RECOGNIZING WEIGHTED AND SEEDED DISK GRAPHS*

Boris Klemz,† *Martin Nöllenburg,*‡ *and Roman Prutkin*§

Abstract. Disk intersection representations realize graphs by mapping vertices bijectively to disks in the plane such that two disks intersect each other if and only if the corresponding vertices are adjacent in the graph. If intersections are restricted to touching points of the boundaries, we call them disk contact representations. Deciding whether a vertex-weighted planar graph can be realized such that the disks' radii coincide with the vertex weights is known to be NP-hard for both contact and intersection representations. In this work, we reduce the gap between hardness and tractability by analyzing the problem for special graph classes. We show that in the contact scenario it remains NP-hard for outerplanar graphs with unit weights and for stars with arbitrary weights, strengthening the previous hardness results. On the positive side, we present a constructive linear-time recognition algorithm for embedded stars with arbitrary weights.

We also consider a version of the problem in which the disks of a representation are supposed to cover preassigned points, called seeds. We show that both for contact and intersection representations this problem is NP-hard for unit weights even if the given graph is a path. If the disks' radii are not prescribed, the problem remains NP-hard for trees in the contact scenario.

1 Introduction

A set of disks in the plane is a *disk intersection representation* of a graph $G = (V, E)$ if there is a bijection between V and the set of disks such that two disks intersect if and only if they are adjacent in G. *Disk intersection graphs* are graphs that have a disk intersection representation; a subclass are *disk contact graphs* (also known as coin graphs), that is, graphs that have a disk intersection representation with interior-disjoint disks. This is also called a *disk contact representation* (DCR) or, if connected, a circle packing. It is easy to see that every disk contact graph is planar and the famous Koebe-Andreev-Thurston circle packing theorem [19] dating back to 1936 (see Stephenson [24] for its history) states that the converse is also true, that is, every planar graph is a disk contact graph.

Application areas for disk intersection/contact graphs include modeling physical problems like wireless communication networks [15], covering problems like geometric facility location [23, 25], visual representation problems like area cartograms [13] and many

*A preliminary version of this paper appeared in the proceedings of Graph Drawing and Network Visualization 2015 [17].

†*Universität Würzburg, Germany*, firstname[dot]lastname[at]uni[dash]wuerzburg[dot]de

‡*TU Wien, Austria*, noellenburg@ac.tuwien.ac.at

§*Karlsruhe Institute of Technology, Germany*, roman.pru@gmail.com

more (various examples are given by Clark et al. [9]). Efficient numerical construction of DCRs has been studied in the past [11, 21]. Often, however, one is interested in recognizing disk graphs or generating representations that do not only realize the input graph, but also satisfy additional requirements. For example, Alam et al. [1] obtained several positive and negative results on the existence of balanced DCRs, in which the ratio of the largest disk radius to the smallest is polynomial in the number of disks. Furthermore, it might be desirable to generate a disk representation that realizes a vertex-weighted graph such that the disks' radii or areas are proportional to the corresponding vertex weights, for example, for value-by-area circle cartograms [16]. Clearly, there exist vertex-weighted planar graphs that cannot be realized as disk contact representations, and the corresponding recognition problem for planar graphs is NP-hard, even if all vertices have the same weight [6]. However, the complexity of recognizing weighted disk contact graphs for many interesting subclasses of planar graphs remained open. Graphs realizable as DCRs with unit disks correspond to so-called 1-ply graphs. This was stated by Di Giacomo et al. [12] who introduced and studied the ply number concept for graphs. They showed that internally triangulated biconnected planar graphs admitting a DCR with unit disks can be recognized in $O(n \log n)$ time. Bowen et al. [5] showed that when the combinatorial embedding is prescribed, the problem is NP-hard even for trees. However, the recognition of trees with a unit disk contact representation remains an interesting open problem if arbitrary embeddings are allowed.[1]

Another interesting problem is the generation of *seeded* disk graph representations in which each disk is required to cover a point in the plane, called *seed*, that is preassigned to its corresponding vertex. Atienza et al. [3] showed that this problem is NP-hard in the contact scenario even if the input graph is outerplanar.

A further related concept are *weak* disk contact representations, where the disks of every two adjacent vertices in G must touch, but two touching disks do not necessarily imply that their vertices are adjacent in G. Chiu et al. [8] and Cleve [10] proved NP-hardness of recognizing weak unit disk contact graphs for embedded caterpillars and for non-embedded trees. This problem remains open in the non-weak setting.

Finally, for unit disk intersection graphs as a superclass of unit disk contact graphs, Bhore et al. [4] recently showed several results, including the NP-hardness of recognizing outerplanar graphs and embedded trees that admit a unit disk intersection representation. For caterpillars, however, they showed that the recognition problem can be solved in linear time, as well as for the related class of so-called lobster graphs in the weak contact model.

Results. In this paper, we examine the aforementioned scenarios more closely and explore disk contact representations for special graph classes. We extend the results of Breu and Kirkpatrick [6] and show that it remains NP-hard to decide whether a DCR with unit disks exists even if the input graph is outerplanar (Section 3). For vertex weights that are not necessarily uniform we show that the recognition problem is strongly NP-hard even for stars (Section 4.1). In contrast, for *embedded* stars we solve the problem in linear time

[1] In a preliminary version of this article [17], we claimed this problem to be linear-time solvable for the subclass of caterpillars (i.e., trees for which a path remains after removing all leaves). We still conjecture that this is true, but our original proof was flawed and the required arguments appear to be more involved than anticipated, see also Section 6.

		non-embedded DCR		embedded DCR	
	graph class	unit disks	weighted disks	unit disks	weighted disks
non-seeded	planar	NP-hard [6]	⟶ NP-hard	NP-hard ↑	⟶ NP-hard ↑
	outerplanar	NP-hard (Thm. 1)	⟶ NP-hard ↑	NP-hard ↑	⟶ NP-hard ↑
	trees	open	NP-hard ↑	NP-hard [5]	⟶ NP-hard
	caterpillars	open	NP-hard ↑	open	open
	stars	trivial	NP-hard (Thm. 2)	trivial	$O(n)$ (Thm. 3)
	paths	trivial	trivial	trivial	trivial

		contact representation		intersection representation
	graph class	arbitrary disks	unit disks	unit disks
seeded	planar	NP-hard ↑	NP-hard ↑	NP-hard ↑
	outerplanar	NP-hard [3]	NP-hard ↑	NP-hard ↑
	trees	NP-hard (Thm. 4)	NP-hard ↑	NP-hard ↑
	paths	open	NP-hard (Thm. 5)	NP-hard (Thm. 6)

Table 1: Overview of the state of the art, new results, and open problems on disk graph recognition. A cell with an arrow indicates a result that is an immediate consequence from the result in the cell at the arrow's tail.

(Section 4.2). This algorithm assumes a *Real RAM* model of computation where a set of basic arithmetic operations (including trigonometric functions and square roots) can be performed in constant time [22]. We strengthen the result by Atienza et al. [3] and show that the seeded version of the problem is NP-hard even for trees (Section 5.1). We also consider a combination of the weighted and seeded problem and show NP-hardness for unit weights even if the input graph is a path (Section 5.1). The result applies to contact, as well as intersection representations. Table 1 summarizes our results, the state of the art, and remaining open recognition problems.

2 Preliminaries

In this section we introduce the planar 3SAT problem and its variations, which will be used for some of our hardness reductions in this paper. A *planar 3SAT* (P3SAT) formula φ is a Boolean 3SAT formula with a set $\mathcal{U}$ of variables and a set $\mathcal{C}$ of clauses such that its *variable-clause-graph* $G_\varphi = (\mathcal{U} \cup \mathcal{C}, E)$ is planar. The set E contains for each clause $c \in \mathcal{C}$ the edge (c, x) if a literal of variable x occurs in c. Deciding the satisfiability of a P3SAT formula is NP-complete [20] and for every P3SAT formula φ there exists a planar drawing $\mathcal{G}_\varphi^R$ of G_φ on a grid of polynomial size such that the variable vertices are placed on a horizontal line ℓ and the clauses are connected in a comb-shaped rectangular fashion from above or below that line [18]. For technical reasons, we additionally add an edge between each pair of consecutive variables along ℓ, Figure 1a illustrates the result. To distinguish between the original edges and these additional edges, we refer to the former as *literal edges*. The drawing can furthermore be slanted to obtain a drawing $\mathcal{G}_\varphi^S$ in which all angles are multiples

of 60 degrees and vertices are placed on an isometric triangular grid [7]. A P3SAT formula φ is *monotone* if each clause contains either only positive or only negative literals and if G_φ has a planar drawing as described above with all clauses of positive literals on one side of the horizontal line with the variable vertices and all clauses of negative variables on the other side, as in Figure 1a. The 3SAT problem remains NP-complete for planar monotone formulae [20] and is called Planar Monotone 3-Satisfiability (PM3SAT).

3 Unit disk contact graphs

In this section we are concerned with the problem of deciding whether a given graph is a *unit disk contact graph* (UDC graph), that is, whether it has a DCR with unit disks. For a UDC graph we also say that it is *UDC-realizable* or simply *realizable*. It is known since 1998 that recognizing UDC graphs is generally NP-hard for planar graphs [6], but it remained open for which subclasses of planar graphs it can be solved in polynomial time and for which subclasses NP-hardness still holds. Here we show that the problem remains NP-hard for outerplanar graphs. For prominent graph classes below outerplanar graphs such as (non-embedded) trees the complexity of the recognition problem remains open, while for embedded trees it is known to be NP-hard [5].

A graph is *outerplanar* if it admits an *outerplane* drawing, that is, a planar drawing in which all vertices lie on the unbounded outer face. As usual, our reduction makes use of several types of gadgets. Arguing about these gadgets becomes a lot easier, if they admit a single unique disk contact representation (up to rotation, translation and mirroring). We call graphs with such a representation *rigid*. In the following lemma we state a sufficient condition for rigid UDC structures.

Lemma 1. *Let $G = (V, E)$ be a biconnected graph realizable as a UDC representation that induces an internally triangulated outerplane drawing of G. Then, G is rigid.*

Proof. Let $\mathcal{G}$ be a UDC representation of G that induces an internally triangulated outerplane drawing Γ of G. We prove the statement by induction on the number n of vertices. The smallest biconnected graph is a triangle which is obviously a rigid graph and, thus, the statement is true for $n = 3$.

For the induction step, consider any $n > 3$ and assume that our hypothesis holds true for all biconnected graphs with at most $n-1$ vertices. By assumption, G is outerplanar and thus there exists a vertex $v_r \in V$ with $\deg(v_r) \leq 2$. Since G is also biconnected it follows that $\deg(v_r) = 2$. Let $v_1, v_2 \in V$ be the neighbors of v_r. Since G is biconnected, the outer face of Γ is a simple cycle. Thus, since $n > 3$ and since the drawing is internally triangulated, v_1 and v_2 are adjacent, and the edge (v_1, v_2) is internal. Removing the disk corresponding to v_r from $\mathcal{G}$ yields a UDC representation $\mathcal{G}'$ that realizes the subgraph $G' = (V', E')$ of G that is induced by the vertex set $V' = V \setminus \{v_r\}$. The induced planar drawing of $\mathcal{G}'$ is obviously still outerplanar and internally triangulated. Moreover, since v_1 and v_2, the only neighbors of v_r, are adjacent, G' is biconnected. Thus, by the induction hypothesis, it is rigid.

We now re-insert the vertex v_r and edges $(v_r, v_1), (v_r, v_2)$ to G' (resulting in G) and add a corresponding disk D_r to $\mathcal{G}'$. Since v_1 and v_2 are adjacent, their disks touch and,

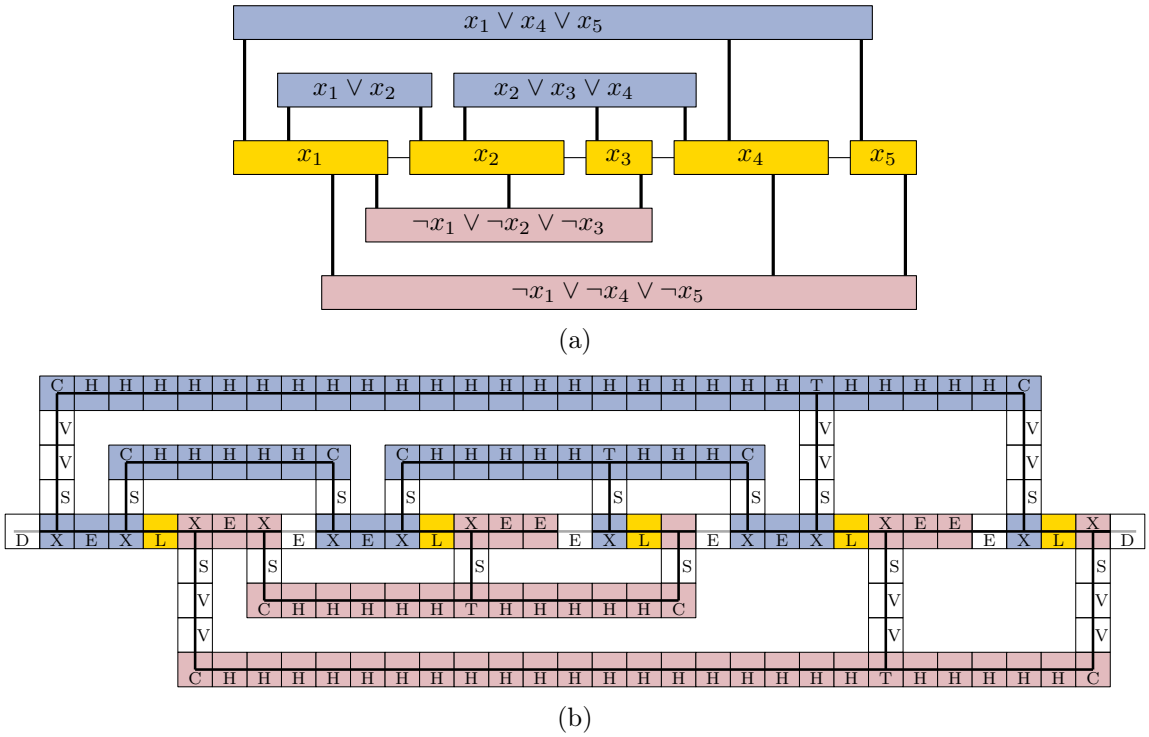


Figure 1: (a) Planar rectangular drawing $\mathcal{G}_\varphi^R$ of a PM3SAT formula φ. (b) Layout of the gadget tiles of our construction, which mimics $\mathcal{G}_\varphi^R$.

thus, there are exactly two possible locations p_r and p_l in the plane for a disk that touches the disks of both v_1 and v_2. Without loss of generality we may assume that p_r is the former location of D_r. Since (v_1, v_2) is an internal edge of Γ and since Γ is internally triangulated, there exists some vertex $v_l \neq v_r$ adjacent to both v_1 and v_2. In both $\mathcal{G}$ and $\mathcal{G}'$, the disk of v_l is placed at p_l. Hence the only possible placement for D_r in $\mathcal{G}'$ is p_r and, thus, G is rigid. □

We note that for a graph G satisfying the conditions of Lemma 1, every internal face is a triangle. The dual graph of the internal faces is a tree of maximum degree three, similarly to the dual graph of a triangulation of a simple polygon. We are now prepared to proceed with the proof of the main result of this section.

Theorem 1. *For outerplanar graphs the UDC recognition problem is* NP*-hard if an arbitrary embedding is allowed.*

Proof. We perform a polynomial time reduction from PM3SAT (see Section 2) to show the NP-hardness of recognizing outerplanar UDC graphs. For the reduction we create, based on the planar rectangular drawing $\mathcal{G}_\varphi^R$ of a PM3SAT formula φ (see Figure 1a) an outerplanar graph that has a UDC representation if and only if the formula φ is satisfiable.

For ease of presentation, we start by describing a reduction from PM3SAT to a more constrained variant of UDC recognition where the positions of a subset of the vertices may be

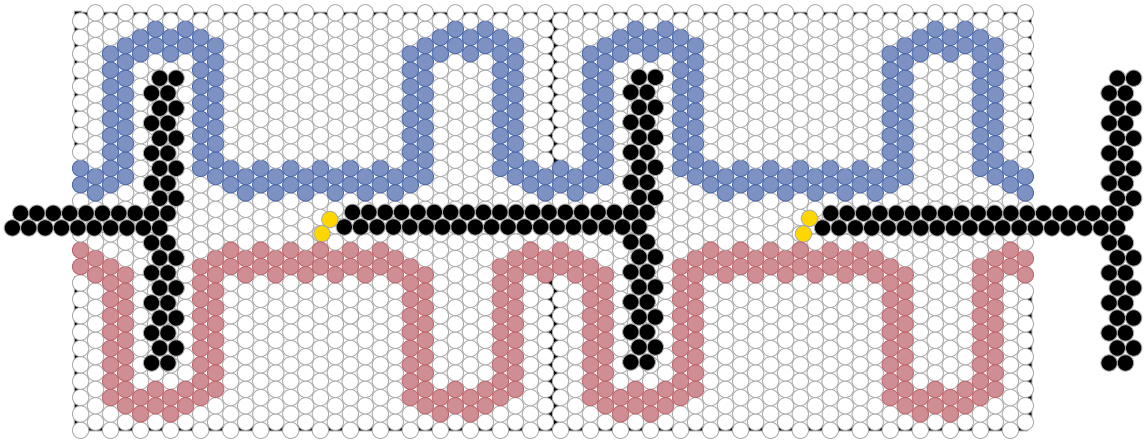


Figure 2: Two neighboring Type H wire gadgets, both of which are oriented to the right. This illustration is not up to scale: the underlying constant sized hexagonal grid of each tile actually needs to be larger to ensure that the other gadget types (in particular the Type T and Type X wire) can be realized. When scaling the Type-H wire to the correct grid size, the length of the vertical and horizontal "segment" of the T-shaped bar is increased accordingly, but its width remains fixed (≈ 2 times the disk diameter). Similarly, the length of the segments of the tunnel structure is increased, but the tunnels remain thin (≈ 4 times the disk diameter), so that the bar has only little wiggle room.

prescribed as part of the input. We call this problem *partial* UDC recognition (PUDC). We then modify the gadgets of our construction to extend our reduction to the standard version of UDC recognition, where the coordinates of the disks are no longer fixed in advance. We will use a packing argument to show that all the disks have to be placed near their intended locations, so that the idea for the correctness of the reduction to PUDC carries over.

Wire gadgets. The main building blocks of the reduction are *wire gadgets* that come in different variations. Each wire gadget occupies a rectangular tile designed on a hexagonal grid of fixed size and orientation (see, e.g., Figure 2) so that different tiles can be flexibly put together in a grid-like fashion (see Figure 1b) to mimic the layout of the planar rectangular drawing $\mathcal{G}_\varphi^R$ (see Figure 1a). Each of the wire gadgets consist of a rigid tunnel-like structure. Most of these gadgets contain a rigid bar that can be flipped into different tunnels around a centrally located disk. The bars stick out of the tiles in order to transfer information to the neighboring tiles, and longer chains of wire gadgets will be used to transport information between the other gadgets of our construction.

Figure 2 illustrates the UDC realization of two neighboring *Type H* (horizontal) wire gadgets. The red and blue disks form the aforementioned thin *tunnel* and their positions are considered to be fixed. By Lemma 1 the two underlying UDC graphs are rigid. The black disks in each of the tiles constitute the T-shaped *bar* of the wire gadget, whose underlying UDC graph is also rigid by Lemma 1. The positions of the disks of the bar are not fixed. However, combinatorially speaking, there are only two possible ways to place the bar since it is attached to a path consisting of two orange *chain* disks (whose positions are not fixed),

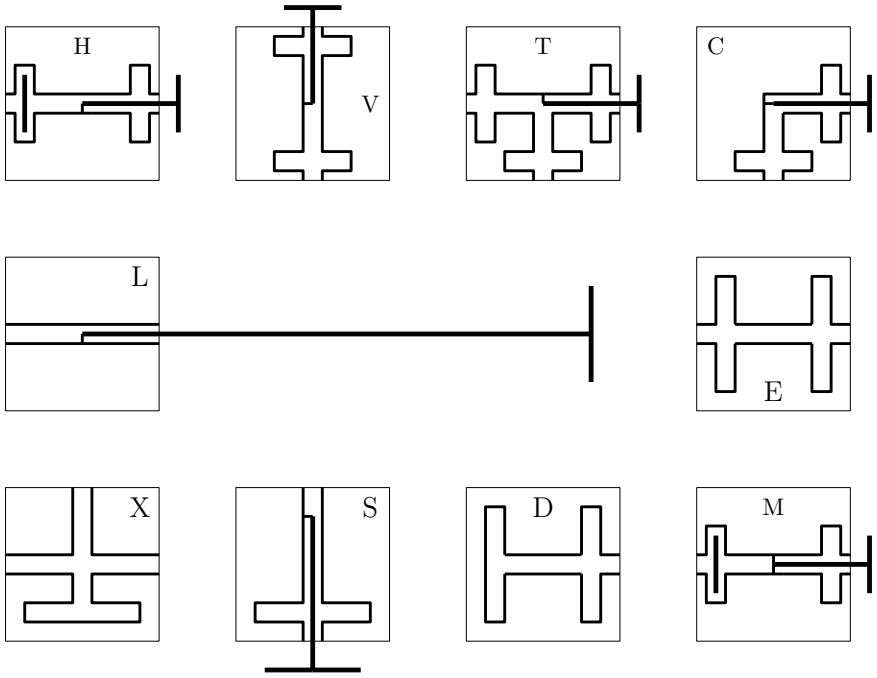


Figure 3: Overview of all wire types in our construction. Some of these tiles come in multiple orientations, e.g., there are four orientations of the Type C wire (bottom-left, bottom-right, top-left, and top-right).

that are attached to a red disk (whose position is fixed) near the center of the tunnel: the bar can be placed such that it sticks out to the right (as in Figure 2) or to the left of the gadget tile. We say that the wire is *oriented* to the left or right, respectively. If the left of the two wires is oriented to the right, the right wire has to be oriented to the right as well; otherwise, their bars would intersect. Conversely, if the right wire is oriented to the left, the left wire has to be oriented to the left as well. Hence, longer chains of consecutive Type H wire gadgets can be used to propagate information along a horizontal row of the grid.

To achieve more flexibility, we introduce further types of wire gadgets, all of which are designed according to the same principals as the Type H wire; for a schematic overview of all wire types see Figure 3. In particular, each tile contains tunnels formed by rigid subgraphs and the positions of the vertices of these subgraphs are considered to be fixed. The *Type V* (vertical) wire gadget works analogously to the Type H wire gadget, except that it propagates information vertically rather than horizontally, see Figure 4. Conceptually, the Type V wire gadget corresponds to a Type H wire rotated by 90°. However, to ensure that the orientation of the underlying hexagonal grid of all tiles is consistent (so that the tiles can indeed be put together in a grid-like fashion), the disks are arranged slightly differently. A *vertical* wire is a Type V wire or a Type S wire, which is a variant of a Type V wire defined below. Similarly, a *horizontal* wire is a Type H wire or a Type M wire, which is a variant of a Type H wire defined below. In a *Type T* wire gadget, the bar has to be placed in one of two horizontal tunnels or in a vertical tunnel, see Figure 4. The *Type C* (corner) wire gadget allows the transition from a horizontal to a vertical wire, see Figure 4.

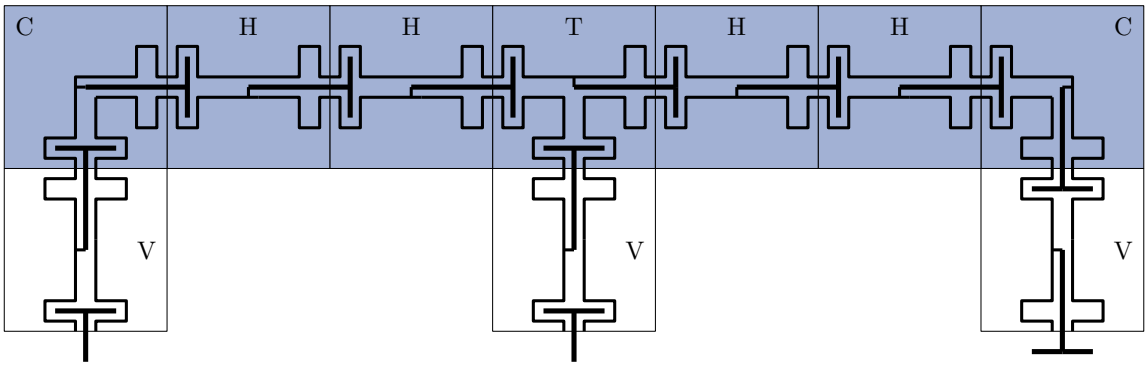


Figure 4: Schematic of a clause gadget.

Literal gadgets. For each literal edge of the variable-clause-graph G_φ of φ, we create a *literal gadget* which consists of several consecutive vertical wires, see Figure 1b. If the bottom-most (top-most) wire of the gadget is oriented towards the top (bottom), then all of its wires are oriented to the top (bottom). This corresponds to the two truth states of the literal.

Clause gadgets. We create a *clause gadget* for each clause c of φ. Each clause gadget consists of multiple consecutive wire gadgets in a horizontal grid row, see Figure 4. Specifically, the left-most and right-most wire is of Type C and attaches to a vertical wire belonging to a literal gadget. One of the wire gadgets between the two Type C wires is of Type T and the remaining wires are horizontal. The Type T wire also attaches to a vertical wire that belongs to a third literal gadget. Hence, the clause gadget is connected to a total of three distinct literal gadgets corresponding to the literal edges incident to c in G_φ. The Type T wire has to be oriented towards one of its neighboring tiles. Since this information is propagated, at least one of the three attached literal gadgets has to be oriented away from the clause gadget. This corresponds to the fact that at least one of the three literals of each clause has to be satisfied, e.g., in Figure 4 the literal corresponding to the right-most literal gadget attached to the clause has to be satisfied. (The above description assumes that the clause has three literals. If the clause has only two literals, we omit the Type T wire. If the clause has only one literal, its gadget consists of a modified vertical wire where one of the two sides is blocked so that the orientation of the wire is fixed. Alternatively, we could transform φ into an equivalent formula where every clause has exactly three literals.)

Variable gadgets. We create a *variable gadget* for each variable x of φ. Each variable gadget consists of an odd number s of consecutive tiles in a horizontal row of the tile grid. The central tile is a *Type L* (long) wire gadget, which is defined like a Type H wire gadget, except that its *long* bar stretches over $(s+1)/2$ tiles, rather than just one (see Figure 5). For each literal edge e incident to x in G_φ, there is one Type X wire X_e in the variable gadget; the remaining wires are of Type E. A *Type E* (empty) wire is defined as a Type H wire, except that it has no rigid bar and no chain disks, so that the long bar of the Type L wire can be embedded in its tunnel, see Figure 5. A *Type X* wire is similar to the

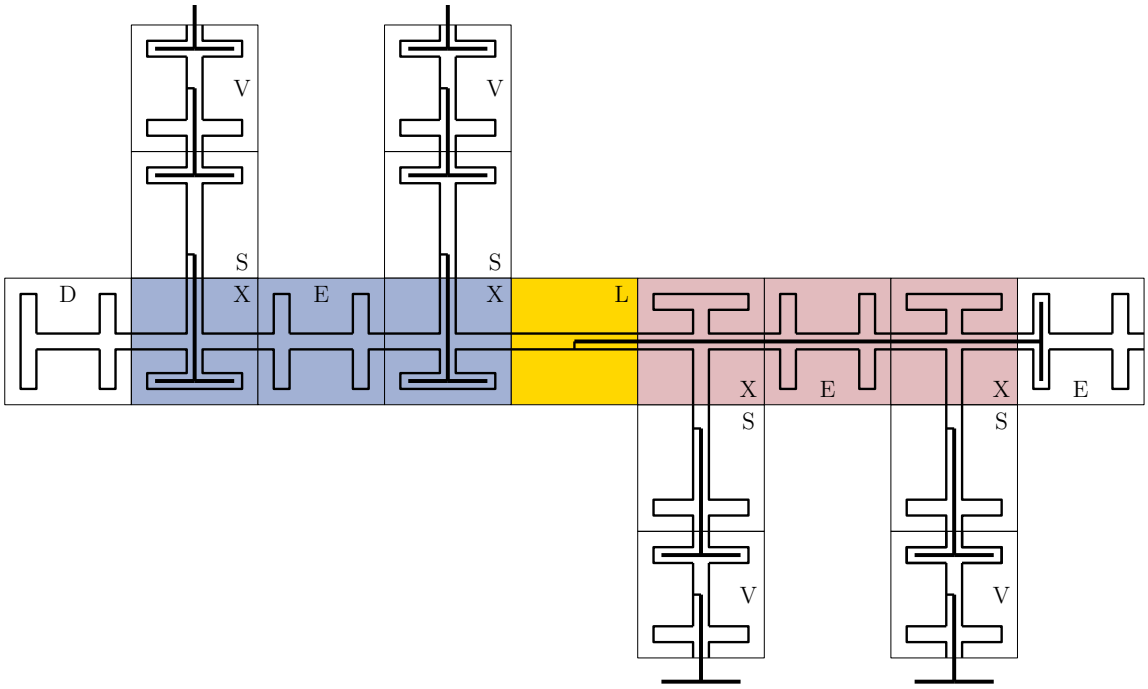


Figure 5: Schematic of a variable gadget.

Type T wire, but it contains no bar and no chain disks: it consist of a horizontal tunnel that allows the long bar of the Type L gadget to pass through the gadget tile and a vertical tunnel that attaches to the horizontal tunnel from above (or below) and extends a bit below (above) the horizontal tunnel, see Figure 5. The Type X wire X_e is located to the left of the Type-L wire if and only if e corresponds to a positive literal, and its vertical tunnel attaches to the literal gadget L_e corresponding to e. The unique vertical wire of L_e that attaches to the vertical tunnel of X_e is of Type S; the remaining (vertical) wires of L_e are of Type V. A *Type S* wire is defined as a Type V wire, except that its bar is a bit longer and the disk to which it is attached via the chain disks is shifted towards the attached variable gadget such that if the literal gadget L_e is oriented towards the variable gadget, the long bar of its Type L wire cannot be embedded in the tunnel of X_e without crossing the bar of the Type S wire, see Figure 5. Hence, if the bar of the Type L wire is oriented to the left (right), all literal gadgets of positive (negative) literals of x have to be oriented away from the variable gadget of x. This corresponds to the two truth states of x.

All variable gadgets are arranged on a common horizontal grid row. We fill the tiles between each pair of consecutive variable gadgets with Type E wires. To the left (right) of the left-most (right-most) variable gadget we place a *Type-D* (dead end) wire, which is defined like the Type E wire, except that one of its sides is blocked (see Figure 5, left).

This concludes our construction. Let G_1 denote the planar graph underlying our construction (we remark that G_1 not outerplanar; we fix this aspect below). Since each tile contains a constant number of disks and the number of tiles is polynomial (actually, linear) in the grid size of the planar rectangular drawing $\mathcal{G}^R_\varphi$, which is polynomial in the size of φ,

the size of G_1 is polynomial in the size of φ. We remark that we do not need to worry about encoding the prescribed disk coordinates since our ultimate goal is a reduction to UDC, where the disk coordinates are no longer fixed.

Correctness. We now show that G_1 has a PUDC realization if and only if the PM3SAT formula $\varphi = (\mathcal{U}, \mathcal{C})$ is satisfiable. If the latter is the case, there exists a truth assignment t for $\mathcal{U}$ that satisfies all clauses in $\mathcal{C}$, i.e., one literal of each clause in $\mathcal{C}$ is satisfied with respect to t. Each literal of a clause $c \in \mathcal{C}$ corresponds to one of the literal gadgets attached to the clause gadget representing c. A truth assignment for $\mathcal{U}$ induces an orientation for each of these gadgets: we orient the gadgets of satisfied literals towards their respective variable gadget and the gadgets of unsatisfied literals towards the clause gadget. Thus, t induces an orientation in which at least one of the literal gadgets attached to the clause gadget of c is oriented towards a variable gadget, which is a necessary and sufficient condition for the realizability of the clause gadget. Furthermore, a truth assignment induces an orientation for the long bar of each variable gadget. We orient a long bar to the right (left) if the corresponding variable is true (false). Recall that all clauses above (below) the horizontal line of variables contain exclusively positive (negative) literals and that the literal gadgets that represent these literals are attached to the left (right) side of each variable gadget. Each satisfied literal gadget L_e is oriented towards its respective variable gadget, thus, the bar of its Type S wire sticks into the Type X wire X_e of the variable gadget. This does not interfere with the realizability of the variable gadgets' subgraphs due to the fact that the orientation of long bars induced by t is chosen such that these bars only prevent gadgets of unsatisfied literals to be oriented towards variable gadgets. The unsatisfied literal wires have to be oriented towards their respective clause gadgets in accordance with the orientation (induced by t) of the vertical wires incident to the clause gadgets. Thus, if φ is satisfiable, G_1 is realizable.

On the other hand, if G_1 is realizable, the orientation of the long bars of the variable gadget induces a satisfying truth assignment t: if such a bar is oriented to the right, we set the variable to true; otherwise to false. Each positive (negative) clause gadget is attached to at least one literal gadget that is oriented away from the clause gadget, which is only possible only if the long bar of the variable gadget attached to the literal gadget is oriented to the right (left), implying that according to t, the corresponding literal is true. Hence, each clause is satisfied.

We proceed by modifying our construction to prepare for the transition to the UDC problem.

Outerplanarity and connectivity. In this paragraph, we show that G_1 is easily turned into an equivalent PUDC instance that is outerplanar. Moreover, we do so while also making the graph connected. The latter property is important to establish some control over the placement of its maximal rigid subgraphs when going from PUDC to UDC.

We begin by studying the structure of G_1. By construction, the subgraph of G_1 induced by the vertices whose positions are fixed (i.e., the vertices that form the tunnel structure) has a PUDC representation. Consider the (unique) plane subgraph Γ_1 induced

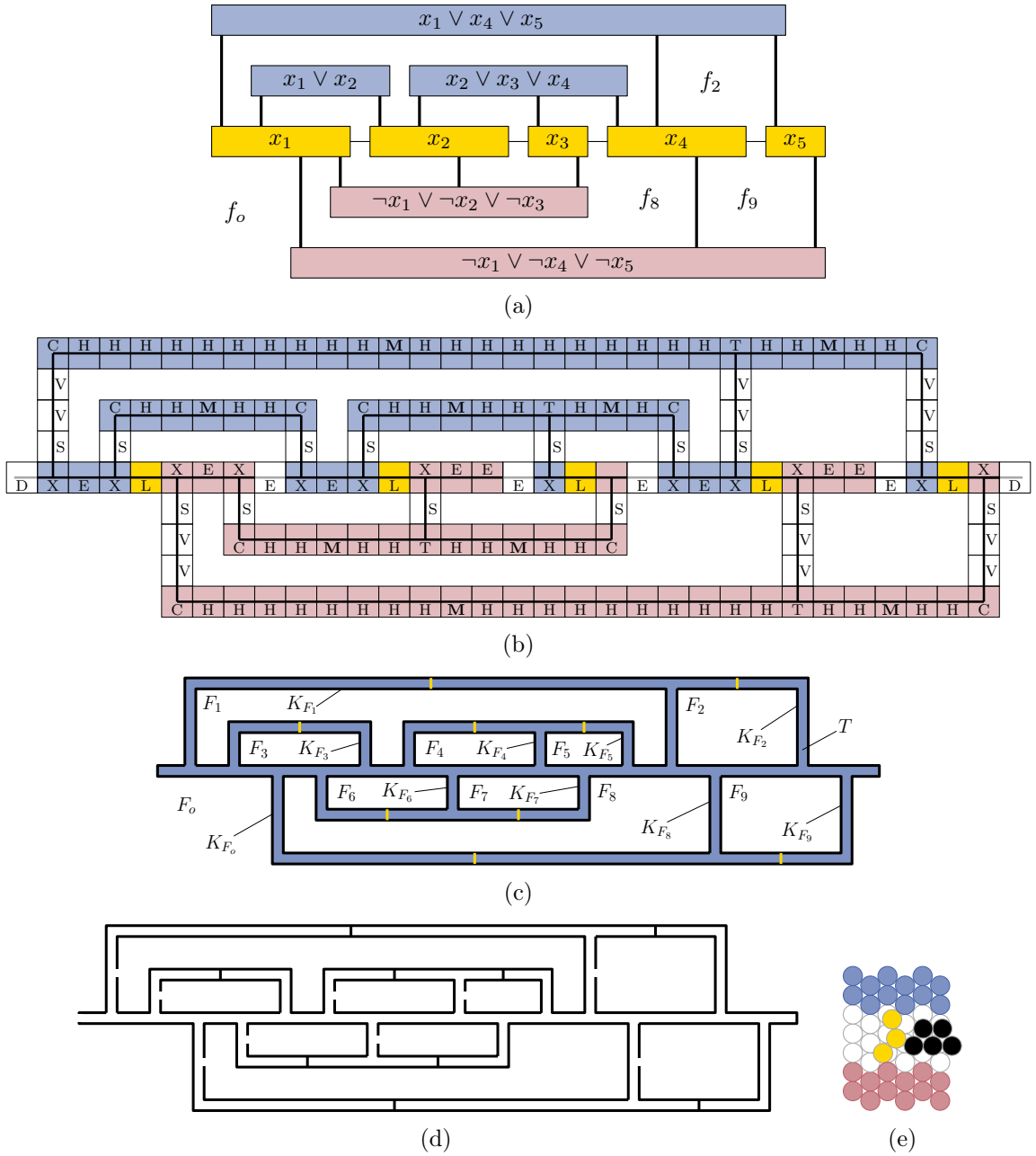


Figure 6: Example of our reduction on multiple levels of detail. (a) Planar rectangular drawing $\mathcal{G}_\varphi^R$ of a PM3SAT formula φ (with selected faces labeled to make the correspondence to Subfigure (c) apparent). (b) Layout of the gadget tiles of our construction, which mimics $\mathcal{G}_\varphi^R$. (c) Structure of the plane graphs Γ_1 and Γ_2. Each of the white faces corresponds to a face in $\mathcal{G}_\varphi^R$. The blue face corresponds to the interior of the tunnels. The short orange lines indicate the paths added by the Type M wires. (d) Structure of the plane graph Γ_2'. (e) The Type M wire is identical to the Type H wire, except that an additional chain disk is added to establish a connection between the red and the blue tunnel boundary.

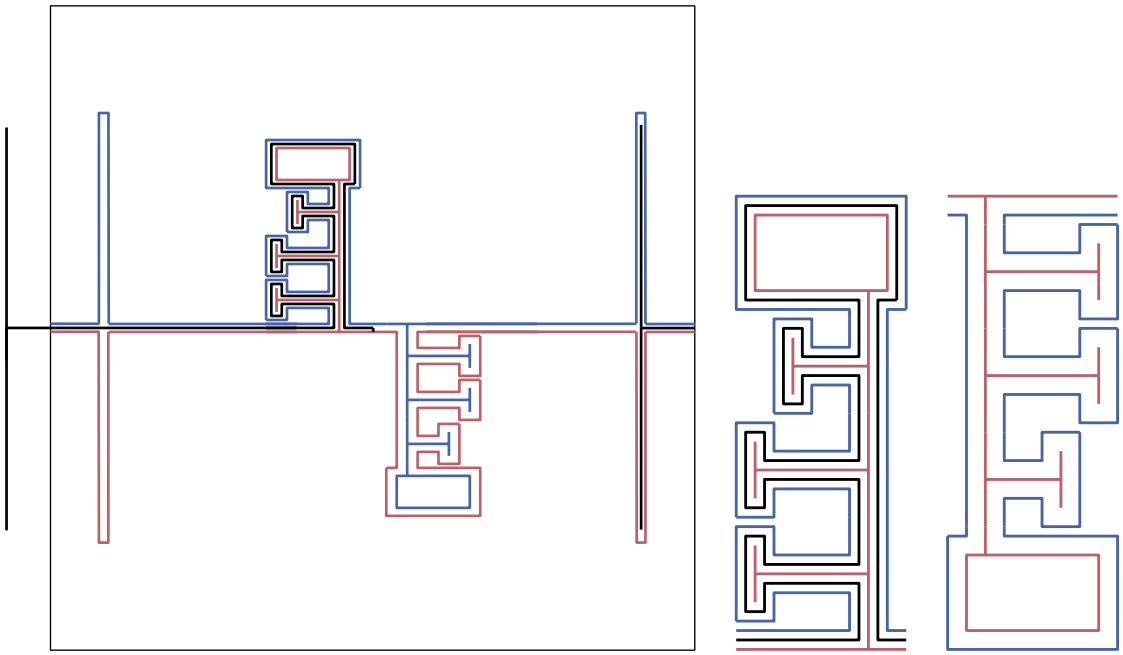
by this representation. Each face f of the planar rectangular drawing $\mathcal{G}_\varphi^R$ (including the outer face) corresponds to a face F in Γ_1 whose boundary is described by a connected component K_F of Γ_1, see Figures 6a and 6c (white). Let $\mathcal{F}$ denote the collection of these faces F of Γ_1. Moreover, Γ_1 contains a face T that corresponds to the interior of the tunnels (the region where the bars and chain disks are placed), see Figure 6c (blue). The remaining faces of Γ_1 are triangular. By construction, each of the vertices of such a triangular face is also incident to T or some $F \in \mathcal{F}$.

We now show how to make G_1 connected. To this end, we introduce the *Type M* (merging) wire, whose definition is analogous to that of the Type H wire, except that we add another chain disk that, together with the other two chain disks, forms a path that connects the two subgraphs forming the tunnel, see Figure 6e. The position of the new disks is not considered to be fixed. Hence, the path of chain disks is a bit flexible and the bar of the Type M wire can be embedded such that it sticks out either to the right or to the left of the gadget tile, just like in a regular Type H wire. For each maximal sequence of consecutive Type H wires (which only occur in the clause gadgets), we pick an arbitrary Type H wire and replace it by a Type M wire, see Figure 6b. The resulting graph G_2 is connected (see Figure 6c) and PUDC realizable if and only if G_1 is PUDC realizable.

Next, we show how to make our instances outerplanar. Let Γ_2 denote the induced plane graph of a PUDC representation of the subgraph of G_2 induced by the union of the vertices whose positions are fixed and the vertices that correspond to chain disks (i.e., we consider all disks except for those of the bars). Note that when going from Γ_1 to Γ_2 (by adding the paths formed by the chain disks) the set of faces does not change, that is, for each path of chain disks, we have that both the face to its left and the face to its right is the face T that corresponds to the interior of the tunnels, see Figure 6c. Let $F \in \mathcal{F}$. Pick a pair of adjacent vertices u, v in K_F such that one of them is incident to F and one of them is incident to T and such that none of them is adjacent to a vertex corresponding to a chain disk. Removing u and v merges F and T. Moreover $K'_F = K_F \setminus \{u, v\}$ is still rigid by Lemma 1. Hence, by repeating this step for each $F \in \mathcal{F}$, we obtain a connected outerplane graph Γ'_2, see Figure 6d. By applying the corresponding modifications to G_2, we obtain an equivalent PUDC instance G'_2 that is outerplanar and connected.

Locks and keys. Since the connections realized by means of the Type M wires are (necessarily) nonrigid, there is still some wiggle room when relaxing the fixed disks of the PUDC instance and interpreting G'_2 as an instance of UDC. To establish more control over the possible placement of the maximal rigid subgraphs, we continue to modify our graph. The main idea is to twist and interlace the tunnels in specific patterns such that the two sides of a tunnel are kept close together. More specifically, we assign a unique id to each individual tile of our construction and then deform the tunnels of each tile depending on the bit representation of its id. We refer to this as a *lock-and-key mechanic*. We start by describing this mechanic in the PUDC setting for the graph G_2 and then transition to the UDC setting and the graph G'_2.

Figures 7 and 8 illustrate the lock-and-key mechanic applied to a tile that contains a horizontal wire. The left and right sides of the wire are deformed symmetrically. On the left



(a) (b) (c)

Figure 7: (a) Schematic of the lock-and-key mechanic applied to a horizontal wire. (b) Close-up of the key and lock on the left side, which also contains the bar. The bit representation of the encoded id is 110. A low-level representation with disks can be found in Figure 8. (c) Close-up of the key and lock on the right side.

side, the red disks form a *key* and the blue disks its *lock*, and on the right side the blue disks form a key (centrally symmetric to the red key) and the red disks form a lock (centrally symmetric to the blue key). A key and its lock consist of two main parts. The part closer to the original tunnel consist of a sequence of T-shaped tunnel segments, one for each bit of the representation of the id of the tile (we assume that the bit representation of each key has the same length, which is logarithmic in the number of tiles). There are two variations of T-shaped tunnel segment, which differ in terms of their length. A long segment is used to encode a 1 and a short segment is used to encode a 0. The other part of the key and its lock is just a rectangle. We also deform the bar of the wire gadget such that it fits in between the key and its lock, see Figures 7 and 8. Since the red and blue keys of the gadget are centrally symmetric, the bar can be embedded such that it sticks out either to the left side or to the right side of the gadget tile. Hence, the information propagation still works just like in the original construction.

The lock-and-key mechanic can be applied analogously to tiles of vertical wires and Type C wires. In tiles of Type T wires, we use three centrally symmetric key-and-lock pairs to ensure that the gadget can be oriented towards either of the three directions as desired, see Figure 9. In the variable gadgets, we have to alter our strategy for choosing and placing the keys slightly since the long bar needs to be deformed to fit through *every* lock-and-key mechanic on both sides of the variable gadgets. In each of the tiles of a variable

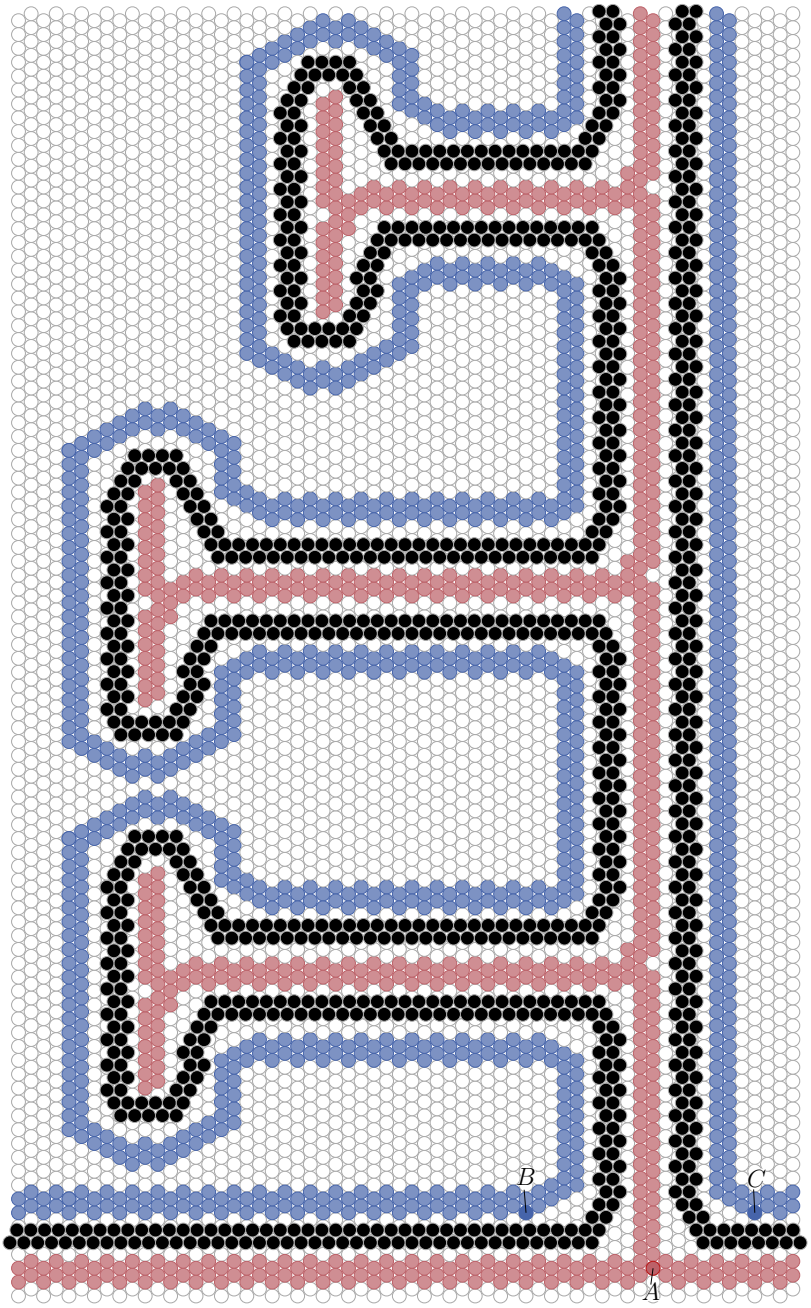


Figure 8: Realization of the bottom part of the lock-and-key mechanic illustrated in Figure 7b.

gadget we introduce two lock and key pairs near the center of the tile, just like we did in the tiles of horizontal wires. In the Type L tile, we choose two centrally symmetric lock and key pairs that encode some bit vector a, as usual. For each of the remaining tiles, we use nonsymmetric lock and key pairs such that the sequence of the bit vectors of keys encountered when walking through the variable gadget from the Type L tile T_0 to the right

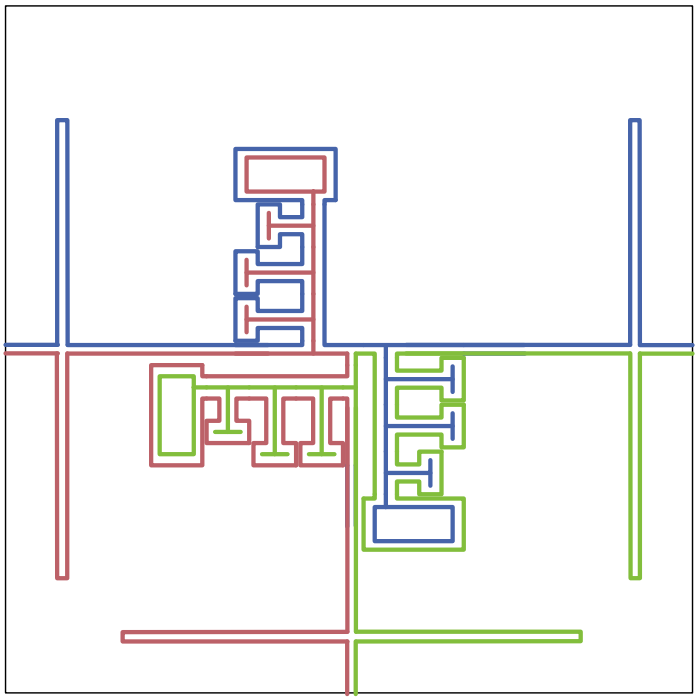


Figure 9: Schematic of the lock-and-key mechanic applied to a Type T wire. The bars have been omitted for the sake of visual clarity.

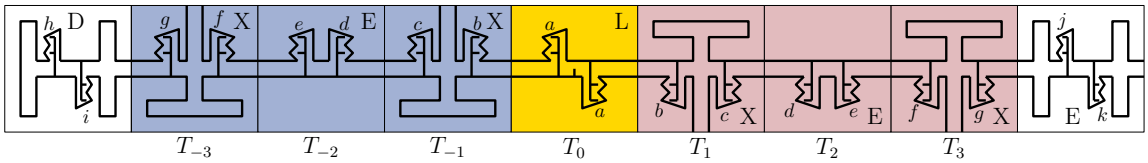


Figure 10: Placement and choice of the keys in the variable gadgets.

is the same as the one encountered when walking from T_0 to the left, for an illustration see Figure 10. More specifically, in the tile T_1 directly to the right of T_0 the left key encodes some bit vector b and the right key encodes some different bit vector c. Symmetrically, in the tile T_{-1} directly to the left of T_0 the right key encodes b and the left key encodes c. More generally, the tile T_i that is located i tiles to the right of T_0 uses two distinct lock and key pairs, and the tile T_{-i} located i tiles to the left of T_0 uses the same pairs of locks and keys but in reverse order. This choice of keys and locks ensures that the long bar can be deformed such that it fits in the tunnels on either of the two sides. In each of the Type D tiles and the Type E tiles that were placed between consecutive variable gadgets, we may use two distinct unique key pairs since our construction does not require any of the bars to traverse the center of these tiles, see Figure 10.

To ensure that there is enough space on the gadget tiles to accommodate the locks and keys, we may need to scale the underlying hexagonal grid by a polynomial factor (actually, the factor is logarithmic in the number of tiles). As already discussed in the caption

of Figure 2, when *scaling* to the desired grid size, the length of the vertical and horizontal "segment" of each T-shaped bar is increased accordingly, but its width remains fixed (at ≈ 2 times the disk diameter). Similarly, the length of the segments of the tunnel structure is increased, but the tunnels remain thin (≈ 4 times the disk diameter), so that the bar has very limited wiggle room.

Let G_3 and G_3'' denote the connected planar graphs obtained by applying the lock and key mechanic to G_2 and G_2', respectively. The graph G_3'' is not outerplanar, but we can turn it into an outerplanar graph G_3' by using the same trick that was applied when going from G_2 to G_2': for each key, we remove two adjacent disks in the part of the key that encloses its rectangular region.

From PUDC to UDC. Clearly, if G_3' has a PUDC realization, then it also has a UDC realization. For the other direction, consider a UDC realization Λ_3^{UDC} of G_3 and let Γ_3^{UDC} denote the restriction of Λ_3^{UDC} to the union of the vertices whose positions were previously considered fixed and the vertices that correspond to the chain disks (i.e., we consider all disks except for those of the bars). Let Γ_3^{PUDC} denote a PUDC realization of the same set of vertices, which exists by construction.

Let F_o denote the face of Γ_3^{PUDC} that corresponds to the outer face of the planar rectangular drawing $\mathcal{G}_\varphi^R$. Let K_{F_o} denote the maximal rigid subgraph that describes the boundary of F_o. Without loss of generality, the coordinates of the disks of K_{F_o} in Γ_3^{UDC} and Γ_3^{PUDC} are identical. Due to the connections that were introduced by means of the Type M wires, all disks of Γ_3^{UDC} are placed in the closed interior of K_{F_o}, as illustrated in Figure 6c. Let T^{UDC} and T^{PUDC} denote the face corresponding to the interior of the tunnels in Γ_3^{UDC} and Γ_3^{PUDC}, respectively, see Figure 6c (blue). It is a priori not clear that the boundaries of T^{UDC} and T^{PUDC} are identical (i.e., the combinatorial embedding of the plane graphs induced by Γ_3^{UDC} and Γ_3^{PUDC} could differ). However, by rigidity, the area A_T of T^{UDC} and T^{PUDC} is identical. Intuitively, this area corresponds to the wiggle room in our construction.

Without loss of generality, we may assume that the rectangular part of each key has an area larger than A_T. Otherwise, we can scale the instance to achieve this property since scaling increases the area of the rectangles quadratically, while the area of T^{UDC} and T^{PUDC} increases only linearly (since the width of the tunnels remains unchanged). In particular, if the area of the rectangular region R of a key is $a \cdot b$, then scaling by a factor $S \geq A_T$ increases the area of R to $\approx a \cdot b \cdot S^2$ and the area of T^{UDC} and T^{PUDC} to $\approx S \cdot A_T \leq S^2$. It follows that in Γ_3^{UDC} each key is placed in *some* lock since otherwise there is a lock whose rectangular region is devoid of any disks by construction and, hence, the area of T^{UDC} is larger than A_T, which would contradict the definition of A_T. Moreover, we will show that in Γ_3^{UDC} each key is placed in its *intended* lock, that is, the same lock it is placed in in Γ_3^{PUDC}.

Each key (and lock) encodes a bit vector, and for each bit vector there are at most three keys (and locks) encoding it. By construction, each key only fits in a lock encoding the same bit vector. Therefore, each key fits in at most three locks. We need to show that it is placed in the intended lock. For lock and key pairs introduced in the horizontal wires this is easy to see since the red key of such a wire cannot possibly be placed in the red

lock, as they belong to the same maximal rigid subgraph. For the vertical wires and the Type C wires we can argue analogously. The intended pairing of the three locks and the three keys in each Type T wire follows from the fact that its adjacent horizontal and vertical wires have the intended lock and key pairings and are rigidly connected to the T wire. For the tiles involved in the variable gadgets, we can argue similarly: each variable gadget is adjacent to a Type D or Type E wire on both its left and right side. The keys and locks used in these wires encode unique bit vectors and, hence, must be paired up as intended. It follows that the keys and locks of the leftmost and rightmost wire tile of the variable gadget are paired up as intended. This correct pairing propagates inductively through the whole variable gadget.

We have established that in Λ_3^{UDC} each key is placed in its intended lock. Consider an arbitrary key placed in its intended lock, say the key is red and its lock is blue as in Figure 8. Let A be a red disk where the key branches off of the main tunnel as marked in Figure 8. Similarly, let B and C be two blue disks where the two sides of the lock branch off of the main tunnel. Recall that we have scaled the instance by some factor S to achieve that the rectangular part of each key has an area larger than A_T. If $S = 1$ (i.e., the scaling was unnecessary), then the distance between A and B, as well as the distance between A and C are bounded by a constant. We may retain this property even for scaling factors $S > 1$. To this end, we scale the instance as usual, except that we keep the sizes of the non-rectangular parts of the keys (i.e., the parts that encode the id of the respective tile) fixed. Now consider two adjacent gadget tiles T_1 and T_2 connected by a tunnel. By applying the above argument to two consecutive key and lock pairs, one from each tile, and since the tunnel between these pairs is rigid, we obtain that the width of the entire tunnel segment is constant. Hence, if T_1 and T_2 are wire gadgets and, say, T_1 is oriented towards T_2, then T_2 cannot be oriented towards T_1. This is due to the fact that each bar ends with a (polynomially) long segment perpendicular to the tunnel and therefore tolerates a constant displacement of the tunnels it is placed in without losing its propagation property. To see that the propagation of truth values at the Type X wires in the variable gadgets also works as intended, we can argue as follows: consider two adjacent gadget tiles T_1 and T_2 where T_1 is of Type X and T_2 is of Type S. By applying the above argument about the constant distances between the disks A, B, and C of each lock and key pair to the two lock and key pairs in T_1 and the lock and key pair of T_2 that is closer to T_1, we obtain that the width of the entire tunnel segment between T_1 and T_2 is again constant. As already shown, the width of the tunnel segment leading to the two horizontal neighbors of T_1 is also constant. With the same argument as for the propagation between two wire tiles, we see that the propagation between the variable gadget and its attached literal gadgets also works as intended. Hence, the graph G_3 has a PUDC realization, as desired.

So far, we have established that G_3 has a UDC realization if and only if it has a PUDC realization. The final step is to extend our argument to our outerplanar graph G_3'. In the above packing argument, we referred to the area A_T of the tunnel face T of Γ_3^{UDC} and Γ_3^{PUDC}. In realizations of G_3', the face T was merged with other faces. However, the gaps introduced to merge these faces have a width of one disk diameter. Hence, no disks can pass through these gaps and it follows that G_3' is UDC realizable if and only if G_3 is UDC realizable.

Altogether, we have shown that the outerplanar graph G_3' is UDC realizable if and only if φ is satisfiable. The number of disks in our construction is polynomial and, hence, the underlying graph can be computed in polynomial time. This concludes the proof. □

4 Weighted disk contact graphs

In this section, we assume that a positive weight $w(v)$ is specified for each vertex v of the graph $G = (V, E)$ as part of the input. The task is to decide whether G has a DCR, in which each disk D_v representing a vertex $v \in V$ has radius proportional to $w(v)$. A DCR with this property is called a *weighted disk contact representation* (WDC representation) and a graph that has a WDC representation is called a *weighted disk contact graph* (WDC graph). Obviously, recognizing WDC graphs is at least as hard as the UDC graph recognition problem from Section 3 by setting $w(v) = 1$ for every vertex $v \in V$. Accordingly, we first show that recognizing WDC graphs is NP-hard even for stars (Section 4.1), however, embedded stars with a WDC representation can still be recognized (and one can be constructed if it exists) in linear time (Section 4.2).

4.1 Hardness for stars

We perform a polynomial reduction from the well-known 3-Partition problem. Given a bound $B \in \mathbb{N}$ and a multiset of positive integers $\mathcal{A} = \{a_1, \ldots, a_{3n}\}$ such that $\frac{B}{4} < a_i < \frac{B}{2}$ for all $i = 1, \ldots, 3n$, deciding whether $\mathcal{A}$ can be partitioned into n triples of sum B each is known to be strongly NP-complete [14]. Let $(\mathcal{A}, B)$ be a 3-Partition instance. We construct a star $S = (V, E)$, i.e., a tree with a single internal vertex $v_c \in V$ and all leaves in $V \setminus \{v_c\}$ adjacent to v_c, and a radius assignment $\mathbf{r} : V \to \mathbb{R}^+$ such that S has a WDC representation respecting $\mathbf{r}$ if and only if $(\mathcal{A}, B)$ is a yes-instance.

Recently, Alt et al. [2] considered the problem of packing disks on a horizontal line, such that the distance between the leftmost point of any disk and rightmost point of any disk is minimized. To show NP-hardness, the authors also used the reduction from the 3-Partition problem. We discussed with the authors whether our problem can be directly transformed to theirs or vice versa, but it remains unclear whether this is possible. There are two main differences to our setting: in the setting by Alt et al. (1) the disks are packed onto a flat surface rather than a curved one (i.e. the disk corresponding to the star center v_c); and (2) the disks are allowed to touch while in our setting only adjacent disks may touch, i.e, the disks corresponding to the leaves of the star must be pairwise disjoint. These differences make the necessary computations in our scenario significantly more involved.

We create a central disk D_c of radius r_c corresponding to the central vertex v_c of S as well as a fixed number of outer disks with identical radius r_o chosen appropriately such that these disks have to be placed closely together around D_c without touching, creating funnel-shaped *gaps* of roughly equal size; see Figure 11. Then, a WDC representation of S exists only if all remaining disks can be distributed among the gaps, and the choice of the gap will induce a partition of the integers $a_i \in \mathcal{A}$. We shall represent each a_i by a single disk called an *input* disk and encode a_i in its radius. Each of the gaps is supposed to be

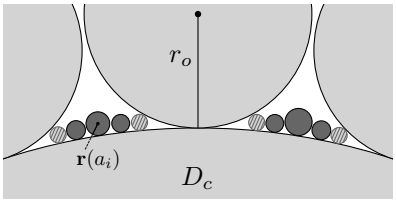


Figure 11: Reducing from 3-Partition to prove Theorem 2. Input disks (dark) are distributed between gaps. Hatched disks are separators.

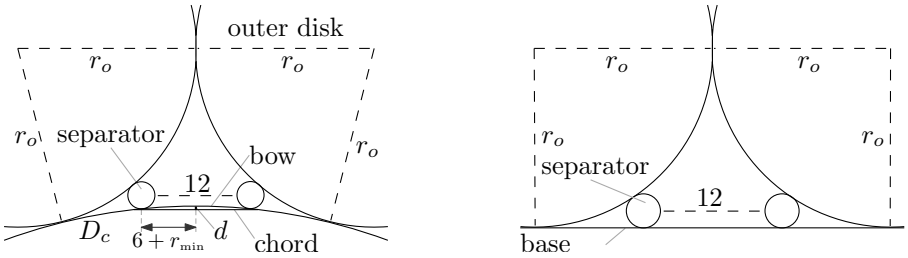
large enough for the input disks that represent a *feasible triple*, i.e., with sum B, to fit inside it, however, the gaps must be too small to contain an *infeasible triple*'s disk representation, i.e., a triple with sum $> B$.

While the principal idea of the reduction is simple, the main challenge is finding a radius assignment satisfying the above property and taking into account numerous additional, nontrivial geometric considerations that are required to make the construction work. For example, we require that the lower boundary of each gap is sufficiently flat. We achieve this by creating additional dummy gaps and ensure that they can not be used to realize a previously infeasible instance. Next, we make sure that additional *separator* disks must be placed in each gap's corners to prevent left and right gap boundaries from interfering with the input disks. Finally, all our constructions are required to tolerate a certain amount of "wiggle room", since, firstly, the outer disks do not touch and, secondly, some radii cannot be computed precisely in polynomial time and must be approximated.

Since S is supposed to be a star, the only adjacencies in our construction are the ones with D_c. However, several of the disks adjacent to D_c are required to be placed very close together without actually touching. We shall, whenever we need to calculate distances, handle these barely not touching disks as if they were actually touching. We will describe how to compute these distances approximately; see Lemma 7. During this step the radius of the central disk increases by a suitably small amount such that no unanticipated embeddings can be created.

Let $B > 12$ and $n > 6$, and let $m \geq n$ be the number of gaps in our construction. In the *original* scenario described above, a gap's boundary belonging to the central disk D_c, which we call the gap's *bow*, is curved as illustrated in Figure 12a. We will, however, first consider a *simplified* scenario in which a gap is created by placing two disks of radius r_o right next to each other on a straight line as depicted in Figure 12b. We refer to this gap's straight boundary as the *base* of the gap. We call a point's vertical distance from the base its *height*. We also utilize the terms *left* and *right* in an obvious manner. Assume for now that we can place two *separator* disks in the gap's left and right corner, touching the base and such that the distance between the rightmost point p_l of the left separator and the leftmost point p_r of the right separator is exactly 12 units. These separator disks are small and have the radius of the smallest possible input disk. We can assume $B \equiv 0 \mod 4$; see Lemma 2. Thus, we know that $a \in \{B/4 + 1, \ldots, B/2 - 1\}$ for any $a \in \mathcal{A}$.

Lemma 2. *For each $m \geq n$, there exists a 3-Partition instance $(\mathcal{A}', B')$ equivalent to the 3-Partition instance $(\mathcal{A}, B)$ described above with $|\mathcal{A}'| = 3m$ and $B' = 180B$.*



(a) original scenario (b) simplified scenario

Figure 12: A gap, bounded in (a) by two outer disks and a bow; in (b) the gap's base replaces its bow. The distance between the separators is 12 in both scenarios.

Proof. Let $n' = m-n$. For each $a_i \in \mathcal{A}$, we add $180a_i$ to $\mathcal{A}'$. Additionally, we add $2n'$ integers with value $60B-5$ and n' integers with value $60B+10$. The resulting instance $(\mathcal{A}', B')$ can be realized if and only if the original 3-Partition instance $(\mathcal{A}, B)$ is a yes-instance. Clearly, if $(\mathcal{A}, B)$ is a yes-instance, then $(\mathcal{A}', B')$ is a yes-instance. For the opposite direction, let S be a solution for $(\mathcal{A}', B')$ and assume that $(\mathcal{A}, B)$ is a no-instance. Then, there exists a triple t of integers in S that contains either one or two integers of $\mathcal{A}' \setminus \{a = 180a_i \mid a_i \in \mathcal{A}\}$. The sum of the integers in t is 5, 10, 20, 50 or 55 $\mod 60$ contradicting to S being a solution for $(\mathcal{A}', B')$ since $B' \equiv 0 \mod 60$. □

Our first goal is to find a function $\mathbf{r} : \{B/4, B/4+1, \ldots, B/2\} \to \mathbf{R}^+$ that assigns a disk radius to each input integer as well as to the values $B/4$ and $B/2$ such that a disk triple t together with two separator disks can be placed on the base of a gap without intersecting each other or the outer disks if and only if t is feasible. In the following, we show that $\mathbf{r}(x) = 2 - (4 - 12x/B)/B$ will satisfy our needs. We choose the radius of the separators to be $r_{\min} = \mathbf{r}(B/4+1) = 2 - (1 - 12/B)/B$, the smallest possible input disk radius. The largest possible input disk has radius $r_{\max} = \mathbf{r}(B/2-1) = 2 + (2 - 12/B)/B$. Note that $\mathbf{r}$ is linear and increasing.

Next, we show for both scenarios that separators placed in each gap's corners prevent the left and right gap boundaries from interfering with the input disks.

Lemma 3. *Let the radii r_c of the central disk and r_o of the outer disk vary arbitrarily, such that the distance between two separator disks placed into the corners of a gap remains 12. Then, for any $a \in \mathcal{A}$ it is not possible that a disk with radius $\mathbf{r}(a)$ intersects one of the outer disks that bound the gap when placed between the two separators.*

Proof. First, we utilize a geometric construction to show that $r_o^u = 38$ is an upper bound for the outer disks' radius r_o and then use this result to prove that even disks with radius $r_{\max}$ placed in a gap right next to a separator do not intersect an outer disk in the original scenario, implying that the input disks can actually be placed inside the gaps.

Let the distance between the two separator disks always remain 12 as in Figure 12b. For fixed $r_{\min}$, if the number of gaps m decreases, then r_c decreases and r_o increases. For fixed m, if $r_{\min}$ increases, so does r_o. We designate a minimum value of $m_{\min} = 6$ to m

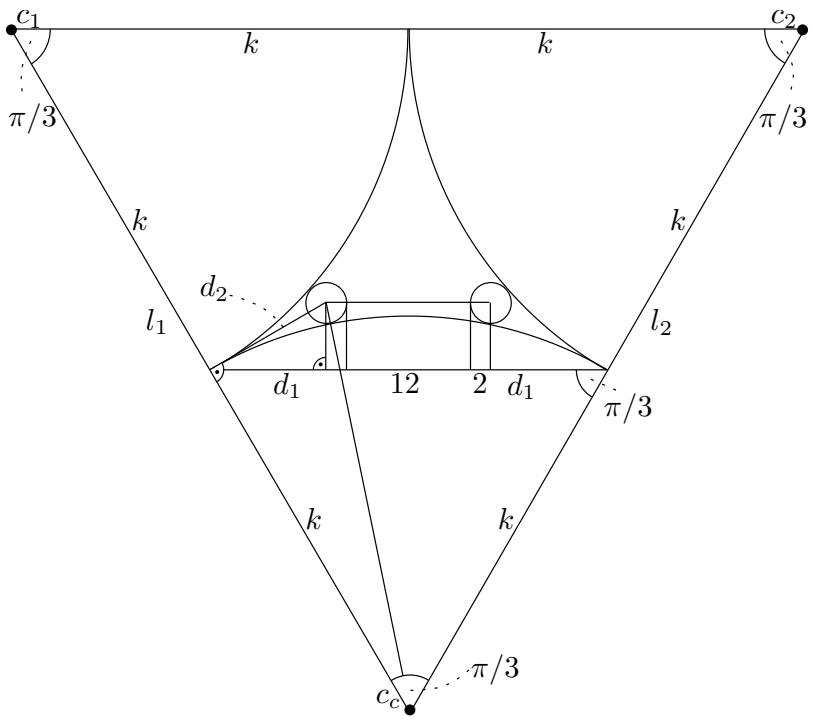


Figure 13: The geometric construction for the proof of Lemma 3.

and observe that $r_{\min} = \mathbf{r}(B/4+1) = 2-(1-12/B)/B < 2$ for any $B > 12$. Consider the extreme case $m = 6$ and $r_{\min} = 2$ in Figure 13. The angle between the two line segments l_1, l_2 bounded by the centers c_1, c_2 of two adjacent outer disks and the center c_c of the central disk D_c is $2\pi/m_{\min} = \pi/3$ and, therefore, l_1 and l_2 together with the line segment $\overline{c_1c_2}$ constitute an equilateral triangle. This implies that the outer disk radius is equal to the radius of the central disk, we denote this radius with k. By using basic trigonometry as well as the Pythagorean Theorem, we obtain that k has to satisfy the equality $k = d_1 + 2 + 12 + 2 + d_1 = 2d_1 + 16$, where $d_1 = \cos(\pi/6) \cdot d_2 = \sqrt{3} \cdot d_2/2$ and $d_2 = \sqrt{(k+2)^2 - k^2} = \sqrt{4k+4} = 2\sqrt{k+1}$. This set of equalities solves for $k = 22 + 4\sqrt{5}\sqrt{3} = 37.4919... < 38 = r_o^u$.

We now show that a disk with radius $r_{\max}$ placed in a gap right next to a separator does not intersect an outer disk. Assume the separator always touches both the outer and the central disk, and consider the angle α between the two tangents on the separator in the two touching points. For fixed r_o and r_c, if $r_{\min}$ decreases, so does α. For fixed $r_{\min}$ and r_o, if r_c increases, then α decreases. Similarly, for fixed $r_{\min}$ and r_c, if r_o increases, then α decreases. Thus, to compute the lower bound for α, we use upper bounds r_c^u and r_o^u for r_c and r_o respectively and a lower bound $r_{\min}^l$ for $r_{\min}$. We will show that a disk with radius $r_{\max}$ always fits inside α right next to the separator. A suitable choice for these values is $r_c^u = \infty$, $r_o^u = 38$ and $r_{\min}^l = 2 - \frac{1}{12}$.

To compute the value of α corresponding to these values, consider Figure 14. It holds: $\beta = \arccos((r_o^u - r_{\min}^l)/(r_o^u + r_{\min}^l))$, $\gamma = \pi/2 - \beta$ and $\alpha = \pi/2 - \gamma \approx 25.3°$. Recall

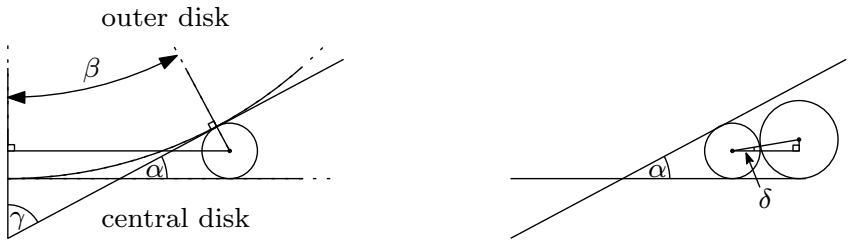


Figure 14: A disk with radius $r_{\max}$ can always be placed right next to a separator in a corner of a gap.

that $r^u_{\max} = 2 + 1/6$ is an upper bound for $r_{\max}$. Assume disks with radii $r_{\min}$ and $r_{\max}$ are placed next to each other on a horizontal line. Then, for the angle δ in Figure 14 it holds: $\delta = \arcsin((r_{\max} - r_{\min})/(r_{\max} + r_{\min})) \leq \arcsin((r^u_{\max} - r^l_{\min})/(r^u_{\max} + r^l_{\min})) \approx 3.51° < \alpha/2$. It follows that the center of the bigger disk lies below the bisector of α. Therefore, the bigger disk fits inside α. □

For our further construction, we need to prove the following property.

Property 1. Each feasible triple fits inside a gap containing two separators and no infeasible triple does.

We will show that Property 1 holds for both the simplified flat scenario and for the original curved one. For the ease of presentation, we will first consider the simplified scenario.

4.1.1 Simplified scenario

It can be easily verified that for x_1, x_2, x_3, $\sum_{i=1}^{3} x_i \leq B$, it is $2\sum_{i=1}^{3} \mathbf{r}(x_i) \leq 12$, implying the first part of Property 1. We define $\sigma = 2r_{\min} + 2\sqrt{(r_{\max} + r_{\min})^2 - (r_{\max} - r_{\min})^2}$. In the proof of Lemma 4, we will see that σ is the horizontal space required for the triple $(r_{\min}, r_{\max}, r_{\min})$, which is the narrowest infeasible triple. Next, let

$$d(\varepsilon, x) = \sqrt{(\mathbf{r}(x) - \varepsilon/2)^2 + (\mathbf{r}(x) - r_{\min})^2}$$

for $\varepsilon > 0$ and $x \in \{B/4+1, \ldots, B/2-1\}$. We will see that $d(\varepsilon, x)$ is an upper bound for the distance between the center of a disk $D(x)$ with radius $\mathbf{r}(x)$ and the rightmost (leftmost) point of the left (right) separator disk, if the overlap of their horizontal projections is at least $\varepsilon/2$.

Claim 1. *For the simplified scenario, consider three disks with radii r_1, r_2, r_3 with $r_{\min} \leq r_i \leq r_{\max}$ for $i = 1, \ldots, 3$.*

i) Let the three disks correspond to a feasible triple. Then, the required horizontal space for the three disks is 12 if $r_1 = r_2 = r_3 = 2$ and less than 12 otherwise.

ii) For $r_1 + r_2 + r_3 \geq 2r_{\min} + r_{\max}$, the required horizontal space for the three disks is minimized for $r_2 = r_{\max}$ and $r_1 = r_3 = r_{\min}$ and in this case it is σ.

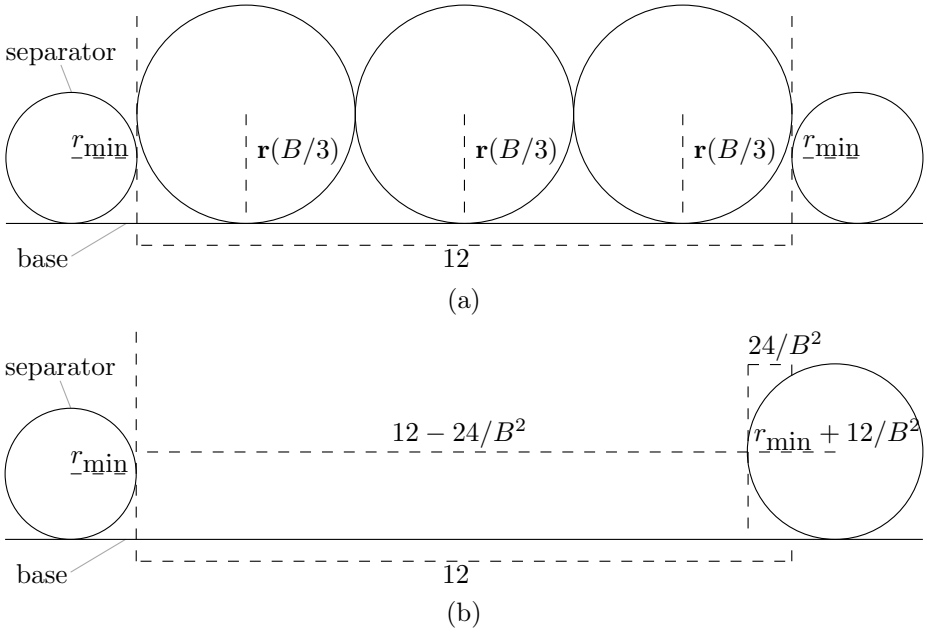


Figure 15: Two illustrations regarding the proof of Lemma 4. (a) Depiction of a feasible input triple's disk representation. This particular representation requires the largest possible amount of horizontal space out of all representations for feasible input triples with sum B. (b) An upper bound for the amount of horizontal space between the two disks placed in a gap's corner.

Proof. For $i = 1, \ldots, 3$, let d_i denote the disk with radius r_i, such that $r_{\min} \leq r_i \leq r_{\max}$ for $i = 1, \ldots, 3$. Consider a packing of the three disks on a horizontal line, such that d_1 touches d_2 and d_2 touches d_3. Using the Pythagorean Theorem, the horizontal space consumption $h(r_1, r_2, r_3)$ is then

$$\begin{aligned} h(r_1, r_2, r_3) &= r_1 + \sqrt{(r_1 + r_2)^2 - (r_1 - r_2)^2} + \sqrt{(r_2 + r_3)^2 - (r_2 - r_3)^2} + r_3 \\ &= r_1 + 2\sqrt{r_1 r_2} + 2\sqrt{r_2 r_3} + r_3 \\ &= 2(r_1 + r_2 + r_3) - (r_1 + r_2 - 2\sqrt{r_1 r_2}) - (r_2 + r_3 - 2\sqrt{r_2 r_3}) \\ &= 2(r_1 + r_2 + r_3) - (\sqrt{r_1} - \sqrt{r_2})^2 - (\sqrt{r_2} - \sqrt{r_3})^2. \end{aligned}$$

i) By the definition of $\mathbf{r}$, we have $2(r_1 + r_2 + r_3) \leq 12$. Therefore, $h(r_1, r_2, r_3) = 12$ only if $r_1 = r_2 = r_3 = 2$, and $h(r_1, r_2, r_3) < 12$ otherwise. ii) For $r_1 + r_2 + r_3 \geq 2r_{\min} + r_{\max}$, we have $h(r_1, r_2, r_3) \geq 2(2r_{\min} + r_{\max}) - 2(\sqrt{r_{\max}} - \sqrt{r_{\min}})^2 = h(r_{\min}, r_{\max}, r_{\min})$. □

Lemma 4. *There exist $\varepsilon, \varepsilon_1, \varepsilon_2, \phi > 0$ with $\varepsilon = \varepsilon_1 + \varepsilon_2$ that satisfy the two conditions:* (I) $12 + \varepsilon \leq \sigma$ *and* (II) $d(\varepsilon_1, x) \leq \mathbf{r}(x) - \phi \quad \forall x \in \{B/4 + 1, \ldots, B/2 - 1\}$.

Proof. Recall that the function $\mathbf{r}$ is linear. A triple of disks with uniform radius $\mathbf{r}(B/3) = 2$ requires a total horizontal space of $2 \cdot 2 \cdot 3 = 12$ if placed tightly next to each other on a

straight line. Therefore, since the radius $r_{\min}$ of the separators is less than 2, it follows that every feasible disk triple fits in the gap since (1) by Claim 1, a triple of disks with uniform radius $\mathbf{r}(B/3)$ yields an upper bound for the amount of horizontal space required by any feasible disk triple and since (2) $\mathbf{r}(B/3) > r_{\min}$, which implies that if the three disks are placed next to each other on the base, the height of the leftmost point of the disk triple is greater than the height of the rightmost point p_l of the left separator and, therefore, these disks do not touch (and the same holds true for the right side respectively), see Figure 15a.

Next, consider the disk triple $t_i = (r_{\min}, r_{\max}, r_{\min})$; see Figure 16a. The sum of the integers corresponding to the disks of t_i is $B/4 + 1 + B/2 - 1 + B/4 + 1 = B + 1$ and, therefore, t_i is infeasible. Since $\mathbf{r}$ is linear and $B + 1$ is the smallest possible sum of any infeasible integer triple, by Claim 1, $\sigma = 2r_{\min} + 2\sqrt{(r_{\max} + r_{\min})^2 - (r_{\max} - r_{\min})^2}$ is the least possible amount of horizontal space required by any infeasible disk triple.

In order to show that $0 < \varepsilon \leq 17/B^2$ is a sufficient choice to satisfy Condition I, for an arbitrary $0 < c \leq 17$ we assign $\varepsilon = c/B^2$ and show that the condition holds and for any $B > 12$.

$$
\begin{aligned}
12 + \varepsilon \leq \sigma \Leftrightarrow \\
12 + c/B^2 \leq 2r_{\min} + 2\sqrt{(r_{\max} + r_{\min})^2 - (r_{\max} - r_{\min})^2} \Leftrightarrow \\
3 + (c/4)/B^2 - (1/2)r_{\min} \leq \sqrt{r_{\max} r_{\min}} \Leftarrow \\
(3 + (c/4)/B^2 - (1/2)(2 - 1/B + 12/B^2))^2 \leq \\
(2 - 1/B + 12/B^2)(2 + 2/B - 12/B^2) \Leftrightarrow \\
9 + (c^2/16)/B^4 + 1 + 1/(4B^2) + 36/B^4 - 1/B + 12/B^2 - 6/B^3 + \\
3c/(2B^2) - 6 \ + 3/B - 36/B^2 - c/(2B^2) + c/(4B^3) - 3c/B^4 \leq \\
4 + 2/B - 2/B^2 + 36/B^3 - 144/B^4 \Leftrightarrow \\
(c^2/16 - 3c + 180)/B^4 + (c/4 - 42)/B^3 + (c - 87/4)/B^2 \leq 0 \Leftarrow \\
c^2/16 - 3c + 180 + (c/4 - 42)B + (c - 87/4)B^2 \leq 0 \Leftarrow \\
17/16 - 3 \cdot 0 + 180 + (17/4 - 42)B + (17 - 87/4)B^2 \leq 0 \Leftrightarrow \\
17/16 + 180 - (151/4)B - (19/4)B^2 \leq 0
\end{aligned}
$$

The last inequality clearly holds true for any $B > 12$, which concludes the proof of this step.

We now show that Condition II holds for $\varepsilon_1 = 16/B^2$ and $0 \leq \phi \leq 1/B^2$. To this end, we substitute $y = x \cdot 12/B$ and show that $d((16/B^2), (y \cdot B/12)) \leq \mathbf{r}(y \cdot B/12) - c/B^2$ for any $y \in \{3 + 1 \cdot 12/B, 3 + 2 \cdot 12/B, \ldots, 6 - 12/B\}$, any $0 \leq c \leq 1$ and any $B > 12$.

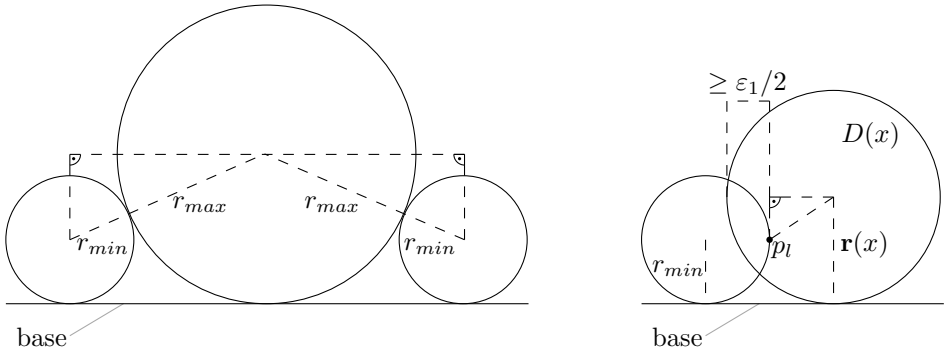


Figure 16: By Lemma 4, it is not possible to place an infeasible disk triple inside a simplified gap. (a) The smallest possible infeasible disk triple. (b) Disk $D(x)$ is intersecting the separator.

$$
\begin{aligned}
d((16/B^2),(y\cdot B/12)) \leq \mathbf{r}(y\cdot B/12) - c/B^2 \Leftrightarrow \\
\sqrt{(\mathbf{r}(y\cdot B/12) - 8/B^2)^2 + (\mathbf{r}(y\cdot B/12) - r_{\min})^2} \leq \mathbf{r}(y\cdot B/12) - c/B^2 \Leftarrow \\
2\cdot \mathbf{r}(y\cdot B/12)^2 - 16\cdot \mathbf{r}(y\cdot B/12)/B^2 + 64/B^4 - 2\mathbf{r}(y\cdot B/12) r_{\min} + (r_{\min})^2 \leq \\
\mathbf{r}(y\cdot B/12)^2 - 2c\cdot \mathbf{r}(y\cdot B/12)/B^2 + c^2/B^4 \Leftrightarrow \\
\mathbf{r}(y\cdot B/12)^2 - 16\cdot \mathbf{r}(y\cdot B/12)/B^2 + 64/B^4 - 2\mathbf{r}(y\cdot B/12) r_{\min} + (r_{\min})^2 \leq \\
-2c\cdot \mathbf{r}(y\cdot B/12)/B^2 + c^2/B^4 \Leftrightarrow \\
(4 + 16/B^2 + y^2/B^2 - 16/B + 4y/B - 8y/B^2) + (-32/B^2 + 64/B^3 - 16y/B^3) + \\
64/B^4 + (-8 + 16/B - 4y/B + 4/B - 8/B^2 + 2y/B^2 - 48/B^2 + \\
96/B^3 - 24y/B^3) + (4 + 1/B^2 + 144/B^4 - 4/B + 48/B^2 - 24/B^3) + \\
(4c/B^2 - 8c/B^3 + 2cy/B^3) - c^2/B^4 \leq 0 \Leftrightarrow \\
(208 - c^2)/B^4 + (64 - 16y + 96 - 24y - 24 - 8c + 2cy)/B^3 + \\
(16 + y^2 - 8y - 32 - 8 + 2y - 48 + 1 + 48 + 4c)/B^2 \leq 0 \Leftrightarrow \\
(208 - c^2)/B^4 + (136 - 40y + 2cy - 8c)/B^3 + (-23 + y^2 - 6y + 4c)/B^2 \leq 0 \Leftarrow \\
208 - c^2 + (136 - 40y + 2cy - 8c)B + (-23 + y^2 - 6y + 4c)B^2 \leq 0 \Leftarrow \\
208 - 0^2 + (136 - 40\cdot 3 + 2\cdot 1\cdot 6 - 8\cdot 0)B + (-23 + 0 + 4\cdot 1)B^2 \leq 0 \Leftrightarrow \\
208 + 28B - 19B^2 \leq 0
\end{aligned}
$$

The last inequality clearly holds true for any $B > 12$, which concludes this part of the proof. □

We now use Conditions I and II from Lemma 4 to prove the second part of Property 1 for the simplified scenario.

Corollary 1. *No infeasible disk triple can be placed in the gap together with two separators.*

Proof. Recall that the distance between the rightmost point p_l of the left separator and the leftmost point p_r of the right separator, which are located at height $r_{\min}$, is exactly 12 units. Condition I from Lemma 4 ensures that all infeasible disk triples take up at least $12+\varepsilon$ units of horizontal space, however, this condition is not sufficient to guarantee that infeasible disk triples can not be placed between the separators since we do not know yet at what height the leftmost and the rightmost point of the disk triple are located. However, it is guaranteed that either the leftmost point of the disk triple is located at least $\varepsilon_1/2$ units to the left of p_l or the rightmost point of the triple is located at least $\varepsilon_1/2$ units to the right of p_r. Let now $x \in \{B/4+1, \ldots, B/2-1\}$ be an input integer. The Pythagorean Theorem implies that the distance between p_l (p_r) and the center of a disk $D(x)$ with radius $\mathbf{r}(x)$ whose center is located between the two separator disks and whose leftmost (rightmost) point is located at least $\varepsilon_1/2$ units to the left (right) of p_l (p_r) is at most $d(\varepsilon_1, x) = \sqrt{(\mathbf{r}(x)-\varepsilon_1/2)^2 + (\mathbf{r}(x)-r_{\min})^2}$, as illustrated in Figure 16b. Condition II ensures that this distance is at most $\mathbf{r}(x)-\phi$, implying that $D(x)$ intersects the left (right) separator. Therefore, Condition I and Condition II together guarantee that infeasible disk triples together with two separators can not be placed inside a gap in the simplified scenario. The significance of ε_2 becomes clear in the proof of Lemma 6, where we tailor our conditions to apply to the original scenario as well. □

So far we assumed that the separators are always placed in the corners of the gap. But in fact, separators could be placed in a different location, moreover, there could even be gaps with multiple separators and gaps with zero or one separator. Since the radius of the separators is $r_{\min}$, which is the radius of the smallest possible input disk, it seems natural to place them in the gaps' corners to efficiently utilize the horizontal space. However, all feasible disk triples (except $(B/3, B/3, B/3)$) require less than 12 units of horizontal space. It might therefore be possible to place a feasible disk triple inside a gap together with two disks that are not necessarily separators but input disks with a radius greater than $r_{\min}$. To account for this problem, we prove the following property.

Property 2. A feasible disk triple can be placed in the gap together with two other disks only if those two disks have radius $r_{\min}$.

In this section, we prove Property 2 for the simplified scenario. The proof for the original scenario will be presented in Section 4.1.2. The following technical lemma is instrumental for the proof.

We define $s_f = 2\mathbf{r}(B/4) + 2\sqrt{(\mathbf{r}(B/2)+\mathbf{r}(B/4))^2 - (\mathbf{r}(B/2)-\mathbf{r}(B/4))^2}$. In the proof of Lemma 5, we will see that s_f is a lower bound for the horizontal space consumption of any feasible triple.

Lemma 5. *There exist $\xi, \xi_1, \xi_2, \psi > 0$ with $\xi = \xi_1+\xi_2$ satisfying the following two conditions:* (III) $12 - 24/B^2 + \xi \le s_f$ *and* (IV) $d(\xi_1, x) \le \mathbf{r}(x) - \psi \ \ \forall x \in \{B/4+1, \ldots, B/2-1\}$.

Proof. To show that $0 < \xi \le 17/B^2$ is a sufficient choice to satisfy Condition III, for an

arbitrary $0 < c \leq 17$ we assign $\xi = c/B^2$ and show that the condition holds for any $B > 12$.

$$\begin{aligned}
12 - 24/B^2 + \xi \leq s_f \Leftrightarrow \\
12 - 24/B^2 + c/B^2 \leq 2\mathbf{r}(B/4) + 2\sqrt{(\mathbf{r}(B/2) + \mathbf{r}(B/4))^2 - (\mathbf{r}(B/2) - \mathbf{r}(B/4))^2} \Leftrightarrow \\
12 + (c - 24)/B^2 - 2\mathbf{r}(B/4) \leq 4\sqrt{\mathbf{r}(B/2) \cdot \mathbf{r}(B/4)} \Leftrightarrow \\
2 + (c/4 - 6)/B^2 + (1/2)/B \leq \sqrt{(2 + 2/B) \cdot (2 - 1/B)} \Leftarrow \\
4 + (c^2/16 - 3c + 36)/B^4 + (1/4)/B^2 + (c - 24)/B^2 + 2/B + (c/4 - 6)/B^3 \leq \\
4 - 2/B + 4/B - 2/B^2 \Leftrightarrow \\
(c^2/16 - 3c + 36)/B^4 + (c/4 - 6)/B^3 + (c - 24 + 1/4 + 2)/B^2 \leq 0 \Leftarrow \\
c^2/16 - 3c + 36 + (c/4 - 6)B + (c - 22 + 1/4)B^2 \leq 0 \Leftarrow \\
17^2/16 - 3 \cdot 0 + 36 + (17/4 - 6)B + (17 - 22 + 1/4)B^2 \Leftrightarrow \\
289/16 + 36 - (7/4)B - (19/4)B^2 \leq 0
\end{aligned}$$

The last inequality can easily be verified to be true for any $B > 12$.

The arguments for showing that Condition IV holds for $\xi_1 = 16/B^2$ and an arbitrary ψ with $0 \leq \psi \leq 1/B^2$ is identical to that of Condition II. □

Corollary 2. *Property 2 holds for the simplified scenario.*

Proof. Recall that the second smallest possible input disk radius is $\mathbf{r}(B/4 + 2) = 2 - (1 - 24/B)/B = 2 - (1 - 12/B)/B + 12/B^2 = r_{\min} + 12/B^2$ and, therefore, $12 - 2 \cdot 12/B^2 = 12 - 24/B^2$ is an upper bound for the remaining horizontal space in a gap in which two disks have been placed such that one of the disks has radius greater than $r_{\min}$, see Figure 15b. The input integers' values are at least $B/4 + 1$ and at most $B/2 - 1$, therefore, the horizontal space consumption of the disk triple $t_f = (\mathbf{r}(B/4), \mathbf{r}(B/2), \mathbf{r}(B/4))$ is a lower bound for the space consumption of any feasible disk triple since the total difference between the radii of adjacent disks in t_f is larger than that of any feasible disk triple. Yet again we utilize the Pythagorean Theorem to describe t_f's required horizontal space as $s_f = 2\mathbf{r}(B/4) + 2\sqrt{(\mathbf{r}(B/2) + \mathbf{r}(B/4))^2 - (\mathbf{r}(B/2) - \mathbf{r}(B/4))^2}$. Condition III therefore ensures that any feasible disk triple consumes at least $12 - 24/B^2 + \xi$ horizontal space and, analogously to Condition II, Condition IV together with

$$d(\xi_1, x) = \sqrt{(\mathbf{r}(x) - \xi_1/2)^2 + (\mathbf{r}(x) - r_{\min})^2}$$

ensures that one of the disks of t_f intersects a separator or one of the replacing disks, implying Property 2. Like with ε_2 the significance of ξ_2 will become apparent in the proof of Lemma 6 when we describe how to apply our conditions to the original scenario. □

We verify in the proofs of Lemmas 4 and 5 that choosing $\varepsilon_1, \xi_1 = 16/B^2$ and $\varepsilon_2, \phi, \xi_2, \psi = 1/B^2$ satisfies our four conditions.

Intuitively, Conditions (I)–(IV) have the following meaning. By (I), the horizontal space consumption of any infeasible triple is greater than 12 by some fixed buffer. By (III),

the horizontal space consumption of any feasible triple is very close to 12. Conditions (II) and (IV) imply that if the overlap of the horizontal projections of a separator and an input disk is large enough, the two disks intersect, implying that triples with sufficiently large space consumption can indeed not be placed between two separators.

4.1.2 Original scenario

We now return to the original scenario. From now on, whenever we consider a fixed pair of separator disks touching the central disk D_c, we assume that their centers lie on a horizontal line and that the center of D_c lies below that line. When we talk about the *horizontal space consumption* of disks touching D_c between the two separators, we mean the distance between the leftmost and rightmost points of their horizontal projection. The amount of *free horizontal space* is the maximum possible horizontal space consumption of disks lying on the boundary of D_c between the two separators. In the following, we will show that the horizontal space consumption of input disk triples changes only insignificantly between the simplified and the original scenario if the gap's bow is flat enough; see Lemma 6. Proving this will let us use Lemmas 4 and 5 and allow us to prove Properties 1 and 2 for the original scenario.

In the original scenario, consider a straight line directly below the two separators. We call this straight line the gap's *chord*, see Figure 12a. The gap's chord has a function similar to the base in the simplified scenario. We still want separators to be placed in the gap's corners. The distance between the rightmost point p_l of the left separator and the leftmost point p_r of the right separator is now allowed to be slightly more than 12. The horizontal space consumption of a disk triple placed on the bow is lower compared to the disk triple being placed on the chord. Moreover, the overlap of the horizontal projections of a separator and an input disk can now be bigger without causing an intersection. However, we show that if the maximum distance d between a gap's bow and its chord is small enough, the original scenario is sufficiently close to the simplified one, and the four conditions still hold, implying the desired properties.

Lemma 6. *In the original scenario, let $d \leq 1/4B^2$, and let the amount of free horizontal space in each gap after inserting the two separators in each corner be between 12 and $12 + 1/4B^2$. Then, Properties 1 and 2 still hold.*

Proof. Obviously, each feasible triple can still fit in the gap together with two separators. We now compute the amount of horizontal space that can be saved in the original scenario compared to the simplified scenario when placing any infeasible or feasible disk triple on the bow instead of on the chord.

Consider the disk triple $t = (\mathbf{r}(B/4), \mathbf{r}(B/2), \mathbf{r}(B/4))$. Yet again utilizing the Pythagorean Theorem, we calculate an upper bound for the amount of horizontal space that can be saved to be

$$s = 2(\sqrt{(\mathbf{r}(B/2) + \mathbf{r}(B/4))^2 - (\mathbf{r}(B/2) - \mathbf{r}(B/4))^2} \\ - \sqrt{(\mathbf{r}(B/2) + \mathbf{r}(B/4))^2 - (\mathbf{r}(B/2) + d - \mathbf{r}(B/4))^2})$$

by simply moving the left and right disk down by d units and as far to the center disk as possible, see Figure 17. Recall that $B/4$ is smaller and $B/2$ is greater than any input integer, and the differences between the radii of adjacent disks in t are larger than in any actual input disk triple. Therefore, the value s is also an upper bound for the amount of horizontal space that can be saved in the original scenario compared to the simplified scenario when placing any infeasible or feasible disk triple.

We now show that for $d \leq 1/4B^2$ an upper bound for s is $1/4B^2$.

$$
\begin{aligned}
s \leq 1/4B^2 \Leftrightarrow \\
2(\sqrt{(\mathbf{r}(B/2)+\mathbf{r}(B/4))^2-(\mathbf{r}(B/2)-\mathbf{r}(B/4))^2} \\
-\sqrt{(\mathbf{r}(B/2)+\mathbf{r}(B/4))^2-(\mathbf{r}(B/2)+d-\mathbf{r}(B/4))^2}) \leq 1/4B^2 \Leftrightarrow \\
2(\sqrt{4\cdot\mathbf{r}(B/2)\cdot\mathbf{r}(B/4)} \\
-\sqrt{4\cdot\mathbf{r}(B/2)\cdot\mathbf{r}(B/4)-2d\cdot\mathbf{r}(B/2)+2d\cdot\mathbf{r}(B/4)-d^2}) \leq 1/4B^2 \Leftrightarrow \\
\sqrt{4\cdot\mathbf{r}(B/2)\cdot\mathbf{r}(B/4)}-1/8B^2 \\
\leq \sqrt{4\cdot\mathbf{r}(B/2)\cdot\mathbf{r}(B/4)-2d\cdot\mathbf{r}(B/2)+2d\cdot\mathbf{r}(B/4)-d^2} \Leftarrow \\
1/64B^4-\sqrt{4\cdot\mathbf{r}(B/2)\cdot\mathbf{r}(B/4)}/4B^2 \leq 2d\cdot\mathbf{r}(B/4)-2d\cdot\mathbf{r}(B/2)-d^2 \Leftrightarrow \\
1/16B^2+8dB^2\cdot(2+2/B)-8dB^2\cdot(2-1/B)+4d^2B^2 \\
\leq \sqrt{4\cdot(2+2/B)\cdot(2-1/B)} \Leftarrow \\
(1/32B^2+4dB^2\cdot(2+2/B-2+1/B)+2d^2B^2)^2 \leq (2+2/B)\cdot(2-1/B) \Leftarrow \\
(5/32B^2+1\cdot(3/B))^2 \leq 4-2/B+4/B-2/B^2 \Leftrightarrow \\
25/1024B^4+30/32B^3+9/B^2 \leq 4-2/B+4/B-2/B^2 \Leftrightarrow \\
25/1024B^4+15/16B^3+11/B^2-2/B-4 \leq 0 \Leftarrow \\
25/1024+15B/16+11B^2-2B^3-4B^4 \leq 0
\end{aligned}
$$

The last inequality clearly holds true for any $B > 12$.

Consider an infeasible triple placed in a gap with two separators in the corners. The triple requires horizontal space at least $\sigma - s \geq 12+\varepsilon_1+\varepsilon_2-s \geq 12+(16+\frac{3}{4})/B^2$, and at most $12+1/4B^2$ is available. Thus, without loss of generality, the left separator and the leftmost disk $D(x)$ of the triple have overlap of horizontal projections of at least $(16+1/2B^2)/2 = 8+1/4B^2$. If $D(x)$ would be placed on the chord instead of the bow, the distance between p_l and the center of $D(x)$ would be at most $d(\varepsilon_1, x) \leq \mathbf{r}(x)-\phi = \mathbf{r}(x)-1/B^2$. When $D(x)$ is moved up and placed on the bow instead, this distance remains at most $\mathbf{r}(x)-1/B^2+d \leq \mathbf{r}(x)-3/4B^2$. Thus, the infeasible triple doesn't fit inside the gap, and Property 1 holds.

Now consider a feasible triple and assume that a separator has been replaced by a bigger disk in the gap's corner. The triple requires horizontal space at least $s_f - s \geq 12-24/B^2+\xi_1+\xi_2-1/4B^2 = 12-7/B^2-1/4B^2$. Replacing a separator by a bigger disk in the gap's corner consumes at least $24/B^2$ horizontal space in the simplified scenario

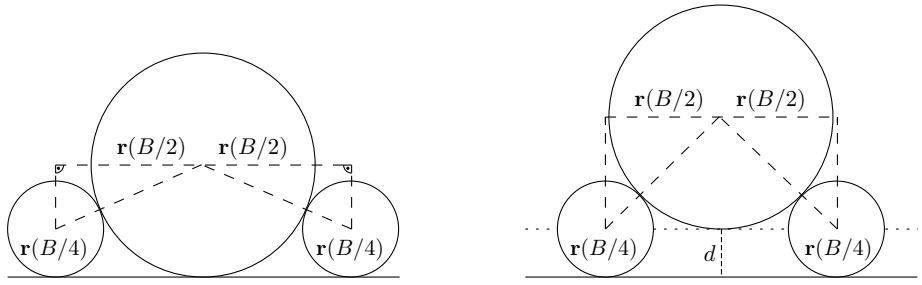


Figure 17: An upper bound for the amount of horizontal space that can be saved by placing a disk triple on the gap's bow instead of its chord can be calculated by replacing the bow by a straight line d units above the chord and comparing the required space to the space required when placing the two outer disks directly on the chord.

(see Figure 15b) and even more in the original scenario. Then, without loss of generality, the overlap of the horizontal projections of $D(x)$ and the disk in the left gap corner is at least $((12 - 7/B^2 - 1/4B^2) - (12 - 24/B^2 + 1/4B^2))/2 = 8/B^2 + 1/4B^2$, and, analogously to the above argument, the two disks intersect. Therefore, the triple can not fit in the gap, and Property 2 follows. □

In order to conclude the hardness proof, it therefore remains to describe how to choose the radii for the central and outer disks and how to create the gaps such that $d \leq 1/4B^2$. Recall that d is the distance between a gap's bow and its chord; see Figure 12b.

Recall that we have a central disk D_c with radius r_c and m outer disks with radius r_o which are tightly packed around D_c such that m equal-sized gaps are created. With basic trigonometry we see that $r_c + r_o = r_o/\sin(\pi/m)$ and, therefore, $r_c = r_o/\sin(\pi/m) - r_o$. Clearly, there always exists a value r_o such that the two separator disks can be placed in each gap's corners and such that the distance between each pair of separators is exactly 12 units. Let $\bar{r_o}$ be this value. Moreover, the maximum distance d between a gap's bow and its chord is of particular importance, see Figure 12a. Using the Pythagorean Theorem, it can be calculated to be $d = r_c - (\sqrt{(r_c + r_{\min})^2 - (6 + r_{\min})^2} - r_{\min})$. The crucial observation is that we do not necessarily need to choose $m = n$. Instead we may choose any $m \geq n$ and thereby decrease d, as long as we make sure that m is still a polynomial in the size of the input or numeric values and that the $m - n$ additional gaps cannot be used to solve an instance which should be infeasible.

Lemma 7. *There exist constants c_1, c_3, c_4, such that for $m = B^{c_1}$, $\varepsilon_3 = 1/B^{c_3}$ and $\varepsilon_4 = 1/B^{c_4}$, there exist values $\widetilde{r_o}$ for r_o and $\widetilde{r_c}$ for r_c, for which it holds $\bar{r_o} < \widetilde{r_o} \leq \bar{r_o} + \varepsilon_3$ and $\bar{r_c} < \widetilde{r_c} \leq \bar{r_c} + \varepsilon_4$ for $\bar{r_c} = \widetilde{r_o}/\sin(\pi/m) - \widetilde{r_o}$. Moreover, the constants can be chosen such that $d \leq 1/4B^2$ and such that the amount of free horizontal space in each gap is between 12 and $12 + 1/4B^2$. Finally, $\widetilde{r_o}$ and $\widetilde{r_c}$ can be computed in polynomial time.*

Proof. **Choosing** m. In order to choose an $m \geq n$ such that $d \leq 1/4B^2$, we require some information about the radius r_o of the outer disks. A precise calculation of this value yields

a complicated formula, however, a lower as well as an upper bound for r_o are sufficient to conclude our argument. Clearly, $r_o^l = 6$ is a lower bound for r_o. In the proof of Lemma 3, we have shown that for $m \geq m_{min} = 6$, $r_o^u = 38$ is an upper bound for r_o. Recalling that $r_{\min}$ is a polynomial in B, that $m \geq m_{min} = 6$ and utilizing that $\sin(x) \leq x, \forall x \geq 0$, we can now prove that m can be chosen as a polynomial in B such that $d \leq 1/4B^2$:

$$
\begin{aligned}
d \leq 1/4B^2 &\Leftrightarrow \\
r_c - (\sqrt{(r_c + r_{\min})^2 - (6 + r_{\min})^2} - r_{\min}) \leq 1/4B^2 &\Leftrightarrow \\
(r_c + r_{\min}) - 1/4B^2 \leq \sqrt{(r_c + r_{\min})^2 - (6 + r_{\min})^2} &\Leftrightarrow \\
1/16B^4 - (r_c + r_{\min})/2B^2 \leq -(6 + r_{\min})^2 &\Leftrightarrow \\
1/8B^2 + 2B^2(6 + r_{\min})^2 - r_{\min} \leq r_c &\Leftrightarrow \\
1/8B^2 + 2B^2(6 + r_{\min})^2 - r_{\min} + r_o \leq r_o/\sin(\pi/m) &\Leftrightarrow \\
\sin(\pi/m) \leq r_o/(1/8B^2 + 2B^2(6 + r_{\min})^2 - r_{\min} + r_o) &\Leftarrow \\
\pi/m \leq r_o/(1/8B^2 + 2B^2(6 + r_{\min})^2 - r_{\min} + r_o) &\Leftrightarrow \\
m \geq (\pi/r_o) \cdot (1/8B^2 + 2B^2(6 + r_{\min})^2 - r_{\min} + r_o) &\Leftarrow \\
m \geq (\pi/r_o^l) \cdot (1/8B^2 + 2B^2(6 + 1 + r_{\min})^2 - r_{\min} + r_o^u + 1) & \\
\geq (\pi/r_o^l) \cdot (1/8B^2 + 2B^2(6 + r_{\min})^2 - r_{\min} + r_o^u) &
\end{aligned}
$$

Therefore, we define $m = B^{c_1}$ where c_1 is a sufficiently large constant. Note that we need to ensure that $m \geq n$, which is however no problem since we can, without loss of generality, assume that B is a multiple of n since we could simply multiply each input integer as well as the bound by n to obtain a problem instance that is a yes-instance if and only if the original instance was a yes instance and whose size is polynomial in the size of the original input.

For the approximate radii, the upper and lower bounds still hold. Similar to the proof of Lemma 3, the upper bound of $12 + 1/12$ for the separator distance provides the equation $k = 2\sqrt{3} \cdot \sqrt{k+1} + (16 + \frac{1}{12})$, which has a solution for $k = 37.6 < 38 = r_o^u$. Since it is $\bar{r_o} \leq \widetilde{r_o}$ and $\bar{r_c} \leq \widetilde{r_c}$, the lower bound $r_o^l = 6$ still holds.

Note that if the promised approximate values $\widetilde{r_o}, \widetilde{r_c}$ for r_o and r_c are used and the distance between the separators is between 12 and $12 + 1/4B^2$, the maximum distance between the bow and the chord changes compared to the precise scenario. It holds now: $d \leq \widetilde{r_c} - (\sqrt{(\widetilde{r_c} + r_{\min})^2 - (6 + 1/8B^2 + r_{\min})^2} - r_{\min})$. However, the upper bound of $1/4B^2$ still holds true because of the following.

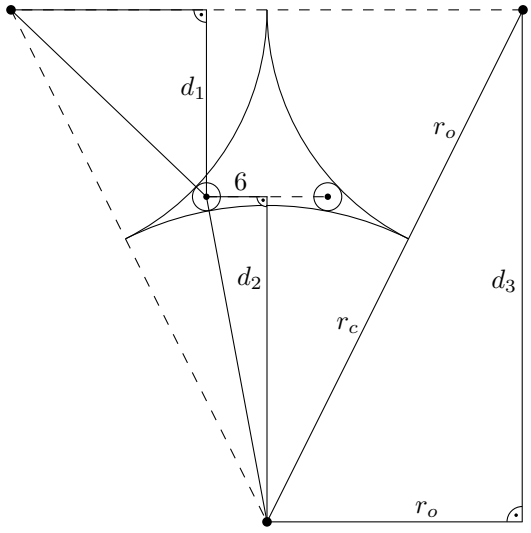


Figure 18: Determining the outer disk radius by utilizing the Pythagorean Theorem.

$$\begin{aligned}
\widetilde{r_c} - (\sqrt{(\widetilde{r_c} + r_{\min})^2 - (6 + 1/8B^2 + r_{\min})^2} - r_{\min}) \leq 1/4B^2 \Leftrightarrow \\
(\widetilde{r_c} + r_{\min}) - 1/4B^2 \leq \sqrt{(\widetilde{r_c} + r_{\min})^2 - (6 + 1/8B^2 + r_{\min})^2} \Leftrightarrow \\
1/16B^4 - (\widetilde{r_c} + r_{\min})/2B^2 \leq -(6 + 1/8B^2 + r_{\min})^2 \Leftrightarrow \\
1/8B^2 + 2B^2(6 + 1/8B^2 + r_{\min})^2 - r_{\min} \leq \widetilde{r_c} \Leftarrow \\
1/8B^2 + 2B^2(6 + 1/8B^2 r_{\min})^2 - r_{\min} + \widetilde{r_o} \leq \widetilde{r_o}/\sin(\pi/m) \Leftrightarrow \\
\sin(\pi/m) \leq \widetilde{r_o}/(1/8B^2 + 2B^2(6 + 1/8B^2 + r_{\min})^2 - r_{\min} + \widetilde{r_o}) \Leftarrow \\
\pi/m \leq \widetilde{r_o}/(1/8B^2 + 2B^2(6 + 1/8B^2 + r_{\min})^2 - r_{\min} + \widetilde{r_o}) \Leftrightarrow \\
m \geq (\pi/\widetilde{r_o}) \cdot (1/8B^2 + 2B^2(6 + 1/8B^2 + r_{\min})^2 - r_{\min} + \widetilde{r_o}) \Leftarrow \\
m \geq (\pi/r_o^l) \cdot (1/8B^2 + 2B^2(6 + 1/8B^2 + r_{\min})^2 - r_{\min} + r_o^u + 1/B^{c_3}) \Leftarrow \\
m \geq (\pi/r_o^l) \cdot (1/8B^2 + 2B^2(6 + 1 + r_{\min})^2 - r_{\min} + r_o^u + 1)
\end{aligned}$$

Choosing the radii. For a tight packing of the outer disks around the central disk, the radius of the central disk can be described as $r_c = r_o/\sin(\pi/m) - r_o$. By using the Pythagorean Theorem, we obtain the following (see Figure 18).

$$\begin{aligned}
\sqrt{(r_o/\sin(\pi/m))^2 - r_o^2} = d_3 = d_1 + d_2 = \\
\sqrt{(r_o + r_{\min})^2 - (r_o - 6 - r_{\min})^2} + \sqrt{(r_o/\sin(\pi/m) - r_o + r_{\min})^2 - (6 + r_{\min})^2}
\end{aligned}$$

By keeping in mind that a lower bound for r_o is $r_o^l = 6$ we can determine a unique solution:

$$r_o = \frac{r_{\min}\sin(\pi/m) - 3r_{\min} - 6 - 2\sqrt{2r_{\min}^2 + 6r_{\min} - 2r_{\min}^2\sin^2(\pi/m) - 6r_{\min}\sin^2(\pi/m)}}{\sin(\pi/m) - 1}$$

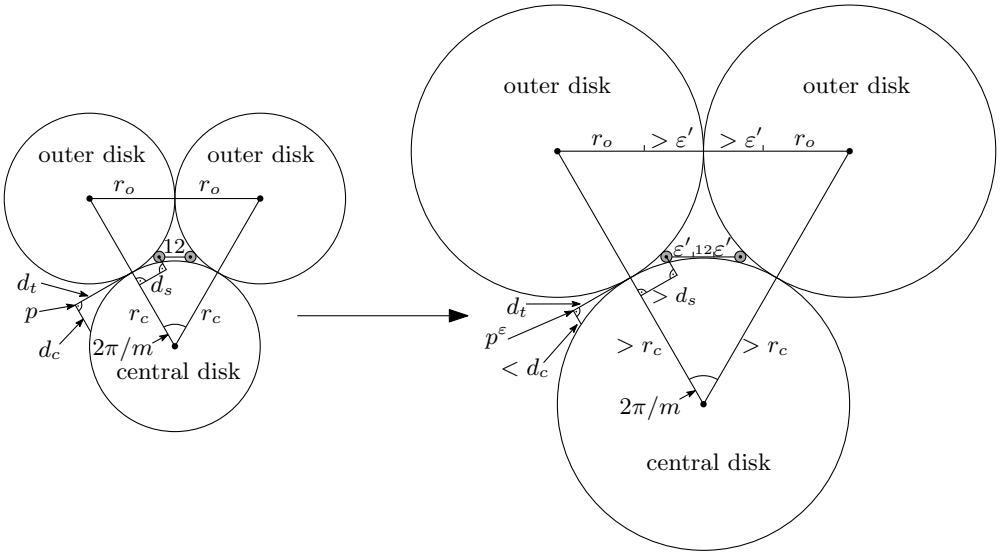


Figure 19: Increasing the separator distance by $2\varepsilon'$ increases the outer disks' radii by at least ε'.

Precise computation of this formula can take a superpolynomial amount of time. We will show later how to compute suitable approximations $\widetilde{r_o}$ and $\widetilde{r_c}$, such that $\bar{r_o} < \widetilde{r_o} \leq \bar{r_o} + \varepsilon_3$ and $\widetilde{r_o}/\sin(\pi/m) - \widetilde{r_o} < \widetilde{r_c} \leq \widetilde{r_o}/\sin(\pi/m) - \widetilde{r_o} + \varepsilon_4$.

First, let us consider a tight packing of the outer disks with approximated outer disk radius $\widetilde{r_o}$, $\bar{r_o} < \widetilde{r_o} \leq \bar{r_o} + \varepsilon_3$ and the corresponding precise central radius $\bar{r_c} = \widetilde{r_o}/\sin(\pi/m) - \widetilde{r_o}$. Note, that for these radii the maximum possible distance between the separator disks in a gap increases to $12 + \varepsilon_s$ for some $\varepsilon_s > 0$.

The left part of Figure 19 illustrates two outer disks bounding a gap with the original radii and the original separator distance of 12. Changing this distance to $12+2\varepsilon'$ (as depicted in the right part of Figure 19) increases the required outer and central radii for a tight packing since we do not change m and, therefore, maintain the angle between the two outer disks. In both packings, consider the tangent line between the respective left outer disk and the central disk. We travel a distance d_t along these tangent lines and arrive at points p and p^ε (see Figure 19). From these points we travel orthogonally (to the tangents) until we reach the central disk. Let d_c be the distance traveled in the original packing and observe that the traveled distance in the modified packing is smaller than d_c since the radius of the central disk is larger. On an intuitive level, this means that the funnel-shaped regions next to the tangent points become more narrow as the radii of the outer and central disks increase. This phenomenon causes separator disks (which maintain their original size) in the modified packing to be pushed farther away from the lines that connect the centers of the central and outer disks than in the original packing (the distance d_s in Figure 19 increases). For this reason, increasing the separator distance from 12 to $12 + 2\varepsilon'$ pushes the centers of the outer disks at least distance ε' to the sides since this is also the distance that the separators move to the left or right respectively. We can conclude that increasing the separator distance by $2\varepsilon'$ increases the radius of the outer disks in a tight packing by at least ε'. The implication

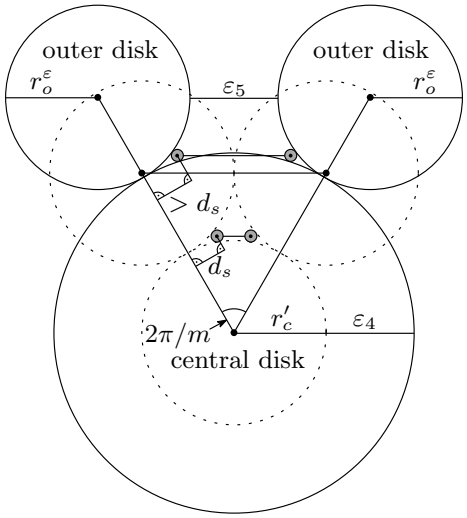


Figure 20: Increasing the central radius by ε_4 creates a distance of ε_5 between the outer disks. The distance between the separators increases by at most ε_5.

is that in a tight packing with outer disk radius $\widetilde{r_o} \leq \bar{r_o} + \varepsilon_3$ and a corresponding (precise) central radius $\bar{r_c} = \widetilde{r_o}/\sin(\pi/m) - \widetilde{r_o}$ the distance between the separators is at most $12 + 2\varepsilon_3$.

Once again, we might be unable to compute the central radius $\bar{r_c}$ precisely. Instead, we approximate it as $\widetilde{r_c}$ with $\bar{r_c} < \widetilde{r_c} \leq \bar{r_c} + \varepsilon_4 = \widetilde{r_o}/\sin(\pi/m) - \widetilde{r_o} + \varepsilon_4$, which basically pushes the outer disks to the outside as depicted in Figure 20. Assuming that the outer disks can not deviate from these positions, this creates some distance ε_5 between the outer disks in each gap. The outer disk radius remains $\widetilde{r_o}$ but the central disk radius is larger than in a tight packing and has the value $\widetilde{r_c} > \bar{r_c}$. Like in the argument in the previous paragraph, this causes the separator disks to be pushed away from the lines that connect the outer and central disks' centers. For this reason, the distance between the separators increases by at most ε_5 from at most $12 + 2\varepsilon_3$ to at most $12 + 2\varepsilon_3 + \varepsilon_5$.

So far, we have assumed that the outer disks can not deviate from their positions even though they are placed distance ε_5 apart from each other. In reality, however, the outer disks can rotate around the central disk and, therefore, the distance between two outer disks can increase to some value $\varepsilon_6 > \varepsilon_5$.

Initially, let every consecutive pair of outer disks have distance $\varepsilon_5 > 0$ to each other. We prove that $\varepsilon_6 < 2m\varepsilon_5$ by showing by induction that if we allow i of the outer disks to move, the maximum distance x_i between two outer disks is smaller than $2(i+1)\varepsilon_5$ for any $0 \leq i \leq m-1$. Clearly this holds true for $x_0 = \varepsilon_5 < 2\varepsilon_5$. Now assume that our hypothesis is true for some fixed $i \leq m-2$. Clearly, the distance x_i is maximized when we place all of the i movable disks close together and thereby create one large gap. One of the neighboring gaps is bounded by two non-movable (in step i) disks such that the distance between these disks is ε_5 as depicted in the left part of Figure 21. The distance x_{i+1} gets maximized by now allowing the previously non-movable disk next to the x_i gap to move such that the two gaps merge as illustrated in the right part of Figure 21. Consider the triangles T_1 (left)

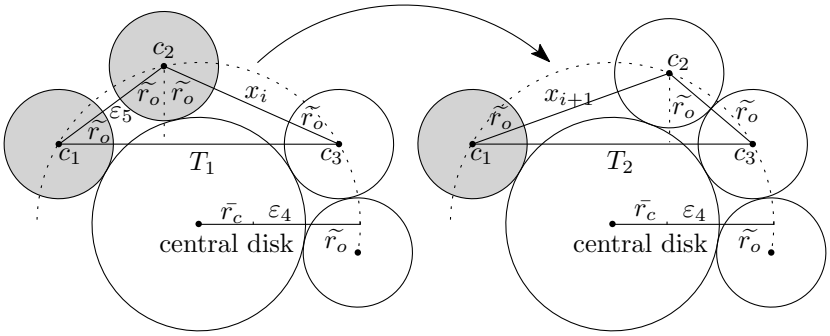


Figure 21: When allowing $i+1$ instead of i outer disks to move (non-movable outer disks in gray), the maximum distance increases at most linear in ε_5.

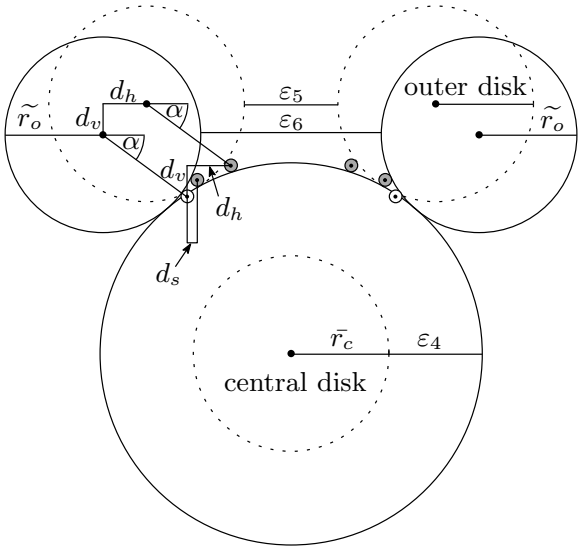


Figure 22: Increasing the distance between two outer disks from ε_5 to ε_6 increases the distance between the separators by at most $\varepsilon_6 - \varepsilon_5$ since $d_s \geq 0$.

and T_2 (right) formed by the centers c_1, c_2, c_3 in Figure 21. The base side of these triangles is identical, the height of T_2 is smaller than the height of T_1 and the circumcircle of both triangles has radius $\bar{r_c}+\varepsilon_4+\widetilde{r_o}$. We can conclude that the area of T_2 is smaller than the area of T_1. Recalling that the area of a triangle T with sides a, b, c can be described as $abc/(4r)$ where r is the radius of the circumcircle of T, we obtain $2\widetilde{r_o}(2\widetilde{r_o}+x_{i+1}) < (2\widetilde{r_o}+\varepsilon_5)(2\widetilde{r_o}+x_i)$ and, hence, $x_{i+1} < x_i + \varepsilon_5 + \varepsilon_5 x_i/(2\widetilde{r_o}) < 2(i+1)\varepsilon_5 + \varepsilon_5 + \varepsilon_5(2(i+1)\varepsilon_5)/(2\widetilde{r_o})$ by our induction hypothesis. By choosing ε_4 and, therefore, ε_5 such that $2m\varepsilon_5 < 1$ we obtain that $x_{i+1} < 2(i+1)\varepsilon_5 + \varepsilon_5 + \varepsilon_5/(2\widetilde{r_o}) < 2(i+1)\varepsilon_5 + \varepsilon_5 + \varepsilon_5 = 2(i+2)\varepsilon_5$, which concludes our induction proof. Thus, the maximum distance between two outer disks increases to at most $\varepsilon_6 < 2m\varepsilon_5$. This increases the maximum distance between the separators to at most $12 + 2\varepsilon_3 + 2m\varepsilon_5$ as illustrated in Figure 22.

With basic trigonometry, we determine that $2\widetilde{r_o} + \varepsilon_5 = 2(\bar{r_c} + \varepsilon_4 + \widetilde{r_o})\sin(\pi/m)$ and $2\widetilde{r_o} = 2(\bar{r_c} + \widetilde{r_o})\sin(\pi/m)$, see Figure 20. We combine these two equalities and ob-

tain $\varepsilon_5 = 2\varepsilon_4 \sin(\pi/m) < 2\varepsilon_4\pi/m < 2\varepsilon_4 \cdot 4/m$. The maximum distance between two separators is, therefore, at most $12 + 2\varepsilon_3 + 16\varepsilon_4$.

Recall that by Lemma 6, Properties 1 and 2 hold true for our radius function $\mathbf{r}$ as long as the maximum distance between two separators is at most $12 + \varepsilon_s$ with $\varepsilon_s = 1/4B^2$. We can now simply choose $\varepsilon_3 = 1/B^{c_3}$ and $\varepsilon_4 = 1/B^{c_4}$ such that $2\varepsilon_3 + 16\varepsilon_4 \leq \varepsilon_s$. Therefore, we can conclude that the approximate radii for the outer and central disks suffice.

Approximating radii in polynomial time. It remains to argue that we can approximate our radii as required. The formulas for the exact radii $\bar{r_o}$ and $\bar{r_o}$ contain a constant number of square root and sine operations. Recall that $m = B^{c_1}$. Redefining and increasing m such that $m = 2^p$ with $2^{p-1} < B^{c_m} \leq 2^p = m$ causes no issues for our construction. Therefore, by using the half-angle formula

$$\cos\left(\frac{1}{2}x\right) = \sqrt{\frac{1+\cos x}{2}} \text{ for } 0 < x < \pi,$$

and using $\sin x = \sqrt{1-\cos^2 x}$ for $0 < x < \pi/2$, we can replace each sine operation in our formulas by $p = \log_2 m$ nested square root operations. In total, we therefore perform $O(\log m) = O(\log B^{c_1})$ square root operations. Individually, each square root approximation can be performed in polynomial time using Heron's quadratically converging method since we can easily determine constant upper and lower bounds for each square root term and use these as the initiation values. In order to approximate the nested square roots, we need to increase the approximation accuracy by an according polynomial amount. □

Lemma 2 already showed how to construct an equivalent 3-Partition instance with $3m \geq 3n$ input integers. We now have all the tools required to prove the main result of this section. Lemmas 2 and 7 show that the construction can be performed in polynomial time. Properties 1 and 2 let us show that a valid distribution of the input and separator disks among the gaps induces a solution of the 3-Partition instance and vice versa.

Theorem 2. *The WDC graph recognition problem is* NP*-hard even for stars if an arbitrary embedding is allowed.*

Proof. Given a 3-Partition instance $(\mathcal{A}, B)$, an equivalent instance $(\mathcal{A}', B')$ as in Lemma 2 and an equivalent WDC graph recognition instance can be constructed in polynomial time. The number of disks is linear in m and, thus, polynomial in B. For the input and separator disks the radius computation does not cause any complications since the output of our radius function $\mathbf{r}$ is always a polynomially bounded rational number. For the inner and outer disks, the radii can be approximated in polynomial time; see Lemma 7. Furthermore, the encoding size of the WDC graph recognition instance is polynomial in the encoding size of $(\mathcal{A}, B)$. A solution of $(\mathcal{A}', B')$ induces a valid distribution of disks among the m gaps by placing each disk triple together with two separators in each gap. Conversely, a valid distribution of the input and separator disks among the m gaps induces a solution of $(\mathcal{A}', B')$, since Properties 1 and 2 ensure that each of the m gaps contains a feasible triple and two separators. □

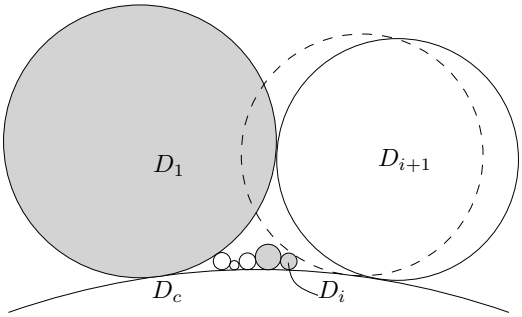


Figure 23: Deciding existence for Theorem 3. Gray disks are in L before inserting D_{i+1}. After that, the two small gray disks will be removed from L.

4.2 Recognizing embedded stars with a weighted disk contact representation

If, however, the order of the leaves around the central vertex of the star is fixed, the existence of a WDC representation can be easily decided by iteratively placing the outer disks D_1, $\ldots$, D_{n-1} tightly around the central disk D_c. A naive approach tests for collisions with all previously added disks and yields a total runtime of $O(n^2)$. However, in the following theorem we improve this to $O(n)$ by maintaining a list containing only disks that might be relevant in the future.

Theorem 3. *On a Real RAM, for an embedded, vertex-weighted star S it can be decided in linear time whether S is a WDC graph. A WDC representation respecting the embedding (if one exists) can be constructed in linear time.*

Proof. Let r_i be the radius of D_i, and assume that D_1 is the largest outer disk. Then, D_2 can be placed next to D_1 clockwise. Suppose we have already added D_2, $\ldots$, D_i. As depicted in Figure 23, tightly placing D_{i+1} next to D_i might cause D_{i+1} to intersect with a disk inserted earlier, even with D_1. Testing for collisions with all previously added disks yields a total runtime of $O(n^2)$; we improve this to $O(n)$ by keeping a list L of all inserted disks that might be relevant for future insertions. Initially, only D_1 is in L. We shall see that L remains sorted by non-increasing radius.

When inserting D_{i+1}, we traverse L backwards and test for collisions with traversed disks, until we find the largest index $j < i$ such that $r_j \in L$ and $r_{i+1} \leq r_j$. Next, we place D_{i+1} tightly next to all inserted disks, avoiding collisions with the traversed disks.

First, note that D_{i+1} cannot intersect disks preceding D_j in L (unless D_{i+1} and D_1 would intersect clockwise, in which case we report non-existence). Next, disks that currently succeed D_j in L will not be able to intersect D_{i+2}, $\ldots$, D_{n-1} and are therefore removed from L. Finally, we add D_{i+1} to the end of L. Since all but one traversed disks are removed during each insertion, the total runtime is $O(n)$. We return the constructed WDC representation if we can insert all disks tightly and there is still space left; otherwise we report non-existence. □

5 Seeded (unit) disk graphs

Let $G = (V, E)$ be a graph and $s : V \to \mathbb{R}^2$ be an injective assignment of point *seeds* to the vertices of G. We consider the task of deciding whether G has a disk representation in which each disk D_v covers the seed $s(v)$ of its respective vertex $v \in V$, i.e. $s(v) \in D_v$. We call such a representation *seeded disk contact/intersection representation* and a graph that admits such a representation a *seeded disk contact/intersection graph* (SDC/SDI graph). Atienza et al. [3] showed that SDC recognition is NP-hard even for outerplanar graphs. In Section 5.1 we extend their construction and show NP-hardness for trees. In Section 5.2 we combine the seeded and weighted version of the problem and show that recognizing seeded unit disk graphs is NP-hard even for paths. This result applies to contact as well as to intersection representations.

5.1 Seeded disk contact graphs

Atienza et al. [3, Theorem 2.3] showed that SDC graph recognition is NP-hard by creating a graph G and a seed assignment s that is realizable if and only if a given P3SAT formula φ is satisfiable. In this section we first summarize their approach and observe that the graph G is outerplanar and non-connected. Then, we describe how to modify G and s to obtain a tree G' together with a seed assignment s' which remain realizable if and only if φ is satisfiable.

Subgraphs called *chains* serve as a basic building block in the reduction by Atienza et al. [3]. Each chain has one of exactly two possible states in any realization and they are used to propagate truth states between gadgets created for the variables and clauses. A small segment of a chain is depicted in Figure 24. The seeds s and s' are the *stopper elements* of the *inner* seed p. Note how the seeds s, t_1, t_2, r_1, r_2, q and $s', t'_1, t'_2, r'_1, r'_2, q'$ are chosen such that the center of the disk D_p that covers p has to be located either in the region I_p^{true} or in the region I_p^{false}. Accordingly, D_p has to be located either to the left (the black disk D_p^{true}) or to the right (the gray disk D_p^{false}). The chosen position is propagated to the next segment of the chain with inner seed $\hat{p}$, i.e. we can use either both black positions or both gray positions.

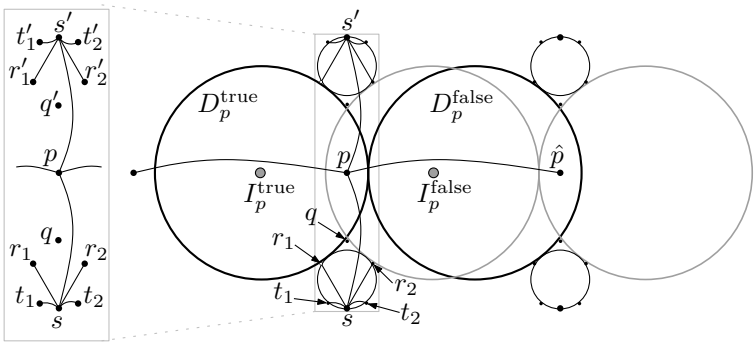
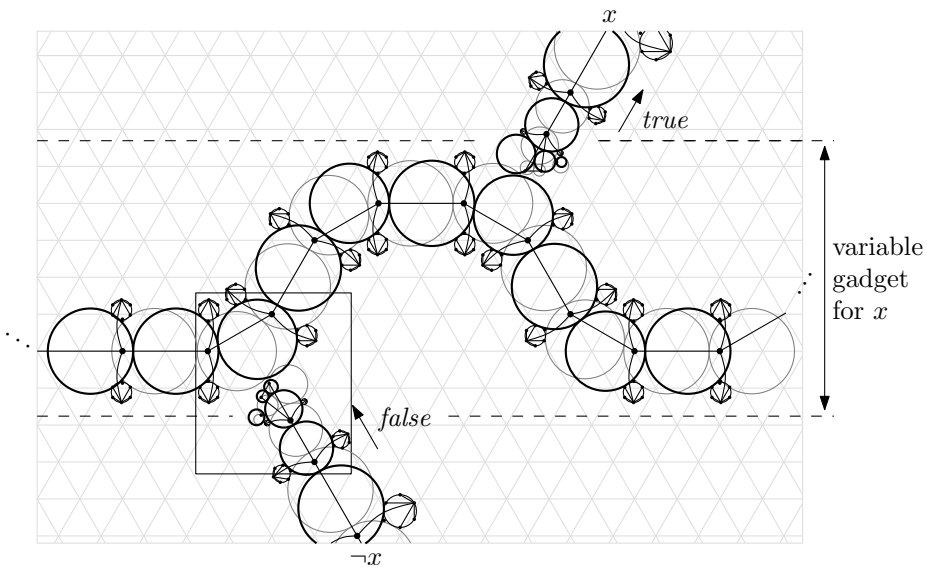
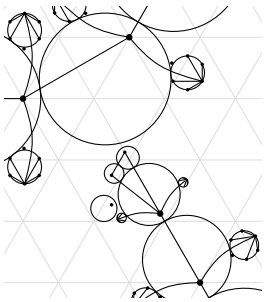


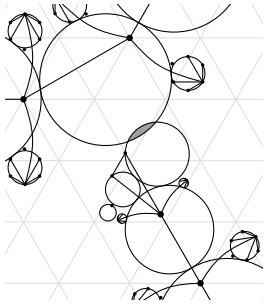
Figure 24: A small part of a chain in the construction by Atienza et al. [3]. Image used with permission of the authors.



(a) Variable gadget for x in true (false) state illustrated by the black (gray) disks.



(b) Valid realization.

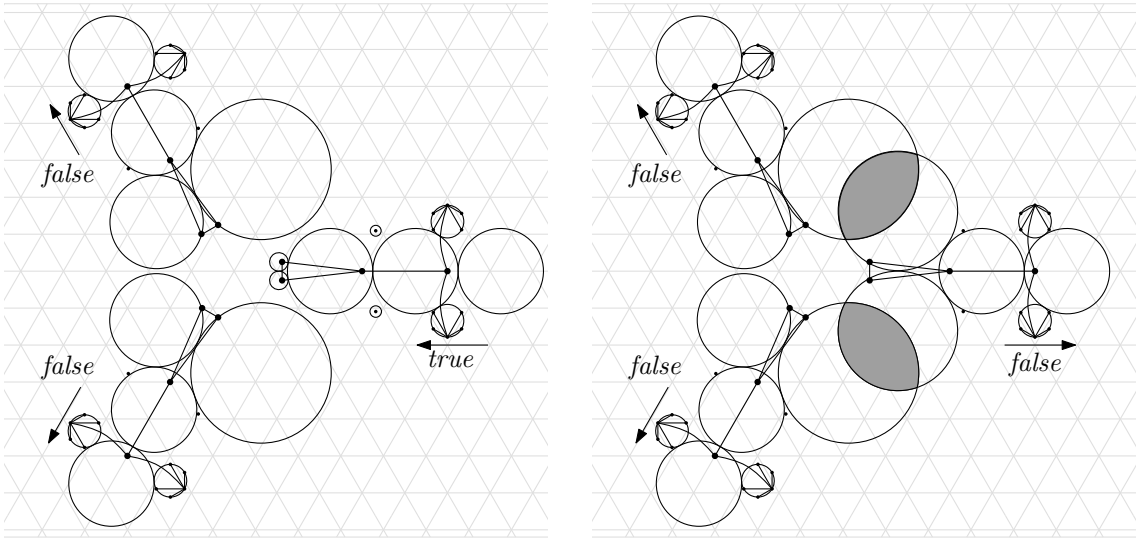


(c) Invalid realization.

Figure 25: A variable gadget in the construction by Atienza et al. [3]. Images used with permission of the authors.

Figure 25a depicts a longer chain that serves as a *variable* gadget with two states. The black disk positions corresponds to the true state and the gray positions corresponds to the false state. Additional chains connect the variable gadget for $x \in U$ to all clause gadgets whose corresponding clauses contain a literal of x. The end of such a chain is designed such that the disks are pulled towards the variable gadget if the clause's literal for x evaluates to false for the respective state of the variable gadget. This is illustrated in Figure 25b and Figure 25c. Note how in Figure 25b the two topmost disks of the bottom chain are realizable since the disks of the chain are embedded towards the variable gadget. However, in Figure 25c we see that if the disks are embedded towards the clause gadget, one of the topmost disks intersects the variable gadget. If a chain's literal evaluates to true, its disks can be embedded in either direction, see the top chain in Figure 25a.

Figure 26a depicts the *clause* gadget. The last two disks of a chain oriented away from the clause gadget are forced to have a large radius. In particular, if all three chains are oriented away from the gadget, it is impossible to find a valid realization, see Figure 26b.



(a) Valid realization. (b) Invalid realization.

Figure 26: A clause gadget in the construction by Atienza et al. [3]. Images used with permission of the authors.

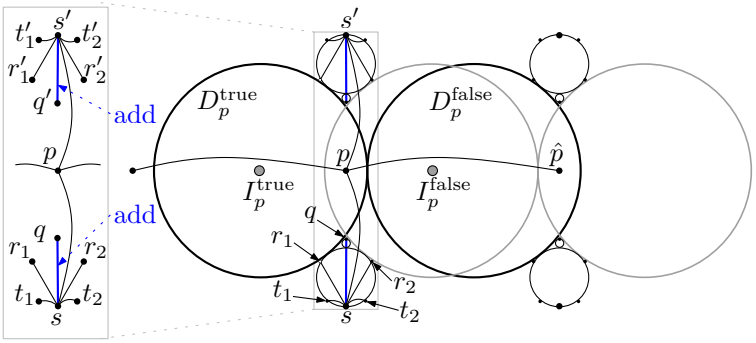
However, if at least one chain is oriented towards the clause gadget there is a valid realization due to the fact the radii of the last two disks can now be chosen very small, see Figure 26a right. This corresponds to the fact that at least one of the literals of each clause is supposed to be be satisfied.

Clearly G is non-connected and outerplanar. By a few modifications we can turn G into a tree G' to show NP-hardness for trees.

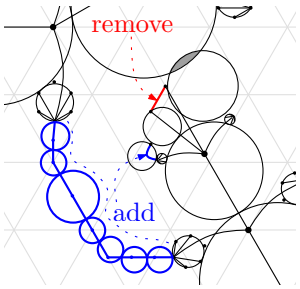
Theorem 4. *SDC graph recognition is* NP*-hard even for trees.*

Proof. The different connected components of G can easily be connected. In each of the chain segments we add two edges, marked blue in Figure 27a. We additionally shift the seeds q (q') slightly towards s (s') such that the additional edges do not change the original purpose of s, s', q and q'. Modifications to the chains connecting variable and clause gadgets are depicted in Figures 27b and 27c. We add the blue edges and vertices/disks to ensure connectivity.

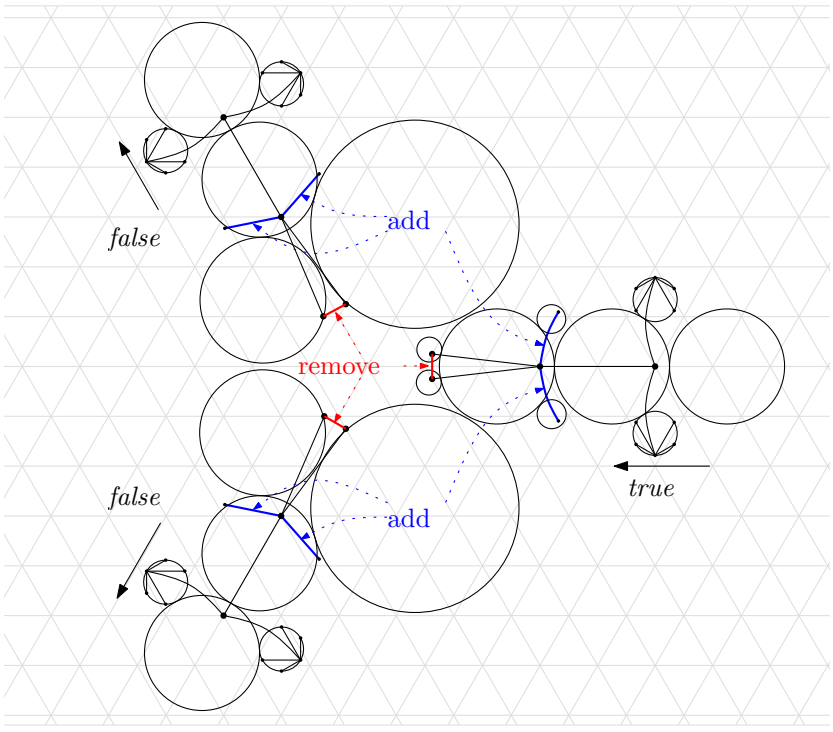
At both ends of each literal chain we find a cycle. Each of these cycles contains exactly one edge connecting two degree-2 vertices. We remove these edges, marked red in Figures 27b and 27c. The removed edges are not necessary to ensure that the disk of one of its vertices intersects the variable gadget in the case that the chain is embedded away from the variable gadget. Finally, we connect all variable gadgets by adding additional simple paths of disks. After these modifications our graph has become a tree, which is still realizable if and only if φ is satisfiable. □



(a) Modified chain.

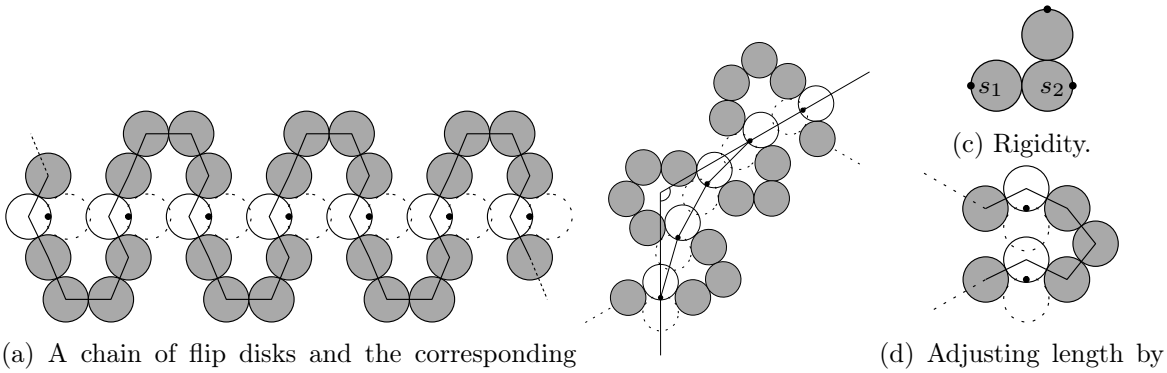


(b) Modified variable gadget.



(c) Modified clause gadget.

Figure 27: The construction of Atienza et al. [3] can be modified such that the resulting graph is a tree.



(a) A chain of flip disks and the corresponding path.

(b) A 60° turn.

(c) Rigidity.

(d) Adjusting length by increasing overlap.

Figure 28: The elemental components for the reduction in Theorem 5.

5.2 Seeded unit disk graphs

In this section we add the requirement that all disks must be unit disks (of radius 1 and diameter 2) and show that recognition of seeded unit disk contact (SUDC) graphs and seeded unit disk intersection (SUDI) graphs is NP-hard already for paths. We first show hardness for SUDC-graphs and then explain how to modify the proof for SUDI-graphs.

Theorem 5. *Recognition of SUDC-graphs is* NP*-hard, even for paths.*

Proof. We perform a reduction from P3SAT (see Section 2) to show that recognizing an SUDC-graph is NP-hard. To this end, we create a graph $G = (V, E)$, which is actually a path, and a seed assignment $s : V \to \mathbb{R}^2$ such that G can be realized as a seeded unit disk contact representation (SUDCR) with respect to s if and only if the given P3SAT formula φ is satisfiable. Due to our seed assignment, realizing representations will resemble the slanted layout $\mathcal{G}_\varphi^S$ on an isometric triangular grid as discussed in Section 2.

If a graph and a seed assignment have a unique SUDCR, we call this SUDCR and its disks *rigid*. It is easy to construct graphs with rigid SUDCRs. Start by placing two touching disks and assign their seeds s_1, s_2 to the extremal points two unit diameters apart from each other. More disks can be rigidly attached to this construction using a similar approach, where the next seed is placed at the diametral point of the intended touching point of the two disks, see Figure 28c. In our figures we show rigid disks in dark gray. A key idea for our reduction is to place two rigid disks D_1, D_2 less than one, but more than $\sqrt{3}-1 \approx 0.73$ unit diameters apart (say, $13/16 = 0.8125$ unit diameters) such that the disk D adjacent to both D_1 and D_2 in the path G can touch both disks D_1 and D_2 simultaneously in exactly two positions and these two positions overlap in a lens, in the center of which we place the seed p for D, see the close-up box in the top part of Figure 29. We say that such a disk D is a *flip* disk. Flip disks are shown in white in our figures. Two flip disks placed sufficiently close to each other (say, about $\sqrt{3}$ unit diameters between their seeds) such that their respective disk positions facing each other overlap in a lens, allow for the following information transfer: The lower flip disk in the box in Figure 29 has to be embedded in its bottom position if the upper disk is embedded in its bottom position. On the other hand, if the lower disk is

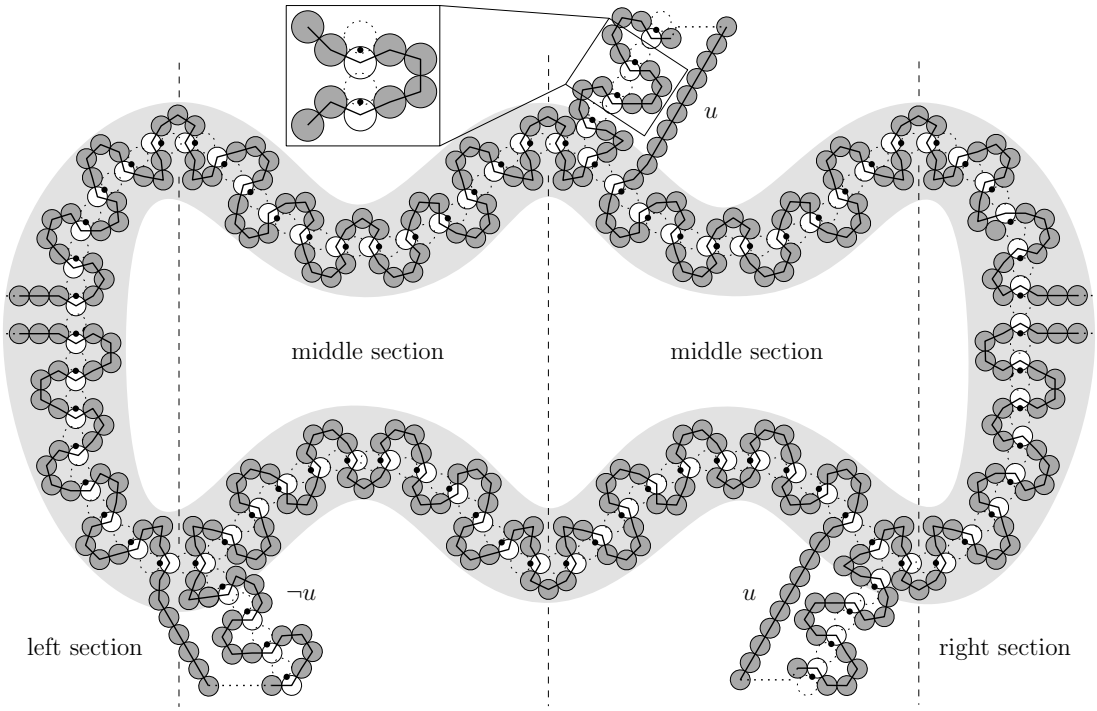
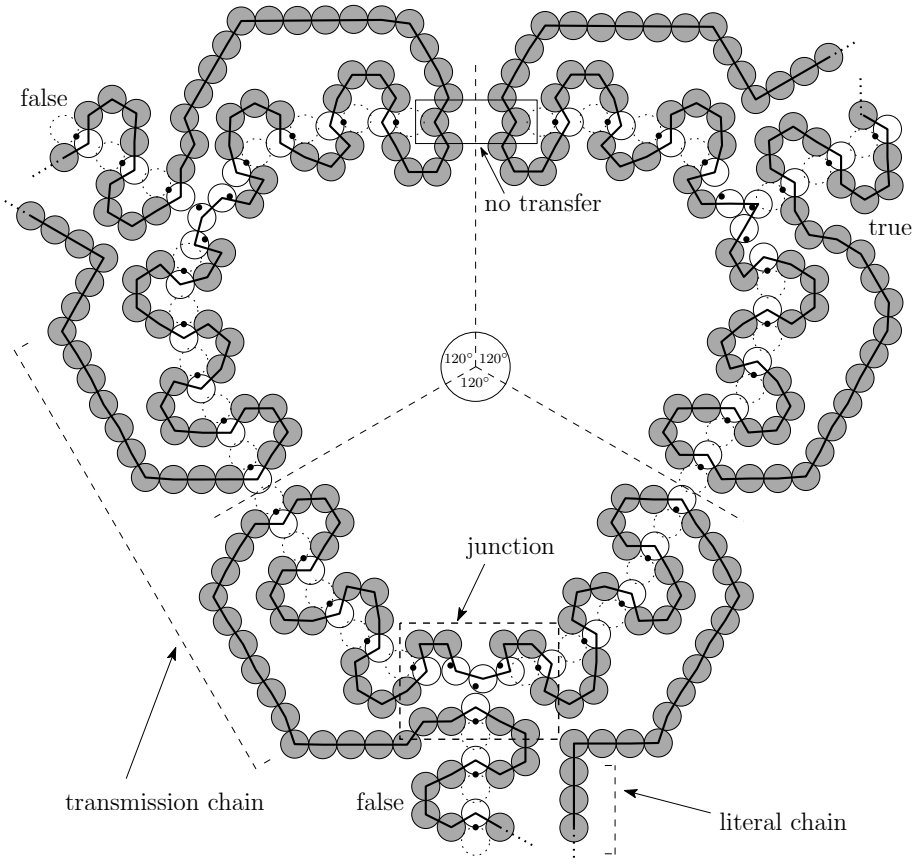


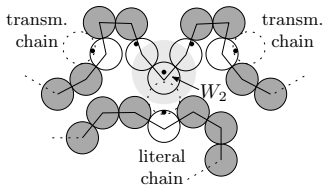
Figure 29: Variable gadget for variable u in state *true*. Gray disks are rigid. White disks are flip disks.

embedded in its top position, the upper disk has to be embedded in its top position, too. Otherwise they would overlap. This idea can be extended to create arbitrarily long *chains* of flip disks that transfer state information, see Figure 28a. Chains are flexible structures: we can implement turns (Figure 28b) and adjust the distance between two consecutive flip disks if needed (Figure 28d).

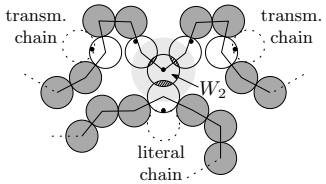
If a chain is *closed* it has exactly two possible states, either all disks are embedded clockwise or counter-clockwise; see the gray highlighted section of Figure 29. This figure depicts a *variable gadget*, one of which is created for each variable vertex of $\mathcal{G}_\varphi^S$. The two states of the closed chain emulate the truth states of the variable. The *left/right sections* are designed such that adjacent variable gadgets can be attached to each other. The required number of *middle sections*, which can easily be duplicated and concatenated, depends on the number of incident literal edges. These edges are represented by *literal* chains attached to the middle sections in one of two ways, see Figure 29, bottom left and bottom right for a negative and a positive literal, respectively. Note how the flip disks of the variable gadgets push the flip disks of the left literal chain away from the variable gadget. However, the flip disks of the right literal chain can be embedded towards the variable gadget. Similarly, for the other truth state of the variable gadget (flip disks embedded clockwise) the flip disks of the right literal chain are pushed away while the flip disks of the left literal chain can be embedded towards the gadget. We attach each literal chain such that its white flip disks are pushed away from the gadget if the corresponding literal evaluates to false with respect to the variable gadget state.



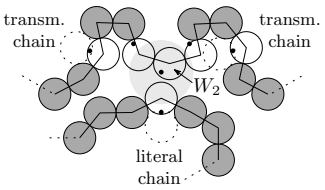
(a) A clause gadget. Two of its literal chains are embedded towards it.



(b) A junction whose literal chain is embedded away from it. Both its transmission chains can be embedded towards it.



(c) A junction whose literal chain is embedded towards it. It is not possible to embed both its transmission chains towards it.



(d) A junction whose literal chain is embedded towards it. One of its transmission chains can be embedded towards it.

Figure 30: Clause gadgets and junctions.

At *clause gadgets*, three literal chains meet symmetrically at 120° angles, see Figure 30a. These gadgets are designed such that flip disks of at most two literal chains can be embedded towards it, i.e., at most two literals can be *false*. This is achieved by three *junctions* connected by *transmission* chains, see Figure 30. A junction pushes flip disks of at least one of its two adjacent transmission chains away if the literal chain connected to the junction is embedded towards it. The reason for this is a volume argument, see Figure 30c. The light-grey disk is centered on the seed of W_2 and has radius 2. Disk W_2 has to be placed inside this light-grey disk, however, if the literal chain and both the adjacent transmission chains are embedded towards the junction there is not enough space to place W_2 without a forbidden disk overlap. Thus, at least one of the two adjacent transmission chains (Figure 30d) or the literal chain (Figure 30b) have to be embedded away from the junction. The clause gadget is not entirely symmetric: The transmission chain connecting the bottom junction and the left junction and the transmission chain connecting the bottom junction and the right junction can be embedded in both directions. However, the transmission chain connecting the left junction and the right junction is split in the middle such that its left part is always embedded towards the left junction and the right part is always embedded to the right junction. Thus, if a literal chain is pushed toward one of the top junctions its corresponding transmission chain has to be embedded towards the bottom. Recalling that literal chains are pushed towards clauses if they evaluate to *false*, we can conclude that G can be realized with respect to s if and only if φ is satisfiable.

Finally, on the left side of the left-most variable gadget, we add two rigid disks connecting the bottom and the top part so that the constructed graph indeed becomes a single path. The size of G is polynomial. It remains to express the seed positions in finite precision. We observe that our construction tolerates a little wiggle room in the exact locations of the seeds of the flip disks and the rigid disks. For each flip disk, it is sufficient to place its seed somewhere in the lens of its two intended disk position. Given that each such lens has positive area, it is clear that there is a small positive $\delta \in \mathbb{Q}$ and a Cartesian grid $\mathcal{C}$ with cell size $\delta \times \delta$ such that each lens contains a lattice point of $\mathcal{C}$, which can be chosen as the seed. To generate the seeds of the rigid disks, we observe that it is actually not necessary to ensure that their positions are unique: consider two consecutive flip disks as in the box in Figure 29. The crucial property required to make the propagation work is that their respective disk positions facing each other overlap in a lens. Given that this lens has positive area, this property tolerates a small perturbation of the positions of the rigid disks, see Figure 31. Consequently, instead of placing the seed of each rigid disk in its ideal position on the boundary of the disk, we may move it slightly towards the interior of the disk onto a lattice point of (a possibly refined version of) $\mathcal{C}$ without losing the desired propagation properties. Finally, for the three nonrigid disks of each junction, a suitable seed on a lattice point is easy to determine as their covering disks permit significant movement by construction. Hence, there is a small positive $\delta' \in \mathbb{Q}$ with $\delta' \leq \delta$ and a Cartesian grid $\mathcal{C}'$ with cell size $\delta' \times \delta'$ where each seed is placed on a lattice point of $\mathcal{C}'$ such that G can be realized with respect to new seed assignment on $\mathcal{C}'$ if and only if φ is satisfiable. Our construction is made up of multiple copies of a small number of constant sized building blocks: the clause gadget, the left, middle and right sections of the variable gadgets, and segments of chains in different orientations. The intended realizations of multiple copies of a building block are just translates of each other. It follows that the value δ' is actually

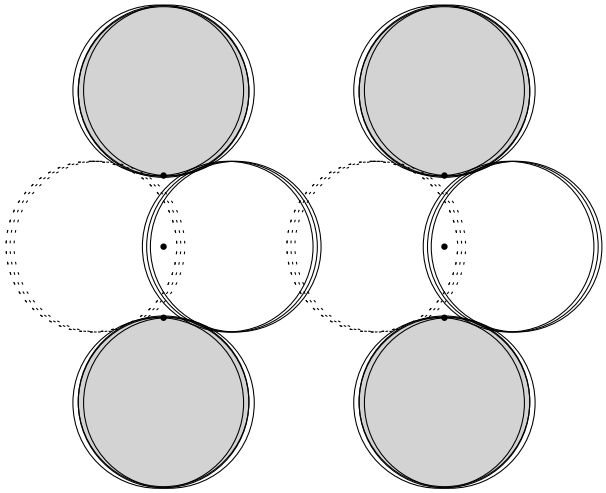


Figure 31: The propagation property of two consecutive flip disks is maintained even of the seeds of their adjacent rigid disks are slightly perturbed, as long as every possible right position of the left flip disk intersects every possible left position of the right flip disk.

independent of the input φ (i.e., δ' is a constant) since it suffices to determine a value of δ' that is suitable for each of the constantly many building blocks of constant size. In other words, the computation of δ' is not part of the reduction itself. Consequently, the size of the grid $\mathcal{C}'$ is polynomial (since the size of the original layout $\mathcal{G}_\varphi^S$ is polynomial) and so is the size of the coordinates encoding our seeds. □

Next, we extend the above reduction for seeded unit disk contact graphs to the case of seeded unit disk intersection graphs.

Theorem 6. *Recognition of SUDI-graphs is* NP*-hard, even for paths.*

Proof. The reduction for intersection graphs is very similar to the reduction of Theorem 5. Therefore we first discuss in the following the similarities and then discuss in detail the few differences. On a high level, we construct the same types of gadgets for variables, literals, and clauses based on a path intersection graph $G = (V, E)$ and a seed assignment $s\colon V \to \mathbb{R}^2$ for a P3SAT formula φ. On a closer look, the gadgets consist again of some rigid disks and some flip disks with two combinatorially different placement options. The flip disks encode the truth values of variables and transfer this information into the clause gadgets. The rigid disks can be represented in exactly the same way as in Theorem 5 (recall Figure 28c): If two unit disks representing an edge $(u, v) \in E$ must cover two seeds s_1 and s_2 with distance 2, they must actually be centered on the line segment $\overline{s_1 s_2}$ and intersect in a single touching point. So if we keep the same seed assignment for the rigid disks as in the proof of Theorem 5, the fact that disks may now overlap has no effect since we force each rigid subpath to be maximally stretched while all rigid disks cover their respective seeds.

It remains to adapt the representation of flip disks and junctions and we first consider the flip disks. Let u be a vertex corresponding to a flip disk D in between of two rigid

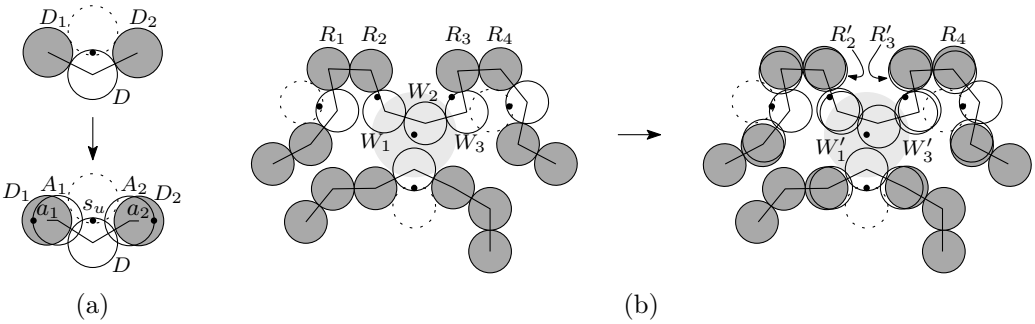


Figure 32: Adaptation of the flip disks (a) and the junctions (b) for SUDI-graph recognition.

neighbors v_1 and v_2, i.e., the reduction in Theorem 5 had edges (v_1, u) and $(u, v_2) \in E$. Here, we subdivide the edges incident to u and create two new vertices w_1 and w_2 acting as *articulation disks* for the flip disk, which are now connected by edges (v_1, w_1), (w_1, u), (u, w_2), and (w_2, v_2) (see Figure 32a). The seeds a_1 and a_2 for w_1 and w_2 are placed near the diametral points of the rigid disks D_1 and D_2 for v_1 and v_2 such that the articulation disks A_1 and A_2 can protrude into the space between D_1 and D_2 as two thin lunes. Now the flip disk D for u must cover its seed s_u centered between D_1 and D_2, it may not intersect either of D_1 and D_2, but it must intersect both A_1 and A_2. Hence this construction maintains the property of the flip disk that it has exactly two combinatorially different extremal positions; it is either flipped to the bottom (solid disk) or to the top (dotted disk), while the positions in between the extremal configurations are excluded due to the disjointness with D_1 and D_2. Finally, the new flip disk gadget uses exactly the same space as before and all chains can remain in their exact same position.

The second change in the gadget construction involves the junctions of the clause gadget. Let $P = (R_1, R_2, W_1, W_2, W_3, R_3, R_4)$ be the subpath consisting of the three white disks and the four adjacent rigid disks in a junction, see Figure 32b (left). We subdivide the four central edges of P and obtain the path $P' = (R_1, R_2, R'_2, W_1, W'_1, W_2, W'_3, W_3, R'_3, R_3, R_4)$ using four extra *twin disks*. The seed s'_2 of R'_2 is chosen in the interior of R_2 very close to the touching point $x \in R_1 \cap R_2$ so that the disk R'_2 has to be placed almost congruent to R_2 since $R_1 \cap R'_2$ has to be empty. For very small distances $||s'_2 - x||$ the possible placements for W_1 almost match the possible placements in the original construction for contact disks, i.e., W_1 is forced to almost touch R_2. Similarly, we can force W_3 to almost touch R_3 by choosing the seed of R'_3 appropriately. Finally the seeds for W'_1 and W'_3 are chosen to be very close to the seeds of W_1 and W_3, respectively, such that the central disk W_2 must be disjoint from W_1 and W_3 but at the same time intersect their twin disks W'_1 and W'_3. Thus, the same volume argument as in the contact representation scenario yields that at least one of the two transmission chains or the literal chain have to be embedded away from the junction (in Figure 32b the transmission chain on the top right).

Finally, the reduction remains of polynomial size and we can use the same grid snapping of the seeds as in the proof of Theorem 5 to obtain a polynomial-size representation of G and s. □

6 Conclusion

In this paper we considered the problem of recognizing weighted and/or seeded disk contact and disk intersection graphs. In particular, we considered these problems for several natural graph classes nested between paths and general planar graphs in order to strengthen previous hardness results and achieve tight gaps between tractability and NP-hardness. In the following we summarize our results and the remaining open problems, recall also Table 1.

- We showed that for non-embedded unit disk contact graphs the recognition problem is NP-hard for the class of outerplanar graphs. For trees the problem remains open. We conjecture that for the subclass of caterpillars it can be decided in linear time whether they are UDC-realizable. More precisely, we conjecture that a caterpillar is UDC-realizable if and only if its maximum vertex degree is 5 and between any two subsequent degree-5 vertices on the caterpillar's backbone path there is at least one backbone vertex of degree at most 3.[2]

- For the case that the weights are not necessarily uniform we showed NP-hardness for stars. This can be considered a tight border of tractability since the problem becomes trivial for paths.

- We also achieved a small gap for the case of recognizing embedded weighted contact graphs by providing a linear-time algorithm for stars. For trees the problem is NP-hard [5]. It remains open whether recognizing weighted and unit disk contact graphs is still NP-hard when restricted to embedded caterpillars.

- We settled the complexity question of recognizing seeded unit disk contact/intersection graphs by showing NP-hardness for paths, one of the most basic graph classes.

- We showed that recognizing non-embedded seeded disk contact graphs (without any assigned weights) is NP-hard for trees. The complexity of recognizing seeded disk contact graphs for paths remains open.

References

[1] M. Alam, D. Eppstein, M. Goodrich, S. Kobourov, and S. Pupyrev. Balanced circle packings for planar graphs. In C. Duncan and A. Symvonis, editors, *Graph Drawing (GD'14)*, volume 8871 of *LNCS*, pages 125–136. Springer Berlin, Heidelberg, 2014.

[2] H. Alt, K. Buchin, S. Chaplick, O. Cheong, P. Kindermann, C. Knauer, and F. Stehn. Placing your coins on a shelf. *Journal of Computational Geometry*, 9(1):312–327, 2018.

[2] The sufficiency of this condition is quite obvious. In an earlier version of this article [17], we also claimed its necessity, but the proof was flawed and the required arguments appear to be more involved than originally anticipated.

[3] N. Atienza, N. de Castro, C. Cortés, M. Á. Garrido, C. I. Grima, G. Hernández, A. Márquez, A. Moreno-González, M. Nöllenburg, J. R. Portillo, P. Reyes, J. Valenzuela, M. T. Villar, and A. Wolff. Cover contact graphs. *Journal of Computational Geometry*, 3(1):102–131, 2012.

[4] S. Bhore, M. Löffler, S. Nickel, and M. Nöllenburg. Unit disk representations of embedded trees, outerplanar and multi-legged graphs. In H. Purchase and I. Rutter, editors, *Graph Drawing and Network Visualization (GD'21)*, volume 12868 of *LNCS*, pages 304–317. Springer Cham, 2021.

[5] C. Bowen, S. Durocher, M. Löffler, A. Rounds, A. Schulz, and C. D. Tóth. Realization of simply connected polygonal linkages and recognition of unit disk contact trees. In E. Di Giacomo and A. Lubiw, editors, *Graph Drawing and Network Visualization (GD'15)*, volume 9411 of *LNCS*, pages 447–459. Springer Cham, 2015.

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

[7] S. Cabello, E. D. Demaine, and G. Rote. Planar embeddings of graphs with specified edge lengths. *Journal of Graph Algorithms and Applications*, 11(1):259–276, 2007.

[8] M. Chiu, J. Cleve, and M. Nöllenburg. Recognizing embedded caterpillars with weak unit disk contact representations is NP-hard. *CoRR*, abs/2010.01881, 2020.

[9] B. N. Clark, C. J. Colbourn, and D. S. Johnson. Unit disk graphs. *Discrete Mathematics*, 86(1–3):165–177, 1990.

[10] J. Cleve. Weak unit disk contact representations for graphs without embedding. *CoRR*, abs/2010.01886, 2020.

[11] C. R. Collins and K. Stephenson. A circle packing algorithm. *Computational Geometry*, 25(3):233 – 256, 2003.

[12] E. Di Giacomo, W. Didimo, S.-H. Hong, M. Kaufmann, S. G. Kobourov, G. Liotta, K. Misue, A. Symvonis, and H.-C. Yen. Low ply graph drawing. In *Information, Intelligence, Systems and Applications (IISA'15)*, pages 1–6. IEEE, 2015.

[13] D. Dorling. Area Cartograms: Their Use and Creation. In *Concepts and techniques in modern geography*. University of East Anglia: Environmental Publications, 1996.

[14] M. R. Garey and D. S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman & Co., 1990.

[15] W. Hale. Frequency assignment: Theory and applications. *Proc. IEEE*, 68(12):1497–1514, 1980.

[16] R. Inoue. A new construction method for circle cartograms. *Cartography and Geographic Information Science*, 38(2):146–152, 2011.

[17] B. Klemz, M. Nöllenburg, and R. Prutkin. Recognizing weighted disk contact graphs. In E. Di Giacomo and A. Lubiw, editors, *Graph Drawing and Network Visualization (GD'15)*, volume 9411 of *LNCS*, pages 433–446. Springer Cham, 2015.

[18] D. E. Knuth and A. Raghunathan. The problem of compatible representatives. *SIAM Journal on Discrete Mathematics*, 5(3):422–427, 1992.

[19] P. Koebe. Kontaktprobleme der konformen Abbildung. In *Ber. Sächs. Akad. Wiss. Leipzig, Math.-Phys. Klasse*, volume 88, pages 141–164, 1936.

[20] D. Lichtenstein. Planar formulae and their uses. *SIAM Journal on Computing*, 11(2):329–343, 1982.

[21] B. Mohar. A polynomial time circle packing algorithm. *Discrete Mathematics*, 117(1):257 – 263, 1993.

[22] F. P. Preparata and M. I. Shamos. *Computational Geometry, An Introduction*. Springer, Heidelberg, 1985.

[23] J.-M. Robert and G. Toussaint. Computational geometry and facility location. In *International Conference on Operations Research and Management Science*, pages 11–15, 1990.

[24] K. Stephenson. Circle packing: A mathematical tale. *Notices of the AMS*, 50(11):1376–1388, 2003.

[25] E. Welzl. Smallest enclosing disks (balls and ellipsoids). In H. Maurer, editor, *New Results and New Trends in Computer Science*, volume 555 of *LNCS*, pages 359–370. Springer Berlin, Heidelberg, 1991.