The Cover Contact Graph of Discs Touching a Line

Stephane Durocher* Saeed Mehrabi* Matthew Skala* Mohammad Abdul Wahid*

Abstract

We answer a question of Atienza et al. [4] by showing that the *circular CCG*$^+$ *problem* is $\mathcal{NP}$-complete. If we cover a set of objects on the plane with discs whose interiors are pairwise disjoint, then we can form a cover contact graph (CCG) that records which of the covering discs touch at their boundaries. When the input objects are themselves discs, and both input and covering discs are constrained to be touching and above the x-axis, then the circular CCG$^+$ problem is to decide the existence of a covering with a connected CCG. We also define an approximate version of this problem by allowing a small overlap between covering discs, and give an algorithm that in polynomial time finds an approximate solution for any yes-instance of the exact problem.

1 Introduction

Given a set S of n disjoint *input discs* in the plane, a covering C of S consists of n *covering discs* such that each covering disc covers exactly one input disc and no two covering discs intersect except on their boundaries. In general, the radii of the covering discs need not all be the same. The *cover contact graph* (CCG) induced by C is a graph $G = (V, E)$ such that each input disc corresponds to a vertex in V and two vertices are connected by an edge if and only if their corresponding covering discs are tangent. In other words, G is the intersection graph of a set of discs in the plane whose interiors are pairwise disjoint. Koebe's theorem [6] states that every planar graph can be represented as a *coin graph*. The coin graph of a set of discs in the plane is in fact the CCG of that set of discs. Problems related to these graphs arise in many application areas, such as wireless communication networks [5] and facility location [7].

Given a set of discs in the plane, the *circular CCG problem* asks if the given set admits a covering whose CCG is connected. Atienza et al. [4] show that the circular CCG problem is $\mathcal{NP}$-hard using a reduction from PLANAR3SAT, a constrained version of 3SAT in which the corresponding *variable-clause graph* must be planar. They also explore a variant in which the input discs are reduced to distinct points, with different kinds of connectivity required for the contact graph. They give algorithms with $O(n \log n)$ worst-case running time for 1-connectivity, and with $O(n^2 \log n)$ expected running time for 2-connectivity. They also study variants in which the covering discs are required to touch the x-axis. While they extensively examine the axis-touching case when the input is limited to distinct points, they leave open the case where the input is a set of discs and both the input and the covering discs are required to touch the x-axis. This is the circular CCG$^+$ problem, for which we show $\mathcal{NP}$-hardness in this paper. Our proof depends on very precise differences in the radii of the discs, and we show that such differences are essential to the hardness of the problem. We define an ϵ-approximate version of the circular CCG$^+$ problem and give a polynomial-time algorithm such that if the circular CCG$^+$ problem has an exact solution, then our algorithm produces an ϵ-approximate solution.

Many related problems are known. One is that of *realizability*. In addition to the set of input discs, we can be given an unlabeled planar graph G and the goal of deciding whether there exists a covering for the given input set whose CCG is G. Atienza et al. [4] show, again by reduction from PLANAR3SAT, that this realizability problem is also $\mathcal{NP}$-hard, even if the input is a set of points.

Notwithstanding Koebe's theorem that every planar graph is realizable as a coin graph (without constraining the centres of the discs), if we fix the coordinates of the vertices to make a *geometric* graph and require the discs to be centred on their respective vertices, then not every geometric graph can be so realized. Under the further constraint that the geometric graph is a tree, Abellanas and Moreno-Jiménez [3] present an $O(n \log n)$-time algorithm that decides if a given tree can be realized as a coin graph with coins centred at the vertices of the tree. For a graph that may not be a tree, they find a spanning tree and adapt their tree algorithm to solve the problem in polynomial time. Moreover, if the answer to this decision problem is affirmative, then the algorithm also computes all possible coin sets.

Abellanas et al. [1] show that given n points and n discs in the plane, it is $\mathcal{NP}$-complete to decide whether the discs can be placed in such a way that each disc is centred at one of the points and no two discs overlap. Abellanas et al. (with a different set of authors) show that the following problem is $\mathcal{NP}$-complete: Given a set of points in the plane, determine whether there are disjoint discs centred at the points such that the

*Department of Computer Science, University of Manitoba, {durocher, mehrabi, mskala, wahid}@cs.umanitoba.ca

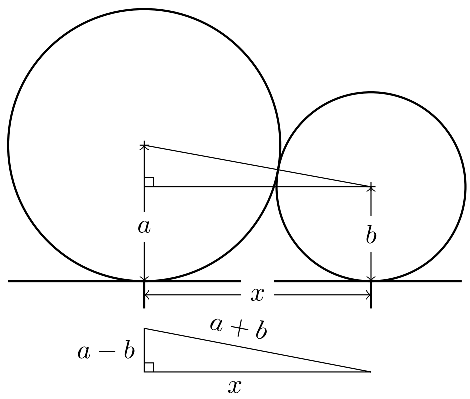

Figure 1: The constraint between two discs. See (1).

CCG of the discs is connected [2]. Note that if we relax the constraint that the discs must be centred at the given points, then the problem is polynomial-time tractable [4].

2 Proof of $\mathcal{NP}$-hardness

In this section, we show that the following problem is $\mathcal{NP}$-hard.

Given n distinct real numbers $x_1 < x_2 < \cdots < x_n$, and n nonnegative real numbers $r_1, r_2, \ldots, r_n$, the *circular CCG*$^+$ *problem* is to decide whether there exist real numbers $y_1, y_2, \ldots, y_n$ such that if C is the set of closed discs $C_1, C_2, \ldots, C_n$ where C_i is centred at (x_i, y_i) and has radius y_i (implying that C_i touches the x axis) with $y_i \geq r_i$, then the interiors of the discs in C are pairwise disjoint and the CCG induced by C is connected.

Consider two discs in the circular CCG$^+$ problem whose radii are a and b with the horizontal distance between their centres equal to x (see Figure 1). The constraint between the radii corresponds to the right triangle shown. We have:

$$\begin{aligned}(a+b)^2 &\leq (a-b)^2 + x^2 \\ \Leftrightarrow ab &\leq x^2/4\,.\end{aligned} \tag{1}$$

The constraint holds as an inequality for every pair of discs. It achieves equality if and only if the discs touch, corresponding to an edge in the CCG$^+$. Taking the logarithm, we have:

$$\log a + \log b \leq 2\log x - \log 4\,.$$

The logarithmic form of the constraint provides intuition for the hardness of the problem. Finding a circular CCG$^+$ means solving a linear program on the logarithms of the disc radii, subject to an additional constraint that the graph formed by the constraints that reach equality is a connected graph. It is intuitively reasonable that linear programming with a connectivity constraint should be $\mathcal{NP}$-hard, because if we could constrain linear programming variables to represent a connected graph, then we could constrain them to represent a 2-regular connected graph, which would be a Hamiltonian cycle.

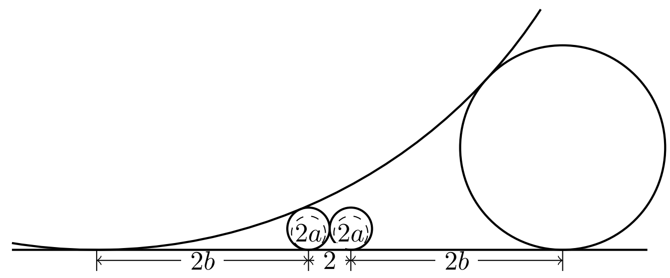

Figure 2: A non-convexity gadget.

Our proof, however, reduces from 3SAT. We will have gadgets and ways to combine them so that different values of the radius of the leftmost disc in the problem will correspond to different assignments of values to boolean variables; then we will manipulate the sets of radii that could satisfy the CCG$^+$ problem so that they correspond to exactly the assignments that satisfy the 3SAT problem.

The first step to create a hard instance of the circular CCG$^+$ problem is to create a non-convex solution set in the related linear programming problem.

2.1 A Non-convex Gadget

Figure 2 shows a gadget for creating non-convexity. The two discs in the middle are constrained to a minimum radius a slightly less than 1; their proximity to each other also gives them a maximum radius slightly greater than 1. Then in order to form a connected contact graph, the two outer discs must touch each other as shown; they cannot both touch the inner discs, but one must, and there is a choice as to which one that is. The radius y_1 of the leftmost disc is constrained to be in one of two non-overlapping intervals depending on that choice. The following lemma describes the behavior and existence of the non-convexity gadget.

Lemma 1 *We can choose the dimensions of a gadget like that shown in Figure 2 to constrain the radius y_1 of the leftmost disc such that the circular CCG$^+$ problem is satisfiable if and only if $(c \leq y_1 \leq d) \vee (e \leq y_1 \leq f)$ is true, for any positive $c, d, e,$ and f such that $1 \leq d/c = f/e < 16/9$ and $1 < e/c < (9/7)^2$.*

Proof. Consider the non-convex gadget shown in Figure 2 and let $C_1, C_2, C_3,$ and C_4 be the four discs, left to right. Their minimum radii are given by $r_1 = r_4 = 0$ and $r_2 = r_3 = a$ for some a slightly less than 1, which we will choose later. The horizontal distances are as shown: $2b$ from C_1 to C_2, 2 from C_2 to C_3, and $2b$ from C_3 to C_4, for some b we will choose later.

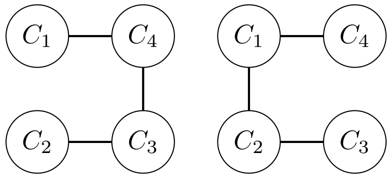

Figure 3: Allowable contact graphs for the non-convex gadget.

Each pair of discs in the gadget corresponds to an inequality constraint, which will achieve equality if and only if the discs touch.

$$y_1 y_2 \leq b^2 \qquad \text{for } (C_1, C_2) \qquad (2)$$
$$y_1 y_3 \leq (b+1)^2 \qquad \text{for } (C_1, C_3) \qquad (3)$$
$$y_1 y_4 \leq (2b+1)^2 \qquad \text{for } (C_1, C_4) \qquad (4)$$
$$y_2 y_3 \leq 1 \qquad \text{for } (C_2, C_3) \qquad (5)$$
$$y_2 y_4 \leq (b+1)^2 \qquad \text{for } (C_2, C_4) \qquad (6)$$
$$y_3 y_4 \leq b^2 \qquad \text{for } (C_3, C_4) \qquad (7)$$

We also have, as a consequence of (5) and the minimum radii $r_2 = r_3 = a \leq 1$, the constraints $a \leq y_2 \leq 1/a$ and $a \leq y_3 \leq 1/a$. For the gadget to work as intended, we must ensure that the only connected contact graphs allowed are those shown in Figure 3. Requiring one of those graphs implies that (4), (5), and either of (2) or (7) can achieve equality, but (3) and (6) cannot, nor (2) and (7) simultaneously.

We can prevent C_1 and C_3 from touching by making C_2 big enough. We have $y_2 \geq a$. That gives $y_1 \leq b^2/a$ from (2), and $y_3 \leq 1/a$ from (5). Therefore $y_1 y_3 \leq b^2/a^2$. Substituting into (3), C_1 and C_3 will be prevented from touching if $a > b/(b+1)$. This relation will hold if we choose a large enough and b small enough; $a \geq 3/4$ and $b < 3$ are sufficient. These conditions also make (6) strict, by symmetry.

It remains to prevent (2) and (7) from both achieving equality at once. Suppose they did that. Then we would have $y_1 y_2 = b^2$ and $y_3 y_4 = b^2$, so $y_1 y_2 y_3 y_4 = b^4$, but by (5), we can eliminate y_2 and y_3 and get $y_1 y_4 \geq b^4$. That will contradict (4) if $b^4 > (2b+1)^2$, or $b^2 > 2b + 1$. Solving the quadratic, (2) and (7) cannot both be equalities if $b > 1 + \sqrt{2} = 2.414\ldots$. Therefore if $3/4 < a \leq 1$ and $1+\sqrt{2} < b < 3$, the connected contact graphs that can be achieved are exactly the ones in Figure 3.

Assume we choose $3/4 < a \leq 1$ and $1 + \sqrt{2} < b < 3$, and consider the possible values for y_1 in a solution to the problem. It must fall in one of two intervals, depending on whether C_1 touches C_2, or C_3 touches C_4. One and only one of those conditions must hold, as described above. If C_1 touches C_2, then because $a \leq y_2 \leq 1/a$, we have:

$$ab^2 \leq y_1 \leq b^2/a\,. \qquad (8)$$

Symmetrically, if C_3 touches C_4, then $ab^2 \leq y_4 \leq b^2/a$, and then since C_1 and C_4 must touch each other, (4) is an equality and we have:

$$\frac{a(2b+1)^2}{b^2} \leq y_1 \leq \frac{(2b+1)^2}{ab^2}\,. \qquad (9)$$

Note that in both (8) and (9), the ratio between the upper and lower limits is equal to $1/a^2$. By choosing an $a > 3/4$, we can choose any value less than $16/9 = 1.777\ldots$ for that ratio. Dividing the lower limits in (8) and (9) gives the ratio $b^4/(2b+1)^2$. Note that a cancels out and so the ratio between the lower ends of the intervals is independent of a. By choosing b between $1+\sqrt{2}$ and 3, we can choose this ratio anywhere between 1 and $(9/7)^2 = 1.653\ldots$. Also note that we can scale the entire gadget arbitrarily, with the effect of scaling all the interval bounds by the same factor. The result follows. □

2.2 Coupling Gadgets and Interval Duplication

The next step is to combine several such gadgets into a chain, by placing them side by side in such a way that they are forced to touch. Then the radius of the leftmost disc is constrained by all the constraints of all the gadgets; we have taken the intersection of the solution sets of the individual gadgets. Figure 4 illustrates the technique.

But linking gadgets side by side is not the only way to apply one gadget's constraints to another; we can also take one gadget, or a chain of them, scale it down to be arbitrarily small, and tuck it underneath another disc as shown in Figure 5. Making it arbitrarily small means we can prevent any other discs in the problem from interfering with the contact. Moving the small disc closer to the large disc cancels out the effect of scaling them smaller, so the effect on the large disc's radius is, if we so choose, no different from placing the gadgets side by side. The difference is that because it does not require access from the sides, only from the bottom arbitrarily close to the centre, we can apply this technique to the inner discs of the gadget from Figure 2; and whatever we do to one of those is done *twice* to the leftmost disc of the gadget. The set of allowed radii for the leftmost disc becomes the union of two copies of the set of allowed radii we apply to the inner disc, separated by an adjustable scaling factor. The following lemma states this property precisely.

Lemma 2 *Given a chain of discs that constrains its leftmost member's radius to be in a set $R \subseteq \mathbb{R}$ with $\sup R \leq \sqrt{2} \inf R$, and positive reals c and d with*

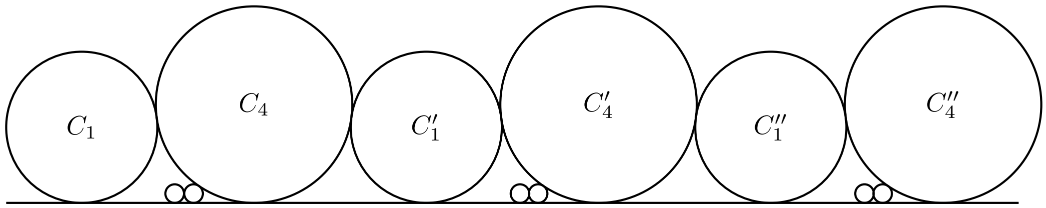

Figure 4: Gadgets coupled into a chain.

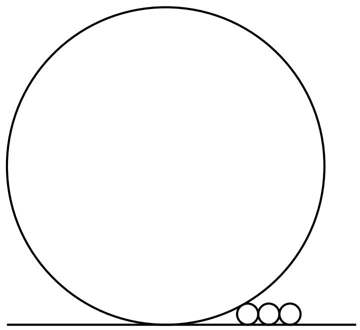

Figure 5: Hiding a chain of discs under a large disc (scale distorted for clarity).

$c < d < (9/7)^2 c$, by adding five more discs we can construct a gadget in which the leftmost disc's radius is constrained to be an element of the set $\{cy \mid y \in R\} \cup \{dy \mid y \in R\}$.

That construction can be repeated a linear number of times to create an exponential number of disjoint intervals, giving the following corollary. The radius of the leftmost disc in the problem is constrained to be in one of the intervals, but so far unconstrained as to which one.

Corollary 3 *For any nonnegative integer k and positive reals c and d with $c > 1$, $d \geq 1$, and $c^{2^k} d \leq \sqrt{2}$, we can construct a gadget using a number of discs linear in k that constrains the radius of its leftmost disc to be in the set $\{y \mid c^i \leq y \leq dc^i$ for some $i \in \{0, 1, \ldots, 2^k - 1\}\}$.*

2.3 Encoding a 3Sat Instance

Choose any instance of 3Sat. We may add a polynomial number of extra variables to it for technical reasons to be described later, but suppose that after adding those it contains n boolean variables $v_1, v_2, \ldots, v_n$. There are 2^n ways to assign values to all the variables. We will associate those with 2^n intervals, disjoint but arbitrarily close to each other and proportionally equally sized and spaced. That is, the ratios between the upper and lower bound of each interval, and between the lower bound of each interval and the lower bound of the next, are the same for all intervals. All the intervals will be contained in $(1, \sqrt{2})$. The intervals are associated with variable

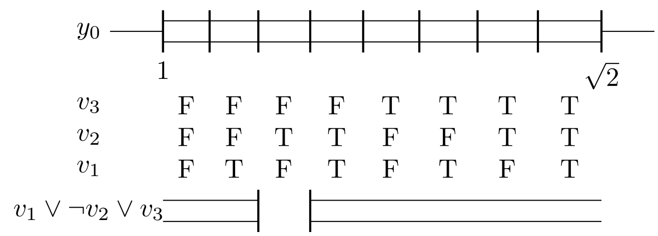

Figure 6: Satisfying a 3-clause.

assignments from all-false to all-true in binary counting order with v_1 as the least significant bit and v_n as the most significant bit. Figure 6 illustrates the encoding.

Figure 6 also illustrates how we can enforce a 3-literal disjunctive clause constraint on this encoding. Provided the clause only involves the three most significant variables, it corresponds to the negation of a single one of the eight intervals. For a clause of the form $(\neg a \vee \neg b \vee \neg c)$ or $(a \vee b \vee c)$, we just increase the minimum radius of the leftmost disc in the problem, or increase the minimum radius of the rightmost in order to have the effect of decreasing the maximum for the leftmost, and we can require the clause to be satisfied. For any other 3-clause over the three most significant variables, we can require satisfaction by intersecting with a union of two intervals that satisfy the numerical requirements of Lemma 1; so adding one more gadget to the right can have the effect of enforcing the clause as a constraint.

The clause constraint gadget only works for clauses in the three most significant variables, so we need all our clauses to be of that form when we apply it. Although other approaches more economical of variables might be possible, we will introduce three new variables for every clause, forcing them equal to the existing variables the clause is intended to constrain. This technique is illustrated in Figure 7. Here v_3 is the new variable being set equal to v_1. Observe that the set of intervals corresponding to $v_1 = v_3$ consists of two halves, and to equate v_i and v_j, $i < j$, each half is a set of 2^{j-i-1} intervals with equal proportional size and equal proportional spacing. Only the gap in the middle is different. We can create one of the halves with the gadget of Corollary 3, and then combine two copies of it with the appropriate spacing in the middle using the more general form of Lemma 2. The following lemma states precisely the ap-

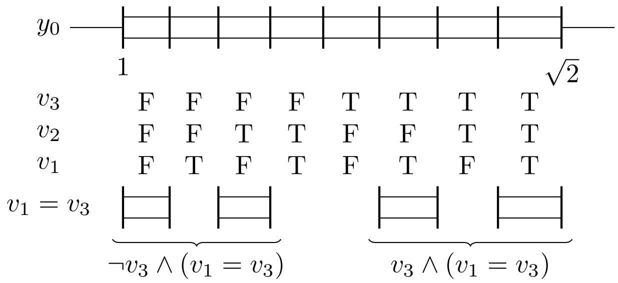

Figure 7: Duplicating a variable.

proach that we use to add a new variable and make it equal to an existing variable.

Lemma 4 *Given that n boolean variables $v_1, v_2, \ldots, v_n$ are encoded by the radius of a disc in a CCG$^+$ instance using a range from 1 to $r \le \sqrt{2}$ according to our encoding scheme, for any integer $1 \le i < n$ we can, with a number of discs linear in n, create a gadget that constrains the radius of its leftmost disc to enforce the constraint $v_i = v_n$.*

Proof. First we apply Corollary 3 with

$$\log c = \frac{\log r}{2^{n-i}},$$
$$\log d = \frac{\log r}{2^{n-i+1}}, \text{and}$$
$$k = n - i - 1$$

to create a gadget that enforces $v_i = v_n$ when v_n is false. Then we apply Lemma 2 to that gadget using

$$c = 1, \text{and}$$
$$\log d = \left[\frac{1}{2} + \frac{1}{2^{n-i+1}}\right] \log r\,.$$

□

The key observation is that the set of intervals corresponding to the statement $v_i = v_n$ may include an exponential number of intervals, but it is of a special form regardless of i and n: it is the union of two copies of a collection of proportionally equally spaced and sized intervals, and we can create it by applying Corollary 3 followed by Lemma 2.

Then we have all the pieces necessary to construct an instance of the circular CCG$^+$ problem equivalent to an instance of 3Sat. First, we calculate the number of variables we will add, which is equal to three times the number of 3-clauses. That gives us the size we need for the smallest intervals in our encoding. Note that this size comes from starting at a constant and scaling down by at most a constant amount, a linear number of times; we can represent the numbers involved in a polynomial number of bits.

We represent the variables from the original problem in a suitably narrow range of radii using Corollary 3. For each clause, we add three new variables with Lemma 4, doubling the number of variable assignments in the encoding with each one. We enforce the 3-clause. Then we proceed to the next. When we are done, we have a polynomial-sized instance of the circular CCG$^+$ problem whose leftmost disc is constrained to have a radius that represents a satisfying assignment for the original 3Sat instance; this is satisfiable if and only if the 3Sat instance was satisfiable. Therefore the following holds:

Theorem 5 *The circular CCG$^+$ problem is $\mathcal{NP}$-hard.*

3 An Approximation Algorithm

The $\mathcal{NP}$-hardness proof depends on high numeric precision. The constraints on disc radii create intervals that are exponentially small, although represented by a number of bits polynomial in the problem size. In this section we show that that precision is essential to the hardness of the problem: if we relax the problem definition in such a way as to permit imprecise solutions, then it becomes tractable.

First, discs cannot grow arbitrarily large or small. This follows from doing two rounds of simple constraint propagation (detailed proof omitted).

Lemma 6 *Any instance of the circular CCG$^+$ problem either is trivially satisfiable, or contains at least two discs with nonzero minimum size, and in the latter case we can in polynomial time compute a nonzero minimum and finite maximum size for every disc in the problem, which must be satisfied by any exact solution.*

Recall that the circular CCG$^+$ problem requires, for each pair of discs whose radii are a and b and whose horizontal distance is x, the constraint $ab \le x^2/4$; and this holds as an equality if and only if the discs are touching each other and correspond to an edge in the contact graph. Let us relax the constraint to say that for any $\epsilon > 0$ two discs are *ϵ-approximately touching* if $x^2/4 \le ab \le (1+\epsilon)x^2/4$. Then other definitions arise by analogy: the *ϵ-approximate contact graph* is the graph with a vertex for each disc and an edge between any two discs that are ϵ-approximately touching, and the *ϵ-approximate CCG$^+$ problem* is like the CCG$^+$ problem but requires a choice of radii for the discs such that the $ab \le (1+\epsilon)x^2/4$ constraint is obeyed by every pair of discs, instead of $ab \le x^2/4$, and the ϵ-approximate contact graph is connected instead of the exact contact graph necessarily being connected.

Allowing ϵ-approximate contacts means that we can reduce the precision of all the disc radii. If we start from an exact solution and round up each disc radius to the next larger integer power of $\sqrt{1+\epsilon}$, we still have an ϵ-approximate solution, giving the following lemma.

Lemma 7 *If there exists an exact solution to an instance of the CCG$^+$ problem, then there exists a solution to the corresponding instance of the ϵ-approximate CCG$^+$ problem in which every disc radius is an integer power of $\sqrt{1+\epsilon}$.*

Therefore we can search for a solution with the radii limited to powers of $\sqrt{1+\epsilon}$, and still be assured of finding an ϵ-approximate solution if an exact solution exists. We can prove the following approximate result.

Theorem 8 *There exists an algorithm such that given a circular CCG$^+$ instance for which an exact solution exists, it finds an ϵ-approximate solution. If no exact solution exists, it may produce an ϵ-approximate solution or fail. Where R is the greatest ratio, for any disc in the problem, between the maximum and minimum radii of Lemma 6, the algorithm runs (unconditionally on solution existence) in time polynomial to the instance size and to $\log R/\log(1+\epsilon)$.*

Note that this is a one-sided test: when an exact solution exists, our algorithm guarantees to find an approximate solution, but if there exists an approximate solution and no exact solution, the algorithm does not guarantee finding the approximate solution.

The approximation algorithm performs dynamic programming on intervals of the left-to-right sequence of discs, using the following observation: if we have three discs left to right and we know that the one in the middle is the largest (possibly tying with either of the other two), then the one on the left and the one on the right cannot touch each other. Thus if we know which disc is largest in the entire problem and its radius, then we can split the problem into two smaller ones whose solutions are independent, which is the necessary structure for dynamic programming. We can try all possibilities for the largest disc, and recurse on each side, memoizing the answers to the recursive subproblems.

Each recursive subproblem corresponds to an interval of the sequence of discs, with specified sizes for the discs on either end, an assumption that no disc in between is larger than either of those, and a choice between a small constant number of cases for whether this subproblem is the leftmost or rightmost in the entire problem and whether or not the two endmost discs are already connected by larger discs outside the subproblem. Lemma 7's limitation on the number of radii we need to consider forces the number of subproblems, and thus the algorithm's time complexity, to be polynomial.

4 Conclusion

In this paper, we considered a circular cover contact graph problem defined by Atienza et al. [4]. We showed that when the input discs and the covering discs are all constrained to touch a line, then the problem of deciding whether the input set has a connected CCG is $\mathcal{NP}$-hard.

We also defined an approximate variation of the problem, where the covering discs are allowed to overlap by a small amount. We gave a polynomial-time algorithm such that if there exists an exact solution to the problem, then the algorithm returns an ϵ-approximate solution. However, if there is no exact solution, then the algorithm does not guarantee to return an approximate solution that might exist. Our algorithm provides an approximate answer to the decision problem of exact solution existence. The decision problem of approximate solution existence is a different problem, and the complexity of that problem remains open.

References

[1] M. Abellanas, S. Bereg, F. Hurtado, A. García Olaverri, D. Rappaport, and J. Tejel. Moving coins. *Computational Geometry: Theory and Applications (CGTA)*, 34:35–48, 2006.

[2] M. Abellanas, N. de Castro, G. Hernández, A. Máarquez, and C. Moreno-Jiménez. Gear system graphs. Manuscript, 2006.

[3] M. Abellanas and C. Moreno-Jiménez. Geometric graphs realization as coin graphs. In *International Conference on Computational Science and Its Applications*, volume 3045, pages 1–10, Assisi, Italy, 2004. Springer.

[4] N. Atienza, N. de Castro, C. Cortés, M. A. 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. In S.-H. Hong, T. Nishizeki, and W. Quan, editors, *Graph Drawing*, volume 4875 of *Lecture Notes in Computer Science*, pages 171–182. Springer, 2007.

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

[6] P. Koebe. Kontaktprobleme der konformen abbildung. *Ber. Sachs. Akad. Wiss. Leipzig Math.-Phys. Kl.*, 88:141–164, 1936.

[7] J.-M. Robert and G. T. Toussaint. Computational geometry and facility location. In *International Conference on Operations Research and Management Sciences*, pages B–1–B–19, Manila, Philippines, 1990.