978–989, 2000.

[39] G. Narasimhan and M. Smid. *Geometric Spanner Networks*. Cambridge University Press, 2007.

[40] E. Ng and H. Zhang. Predicting Internet network distance with coordinates-based approaches. In *Proc. IEEE INFOCOM*, pages 170–179, 2002.

[41] D. Peleg. *Distributed Computing: A Locality Sensitive Approach*. Monographs on Discrete Mathematics and Applications. SIAM, 2000.

[42] C. G. Plaxton, R. Rajaraman, and A. W. Richa. Accessing nearby copies of replicated objects in a distributed environment. In *Proc. ACM Symposium on Parallel Algorithms and Architectures*, pages 311–320, 1997.

[43] S. Ratnasamy, M. Handley, R. Karp, and S. Shenker. Topologically-aware overlay construction and server selection. In *Proceedings of the 21th Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM'05)*, volume 3, pages 1190–1199, 2002.

[44] S. Schosser, K. Böhm, R. Schmidt, and B. Vogt. Incentives engineering for structured p2p systems - a feasibility demonstration using economic experiments. In *EC '06: Proceedings of the 7th ACM conference on Electronic commerce*, pages 280–289, New York, NY, USA, 2006. ACM.

[45] S. Schosser, K. Böhm, and B. Vogt. Indirect partner interaction in peer-to-peer networks: stimulating cooperation by means of structure. In *EC '07: Proceedings of the 8th ACM conference on Electronic commerce*, pages 124–133, New York, NY, USA, 2007. ACM.

[46] A. Slivkins. Distance estimation and object location via rings of neighbors. In *PODC '05: Proceedings of the twenty-fourth annual ACM symposium on Principles of distributed computing*, pages 41–50, 2005.

[47] M. Smid. *The Weak Gap Property in Metric Spaces of Bounded Doubling Dimension*, pages 275–289. Efficient Algorithms, Springer-Verlag, Berlin, Heidelberg, 2009.

[48] A. S. Tanenbaum. *Computer networks (3rd ed.)*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 1996.

[49] W. Wang, C. Jin, and S. Jamin. Network overlay construction under limited end-to-end reachability. In *Proceedings of the 24th Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM'05)*, volume 3, pages 2124–2134, March 2005.

[50] X. Zhang, Z. Li, and Y. Wang. A distributed topology-aware overlays construction algorithm. In *MG '08: Proceedings of the 15th ACM Mardi Gras conference*, pages 1–6, New York, NY, USA, 2008. ACM.