On Embeddability of Unit Disk Graphs onto Straight Lines

Onur Çağırıcı(✉)

Masaryk University, Brno, Czech Republic
onur@mail.muni.cz

Abstract. Unit disk graphs are the intersection graphs of unit radius disks in the Euclidean plane. Deciding whether there exists an embedding of a given unit disk graph, i.e., unit disk graph recognition, is an important geometric problem, and has many application areas. In general, this problem is known to be ∃ℝ-complete. In some applications, the objects that correspond to unit disks have predefined (geometrical) structures to be placed on. Hence, many researchers attacked this problem by restricting the domain of the disk centers. One example to such applications is wireless sensor networks, where each disk corresponds to a wireless sensor node, and a pair of intersecting disks corresponds to a pair of sensors being able to communicate with one another. It is usually assumed that the nodes have identical sensing ranges, and thus a unit disk graph model is used to model problems concerning wireless sensor networks. We consider the unit disk graph realization problem on a restricted domain, by assuming a scenario where the wireless sensor nodes are deployed on the corridors of a building. Based on this scenario, we impose a geometric constraint such that the unit disks must be centered onto given straight lines. In this paper, we first describe a polynomial-time reduction which shows that deciding whether a graph can be realized as unit disks onto given straight lines is NP-hard, when the given lines are parallel to either the x-axis or y-axis. Using the reduction we described, we also show that this problem is NP-complete when the given lines are only parallel to the x-axis (and one another). We obtain these results using the idea of the logic engine introduced by Bhatt and Cosmadakis in 1987.

1 Introduction

An *intersection graph* is a graph that models the intersections among geometric objects. In an intersection graph, each vertex corresponds to a geometric object, and each edge corresponds to a pair of intersecting geometric objects. A *unit disk graph* is the intersection graph of a set of unit disks in the Euclidean plane. Some well-known NP-hard problems, such as chromatic number, independent set, and dominating set, remain hard on unit disk graphs [4,6,11]. We are particularly interested in the unit disk recognition problem, i.e., given a simple

This work is supported by the Czech Science Foundation, project no. 20-04567S.

H. Fernau (Ed.): CSR 2020, LNCS 12159, pp. 184–197, 2020.
https://doi.org/10.1007/978-3-030-50026-9_13

graph, deciding whether there exists an embedding of disks onto the plane which corresponds to the given graph. This problem is known to be NP-hard [9], and even ∃ℝ-complete [16] in general.

A major application area of unit disk graphs is wireless sensor networks, since it is an accurate model (in an ideal setting) of communicating wireless sensor nodes with identical range [3,12]. In a wireless sensor network, the sensor nodes are deployed on bounded areas [2,10,19]. Thus, it becomes more interesting to observe the behavior of the unit disk graph recognition problem when the domain is restricted [1,8,13,15].

We assume that the sensor nodes are deployed onto the corridors in a building, and the floor plans are available. We model the corridors on a floor as straight lines, and consider the recognition problem where the unit disks are centered on the given lines. We show that this problem is NP-hard, even when the given straight lines are either vertical or horizontal, i.e., any pair of lines is either parallel, or perpendicular to each other. In addition, we show that if there are no pairs of perpendicular lines i.e., all lines are parallel to x-axis, then the recognition problem is NP-complete.

Due to space restrictions, the proofs of some statements are omitted, and those statements are marked with (*). The full version of this paper is available online at http://arxiv.org/abs/1811.09881.

Related Work

Breu and Kirkpatrick showed that the unit disk graph recognition problem is NP-hard in general [9]. Later on, this result was extended, and it was proved that the problem is also ∃ℝ-complete [16,18]. Kuhn et al. showed that finding a "good" embedding is not approximable when the problem is parameterized by the maximum distance between any pair of disk centers [17]. In the very same paper, they also give a short reduction that the realization problem and the recognition problem on unit disk graphs are polynomially equivalent [17].

Intuitively, the most restricted domain for unit disk graphs is when the disks are centered on a single straight line in the Euclidean plane. In this case, the unit disks become *unit intervals* on the line, and they yield a *unit interval graph* [14]. To recognize or realize whether a given graph is a unit interval graph is a linear-time task [7]. Our domain is restricted to not only one straight line, but to a set of straight lines given by their equations. Given a simple graph, and a set of straight lines, we ask the question "can this graph be realized as unit disks on the given set of straight lines?" We show that even though these lines are restricted to be parallel to either the x-axis or y-axis, it is NP-hard to determine whether the given graph can be embedded onto the given lines (Theorem 1). We, however, do not know whether this variant belongs to the class NP, or is possibly ∃ℝ-complete. If, on the other hand, the lines are restricted to be parallel only to the x-axis, then we show that the problem belongs to NP and thus is NP-complete.

2 Basic Terminology and Notations

A *unit disk* around a point p is the set of points in the plane whose distance from p is one unit. Two unit disks, centered at two points p and q, intersect when the Euclidean distance between p and q is less than or equal to two units. A graph $G = (V, E)$ is called a *unit disk graph* when every vertex $v \in V$ corresponds to a disk $\mathcal{D}_v$ in the Euclidean plane, and an edge $uv \in E$ exists when $\mathcal{D}_u$ and $\mathcal{D}_v$ intersect.

The *unit disk recognition problem* is deciding whether a given graph $G = (V, E)$ is a unit disk graph. That is, determining whether there exists a mapping $\Sigma : V \to (\mathbb{R} \times \mathbb{R})$, such that each vertex is the center of a unit disk without violating the intersection property. The mapping Σ is also called the *embedding of G by unit disks.* We use the domain of *axes-parallel straight lines* which is a set of lines in 2D, where the angle between a pair of lines is either 0 or $\pi/2$. This implies that the equation of a straight line is either $y = a$ if it is a horizontal line, or $x = b$ if it is a vertical line, where $a, b \in \mathbb{R}$. The input for axes-parallel straight lines recognition problem contains two sets, $\mathcal{H}, \mathcal{V} \subset \mathbb{R}$, where $\mathcal{H}$ contains the Euclidean distance of each horizontal line from the x-axis, and $\mathcal{V}$ contains the Euclidean distance of each vertical line from the y-axis. Thereby in the domain that we use, each vertex is mapped either onto a vertical line, or onto a horizontal line. We denote the class of axes-parallel unit disk graphs as APUD(k, m) where k is the number of horizontal lines, and m is the number of vertical lines. Formally, we define the problem as follows.

Definition 1 (Axes-parallel unit disk graph recognition on k horizontal and m vertical lines). *The input is a graph $G = (V, E)$, where $V = \{1, 2, \ldots, n\}$, and two sets $\mathcal{H}, \mathcal{V} \subset \mathbb{Q}$ of rational numbers with $|\mathcal{H}| = k$ and $|\mathcal{V}| = m$. The task is to determine whether there exists a mapping $\Sigma : V \to (\mathbb{R} \times \mathcal{H}) \cup (\mathcal{V} \times \mathbb{R})$ such that there is a unit disk realization of G in which $u \in \ell_{\Sigma(u)}$ for each $u \in V$.*

3 APUD(k, m) Recognition Is NP-Hard

We prove that axes-parallel unit disk recognition (APUD(k, m) recognition with k and m given as input) is NP-hard by giving a reduction from the *Monotone not-all-equal 3-satisfiability* (NAE3SAT) problem[1]. NAE3SAT is a variation of 3SAT where three values in each clause are not all equal to each other, and due to Schaefer's dichotomy theory, the problem remains NP-complete when all clauses are monotone (i.e., none of the literals are negated) [20]. Our main theorem is as follows.

Theorem 1. *There is a polynomial-time reduction of any instance Φ of Monotone NAE3SAT to some instance Ψ of* APUD(k, m) *such that Φ is a YES-instance if, and only if Ψ is a YES-instance.*

[1] This problem is equivalent to the 2-coloring of 3-uniform hypergraphs. We choose to give the reduction from Monotone NAE3SAT as it is more intuitive to construct for our problem.

We construct our hardness proof using the scheme called a *logic engine*, which is used to prove the hardness of several geometric problems [5]. For a given instance Φ of Monotone NAE3SAT, there are two main components in our reduction. First, we construct a backbone for our gadget. The backbone models only the number of clauses and the number of literals. Next, we model the relationship between the clauses and literals, i.e., which literal appears in which clause.

Let us begin by describing the input graph. For the sake of simplicity, we assume that the given formula has 3 clauses, A, B, C, and 4 literals, q, r, s, t for the moment. In general, we denote the clauses by $C_1, \ldots, C_k$, and the literals by $x_1, \ldots, x_m$. Later on, we explain how to generalize the input graph according to any given instance of Monotone NAE3SAT formula. For the following part, we describe the input graph given in Fig. 1a. Throughout the manuscript, we index the vertices from left to right, and from bottom to top, in ascending order.

Three essential components of the input graph are the following induced paths $P_\alpha = (\alpha_1, \alpha_2, \ldots, \alpha_{11})$, $P_L = (L_1, L_2, \ldots, L_{15})$, and $P_R = (R_1, R_2, \ldots, R_{15})$. The length of P_α is $2m + 3$ for m literals. In our case, $(2 \times 4) + 3 = 11$. The lengths of P_L and P_R are the same, equal to $3 + 4k$ for k clauses. In our case, $3 + (4 \times 3) = 15$.

The middle vertices of P_L and P_R are the end vertices of P_α. That is, $\alpha_1 = L_8$, and $\alpha_{11} = R_8$. The paths P_L and P_R define the left and the right boundary for our gadget, respectively.

For $i = q, r, s, t$, there is an induced path $P_i = (i_1, \ldots, i_{15})$ for each literal, with 15 vertices. In general, we denote those paths by $P^1, P^2, \ldots, P^m$ for m literals. The vertices of these paths are denoted by blue circles in Fig. 1a, they are mutually disjoint, but each of them shares one vertex with P_α. The shared vertices are precisely the middle vertices, which are indicated by green rectangles in the figure. That is, $\alpha_3 = q_8$, $\alpha_5 = r_8$, $\alpha_7 = s_8$, and $\alpha_9 = t_8$. Moreover, i_1 is a vertex of an induced 4-cycle, and i_{15} is a vertex of another induced 4-cycle for $i = q, r, s, t$. The three vertices in a 4-cycle, except the one in one of the induced paths, are indicated by the red color in the figure. Precisely two of them, that are adjacent to a blue vertex (either i_1 or i_{15}) are indicated by squares, and the remaining is indicated by a triangle.

Starting from the second edge of P_L (respectively P_R), every second edge is a chord of a 4-cycle (C_4). Throughout the paper, we refer to such 4-cycles with a chord as a *diamond*. Two vertices of these diamonds are of P_L (respectively P_R), and remaining two are denoted by red triangles in Fig. 1a.

Remember that the problem takes two inputs: a graph, and a set of lines determined by their equations (or rather by two sets of rational numbers, since every line is parallel to either the x- or y- axis). For a Monotone NAE3SAT formula with 3 clauses and 4 literals, we have described the input graph above. Now, let us discuss the input lines of our gadget. The input graph is given in Fig. 1a, and the corresponding lines are given in Fig. 1b. We claim that the given graph can be embedded onto the given lines with ε flexibility, and the resulting realization looks like the set of unit disks given in Fig. 1c.

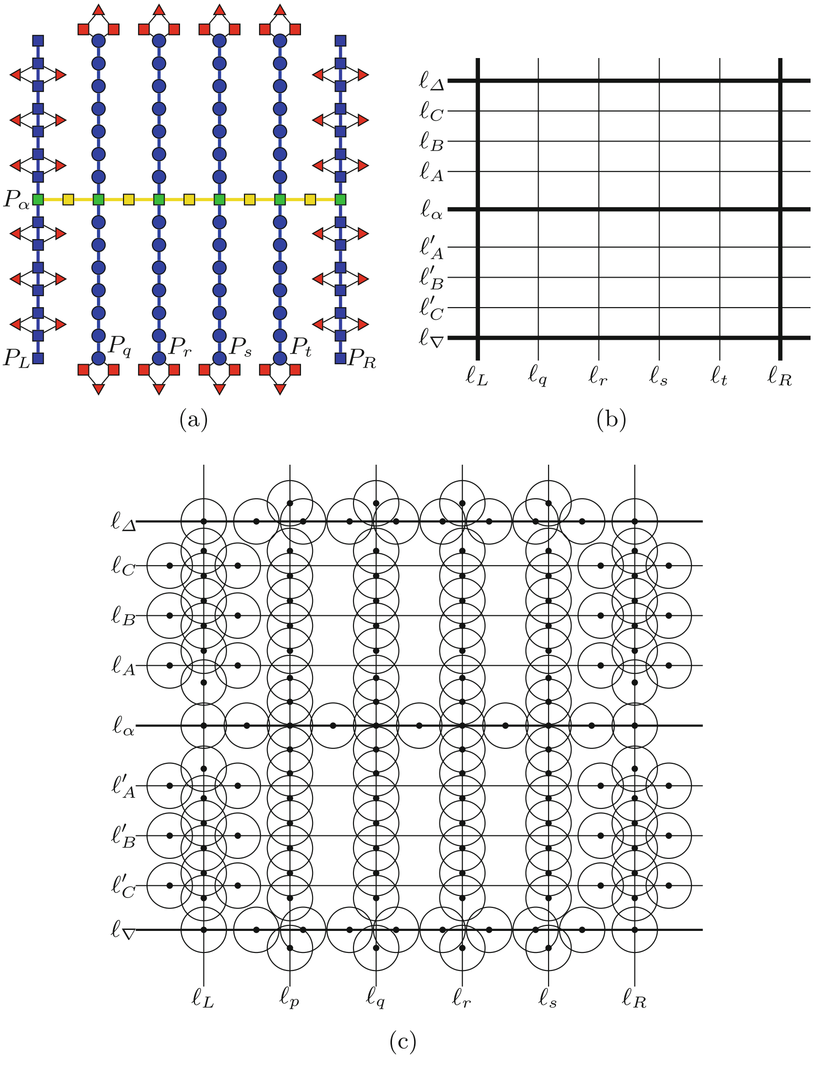


Fig. 1. (a) Skeleton of the input graph for Φ. The consecutive induced paths, labeled as P_q, P_r, P_s, P_t, are to be embedded on the literal lines $\ell_q, \ell_r, \ell_s, \ell_t$ in Fig. 1b, respectively. The vertices in the long induced paths P_L and P_R in 1a (indicated by rectangles) must be embedded on the lines ℓ_L and ℓ_R given in 1b. Similarly, the vertices in P_α (indicated by blue and green rectangles) must be embedded on the line ℓ_α given in 1b.
(b) The line set of the configuration for a Monotone NAE3SAT formula Φ with 4 literals (q, r, s, t) and 3 clauses (A, B, C).
(c) Realization of the graph given in 1a onto the lines given in 1b. (Color figure online)

In order to force such an embedding, we adjust the Euclidean distance between each pair of parallel lines carefully. We start by defining the horizontal line ℓ_α. This line is the axis of horizontal symmetry for our line configuration. Thus, it is safe to assume that ℓ_α is the x-axis. On the positive side of the y-axis, for each clause A, B, and C, there is a straight line parallel to ℓ_α, and another horizontal line acting as the top boundary of the configuration. These lines are denoted by ℓ_A, ℓ_B, ℓ_C, and ℓ_Δ, and their equations are $y = a$, $y = b$, $y = c$, and $y = \Delta$, respectively, where $a < b < c < \Delta$. For every pair of consecutive horizontal lines, the Euclidean distance between them is precisely 2.01 units. That is, $a = 2.01$, $b = 4.02$ and $c = 6.03$, and $\Delta = 8.04$. For every horizontal line described above, there is another horizontal line symmetric to it about the x-axis. These lines are $\ell'_A, \ell'_B, \ell'_C$, and ℓ_∇ (see Fig. 1b).

The leftmost vertical line is ℓ_L, which is the left boundary of our configuration. We can safely assume that ℓ_L is the y-axis for the sake of simplicity. For each literal q, r, s and t, there exists a vertical line parallel to ℓ_L, and another vertical line that defines the right boundary of our configuration. These lines are denoted by ℓ_q, ℓ_r, ℓ_s, ℓ_t, and ℓ_R, and their equations are $x = q$, $x = r$, $x = s$, $x = t$ and $x = R$, respectively, where $q < r < s < t < R$. The Euclidean distance between each pair of consecutive vertical lines is precisely 3.8 units. That is $q = 3.8$, $r = 7.6$, $s = 11.4$, $t = 15.2$, and $R = 19$.

Up to this point, we have described the input graph, and the input lines for a given Monotone NAE3SAT formula with 3 clauses and 4 literals. In general, for a given Monotone NAE3SAT formula Φ with k clauses $C_1, C_2, \ldots, C_k$, and m literals $x_1, x_2, \ldots, x_m$, our gadget has the following components.

1. An induced path $P_\alpha = (\alpha_1, \alpha_2, \ldots, \alpha_{2m+3})$ with $2m + 3$ vertices.
2. m induced paths $P^1 = (P^1_1, P^1_2, \ldots, P^1_{4k+3}), \ldots, P^m = (P^m_1, P^m_2, \ldots, P^m_{4k+3})$, each with $4k+3$ vertices, where $\alpha_3 = P^1_{2k+2}$, $\alpha_5 = P^2_{2k+2}, \ldots, \alpha_{2k+1} = P^m_{2k+2}$, and induced 4-cycles containing the first and the last vertices of each of these paths.
3. Two induced paths $P_L = (L_1, \ldots, L_{4k+3})$ and $P_R = (R_1, \ldots, R_{4k+3})$, each with $4k+3$ vertices, where the edges L_2L_3, L_4L_5, $\ldots$, $L_{2k}L_{2k+1}$, $L_{2k+3}L_{2k+4}$, $\ldots$, $L_{4k+1}L_{4k+2}$, and R_2R_3, R_4R_5, $\ldots$, $R_{2k}R_{2k+1}$, $R_{2k+3}R_{2k+4}$, $\ldots$, $R_{4k+1}R_{4k+2}$ are chords of disjoint 4-cycles.
4. $2k + 3$ horizontal lines $\ell_\nabla, \ell'^C_k, \ell'^C_{k-1}, \ldots, \ell_\alpha, \ell^C_1, \ell^C_2, \ldots, \ell^C_k, \ell_\Delta$, with equations $\ell_\nabla : y = -2.01(k + 1)$, $\ell_\Delta : -\ell_\nabla$, $\ell_\alpha : y = 0$, $\ell'^C_i = -2.01i$, and $\ell^C_i = 2.01i$ for $i = 1, 2, \ldots, k$.
5. $m + 2$ vertical lines $\ell_L, \ell^x_1, \ell^x_2, \ldots, \ell^x_m, \ell_R$, with equations $\ell_L : x = 0$, $\ell_R : x = 3.8(m + 1)$, and $\ell^x_i : 3.8i$ for $i = 1, 2, \ldots, m$.

In total, for the given formula Φ with k clauses and m literals, our gadget is an instance of APUD$(2k + 3, m + 2)$. Here, we conclude the proof of Theorem 1.

Now, let us show that the given graph has a unique embedding onto the given lines, up to ε flexibility.

Claim. The vertices indicated by rectangles in Fig. 1a can only be embedded on the bold lines in Fig. 1b.

Let us start by discussing the embedding of P_L onto ℓ_L (and respectively P_R onto ℓ_R). We give the following two trivial lemmas as preliminaries for the proof of our claim.

Lemma 1 (*). *Consider two disks A and B, centered on $(a, 0)$ and $(b, 0)$ with $0 < |a| < |b|$. Another disk, C that is centered on $(0, c)$ cannot intersect B without intersecting A.*

Lemma 2 (*). *An induced 4-star ($K_{1,4}$) can be realized as a unit disk graph on two perpendicular lines, but not on two parallel lines.*

Now, with the help of Lemmas 1 and 2, we state the following lemmas, and prove our claim.

Lemma 3. *The induced paths P_L, P_R and P_α in the input graph (Fig. 1a) can only be embedded onto ℓ_L, ℓ_R, and ℓ_α, respectively (Fig. 1b).*

Proof (Sketch). The diamonds on the left and the right side of the figure should be embedded around an intersection. There are a total of six diamonds, and thus six intersections are required.

P_α has induced 4-stars, and those 4-stars are vertices of long induced paths, P_α cannot be embedded on multiple lines (via bending etc.). The middle vertices of P_L and P_R are the two ends of P_α. Since P_α is realized on a single line, another intersection is required to realize $K_{1,3}$ consists of $L_8; L_7, L_9, \alpha_2$.

In total, P_L requires seven intersections. Those seven intersections are between a vertical line and seven horizontal lines, excluding ℓ_Δ and ℓ_∇. The same argument applies to P_R up to symmetry. □

Claim. For the given input graph for 3 clauses and 4 literals, the following hold:

i) The induced paths $P_q = (q_1, \ldots, q_{15})$, $P_r = (r_1, \ldots, r_{15})$, $P_s = (s_1, \ldots, s_{15})$ and $P_t = (t_1, \ldots, t_{15})$ in the input graph given in Fig. 1a can only be embedded onto ℓ_q, ℓ_r, ℓ_s, and ℓ_t, respectively.
ii) The center of each disk that correspond to a vertex of those induced paths must be between ℓ_Δ and ℓ_∇.
iii) A pair of non-intersecting disks that are included in an induced 4-cycle, but not included in any of P_q, P_r, P_s, P_t, must lie on either ℓ_Δ or ℓ_∇ (red rectangles in Fig. 1a).

Proof (Sketch). Due to Lemma 3, we know that P_α is realized on ℓ_α, thus (i) holds. 4-cycles require at least two lines, and those two lines cannot be two parallel lines, as the Euclidean distance between each pair of consecutive parallel lines is larger than 2. Thus, (ii) holds. The induced paths P_q, P_r, P_s, P_t can be squeezed enough to be realized between ℓ_α and ℓ_C because $\ell_C : y = 6.03$, but then the 4-cycles cannot be realized. Therefore, the disks that correspond to two vertices of these 4-cycles must be centered on ℓ_Δ (and symmetrically on ℓ_∇). Thus, (iii) holds. □

Lemma 4 (*). *For the given input graph for k clauses and m literals, the following hold:*

i) The induced paths $P_1 = (P_1^1, \dots, P_{4k+3}^1)$, $P^2 = (P_1^2, \dots, P_{4k+3}^2)$, ... , $P^n = (P_1^n, \dots, P_{4k+3}^n)$ in the input graph can only be embedded onto ℓ_1^x, ℓ_2^x, ... ℓ_m^x, respectively.

ii) The center of each disk that correspond to a vertex of those induced paths must be between ℓ_Δ and ℓ_∇.

iii) A pair of non-intersecting disks that are included in an induced 4-cycle, but not included in any of $P^1, P^2, \dots, P^n$, must lie on either ℓ_Δ or ℓ_∇ (red rectangles in Fig. 1a).

With Lemmas 3 and 4, we have shown that the vertices denoted by rectangles in Fig. 1a must be embedded onto the bold lines in Fig. 1b.

Using the backbone we have described, we now show how to model the relationship between the clauses and the literals. To make it easier to follow, we also refer to Fig. 1a in parentheses in the following description. Consider a sub-path $(P_{2k+3}^i, P_{2k+4}^i, \dots, P_{4k+3}^i)$ of the induced path P^i. This part corresponds to the literal x_i of the given Monotone NAE3SAT formula (corresponding to $(q_9, \dots, q_{15})$ of P_q in our example). The edges $P_{2k+3}^i P_{2k+4}^i$, $P_{2k+5}^i P_{2k+6}^i$, ... , $P_{4k+1}^i P_{4k+2}^i$ (corresponding to $q_9 q_{10}$, $q_{11} q_{12}$, and $q_{13} q_{14}$ in P_q in our example) are used to model membership of x_i in the clauses $C_1, C_2, \dots, C_k$ (corresponding to the clauses A, B, and C in our example), respectively.

If x_i appears in a clause C_j, then we do nothing for the edges correspond do those clauses. Otherwise, if x_i does not appear in C_j, then we introduce a *flag vertex* in the graph, which is adjacent to $P_{2(k+j)+1}^i$ and $P_{2(k+j)+2}^i$. Due to the rigidity of the backbone (up to ε flexibility), this flag vertex lies on ℓ_j^C. Similarly, in our example, if q appears in B, then $q_{11} q_{12}$ stays as is, but otherwise, a flag vertex is introduced, adjacent to both q_{11} and q_{12}.

Every clause has 3 literals. Thus, on each horizontal line, 3 out of m possible flag vertices will be missing. That sums up to a total of $k(m-3)$ flag vertices for this part of the graph. For the remaining sub-path $(P_1^i, \dots, P_{2k+2}^i)$ of P^i (corresponding to $(q_1, \dots, q_8)$ of P_q in our example), we introduce the flag vertices for the pairs (P_2^i, P_3^i), (P_4^i, P_5^i), ... , P_{2k}^i, P_{2k+1}^i (corresponding to (q_6, q_7) (q_2, q_3), (q_4, q_5), (q_6, q_7) in our example). That is a total number of km flag vertices for this part of the graph. In the whole graph, there are precisely 2 km–3 km flag vertices.

Realize that the embeddings on some vertical lines must be flipped upside-down to create space for the flag vertices. This operation corresponds to the truth assignment of the literal that corresponds to that vertical line. The configuration forces at least one literal to have a different truth assignment, because for a pair of symmetrical horizontal lines, say ℓ_A and ℓ'_A, there must be at least one missing flag, and at most two missing flags for the disks to fit between ℓ_L and ℓ_R.

The input graph, a YES-instance, and the realization of the YES-instance of the Monotone NAE3SAT formula $\Phi = (q \vee s \vee t) \wedge (q \vee r \vee t) \wedge (q \vee r \vee s)$ is given in Fig. 2.

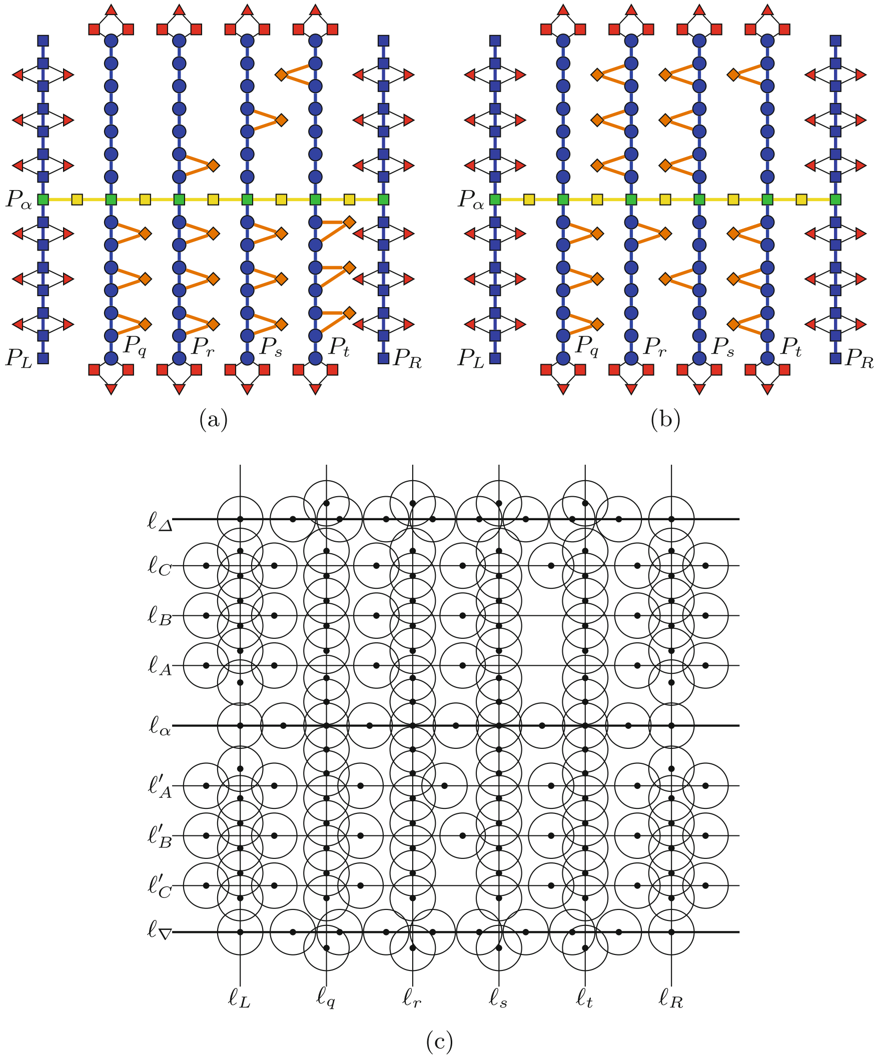


Fig. 2. (a) The input graph for the Monotone NAE3SAT formula Φ with literals q, r, s, t, and clauses $A, B, C.\Phi = A \wedge B \wedge C$ where $A = (q \vee s \vee t)$, $B = (q \vee r \vee t)$, and $C = (q \vee r \vee s)$. The flag vertices, indicated by orange diamonds, are adjacent to the vertices that correspond to a clause on an induced path, if the literal does not appear in that clause.
(b) A truth assignment that satisfies the formula given in 2a: $q =$ TRUE, $r =$ FALSE, $s =$ FALSE, and $t =$ TRUE.
(c) Realization of the graph given in 2b. (Color figure online)

Corollary 1. *Given a graph $G = (V, E)$, deciding whether G is a unit disk graph is an NP-hard problem when the size of the largest induced cycle in G is of length* 4*.*

4 APUD(k, 0) Recognition Is NP-Complete

In this section, we show that the recognition of axes-parallel unit disk graphs is NP-complete when all the given lines are parallel to each other. This version of the problem is referred to as APUD$(k, 0)$, as there are k horizontal lines given as input, but no vertical lines. We use the reduction given in Sect. 3.

Theorem 2. APUD$(k, 0)$ *recognition is NP-hard.*

Proof. Consider the realization given in Fig. 1c. Notice that the length of the paths $P^1, P^2, \dots, P^m$ (P_q, P_r, P_s, P_t in our example), and thus the number of disks on vertical lines, is equal. Lemma 3 (ii) implies that those disks must be centered between ℓ_∇ and ℓ_Δ. Thus, for the disks that correspond to the vertices on these paths, we do not need any vertical line. We can simply remove the vertical lines, and add an extra horizontal line for each clause. For the disks that are adjacent to, but not on P_L and P_R, we can simply add another horizontal line. That is an extra horizontal line for each clause. As a result, for a given instance Φ of Monotone NAE3SAT formula with k clauses and m literals, we have an instance Ψ of APUD$(2k + 3, m + 2)$ to prove NP-hardness with vertical lines, and an instance Ψ' of APUD$(4k + 3, 0)$ to prove NP-completeness without vertical lines. For each clause, we have 3 horizontal lines.

In Ψ', only disks that can "jump" from one horizontal line to another are the ones that are on the top line of ℓ_1^C and bottom line of $\ell_1'^C$. And those jumps do not change the overall configuration. □

To show that APUD$(k, 0)$ recognition is in NP, we need to prove that a given solution can be verified in polynomial time and additionally that any feasible input will have a solution that takes up polynomial space, with respect to the input size. Thus, we show that for any graph $G \in$ APUD$(0, k)$, there exists an embedding where the disk centers are represented using polynomially many decimals with respect to the input size.

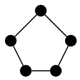
Below, we describe a procedure to show the existence of an embedding with polynomially many digits for every feasible output. Let $h_0, h_1, h_2 \dots, h_k$ denote the rational numbers that correspond to the horizontal lines given as input.

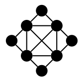
1. Let $G_0, G_1, \dots, G_k$ denote the disjoint induced subgraphs of G, such that the vertices of G_i correspond to the disks centered on line $y = h_i$.
2. Embed G_0 on x-axis with small perturbations which results in all disks on $y = 0$ having polynomially many decimals.
3. For each $1 \le i \le k$, find an embedding of G_i onto the $y = h_i$ line by only considering neighbors from $\bigcup_{j<i} G_j$.

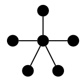
Therefore, APUD$(k, 0)$ recognition is NP-complete.

5 APUD(1, 1) Recognition Is Open

In this section, we discuss a natural basis for the APUD(k, m) recognition problem, that is, $k = 1$ and $m = 1$. For the sake of simplicity, we can assume that our given two lines are the x-axis and the y-axis. First, we give some forbidden induced subgraphs for APUD(1, 1). Namely, those subgraphs are the 5-cycle (C_5), the 4-sun (S_4), and the 5-star ($K_{1,5}$).


C_5


S_4


$K_{1,5}$

Lemma 5 (*). $C_5, S_4, K_{1,5} \notin$ APUD(1, 1).

Lemma 6. *A given graph G can be embedded on x-axis and y-axis as a unit disk intersection graph, without using negative coordinates for the disk centers if, and only if, G is a unit interval graph.*

Proof. In this proof, let us denote the class of graphs that can be embedded on the x- and y- axes as unit disks, using positive coordinates only by $(xy)^+$. We show that the disks on the y-axis can be rotated by $\pi/2$ degrees counterclockwise, and the intersection relationships can be preserved as given in G.

$\mathbf{G} \in (\mathbf{xy})^+ \Rightarrow \mathbf{G} \in \mathbf{UIG}$**:** Consider two disks, A and B, whose centers are $(a, 0)$ and $(0, b)$, respectively, where a and b are both positive numbers. If A and B do not intersect, then $\sqrt{a^2 + b^2} > 2$. After the rotation, the center of B will be on $(-b, 0)$. The new distance between the centers is $a + b$. Since $(a + b)^2 > a^2 + b^2 > 4$, the inequality $a + b > 2$ holds.

If A and B intersect, then $\sqrt{a^2 + b^2} \leq 2$. After the rotation, it might still be the case that $a+b > 2$. However, now we can safely move the center of B and the other centers that have negative coordinates closer to the center of A, recovering the intersection. Note that if a disk C is centered in between the centers of A and B after the rotation, then both A and B must intersect C by Lemma 1.

$\mathbf{G} \in \mathbf{UIG} \Rightarrow \mathbf{G} \in (\mathbf{xy})^+$**:** Since G is a unit interval graph, we can assume that every interval is a unit disk, and the graph is embedded on x-axis. Consider two disks, A and B, whose centers are $(-a, 0)$ and $(b, 0)$, respectively, where a and b are both positive numbers. If A and B are intersecting, then $a + b \leq 2$ Then, after the rotation, since $a + b \leq 2$ holds, then $\sqrt{a^2 + b^2} \leq 2$ also holds.

If A and B are not intersecting, then $a+b > 2$. After the rotation $\sqrt{a^2 + b^2} \leq 2$ might hold, creating an intersection between A and B. However, we can simply shift the center of A (along with the other centers that are on y-axis) far away from the center of B, separating A and B. ☐

Lemma 6 shows APUD$^+(1, 1)$ = APUD$^+(1, 0)$ = UIG if we use only non-negative coordinates. This also applies if we use only non-positive coordinates. Thus, a given APUD(1, 1) can always be partitioned into two unit interval graphs. Considering the embedding, one of these two partitions contains the disks that are centered on the positive sides of x- and y-axes, and the other partition contains the disks centered on the negative sides of x- and y-axes.

Lemma 7. *A graph $G \in \mathrm{APUD}(1,1)$ can be vertex-partitioned into four parts, such that any two form a unit interval graph.*

Proof. Let $\Sigma(G)$ be an embedding of G onto x- and y- axes as unit disks. Denote the set of unit disks in $\Sigma(G)$ that are centered on the positive side of the x-axis, and the positive side of the y-axis by $\Sigma^+(G)$. Similarly, by $\Sigma^-(G)$ denote the set of unit disks that are centered on the negative side of the x-axis, and the negative side of the y-axis. By Lemma 6, both $\Sigma^+(G)$ and $\Sigma^-(G)$ yield separate interval graphs. The vertices corresponding to the disks in $\Sigma^+(G)$ yield a unit interval graph, as do the vertices corresponding to the disks in $\Sigma^-(G)$. If there exists a disk centered at $(0,0)$, then it can be included in one of the partitions arbitrarily. Hence, we can vertex-partition G into two unit interval graphs. □

Up to this point, we showed that if a unit disk graph can be embedded onto two orthogonal lines, then it can be partitioned into two interval graphs. However, this implication obviously does not hold the other way around. Thus, we now identify some structural properties of $\mathrm{APUD}(1,1)$.

Remark 1. Consider four unit disks A, B, C, D, embedded onto x-axis and y-axis. If they induce a 4-cycle, then the centers of those disks will be at $(a, 0)$, $(0, b)$, $(-c, 0)$, $(0, -d)$, respectively, where a, b, c, d are non-negative numbers.

For the upcoming lemma, we will utilize Remark 1. The lemma is an important step towards describing a characterization of $\mathrm{APUD}(1,1)$.

Lemma 8 (*). *Consider eight unit disks embedded onto x-axis and y-axis, around the origin, whose intersection graph contains two induced 4-cycles. Then, this intersection contains at least four 4-cycles, each with a chord, not necessarily as induced subgraphs. Moreover, those 4-cycles are formed by pairs of disks on the same direction ($+x$, $+y$, $-x$, $-y$) with respect to the origin.*

The lemmas imply that for a connected graph $G \in \mathrm{APUD}(1,1)$, we can deduce:

(i) G does not contain either of the 4-sun (S_4) or the 5-star($K_{1,5}$) as an induced subgraph, and the largest induced cycle in G is of length 4 (by Lemma 5).
(ii) G can be vertex partitioned into into four parts, such that any two form a unit interval graph (by Lemma 7).
(iii) Given two 4-cycles, (a, b, c, d) and (u, v, w, x) in G, each one of the quadruplets $\{a, b, u, v\}$, $\{b, c, v, w\}$, $\{c, d, w, x\}$, and $\{d, a, x, u\}$ is either a diamond or a K_4 (by Lemma 8).

Although this characterization gives a rough idea regarding the structure of a graph $G \in \mathrm{APUD}(1,1)$, it is not clear if the recognition can be done in polynomial time. Our characterization is a necessary step through recognition, but it is not yet known whether it is sufficient. Hence, we conjecture that given a graph G, it can be determined whether $G \in \mathrm{APUD}(1,1)$ in polynomial time.

Acknowledgments. The author wants to thank Petr Hliněný for his insight on the hardness proof. In addition, he thanks Deniz Ağaoğlu and Michał Dębski for their extensive comments and generous help during the preparation of this manuscript.

References

1. Alber, J., Fiala, J.: Geometric separation and exact solutions for the parameterized independent set problem on disk graphs. J. Algorithms **52**(2), 134–151 (2004)
2. Alomari, A., Aslam, N., Phillips, W., Comeau, F.: Three-dimensional path planning model for mobile anchor-assisted localization in Wireless Sensor Networks. In: 30th IEEE Canadian Conference on Electrical and Computer Engineering, CCECE, pp. 1–5 (2017)
3. Aspnes, J., et al.: A theory of network localization. IEEE Trans. Mob. Comput. **5**(12), 1663–1678 (2006)
4. Balasundaram, B., Butenko, S.: Optimization problems in unit-disk graphs. In: Floudas, C., Pardalos, P. (eds.) Encyclopedia of Optimization, pp. 2832–2844. Springer, Boston (2009). https://doi.org/10.1007/978-0-387-74759-0
5. Bhatt, S.N., Cosmadakis, S.S.: The complexity of minimizing wire lengths in VLSI layouts. Inf. Process. Lett. **25**(4), 263–267 (1987)
6. Bonnet, É., Giannopoulos, P., Kim, E.J., Rzążewski, P., Sikora, F.: QPTAS and subexponential algorithm for maximum clique on disk graphs. In: Speckmann, B., Tóth, C.D. (eds.) 34th International Symposium on Computational Geometry, SoCG. LIPIcs, vol. 99, pp. 12:1–12:15. Schloss Dagstuhl - Leibniz-Zentrum für Informatik (2018)
7. Booth, K.S., Lueker, G.S.: Testing for the consecutive ones property, interval graphs, and graph planarity using PQ-tree algorithms. J. Comput. Syst. Sci. **13**(3), 335–379 (1976)
8. Breu, H.: Algorithmic aspects of constrained unit disk graphs. Ph.D. thesis, University of British Columbia (1996)
9. Breu, H., Kirkpatrick, D.G.: Unit disk graph recognition is NP-hard. Comput. Geom. **9**(1–2), 3–24 (1998)
10. Çağırıcı, O.: Exploiting coplanar clusters to enhance 3D localization in wireless sensor networks. Master's thesis, Izmir University of Economics (2015). http://arxiv.org/abs/1502.07790
11. Clark, B.N., Colbourn, C.J., Johnson, D.S.: Unit disk graphs. Discrete Math. **86**(1–3), 165–177 (1990)
12. Dil, B., Dulman, S., Havinga, P.: Range-based localization in mobile sensor networks. In: Römer, K., Karl, H., Mattern, F. (eds.) EWSN 2006. LNCS, vol. 3868, pp. 164–179. Springer, Heidelberg (2006). https://doi.org/10.1007/11669463_14
13. Evans, W., van Garderen, M., Löffler, M., Polishchuk, V.: Recognizing a DOG is hard, but not when it is thin and unit. In: Demaine, E.D., Grandoni, F. (eds.) 8th International Conference on Fun with Algorithms, FUN. LIPIcs, vol. 49, pp. 16:1–16:12. Schloss Dagstuhl - Leibniz-Zentrum für Informatik (2016)
14. Fishburn, P.C.: Interval Orders and Interval Graphs – A Study on Partially Ordered Sets. Wiley, Hoboken (1985)
15. Ito, H., Kadoshita, M.: Tractability and intractability of problems on unit disk graphs parameterized by domain area. In: Zhang, X.S., Liu, D.G., Wu, L.Y., Wang, Y. (eds.) Operations Research and Its Applications, 9th International Symposium, ISORA. Lecture Notes in Operations Research, vol. 12, pp. 120–127 (2010)
16. Kang, R.J., Müller, T.: Sphere and dot product representations of graphs. Discrete Comput. Geom. **47**(3), 548–568 (2012). https://doi.org/10.1007/s00454-012-9394-8
17. Kuhn, F., Moscibroda, T., Wattenhofer, R.: Unit disk graph approximation. In: Basagni, S., Phillips, C.A. (eds.) Proceedings of the DIALM-POMC Joint Workshop on Foundations of Mobile Computing, pp. 17–23. ACM (2004)

18. McDiarmid, C., Müller, T.: Integer realizations of disk and segment graphs. J. Comb. Theory Ser. B **103**(1), 114–143 (2013)
19. Neto, M.F., Goussevskaia, O., dos Santos, V.F.: Connectivity with backbone structures in obstructed wireless networks. Comput. Netw. **127**, 266–281 (2017)
20. Schaefer, T.J.: The complexity of satisfiability problems. In: Lipton, R.J., Burkhard, W.A., Savitch, W.J., Friedman, E.P., Aho, A.V. (eds.) Proceedings of the 10th Annual ACM Symposium on Theory of Computing, STOC, pp. 216–226. ACM (1978)