Recognizing Weighted Disk Contact Graphs

Boris Klemz* Martin Nöllenburg* Roman Prutkin*

Abstract

Disk contact representations realize graphs by mapping vertices to interior-disjoint disks in the plane such that disks touch each other if and only if the corresponding vertices are adjacent. Deciding whether a vertex-weighted graph can be realized so that the disks' radii coincide with the vertex weights has been proven NP-hard. In this work, we analyze the problem for special graph classes and show that it remains hard even for very basic ones, thereby strengthening previous NP-hardness results. On the positive side, we present linear-time algorithms for two restricted versions of the problem. $Perimeter(S)$ vs $Perimeter(S)$

1 Introduction

A disk intersection representation is a set of disks in the plane that can be interpreted as a graph containing a vertex for each of its disks and an edge for each pair of intersecting disks. *Disk intersection graphs* are graphs that have a disk intersection representation and generalize *disk contact graphs*, that is, graphs that have a disk intersection (or contact) representation with interior-disjoint disks. Koebe's Theorem [9] is a classic result in graph theory that states that any planar graph is a disk contact graph, and for any disk contact representation it is easy to obtain a planar drawing of the realized graph.

Application areas for disk intersection/contact graphs include modeling physical problems like wireless communication networks [6], covering problems like geometric facility location [10, 11], visual representation problems like the generation of area cartograms [4] and many more (various examples are given by Clark et al. [3]). Often, 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] recently studied graphs having disk contact representations, 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 disk radii or areas reflect the corresponding vertex weights, for example, for value-by-area circle cartograms [7]. Clearly, there exist vertex-weighted planar graphs that can not be realized as disk contact graphs, and the corresponding recognition problem is NP-hard even if all vertices are weighted uniformly [2].

We examine the aforementioned scenario more closely and explore disk contact representations for special graph classes. We show that it is NP-hard to decide whether a realization with uniform radii exists even if the input graph is outerplanar and even if a combinatorial embedding is provided. On the other hand, we can decide in linear time whether a caterpillar is a disk contact graph with uniform disk radii. If the vertex weights are not necessarily uniform, the recognition problem becomes NP-hard even for stars, but it can be solved in linear time for a given combinatorial embedding.

2 Unit disk contact graphs

In this section we are concerned with the problem of deciding whether a given graph is a unit disk contact (UDC) graph, that is, whether it can be realized as a unit disk contact representation. It is known that this is generally NP-hard for planar graphs [2], but it remained open for which subclasses of planar graphs the realizability problem can be efficiently decided and for which subclasses NP-hardness still holds.

We show that for caterpillars we can decide the realizability problem (and construct a representation if it exists) in linear time, whereas the problem remains NP-hard for outerplanar graphs.

Recognizing realizable caterpillars. Let $G = (V, E)$ be a caterpillar graph, that is, a tree for which a path remains after removing all leaves. Let $P = (v_1, \ldots, v_k)$ be this so-called *inner path* of G. It is well known that six unit disks can be tightly packed around one central unit disk, but then any two consecutive outer disks necessarily touch and form a triangle with the central disk. This is not permitted in a caterpillar and we obtain that in any realizable caterpillar the maximum degree $\Delta \leq 5$. For $\Delta \leq 4$ it is easy to see that G can always be realized as shown in Fig. 1.

However, not all caterpillars with $\Delta = 5$ can be realized. For example, two degree-5 vertices on P separated by zero or more degree-4 vertices cannot be realized, as they would again require tightly packed disks inducing cycles in the contact graph.

*Institute of Theoretical Informatics, Karlsruhe Institute of Technology (KIT), Germany. boris.klemz@student.kit.edu, noellenburg@kit.edu, roman.prutkin@kit.edu

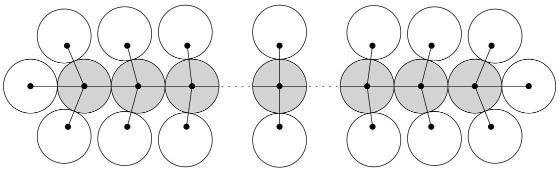

Fig. 1: For $\Delta \leq 4$ any caterpillar can be realized.

It turns out that a simple iterative pass along P suffices to decide the realizability of G and find a realization if possible. We place a disk D_1 for v_1 at the origin and attach its leaf disks *leftmost*, that is, symmetrically pushed to the left with a sufficiently small distance between them. In each subsequent step, we place the next disk D_i for v_i on the bisector of the *free space*, i.e. the maximum cone with origin in D_{i-1}'s center containing no previously inserted neighbors of D_{i-1} or D_{i-2} and attach the leaves of D_i in a leftmost and balanced way, see Fig. 2. For odd numbers of leaves this leads to a change in direction of P, but by alternating upward and downward bends for subsequent odd-degree vertices we can maintain a horizontal monotonicity, which ensures that leaves of D_i can only collide with leaves of D_{i-1} and D_{i-2}. If we fail to place the disks correctly, we claim that no UDC representation of G exists; otherwise we return the constructed realization.

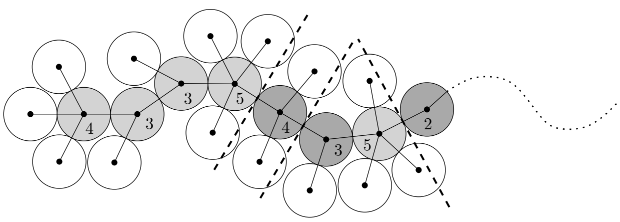

Fig. 2: Incremental construction of a realization. Narrow disks are dark gray, wide disks are light gray.

For a sketch of correctness, consider the tangent line ℓ_i between two adjacent disks D_{i-1} and D_i on the inner path. We say that P is *narrow* at v_i if some leaf disk attached to D_{i-1} intersects ℓ_i; otherwise P is *wide* at v_i. We observe that in our construction P gets narrow precisely when a degree-5 vertex of P is encountered. But it is generally true in any representation that P gets narrow after a degree-5 vertex. If P is narrow at v_i this means that at most three disjoint disks touching D_i can still be placed and thus it must be $\deg(v_i) \leq 4$. Each vertex of degree 4 inherits the narrow/wide status of its predecessor. Vertices with degree 3 or less make P wide again.

This idea leads to a combinatorial characterization (and decision algorithm) of caterpillars with a UDC representation, based on the property that between any two degree-5 vertices on P there must be at least one vertex of degree at most 3.

Theorem 1 *For a caterpillar G it can be decided in linear time whether G is a UDC graph. A realization (if one exists) can be constructed in linear time on a Real RAM.*

Hardness for outerplanar graphs. We perform a polynomial reduction from the classic NP-complete 3SAT problem to show NP-hardness of the UDC-realizability problem for outerplanar graphs. Here, we sketch only the main ideas of the reduction and refer to Klemz [8, Chapter 2] for more details.

Let φ be a Boolean 3SAT formula with a set $\mathcal{U} = \{x_1, \ldots, x_n\}$ of n variables and a set $\mathcal{C} = \{c_1, \ldots, c_m\}$ of m clauses, where each c_i contains three literals over $\mathcal{U}$. We create the *literal-clause-graph* $G_\varphi = (\mathcal{U} \cup \bar{\mathcal{U}} \cup \mathcal{C}, E)$, where $\bar{\mathcal{U}}$ is the set of negative literals over $\mathcal{U}$. The set E contains for each clause $c \in \mathcal{C}$ the edge (c, x) if literal x occurs in c and the edge $(c, \bar{x})$ if literal $\bar{x}$ occurs in c. Based on G_φ we create an outerplanar graph G'_φ that has a UDC representation if and only if the formula φ is satisfiable.

Arguing about UDC realizations of certain subgraphs becomes a lot easier, if there is only a single unique geometric representation (up to rotation, translation and mirroring). We call such a representation *rigid*. Using an inductive argument, we can show the following lemma about rigid UDC structures.

Lemma 2 *A unit disk contact representation whose UDC graph is biconnected, internally triangulated and outerplanar is rigid.*

The main building block of the reduction is a *wire gadget* in G'_φ that comes in different variations but always consists of a rigid tunnel structure containing a rigid bar that can be flipped into different tunnels around its centrally located articulation vertex. Each wire gadget occupies a square tile of fixed dimensions so that different tiles can be flexibly put together in a grid-like fashion. The bars stick out of the tiles in order to transfer information to the neighboring tiles. Some special tiles of the variable gadgets consist of tunnels without bars or with very long bars. Finally, we construct *crossing gadgets* that correctly transmit information along both axes of a tunnel crossing. Figure 3 shows a schematic view of how the gadget tiles are arranged to form a layout of G_φ.

The main idea behind the reduction is as follows. Each variable gadget contains one long horizontal bar that is either flipped to the left (*false*) or to the right (*true*), see Fig. 4(b). Consequently, each wire gadget of a literal edge connecting a variable gadget to a clause gadget must flip its chain of bars towards the clause if the literal is false. Finally, each clause gadget has one central T-shaped wire gadget, whose bar needs to be placed inside one of the three incoming

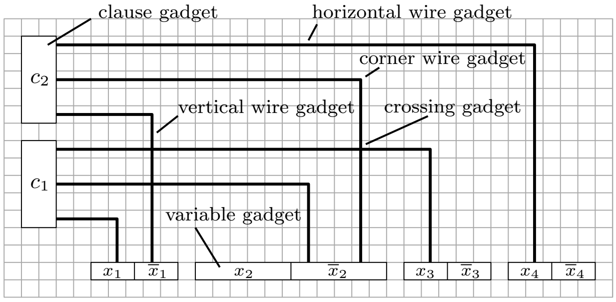

Fig. 3: High-level structure of the construction for the 3SAT formula $\varphi = (x_1 \vee \bar{x}_2 \vee x_3) \wedge (\bar{x}_1 \vee \bar{x}_2 \vee x_4)$.

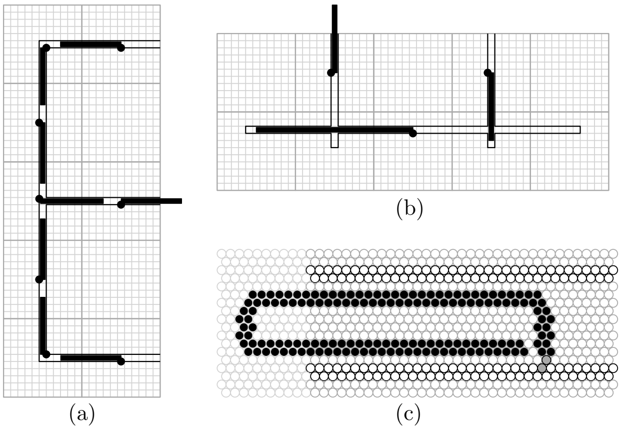

Fig. 4: (a) Clause gadget with two false inputs (top and bottom) and one true input, (b) variable gadget in the state *false* with one positive (left) and one negative literal (right), (c) detailed view of a horizontal wire gadget with a rigid bar (black disks) inside a horizontal tunnel (white disks).

tunnels. This is possible if and only if at least one of the literals evaluates to *true*, see Fig. 4(a).

Clearly, all gadgets need to be realized by rigid unit disk contacts. Figure 4(c) shows a close-up of a horizontal wire gadget. The position of the bars inside the tunnels admits some slack, but it does not affect the combinatorial properties.

Finally, one needs to take care that the constructed graph is actually outerplanar and connected. This is not obvious, but can be done by introducing small gaps and a modification in the attachment of the bar in some of the horizontal wire gadgets. Moreover, the reduction can be further modified so that it remains valid for outerplanar graphs with a fixed embedding; details can be found in [8].

Theorem 3 *The UDC graph recognition problem is* NP*-hard, even for outerplanar graphs and even if a combinatorial embedding is given.*

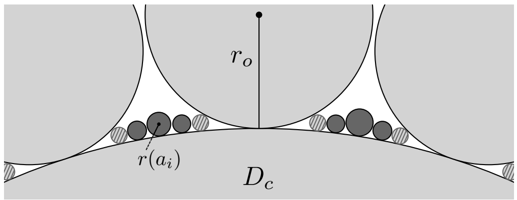

Fig. 5: Reducing from 3-Partition to prove Theorem 4. Input disks (dark) have to be distributed between gaps. Striped disks are separators.

3 Weighted disk contact graphs

In this section, we assume that each graph vertex has a positive weight, which corresponds to the disk radius of the representing disk. Deciding whether a weighted disk contact (WDC) representation respecting this radius assignment exists is obviously at least as hard as the UDC problem from Section 2.

Hardness for stars. Compared to Section 2, for an arbitrary radius assignment the corresponding recognition problem is hard for even simpler graph classes.

We perform a polynomial reduction from the well-known 3-Partition problem. Given a bound $B \in \mathbb{N}$ and a set 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 [5].

Let $(\mathcal{A}, B)$ be a 3-Partition instance. We construct a star S as well as a radius assignment r such that S has a disk contact representation respecting r if and only if $(\mathcal{A}, B)$ is a yes-instance.

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 uniform radius r_o chosen appropriately such that these disks have to be placed close together around D_c without touching, creating funnel-shaped *gaps* of equal size; see Fig. 5. Then, a contact representation 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 large enough for the input disks that represent a feasible triple to fit inside it, however, the gaps should be too small to contain an infeasible triple's disk representation.

The main challenge is finding a radius assignment satisfying the above property, although numerous additional nontrivial geometric considerations 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, which in any realization must be completely filled by special *dummy* disks, such that there

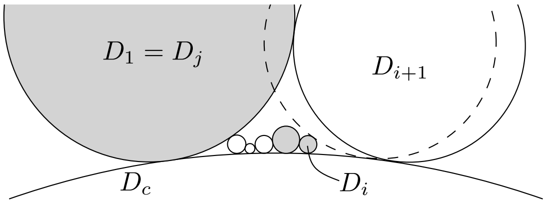

Fig. 6: Deciding existence for Theorem 5. Gray disks are in L before inserting D_{i+1}. After that, the two small gray disks will be removed from L.

are still only n gaps to distribute the input disks. 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. Again, we refer to [8, Chapter 3] for a detailed proof.

Theorem 4 *The WDC graph recognition problem is* NP*-hard even for stars.*

Stars with fixed embedding. If the order of the leaves around the central vertex of the star is fixed, the existence of a WDC representation can be decided by tightly placing the outer disks $D_1, \ldots, D_{n-1}$ around the central disk D_c iteratively. By keeping track of possible positions of the next disk, we can achieve $O(n)$ runtime.

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 Fig. 6, 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. Simply testing for collisions with all previously added disks would yield a total runtime of $O(n^2)$, which we improve to $O(n)$ by keeping a list L of inserted disks which 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 all 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 report existence if we can insert all disks tightly and there is still space left.

Theorem 5 *On a Real RAM, for a vertex-weighted star S with a given embedding it can be decided in linear time whether S is a WDC graph. A representation respecting the embedding (if one exists) can be constructed in linear time.*

4 Conclusion

We presented hardness results as well as linear-time algorithms for variants of the weighted disk contact graph recognition problem. An interesting open problem is the recognition of trees with a UDC representation. For more results, for example, regarding disk contact representations in which disks have to cover specified points, we refer to Klemz [8].

References

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

[2] H. Breu and D. G. Kirkpatrick. Unit Disk Graph Recognition is NP-hard. *Computational Geometry*, 9(1-2):3–24, Jan. 1998.

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

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

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

[6] W. Hale. Frequency Assignment: Theory and Applications. *Proc. IEEE*, 68(12):1497–1514, 1980.

[7] R. Inoue. A New Construction Method for Circle Cartograms. *Cartography and Geographic Information Science*, 38(2):146–152, 2011.

[8] B. Klemz. Weighted Disk Contact Graphs. Diploma thesis, Karlsruhe Institute of Technology, 2014.

[9] P. Koebe. Kontaktprobleme der konformen Abbildung. In *Berichte über die Verhandlungen der Sächsischen Akadmie der Wissenschaften zu Leipzig. Math.-Phas. Klasse*, volume 88, pages 141–164, 1936.

[10] J.-M. Robert and G. Toussaint. Computational Geometry and Facility Location. In *Int. Conf. Operations Research and Management Science*, pages 11–15, 1990.

[11] 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, 1991.