Computing Diameter+2 in Truly-Subquadratic Time for Unit-Disk Graphs

Hsien-Chih Chang
Department of Computer Science, Dartmouth College, Hanover, NH, USA

Jie Gao
Department of Computer Science, Rutgers University, Piscataway, NJ, USA

Hung Le
Manning CICS, University of Massachusetts Amherst, MA, USA

Abstract

Finding the diameter of a graph in general cannot be done in truly subquadratic assuming the Strong Exponential Time Hypothesis (SETH), even when the underlying graph is unweighted and sparse. When restricting to concrete classes of graphs and assuming SETH, planar graphs and minor-free graphs admit truly subquadratic algorithms, while geometric intersection graphs of unit balls, congruent equilateral triangles, and unit segments do not. *Unit-disk graphs* is one of the major open cases where the complexity of diameter computation remains unknown. More generally, it is conjectured that a truly subquadratic time algorithm exists for pseudo-disk graphs where each pair of objects has at most two intersections on the boundary.

In this paper, we show a truly-subquadratic algorithm of running time $\tilde{O}(n^{2-1/18})$, for finding the diameter in a unit-disk graph, whose output differs from the optimal solution by at most 2. This is the first algorithm that provides an additive guarantee in distortion, independent of the size or the diameter of the graph. Our algorithm requires two important technical elements. First, we show that for the intersection graph of pseudo-disks, the *graph VC-dimension* – either of k-hop balls or the distance encoding vectors – is 4. This contrasts to the VC dimension of the pseudo-disks themselves as geometric ranges (which is known to be 3). Second, we introduce a *clique-based r-clustering* for geometric intersection graphs, which is an analog of the r-division construction for planar graphs. We also showcase the new techniques by establishing new results for distance oracles for unit-disk graphs with subquadratic storage and $O(1)$ query time. The results naturally extend to unit L_1 or L_∞-disks and fat pseudo-disks of similar size. Last, if the pseudo-disks additionally have bounded ply, we have a truly subquadratic algorithm to find the *exact* diameter.

2012 ACM Subject Classification Theory of computation → Shortest paths; Theory of computation → Computational geometry

Keywords and phrases Unit-disk graph, pseudo-disks, r-division, VC-dimension, distance oracle, clique-based separator

Digital Object Identifier 10.4230/LIPIcs.SoCG.2024.38

Related Version *Full Version*: `https://arxiv.org/abs/2401.12881`

Funding *Jie Gao*: Supported partially by NSF DMS-2220271, IIS-2229876, CCF-2208663, CCF-2118953.
Hung Le: Supported by the NSF CAREER Award No. CCF-2237288 and an NSF Grant No. CCF-2121952.

1 Introduction

Given a set F of n objects in the d-dimensional Euclidean space $\mathbb{R}^d$, the *geometric intersection graph* $F^\times$ has vertices representing the objects in F and edges representing two overlapping objects. When the objects F are disks of radius 1, the intersection graph is called the *unit-disk graph*, where the vertices are centers of the disks in F and two vertices are connected if and only if their distance is no more than 2. Unit-disk graphs have been widely used to model wireless communication. It is also an interesting family of graphs that admit approximation schemes for many graph optimization problems [25, 31].

Geometric intersection graphs, unlike planar graphs, can be dense. But such graphs can be implicitly represented by storing only the set of objects, and the existence of an edge in the graph can often be verified by directly examining the two corresponding objects. Thus many algorithms on geometric intersection graphs avoid computing the set of edges explicitly. For example, single-source shortest paths in (unweighted) unit-disk graphs can be done in time $O(n \log n)$ [19, 11, 12], even though the graph may have $\Theta(n^2)$ many edges. All-pairs shortest paths can be solved in near-quadratic time for several geometric intersection graphs, including disks, axis-parallel segments, fat triangles in the plane, and boxes in constant dimensional spaces [13].

In this paper, we examine two distance-related problems, namely, the graph diameter problem and the distance oracle problem for geometric intersection graphs, in particular for unit-disk graphs. A fundamental problem in this area is to determine whether Diameter problem can be solved in truly subquadratic time for geometric intersection graphs. This is answered negatively for many types of geometric intersection graphs [9] using a reduction from the Orthogonal Vector Conjecture [32] (which is implied by SETH): Deciding if diameter is at most 2 for unit segments in $\mathbb{R}^2$, congruent equilateral triangles in $\mathbb{R}^2$, axis-parallel hypercubes in $\mathbb{R}^{12}$; and deciding if diameter is at most k for unit balls in $\mathbb{R}^3$, axis-parallel unit cubes in $\mathbb{R}^3$ and axis-parallel line segments in $\mathbb{R}^2$. On the positive side, one can decide in $O(n \log n)$ time whether graph diameter is *at most two* for unit-square graphs in $\mathbb{R}^2$. However, for unit-disk graphs, arguably the most basic intersection graphs, the complexity of Diameter problem remains wide open.

► **Question 1.** *Can we compute the diameter of unit-disk graphs in truly-subquadratic time?*

Currently, there is no strong evidence that the answer of Question 1 is positive or negative. As we mentioned above, Diameter for unit-ball graphs in dimension at least 3 does not have a truly-subquadratic time algorithm unless the Orthogonal Vector Conjecture is false. On the other hand, dimension 2 is fundamentally different from dimension 3 or above, and there exist problems that are hard for dimension 3 or above but become much easier in $\mathbb{R}^2$ [9].

Given the lack of progress on Question 1, it is natural to consider approximation algorithms. When edges in unit-disk graphs are given their Euclidean distances as weights, finding $(1+\varepsilon)$-approximation of the graph diameter takes $\tilde{O}(n^{3/2})$ time [22]; this is later improved to near-linear time [14]. Their approach could be modified to handle unweighted unit-disk graphs to get a *hybrid* $(1+\varepsilon, 4+2\varepsilon)$*-approximation* algorithm for Diameter, meaning that the returned approximate diameter is at most $(1+\varepsilon)D + (4+2\varepsilon)$ where D is the true diameter. The difference is because, when the edges are weighted, for a dense set of disks (e.g., forming cliques of arbitrary size) we can use a subset of disks of density $O(1/\varepsilon^2)$ to obtain a $(1+\varepsilon)$-multiplicative distance approximation; this is no longer true in the unweighted setting – even removing one disk can potentially introduce a constant additive error to the diameter. While these results indicate that being on the Euclidean plane helps, stronger evidence supporting the positive answer for Question 1 would be a $+\beta$*-additive approximation*, where the returned diameter lies in between D and $D + \beta$.

► **Question 2.** *Can we compute* $+\beta$*-approximation of the diameter of (unweighted) unit-disk graphs for some constant* β *in truly subquadratic time?*

A much more general and harder problem is to compute the diameter for the intersection graphs of pseudo-disks [9]. Not surprisingly, we are very far from having the answer, given that the unit-disk case remains wide open (Question 1). Unlike the unit-disk graphs, to the best of our knowledge, there are no known *non-trivial approximation* of the diameter in truly-subquadratic time, even for pseudo-disks with constant complexity. In this work, we consider the possibility of obtaining a purely additive approximation of diameter for the intersection graphs of pseudo-disks with constant complexity that have reasonable shapes. Specifically, we assume that the pseudo-disks are *fat objects* that have *constant complexity* and are similar in size – those that can be sandwiched between two disks of the same center of radius r and R, where $r \leq R$ being two universal constants. These objects include unit L_p-disks, as well as same-size constant-sided convex polygons.

▶ **Question 3.** *Can we compute $+\beta$-approximation of the diameter of (unweighted) intersection graphs of similar-size pseudo-disks with constant complexity for* some constant β *in truly-subquadratic time?*

One source of difficulty in computing diameter in truly-subquadratic time of geometric intersection graphs is that the explicit representation of the intersection graphs could have $\Theta(n^2)$ edges. This naturally raises the question of obtaining such an algorithm for *sparse intersection graphs*, where the number of edges is $O(n^{2-\delta})$ for some constant $\delta > 0$. The answer to this question also remains open. A significant progress toward answering this question would be the case of *constant ply*. A set of objects is said to have *ply k* if every point in the space can stab at most k objects in the set.

▶ **Question 4.** *Can we compute the* exact diameter *of (unweighted) intersection graphs of similar-size pseudo-disks with constant complexity and ply in truly-subquadratic time?*

A positive answer to Question 4 also provides strong evidence for a positive answer to Question 1, as unit-disk graphs of constant ply is a special case of similar-size pseudo-disks with constant complexity and ply.

1.1 Main Results

In this work, we resolve Questions 2, 3, and 4 affirmatively. We can even set the additive approximation constant β as small as 2, which is almost close to the true diameter. First, we present our results for unit-disk graphs.

▶ **Theorem 1.** *There is an algorithm computing a $+2$-approximation of the diameter of any given unweighted unit-disk graph with n vertices in $\tilde{O}(n^{2-1/18})$ time.*

Our algorithm is a combination of two technical ingredients. (1) We show that both the distance encoding vectors defined by Le and Wulff-Nilsen [28] as well as the set of k-neighborhood balls defined by Ducoffe, Habib, and Viennot [17] have VC-dimension of 4 for unit-disk graphs and pseudo-disk graphs in general. (2) We develop a new *clique-based r-clustering* which is analogous to an r-division for planar and minor-free graphs [20, 33]. The combination is inspired by recent developments in computing diameter in truly-subquadratic time for minor-free graphs [28]; we will discuss these technical ideas in detail in Section 1.2. We then generalize our algorithm for unit-disk graphs to work with similar-size pseudo-disks with constant complexity.

▶ **Theorem 2.** *Given an unweighted n-vertex similar-size pseudo-disk graphs with constant complexity, we can compute a $+2$-approximation of the diameter in $\tilde{O}(n^{2-1/18})$ time.*

In this general case, we need an additional component: a single-source shortest path (SSSP) algorithm with $\tilde{O}(n)$ running time for the intersection graphs of similar-size pseudo-disks with constant complexity. SSSP algorithms with running time $\tilde{O}(n)$ are known for some special cases, including unit-disk graphs [19, 14], unit L_p-disks for $p = 1$ or $p = \infty$ [27], and arbitrary disks [26].

When the objects have bounded ply (or even n^δ-ply for small δ), the intersection graphs have truly-sublinear separators, using the observation by de Berg *et al.* [4] that the intersection graph of fat objects has sublinear clique-based separators. (Indeed, the objects in each clique of the clique-based separators are stabbed by a single point.) We use this fact combined with our VC-dimension result for pseudo-disk graphs to prove the following theorem.

▶ **Theorem 3.** *Let G be an unweighted n-vertex similar-size pseudo-disk graphs of with constant complexity, and let k be the ply of G. We can compute the exact diameter in $\tilde{O}(k^{11/9}n^{2-1/18})$ time.*

The running time of Theorem 3 is truly subquadratic when $k = O(n^{1/22-\varepsilon})$ for any constant ε, including the special case of $k = O(1)$ as asked in Question 4.

Next, we showcase another application of our technique in constructing a *distance oracle* for (unweighted) unit-disk graphs. The same technique in Chan and Skrepetos [14] for the diameter problem mentioned above gives a distance oracle returning a hybrid $(1+\varepsilon, 4+2\varepsilon)$-approximation of the shortest distance using $O(n\log^3 n)$ space and $O(1)$ query time. In the weighted setting, they got a multiplicative $(1+\varepsilon)$-approximation with the same space and query time, improving upon an earlier result [22]. Mark de Berg [3] considered the *transmission graph* where each point has a transmission radius and can reach any vertex within the transmission radius. On this graph (which by definition is directed and unweighted), de Berg presented a distance oracle of size $\tilde{O}(n^{3/2}/\varepsilon)$ that can answer approximate distance queries with a hybrid $(1+\varepsilon, +1)$-approximation in time $\tilde{O}(n^{1/2}/\varepsilon)$. The question is: can we develop a distance oracle with truly-subquadratic space and *constant query time*, returning a purely additive approximation of shortest distances? We use the same technique developed for the diameter problem to answer this question positively.

▶ **Theorem 4.** *Given an unweighted unit-disk graph with n vertices, we can construct a distance oracle with $O(n^{2-1/18})$ space and $O(1)$ query time, returning a $+2$-approximation of the true distances.*

Theorem 4 extends to pseudo-disk graphs as well.

▶ **Theorem 5.** *Given an unweighted n-vertex similar-size pseudo-disk graph with constant complexity, we can construct a distance oracle with $O(n^{2-1/18})$ space and $O(1)$ query time, returning a $+2$-approximation of the true distances.*

Due to space limitation, we defer all proofs and analysis to the full version.

1.2 Technical Ideas

Our technique is inspired directly by recent developments in computing exact diameters for minor-free graphs [17, 28] that combine two well-known tools in geometric algorithms: *VC-dimension* and *r-division*. An r-division is a decomposition of the graph into $\Theta(n/r)$ pieces, each with $O(r)$ vertices and $O(\sqrt{r})$ boundary vertices that are incident to other pieces. The result by Chepoi, Estellon, and Vaxès [16] showed that the set of all k-neighborhood balls in a K_h-minor-free graph, when treated as a set system over the vertices, has VC-dimension at most $h-1$. Ducoffe *et al.* [17] was the first to combine the VC-dimension result [16] and r-division to design truly-subquadratic time algorithm for minor-free graphs. Le and Wulff-Nilsen [28] designed a different VC set system based on that of Li and Parter [29], which is easier to combine with r-division. They obtained, among other things, an improved algorithm for computing exact diameter in minor-free graphs.

We follow a path similar to the one taken for minor-free graphs [28] to design an algorithm for unit-disk graphs. To carry out this plan, we have to develop the two corresponding technical components in the geometric setting: An appropriate VC set system and an r-division for unit-disk graphs. There are two main challenges. The first challenge is that while the definitions of the VC set systems proposed in [29, 28] are naturally applicable to any graphs, their proof technique heavily depends on graphs being minor-free (by building a minor directly from a system of high VC-dimension), and in some case involves tedious case analysis. Our proof for unit disks only relies on their topological property of being *pseudo-disks*. The second challenge is rooted from the reality that r-division does not exist for unit-disk graphs. Here we introduced a new notion called *clique-based r-clustering*, which allows cliques to be on the boundary of each region (called *cluster* in our terminology). Our notion of clique-based r-clustering is inspired by clique-based balanced separators for geometric intersection graphs [4, 5]. However, formulating the right definition for clique-based r-clustering ends up to be delicate and challenging; the paragraphs after Remark 6 explain why several naïve approaches do not work, and our eventual solution. We now elaborate on the two main technical components in more details.

Various definitions of VC-dimensions on graphs. In computational geometry literature, VC-dimension has been used to characterize the complexity and "richness" of geometric shapes [15]. The VC-dimension for unit-disks (as well as disks of all possible radii) is 3 – no four points can be shattered by disks in the plane [30]. We remark that in prior work the *VC-dimension of a graph* is defined on the set system of the closed *immediate* neighborhoods, i.e., for each vertex v, the set of vertices including v and its one-hop neighbors [24, 2, 6]. In this definition, the VC-dimension of a unit-disk graph is 3 as well [6].

Here we study the VC-dimension of two set systems: (1) the set of balls in the geometric intersection graph with radius r ranges over all possible non-negative integers – this is referred to as the VC-dimension of the *ball hypergraph* of G, also called the *distance VC-dimension* of G [16, 7, 17]; (2) the distance encoding vectors as defined in [28] in a unit-disk graph with respect to a set S of k vertices. For both cases we show that the VC-dimension is exactly 4 (not 3) – and we have an example of 4 points that are shattered. In fact, we present a proof that is purely topological and thus can be generalized to the intersection graphs of pseudo-disks – topological disks in the plane bounded by Jordan curves such that the boundaries of any two objects have at most two intersection points. The pseudo-disk requirement is actually crucial and cannot be dropped. For example, we can construct n unit-size equilateral triangles (possibly with rotations) with VC-dimension $\Omega(\log n)$ by modifying the fine-grained hardness construction in Bringmann *et al.* [9, Theorem 17].

▶ Remark 6. In [1] it is shown that unit-disk graph has distance VC-dimension 4. As we completed the VC-dimension results for pseudo-disk graphs, we discovered an independent work posted on arXiv by Duraj, Konieczny and Potępa [18]. They showed that geometric intersection graphs of objects that are closed, bounded, convex, and center symmetric has distance VC-dimension at most 4, which is a subset of our result. Both proof techniques rely on geometry. Our proof approach is purely topological, and applies for general pseudo disk graphs and distance encoding vectors.

Duraj, Konieczny and Potępa [18] combined their distance VC dimension bound with an (improved) argument along the lines of Ducoffe, Habib, and Viennot [17] to design truly-subquadratic time algorithms for intersection graphs of unit squares and translations of convex polygons with center of symmetry when the diameter is small. However, as noted above, it remains an open problem if the same result could be obtained for unit-disk graphs even when the diameter is small – one missing element is a data structure that can efficiently build the r-neighborhood with increasing r. It is unclear if such a data structure could be constructed for unit-disk graphs.

Clique-based r-clustering. As we mentioned above, r-division does not exist in unit-disk graphs. Here we develop an analogous clique-based r-clustering. A *δ-balanced clique-based separator* of a geometric intersection graph G [4, 5] is a collection $\mathcal{C}$ of vertex-disjoint cliques whose removal will partition the graph into two parts of size at most δn, with no edges between the parts. The *clique size* of $\mathcal{C}$ is the number of cliques in $\mathcal{C}$, and the *vertex size* of $\mathcal{C}$ is the total number of vertices in all cliques in $\mathcal{C}$.

As alluded to earlier, the definition and construction of the clique-based analog of r-division requires handling several subtleties. To explain these subtleties, we will suggest some natural ideas and discuss why these ideas do not work.

- *First attempt:* Let D be the input set of n disks, whose intersection graph is G. We could apply the clique-based separator [4, 3] to find a set of $\sqrt{n}$ cliques $\mathcal{S}$ such that $D \setminus \mathcal{S}$ could be partitioned into sets $\{D_1, D_2, \ldots\}$ such that each set of disk D_i has size at most $2n/3$ and induces a maximally connected intersection graph. We call cliques in $\mathcal{S}$ *boundary cliques* and disks in $\mathcal{S}$ *boundary disks*. Ideally, we want to recursively apply the clique-based separators to each set D_i until we obtain the set of clusters $\mathcal{R}$ of size at most r each. The issue here is that the number of boundary cliques adjacent to each region in $\mathcal{R}$ could be arbitrarily large, up to $\Omega(\sqrt{n})$. Note that we want each set to have only $O(r)$ boundary cliques in the same way that r-division guarantees each region to have $O(r)$ boundary vertices.
- *Second attempt:* Instead of separating each D_i directly, we could add the boundary disks in $\mathcal{S}$ back to D_i, and then recursively apply the clique-based separator theorem on each resulting D_i, as done in algorithms for constructing an r-division of planar and minor-free graphs [20, 33]. There are several issues, and one of them is running time. Specifically, $\mathcal{S}$ could contain up to $\Omega(n)$ disks, and by reinserting the boundary disks across different D_i, the number of disks (counted with multiplicity) might be more than n, and hence the total number of disks arising over the course of the entire recursion could be up to $\Omega(n^2)$.
- *Third attempt:* One way to avoid adding too many boundary disks to D_i is to add only *one* boundary disk per clique in $\mathcal{S}$. Specifically, for each clique in $\mathcal{S}$, we choose a disk in the clique to be its *representative*. Next, we add the representative of each clique to D_i, if the clique intersects at least one disk in D_i. We then recursively apply the clique-based separator to the resulting set of disks. Here the total number of disks, counted with multiplicity, is $n + O(\sqrt{n})$ at the second level, and $O(n)$ over all levels.
  However, there is another technical issue with using representative disks of cliques in $\mathcal{S}$. Suppose that we apply the clique-based separator to D_i (after adding the representative boundary disks) to find a clique-based separator $\mathcal{S}_i$. Removing $\mathcal{S}_i$ partitions D_i into two balanced sets of disks X_1 and X_2. There could be a representative disk $x \in D_i$ of a clique in $\mathcal{S}$ that is assigned to X_1 and not to X_2. Yet, the clique represented by x might contain a disk (other than x) that intersects disks in X_2. As x is not in X_2, the algorithm does not correctly capture the boundary disks of X_2, and hence, when the algorithm terminates, the number of boundary cliques of each region could still be $\Omega(\sqrt{n})$.

We ended up with the following (rather delicate) definition of a clique-based r-clustering.

▶ **Definition 7** (Clique-based r-clustering). *Let $r \geq 1$ be a parameter. A* clique-based r-clustering *of a geometric intersection graph G is a pair $(\mathcal{R}, \mathcal{C})$ where $\mathcal{R}$ contains subsets of $V(G)$ called* clusters*, and $\mathcal{C}$ is a set of vertex-disjoint cliques of G such that:*

1. *Every set $R \in \mathcal{R}$ induces a connected subgraph of G. Furthermore, $|\mathcal{R}| = O(n/\sqrt{r})$.*
2. *Every cluster $R \in \mathcal{R}$ can be partitioned into two parts,* boundary *∂R and* interior *R°, such that all vertices in R having neighbors outside R belongs to ∂R, and furthermore, (i) R° has at most r vertices and (ii) ∂R contains at most r cliques in $\mathcal{C}$, denoted by $\mathcal{C}(\partial R)$.*
3. *$\sum_{R\in\mathcal{R}} |\mathcal{C}(\partial R)| = O(n/\sqrt{r})$. This in particular implies that $|\mathcal{C}| = O(n/\sqrt{r})$.*
4. *Every vertex of G either belongs to a clique in $\mathcal{C}$ or to R° for some cluster $R \in \mathcal{R}$.*

There are several differences between our clique-based r-clustering and an r-division in planar graph literature [20]. First, in our clique-based r-clustering we can only guarantee that the internal part R° of each cluster R has size at most r. Indeed, the size of R could be $\Omega(n)$, thus, we only compute an implicit representation of $\mathcal{R}$. Second, the fact that R could have size $\Omega(n)$ makes other algorithms relying on clique-based r-clustering more challenging: we cannot go through every vertex of R to do the computation in the way other planar algorithms do. Third, the number of cliques in the boundary of R is $O(r)$ in the clique-based r-clustering, instead of $O(\sqrt{r})$ in a standard planar r-division. Last but not least, to compute a clique-based r-clustering, we have to rely on a *well-separated clique-based separator*, such that the remaining disks can be partitioned into two sets that are far from each other relative to the radii of the disks.

Specifically, we will compute an *implicit representation* of $\mathcal{R}$: for each clique in $\mathcal{C}$, we will choose an arbitrary vertex to be the *representative* of the clique, and for each cluster $R \in \mathcal{R}$, we explicitly store vertices in R° and all representatives of the cliques in $\mathcal{C}(\partial R)$, denoted by $\mathsf{rep}(R)$. Furthermore, for each vertex $u \in R^\circ$, we will maintain a list of representatives x by which u has a neighbor in the clique represented by x.

▶ **Lemma 8.** *For any given integer r and an n-vertex unit-disk graph G, we can find the implicit representation of a clique-based r-clustering $(\mathcal{R}, \mathcal{C})$ of G in $O(n \log^2 n)$ time.*

Pseudo-disk graphs with constant ply. Ducoffe *et al.* [17] showed that if a monotone class of graphs $\mathcal{G}$ has truly-sublinear balanced separators and distance VC-dimension at most d, then we can compute the diameter in time $O(n^{2-\varepsilon_{\mathcal{G}}(d)})$ where $\varepsilon_{\mathcal{G}}(d) = 1/2^{O_{\mathcal{G}}(d)}$; the $O_{\mathcal{G}}(\cdot)$ notation hides a dependency on the family $\mathcal{G}$. Our results above imply that the family of intersection graphs of similar-size pseudo-disks of constant complexity and ply has truly-sublinear balanced separators and distance VC-dimension at most d. Thus, we can solve diameter exactly in time $O(n^{2-\varepsilon_k(d)})$ where the constant $\varepsilon_k(d)$ depends on the ply k using the algorithm of Ducoffe *et al.* [17] as a black box. However in their algorithm the dependency on k is not explicitly computed, and furthermore, the dependency on d is exponentially diminishing. Instead, we modify our approximation algorithm for unit-disk graphs to obtain a better dependency on k and a smaller constant in the exponent of n. The basic idea is that we could now use r-division instead of clique-based r-division. We note that r-division for low-ply geometric intersection graphs was used earlier to solve different problems by Har-Peled and Quandrud [23].

2 VC-dimension of Unit-Disk and Pseudo-Disk Graphs

2.1 Unit-Disk Graphs and Pseudo-Disk Graphs

An undirected, unweighted *unit-disk graph* is a graph obtained from a set of points P in the plane such that two points are connected by an edge if and only if their Euclidean distance is at most 2. A unit-disk graph is a special type of *geometric intersection graph*, which can be defined for a set F of objects in $\mathbb{R}^d$ where an edge exists between two vertices if and only if the two corresponding objects overlap.

One interesting family of geometric intersection graphs is when the objects are *pseudo-disks*. Specifically, a simple closed Jordan curve C partitions the plane into two regions, one of them is bounded, called the *interior* of C. A family of simple closed Jordan curves is called *pseudo-circles* if every two curves are either disjoint or properly crossed at precisely two points. (Without loss of generality we assume there are no tangencies.) In a family of pseudo-circles, the interior of each pseudo-circle is called a *pseudo-disk*. Each pseudo-disk is a simply connected set and the intersection of a pair of pseudo-disks is either empty or is a connected set [10]. For a family of pseudo-disks $D_1, D_2, \ldots, D_n$, we can construct the intersection graph G of the pseudo-disks – combinatorially, we use a set of vertices with v_i corresponding to D_i and connect an edge for v_i and v_j if and only if D_i and D_j have non-empty intersections.

The following property of unit-disk graphs is folklore (e.g., see Breu [8, Lemma 3.3]).

▶ **Lemma 9.** *If two edges ab and cd in a unit-disk graph intersect, then one of the four vertices a, b, c, d is connected to the rest of three vertices.*

We now prove an analog of Lemma 9 for pseudo-disks. The proof uses only the topological properties of pseudo-disks. If two pseudo-disks D and D' intersect, for any two points $p \in D$ and $p' \in D'$, we can find a curve $\pi(p, p')$ from p to p' inside $D \cup D'$ such that this path can be partitioned into three pieces, at point q, q' with $\pi(p, q) \in D \setminus D'$, $\pi(q, q') \in D \cap D'$ and $\pi(q', p') \in D' \setminus D$. Any of the three pieces may be empty. See Figure 1 for an example. We call the curve $\pi(p, p')$ a *proper curve* connecting p and p', and q, q' the *entrance* and *exit* point respectively.

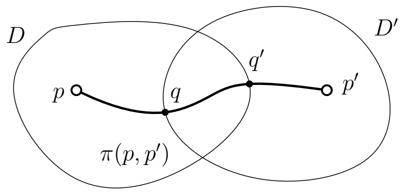

Figure 1 A proper curve $\pi(p, p')$.

▶ **Lemma 10.** *For four pseudo-disks $\{D_a, D_b, D_c, D_d\}$ with D_a intersects D_b and D_c intersects D_d, take four points $a \in D_a$, $b \in D_b$, $c \in D_c$ and $d \in D_d$ and proper curves $\pi(a, b), \pi(c, d)$. If $\pi(a, b), \pi(c, d)$ have an odd number of intersections and none of point $i \in \{a, b, c, d\}$ stays inside any pseudo-disk D_j with $j \in \{a, b, c, d\}$ and $j \neq i$, then one of the four pseudo-disks intersects all three other pseudo-disks.*

Now we consider a pseudo-disk graph. We can find a planar drawing of the pseudo-disk graph in the plane in the following manner: for each pseudo-disk D, we take one *representative point* $p \in D$. If two pseudo-disks D and D' intersect, we connect their representative points $p \in D$ and $p' \in D'$ by a proper curve $\pi(p, p')$. This graph is unweighted, i.e., the proper curve $\pi(p, p')$ has length of 1. *Path* $P(p, q)$ for two pseudo-disks represented by p and q consists of several proper curves visiting the representative points of the pseudo-disks on the path. We use $|P(p, q)|$ to denote the hop length of a path $P(p, q)$.

We can prove a generalized version of Lemma 10 for the hop distances of paths in the pseudo-disk graph. This will be useful for bounding the VC-dimension of set systems defined on the pseudo-disk graph. Consider four vertices a, b, c, d representing four pseudo-disks D_a, D_b, D_c, D_d and assume that there are two paths $P(a, b)$ and $P(c, d)$. We define a *local crossing pattern* to be four distinct vertices a', b', c', d' with a', b' on path $P(a, b)$ (with a' closer to a than b') and c', d' on path $P(c, d)$ (with c' closer to c than d') such that one of the four vertices a', b', c', d' has edges to all the other three vertices.

▶ **Lemma 11.** *Consider four vertices a, b, c, d representing four pseudo-disks D_a, D_b, D_c, D_d and assume that there is a local crossing pattern of the two paths $P(a, b)$ and $P(c, d)$, then the followings are true:*
1. *Either there is a path $P'(a, c)$ whose hop length is at most $|P(c, d)|$ or there is a path $P'(b, d)$ whose hop length is at most $|P(a, b)|$.*
2. *Either there is a path $P'(a, c)$ whose hop length is at most $|P(a, b)|$ or there is a path $P'(b, d)$ whose hop length is at most $|P(c, d)|$.*

2.2 VC-dimension of Unit-Disk Graphs and Pseudo-Disk Graphs

The VC-dimension of a set system $(P, \mathcal{R})$ with $\mathcal{R}$ containing subsets of P is the largest cardinality of a subset $S \subseteq P$ that can be *shattered*, i.e., all subsets of S can be obtained by the intersection of some sets in $\mathcal{R}$ with P. Here we consider the VC-dimension of two other set systems defined on a unit-disk graph or a pseudo-disk graph G, namely the *distance VC-dimension* of G and the *distance encoding VC-dimension* of G.

Distance VC-dimension. In an unweighted graph G, consider the collection of balls $B(v, r)$, which is the set of all points within a hop distance of r from a vertex v. Since we consider unweighted graphs, we assume r to be non-negative integers. We define the *ball system* of a graph G on points P as the sets

$$\mathcal{B}(G) \coloneqq \left\{B(v, r) : \forall v \in P,\, \forall r \in \mathbb{Z},\, r \geq 0\right\}.$$

The VC-dimension of the set of balls with radius r for all possible non-negative integers is referred to as the VC-dimension of the ball hypergraph of G, also called the *distance VC-dimension* of G [17]. It is known that the set system of balls of any undirected (weighted) K_h-minor-free graphs have VC-dimension at most $h - 1$ [16]. Thus the set of balls for planar graphs has VC-dimension at most 4, since a planar graph does not have K_5 as a minor. Notice that a unit-disk graph can be a complete graph thus is not K_h-minor-free for any $h \leq n$. Thus the above result does not immediately apply to a unit-disk graph. Ducoffe *et al.* [17] also showed that interval graphs have distance VC-dimension of two. Unit-disk graph is a natural extension of the interval graph to two dimensional space.

▶ **Theorem 12.** *The distance VC-dimension of a pseudo-disk graph is* 4*.*

Distance encoding VC-dimension. Li and Parter [29] defined a distance encoding function in a graph and used it for computing diameter in a planar graph. Later Le and Wulff-Nilsen [28] used a slighly revised one. We take the definition by Le and Wulff-Nilsen [28] and argue that this set defined on a unit-disk graph also has low VC-dimension.

▶ **Definition 13.** *Let $M \subseteq \mathbb{R}$ be a set of real numbers. Let $S = \{s_0, s_1, \ldots, s_{k-1}\}$ be a sequence of k vertices in an undirected weighted graph $G = (V, E)$. For every vertex v define*

$$X(v) \coloneqq \left\{(i, \Delta) : 1 \leq i \leq k - 1, \Delta \in M, d(v, s_i) - d(v, s_0) \leq \Delta\right\}.$$

Let $L_{G,M}(S) \coloneqq \{X(v) : v \in V\}$ be a set of subsets of the ground set $[k - 1] \times M$.

The set $L_{G,M}(S)$ is a set of "ranges" where each range $X(v)$ corresponds to a vertex $v \in V$, which captures the distance to vertices in S compared to the distance to s_0. We note that s_i and s_{i+1} may not be adjacent in G.

▶ **Theorem 14.** *Let S be as any set of vertices of a pseudo-disk graph G and $M \subseteq \mathbb{R}$ be any set of real numbers. $L_{G,M}(S)$ has VC-dimension at most* 4*.*

3 +2-Approximation for Diameter in Unit-Disk Graphs

As we discussed in Section 1.2, we will combine the distance encoding with the clique-based r-division in Lemma 8, along the line of Le and Wulff-Nilsen [28].

Distance encoding. Fix a sequence of vertices $S = \langle s_0, s_1, \dots, s_{k-1} \rangle$. Following previous work [21, 28], we define a *pattern* of v with respect to S, denoted by $\mathbf{p}_v$, to be a k-dimensional vector where:

$$\mathbf{p}_v[i] \coloneqq d_G(v, s_i) - d_G(v, s_0) \quad \text{for every } 0 \le i \le k-1. \tag{1}$$

Note that $\mathbf{p}_v[0] = 0$ by definition. The following lemma is from [28, 21].

▶ **Lemma 15** ([28, 21]). *Let G be an unweighted graph and S be a sequence of vertices. Suppose that $L_{G,M}(S)$ has VC-dimension at most d for any M. Let $\mathbf{P} \coloneqq \{\mathbf{p}_v : v \in V\}$ be the set of all patterns with respect to S. Suppose that $d_G(s_i, s_0) \le \Delta$ for every $i \in [k-1]$, then $|\mathbf{P}| = O((k \cdot \Delta)^d)$.*

Computing approximate diameter. Our algorithm will be based on a clique-based r-clustering. Let $(\mathcal{R}, \mathcal{C})$ be a clique-based r-clustering for a parameter r to be chosen later. Recall that for each cluster $R \in \mathcal{R}$ the boundary vertices ∂R belong to $O(r)$ cliques in $\mathcal{C}$.

For subgraph $R \in \mathcal{R}$, we define a sequence $S_R = \langle s_0, s_1, \dots, s_{k_R} \rangle$ from all the clique representatives in $\mathsf{rep}(R)$. Note that $k_R = O(r)$ and $\mathsf{rep}(R) \subseteq \partial R$ by Definition 7. Since R is connected, by the triangle inequality, $d_G(s_i, s_0) \le |V(R^\circ)| + |\mathsf{rep}(R)| + 2 = O(r)$. For each vertex $u \in V$, we form a pattern $\mathbf{p}_u$ with respect to S_R, and let $\mathbf{P}_R \coloneqq \{\mathbf{p}_u : u \in V\}$.

Given a pattern $\mathbf{p}_u$ of u and a vertex $v \in G$, we want to estimate the distance $d_G(u, v)$ via $\mathbf{p}_u$. This leads to the definition of distance $d(\mathbf{p}, v)$ between a pattern $\mathbf{p}$ and a vertex v:

$$d(\mathbf{p}, v) \coloneqq \min_i \left\{ d_G(v, s_i) + \mathbf{p}[i] \right\} \tag{2}$$

Previous work [21, 28] showed that if the sequence S_R contains *all* vertices of ∂R, then $d_G(u, v) = d(\mathbf{p}_u, v) + d_G(u, s_0)$. In our setting S_R only contains a *subset* of vertices of ∂R, so recording $d(\mathbf{p}, v)$ does not give us exact distances; however, we get a +2-approximation as shown by the following lemma.

▶ **Lemma 16.** *Suppose that $\pi(u, v, G) \cap \partial R \neq \varnothing$ where $\pi(u, v, G)$ is a shortest path from u to v in G. Let*

$$\tilde{d}_G(u, v) \coloneqq d_G(u, s_0) + d(\mathbf{p}_u, v). \tag{3}$$

Then, $d_G(u, v) \le \tilde{d}_G(u, v) \le d_G(u, v) + 2$.

We now describe our algorithm. The *eccentricity* of a vertex u is defined to be $\mathsf{ecc}(u) \coloneqq \max_{v \in V(G)} d_G(u, v)$. We will compute the approximate diameter by computing the approximate eccentricities for all vertices in G; that is, for each vertex u, we will compute an approximation $\widetilde{\mathsf{ecc}}(u)$, and then output $\max_u \widetilde{\mathsf{ecc}}(u)$. Our algorithm is similar to the algorithm of Le and Wulff-Nilsen [28] for computing exact diameter in minor-free graphs. Here, we use clique-based r-clustering in place of an r-division and have to handle the cliques in $\mathcal{C}$. We also have to be more careful in the way we handle clusters in $\mathcal{R}$ as a cluster could have a very large size. The algorithm has three steps:

- Step 1. Construct a clique-based r-clustering $(\mathcal{R}, \mathcal{C})$ of G. For each clique $K(x)$ in $\mathcal{C}$ represented by a vertex x, we find the shortest path distances from x to all other vertices of G using a single-source shortest path algorithm [19]. For each cluster $R \in \mathcal{R}$, form a sequence of boundary vertices S_R as described above. We compute a set of patterns $\mathbf{P}_R = \{u \in V : \mathbf{p}_u\}$ with respect to S_R. We store all the information computed in this step in a table $T_R^{(1)}$.
- Step 2. For each cluster $R \in \mathcal{R}$, each pattern $\mathbf{p} \in \mathbf{P}_R$, and each vertex $v \in R^\circ$, we compute $d(v, \mathbf{p})$. Then we find $v_\mathbf{p} := \arg\max_{v \in V(R^\circ)} d(v, \mathbf{p})$, which is the *furthest vertex* from $\mathbf{p}$. We store both $d(v, \mathbf{p})$ and $d(\mathbf{p}, v_\mathbf{p})$ in a table $T_R^{(2)}$.
- Step 3. We now compute $\widetilde{\mathsf{ecc}}(u)$ for each vertex $u \in V$. For each cluster $R \in \mathcal{R}$, we compute the approximate distance from u to the vertex $v \in R^\circ$ furthest from u, denoted by $\Delta(u, R^\circ)$, as follows. Let $\mathbf{p}_u$ be the pattern of u with respect to S_R computed in Step 1.
  - Step 3a. If $u \notin R^\circ$, let v be the furthest vertex from $\mathbf{p}_u$, computed in Step 2. We return $\Delta(u, R^\circ) := d_G(u, s_0) + d(\mathbf{p}_u, v)$ where s_0 is the first vertex of S_R.
  - Step 3b. If $u \in R^\circ$, then we compute a distance $d_{R^\circ}(u, v)$ from u to every $v \in R^\circ$ where this distance is in the intersection graph of the disks in R°. Then, compute $\tilde{d}_G(u, v) = d_G(u, s_0) + d(\mathbf{p}_u, v)$ and finally return $\Delta(u, R^\circ) := \max_{v \in R^\circ} \min\left\{\tilde{d}_G(u, v), d_{R^\circ}(u, v)\right\}$.

  We are not done yet: we have to compute the maximum approximate distance, denoted by $\Delta(u, \mathcal{C})$ from u to vertices in cliques in $\mathcal{C}$:

  $$\Delta(u, \mathcal{C}) = 1 + \max_{K(x) \in \mathcal{C}} d_G(u, x) \tag{4}$$

  Here $K(x)$ is the clique in $\mathcal{C}$ represented by x. The distance $d_G(u, x)$ was computed and store in $T_R^{(1)}$ in Step 1. Finally, we compute:

  $$\widetilde{\mathsf{ecc}}(u) = \max\left\{\max_{R \in \mathcal{R}} \Delta(u, R^\circ), \Delta(u, \mathcal{C})\right\} .$$

4 Well-Separated Clique-Based Separator Decomposition

In this section, we prove Lemma 8; see Section 1.2 for an overview of the argument.

▶ **Definition 17** (Well-separated clique-based separators). *Let D be a set of n unit-disks. Let G be its geometric intersection graph. We say a family of disjoint subsets of cliques of G, denoted by $\mathcal{S}$, is a* well-separated clique-based separator *of D if all following conditions hold:*
- **Balanced.** *Every connected component of $G \setminus \mathcal{S}$ contains at most $2n/3$ disks.*
- **Well-separated.** *For every two disks a and b in different components of $G \setminus \mathcal{S}$, the minimum Euclidean distance $\|a, b\|$ between points in a and points in b is greater than 2.*
- **Low-ply.** *The disks in each clique in $\mathcal{S}$ are stabbed by a single point. Furthermore, we could choose for each clique $X \in \mathcal{S}$ a* representative disk *x such that the ply with respect to the intersection graph of all representative disks in $\mathcal{S}$ is $O(1)$.*

We say that the size of $\mathcal{S}$ is the number of cliques in $\mathcal{S}$.

We will show that by adapting the clique-based separator theorem for geometric intersection graphs [4, 3], we can construct a well-separated clique-based separator for unit-disk graphs in near-linear time.

▶ **Lemma 18.** *Let D be a set of n unit disks. We can construct a well-separated clique-based separator $\mathcal{S}$ for D of size $O(\sqrt{n})$ in $O(n \log n)$ time, such that for every disk y, there are $O(1)$ cliques in $\mathcal{S}$ that intersect y. Furthermore, we can compute the list of the representative disks of cliques in $\mathcal{S}$ that intersect y for every disk y in a total of $O(n \log n)$ time.*

Clique-based r-clustering algorithm. Let D be the set of n disks and G be its geometric intersection graph. In this step, we recursively partition D into a family of sets of disks such that each set has at most r disks and at most r boundary cliques. We also maintain a (global) set of cliques $\mathcal{C}$ and their representative disks. For each representative disk x, let $K(x)$ be the clique in $\mathcal{C}$ represented by x.

At each intermediate recursive step, we will maintain an (explicit) set $\hat{D}$ of size at least r that includes two types of disks: *regular disks* and *representative disks*. We assume that $|\hat{D}| \geq r$; otherwise, the algorithm will stop in the previous step. For each regular disk $y \in \hat{D}$, we maintain a list of representative disks, denoted by $\rho(y)$, in $\hat{D}$ such that for each $x \in \rho(y)$ the clique $K(x)$ it represents has at least one disk that intersects with y. (Notice that the representative x itself might not intersect y.) Furthermore, we will show that every neighbor of y in G is in a clique represented by some disk in $\rho(y)$. Let $\Gamma(\hat{D})$ be the graph obtained from $\hat{D}$ by first taking the intersection graph of $\hat{D}$ and, for every regular disk y, adding an edge (y, x) for every $x \in \rho(y)$. (Intuitively, we pretend as if the representative x itself intersects y instead of the clique $K(x)$.) We call $\Gamma(\hat{D})$ the *extended intersection graph* of $\hat{D}$. We will ensure that $\Gamma(\hat{D})$ is a connected graph. Note that we will not explicitly maintain $\Gamma(\hat{D})$ as it could have super-linear many edges, where as our goal is near-linear time. Initially, $\hat{D} = D$ and $\Gamma(\hat{D}) = G$, and all disks in $\hat{D}$ are regular disks.

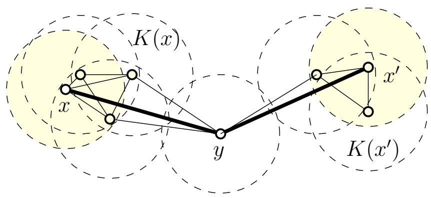

Figure 2 The extended intersection graph $\Gamma(\hat{D})$. The edges connecting a regular disk y and representative disks in $\rho(y)$ are shown in solid edges.

We then apply Lemma 18 to construct a well-separated clique-based separator $\hat{\mathcal{S}}$ for $\hat{D}$. If there is a representative disk x in $\hat{D}$ contained in a clique in $\hat{\mathcal{S}}$, we split x out of the clique and consider x an independent clique in $\hat{\mathcal{S}}$. We then add new cliques in $\hat{\mathcal{S}}$ to $\mathcal{C}$.

Let $R(\hat{\mathcal{S}})$ be the set of representative disks of cliques in $\hat{\mathcal{S}}$. We partition $\hat{D} \setminus \hat{\mathcal{S}}$ into two set of disks D_1, D_2 each contains at most $2|\hat{D}|/3$ disks. For each D_i, we construct a spanning forest F of the extended intersection graph $\Gamma(D_i \cup R(\hat{\mathcal{S}}))$. For each connected component, say T of F, if T has at most r vertices, we then form a cluster R_T containing all regular and representative disks of T, and all disks in the clique of the representative disks of T, and add R_T to $\mathcal{R}$. Otherwise, T has at least r vertices, we recurse on the set of disks, say $\hat{D}_i$, corresponding to vertices of T. The extended intersection graph of $\hat{D}_i$ will be connected.

5 Open Problems

The obvious open question is whether a truly subquadratic time algorithm can be found for the *exact* diameter, thus resolving the long-standing open question. Another open problem is whether the results can be extended to the intersection graph of disks of possibly different radii. The challenge there is to develop something similar to an r-division. While the clique-based separator by de Berg [3] works for general disk graphs, there are challenges applying the separator (or some other variants) recursively to get a nice subdivision with bounded boundary size per piece.

References

1 A. Karim Abu-Affash, Paz Carmi, Anil Maheshwari, Pat Morin, Michiel Smid, and Shakhar Smorodinsky. Approximating maximum diameter-bounded subgraph in unit disk graphs. *Discrete Comput. Geom.*, 66(4):1401–1414, December 2021.

2 Noga Alon, Graham Brightwell, H. A. Kierstead, A. V. Kostochka, and Peter Winkler. Dominating sets in k-majority tournaments. *J. Combin. Theory Ser. B*, 96(3):374–387, May 2006.

3 Mark de Berg. A note on reachability and distance oracles for transmission graphs. *Computing in Geometry and Topology*, page Vol. 2 No. 1 (2023), 2023. doi:10.57717/CGT.V2I1.25.

4 Mark de Berg, Hans L. Bodlaender, Sándor Kisfaludi-Bak, Dániel Marx, and Tom C. van der Zanden. A framework for exponential-time-hypothesis–tight algorithms and lower bounds in geometric intersection graphs. *SIAM Journal on Computing*, 49(6):1291–1331, 2020. doi:10.1137/20m1320870.

5 Mark de Berg, Sándor Kisfaludi-Bak, Morteza Monemizadeh, and Leonidas Theocharous. Clique-based separators for geometric intersection graphs. *Algorithmica*, 85(6):1652–1678, 2023.

6 Nicolas Bousquet, Aurélie Lagoutte, Zhentao Li, Aline Parreau, and Stéphan Thomassé. Identifying codes in hereditary classes of graphs and VC-dimension. *SIAM Journal on Discrete Mathematics*, 29(4):2047–2064, 2015. doi:10.1137/14097879X.

7 Nicolas Bousquet and Stéphan Thomassé. VC-dimension and Erdős–Pósa property. *Discrete Math.*, 338(12):2302–2317, December 2015.

8 Heinz Breu. *Algorithmic Aspects of Constrained Unit Disk Graphs*. PhD thesis, University of British Columbia, April 1996.

9 Karl Bringmann, Sándor Kisfaludi-Bak, Marvin Künnemann, André Nusser, and Zahra Parsaeian. Towards sub-quadratic diameter computation in geometric intersection graphs. In Xavier Goaoc and Michael Kerber, editors, *38th International Symposium on Computational Geometry (SoCG 2022)*, pages 21:1–21:16. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, June 2022.

10 Sarit Buzaglo, Rom Pinchasi, and Günter Rote. Topological hypergraphs. *Thirty Essays on Geometric Graph Theory*, pages 71–81, 2013.

11 Sergio Cabello and Miha Jejčič. Shortest paths in intersection graphs of unit disks. *Comput. Geom.*, 48(4):360–367, May 2015.

12 Timothy M. Chan and Dimitrios Skrepetos. All-pairs shortest paths in unit-disk graphs in slightly subquadratic time. In *LIPIcs-Leibniz International Proceedings in Informatics*, volume 64, 2016.

13 Timothy M. Chan and Dimitrios Skrepetos. All-Pairs shortest paths in geometric intersection graphs. In *Algorithms and Data Structures*, pages 253–264. Springer International Publishing, 2017.

14 Timothy M. Chan and Dimitrios Skrepetos. Approximate shortest paths and distance oracles in weighted Unit-Disk graphs. *J. Comput. Geom.*, 10(2), 2019.

15 Bernard Chazelle and Emo Welzl. Quasi-optimal range searching in spaces of finite VC-dimension. *Discrete & Computational Geometry*, 4(5):467–489, October 1989. doi:10.1007/BF02187743.

16 Victor Chepoi, Bertrand Estellon, and Yann Vaxès. Covering planar graphs with a fixed number of balls. *Discrete Comput. Geom.*, 37(2):237–244, February 2007.

17 Guillaume Ducoffe, Michel Habib, and Laurent Viennot. Diameter, eccentricities and distance oracle computations on h-minor free graphs and graphs of bounded (distance) vapnik–chervonenkis dimension. *SIAM Journal on Computing*, 51(5):1506–1534, 2022.

18 Lech Duraj, Filip Konieczny, and Krzysztof Potępa. Better diameter algorithms for bounded VC-dimension graphs and geometric intersection graphs, 2023. arXiv:2307.08162.

19 Alon Efrat, Alon Itai, and Matthew J Katz. Geometry helps in bottleneck matching and related problems. *Algorithmica*, 31(1):1–28, September 2001.

20 Greg N. Frederickson. Fast algorithms for shortest paths in planar graphs, with applications. *SIAM Journal on Computing*, 16(6):1004–1022, 1987. doi:10.1137/0216064.

21 Viktor Fredslund-Hansen, Shay Mozes, and Christian Wulff-Nilsen. Truly subquadratic exact distance oracles with constant query time for planar graphs. In *32nd International Symposium on Algorithms and Computation (ISAAC 2021)*, Leibniz International Proceedings in Informatics (LIPIcs), pages 25:1–25:12, 2021. doi:10.4230/LIPIcs.ISAAC.2021.25.

22 Jie Gao and Li Zhang. Well-separated pair decomposition for the unit-disk graph metric and its applications. *SIAM Journal on Computing*, 35(1):151–169, 2005. doi:10.1137/S0097539703436357.

23 Sariel Har-Peled and Kent Quanrud. Approximation algorithms for polynomial-expansion and low-density graphs. *SIAM Journal on Computing*, 46(6):1712–1744, 2017. doi:10.1137/16m1079336.

24 David Haussler and Emo Welzl. Epsilon-nets and simplex range queries. In *Proceedings of the second annual symposium on Computational geometry*, SCG '86, pages 61–71, New York, NY, USA, August 1986. Association for Computing Machinery.

25 Harry B. Hunt III, Madhav V. Marathe, Venkatesh Radhakrishnan, Shankar S. Ravi, Daniel J. Rosenkrantz, and Richard E. Stearns. NC-Approximation schemes for NP- and PSPACE-Hard problems for geometric graphs. *J. Algorithm. Comput. Technol.*, 26(2):238–274, February 1998.

26 Haim Kaplan, Matthew J. Katz, Rachel Saban, and Micha Sharir. The Unweighted and Weighted Reverse Shortest Path Problem for Disk Graphs. In *31st Annual European Symposium on Algorithms (ESA 2023)*, pages 67:1–67:14, 2023. doi:10.4230/LIPIcs.ESA.2023.67.

27 Katharina Klost. An algorithmic framework for the single source shortest path problem with applications to disk graphs. *Comput. Geom.*, 111:101979, April 2023.

28 Hung Le and Christian Wulff-Nilsen. VC set systems in minor-free (di)graphs and applications. arXiv 2304.01790, 2023.

29 Jason Li and Merav Parter. Planar diameter via metric compression. In *Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing*, STOC 2019, pages 152–163, New York, NY, USA, June 2019. Association for Computing Machinery.

30 Jiří Matoušek, Raimund Seidel, and Emo Welzl. How to net a lot with little: small ϵ-nets for disks and halfspaces. In *Proceedings of the sixth annual symposium on Computational geometry*, SCG '90, pages 16–22, New York, NY, USA, May 1990. Association for Computing Machinery.

31 Tim Nieberg, Johann Hurink, and Walter Kern. A robust PTAS for maximum weight independent sets in unit disk graphs. In *Graph-Theoretic Concepts in Computer Science*, pages 214–221. Springer Berlin Heidelberg, 2005.

32 Ryan Williams. A new algorithm for optimal 2-constraint satisfaction and its implications. *Theoretical Computer Science*, 348(2-3):357–365, 2005.

33 Christian Wulff-Nilsen. Separator theorems for minor-free and shallow minor-free graphs with applications. In *2011 IEEE 52nd Annual Symposium on Foundations of Computer Science*, pages 37–46, 2011. doi:10.1109/FOCS.2011.15.