Obedient Plane Drawings for Disk Intersection Graphs

Bahareh Banyassady[1,⋆], Michael Hoffmann[2],
Boris Klemz[1], Maarten Löffler[3,⋆⋆], Tillmann Miltzow[4,⋆⋆⋆]

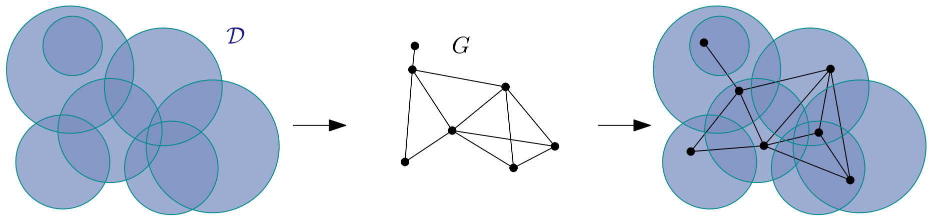
[1] Institute of Computer Science, Freie Universität Berlin, Germany
[2] Department of Computer Science, ETH Zürich, Switzerland
[3] Dept. of Computing and Information Sciences, Utrecht University, the Netherlands
[4] Université libre de Bruxelles (ULB), Brussels, Belgium

Abstract. Let $\mathcal{D}$ be a set of disks and G be the intersection graph of $\mathcal{D}$. A drawing of G is *obedient* to $\mathcal{D}$ if every vertex is placed in its corresponding disk. We show that deciding whether a set of unit disks $\mathcal{D}$ has an obedient plane straight-line drawing is $\mathcal{NP}$-hard regardless of whether a combinatorial embedding is prescribed or an arbitrary embedding is allowed. We thereby strengthen a result by Evans *et al.*, who show $\mathcal{NP}$-hardness for disks with arbitrary radii in the arbitrary embedding case. Our result for the arbitrary embedding case holds true even if G is *thinnish*, that is, removing all triangles from G leaves only disjoint paths. This contrasts another result by Evans *et al*. stating that the decision problem can be solved in linear time if $\mathcal{D}$ is a set of unit disks and G is *thin* , that is, (1) the (graph) distance between any two triangles is larger than 48 and (2) removal of all disks within (graph) distance 8 of a triangle leaves only *isolated* paths. A path in a disk intersection graph is isolated if for every pair A, B of disks that are adjacent along the path, the convex hull of $A \cup B$ is intersected only by disks adjacent to A or B. Our reduction can also guarantee the triangle separation property (1). This leaves only a small gap between tractability and $\mathcal{NP}$-hardness, tied to the path isolation property (2) in the neighborhood of triangles. It is therefore natural to study the impact of different restrictions on the structure of triangles. As a positive result, we show that an obedient plane straight-line drawing is always possible if all triangles in G are *light* and the disks are in general position (no three centers collinear). A triangle in a disk intersection graph is light if all its vertices have degree at most three or the common intersection of the three corresponding disks is empty. We also provide an efficient drawing algorithm for that scenario.

1 Introduction

Disk intersection graphs have been long studied in mathematics and computer science due to their wide range of applications in a variety of domains. They can be used to model molecular bonds as well as the structure of the cosmic web, interference in communication networks and social interactions in crowds. Finding planar realizations of

⋆ supported by DFG project MU/3501-2
⋆⋆ supported by the Netherlands Organisation for Scientific Research (NWO) under project no. 639.021.123 and 614.001.504
⋆⋆⋆ supported by the ERC grant PARAMTIGHT: "Parameterized complexity and the search for tight complexity results", no. 280152



(a)

Fig. 1: A set $\mathcal{D}$ of disks, the induced graph G, and an obedient plane straight-line drawing of G.

disk intersection graphs is an important tool to model sensor networks so as to translate the network connectivity information into virtual coordinates [11]. The planar model can help with topology extraction [10, 15] and enable geometric routing schemes [5, 12].

Rapidly increasing data collection rates call for efficient algorithms to run simulations. But computational problems involving sets of disks are notoriously difficult to grasp: they often lie in a transition zone between tractable and intractable classes of shapes. In this paper, we investigate the computational complexity of one such problem: determining whether a disk intersection graph admits an obedient plane straight-line drawing.

Disk-obedient Drawings. A set $\mathcal{D}$ of disks in the plane induces a graph $G = (V, E)$, called the *disk intersection graph*, which has a vertex for every disk in $\mathcal{D}$ and an edge between two vertices whose (closed) disks intersect. A straight-line *drawing* of G is an injective map $\varphi : V \to \mathbb{R}^2$ so that for every edge $uv \in E$ the open line segment $\varphi(u)\varphi(v)$ is disjoint from $\varphi(V)$. A drawing is *plane* (also called an *embedding*) if the edges do not intersect except at common endpoints. A drawing is *obedient* if every vertex is contained in its corresponding disk. Disk-obedient drawings were introduced by Evans *et al*. [8], who prove that recognizing whether G admits an obedient plane straight-line drawing is $\mathcal{NP}$-hard. This decision problem is called PLANAR DISK OBEDIENCE RECOGNITION. The motivation to study disk-obedience stems from dealing with data uncertainty [7, 13]. The problem is strongly related to *Anchored* Planar Graph Drawing (AGD) (shown to be $\mathcal{NP}$-hard by Angelini *et al*. [2]): Given a planar graph G and an associated unit disk for each circle, produce a planar embedding of G such that each vertex is contained in its disk. Our problem is different from AGD since for us, G itself is defined by the disks. Keszegh *et al*. [9] also study a related problem of placing vertices of a disk intersection graph inside their respective disks such that the resulting drawing is C-oriented; however, the disks in [9] can only touch (their interiors are disjoint), ensuring the drawing is always a plane embedding.

Results and Overview. In this paper we show that several natural restrictions of PLANAR DISK OBEDIENCE RECOGNITION remain $\mathcal{NP}$-hard: even when the disks are unit disks (Section 2), and / or the combinatorial embedding of the graph is given (Section 4), the problem is still hard. In the former case, our result holds true even if removing all triangles leaves only disjoint paths and even if all triangles are far apart. This creates an interesting contrast to results by Evans et al. [8]. Since it is unclear whether the problem is in $\mathcal{NP}$ (see Section 5), our results indicate that the problem is indeed very hard, and it is probably difficult to attack the general problem using a combinatorial approach. On the positive side, we show that the problem can be solved efficiently if the degree of vertices belonging to ply-3 triangles (that is, the three corresponding disks have a common intersection) is bounded by three (Section 3). In this result no assumption concerning the uniformity of the disks' radii is needed. Due to space restrictions, some of the proofs are sketched or omitted in this short paper.

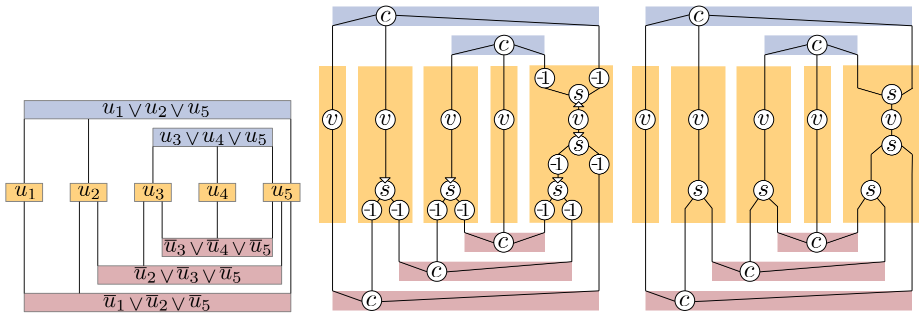
Notation. Throughout this paper disks are closed disks in the plane. Let $\mathcal{D}$ be a set of disks, let $G = (V, E)$ be the intersection graph of $\mathcal{D}$ and let Γ be a straight-line drawing of G. We use capital letters to denote disks and non-capital letters to denote the corresponding vertices, e.g., the vertex corresponding to disk $D \in \mathcal{D}$ is $d \in V$. We use the shorthand uv to denote the edge $\{u, v\} \in E$. Further, uvw refers to the triangle composed of the edges uv, vw and wu. We identify vertices and edges with their geometric representations, e.g., uv also refers to the line segment between the points representing u and v in Γ. We use $\text{int}(D)$ to refer to the interior of a disk $D \in \mathcal{D}$. The *ply* of a point $p \in \mathbb{R}^2$ with respect to $\mathcal{D}$ is the cardinality $|\{D \in \mathcal{D} \mid p \in D\}|$. The *ply* of $\mathcal{D}$ is the maximum ply of any point $p \in \mathbb{R}^2$ with respect to $\mathcal{D}$.

Planar Montone 3-Satisfiability. Let $\varphi = (\mathcal{U}, \mathcal{C})$ be a 3-SATISFIABILITY (3SAT) formula where $\mathcal{U}$ denotes the set of variables and $\mathcal{C}$ denotes the set of clauses. We call the formula φ *monotone* if each clause $c \in \mathcal{C}$ is either *positive* or *negative*, that is, all literals of c are positive or all literals of c are negative. Note that this is not the standard notion of monotone Boolean formulas. Formula $\varphi = (\mathcal{U}, \mathcal{C})$ is *planar* if its *variable clause graph* $G_\varphi = (\mathcal{U} \uplus \mathcal{C}, E)$ is planar. The graph G_φ is bipartite and every edge in E is incident to both a *clause* vertex from $\mathcal{C}$ and a *variable* vertex from $\mathcal{U}$. The edge $\{c, u\}$ is contained in E if and only if a literal of variable $u \in \mathcal{U}$ occurs in $c \in \mathcal{C}$. In the decision problem PLANAR MONOTONE 3-SATISFIABILITY we are given a planar and monotone 3SAT formula φ together with a *monotone rectilinear representation* $\mathcal{R}$ of the variable clause graph of φ and we need to decide whether φ is satisfiable. The representation $\mathcal{R}$ is a contact representation on an integer grid, see Figure 2a. In $\mathcal{R}$ the variables are represented by horizontal line segments arranged on a line ℓ. The clauses are represented by E-shapes. All positive clauses are placed above ℓ and all negative clauses are placed below ℓ. PLANAR MONOTONE 3-SATISFIABILITY is $\mathcal{NP}$-complete [4].

2 Thinnish Unit Disk Intersection Graphs

Evans et al. [8] showed that PLANAR DISK OBEDIENCE RECOGNITION is $\mathcal{NP}$-hard. Further, they provide a polynomial time algorithm to recognize disk-obedience graphs for the case that the respective disk intersection graph is thin and unit. A disk intersection graph G is *thin* if (i) the graph distance between any two triangles of G is larger than 48 and (ii) removal of all disks within graph distance 8 of a triangle decomposes the graph into *isolated* paths. A path is *isolated* if for any pair of adjacent disks A and B of the path, the convex hull of $A \cup B$ is intersected only by disks adjacent to A or B.

In this section we strengthen the $\mathcal{NP}$-hardness result by Evans et al. by showing that PLANAR DISK OBEDIENCE RECOGNITION is $\mathcal{NP}$-hard even for *unit* disks. Further, we show that the path-isolation property of thin disk intersection graphs is essential to make the problem tractable. This is implied by the fact that our result holds even for disk intersection graphs that are *thinnish*, that is, removing all disks that belong to a



(a) **(b)** **(c)**

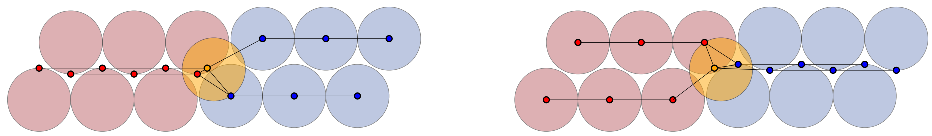
Fig. 2: (a) A monotone rectilinear representation. For the purposes of illustration, the line segments representing variables and clauses are thickened and labeled. **(b)** Gadget layout mimicking the monotone rectilinear representation from (a) in the proof of Theorem 1. The v-nodes are variable gadgets, the c-nodes are clause gadgets, the s-nodes are splitter gadgets and the (-1)-nodes are inverter gadgets. The input of the splitter gadgets are marked with an arrow. The black polygonal paths represent wires. **(c)** Gadget layout for the proof of Theorem 3

triangle decomposes the graph into disjoint paths (which are not necessarily isolated). Being thinnish does not impose any distance or other geometric constraint on the set of disks and its intersection graph. Nevertheless, our reduction also works if the distance between any two triangles is lower bounded by some value. In particular, this implies that spacial separation of triangles is not sufficient for tractability.

Theorem 1. PLANAR DISK OBEDIENCE RECOGNITION *is $\mathcal{NP}$-hard even under any combination of the following restrictions: (1) all disks have the same radius (unit disks); (2) the intersection graph of the disks is thinnish; (3) the graph distance between any two triangles is lower bounded by some value that is polynomial in the number of disks.*

Proof. We describe a polynomial-time reduction from PLANAR MONOTONE 3-SAT. Let $\varphi = (\mathcal{U}, \mathcal{C})$ be a planar monotone 3SAT formula where $\mathcal{U}$ is the set of variables and $\mathcal{C}$ is the set of clauses. On an intuitive level our reduction works as follows. We introduce five different types of gadgets. A variable gadget is created for each variable of $\mathcal{U}$. The gadget has two combinatorial states that are used to encode the truth state of its variable. Wire gadgets are used to propagate these states to other gadgets. In particular, we create a clause gadget for each clause $c \in \mathcal{C}$ and use wires to propagate the truth states of the variables occurring in c to the clause gadget of c. The purpose of the clause gadget is to enforces that at least one of the literals of c is satisfied. In order to appropriately connect the variables with the clauses we require two more gadgets. The splitter gadget splits a wire into two wires and the inverter gadget inverts the state transported along a wire. The gadgets are arranged according to the monotone rectilinear representation $\mathcal{R}$ for φ, see Figure 2b. We proceed by describing our gadgets in detail.

Variables, Wires and Inverters. For each variable vertex we create a *variable* gadget as depicted in Figure 3a. Note how on the left side of the obedient plane straight-line drawing of the disk intersection graph the vertices of the lower path belong to the



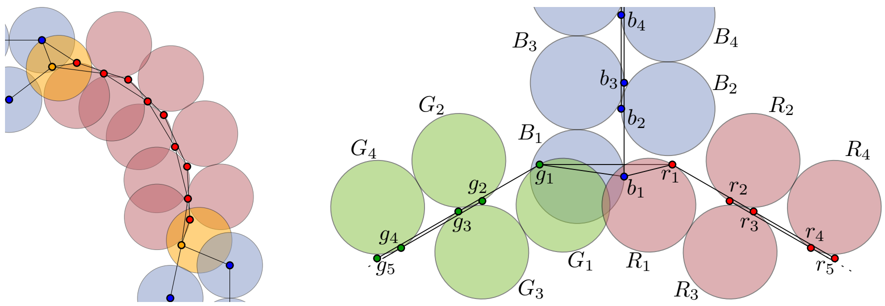
(a) **(b)**

Fig. 3: The variable gadget has two possible states: narrow and wide. It can also be used as an inverter as either the wire to the left or to the right has to be in narrow position.

upper path of disks and vice versa. We say these paths are in a *narrow* position. On the other hand, the paths on the right side are in a *wide* position: the upper path in the obedient plane straight-line drawing of the disk intersection graph belongs to the upper path of disks. By flipping the combinatorial embedding of the subgraph induced by the edges incident to the triangle vertices of the gadget, we can allow the left paths to be in wide position and the right path to be in narrow position; see Figure 3b. However, observe that (i) without flipping the embedding it is not possible to switch between the narrow and the wide positions and (ii) exactly one side has to be in narrow and the other in wide position in order to avoid edge-crossings. We shall use these two states of the variable gadget to encode the truth state of the corresponding variable. The parallel paths of disks to either side of the triangle act as *wires* that propagate the state of the variable gadget. Figure 4a illustrates the information propagation and shows that it is not necessary that the disk centers of the two parallel paths are collinear. Thus, wires are very flexible structures that allow us to transmit the truth states of the variable gadgets to other gadgets in our construction. Observe that our variable gadget can also be used as an *inverter*. If the narrow state is propagated to the triangle from one side, the other side is forced to propagate the wide state and vice versa.

Clauses. For each clause vertex we create a *clause* gadget as depicted in Figure 4b. Each of the three sets of disks $\{R_1, R_2, ...\}$, $\{G_1, G_2, ...\}$ and $\{B_1, B_2, ...\}$ belong to one wire. Note that if a wire is in narrow position, the position of the vertices of the wire is essentially *unique* up to an arbitrarily small wiggle room. The clause gadget is designed such that if all three of its wires are in narrow position the disk intersection graph can not be drawn obediently. The reason for this is that the unique positions of the vertices of the triangle $r_1g_1b_1$ in the middle of the gadget enforce a crossing, see Figure 4b. However, if at least one of the incident wires is in wide position and, thus, at least one of u, v or w can be placed freely in its disk, then the gadget can be drawn obediently.

Splitters. The final gadget in our construction is the *splitter* gadget, see Figure 5a. It works as follows. The three sets of disks $\{R_1, R_2, ...\}$, $\{G_1, G_2, ...\}$ and $\{B_1, B_2, ...\}$ belong to three wires r, g, b respectively. We also created a disk O which almost completely overlaps with G_2. Without the disk O the gadget would contain a vertex of degree 3 that is not part of a triangle. The disk O artificially creates a triangle so that the resulting disk intersection graph is thinnish. We refer to the wire b as the *input* of the gadget and to the wires r and g as the *outputs*. If the input is narrow then both outputs have to be wide due to the unique positions of the vertices in the narrow wires. However,



(a) **(b)**

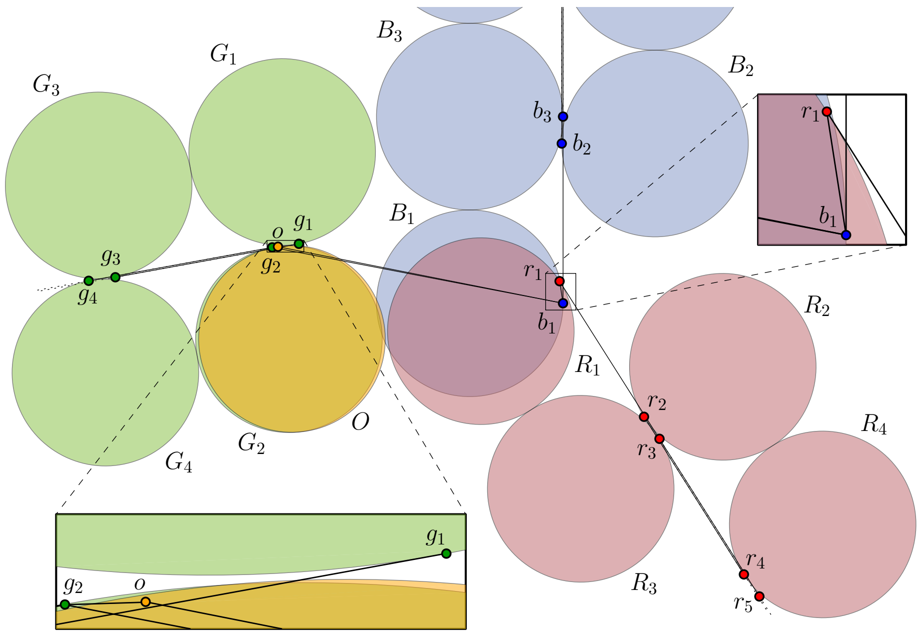
Fig. 4: **(a)** A curved wire that propagates the narrow position between orange disks. **(b)** The clause gadget.

if the input is wide, then any of the outputs can have any combination of states. For instance, both can be narrow. For our reduction we shall always connect the two outputs to inverters so that if the input of the splitter is narrow, then the inverted outputs are also narrow.

Layout. Figure 2b illustrates how we layout and combine our gadgets. We mimic the monotone rectilinear representation $\mathcal{R}$ for φ. We place the variable gadgets and clause gadgets according to $\mathcal{R}$. Consider a variable $u \in \mathcal{U}$. From the variable gadget of u one wire w_t leads to the top; another wire w_b leads to the bottom. If u occurs as a literal only once in a positive clause, the top wire w_t leads directly to the corresponding clause gadget. Otherwise, it leads to the input of a splitter. As stated earlier, we connect the outputs of the splitter to inverters. We split the resulting wires recursively until we have created as many wires as there are literals of u in positive clauses. We call the created wires the *children* of w_t and w_t is their *origin*. Similarly, the wire w_b that leads from the variable gadget of u to the bottom is connected to the negative clauses in which u occurs and the resulting wires are the *children* of w_b and w_b is their *origin*. Further, we refer to u as the *variable* of w_t and w_b. Note that while in some of our gadgets we require very precise coordinates, the required precision does not depend on the input size. Thus, the construction can be carried out in polynomial time.

Correctness. It remains to argue that our reduction is correct. Recall that if the input of a splitter is narrow then the outputs are wide. Since we place inverters at the outputs of each splitter it follows that all children of a narrow wire w are also narrow. Conversely, if a wire connected to a clause gadget is wide then it is a child of a wide wire.

Assume there exists an obedient plane straight-line drawing of the disk intersection graph of the set of disks we created. We create a satisfying truth assignment for φ. In an obedient plane straight-line drawing, for each clause gadget c there is at least one wire w connected to c that is wide; otherwise there is a crossing in the subdrawing of the clause gadget. Consequently, the origin of w is wide as well. If c is positive, we set the variable of w to true and if c is negative we set the variable of w to false. Thus, we have created a truth assignment in which for each clause, there is at least one satisfied literal. Note that it is not possible that we set a variable to both true and false since a wire can



(a)

Fig. 5: The splitter gadget.

only be the origin of either positive of negative clauses due to the fact that in a monotone rectilinear representation all negative clauses are below and all positive clauses are above the variable line.

Finally, assume that φ is satisfiable. For each variable u we orient its variable gadget such that the wire that leads to the positive clauses is wide if and only if u is true. We draw the splitter gadgets such that all children of wide wires are wide. Since every clause has a satisfied literal, the corresponding clause gadget is connected to a wide wire and, thus, can be drawn without introducing crossing.

Spacial Separation. Note that the splitter, variable, clause and inverter gadgets each contain one triangle and recall that all gadgets are connected by wires, which do not contain triangles. Thus, it is straightforward to ensure a minimum distance between any two triangles by simply increasing the length of our wires accordingly. □

3 Disk Intersection Graphs with Light Triangles

In Section 2 we give an $\mathcal{NP}$-hardness proof for a very restricted class of instances of PLANAR DISK OBEDIENCE RECOGNITION. A key ingredient of our reduction from 3SAT is a triangle in the middle of each variable gadget. The two truth states of the gadget are encoded by the combinatorial embedding of the subgraph induced by edges incident to the triangle vertices. It seems natural to study recognition of disk obedience graphs where the degree of vertices in triangles and, thus, the number of combinatorial

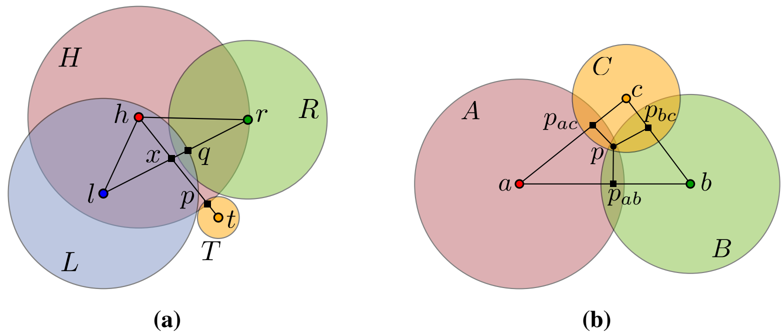


Fig. 6: (a) Every crossing between centered edges implies the existence of an arrow. **(b)** In a centered drawing, every point in a ply-3 triangle abc belongs to one of the three disks.

embeddings for the triangle-induced subgraphs is bounded. This motivates the following definition. Let $\mathcal{D}$ be a set of disks and $G = (V, E)$ be the intersection graph of $\mathcal{D}$. A triangle abc of G is called *light* if $\deg(a), \deg(b), \deg(c) \leq 3$ or if $A \cap B \cap C = \emptyset$. We say that $\mathcal{D}$ is *light* if every triangle of G is light. In this section we show that for any light set $\mathcal{D}$ of disks there always exists an obedient plane straight-line drawing of the intersection graph of $\mathcal{D}$. Note that we do not require the disks in $\mathcal{D}$ to have unit radius.

We begin by introducing some notations and by stating some helpful observations. A set of disks is *connected* if the union of all disks is connected. A set $\mathcal{D}$ of disks is said to be in *general position* if for any connected subset $\mathcal{D}' \subseteq \mathcal{D}$, $|\mathcal{D}'| = 3$ the disk centers of $\mathcal{D}'$ are non-collinear. Let $G = (V, E)$ be the intersection graph of a set of disks $\mathcal{D}$ and Γ be a straight-line drawing of G. A vertex $v \in V$ that is placed at its respective disk's center in Γ is called *centered*. An edge $e \in E$ between two centered vertices in Γ is called *centered*. The drawing Γ is called *centered* if all vertices in V are centered. An *arrow* (h, t, l, r) in a straight-line drawing Γ of a graph $G = (V, E)$ is a sequence of vertices $h, t, l, r \in V$ such that $ht, hl, hr, lr \in E$ and such that ht and lr cross in Γ, see Figure 6a. We refer to h, t, l, r as the arrow's *head*, *tail*, *left* and *right* vertex respectively.

Evans et al. [8] show that any set of disks with ply 2 in general position admits an obedient plane straight-line drawing. We observe that with some minor adaptations, their observation furthermore yields an explicit statement regarding the graph structure in non-plane centered drawings. We restate their proof together with our modifications to show that every crossing between centered edges implies the existence of an arrow. Furthermore, we strengthen this statement by showing that if the intersection point x of the two edges is contained in the interior of one of the corresponding disks, then there always is an arrow whose head's disk contains x in its interior.

Lemma 1. *Let $G = (V, E)$ be the disk intersection graph of a set of disks $\mathcal{D}$ in general position. Let Γ be a straight-line drawing of G. For any crossing in Γ between centered edges ab and cd there exists an arrow (h, t, l, r) where $H \cap L \cap R \cap lr \neq \emptyset$ and either (i) $ht = ab$ and $lr = cd$, or (ii) $ht = cd$ and $lr = ab$. Furthermore, if $x = ab \cap cd$ is contained in the interior of at least one of A, B, C, D, then $x \in \operatorname{int}(H)$.*

Next, we observe that if $A \cap B \cap C \neq \emptyset$, then in a centered drawing any point in the triangle abc is contained in one of the disks A, B or C.

Observation 1. *Let $\mathcal{D} = \{A, B, C\}$ be a set of disks whose intersection graph G is the triangle abc and let Γ be a centered drawing of G. If the closed triangle abc contains a point $p \in A \cap B \cap C$ then any point q in the closed triangle abc is contained in $A \cup B \cup C$.*

Our final auxiliary result states that under certain conditions, the number of crossings along one edge in a centered drawing is at most one.

Lemma 2. *Let $\mathcal{D}$ be a light set of disks in general position and let $G = (V, E)$ be the intersection graph of $\mathcal{D}$. Let Γ be a straight-line drawing of G. Let (a, b, c, d) be an arrow in Γ where a, b, c, d are centered and $A \cap C \cap D \cap cd \neq \emptyset$. Then there exists no centered edge other than ab that crosses cd in Γ .*

Proof Sketch. According to the definition of an arrow, we know that $ab, cd, ac, ad \in E$, see Figure 7c (left). Assume that some centered edge $fg \neq ab$ crosses cd. According to Lemma 1, the intersection of fg and cd implies existence of an arrow $\mathcal{A}$ consisting of c, d and two other vertices f and g, at least one of which has to be different from a and b. Since acd is a light triangle in G and since $A \cap C \cap D \neq \emptyset$, the degree of a, c, d is bounded by 3. Due to $\deg(a) \leq 3$ the vertex a can not be connected to any vertex other than b, c, d, which means that a cannot be an endpoint of the edge fg. Note that b could be equal to f or g. We perform a case distinction regarding the containment of the edges bc and bd in E.

First, assume that $bc \in E$ and $bd \in E$, see Figure 7a (left). Because of the degree restrictions for c and d, neither of them can be adjacent to a vertex different from a, b, c, d. Thus, cd can not be the head-tail edge of $\mathcal{A}$ since the head of $\mathcal{A}$ is connected to f and g. If cd is the left-right edge of $\mathcal{A}$, then the head of $\mathcal{A}$ has to be b and without loss of generality f is equal to b. Due to Lemma 1 we know that $B \cap C \cap D \neq \emptyset$ and, thus, since bcd is light the degree of b is bounded by 3. However, this is a contradiction to the fact that b is connected to g.

Now, assume that $bc \in E$ and $bd \notin E$, see Figure 7b. Similar to the last case, the degree restriction for c implies that c can not be adjacent to any vertex other than a, b, d. This implies that c can not be the head of $\mathcal{A}$ and it can be the right or the left vertex of $\mathcal{A}$ only if b is the head of $\mathcal{A}$. In this case cd is the left-right edge of $\mathcal{A}$ and d has to be connected to the head b of $\mathcal{A}$, which contradicts $bd \notin E$. Thus, c can only be the tail of $\mathcal{A}$, and so d is the head. The head d is connected to f and g both of which have to be different from a and b due to $\deg(a) \leq 3$ and due to $bd \notin E$ respectively. This is a contradiction to $\deg(d) \leq 3$.

We sketch the final case. Assume $bc \notin E$ and $bd \notin E$, see Figure 7c (left). Due to degree restrictions we see that cd is the left-right edge of $\mathcal{A}$ and w.l.o.g. $g \neq b$ is the head. Head g can not be located inside triangle acd since this would imply an additional crossing with ab, which contradicts degree restrictions. If g is exterior to acd and f is interior, Observation 1 implies a contradiction to the degree bounds of a, c, d. If g and f are exterior to acd, fg has to cross ad or ac, which implies the existence of another arrow, again contradicting degree restrictions. $\square$

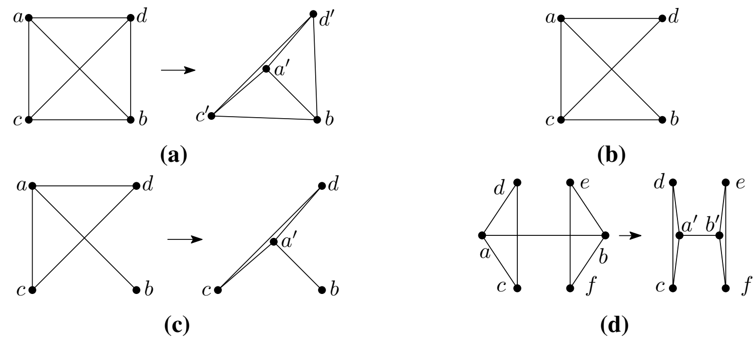


Fig. 7: Removing crossing by moving vertices.

Theorem 2. *Let $\mathcal{D}$ be a light set of disks in general position whose intersection graph is G. Then G has a plane straight-line drawing obedient to $\mathcal{D}$.*

Proof. We describe an iterative approach that transforms a centered drawing of G into a crossing-free drawing obedient to $\mathcal{D}$. In each *step* we change the position of precisely one, two or three vertices to remove one or two crossings. During the entire procedure, each vertex is moved at most once. We maintain the following *invariant*: After and before each step, all crossing edges are centered. We proceed by describing our algorithm. After that we show that the invariant is maintained and, thus, that the algorithm is correct.

Algorithm. Let ab and cd be two centered edges that cross in a point x. By Lemma 1 there exists an arrow consisting of a, b, c, d, w.l.o.g. (a, b, c, d), where $A \cap C \cap D \cap cd \neq \emptyset$, see Figure 7c (left). Note that this implies that the degree of a, c, d is bounded by 3 since $\mathcal{D}$ is light and since $A \cap C \cap D \neq \emptyset$. In order to remove the crossing we move some of a, b, c, d and we use a', b', c', d' to denote the new postions of these vertices.

We distinguish two cases. First assume that $x \in A \cap B \cap C \cap D$ and $x \notin \mathrm{int}(A) \cup \mathrm{int}(B) \cup \mathrm{int}(C) \cup \mathrm{int}(D)$, i.e. x is on the boundary of all four disks. In this case we set $a' = x$ and we move vertices c, d by a distance $\varepsilon \in \mathbb{R}^+$ in the direction given by the vector $\overrightarrow{ba}$, see Figure 7a. Value ε should be chosen small enough such that $c' \in C$, $d' \in D$ and $c'd'$ does not cross an edge ef, unless cd already crosses ef.

Now we consider the case that x is in the interior of at least one of A, B, C, D. By Lemma 1 we can assume without loss of generality that $x \in \mathrm{int}(A)$. In order to remove the crossing we move a in the direction given by the vector $\overrightarrow{ab}$ for distance $|ax| + \varepsilon < |ab|$, see Figure 7c. The value $\varepsilon \in \mathbb{R}^+$ should be chosen small enough such that $a' \in A$ and such that there is no crossing between $a'c$ or $a'd$ and any other edge ef, unless ef also intersects cd. To shorten notation, we refer to this procedure as 'removing the crossing by moving a to x', although technically a' is close to x but $a' \neq x$. It remains to treat the special case that ab has a crossing with some additional edge $ef \neq cd$. In this case, in addition to moving a to x, we move b to $y = ab \cap ef$, see Figure 7d. In the following paragraphs we will see that there exists at most one such additional crossing edge ef and that e, f are distinct from c, d as illustrated.

Correctness. Clearly our invariant holds for the initial centered drawing. In order to show that our invariant is maintained it suffices to show that none of the moved vertices is incident to an edge that has a crossing. In our algorithm we considered two main cases. In the first case we moved vertices a, c and d, which formed a complete graph with vertex b. Note that all these vertices have exactly degree three due to the degree restrictions. Therefore none of them can be adjacent to a vertex different from a, b, c, d. In the second case we moved vertex a, which is adjacent to exactly b, c and d due the degree restriction on a. Therefore, in both cases the moved vertices can only be incident to edges with crossing if there exists some edge $ef \neq ab, cd$ that has a crossing with ab or cd. According to Lemma 2 there is no edge that crosses cd except for ab. Due to Lemma 1, if ef crosses ab, there exists an arrow $\mathcal{A}$ composed of the edge ef and ab. According to Lemma 2, ab can not be the left-right edge of $\mathcal{A}$. The degree of a is bounded by 3 and, thus, a has to be the tail, b has to be the head and e, f have to be the left and right vertex of $\mathcal{A}$.

We perform a case distinction regarding the equivalence of e, f and c, d. First, asumme that $\{e, f\} = \{c, d\}$ and without loss of generality $e = d$ and $f = c$. Then $\mathcal{A}$ consists of a, b, c, d. Next, assume that exactly one of e, f is equal to one of c, d and without loss of generality $e = d$ and $f \neq a, b, c, d$. Then d is adjacent to a, c, b, f, which contradicts the degree bound for d. Finally assume that both e, f are distinct from a, b, c, d. By Lemma 2, neither cd nor ef can have a crossing with any edge other than ab. Hence, the situation looks like the one illustrated in Figure 7d. In this case, in addition to moving a to x, we move b to $y = ab \cap ef$ as described above. Now, $a'b'$ does not have any crossing and, thus, a' is not incident to an edge with a crossing. For symmetric reasons, neither is b'. □

4 Embedded Unit Disk Intersection Graphs

In Section 2 we proved that PLANAR DISK OBEDIENCE RECOGNITION is $\mathcal{NP}$-hard even for disk intersection graphs that are unit and thinnish. In the reduction from 3SAT used for the proof, the truth state of a variable gadget corresponds to the combinatorial embedding of the respective subgraph. The $\mathcal{NP}$-hardness proof by Evans et al. [8] also establishes a correspondence between truth states and combinatorial embeddings. This raises the question, whether $\mathcal{NP}$-hardness holds if a combinatorial embedding is prescribed. The following theorem answers this question in the affirmative. On a high level, the proof idea is a reduction from PLANAR MONOTONE 3-SATISFIABILITY similar to the one in the proof of Theorem 1. However, for the reduction in Theorem 3 we have to heavily rely on geometric arguments rather than combinatorial embeddings to encode the truth states of variable and wire gadgets.

Theorem 3. PLANAR DISK OBEDIENCE RECOGNITION *is $\mathcal{NP}$-hard even for embedded unit disk intersection graphs.*

5 Remarks and Open Problems

Other Shapes. The notion of obedient drawings naturally extends to other shapes. The reduction strategies used in the hardness proofs in this paper and the paper by Evans et al. [8] seem to apply for several other shapes as well, e.g., for unit squares. This raises the interesting question whether a more general statement can be made that captures all these hardness results at once.

NP-Membership. For many combinatorial problems, showing $\mathcal{NP}$-membership is an easy exercise. For disk-obedience the question turns out to be much more intricate. A naive idea to show NP-membership would be to guess the coordinates of all vertices. However, it is not obvious that there always exists a rational representation of bounded precision. Indeed, there are several geometric problems where this approach is known to fail. In some cases an explicit rational representation may require an exponential number of bits, in others optimal solutions may require irrational coordinates, see [1, 3, 14]. Many problems initially not known to lie in $\mathcal{NP}$ turned out to be $\exists\mathbb{R}$-complete. The complexity class $\exists\mathbb{R}$ captures all computational problems that are equivalent under polynomial time reductions to the satisfiability of arbitrary polynomial equations and inequalities over the reals, see [6, 14]. We leave it as an open problem to determine the relation of disk obedient plane straight-line drawings with respect to $\mathcal{NP}$ and $\exists\mathbb{R}$.

Acknowledgments. This work was initiated during the *Fixed-Parameter Computational Geometry* Workshop at Lorentz Center, April 4–8, 2016. We thank the organizers and all participants for the productive and positive atmosphere.

References

1. Abrahamsen, M., Adamaszek, A., Miltzow, T.: Irrational guards are sometimes needed. arXiv preprint arXiv:1701.05475 (2017)
2. Angelini, P., Lozzo, G.D., Bartolomeo, M.D., Battista, G.D., Hong, S., Patrignani, M., Roselli, V.: Anchored drawings of planar graphs. In: Graph Drawing. Lecture Notes in Computer Science, vol. 8871, pp. 404–415. Springer (2014)
3. Bajaj, C.: The algebraic degree of geometric optimization problems. Discrete & Computational Geometry 3(2), 177–191 (1988)
4. de Berg, M., Khosravi, A.: Optimal binary space partitions for segments in the plane. Int. J. Comput. Geometry Appl. 22(3), 187–206 (2012)
5. Bose, P., Morin, P., Stojmenović, I., Urrutia, J.: Routing with guaranteed delivery in ad hoc wireless networks. Wireless networks 7(6), 609–616 (2001)
6. Cardinal, J.: Computational geometry column 62. SIGACT News 46(4), 69–78 (2015)
7. Evans, W., Kirkpatrick, D., Löffler, M., Staals, F.: Minimizing co-location potential of moving entities. SIAM J. Comput. 45(5), 1870–1893 (2016)
8. Evans, W.S., van Garderen, M., Löffler, M., Polishchuk, V.: Recognizing a DOG is hard, but not when it is thin and unit. In: FUN 2016. pp. 16:1–16:12 (2016)
9. Keszegh, B., Pach, J., Pálvölgyi, D.: Drawing planar graphs of bounded degree with few slopes. SIAM Journal on Discrete Mathematics 27(2), 1171–1183 (2013)
10. Kröller, A., Fekete, S.P., Pfisterer, D., Fischer, S.: Deterministic boundary recognition and topology extraction for large sensor networks. In: SoDA 2006. pp. 1000–1009 (2006)
11. Kuhn, F., Moscibroda, T., Wattenhofer, R.: Unit disk graph approximation. In: Proceedings of the 2004 joint workshop on Foundations of mobile computing. pp. 17–23. ACM (2004)
12. Kuhn, F., Wattenhofer, R., Zhang, Y., Zollinger, A.: Geometric ad-hoc routing: of theory and practice. In: PoDC. pp. 63–72 (2003)
13. Löffler, M.: Data Imprecision in Computational Geometry. Ph.D. thesis, Utrecht University (2009)
14. Matoušek, J.: Intersection graphs of segments and $\exists\mathbb{R}$. arXiv preprint arXiv:1406.2636 (2014)
15. Wang, Y., Gao, J., Mitchell, J.S.B.: Boundary recognition in sensor networks by topological methods. In: MobiCom 2006. pp. 122–133 (2006)