Well-Separated Pair Decomposition for the Unit-Disk Graph Metric and its Applications

Jie Gao[*] Li Zhang[†]

ABSTRACT

We extend the classic notion of well-separated pair decomposition [10] to the (weighted) unit-disk graph metric: the shortest path distance metric induced by the intersection graph of unit disks. We show that for the unit-disk graph metric of n points in the plane and for any constant $c \geq 1$, there exists a c-well-separated pair decomposition with $O(n \log n)$ pairs, and the decomposition can be computed in $O(n \log n)$ time. We also show that for the unit-ball graph metric in k dimensions where $k \geq 3$, there exists a c-well-separated pair decomposition with $O(n^{2-2/k})$ pairs, and the bound is tight in the worst case. We present the application of the well-separated pair decomposition in obtaining efficient algorithms for approximating the diameter, closest pair, nearest neighbor, center, median, and stretch factor, all under the unit-disk graph metric.

Categories and Subject Descriptors

E.1 [**Data**]: data structures—*graphs and networks*; F.2.2 [**Theory of Computation**]: analysis of algorithms and problem complexity—*non-numerical algorithms and problems*; G.2.2 [**Discrete Mathematics**]: graph theory—*graph algorithms*

General Terms

Algorithms

Keywords

well separated pair decomposition, unit-disk graph, approximation algorithm

[*]Department of Computer Science, Stanford University, Stanford, CA 94305. E-mail: jgao@cs.stanford.edu.

[†]Systems Research Center, Hewlett-Packard Labs, 1501 PageMill Road, Palo Alto, CA 94304. E-mail: l.zhang@hp.com.

STOC'03, June 9–11, 2003, San Diego, California, USA.

1. INTRODUCTION

Well-separated pair decomposition, introduced by Callahan and Kosaraju [10], has found numerous applications in solving proximity problems for points in the Euclidean space [7, 10, 9, 5, 4, 28, 24, 18, 14]. A pair of point sets (A, B) is *c-well-separated* if the distance between A, B is at least c times the diameters of both A and B. A well-separated pair decomposition of a point set consists of a set of well-separated pairs that "cover" all the pairs of distinct points, i.e. any two distinct points belong to the different sets of some pair. In [10], Callahan and Kosaraju showed that for any point set in an Euclidean space and for any constant $c \geq 1$, there always exists a c-well-separated pair decomposition with linearly many pairs. This fact has been very useful in obtaining nearly linear time algorithms for many problems such as computing k-nearest neighbors, N-body potential fields, geometric spanners, approximate minimum spanning trees etc. Well-separated pair decomposition is also shown very useful in obtaining efficient dynamic, parallel, and external memory algorithms [7, 9, 10, 8, 17].

The definition of well-separated pair decomposition can be naturally extended to any metric space. Curiously enough however, there has been no work for such an extension. One reason is probably that a general metric space may not admit a well-separated pair decomposition with a sub-quadratic size. Indeed, even for the metric induced by a star tree with unit weight on each edge[1], any well-separated pair decomposition requires quadratically many pairs. This makes the well-separated pair decomposition useless for such a metric. In this paper, we will show that for a certain metric, there do exist well-separated pair decompositions with almost linear size, and therefore many proximity problems under that metric can be solved efficiently. The metric we investigate is the so called unit-disk graph metric.

For a point set S in the plane, its unit-disk graph [12] is formed by connecting two points p, q in S if the Euclidean distance $\mathrm{d}(p, q)$ is at most 1. A unit-disk graph can also be viewed as the intersection graph of a set of unit disks centered at the points in S. We consider the weighted unit-disk graphs where each edge (p, q) receives the weight $\mathrm{d}(p, q)$. Such graphs have been used extensively to model the communication or influence between objects [26, 20] and studied in many different contexts [12, 6, 21, 15]. Recently, there have been many applications of unit-disk graphs in wireless networks [22, 29, 30], as two wireless nodes can directly communicate with each other only if they are within certain dis-

[1]A metric induced by a graph (with positive edge weights) is the shortest path distance metric of the graph.

tance. In this paper, we call the well-separated pair decomposition in the Euclidean space the *geometric well separated pair decomposition*, to be distinguished from the decomposition in graph metrics.

In this paper, we show that for the metric induced by the unit-disk graph on n points and for any constant $c \geq 1$, there does exist a c-well-separated pair decomposition with $O(n \log n)$ pairs, and such a decomposition can be computed in $O(n \log n)$ time. We also show that the bounds can be extended to higher dimensions: for $k \geq 3$, there always exists a c-well-separated pair decomposition with size $O(n^{2-2/k})$ for the unit-ball graph metric on n points, and the bound is tight in the worst case. The construction time is $O(n^{4/3} \operatorname{polylog} n)$ for $k = 3$ and $O(n^{2-2/k})$ for $k \geq 4$.

The difficulty in obtaining a well-separated pair decomposition for unit-disk graph metric is that two points that are close in the space are not necessarily close under the graph metric. We first prove the bound for the point set with constant-bounded density, i.e. a point set where any unit disk covers only a constant number of points in the set, by using a packing argument similar to the one in [19]. For a point set with unbounded density, we apply the clustering technique similar to the one used in [16] to the point set and obtain a set of "clusterheads" with a bounded density. We then apply the result for bounded density point set on those clusterheads. Then, by combining the well-separated pair decomposition for the bounded density point sets and for the Euclidean metric, we are able to show that the bound holds for any point sets.

For a pair of well-separated sets, the distance between two points from different sets can be approximated by the "distance" between the two sets or the distance between any pair of points in different sets. In other words, a well-separated pair decomposition can be thought as a compressed representation to approximate the $\Theta(n^2)$ pairwise distances. Many problems that require to check the pairwise shortest distances can therefore be approximately solved by examining those distances between the well-separated pairs of sets. When the size of the well-separated pair decomposition is sub-quadratic, it often gives us more efficient algorithms than examining all the pairwise distances. Indeed, this is the intuition behind many applications of the geometric well-separated pair decomposition. By using the same intuition, we show the application of well-separated pair decomposition in several proximity problems under the unit-disk graph metric.

Specifically, we consider the following natural proximity problems. Assume that $S_1 \subseteq S$.

- **Furthest neighbor, diameter, center.** The furthest neighbor of $p \in S_1$ is the point in S_1 that maximizes the distance to p. Related problems include computing the *diameter*, the maximum pairwise shortest distance for points in S_1, and the *center*, the point that minimizes the maximum distance to all the other points.

- **Nearest neighbor, closest pair.** The nearest neighbor of $p \in S_1$ is the point in S_1 with the minimum distance to p. Related problems include computing the *closest pair*, the pair with minimum shortest distance, and the *bichromatic closest pair*, the pair that minimizes distance between points from two different sets.

- **Median.** The median of S is the point in S that minimizes the average (or total) distance to all the other points.

- **Stretch factor.** For a graph G defined on S, its stretch factor with respect to the unit-disk graph metric is defined to be the maximum ratio $\pi_G(p,q)/\pi(p,q)$ where π_G, π are the distances induced by G and by the unit-disk graph, respectively.

All the above problems can be solved or approximated efficiently for points in the Euclidean space. However, for the metric induced by a graph, even for planar graphs, very little is known other than solving the expensive all-pairs shortest path problem. For computing diameter, there is a simple linear time method that achieves a 2-approximation and a 4/3-approximate algorithm with running time $O(m\sqrt{n \log n} + n^2 \log n)$, for a graph with n vertices and m edges, by Aingworth *et al.* [2]. By using the powerful tool of the well-separated pair decomposition, we are able to obtain, for all the above problems, nearly linear time algorithms for computing 2.42-approximation[2] and $O(n\sqrt{n \log n}/\varepsilon^3)$ time algorithms for computing $(1+\varepsilon)$-approximation for any $\varepsilon > 0$. In addition, the well-separated pair decomposition can be used to obtain an $O(n \log n/\varepsilon^4)$ space distance oracle so that any $(1+\varepsilon)$ distance query in the unit-disk graph can be answered in $O(1)$ time.

While the existence of almost linear size well-separated pair decomposition has reduced the number of pairs needed to examine when solving many proximity problems, we still need good approximation of the distances between those pairs. Our construction algorithm only produces well separated pair decompositions without knowing an accurate approximation of the distances. For approximation algorithms, we need accurate estimation of shortest distances between $O(n \log n)$ pairs of points in the unit-disk graph. Indeed, the approximation ratio and the running time of our algorithms are dominated by the efficiency of such algorithms. Once the distance estimation has been made, the rest of computation only takes almost linear time.

For a general graph, it is unknown whether $O(n \log n)$ pairs shortest path distances can be computed significantly faster than all pairs shortest path distances. For the planar graph, one can compute $O(n \log n)$ pairs shortest path distance in $O(n\sqrt{n \log n})$ time by using separators with $O(\sqrt{n})$ size [3]. This method extends to the unit-disk graph with constant bounded density since such graphs enjoy similar separator property as the planar graphs [27, 31]. As for approximation, Thorup [33] recently discovered an algorithm for planar graphs that can answer any $(1+\varepsilon)$-shortest distance query in $O(1/\varepsilon)$ time after almost linear time preprocessing. Unfortunately, Thorup's algorithm uses balanced shortest-path separators in planar graphs which do not obviously extend to the unit-disk graphs. On the other hand, it is known that there does exist planar 2.42-spanner for a unit-disk graph [25]. By applying Thorup's algorithm to that

[2]For a minimization problem, a quantity $\hat{\ell}$ is a c-approximation of ℓ if $\ell \leq \hat{\ell} \leq c\ell$. An object $\hat{O}$ is a c-approximation of O with respect to a cost function f if $f(O) \leq f(\hat{O}) \leq cf(O)$. For a maximization problem, $\hat{\ell}$ is a c-approximation of ℓ if $\ell/c \leq \hat{\ell} \leq \ell$, and $\hat{O}$ is a c-approximation of O if $f(O)/c \leq f(\hat{O}) \leq f(O)$.

planar spanner, we can compute 2.42-approximate shortest path distance for $O(n \log n)$ pairs in almost linear time.

Another application of well-separated pair decomposition is that we are able to obtain an almost linear size data structure to answer $(1+\varepsilon)$-approximate shortest path query in $O(1)$ time. Approximate distance oracles have been studied where the emphasis is often on the size of the oracles (for a survey, see [36]). For general graphs, it has been shown that it is possible to construct a $(2k-1)$-approximate distance oracle with size $O(kn^{1+1/k})$ [34]. It is also shown in [34] that this bound is tight for some small k's and is conjectured to be tight for all the k's. For planar graphs, Thorup [33] and Klein [23] have shown that there exists $(1+\varepsilon)$-approximate distance oracle by using almost linear space for any $\varepsilon > 0$. As mentioned before, their results do not extend to the unit-disk graph. In addition, the query time of their algorithm is $O(1/\varepsilon)$. Recently, Gudmundsson *et al.* showed that when a geometric graph is an Euclidean spanner, there does exist an almost linear time (and therefore almost linear space) method to construct $(1+\varepsilon)$-approximate and $O(1)$ query time distance oracles [18]. Again, a unit-disk graph is not necessarily an Euclidean spanner with bounded stretch factor, and their technique does not extend.

2. DEFINITIONS

Unit-disk graphs. Denote by $\mathrm{d}(\cdot,\cdot)$ the Euclidean metric. For a set of points S in the plane, the unit-disk graph $I(S) = (S, E)$ is defined to be the weighted graph where an edge $e = (p, q)$ is in the graph if $\mathrm{d}(p, q) \leq 1$, and the weight of e is $\mathrm{d}(p, q)$. Likewise, we can define the unit-ball graph for points in higher dimensions.

Metric space. Suppose that (S, π) is a metric space where S is a set of elements and π the distance function defined on $S \times S$. For any subset $S_1 \subseteq S$, the *diameter* $D_\pi(S_1)$ (or $D(S_1)$ when π is clear from the context) of S is defined to be $\max_{s_1,s_2 \in S_1} \pi(s_1, s_2)$. The *distance* $\pi(S_1, S_2)$ between two sets $S_1, S_2 \subseteq S$ is defined to be $\min_{s_1 \in S_1, s_2 \in S_2} \pi(s_1, s_2)$.

Any connected graph G containing edges with positive weights induces a metric π_G on its nodes where the distance between any two nodes is defined to be the length of the shortest path between them. In this paper, we are interested in the *unit-disk graph metric* $\pi = \pi_{I(S)}$ induced by the unit-disk graph of a set of points S in the plane.

Well-separated pair decomposition. For a metric space (S, π), two non-empty subsets $S_1, S_2 \subseteq S$ are called *c-well-separated* if $\pi(S_1, S_2) \geq c \cdot \max(D_\pi(S_1), D_\pi(S_2))$.

Following the definition in [10], for any two sets A and B, a set of pairs $\mathcal{P} = \{P_1, P_2, \cdots, P_m\}$, where $P_i = (A_i, B_i)$, is called a *pair decomposition* of (A, B) (or of A if $A = B$) if

- for all the i's, $A_i \subseteq A$, and $B_i \subseteq B$;
- $A_i \cap B_i = \emptyset$;
- for any two elements $a \in A$ and $b \in B$, there exists a unique i such that $a \in A_i$, and $b \in B_i$. We call (a, b) is *covered* by the pair (A_i, B_i).

If in addition, every pair in $\mathcal{P}$ is c-well-separated, $\mathcal{P}$ is called a *c-well-separated pair decomposition* (or c-WSPD in short). Clearly, any metric space admits a c-WSPD with quadratic size by using the trivial family that contains all the pairwise elements.

3. WELL-SEPARATED PAIR DECOMPOSITION FOR UNIT-DISK GRAPH METRIC

We start with the point set with constant bounded density. Then, by combining with geometric well-separated pair decomposition, we show the extension of the result to arbitrary point sets. We will focus our discussion on points in the plane, but most results extend to higher dimensions, resulting sub-quadratic size well-separated pair decomposition. We also show that our bounds in $\mathbb{R}^k$ for $k \geq 3$ are tight.

3.1 Point sets with constant bounded density

The density α of a point set S is defined to be the maximum number of points in S covered by a unit disk. S has constant bounded density if its density is $O(1)$. We assume that the unit-disk graph on S is connected; otherwise, we can consider each connected component separately.

To construct a well-separated pair decomposition, we first compute the unit-disk graph $I(S)$ of S and then a spanning tree T of $I(S)$ where the maximum degree of T is 6. This can be done by computing the relative neighborhood graph of S [35] and keeping those edges with length at most 1. Let G be the resulted graph. It is well known that G is connected, and the degree of G is at most 6. We then compute a spanning tree of G. This step takes $O(n \log n)$ time [32]. It is also known that any n-vertex tree with maximum degree $\beta - 1$ can be divided into two parts by removing a single edge so that each subtree contain at least n/β vertices. We now recursively apply the balanced partitioning to obtain a balanced hierarchical decomposition of T. The decomposition can be represented as a rooted binary tree T' where each node $v \in T'$ corresponds to a (connected) subtree $T(v)$ of T. The root of T' corresponds to T, and for a node $v \in T'$, v's two children v_1, v_2 represent the two connected subtrees $T(v_1)$ and $T(v_2)$ obtained by removing an edge from $T(v)$. We denote by $S(v)$ the set of points in the subtree in $T(v)$. For a node $v \in T'$, denote by $P(v)$ the parent node of v in T'. We also use $P(S(u))$ to denote $S(P(u))$. The height of the tree T' is obviously $O(\log n)$.

Now, we describe a procedure to produce a c-WSPD of S. For each node $v \in T'$, we pick an arbitrary point from $S(v)$ as a representative of $S(v)$ and denote it by $\sigma(S(v))$ (or $\sigma(v)$). We place in a queue the pair $(S(r), S(r))$ where r is the root of T'. We run the following process until the queue becomes empty: repeatedly remove the first element $(S(v_1), S(v_2))$ from the queue. There are two cases:

- $\mathrm{d}(\sigma(v_1), \sigma(v_2)) \geq (c+2) \cdot \max(|S(v_1)| - 1, |S(v_2)| - 1)$. In this case, we include the pair to $\mathcal{P}$.

- $\mathrm{d}(\sigma(v_1), \sigma(v_2)) < (c+2) \cdot \max(|S(v_1)| - 1, |S(v_2)| - 1)$. If $|S(v_1)| = |S(v_2)| = 1$, then it must be the case that $S(v_1)$ and $S(v_2)$ contain the same point. In this case, we simply discard the pair. Otherwise, suppose that $|S(v_1)| \geq |S(v_2)|$ and that u_1, u_2 are two children of v_1. We add to the queue two pairs $(S(u_1), S(v_2))$ and $(S(u_2), S(v_2))$.

The above process is very similar to the collision detection algorithm in [19] except that here a pair is produced when they are c-well-separated. We now make the following claims.

Lemma 3.1. *$\mathcal{P}$ is a c-WSPD of S. Furthermore, each ordered pair of distinct points (p, q) is covered by exactly one pair in $\mathcal{P}$.*

Proof. By the construction, a pair $(S(v_1), S(v_2))$ is included in $\mathcal{P}$ only if $\mathrm{d}(\sigma(v_1), \sigma(v_2)) \geq (c+2) \cdot \max(|S(v_1)| - 1, |S(v_2)| - 1)$. Since for any $v \in T'$, $S(v)$ is connected, $D_\pi(S(v)) \leq |S(v)| - 1$. In addition, $\pi(p, q) \geq \mathrm{d}(p, q)$. Thus, we have that

$$\begin{aligned} & \pi(S(v_1), S(v_2)) \\ \geq\ & \pi(\sigma(v_1), \sigma(v_2)) - (D_\pi(S(v_1)) + D_\pi(S(v_2))) \\ \geq\ & \mathrm{d}(\sigma(v_1), \sigma(v_2)) - 2\max(|S(v_1)| - 1, |S(v_2)| - 1) \\ \geq\ & c \cdot \max(|S(v_1)| - 1, |S(v_2) - 1|) \\ \geq\ & c \cdot \max(D_\pi(S(v_1)), D_\pi(S(v_2)))\,. \end{aligned}$$

That is, every pair in $\mathcal{P}$ is a c-well-separated pair. The process clearly ends. To argue that $\mathcal{P}$ covers all the pairs of distinct points, we observe that we begin with the pair $(S(r), S(r))$ that covers all the pairs, and each time when we split a node, the union of the pairs covered remain the same. The pairs we discard are of the form $(\{p\}, \{p\})$. Thus, all the ordered pairs of distinct points are covered by $\mathcal{P}$. Since the splitting produces two disjoint sets, each ordered pair is covered exactly once. $\square$

The following lemma shows that the sizes of two sets in the same pair do not differ too much.

Lemma 3.2. *Each pair (A, B) that ever appears in the queue satisfies $1/\beta \leq |A|/|B| \leq \beta$.*

Proof. The proof is done by induction. Clearly, it is true for the pair $(S(r), S(r))$. Now, consider the splitting that generates the pair (A, B). Without loss of generality, assume that we split $P(B)$, the parent node of B. By the splitting rule, we have that $|A| \leq |P(B)|$. By induction hypothesis, $|A| \geq |P(B)|/\beta \geq |B|/\beta$. Since the splitting is balanced, $|B| \geq |P(B)|/\beta \geq |A|/\beta$. Therefore $1/\beta \leq |A|/|B| \leq \beta$. $\square$

Now, we bound the size of of $\mathcal{P}$.

Lemma 3.3. *If $(A, B_i) \in \mathcal{P}$, $i = 1, \cdots m(A)$, then $B_i \cap B_j = \emptyset$, and $m(A) = O(c^2|A|)$.*

Proof. By Lemma 3.1, each pair of points can only be covered once, thus $B_i \cap B_j = \emptyset$ if both (A, B_i) and (A, B_j) are in $\mathcal{P}$.

If $(A, B_i) \in \mathcal{P}$, then $(P(A), P(B_i))$ is not in $\mathcal{P}$. So we have that $\mathrm{d}(\sigma(P(A)), \sigma(P(B_i))) < (c+2) \cdot \max(|P(A)| - 1, |P(B_i)| - 1)$. Set $R = \beta|P(A)| \leq \beta^2|A|$. If we split $P(B_i)$ to get the pair (A, B_i), then $(A, P(B_i))$ appeared in the queue, by Lemma 3.2, we have $|P(B_i)| \leq \beta|A| \leq \beta|P(A)| = R$. If we split $P(A)$ to get the pair (A, B_i), then $|B_i| \leq |P(A)|$, so $|P(B_i)| \leq \beta|B_i| \leq \beta|P(A)| = R$. Then,

$$\mathrm{d}(\sigma(P(A)), \sigma(P(B_i))) < (c+2)R, D_\pi(P(B_i)) \leq R\,.$$

Then all the points in B_i must be inside a disk of radius $(c+3)R$ centered at $\sigma(P(A))$. Therefore we have that $|\cup_{i=1}^{m(A)} B_i| = O((c+3)^2R^2)$ because S has constant bounded density. By Lemma 3.2, we know that $|B_i| \geq |A|/\beta \geq |P(A)|/\beta^2$. Thus, $|B_i| \geq R/\beta^3$. Then, we have that $m(A) = O((c+3)^2R^2/(R/\beta^3)) = O(c^2R) = O(c^2|A|)$. $\square$

Lemma 3.4. $|\mathcal{P}| = O(c^2 n \log n)$.

Proof. Define $V_i = \{v \in T' \mid |S(v)| \in [2^i, 2^{i+1})\}$, for $0 \leq i \leq \log n$. Clearly, $|V_i| = O(n/2^i)$. Define $\Sigma_i = \{(S(v), B) \in \mathcal{P} \mid v \in V_i\}$. Denote by $m(S(v))$ the total number of pairs in which $S(v)$ is involved. By Lemma 3.3, we have that

$$\begin{aligned} |\Sigma_i| &= \textstyle\sum_{v \in V_i} m(S(v)) = \sum_{v \in V_i} O(c^2|S(v)|) \\ &= O(c^2 2^{i+1} \cdot n/2^i) = O(c^2 n)\,. \end{aligned}$$

Thus, $|\mathcal{P}| = \sum_{i=0}^{\log n} |\Sigma_i| = O(c^2 n \log n)$. $\square$

Combining the above result, we now have that

Theorem 3.1. *For any n points with constant-bounded density in the plane and any $c \geq 1$, there exists a c-WSPD with $O(c^2 n \log n)$ pairs, which can be computed in $O(c^2 n \log n)$ time.*

Proof. Clearly, the time needed is proportional to the number of pairs that ever appear in the queue. We can represent the construction as a tree: each pair corresponds to a node in the tree, and when a pair is split, we treat those two resulting pairs as the children of the pair. Clearly, the leave of the tree correspond to those pairs included in $\mathcal{P}$ and the pairs discarded. All the discarded pairs have the form $(\{p\}, \{p\})$, and there are $O(n)$ such pairs. Thus, the total number of nodes in the tree is bounded by $O(|\mathcal{P}|) = O(c^2 n \log n)$. Each split costs $O(1)$. Therefore, the total computation cost is $O(c^2 n \log n)$. $\square$

The result can be easily extended to the point set with maximum density α.

Corollary 3.5. *For a point set with maximum density α, for any $c \geq 1$, a c-WSPD with $O(\alpha c^2 n \log n)$ pairs can be constructed in $O(\alpha c^2 n \log n)$ time.*

Proof. If the point set has maximum density α, Lemma 3.3 still holds if we change $m(A)$ to $O(\alpha c^2|A|)$. Plug it in Lemma 3.4, we have that $|\mathcal{P}| = O(\alpha c^2 n \log n)$. The claim then follows from Theorem 3.1. $\square$

By a similar argument, we can extend the result to higher dimensions.

Theorem 3.6. *Given a point set in $\mathbb{R}^k$, where $k \geq 3$, with constant bounded density and any constant $c \geq 1$, there exist a c-WSPD with $O(n^{2-\frac{2}{k}})$ pairs for the unit-ball graph metric. This bound is tight in the worst case. And the decomposition can be computed in $O(n^{2-\frac{2}{k}})$ time.*

Proof. We first compute a spanning tree of S with constant maximum degree β_k, a constant dependent on k only. This can be done by using the technique in [4]. We then follow the same process as we described above. The upper bound follows from the same packing argument as in Lemma 3.4. Lemma 3.3 can be changed so that the number of pairs associated with a node A is $m(A) = O(|A|^{k-1})$. In addition, by Lemma 3.2, for any pair $(A, B) \in P$, $1/\beta_k \leq |A|/|B| \leq \beta_k$. Thus, $m(A) = O(n/|A|)$. Define V_i as in Lemma 3.4, $|V_i| = O(n/2^i)$. When $0 \leq i \leq \frac{1}{k}\log n$, $|\Sigma_i| = \sum_{v \in V_i} m(S(v)) = O(\sum_{v \in V_i} |S(v)|^{k-1}) = O(2^{i(k-1)} \cdot n/2^i) = O(n2^{i(k-2)})$. When $i > \frac{1}{k}\log n$, $|\Sigma_i| = \sum_{v \in V_i} m(S(v)) = O(\sum_{v \in V_i} n/|S(v)|) = O((n/2^i)^2)$. Therefore,

$$\begin{aligned} |\mathcal{P}| &= \sum_{0 \leq i \leq \frac{1}{k}\log n} n2^{i(k-2)} + \sum_{\frac{1}{k}\log n < i \leq \log n} O(n^2/2^{2i}) \\ &= O(n^{2-2/k})\,. \end{aligned}$$

As for the lower bound, consider the points on the k-dimensional grid $[0, n^{1/k}) \times \ldots \times [0, n^{1/k})$. Define a graph G with edges between the pairs of points $(x_1, \ldots, x_i, \ldots, x_k)$ and $(x_1, \ldots, x_i + 1, \ldots, x_k)$ for $i = 1$, or $x_1 = 0$ and $i \geq 2$. A point $(n^{1/k} - 1, x_2, \ldots, x_k)$ for $0 \leq x_i < n^{1/k}$ is called a *tip point*. Intuitively, G can be thought as a graph where the tip points dangle down from a $k-1$ dimensional mesh. Clearly, we can perturb the point set so that its unit-ball graph has the same topology as G. The metric defined by G has the following property: (i) the diameter of G is $kn^{1/k}$; (ii) the distance between any two tip points is at least $2n^{1/k}$. Therefore, when $c > k/2$, a c-WSPD cannot have two tip points in the same set of a pair. Since there are $\Theta(n^{1-1/k})$ tip points, $\Omega(n^{2-\frac{2}{k}})$ pairs are needed, only to separate those tip points.

By the same argument as in Theorem 3.1, it is easily seen that the c-WSPD can be computed in $O(n^{2-2/k})$ time. $\square$

3.2 Arbitrary point sets

The packing argument fails for the unit-disk graph of point sets with unbounded density. However, we can reduce the problem to the constant density case by first clustering the points and then considering those crowded points separately by using geometric well-separated pair decompositions.

For $0 \leq \delta \leq 1$, a point p is δ-covered (or simply covered) by a point s if $\mathrm{d}(s, p) \leq \delta$. Denote by $U(s)$ the set of points δ-covered by s. A subset $X \subseteq S$ is called a δ-cover of S if any point in S is δ-covered by some point in X. We call the points in a δ-cover X *clusterheads*. For each point in S, we assign it to the nearest clusterhead. Thus X induces a partitioning of S into sets $C(s) = S \cap \mathrm{Vor}(s)$, where $\mathrm{Vor}(s)$ denotes the Voronoi cell of s in X. Clearly, for any $p \in C(s)$, $\mathrm{d}(s, p) \leq \delta$, i.e. $C(s) \subseteq U(s)$. A δ-cover is called *minimal* if no two points in X are within distance δ to each other. For any set $A \subseteq X$, denote by $\hat{A}$ the set $\hat{A} = \cup_{s \in A} C(s)$.

To deal with an arbitrary point set S, we first compute a minimal cover X of S with an appropriately chosen δ. We then apply our results on constant-bounded density point sets to X. Note that we can not use the unit-disk graph on X because it may not have the same connectivity as the unit-disk graph on S. For any two points s_1, s_2 in X, they are *neighbors* if $\mathrm{d}(s_1, s_2) > 1$, and there exist two points $p_1 \in C(S_1)$, and $p_2 \in C(s_2)$ such that $\mathrm{d}(p_1, p_2) \leq 1$. We call the pair (p_1, p_2) a *bridge* between s_1 and s_2. For each neighboring pair, we only pick one bridge arbitrarily. Let Y denote the set of all bridge points. Consider the point set $Z = X \cup Y$. Let π' denote the unit-disk graph metric on the set Z. Now, we claim that

Lemma 3.7. *X has $O(1/\delta^2)$-density. Z can be computed in $O(n \log n/\delta^2)$ time.*

PROOF. For any two points s_1, s_2 in a minimal cover X, they are of at least distance δ away from each other. Therefore, there are $O(1/\delta^2)$ points of X inside any unit disk. So X has $O(1/\delta^2)$ density.

To compute X, we can use a greedy algorithm with the assistance of a dynamic point location data structure of unit disks [11]. The algorithm runs in $O(n \log n)$ time. To compute all the neighboring pairs, we can enumerate all the pairs (s_1, s_2) where s_2 is inside the square centered at s_1 and with side-length $2(1 + 2\delta)$. There are $O(n/\delta^2)$ such pairs according to Lemma 3.7 and can be computed in $O(n \log n/\delta^2)$ time by using a standard rectangular range searching data structure. Call such pairs *candidate pairs*. Clearly, only a candidate pair can possibly be a neighboring pair. To find a bridge between two clusterheads s_1, s_2 of a candidate pair, we can compute the bichromatic closest pair between two sets $C(s_1), C(s_2)$. In the plane, this can be done in $O(|U(s_1) \cup U(s_2)| \log n)$ time. Since we only need to examine each clusterhead against $O(1/\delta^2)$ clusterheads, the total computation time is bounded by $O(n \log n/\delta^2)$. $\square$

Now, we show that π' approximates π well on the set X.

Lemma 3.8. *For any two points $p, q \in X$,*

$$\pi(p, q) \leq \pi'(p, q) \leq (1 + 12\delta)\pi(p, q) + 12\delta\,.$$

PROOF. Since $Z \subseteq S$, $\pi(p, q) \leq \pi'(p, q)$. On the other hand, assume that $p_0 p_1 \cdots p_m$, where $p_0 = p$ and $p_m = q$, is the shortest path between p and q in the unit-disk graph of S. For $0 \leq i \leq m$, suppose that s_i is the clusterhead that covers p_i. Note that $s_0 = p$ and $s_m = q$ as $p, q \in X$.

Consider two consecutive points p_i, p_{i+1}. If $s_i = s_{i+1}$, then we have that $\mathrm{d}(p_i, p_{i+1}) \leq 2\delta$. Otherwise, suppose that $s_i \neq s_{i+1}$. If $\mathrm{d}(s_i, s_{i+1}) \leq 1$, then $\pi'(s_i, s_{i+1}) = \mathrm{d}(s_i, s_{i+1}) \leq \mathrm{d}(p_i, p_{i+1}) + 2\delta$. If $\mathrm{d}(s_i, s_{i+1}) > 1$, then s_i, s_{i+1} must be a neighboring pair since $\mathrm{d}(p_i, p_{i+1}) \leq 1$. In this case, it is easy to verify that $\pi'(s_i, s_{i+1}) \leq \mathrm{d}(p_i, p_{i+1}) + 6\delta$. Thus,

$$\begin{aligned} \pi'(p, q) &\leq \textstyle\sum_{i=0}^{m-1} \pi'(s_i, s_{i+1}) \\ &\leq \textstyle\sum_{i=0}^{m-1} \mathrm{d}(p_i, p_{i+1}) + 6m\delta \leq \pi(p, q) + 6m\delta\,. \end{aligned}$$

Since $p_0 p_1 \cdots p_m$ is the shortest path, $\mathrm{d}(p_i, p_{i+2}) \geq 1$ for any $0 \leq i \leq m - 2$ because otherwise the path could be shortened due to triangular inequality. That is, $\pi(p, q) \geq \lfloor m/2 \rfloor > m/2 - 1$, i.e. $m < 2(\pi(p, q) + 1)$. Thus we have that $\pi'(p, q) \leq (1 + 12\delta)\pi(p, q) + 12\delta$. $\square$

Before we describe the construction of c-WSPD for S, we need a straight-forward extension of geometric well-separated pair decomposition in [10] to two separable point sets.

Lemma 3.2. *Suppose that A and B are two point sets that can be separated by a line and have n points in total. For any constant $c \geq 1$, there exists a geometric c-well-separated pair decomposition of (A, B) with $O(n)$ pairs.*

PROOF. This can be done by modifying the algorithm in [10] so that the first split of the point set of $A \cup B$ is by the line that separates A and B. $\square$

Now, we describe a process that produces a c-WSPD of S for any $c \geq 1$. Set $\delta = 1/(2c + 4)$, and $c' = 9(c + 14)$. We first construct a minimal δ-cover X and the set Z as described above. Next we compute a c'-well-separated pair decomposition of the clusterheads X in the unit-disk graph metric of point set Z. Specifically, we give weight 1 to points in X and 0 to bridge points. We find the spanning tree T of the unit-disk graph $I(Z)$. T has total weight $|X|$. We then recursively find balanced weighted decomposition of T: by removing an edge, each subtree has weight at least $1/\beta$ times the weight of the parent. Since X has a bounded density $O(1/\delta^2)$, the packing argument is still valid and we can compute a c'-well-separated pair decomposition for X. Suppose the decomposition obtained is $\mathcal{P} = \{P_1, P_2, \ldots, P_m\}$ where $P_i = (A_i, B_i)$, $A_i \subseteq X$, $B_i \subseteq X$. We now create a set of pairs $\mathcal{P}' = \mathcal{P}'_1 \cup \mathcal{P}'_2 \cup \mathcal{P}'_3$ as follows.

1. For each $P_i \in \mathcal{P}$, if $|A_i| > 1$ or $|B_i| > 1$, we include in $\mathcal{P}'_1$ the pair $P'_i = (\hat{A}_i, \hat{B}_i)$. Recall that $\hat{A} = \cup_{s \in A} C(s)$.
2. If $|A_i| = |B_i| = 1$, suppose that $A_i = \{a\}$ and $B_i = \{b\}$. If $\mathrm{d}(a, b) \geq (2c + 2)\delta$, we then include in $\mathcal{P}'_1$ the pair $P'_i = (\hat{A}_i, \hat{B}_i)$. Otherwise, any pair of points in $\hat{A}_i \bigcup \hat{B}_i$ are within distance $(2c + 2)\delta + 2\delta = 1$. Since $\hat{A}_i \subset \mathrm{Vor}(a)$, and $\hat{B}_i \subset \mathrm{Vor}(b)$, $\hat{A}_i$ and $\hat{B}_i$ are separable by a line. Per Lemma 3.2, we compute a geometric c-WSPD of $(\hat{A}_i, \hat{B}_i)$ and include into $\mathcal{P}'_2$ all the pairs produced this way.
3. For every $s \in X$, we compute a geometric c-WSPD of $C(s)$ and include into $\mathcal{P}'_3$ all the pairs produced.

Now, we claim that

Lemma 3.9. *$\mathcal{P}'$ is a c-WSPD of S.*

PROOF. We first argue that $\mathcal{P}'$ is a pair decomposition of S. For any pair of points $s_1, s_2 \in S$, suppose that the clusterheads covering them are s'_1 and s'_2, respectively. If $s'_1 \neq s'_2$, then (s_1, s_2) is covered by a pair in $\mathcal{P}'_1 \cup \mathcal{P}'_2$. Otherwise, it is covered by a pair in $\mathcal{P}'_3$. It is also easily verified that each ordered pair is covered exactly once.

Now, we show that all the pairs in $\mathcal{P}'$ are c-well-separated with respect to the unit-disk graph metric. Since $\delta = 1/(2c+4)$, for all the pairs in $\mathcal{P}'_2$, the Euclidean distance between any two points in $\hat{A}_i \cup \hat{B}_i$ is at most $(2c+4)\delta = 1$. Therefore, the unit-disk graph on the subset $\hat{A}_i \cup \hat{B}_i$ is a complete graph, i.e. every pair in $\mathcal{P}'_2$ is c-well-separated under the unit-disk graph metric. The same argument applies to $\mathcal{P}'_3$ as the distance between two points in $C(s)$ is at most $2\delta \leq 1$.

Now, consider a pair $(\hat{A}_i, \hat{B}_i) \in \mathcal{P}'_1$. We distinguish two cases.

(1). When $|A_i| = |B_i| = 1$. Then we must have $\pi(A_i, B_i) \geq (2c + 2)\delta$ according to the construction rule, and thus

$$\pi(\hat{A}_i, \hat{B}_i) \geq \pi(A_i, B_i) - 2\delta \geq 2c\delta = c/(c + 2)\,.$$

On the other hand, $D(\hat{A}_i), D(\hat{B}_i) \leq 2\delta = 1/(c + 2)$. Therefore, $(\hat{A}_i, \hat{B}_i)$ is c-well-separated.

(2). When $|A_i| > 1$ or $|B_i| > 1$. In what follows, we use D and D' to denote $D_\pi, D_{\pi'}$, respectively. Clearly,

$$\pi(\hat{A}_i, \hat{B}_i) \geq \pi(A_i, B_i) - 2\delta\,, \text{ and } D(\hat{A}) \leq D(A) + 2\delta\,.$$

Since A_i or B_i contains at lease two clusterheads, it must be true that $\max(D(A_i), D(B_i)) \geq \delta$ as the distance between two clusterheads is at least δ. So, $\max(D(\hat{A}_i), D(\hat{B}_i)) \geq \delta$, and $\max(D(\hat{A}_i), D(\hat{B}_i)) \leq \max(D(A_i), D(B_i)) + 2\delta \leq 3\max(D(A_i), D(B_i))$.

As A_i, B_i are c'-well-separated under π', $\pi'(A_i, B_i) \geq c' \cdot \max(D'(A_i), D'(B_i))$. Therefore,

$$\begin{aligned}
\pi(\hat{A}_i, \hat{B}_i) &\geq \pi(A_i, B_i) - 2\delta \\
&\geq (\pi'(A_i, B_i) - 12\delta)/(1 + 12\delta) - 2\delta \\
&\quad \text{by Lemma 3.8} \\
&\geq c'/(1 + 12\delta) \cdot \max(D'(A_i), D'(B_i)) - 14\delta \\
&\geq c'/(1 + 12\delta) \cdot \max(D(A_i), D(B_i)) - 14\delta \\
&\geq (c'/(3(1 + 12\delta)) - 14) \cdot \max(\hat{D}(A_i), \hat{D}(B_i)) \\
&\geq c \max(D(\hat{A}_i), D(\hat{B}_i))\,. \\
&\quad \text{by } c \geq 1, \delta = 1/(2c + 4), \text{ and } c' = 9(c + 14).
\end{aligned}$$

In both cases, $\hat{A}_i, \hat{B}_i$ are c-well-separated, i.e. all the pairs in $\mathcal{P}'_1$ are c-well-separated. □

Now, we are ready to claim that

Theorem 3.10. *For any set S of n points in the plane and any $c \geq 1$, there exists a c-WSPD $\mathcal{P}$ of S under the unit-disk graph metric where $\mathcal{P}$ contains $O(c^4 n \log n)$ pairs and can be computed in $O(c^4 n \log n)$ time.*

PROOF. By combining Corollary 3.5 and Lemma 3.7, we have that $|\mathcal{P}'_1| \leq |\mathcal{P}| = O(c^2 n \log n/\delta^2) = O(c^4 n \log n)$. If $|A_i| = 1$, then the number of pairs $(A_i, B_i) \in \mathcal{P}'_2$ where $|B_i| = 1$ is bounded by $O(1/\delta^2) = O(c^2)$. Since the size of the geometric well-separated pair decomposition is linear in terms of the number of points [10], $|\mathcal{P}'_2| = O(c^2 n)$. Clearly, $|\mathcal{P}'_3| = O(n)$. Sum them up, we have that $|\mathcal{P}'| = O(c^4 n \log n)$.

By Theorem 3.1 and Lemma 3.7, it is easy to see that the total time needed is $O(c^4 n \log n)$. □

Similarly, in higher dimensions, we have that

Corollary 3.11. *For any set S of n points in $\mathbb{R}^k$, for $k \geq 3$, and for any constant $c \geq 1$, there exist a c-WSPD $\mathcal{P}$ of S under the unit-ball graph metric where $\mathcal{P}$ contains $O(n^{2-2/k})$ pairs and can be constructed in $O(n^{4/3} \operatorname{polylog} n)$ time for $k = 3$ and in $O(n^{2-2/k})$ time for $k \geq 4$.*

PROOF. For simplicity of computation, we use boxes instead of balls to find clusterheads with constant bounded density. A point p is covered by a point s if p is inside the box with size 2δ centered at s. Finding the minimal cover can be done by using a dynamic rectilinear range search tree in k-dimension [11]. The running time is $O(n \operatorname{polylog} n)$. Notice that every point can be covered by at most a constant number of clusterheads, thus we can find the nearest clusterhead for every point in linear time in total. To find a bridge between two clusterheads s_1, s_2, we compute the bichromatic closest pair between two sets $C(s_1), C(s_2)$. Let $m_1 = |C(s_1)|$ and $m_2 = |C(s_2)|$. According to [1], when $k = 3$, it takes $O((m_1 m_2)^{2/3} \operatorname{polylog} n)$ time; and when $k = 4$, it is

$$\begin{aligned}
&O((m_1 m_2)^{1-1/(\lceil k/2 \rceil + 1) + \varepsilon} + m_1 \log m_2 + m_2 \log m_1) \\
= \;&O((m_1 m_2)^{1-1/k} + m_1 \log m_2 + m_2 \log m_1)\,.
\end{aligned}$$

Since each set is only involved in $O(1)$ bichromatic closest pair computation, the total time is $O(n^{4/3} \operatorname{polylog} n)$ when $k = 3$ and $O(n^{2-2/k})$ for $k \geq 4$. Computing the WSPD on the clusterheads takes $O(n^{2-2/k})$ time, according to Theorem 3.1. □

3.3 Estimating distance between pairs

In the above, we showed how to construct well separated pair decomposition for unit-disk and unit-ball graphs. As mentioned in the introduction, to apply WSPD in solving proximity problems in the unit-disk graphs, we first need to estimate the shortest path distances between $O(n \log n)$ pairs of the WSPD. Note that in our construction for the point sets with constant bounded density, we use Euclidean distance as a lower bound for the unit-disk graph distance and the size of the point set as an upper bound for the diameter. While these approximations are sufficient for bounding the size of WSPD's, it is too coarse for obtaining good approximation. Recall that $\sigma(A)$ is an (arbitrary) point picked from a set A. For a c-well-separated pair (A, B), we can use the estimated distance $\tilde{\pi}(\sigma(A), \sigma(B))$ to approximate all the pairwise distances between points in A and points in B. In this section, we show several tradeoffs for measuring the distance between m pairs of points in the unit-disk graph.

Denote by $\tau(n, c, m)$ the time needed to compute m-pairs c-approximate distance in a unit disk graph. In what follows, we set $c_0 = 2.42 > \frac{4\sqrt{3}}{9}\pi$ and c_1 a number slightly smaller than c_0 but greater than $\frac{4\sqrt{3}}{9}\pi$. We have that:

Lemma 3.12. *1. $\tau(n, c_1, m) = O(n \log^3 n + m)$.*

2. $\tau(n, 1+\varepsilon, m) = O(n^2/(\varepsilon r) + mr/\varepsilon)$, for any $1 \le r \le n$.

Proof. 1. We first construct a planar $\frac{4\sqrt{3}}{9}\pi$-spanner of the unit disk graph. Such spanner exists and can be computed in $O(n \log n)$ time [25]. Now, we apply Thorup's construction of $(1+\varepsilon)$-approximate distance oracle [33] to that planar spanner, for a sufficiently small constant $\varepsilon > 0$. The bound follows immediately from the preprocessing and query time bounds of Thorup's algorithm.

2. We again cluster the points and consider the set of clusterheads, X. Suppose that we have constructed a $(1+\varepsilon/2)$-approximate shortest distance oracle for X. For two query points q_1, q_2, if $\mathrm{d}(q_1, q_2) \le 1$, we return $\mathrm{d}(q_1, q_2)$. Otherwise, we find the clusterheads s_1, s_2 that cover q_1 and q_2, respectively, and return $\hat{\pi}(q_1, q_2) = \pi'(s_1, s_2) + 2\delta$ as an approximation of $\pi(q_1, q_2)$. It is easily verified that $\hat{\pi}(q_1, q_2)$ is a $(1+\varepsilon)$-approximation for $\delta = O(\varepsilon)$. The density of X is $O(1/\delta^2) = O(1/\varepsilon^2)$. The graph formed by connecting neighboring pairs in X is an $O(1/\varepsilon^2)$-overlap graph as defined in [27] and therefore admits a balanced separator with size $O(\sqrt{n}/\varepsilon)$. Furthermore, it can be computed in deterministic linear time by the method in [13].

Now, it is easy to extend the shortest distance algorithm for planar graphs in [3] to the above geometric graph on X. By using the same technique, we can obtain a tradeoff with $O(n^2/(\varepsilon r))$ preprocessing time and $O(r/\varepsilon)$ query time for any $1 \le r \le \sqrt{n}$. □

4. APPLICATIONS

In this section, we show the application of the well separated pair decomposition in obtaining efficient algorithms for approximating the furthest neighbor (diameter, center), nearest neighbor (closest pair), median, and stretch factor, all under the unit-disk graph metric. Since the running time of the algorithms for computing c_0-approximate and $(1+\varepsilon)$-approximate distance are different, we will be describing the bounds for both approximations (recall that $c_0 = 2.42$). Roughly speaking, our algorithms for computing c_0-approximation is about linear and for computing $(1+\varepsilon)$-approximation is about $O(n\sqrt{n})$, dominated by the distance estimation.

We should note that for the problems of computing diameter and center, there is a simple linear time method to achieve 2-approximation. It is therefore not interesting to present algorithms to obtain c_0-approximation for those problems. For the other problems, it is still interesting as we are not aware of any algorithms that achieve comparable approximation ratio in sub-quadratic time, even for planar graphs.

We need to first describe the well-separated pair decomposition we will be using. In what follows, we also include the time for measuring the distances between pairs into the construction time. For c_0-approximation, we construct a c-well-separated pair decomposition $\mathcal{P}_1$ for sufficiently large constant c and, for each pair (A, B) in the WSPD, compute c_1-approximate distance $\hat{\pi}_1(A, B)$ between $\sigma(A)$ and $\sigma(B)$ according to Lemma 3.12.1. For $(1+\varepsilon)$-approximation, we compute a c-well-separated pair decomposition $\mathcal{P}_2$ for $c = O(1/\varepsilon)$ and, for each pair (A, B), compute the $(1+\varepsilon/2)$-approximate distance $\hat{\pi}_2(A, B)$ between $(\sigma(A), \sigma(B))$ by Lemma 3.12.2 and by setting $r = \varepsilon^2\sqrt{n/\log n}$. The following is immediate.

Lemma 4.1. *$\mathcal{P}_1$ contains $O(n \log n)$ pairs and can be computed in $O(n \log^3 n)$ time. $\mathcal{P}_2$ contains $O(n \log n/\varepsilon^4)$ pairs and can be computed in $O(n\sqrt{n \log n}/\varepsilon^3)$ time. For any pair of points (p, q), suppose that its covering pair in $\mathcal{P}_1(\mathcal{P}_2)$ is (A, B), then $\hat{\pi}_1(A, B)$ $(\hat{\pi}_2(A, B))$ is a c_0-approximation ($(1+\varepsilon)$-approximation) of $\pi(p, q)$.*

In the process of producing a well-separated pair decomposition, we constructed several trees, the balanced hierarchical decomposition tree for constant bounded density points and the fair split trees for geometric well-separated pair decomposition [10]. For presentation simplicity, we treat them as a single tree T_1' and T_2', for $\mathcal{P}_1$ and $\mathcal{P}_2$ respectively, by joining the trees created in the geometric well-separated pair decomposition to the clusterheads appropriately. In what follows, $\mathcal{P}, T', \hat{\pi}$ mean that they could be either case.

4.1 $(1+\varepsilon)$-distance oracle

While the computation for $\mathcal{P}_2$ takes time $O(n\sqrt{n \log n}/\varepsilon^3)$, the space needed is only $O(n \log n/\varepsilon^4)$. We can use it to answer $(1+\varepsilon)$-approximate distance query between any two points (p_1, p_2) by first locating the pair (A, B) that covers (p, q) and returning $\hat{\pi}(A, B)$. The query time is the time needed to discover a pair in $\mathcal{P}_2$ that covers the query pair. We show this can be done in $O(1)$ time by using the properties of WSPD.

Corollary 4.1. *For a unit-disk graph on n points and for any $\varepsilon > 0$, we can preprocess it into a data structure with $O(n \log n/\varepsilon^4)$ size so that for any query pair, a $(1+\varepsilon)$-approximate distance can be answered in $O(1)$ time.*

Proof. It suffices to prove it for constant-bounded density point sets. We store all the pairs in $\mathcal{P}$ in a hash table indexed by the pairs. We will show that for each query pair (p, q), we can find $O(1)$ candidate pairs that are guaranteed to contain the pair in $\mathcal{P}$ that covers (p, q). Then, we simply query the hash table using those candidate pairs and discover the one that does cover (p, q).

We modify our construction in Section 3.1 so that we are more careful on deciding when to include a pair in $\mathcal{P}$. We use a c_1-approximate distance oracle as constructed in Lemma 3.12.1. When producing $\mathcal{P}$, we include a pair in $\mathcal{P}$ if $\hat{\pi}(A, B) > (cc_1+2)\max(|A|-1, |B|-1)$. Then there is a constant $c_2 > 0$, such that for any $c \ge 2$ and any pair $(A, B) \in \mathcal{P}$, $cc_1 s \le \pi(A, B) \le cc_2 s$, where $s = \max(|A|-1, |B|-1)$.

Now, to answer a query (p, q), we first use the c_1-approximate distance oracle to compute an approximation ℓ of $\pi(p, q)$, i.e. $\pi(p, q) \le \ell \le c_1\pi(p, q)$. Suppose that $(A, B) \in \mathcal{P}$ is the pair that covers (p, q). Without loss of generality, let us assume that $|A| \ge |B|$, i.e. $s = |A| - 1$. Then, we have

$$s \le \pi(A, B)/(cc_1) \le \pi(p, q)/(cc_1) \le \ell/(cc_1)\,.$$

On the other hand $s \ge \pi(A, B)/(cc_2) \ge (\pi(p, q) - 2s)/(cc_2)$. That is, $s \ge \pi(p, q)/(cc_2 + 2) \ge \ell/(c_1(cc_2 + 2))$.

Set $\hat{\ell} = \ell/(cc_1)$. Then, for (A, B) to cover (p, q), A has to be an ancestor of p in T', and the size of A is sandwiched by $\hat{\ell}/(c_2+1)$ and $\hat{\ell}$. Notice that c_1, c_2 are constants independent of c. There are only $O(1)$ such nodes in T'. Since $|A|/\beta \le |B| \le |A|$ for a constant β, there are only $O(1)$ such B's as well. We now simply form the $O(1)$ candidate pairs by joining every possible A and B. □

4.2 Furthest neighbor

Suppose that $S_1 \subseteq S$. For any p, define the (relative) furthest neighbor of p to be $\xi(p) = \arg\max_{q \in S_1} \pi(p, q)$ in S_1. Then the diameter of S_1 is $D(S_1) = \max_{p \in S_1} \pi(p, \xi(p))$. The center of S_1 is the point that minimizes the maximum distance to the other points, i.e. $\arg\min_{p \in S_1} \pi(p, \xi(p))$. Therefore, once we have compute approximate furthest neighbors for all the p's, we also obtain approximate diameter and center.

Consider any WSPD. To compute the furthest neighbor of S_1, we traverse the balanced hierarchical decomposition tree T' and mark all the nodes $v \in T'$ where $S(v) \cap S_1 \neq \emptyset$. This can be done in $O(n)$ time in a post-order visit of the tree. A pair $P = (S(u), S(v))$ is called *marked* if both u and v are marked. Let

$$R_1(u) = \max\{\hat{\pi}(S(u), B) | (S(u), B) \text{ is marked}\},$$

and 0 if there is no such pair. With each node u, we also record $\ell(u)$, the node that achieves $R_1(u)$.

For any $p \in S_1$, consider the path P in T' from p to the root. Suppose that u is the node that maximizes $R_1(u)$ among all the nodes on P. Now, we pick any point, say q, from $S(\ell(u)) \cap S_1$ (since $\ell(u)$ is marked, $S(\ell(u)) \cap S_1 \neq \emptyset$) and claim that it is an approximate furthest neighbor with the approximation ratio 2.42, if the above process is applied to $\mathcal{P}_1$, or $1+\varepsilon$, if applied to $\mathcal{P}_2$. For the correctness, consider the (marked) pair in $\mathcal{P}$ that covers $(p, \xi(p))$. Suppose it is $(S(u), S(v))$. Then $R_1(u) \ge \hat{\pi}(S(u), S(v))$. Since the pairs are well-separated, it is easy to see that q is an approximate furthest neighbor of p with the approximation ratio determined by the WSPD we use. After we have computed the approximate furthest neighbor, it is simple to compute the diameter and the center. Therefore, we have that

Corollary 4.2. *For any set S of n points in the plane and any $S_1 \subseteq S$, we can compute*

- *c_0-approximate furthest neighbor for all the points in S_1 in $O(n \log^3 n)$ time; and*
- *$(1+\varepsilon)$-approximation, for any $\varepsilon > 0$, of the furthest neighbor, the diameter of S_1, and the center of S_1 in $O(n\sqrt{n \log n}/\varepsilon^3)$ time.*

Remark. We did not list c_0-approximation for the diameter and the center because there is a simple linear time 2-approximate algorithm.

4.3 Nearest neighbor, closest pair

Computing the nearest neighbor or closest pair in S under the unit-disk graph metric is trivial — it is the same as under the Euclidean metric as long as the graph is connected. However, the problem becomes harder if we restrict our attention to a subset $S_1 \subseteq S$, i.e. computing the nearest neighbor in S_1 for each point in S_1 or computing the closest pair between points in S_1. For any two sets S_1, S_2, we can also define the bichromatic closest pair to be $\arg\min_{p \in S_1, q \in S_2} \pi(S_1, S_2)$.

By using the same technique in the previous section, we are able to show:

Corollary 4.3. *For any set S of n points in the plane, and any $S_1, S_2 \subseteq S$, we can compute*

- *c_0-approximation for the nearest neighbor for all the points in S_1, the closest pair in S_1, the bichromatic closest pair of S_1, S_2, in time $O(n \log^3 n)$; and*
- *$(1+\varepsilon)$-approximation for the same problems in time $O(n\sqrt{n \log n}/\varepsilon^3)$.*

Remark. We should note that by applying the technique in [10], we can actually enumerate $O(n)$ pairs of points that is guaranteed to include the closest pair. However, since our distance oracle is approximate, we can only compute approximate closest pair, unlike in the geometric case.

4.4 Median

Similar to the definition of center, median is defined to be the point that minimizes the average (or total) distance to all the other points. Let $\rho(p) = \sum_{q \in S_1} \pi(p, q)$. Then the median of S_1 is the point that minimizes $\rho(p)$.

By using similar technique, we can show that

Corollary 4.4. *For any planar point set S with n points and $S_1 \subseteq S$, a c_0-approximate median of S_1 can be computed in $O(n \log^3 n)$ time, and for any $\varepsilon > 0$, a $(1+\varepsilon)$-approximation can be computed in $O(n\sqrt{n \log n}/\varepsilon^3)$ time.*

Proof. Computing approximate median is similar to computing the furthest neighbor. The only difference is that instead of computing $R_1(u)$, we compute

$$R_2(u) = \sum_{(S(u), B) \in \mathcal{P}} \hat{\pi}(S(u), B) \cdot |B|,$$

and then for each point p and the path P from p to the root, compute $\hat{\rho}(p) = \sum_{u \in P} R_2(u)/(n-1)$, as an approximation of $\rho(p)$. The correctness is guaranteed by the property of pair decomposition that every pair of points is covered by a unique pair in the decomposition. Again, we pick the point with the minimum $\hat{\rho}(p)$ to be the approximate median. The approximation ratio and running time bounds follow immediately. □

4.5 Stretch factor

For a graph G defined on S, the *stretch factor* of G with respect to π is defined as $\max_{p,q \in S} \pi_G(p, q)/\pi(p, q)$. Narasimhan and Smid [28] gave an algorithm to approximate the stretch factor of a geometric graph to the Euclidean metric using the geometric well-separated pair decomposition. By following the same argument we can approximate the stretch factor of an arbitrary graph G with respect to the unit-disk graph metric. Again, we consider the well-separated pair decomposition $\mathcal{P}$. For each pair $(A, B) \in \mathcal{P}$, we pick any two points (p, q) where $p \in A$ and $q \in B$ and compute the approximate shortest path $\hat{\pi}_G(p, q)$ in G and $\hat{\pi}(p, q)$ in I. The maximum ratio of $\hat{\pi}_G(p, q)/\hat{\pi}(p, q)$ over all pairs in $\mathcal{P}$ is an approximation to the stretch factor by the same argument in [28].

Corollary 4.5. *For any graph G on S, we can compute an $O(1)$-approximate stretch factor of G in time $O(\tau_1'(n \log n))$ where $\tau_1'(m)$ is the time to compute m $O(1)$-approximate* *shortest path queries in G. In particular, if G is a subgraph of I, an $O(1)$-approximate can be computed in time $O(n \log^3 n)$. Similarly, we can compute for any $\varepsilon > 0$, a $(1+\varepsilon)$-approximate stretch factor of G in time $O(\tau_2'(n \log n/\varepsilon^4) + n\sqrt{n \log n}/\varepsilon^3)$, where $\tau_2'(m)$ is the time to compute m $(1+\varepsilon)$-approximate shortest path queries in G. When G is a subgraph of I, this can be done in $O(n\sqrt{n \log n}/\varepsilon^3)$ time.*

5. EXTENSIONS

There are several direct extensions of our techniques. Here, we outline the extension to the intersection graph of disks with bounded radii ratio and to the unweighted unit-disk graph.

When the size of the disks are not uniform, it is generally not possible to obtain sub-quadratic well-separated pair decomposition of the metric induced by the intersection graph. This can be shown by the example where there is a big disk and $n-1$ pairwise disjoint small disks intersecting it. Indeed, the intersection graph of this example is a tree with one internal node and $n-1$ leaves.

However, if the ratio between the radii of any two disks (or balls) is upper bounded by a constant, then the packing property (Lemma 3.3) still holds. We can obtain similar results for the intersection graph of disks (or balls in high dimensions) with bounded radii ratio.

In the paper, we only consider weighted unit-disk graphs. There are applications in which people are interested in the unweighted unit-disk graph. Our results for point set with constant bounded density can be directly extended to unweighted unit-disk graphs. If the density is unbounded, then it is impossible to obtain a sub-quadratic size well-separated pair decomposition as shown by the example of the unweighted complete graph. But again we can apply the clustering technique to reduce it to the problem for point sets with constant unbounded density. The clustering will increases the approximation ratio by a multiplicative factor of 3. This gives us almost linear time algorithms to find $3c_0$-approximate solutions to the following problems: the furthest neighbor, nearest neighbor, closest pair, bichromatic closest pair, median, and stretch factor, all with respect to the unweighted unit-disk graph metric. Again, we did not list the problems of computing diameter and center because there are trivial 2-approximate algorithms.

6. CONCLUSION

In this paper, we extend the well-separated pair decomposition, originally developed in the Euclidean metric, to the unit-disk and unit-ball graph metrics. This allows us to obtain almost linear time 2.42-approximate and sub-quadratic time $(1+\varepsilon)$-approximate algorithms for several proximity problems where no efficient methods were previously known. The combinatorial bounds in $\mathbb{R}^k$ for $k \geq 3$ are also tight.

The most notable open problem is the gap between $\Omega(n)$ and $O(n \log n)$ on the number of pairs needed in the plane. Also, the time bound for $(1+\varepsilon)$-approximation is still about $O(n\sqrt{n})$ because of the lack of efficient method for computing $(1+\varepsilon)$-approximate shortest distance between $O(n)$ pairs of points. Any improvement to the algorithm for that problem will immediately lead to improvement to all the $(1+\varepsilon)$-approximate algorithms presented in this paper.

7. REFERENCES

[1] P. K. Agarwal, H. Edelsbrunner, O. Schwarzkopf, and E. Welzl. Euclidean minimum spanning trees and bichromatic closest pairs. *Discrete Comput. Geom.*, 6(5):407–422, 1991.

[2] D. Aingworth, C. Chekuri, and R. Motwani. Fast estimation of diameter and shortest paths (without matrix multiplication). In *Proc. 7th ACM-SIAM Symposium on Discrete Algorithms*, pages 547–553, 1996.

[3] S. R. Arikati, D. Z. Chen, L. P. Chew, G. Das, M. H. M. Smid, and C. D. Zaroliagis. Planar spanners and approximate shortest path queries among obstacles in the plane. In J. Díaz and M. Serna, editors, *Fourth Annual European Symposium on Algorithms*, pages 514–528, 1996.

[4] 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.

[5] 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.

[6] H. Breu and D. G. Kirkpatrick. Unit disk graph recognition is NP-hard. *Computational Geometry: Theory and Applications*, 9(1-2):3–24, 1998.

[7] 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.

[8] P. B. Callahan and S. R. 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 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.

[10] 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.

[11] Y.-J. Chiang and R. Tamassia. Dynamic algorithms in computational geometry. *Proc. IEEE*, 80(9):1412–1434, Sept. 1992.

[12] B. N. Clark, C. J. Colbourn, and D. S. Johnson. Unit disk graphs. *Discrete Mathematics*, 86:165–177, 1990.

[13] D. Eppstein, G. L. Miller, and S.-H. Teng. A deterministic linear time algorithm for geometric separators and its applications. In *Proc. 9th Annu. ACM Sympos. Comput. Geom.*, pages 99–108, 1993.

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

[15] T. Erlebach, K. Jansen, and E. Seidel. Polynomial-time approximation schemes for geometric graphs. In *Proceedings of the 12th ACM-SIAM Symposium on Discrete Algorithms*, pages 671–679, Washington, DC, 7-9 2001.

[16] J. Gao, L. J. Guibas, J. Hershberger, L. Zhang, and

A. Zhu. Geometric spanner for routing in mobile networks. In *Proc. 2nd ACM Symposium on Mobile Ad Hoc Networking and Computing*, pages 45–55, 2001.

[17] S. Govindarajan, T. Lukovszki, A. Maheshwari, and N. Zeh. I/O efficient well-separated pair decomposition and its applications. In *Proc. 8th European Symposium on Algorithms*, pages 220–231, 2000.

[18] 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.

[19] L. Guibas, A. Ngyuen, D. Russel, and L. Zhang. Collision detection for deforming necklaces. In *Proc. 18th ACM Symposium on Computational Geometry*, pages 33–42, 2002.

[20] W. K. Hale. Frequency assignment: Theory and applications. *Proceedings of the IEEE*, 68(12):1497–1513, 1980.

[21] H. B. Hunt, M. V. Marathe, V. Radhakrishnan, S. S. Ravi, D. J. Rosenkrantz, and R. E. Stearns. NC-approximation schemes for NP- and PSPACE-hard problems for geometric graphs. *J. Algorithms*, 26(2):238–274, 1998.

[22] D. B. Johnson and D. A. Maltz. Dynamic source routing in ad hoc wireless networks. In Imielinski and Korth, editors, *Mobile Computing*, volume 353. Kluwer Academic Publishers, 1996.

[23] P. Klein. Preprocessing an undirected planar network to enable fast approximate distance queries. In *Proc. 13th ACM-SIAM Symposium on Discrete Algorithms*, pages 820–827, 2002.

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

[25] X.-Y. Li, G. Calinescu, and P.-J. Wan. Distributed Construction of a Planar Spanner and Routing for Ad Hoc Wireless Networks. In *IEEE INFOCOM 2002*, New York, NY, June 23–27 2002.

[26] C. A. Mead and L. Conway. *Introduction to VLSI Systems*. Addison-Wesley, 1980.

[27] G. L. Miller, S.-H. Teng, and S. A. Vavasis. An unified geometric approach to graph separators. In *Proc. 32nd Annu. IEEE Sympos. Found. Comput. Sci.*, pages 538–547, 1991.

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

[29] C. Perkins and P. Bhagwat. Highly dynamic destination-sequenced distance-vector routing (DSDV) for mobile computers. In *Proc. of ACM SIGCOMM'94 Conference on Communications Architectures, Protocols and Applications*, pages 234–244, 1994.

[30] C. E. Perkins and E. M. Royer. Ad hoc on-demand distance vector routing. In *Proc. IEEE Workshop on Mobile Computing Systems and Applications*, pages 90–100, 1999.

[31] W. D. Smith and N. C. Wormald. Geometric separator theorems and applications. In *Proc. 39th IEEE Symposium on Foundations of Computer Science*, pages 232–243, 1998.

[32] K. J. Supowit. The relative neighborhood graph with an application to minimum spanning trees. *J. ACM*, 30(3):428–448, 1983.

[33] M. Thorup. Compact oracles for reachability and approximate distances in planar digraphs. In *Proc. 42nd IEEE Symposium on Foundations of Computer Science*, pages 242–251, 2001.

[34] M. Thorup and U. Zwick. Approximate distance oracles. In *Proc. ACM Symposium on Theory of Computing*, pages 183–192, 2001.

[35] G. Toussaint. The relative neighborhood graph of a finite planar set. *Pattern Recognition*, 12(4):261–268, 1980.

[36] U. Zwick. Exact and approximate distances in graphs - a survey. In *Proc. of 9th ESA*, pages 33–48, 2001.