Finding a Maximum Clique in a Disk Graph

Jared Espenant
Department of Computer Science, University of Saskatchewan, Saskatoon, Saskatchewan, Canada

J. Mark Keil
Department of Computer Science, University of Saskatchewan, Saskatoon, Saskatchewan, Canada

Debajyoti Mondal[1]
Department of Computer Science, University of Saskatchewan, Saskatoon, Saskatchewan, Canada

Abstract

A disk graph is an intersection graph of disks in the Euclidean plane, where the disks correspond to the vertices of the graph and a pair of vertices are adjacent if and only if their corresponding disks intersect. The problem of determining the time complexity of computing a maximum clique in a disk graph is a long-standing open question that has been very well studied in the literature. The problem is known to be open even when the radii of all the disks are in the interval $[1, (1+\varepsilon)]$, where $\varepsilon > 0$. If all the disks are unit disks then there exists an $O(n^3 \log n)$-time algorithm to compute a maximum clique, which is the best-known running time for over a decade. Although the problem of computing a maximum clique in a disk graph remains open, it is known to be APX-hard for the intersection graphs of many other convex objects such as intersection graphs of ellipses, triangles, and a combination of unit disks and axis-parallel rectangles. Here we obtain the following results.

- We give an algorithm to compute a maximum clique in a unit disk graph in $O(n^{2.5} \log n)$-time, which improves the previously best known running time of $O(n^3 \log n)$ [Eppstein '09].
- We extend a widely used "co-2-subdivision approach" to prove that computing a maximum clique in a combination of unit disks and axis-parallel rectangles is NP-hard to approximate within $4448/4449 \approx 0.9997$. The use of a "co-2-subdivision approach" was previously thought to be unlikely in this setting [Bonnet et al. '20]. Our result improves the previously known inapproximability factor of $7633010347/7633010348 \approx 0.9999$.
- We show that the parameter minimum lens width of the disk arrangement may be used to make progress in the case when disk radii are in $[1, (1+\varepsilon)]$. For example, if the minimum lens width is at least 0.265 and $\varepsilon \leq 0.0001$, which still allows for non-Helly triples in the arrangement, then one can find a maximum clique in polynomial time.

2012 ACM Subject Classification Theory of computation → Computational geometry

Keywords and phrases Maximum clique, Disk graph, Time complexity, APX-hardness

Digital Object Identifier 10.4230/LIPIcs.SoCG.2023.30

Related Version *Full Version*: `https://arxiv.org/abs/2303.07645` [15]

Funding The work is supported in part by the Natural Sciences and Engineering Research Council of Canada (NSERC).

1 Introduction

An *intersection graph* of a set S of geometric objects is a graph where each object in S corresponds to a vertex in G and two vertices in G are adjacent if and only if the corresponding objects intersect. A set of vertices $C \subseteq V$ is called a *clique* if they are mutually adjacent. In this paper, we are interested in the problem of finding a *maximum clique*, i.e., a largest set of mutually adjacent vertices. We mainly focus on *disk graphs*, i.e., the intersection graphs of disks in $\mathbb{R}^2$ (Figure 1). Disk graphs are often used to model ad-hoc wireless networks [22].

[1] Corresponding author

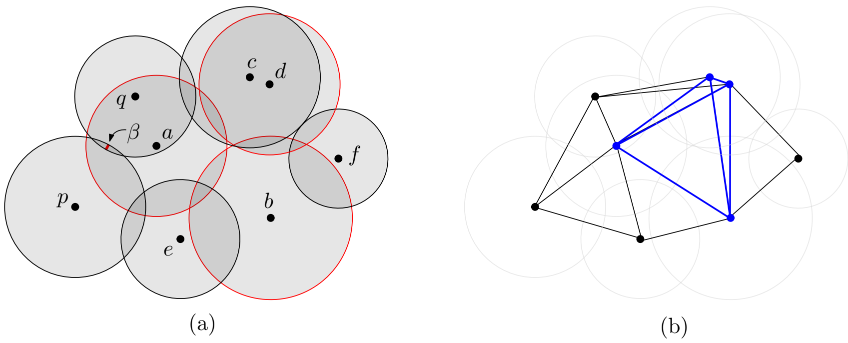

Figure 1 (a) An arrangement of disks $\mathcal{A}$. The minimum width β over all the lenses is determined by the disks centered at p and q. A non-Helly triple is shown in red. (b) A disk graph corresponding to $\mathcal{A}$, where a maximum clique is shown in blue.

The time complexity question for finding a maximum clique in a disk graph is known to be open for over two decades [3, 4, 17]. The question is open even in severely restricted setting such as when the radii of the disks are of two types [9] or when the disk radii are in the interval $[1, 1+\varepsilon]$ for a fixed $\varepsilon > 0$ [5]. However, there exists randomized EPTAS, deterministic PTAS, and subexponential-time algorithms for computing a maximum clique in arbitrary disk graphs [5, 6]. For unit disk graphs, i.e., when all the radii are the same, Clark et al. [13] showed that a maximum clique can be found in $O(n^{4.5})$-time. Their algorithm searches for a maximum clique over all the lenses of pairwise intersecting disks. Later, Eppstein [14] showed how the algorithm could be implemented in $O(n^3 \log n)$-time by searching through a careful ordering of the lenses and using a data structure of [2] to maintain a maximum clique throughout the search. Faster algorithms are known in constrained settings where the centers of the disks lie within a narrow horizontal strip [8]. Polynomial-time algorithms exist for many other intersection graph classes such as for circle graphs [29], trapezoid graphs [16], circle trapezoid graphs [16], intersection graphs of axis-parallel rectangles [23], and so on.

Although the maximum clique problem is open for disk graphs, a number of APX-hardness results are known in the literature for intersection graphs of other geometric objects. A common approach to prove the NP-hardness result for computing a maximum clique in an intersection graph class $\mathcal{I}$ is to take a *co-$2k$-subdivision approach*, as follows. A *$2k$-subdivision*, where k is a positive integer, of a graph G is obtained by replacing each edge (u, v) of G with a path $(u, d_1, \ldots, d_{2k}, v)$ of $2k$ division vertices. A co-$2k$-subdivision approach takes a graph class for which finding a maximum independent set is NP-hard and shows that the complement graph of its $2k$-subdivision has an intersection representation in class $\mathcal{I}$. Since the NP-hardness of computing a maximum independent set is preserved by the even subdivision [12] and since a maximum independent set in a graph corresponds to a maximum clique in the complement graph, this establishes the NP-hardness result for computing a maximum clique in class $\mathcal{I}$. Some of the intersection graph classes for which the maximum clique problem has been proved to be APX-hard using the co-$2k$-subdivision approach are intersection graphs of ellipses [3], triangles [3], string graphs [27], grounded string graphs [25], and so on. Cabello [10] used the co-$2k$-subdivision approach to prove the NP-hardness of computing a maximum clique in the intersection graph of rays, which settled a 21-year-old open problem posed by Kratochvíl and Nešetřil [26]. To the best of our knowledge, no hardness of approximation result is known for this graph class.

Since there is strong evidence that a co-$2k$-subdivision approach may not be sufficient to prove the NP-hardness of computing a maximum clique in a disk graph [6], Bonnet et al. [7] attempted to explore alternative approaches. While they were not able to prove the

NP-hardness for disk graphs, they showed that the problem of computing a maximum clique in an intersection graph that contains both unit disks and axis-parallel rectangles is not approximable within a factor of 7633010347/7633010348 in polynomial time, unless P=NP. This result is interesting since the maximum clique problem is polynomial-time solvable when all objects are either unit disks [13] or axis-parallel rectangles [23]. To obtain this result, Bonnet et al. [7] introduced a new problem called "Max Interval Permutation Avoidance", proved it to be APX-hard, and reduced it to the problem of computing a maximum clique in a combination of unit disks and axis-parallel rectangles. Furthermore, they stated that the intersection graph of unit disks and axis-parallel rectangles is "a class for which the co-2-subdivision approach does not seem to work".

Our Contribution

In this paper we make significant progress on the maximum clique problem for unit disk graphs, disk graphs with disk radii lying in the interval $[1, 1+\varepsilon]$, and intersection graphs of unit disks and axis-parallel rectangles.

Unit disk graph. We give an algorithm to compute a maximum clique in a unit disk graph in $O(n^{2.5} \log n)$-time, which improves the previously best known running time of $O(n^3 \log n)$ [14]. Our algorithm is based on a divide-and-conquer approach that, unlike the previous algorithms that search a clique over all the lenses, shows how to efficiently merge solutions to the subproblems to achieve a faster time complexity. Such techniques have previously been used to accelerate computation for other computational geometry problems, e.g., when finding a closest pair in a point set [28], but appeared to be highly non-trivial while adapting it for the unit disk graph setting.

Intersection graph of unit disks and axis-parallel rectangles. We extend the co-2-subdivision approach to prove a $(4448/4449 \approx 0.9997)$-inapproximability result for computing a maximum clique in an intersection graph that contains both unit disks and axis-parallel rectangles, and thus improve the previously known inapproximability factor of $7633010347/7633010348 \approx 0.9999$ [7]. Note that the use of a co-2-subdivision approach was previously thought to be unlikely in this setting by Bonnet et al. [7]. The key idea behind our NP-hardness reduction is to show that every Hamiltonian cubic graph admits a well-behaved edge orientation and vertex labeling, i.e., its vertices can be labeled and the edges can be oriented such that every vertex has two outgoing or two incoming edges where the labels of these corresponding neighbors are consecutive. While such orientation and labeling are of independent interest, they allow us to represent the complement of the 2-subdivision of a Hamiltonian cubic graph using a combination of unit disks and axis-parallel rectangles.

(ε, β)-disk graph. In an attempt to make progress on the case when the disk radii are in the interval $[1, 1+\varepsilon]$, we introduce (ε, β)-disk graphs. A (ε, β)-*disk graph*, where ε and β are positive constants, is a disk graph where the radii of the disks are in the interval $[1, 1+\varepsilon]$ and every lens is of width at least β. The parameter β can be thought of as the minimum width over all the lenses in the disk arrangement, where a *lens* is the convex intersection region of a pair of disks (Figure 1(a)). We show that the parameter β, i.e., the minimum lens width of the disk arrangement, may be used to make progress in the case when disk radii are in $[1, (1+\varepsilon)]$. For example, if the minimum lens width is at least 0.265, then one can find a maximum clique for $\varepsilon \leq 0.0001$ in polynomial time.

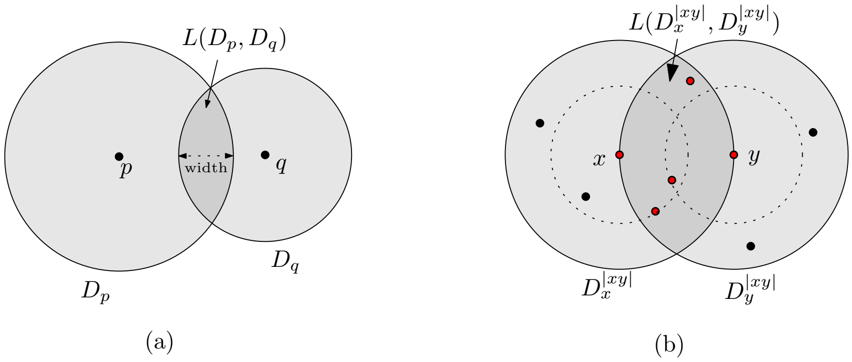

Figure 2 (a) Illustration for a lens. (b) Computation of a maximum clique, where the centers of the disks are shown in dots. The unit disks centered at x and y are shown in dotted circles. The centers inside $L(D_x^{|xy|}, D_y^{|xy|})$ are shown in red.

The existence of *non-Helly triple* in a disk arrangement, i.e., three pairwise intersecting disks without any common point of intersection (Figure 1(a)), typically makes the problem of finding a clique challenging (see Sec. 5). Since β is a lower bound on the width of every lens, a natural question is whether our choice for $\beta \geq 0.265$ already forbids the existence of non-Helly triples. We note that our choice for β still allows for non-Helly triples, and thus the result is non-trivial. We show that the lower bound on β could be leveraged to find for each non-Helly triple, a maximum clique that includes this triple. This extends the prior approach of finding a maximum clique in a unit disk graph that searches over all the pairwise intersecting disks [13] to a more general setting where the disk radii are in $[1, 1.0001]$. We believe that our proposed approach is interesting from the perspective of finding a way to make progress beyond unit disks even though the lower bound on β is large and the gain on ε is small.

2 Preliminaries

By D_q^r we denote a disk with radius r and center q. For the simplicity of the presentation, sometimes we omit the radius and simply use D_q to denote a disk with center q. Let D_p and D_q be a pair of disks. By $L(D_p, D_q)$ we denote the lens (i.e. the intersection region) of these disks (Figure 2(a)). For a line segment ab, we denote by $|ab|$ the length of the segment or the Euclidean distance between the points a and b. The *width* of a lens $L(D_p, D_q)$ is the length of the line segment determined by the intersection of pq and $L(D_p, D_q)$.

Let G be a graph. The *complement graph* $\overline{G}$ of G is a graph on the same set of vertices where $\overline{G}$ contains an edge if and only if it does not appear in G. A set S of vertices in G is called *independent* if no two vertices in S are adjacent in G. A *maximum independent set* $\alpha(G)$ is an independent set of largest cardinality. G is called *bipartite* if its vertices can be partitioned into two independent sets. G is called a *cobipartite graph* if the complement graph of G is a bipartite graph. G is called *cubic* if every vertex of G is of degree three. G is *Hamiltonian* if it has a cycle that contains each vertex of G exactly once.

3 Unit Disk Graph (UDG)

In this section we provide an $O(n^{2.5} \log n)$-time algorithm to compute a maximum clique in a unit disk graph, where a geometric representation of the graph is given as an input.

Clark et al. [13] gave an $O(n^{4.5})$-time algorithm to compute a maximum clique in a unit disk graph. The idea of the algorithm is as follows. For each edge (x, y) of the graph, consider two disks D_x and D_y such that their boundaries pass through y and x, respectively. Let S be the set of unit disks with centers in $L(D_x^{|xy|}, D_y^{|xy|})$. Clark et al. showed that the subgraph of G induced by the vertices corresponding to S is a cobipartite graph $G(S)$. One can thus find a maximum clique in $G(S)$ by computing a maximum independent set in the bipartite graph $\overline{G(S)}$. If (x, y) is the longest edge of a maximum clique M in G, then S must include all the centers of the disks in M and $G(S)$ will contain the largest clique in G. Therefore, one can try the above strategy over all edges and find a maximum clique in G. Since G contains $O(n^2)$ edges and since a maximum independent set in a bipartite graph can be computed in $O(n^{2.5})$ time by leveraging a maximum matching [21], the running time becomes $O(n^{4.5})$.

Breu [8] observed that Clark et al.'s approach [13] to find a maximum clique can be implemented in $O(n^{3.5} \log n)$ time using a result of Aggarwal et al. [2]. Specifically, Aggarwal et al. [2] showed how to compute a maximum independent set in $\overline{G(S)}$ in $O(n^{1.5} \log n)$ time using a data structure of [20, 24], and hence over $O(n^2)$ lenses the running time becomes $O(n^{3.5} \log n)$.

Eppstein [14] observed that while searching through the lenses, instead of computing the maximum independent set from scratch, one can exploit geometric properties to efficiently update and maintain a maximum independent set as follows. For a unit disk center p, let q be a point on the plane such that $|pq| = 2$. Consider two disks D_p and D_q such that their boundaries pass through q and p, respectively. One can now rotate the lens $L(D_p^{|pq|}, D_q^{|pq|})$ around p and update the maximum independent set in the graph corresponding to $L(D_p^{|pq|}, D_q^{|pq|})$ each time a point (i.e., a center of a unit disk) enters or exists from the lens. An update can be processed by an alternating path search in $O(n \log n)$ time [2]. Since the number of changes to $L(D_p^{|pq|}, D_q^{|pq|})$ is bounded by $O(n)$, the time spent for p is $O(n^2 \log n)$. Hence the overall running time is $O(n^3 \log n)$.

3.1 Idea of Our Algorithm

Let G be a disk graph with n vertices, where each disk is of radius r. Let P be the set of centers of the disks corresponding to the vertices of G. To find a maximum clique we take a divide-and-conquer approach as follows.

We rotate the plane so that no two points are in the same vertical or horizontal line. It is straightforward to perform such a rotation in $O(n^2)$ time. We sort the points in P with respect to their x-coordinates and find a vertical line V through a median x-coordinate such that at most $\lceil n/2 \rceil$ points of P are on each half-plane of V. Let P_l and P_r be the points on the closed left halfplane and closed right halfplane of V, respectively. We will find a maximum clique in P_l and P_r recursively.

Let M be a maximum clique in G. If the set of disk centers corresponding to M is a subset of either P_l or P_r, then such a clique must be returned as a solution to one of these two subproblems. Otherwise, each of P_l and P_r contains some points of M. To tackle such a case, it suffices to find a maximum clique in the vertical slab between the vertical lines V_l and V_r, where V_l and V_r are $2r$ units apart from V on the left halfplane and right halfplane, respectively. Let $Q \subseteq P$ be the set of points in the vertical slab. Then the maximum clique of G is the maximum clique found over the disks corresponding to the sets P_l, P_r and Q.

Let $T(n)$ be the time to compute a maximum clique in G. Let $F(n)$ be the time to compute a maximum clique in the vertical slab. Then $T(n)$ is defined as follows.

$$T(n) = 2T\left(\frac{n}{2}\right) + F(n). \quad (1)$$

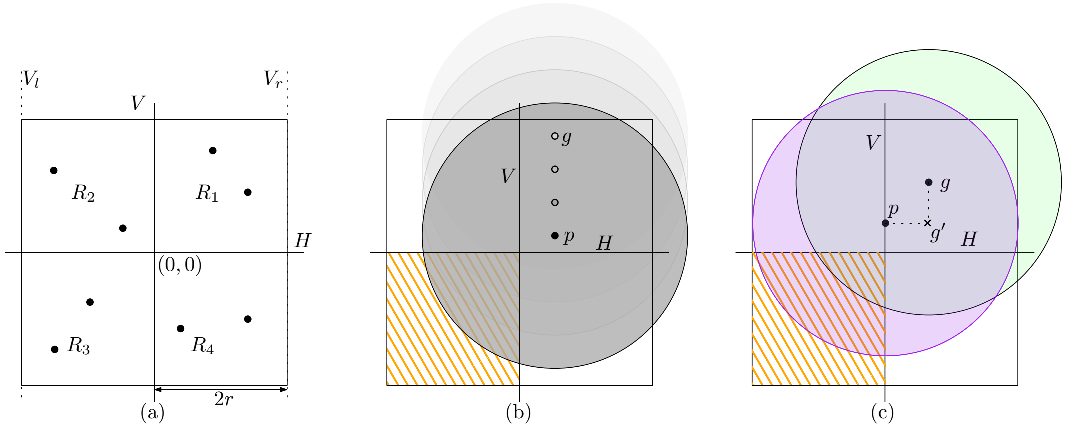

Figure 3 (a) The square S with the disk centers in black dots. (b)–(c) Illustration for Remark 2.

We now sort the points of Q with respect to their y-coordinates and find a horizontal line H through the median y-coordinate such that at most $\lceil |Q|/2 \rceil$ points of Q are on each half-plane of H. Let Q_t and Q_b be the points on the closed top halfplane and closed bottom halfplane of H, respectively. We now find a maximum clique in Q_t and Q_b recursively. If the set of disk centers corresponding to M is a subset of either Q_t or Q_b, then such a clique must be returned as a solution to one of these two subproblems. Otherwise, each of Q_t and Q_b contains some points of M. It now suffices to find a maximum clique in the square S of side length $4r$ with its center located at the intersection point of V and H (Figure 3(a)). Let $B(n)$ be the time to compute the maximum clique in S. Then $F(n)$ is defined as follows.

$$F(n) = 2F\left(\frac{n}{2}\right) + B(n). \quad (2)$$

In the following, we will show that a maximum clique in S can be computed in $O(n^{2.5} \log n)$ time. Consequently, $B(n) \in O(n^{2.5} \log n)$ and by Equation 2 and master theorem, $F(n) \in O(n^{2.5} \log n)$. Consequently, the time complexity determined by Equation 1 is $O(n^{2.5} \log n)$. Note that computing a maximum clique in the square S of side length $4r$ appears to be the bottleneck of our algorithm.

3.2 Computing a Maximum Clique in the Square S

Let M be a maximum clique in G and let C be the centers of the disks in M. Assume that $C \not\subseteq P_l$, $C \not\subseteq P_r$, $C \not\subseteq Q_t$ and $C \not\subseteq Q_b$, i.e., C is a subset of the points in S. We now show how to find M. Let o be the center of S, i.e., the intersection point of H and V. Without loss of generality assume that o is at $(0, 0)$. Let R_i, where $1 \leq i \leq 4$, be the region determined by the intersection of the ith quadrant and S (Figure 3(a)). We now give two remarks. Remark 1 follows directly from our assumption that $C \not\subseteq P_l$, $C \not\subseteq P_r$, $C \not\subseteq Q_t$ and $C \not\subseteq Q_b$.

▶ Remark 1. C must satisfy at least one of the following two conditions. (a) R_1 and R_3 each contains a point from C. (b) R_2 and R_4 each contains a point from C.

▶ Remark 2. Let p and g be two points inside R_1 where the x- and y-coordinates of g are at least as large as that of p. Let D_p^{2r} and D_g^{2r} be two disks of the same radius $2r$ centered at p and g, respectively. Then $(R_3 \cap D_g^{2r}) \subseteq (R_3 \cap D_p^{2r})$.

Proof. Consider first the case when g and p lie on the same vertical line (Figure 3(b)). Note that the interval $(H \cap D_g^{2r})$ increases as we move the center g vertically downward and the interval reaches the maximum when g hits H. Therefore, $(H \cap D_g^{2r}) \subseteq (H \cap D_p^{2r})$. Since g is vertically above p and both have the same radius, $(R_3 \cap D_g^{2r}) \subseteq (R_3 \cap D_p^{2r})$. The argument when g and p lie on the same horizontal line is symmetric.

Consider now the case when x- and y- coordinates of g are larger than that of p (Figure 3(c)). We can move D_g^{2r} vertically down to reach a point g' that has the same y-coordinate as that of p. Consequently, $(R_3 \cap D_g^{2r}) \subseteq (R_3 \cap D_{g'}^{2r})$. Finally, we move $D_{g'}^{2r}$ towards p. Hence we obtain $(R_3 \cap D_{g'}^{2r}) \subseteq (R_3 \cap D_p^{2r})$. ◀

We are now ready to describe the algorithm. The algorithm considers two cases depending on whether every disk center in C is within a distance of $2r$ from o. It processes each case in $O(n^{2.5} \log n)$ time, and then returns the maximum clique found over the whole process.

The high-level idea for finding a maximum clique is as follows. For the first case, we assume every disk center in C to be within a distance of $2r$ from o. The algorithm makes a guess for the farthest disk center q in C from o and then finds the other disks in the maximum clique by defining a lens that would contain all the disk centers of C. For the second case, we assume that at least one disk center in C has a distance of more than $2r$ from o. The algorithm makes a guess for the first point $p \in C$ in some particular point ordering and then finds the other disks in the maximum clique by defining a lens that would contain all the disk centers of C. We now describe the details.

Case 1 (Every disk center in C is within a distance of $2r$ from o)

Let q be a point of C that has the largest distance from o. Without loss of generality assume that q lies in R_2. We now order the points of R_2 that are within distance $2r$ from o in decreasing order of their distances from o (breaking ties arbitrarily). Figure 4(a) illustrates this order in orange concentric circles. Let σ be the resulting point ordering. We iteratively consider each point in σ to be q and then find a maximum clique as follows.

Let $D_o^{|oq|}$ be a disk centered at $o = (0, 0)$ such that its boundary passes through q (Figure 4(b)). Since q is the furthest point of C from o, every point of C is contained in $D_o^{|oq|}$. Let t be a point in R_4 that lies on the line through o and q at a distance of $2r$ from q.

We now show that every point of C is in $L(D_q^{2r}, D_t^{2r})$. Suppose for a contradiction that there exists a point $e \in C$ which is not in $L(D_q^{2r}, D_t^{2r})$. If e belongs to $S \setminus D_q^{2r}$, then D_e cannot intersect D_q. Therefore, e must lie in the region $D_o^{|oq|} \cap (D_q^{2r} \setminus D_t^{2r})$. Note that q, o and t lie on the same line. Since the boundaries of both $D_o^{|oq|}$ and D_t^{2r} pass through q and $|qt| \geq |qo|$, we have $D_o^{|oq|} \subseteq D_t^{2r}$, and hence the point e cannot exist.

Since the intersection graph induced by the disks with centers in $L(D_q^{2r}, D_t^{2r})$ is a cobipartite graph [13], a maximum clique in this graph can be computed in $O(n^{1.5} \log n)$ time [2]. Over all choices for q in σ, the running time becomes $O(n^{2.5} \log n)$.

Case 2 (There exists a disk center in C with distance more than $2r$ to o)

Without loss of generality assume that R_2 contains a disk center that belongs to C and has a distance of more than $2r$ from o. We now consider the points of R_2 that have a distance of more than $2r$ from o and order them by sweeping a disk as described below. Let o' be the top left corner of R_2. Let D_a^{2r} be a disk of radius $2r$ such that its boundary passes through o' and its center a lies on the line oo' (Figure 5(a)). We now move D_a^{2r} along the line $o'o$

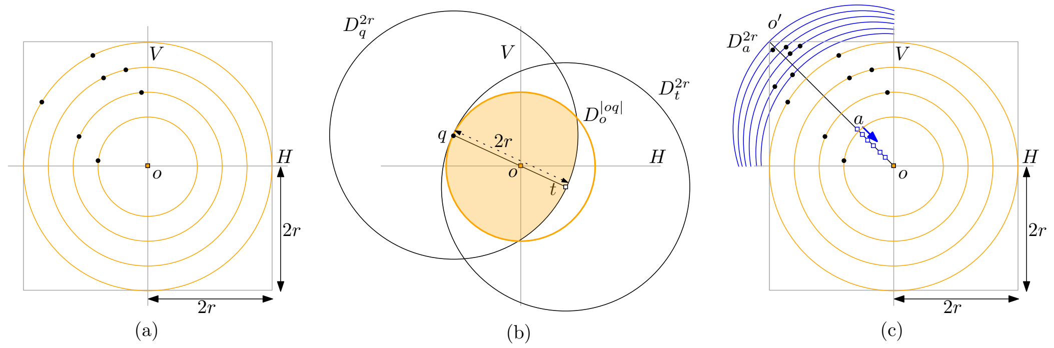

Figure 4 (a) Illustration for point ordering in R_2 in Case 1. (b) Illustration for D_q^{2r} and D_t^{2r}. (c) Point ordering in Case 2.

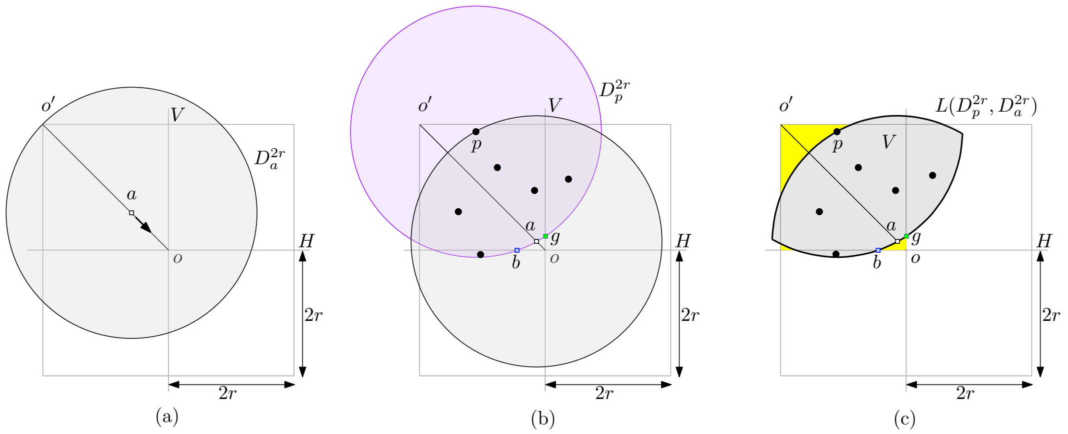

Figure 5 (a)–(b) Illustration for sweeping D_a^{2r}. (c) Illustration for Case A of Lemma 3.

by moving the center a towards o. We order the points in the order D_a^{2r} hits them at its boundary as we move a towards o (breaking ties arbitrarily). Figure 4(c) illustrates this order in blue circular arcs. Let σ' be the resulting point ordering.

Let p be the first point of C in R_2 that is hit by D_a^{2r} at its boundary. Then the boundary of D_p^{2r} passes through a (Figure 5(b)). Let b be the point of intersection between D_p^{2r} and H that is closer to o. Let g be the point of intersection between D_p^{2r} and V that is closer to o.

In the following lemma (Lemma 3), we show that every point of C belongs to the lens $L(D_p^{2r}, D_a^{2r})$. Since the corresponding intersection graph is a cobipartite graph [13], a maximum clique in this graph can be computed in $O(n^{1.5} \log n)$ time [2]. Over all choices for p in σ', the running time becomes $O(n^{2.5} \log n)$.

▶ **Lemma 3.** *Let p be the first point of C in R_2 that is hit by D_a^{2r} at its boundary. Then every point of C belongs to the lens $L(D_p^{2r}, D_a^{2r})$.*

Proof. Suppose for a contradiction that there exists a point $e \in C$ that does not belong to $L(D_p^{2r}, D_a^{2r})$. We now consider the following three subcases and in each case we show that such a point e cannot exist.

Case A ($e \in R_2$): Figure 5(c) highlights the potential locations for e in yellow. If $e \in (D_a^{2r} \setminus D_p^{2r})$, then e fails to intersect p. If $e \in (D_p^{2r} \setminus D_a^{2r})$, then $e \in C$ must be the first point (instead of p) in R_2 that is hit by D_a^{2r}, which leads to a contradiction.

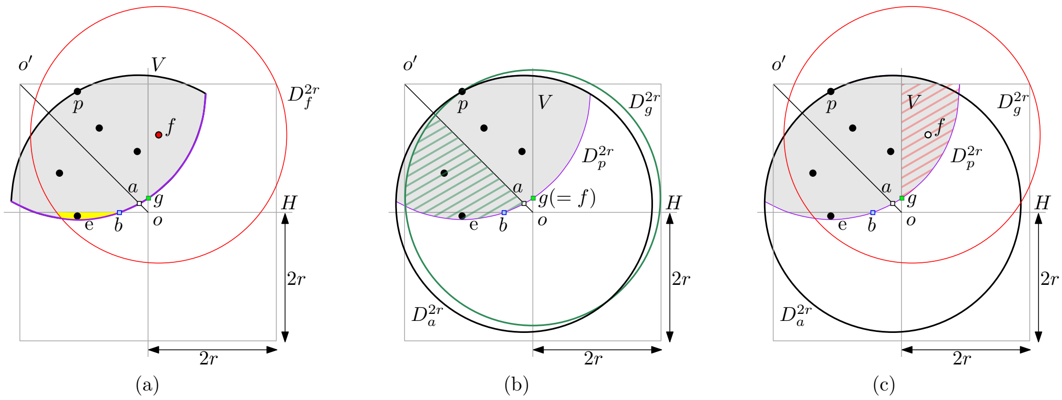

Figure 6 Illustration for Case C of Lemma 3. (a) The boundaries of D_f^{2r} and D_p^{2r} are shown in red and purple, respectively. (b) The scenario when f coincides with g. (c) The scenario when $f \neq g$.

Case B ($e \in R_4$): Since D_p^r and D_e^r intersect, e must lie inside D_p^{2r}. Note that $|po| > 2r$. Since p belongs to C, we have $C \subseteq D_p^{2r}$, and hence, R_4 cannot contain any point of C.

Case C ($e \in R_1$ or $e \in R_3$): Without loss of generality assume that $e \in R_3$. The argument when $e \in R_1$ is symmetric. We have explained in Case B that R_4 cannot contain any point of C. Therefore, by Remark 1, R_1 and R_3 each contains a point from C. Let $f \in C$ be a point in R_1. Then D_e^r must intersect D_f^r. In other words, e must belong to D_f^{2r} (Figure 6(a)). It now suffices to show that the region $R_3 \cap L(D_f^{2r} \cap D_p^{2r})$, which is the potential location for e (shown in yellow), is a subset of $L(D_p^{2r}, D_a^{2r})$.

Consider first the case when f coincides with g. Since g is on the boundary of D_p^{2r}, the boundary of D_g^{2r} passes through p. Note that if we walk along the boundary of D_p^{2r} starting at g clockwise, then we first hit g and then a. If we keep walking then we must hit the boundary of D_g^{2r} before the boundary of D_a^{2r}. Therefore, the region of $L(D_p^{2r}, D_g^{2r})$ on the left halfplane of line oo' (as shown in rising pattern in Figure 6(b)) is a subset of $L(D_p^{2r}, D_a^{2r})$. Consequently, $R_3 \cap L(D_p^{2r}, D_g^{2r})$ is a subset of $L(D_p^{2r}, D_a^{2r})$.

Consider now the case when $f \neq g$ and $f \in (R_1 \cap D_p^{2r})$ (Figure 6(c)). Since the x- and y-coordinates of f are at least as large as that of g, by Remark 2, we obtain $(R_3 \cap D_f^{2r}) \subseteq (R_3 \cap D_g^{2r})$. Together with the argument that $R_3 \cap L(D_p^{2r}, D_g^{2r})$ is a subset of $L(D_p^{2r}, D_a^{2r})$, one can observe that $R_3 \cap L(D_p^{2r}, D_f^{2r})$ is a subset of $L(D_p^{2r}, D_a^{2r})$. ◀

Since a maximum clique in S can be computed in $O(n^{2.5} \log n)$ time, the strategy of Section 3.1 yields a running time of $O(n^{2.5} \log n)$.

▶ **Theorem 4.** *Given a set of n unit disks in the Euclidean plane, a maximum clique in the corresponding disk graph can be computed in $O(n^{2.5} \log n)$ time.*

4 Combination of Unit Disks and Axis-Parallel Rectangles

In this section we show that the maximum clique problem for an intersection graph of unit disks and axis-parallel rectangles is NP-hard to approximate within a factor of $\frac{4448}{4449} \approx 0.9997$. We first show an inapproximability result for computing a maximum independent set and then use this result to prove the APX-hardness for computing a maximum clique.

4.1 Inapproximability of Computing a Maximum Independent Set

The proof of the following theorem is obtained by leveraging an inapproximability result of [11] and a graph transformation technique of [18] (see the full version [15]).

▶ **Theorem 5.** *The problem of computing a maximum independent set in a 2-subdivision of a Hamiltonian cubic graph is NP-hard to approximate within* $\frac{4448}{4449} \approx 0.9997$*, even when a Hamiltonian cycle is given as an input.*

Let G be a Hamiltonian cubic graph with n vertices and let H be a 2-subdivision of G. In Section 4.2 we show that given a Hamiltonian cycle of G, $\overline{H}$ can be represented as an intersection graph of unit disks and axis-parallel rectangles in polynomial time. Since a maximum independent set in H corresponds to a maximum clique in $\overline{H}$ and vice versa, the inapproximability result follows from Theorem 5.

4.2 Representing $\overline{H}$ with Unit Disks and Axis-parallel Rectangles

The number of edges in G is $m = 3n/2$. We first show that the edges of G can be oriented and labeled with distinct positive integers from 1 to $3n/2$ such that each vertex has exactly two of its incident edges with the same orientation and they are labeled with consecutive numbers. We will refer to such labeling as a *pair-oriented labeling.* Figure 7(i) illustrates a pair-oriented labeling of a cubic graph, e.g., v_5 has two incoming edges which are labeled with 2 and 3, and v_7 has two outgoing edges which are labeled with 8 and 9. We will use this labeling to construct the required intersection representation for $\overline{H}$.

▶ **Lemma 6.** *Let G be a Hamiltonian cubic graph with n vertices. Then G admits a pair-oriented labeling. Furthermore, given a Hamiltonian cycle C in G, a pair-oriented labeling for G can be computed in polynomial time.*

Proof (Outline). We first orient the edges of G, as follows. Let $(v_1, v_2, \ldots, v_n)$ be the ordering of the vertices of G on C. For each edge (v_i, v_j), where $i < j$, we orient the edge from v_i to v_j, as illustrated in Figure 7(a).

We now give an incremental construction for the edge labeling. We first find the smallest index k such that v_k has a pair of outgoing edges that are not yet labeled. We now find a *maximal edge sequence* S_k of non-Hamiltonian edges $e_1, e_2, \ldots, e_q$ such that for each i from 1 to $q-1$, there is a Hamiltonian edge that connects the source vertex of e_{i+1} to the target vertex of e_i. Figure 7(c) illustrates such a maximal edge sequence e_1, e_2, where $e_1 = (v_1, v_5)$ and $e_2 = (v_4, v_6)$, and the edge (v_4, v_5) is a Hamiltonian edge. Let ℓ be the largest number that has been used for edge labeling so far. We then label the edges $e_1, e_2, \ldots, e_q$ with $\ell + 2, \ell + 4, \ldots, \ell + 2q$ and the Hamiltonian edges that they nest with $\ell + 1, \ell + 3, \ldots, \ell + (2q + 1)$. Let V_k be the set of vertices that appear on the edges of S_k. It is now straightforward to verify that every vertex of V_k has two edges with the same orientation and these edges are labeled with consecutive numbers. We repeatedly find such maximal edge sequences starting at the vertex with the smallest index that has two outgoing edges that are not yet labeled (Figure 7(a)–(g)). The remaining unlabeled edges are then labeled arbitrarily using remaining integers (Figure 7(h)) and finally, the orientation of the edge (v_1, v_n) is reversed to obtain the required pair-oriented labeling (Figure 7(i)). ◀

Let γ be a pair-oriented labeling of G (Figure 8(a)–(b)). By $\gamma(e)$ we denote the label of an edge e. Let e be an edge of G with source s and target t. We now label the two division vertices corresponding to e in the 2-subdivision H. The division vertex adjacent to s receives

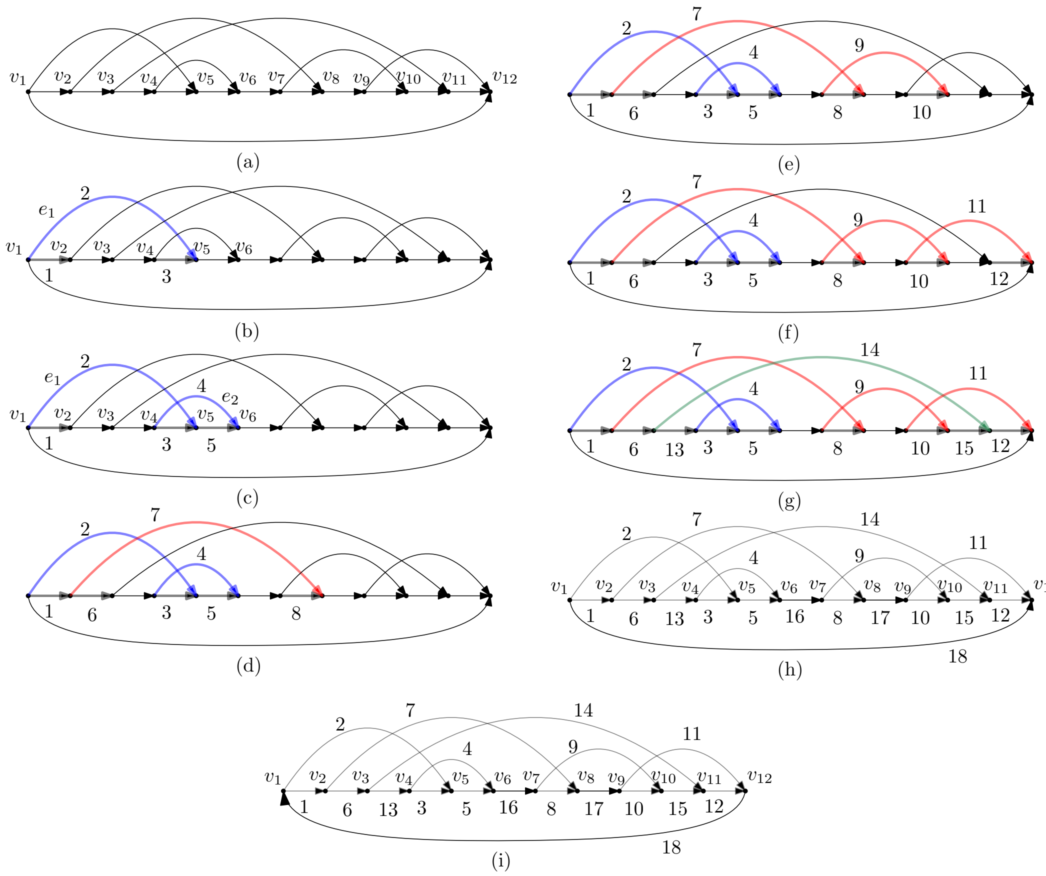

Figure 7 Construction of a pair-oriented labeling of a Hamiltonian cubic graph. The first maximal edge sequence S_1 is shown in blue. S_1 consists of the non-Hamiltonian edges e_1, e_2, and the set V_1 consists of their end vertices, i.e., $\{v_1, v_4, v_5, v_6\}$. The second and third maximal edge sequences are shown in red and green, respectively.

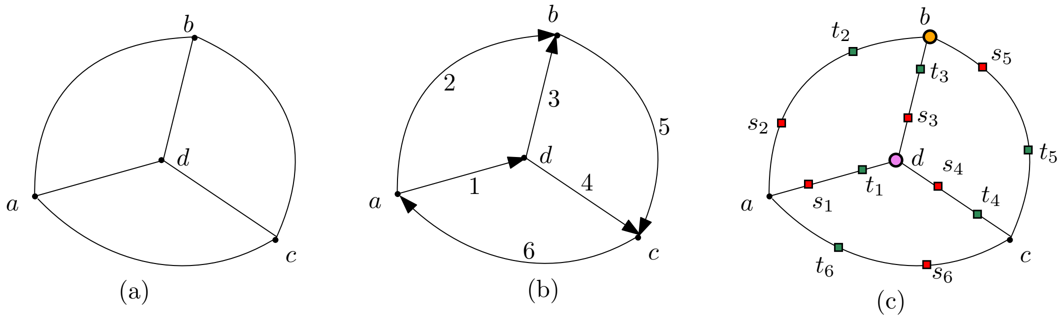

Figure 8 (a)–(b) G, and its pair-oriented labeling. (c) Labeling of the division vertices of H.

the label $s_{\gamma(e)}$ and the division vertex adjacent to t receives the label $t_{\gamma(e)}$. We refer to $s_{\gamma(e)}$ and $t_{\gamma(e)}$ as a *type-s* and *type-t* label, respectively. By the property of γ (Lemma 6), each original vertex v in H is now adjacent to exactly two division vertices of the same type with their indices numbered with consecutive numbers. For example in Figure 8(c), the vertices d and b are adjacent to s_3, s_4 and t_2, t_3, respectively.

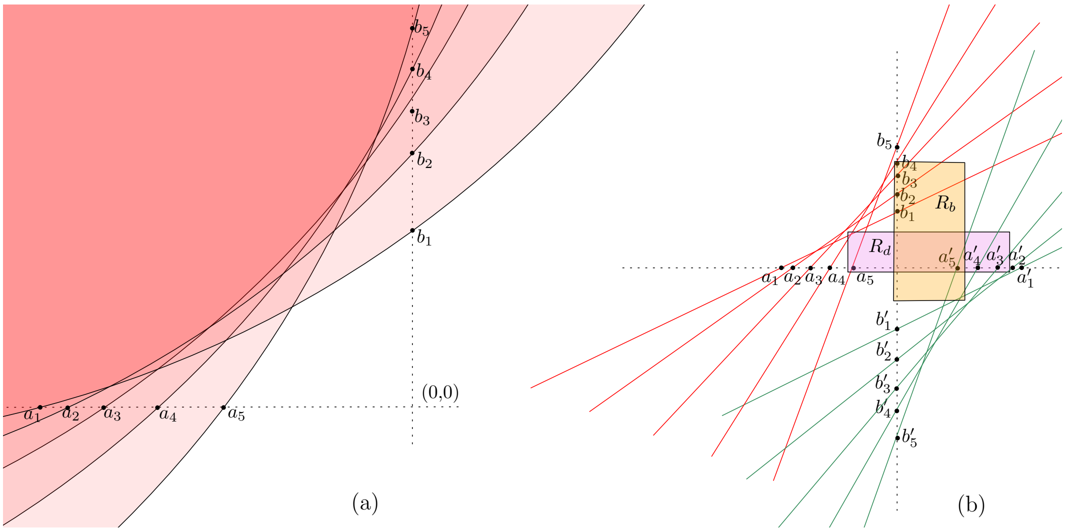

Figure 9 (a) Arrangement of the unit disks corresponding to type-s division vertices. (b) Illustration for the intersection representation of $\overline{H}$. Only a subset of unit disks and the rectangles corresponding to b and d are shown for better readability.

The intersection representation of $\overline{H}$ now follows from the construction of [7]. We briefly describe the construction at a high level for completeness. Consider a set of unit disks C_s for the type-s vertices with centers in the second quadrant such that they intersect the negative x-axis and positive y-axis, but not the positive x-axis or negative y-axis. Furthermore, the ordering of the disks obtained by walking from $(0,0)$ to $(-\infty,0)$ is reversed when walking from $(0,0)$ to $(0,\infty)$ (Figure 9(a)). Let $\mathcal{I}_s$ be the convex region determined by the intersection of all type-s disks. The set of unit disks C_t for type-t vertices is placed on the 4th quadrant symmetrically. Let $\mathcal{I}_t$ be the convex region determined by the intersection of all the type-t disks. For a sufficiently large radius, the disk boundaries appear similar to a set of halfplanes (Figure 9(b)) and each disk in C_s intersects every disk in C_t except for the one with the same label. Therefore, all the intersections between division vertices of $\overline{H}$ are realized. Note that the original vertices of H in $\overline{H}$ form a clique in $\overline{H}$ and each original vertex is adjacent to all but three division vertices in $\overline{H}$. We now represent the original vertices of $\overline{H}$ with rectangles such that all of them enclose the point $(0,0)$. Let b be an original vertex of $\overline{H}$. Without loss of generality assume that b has two type-t neighbors and one type s neighbor (Figure 8(c)). By the property of the pair-oriented labeling, the type-t neighbors are labeled consecutively. Let t_i, t_{i+1}, s_j be the neighbors of b. We now create a rectangle R_b to represent b. We place the top-left corner of R_b near the intersection point of the boundaries of the disks for t_i, t_{i+1} such that R_b does not intersect these disks but intersects all other type-t disks. We place the bottom-right corner of R_b near the circular segment determined by the disk for s_j on $\mathcal{I}_t$ such that R_b does not intersect the disk for s_j but intersects all other type-s disks. We refer to [7] for a formal reduction.

▶ **Theorem 7.** *The problem of computing a maximum clique in an intersection graph of unit disks and axis-parallel rectangles is NP-hard to approximate within a factor of* $\frac{4448}{4449} \approx 0.9997$.

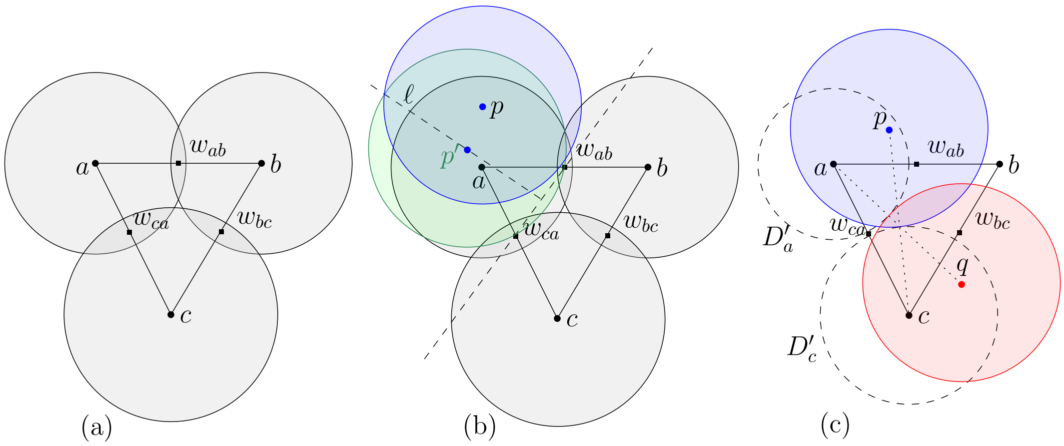

Figure 10 (a) A non-Helly triple $\{D_a, D_b, D_c\}$. Illustration for (b) Case 1 and (c) Case 2.

5 Finding a Maximum Clique in an (ε, β)-disk graph

In this section, we give a polynomial-time algorithm to compute a maximum clique in an (ε, β)-disk graph. By definition, the radii of the disks are in $[1, 1+\varepsilon]$ and every lens is of width at least β. We give an $O(n^4)$-time algorithm when $\beta \geq 0.265$ and $\varepsilon \leq 0.0001$. Although β could be expressed as a function of ε, for simplicity of the presentation, we set specific values to β and ε and often use crude bounds to simplify the arguments. Therefore, we believe one can choose slightly better parameters by using a tedious case analysis.

Let G be an (ε, β)-disk graph. Let M be a set of disks determining a maximum clique in G. In the following, we will use Helly's theorem [19], i.e., for a collection of convex sets in $\mathbb{R}^d$, if the intersection of every $(d+1)$ of these sets is non-empty then the collection must have a non-empty intersection. Recall that for three disks $\{D_a, D_b, D_c\}$, if $(D_a \cap D_b \cap D_c) = \emptyset$, then we call them a *non-Helly triple*. Otherwise, we refer to them as a *Helly triple*.

Consider three unit disks with centers at the corners of an equilateral triangle. If these disks intersect exactly at one point, then the width of each lens is $(2 - 2\cos 30°) \approx 0.2679$, which is larger than β. Therefore, in an (ε, β)-disk graph, we may have non-Helly triples. We now consider two cases depending on whether M contains a non-Helly triple or not.

5.1 M does not contain any non-Helly triple

If M does not contain any non-Helly triple, then by Helly's theorem [19], the disks in M have a non-empty intersection. Let $\mathcal{R}$ be the set of connected regions or cells determined by the arrangement of the disk boundaries. We examine for each cell r in $\mathcal{R}$, the number of disks that contains r, and find a maximum set of mutually intersecting disks. It is straightforward to compute the arrangement of n disks in $O(n^3)$-time (even faster algorithms exist [1]) and each cell can be checked in $O(n)$ time. Hence finding a maximum clique takes $O(n^4)$ time.

5.2 M contains a non-Helly triple

If M contains a non-Helly triple, then let $\{D_a, D_b, D_c\}$ be such a non-Helly triple in M (Figure 10(a)). Let w_{ab}, w_{bc}, w_{ca} be the midpoint of the lenses $L(D_a, D_b)$, $L(D_b, D_c)$ and $L(D_c, D_a)$, respectively. The following two lemmas give some properties corresponding to the non-Helly triple and their proof is included in the full version [15].

▶ **Lemma 8.** *If* $\{D_a, D_b, D_c\}$ *is a non-Helly triple,* $\beta \geq 0.265$*, and* $\varepsilon \leq 0.0001$*, then the lenses* $L(D_a, D_b)$*,* $L(D_b, D_c)$ *and* $L(D_c, D_a)$ *are of width less than 0.275. Furthermore, the interior angles of* Δabc *are in the interval* $[58.024°, 60.988°]$.

▶ **Lemma 9.** *If* $\{D_a, D_b, D_c\}$ *is a non-Helly triple,* $\beta \geq 0.265$*, and* $\varepsilon \leq 0.0001$*, then* Δabc *and* $\Delta w_{ab}w_{bc}w_{ca}$ *satisfy the following properties. (a)* $1.725 \leq |ab|, |bc|, |ca| \leq (1.735+2\varepsilon)$*. (b) For each point* q *in* $\{w_{ab}, w_{bc}, w_{ca}\}$*, the distances from* q *to the center of the two disks containing* q *are in the interval* $[0.8625, (0.8675 + \varepsilon)]$*. (c) The length of each side of* $\Delta w_{ab}w_{bc}w_{ca}$ *is in the interval* $[0.883, 0.887]$.

Let O be the disks in the disk graph representation and let O' be $O \setminus \{D_a, D_b, D_c\}$. We refer to a disk in O' as *type-k*, where $0 \leq k \leq 3$, if it contains exactly k points from $\{w_{ab}, w_{bc}, w_{ca}\}$. In the following we show that for a pair of disks D_p, D_q, if each of them intersects all the disks in $\{D_a, D_b, D_c\}$, then they must mutually intersect. As a consequence, we can find a maximum clique including $\{D_a, D_b, D_c\}$ in $O(n)$ time and a maximum clique over all possible $O(n^3)$ choices of non-Helly triples in $O(n^4)$ time.

Case 1 (At least one of D_p and D_q is of Type-0): We show that this case is trivial because a type-0 disk that intersects all the disks in the non-Helly triple but avoids $\{w_{ab}, w_{bc}, w_{ca}\}$ cannot exist.

Consider the disk D_p. We first show that if p lies inside $\Delta w_{ab}w_{bc}w_{ca}$, then D_p must contain a corner of $\Delta w_{ab}w_{bc}w_{ca}$. By Lemma 9, the maximum side length of $\Delta w_{ab}w_{bc}w_{ca}$ is at most 0.887. Therefore, the circumradius for $\Delta w_{ab}w_{bc}w_{ca}$ is bounded by $\frac{0.887}{\sqrt{3}} \leq 1$. Hence D_p must contain a corner of $\Delta w_{ab}w_{bc}w_{ca}$.

We now show that if p lies outside of $\Delta w_{ab}w_{bc}w_{ca}$, then D_p cannot create a lens of width β with D_a, D_b, D_c. Without loss of generality assume that the left-halfplane of the line through $w_{ab}w_{ca}$ contains p and the right-halfplane contains the centers b, c (Figure 10(b)).

Consider a disk $D_{p'}$ with the same radius as that of D_p such that its center p' lies outside of Δabc and its boundary passes through w_{ab} and w_{ca}. The following lemma gives an upper bound on $|ap'|$ and $\angle acp'$ and its proof is included in the full version [15].

▶ **Lemma 10.** *Let* $D_{p'}$ *be a disk such that the boundary of* $D_{p'}$ *passes through* w_{ab} *and* w_{ca} *and the center* p' *lies outside of* Δabc*. Then* $|ap'| < (0.25 + \varepsilon)$ *and* $\angle acp' \leq 17.5°$.

We now show that $D_{p'}$ cannot create a lens of width β with D_c. Since $|p'w_{ca}|$ is fixed, the distance $|p'c|$ decreases with the increase in $\angle p'cw_{ca}$ and decrease in $|cw_{ca}|$. Since $\angle p'cw_{ca} < 17.5°$ and $|cw_{ca}| \geq 0.8625$, by using basic trigonometry on $\Delta p'cw_{ca}$ one can observe that $|p'c| \geq 1.78 > (2 - \beta)$. Therefore, $D_{p'}$ cannot create a lens of width β with D_c.

Since D_p does not contain w_{ab} and w_{bc}, p lies above or below the bisector ℓ of $w_{ab}w_{ca}$. Consider moving p' to p. Since moving p' above or below ℓ decreases the width of either $L(D_{p'}, D_b)$ or $L(D_{p'}, D_c)$, D_p cannot have a lens of width β with D_b and D_c simultaneously.

Case 2 (D_p and D_q are of Type-1): Without loss of generality assume that D_p and D_q contains w_{ab} and w_{bc}, respectively (Figure 10(c)). Let a_r, c_r, p_r, q_r be the radii of D_a, D_c, D_p, D_q, respectively. It now suffices to show that $|pq| \leq p_r + q_r$, i.e., D_p and D_q must intersect. Note that by the property of (ε, β)-graph, an intersection would imply a lens of width at least β, and hence we only show that $|pq| \leq p_r + q_r$.

Let D'_a and D'_c be the disks obtained by shrinking the radii of D_a and D_c by β. Since the width of the lenses created by the non-Helly triple is less than 0.275, the points w_{ab}, w_{bc}, w_{ca} lie outside of D'_a and D'_c. Since the width of each lens is at least β, D_p must intersect D'_c. Consider a line ℓ through ac with b on its right half-plane.

Consider first the scenario when p and q are on the right half-plane of ℓ. If p is above the line through bc and q is below the line through ab, then aq and pc intersect (Figure 11(a)). Therefore, $|pq| \leq |aq| + |pc| - |ac| \leq (a_r + q_r - 0.265) + (p_r + c_r - 0.265) - (a_r + c_r - 0.275) < (p_r + q_r)$. Otherwise, p, q lie on the right halfplane of the line through $w_{ab}w_{bc}$ in the wedge determined by $\angle abc$ and its opposite angle, as shaded in orange in Figure 11(b). Since $\max\{|w_{ab}w_{bc}|, |bw_{ab}|, |bw_{bc}|\} \leq 1+\varepsilon$, it is straightforward to observe that D_p and D_q intersect.

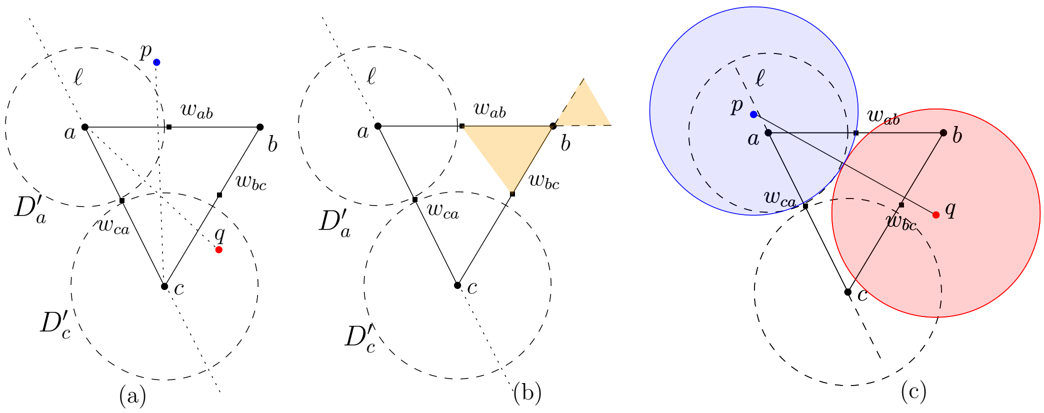

Figure 11 Illustration for the locations of p and q. (a)–(b) Case 1. (c) Case 2.

If p and q are on different sides of ℓ (Figure 11(c)), then without loss of generality assume that p lies on the left half-plane and q lies on the right half-plane. Here we show that $|ap| \leq (0.25 - \varepsilon)$ (see the full version [15] for details). Consequently, $|pq| \leq |ap| + |aq| \leq (0.25 - \varepsilon) + (a_r + q_r - 0.265) = (p_r + q_r) + (a_r - p_r) - \varepsilon - 0.015 < (p_r + q_r)$.

Case 3 (D_p and D_q are of Type-2 or Type-3): Since D_p and D_q each contains at least two points from $\{w_{ab}, w_{bc}, w_{ca}\}$, they must intersect.

Case 4 (One of D_p and D_q is of type-1 and the other is of type-2 or type-3): The case when D_p and D_q contains a common point from $\{w_{ab}, w_{bc}, w_{ca}\}$ is trivial. Therefore, without loss of generality assume that D_p is of type-1 and contains w_{ab}, and D_q is of type-2 and contains w_{bc} and w_{ca}. We use the same setting as in Case 2, i.e., ℓ is the line through ac and b lies on the right half-plane. We now move D_q counter-clockwise without changing the distance of $|pq|$ and stop as soon as w_{ca} hits the boundary of D_q. By an analysis similar to Case 2, we now can observe that $|pq| \leq p_r + q_r$, and hence D_p and D_q must intersect.

▶ **Theorem 11.** *Given a set of n disks in the Euclidean plane such that the width of every lens is at least 0.265 and the radii are in the interval $[1, 1.0001]$, a maximum clique in the corresponding disk graph can be computed in $O(n^4)$ time.*

6 Conclusion and Directions for Future Work

We gave an $O(n^{2.5} \log n)$-time algorithm to compute a maximum clique in a unit disk graph. A natural avenue for future research would be to improve the time complexity of the algorithm. We proved that for the combination of unit disks and axis-parallel rectangles, a maximum clique is NP-hard to approximate within a factor of 4448/4449. We obtained the result using a co-2-subdivision approach, and along the way, we showed that every Hamiltonian cubic graph admits a pair-oriented labeling. It would be interesting to improve the inapproximability factor, and one way to achieve this would be to examine whether pair-oriented labelings exist also for non-Hamiltonian cubic graphs. We showed that if the width of every lens is at least 0.265, then one can find a maximum clique in polynomial time in a more general setting where the disk radii are in $[1, 1.0001]$. We believe that with tedious case analysis, these numbers may be improved slightly, however, it would be challenging to lower β down to 0.2 using the current technique.

References

1 Pankaj K. Agarwal and Micha Sharir. Arrangements and their applications. In Jörg-Rüdiger Sack and Jorge Urrutia, editors, *Handbook of Computational Geometry*, pages 49–119. Elsevier, 2000. doi:10.1016/b978-0-444-82537-7.x5000-1.

2 Alok Aggarwal, Hiroshi Imai, Naoki Katoh, and Subhash Suri. Finding k points with minimum diameter and related problems. *J. Algorithms*, 12(1):38–56, 1991. doi:10.1016/0196-6774(91)90022-Q.

3 Christoph Ambühl and Uli Wagner. The clique problem in intersection graphs of ellipses and triangles. *Theory of Computing Systems*, 38(3):279–292, 2005. doi:10.1007/s00224-005-1141-6.

4 J. Bang-Jensen, B. Reed, M. Schacht, R. Šámal, B. Toft, and U. Wagner. *Topics in Discrete Mathematics, Dedicated to Jarik Nešetřil on the Occasion of his 60th birthday*, volume 26 of *Algorithms and Combinatorics*, pages 613–627. Springer, 2006.

5 Marthe Bonamy, Édouard Bonnet, Nicolas Bousquet, Pierre Charbit, Panos Giannopoulos, Eun Jung Kim, Pawel Rzazewski, Florian Sikora, and Stéphan Thomassé. EPTAS and subexponential algorithm for maximum clique on disk and unit ball graphs. *J. ACM*, 68(2):9:1–9:38, 2021. doi:10.1145/3433160.

6 Édouard Bonnet, Panos Giannopoulos, Eun Jung Kim, Pawel Rzazewski, and Florian Sikora. QPTAS and subexponential algorithm for maximum clique on disk graphs. In Bettina Speckmann and Csaba D. Tóth, editors, *Proceedings of the 34th International Symposium on Computational Geometry (SoCG)*, volume 99 of *LIPIcs*, pages 12:1–12:15. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2018. doi:10.4230/LIPIcs.SoCG.2018.12.

7 Édouard Bonnet, Nicolas Grelier, and Tillmann Miltzow. Maximum Clique in Disk-Like Intersection Graphs. In Nitin Saxena and Sunil Simon, editors, *Proceedings of the 40th IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science (FSTTCS 2020)*, volume 182 of *LIPIcs*, pages 17:1–17:18. Schloss Dagstuhl–Leibniz-Zentrum für Informatik, 2020. doi:10.4230/LIPIcs.FSTTCS.2020.17.

8 Heinz Breu. *Algorithmic aspects of constrained unit disk graphs*. PhD thesis, University of British Columbia, 1996.

9 Sergio Cabello. Maximum clique for disks of two sizes. Open problems from Geometric Intersection Graphs: Problems and Directions, CG Week Workshop, 2015.

10 Sergio Cabello, Jean Cardinal, and Stefan Langerman. The clique problem in ray intersection graphs. *Discret. Comput. Geom.*, 50(3):771–783, 2013. doi:10.1007/s00454-013-9538-5.

11 Miroslav Chlebík and Janka Chlebíková. Complexity of approximating bounded variants of optimization problems. *Theor. Comput. Sci.*, 354(3):320–338, 2006. doi:10.1016/j.tcs.2005.11.029.

12 Miroslav Chlebík and Janka Chlebíková. The complexity of combinatorial optimization problems on d-dimensional boxes. *SIAM J. Discret. Math.*, 21(1):158–169, 2007. doi:10.1137/050629276.

13 Brent N. Clark, Charles J. Colbourn, and David S. Johnson. Unit disk graphs. *Discret. Math.*, 86(1-3):165–177, 1990. doi:10.1016/0012-365X(90)90358-O.

14 David Eppstein. Graph-theoretic solutions to computational geometry problems. In Christophe Paul and Michel Habib, editors, *Proceedings of the 35th International Workshop on Graph-Theoretic Concepts in Computer Science (WG)*, pages 1–16, 2009. doi:10.1007/978-3-642-11409-0_1.

15 Jared Espenant, J. Mark Keil, and Debajyoti Mondal. Finding a maximum clique in a disk graph, 2023. doi:10.48550/ARXIV.2303.07645.

16 Stefan Felsner, Rudolf Müller, and Lorenz Wernisch. Trapezoid graphs and generalizations, geometry and algorithms. *Discret. Appl. Math.*, 74(1):13–32, 1997. doi:10.1016/S0166-218X(96)00013-3.

17 Aleksei V. Fishkin. Disk graphs: A short survey. In Klaus Jansen and Roberto Solis-Oba, editors, *Approximation and Online Algorithms, First International Workshop, WAOA 2003, Budapest, Hungary, September 16-18, 2003, Revised Papers*, volume 2909 of *Lecture Notes in Computer Science*, pages 260–264. Springer, 2003. doi:10.1007/978-3-540-24592-6_23.

18 Herbert Fleischner, Gert Sabidussi, and Vladimir I. Sarvanov. Maximum independent sets in 3- and 4-regular hamiltonian graphs. *Discret. Math.*, 310(20):2742–2749, 2010. doi:10.1016/j.disc.2010.05.028.

19 E. Helly. über mengen konvexer körper mit gemeinschaftlichen punkten. *Jahresbericht der Deutschen Mathematiker-Vereinigung*, 32:175–176, 1923.

20 John Hershberger and Subhash Suri. Finding tailored partitions. *J. Algorithms*, 12(3):431–463, 1991. doi:10.1016/0196-6774(91)90013-O.

21 John E. Hopcroft and Richard M. Karp. An $n^{5/2}$ algorithm for maximum matchings in bipartite graphs. *SIAM J. Comput.*, 2(4):225–231, 1973. doi:10.1137/0202019.

22 Mark L. Huson and Arunabha Sen. Broadcast scheduling algorithms for radio networks. In *Proceedings of MILCOM'95*, volume 2, pages 647–651. IEEE, 1995.

23 Hiroshi Imai and Takao Asano. Finding the connected components and a maximum clique of an intersection graph of rectangles in the plane. *J. Algorithms*, 4(4):310–323, 1983. doi:10.1016/0196-6774(83)90012-3.

24 Hiroshi Imai and Takao Asano. Efficient algorithms for geometric graph search problems. *SIAM J. Comput.*, 15(2):478–494, 1986. doi:10.1137/0215033.

25 J. Mark Keil, Debajyoti Mondal, Ehsan Moradi, and Yakov Nekrich. Finding a maximum clique in a grounded 1-bend string graph. *Journal of Graph Algorithms and Applications*, 26(4), 2022. doi:10.7155/jgaa.00608.

26 Jan Kratochvíl and Jaroslav Nešetřil. Independent set and clique problems in intersection-defined classes of graphs. *Commentationes Mathematicae Universitatis Carolinae*, 31(1):85–93, 1990.

27 Matthias Middendorf and Frank Pfeiffer. The max clique problem in classes of string-graphs. *Discret. Math.*, 108(1-3):365–372, 1992. doi:10.1016/0012-365X(92)90688-C.

28 Michael Ian Shamos and Dan Hoey. Closest-point problems. In *Proceedings of the 16th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 151–162, 1975. doi:10.1109/SFCS.1975.8.

29 Alexander Tiskin. Fast distance multiplication of unit-monge matrices. *Algorithmica*, 71(4):859–888, 2015. doi:10.1007/s00453-013-9830-z.