Embedding Ray Intersection Graphs and Global Curve Simplification

Mees van de Kerkhof[1](✉), Irina Kostitsyna[2], and Maarten Löffler[1]

[1] Department of Computing and Information Sciences, Utrecht University, Utrecht, The Netherlands
{m.vandekerkhof,m.loffler}@uu.nl

[2] Department of Mathematics and Computer Science, TU Eindhoven, Eindhoven, The Netherlands
i.kostitsyna@tue.nl

Abstract. We prove that circle graphs (intersection graphs of circle chords) can be embedded as intersection graphs of rays in the plane with polynomial-size bit complexity.

We use this embedding to show that the global curve simplification problem for the directed Hausdorff distance is NP-hard. In this problem, we are given a polygonal curve P and the goal is to find a second polygonal curve P' such that the directed Hausdorff distance from P' to P is at most a given constant, and the complexity of P' is as small as possible.

1 Introduction

Problems in the area of graph drawing often find application in complexity theory by providing a basis for NP-hardness proofs for geometric problems. In this paper, we study an application of embedding *circle graphs* (intersection graphs of chords of a circle) as *ray graphs* (intersection graphs of half-lines) to the analysis of the complexity of *global curve simplification*. In particular, we prove (refer to Sect. 2 for precise problem definitions):

- All circle graphs are ray graphs that have a representation as a set of intersecting rays described by coordinates that have a polynomial number of bits (Theorem 1);
- Hamiltonian Path is NP-hard on such ray graphs (Corollary 1);
- Directed Curve Simplification is NP-hard (Theorem 4).

1.1 Global Curve Simplification

Curve simplification is a long-studied problem in computational geometry and has applications in many related disciplines, such as graphics, and geographical information science (GIS). Given a polygonal curve P with n vertices, the goal is to find another polygonal curve P' with a smaller number of vertices such that P' is sufficiently similar to P. Methods proposed for this problem famously

H. C. Purchase and I. Rutter (Eds.): GD 2021, LNCS 12868, pp. 358–371, 2021.
https://doi.org/10.1007/978-3-030-92931-2_26

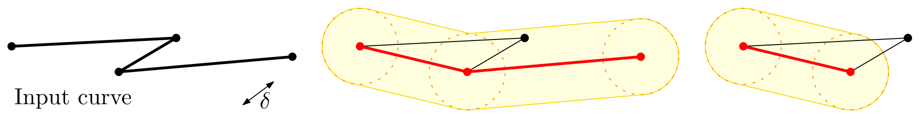

Fig. 1. For a target Hausdorff distance δ, the boldened red curve (middle) is a global simplification of the input curve (left), but it is not a local simplification, since the first shortcut does not closely represent its corresponding curve section (right). (Color figure online)

include a simple heuristic scheme by Douglas and Peucker [12], and a more involved classical algorithm by Imai and Iri [18]; both are frequently implemented and cited. Since then, numerous further results on curve simplification, often in specific settings or under additional constraints, have been obtained [1–5,9,10, 14,17].

Recently, the distinction was made between *global* simplification, when a bound on a distance measure must be satisfied between P and P', and *local* simplification when a bound on a distance measure must be satisfied between each edge of P' and its corresponding section of P [20]. A local simplification is also a global simplification, but the reverse is not necessarily true, see Fig. 1.

Agarwal *et al.* [2] were first to consider the idea of global simplification under the *Fréchet distance*. They introduce what they call a *weak simplification*: a model in which the vertices of the simplification are not restricted to be a subset of the input vertices, but can lie anywhere in the ambient space. Kostitsyna *et al.* [22] present a polynomial-time algorithm for this model but for the *Hausdorff distance*; in particular, the directed Hausdorff distance from the simplification curve to input curve. Van Kreveld *et al.* [24] consider a different setting in which the output vertices should be a subsequence of the input, and they also consider the Hausdorff distance. They give a polynomial-time algorithm for the directed Hausdorff distance from the simplification curve to input curve, but they show the problem is NP-hard for the directed Hausdorff distance in the opposite direction, and also for the symmetric (undirected) Hausdorff distance. Van de Kerkhof *et al.* [20] prove that the hardness result for the unrestricted Hausdorff distance can be extended to the non-restricted case as well; in addition, they introduce an intermediate *curve-restricted* model where the vertices of the simplified curve should lie on the input curve. Surprisingly, the problem is hard under this model for all three variants of the Hausdorff distance. Table 1 summarizes the state of the art for global curve simplification under the Hausdorff distance.

1.2 Embeddings of Geometric Intersection Graphs

Geometric intersection graphs have long been studied due to their wide range of applications, and lie on the interface between computational geometry, graph theory, and graph drawing [26]. The graph classes corresponding to intersections

Table 1. Results for global curve simplification under the Hausdorff distance between the curve P and its simplification P'. The result in **bold** is from this work.

Distance	Vertex-restricted ($\mathcal{V}$)	Curve-restricted ($\mathcal{C}$)	Non-restricted ($\mathcal{N}$)
$\overrightarrow{\mathsf{H}}(P, P')$	NP-hard [24]	NP-hard [20]	**NP-hard**
$\overrightarrow{\mathsf{H}}(P', P)$	$O(n^4)$ [24] $O(n^2 \text{polylog } n)$ [20]	NP-hard [20]	poly(n) [22]
$\mathsf{H}(P, P')$	NP-hard [24]	NP-hard [20]	NP-hard [20]

of geometric shapes form a natural hierarchy that links to the complexity of those shapes: more complex shapes allow to represent more graphs. Arguably the most restricted class in this family are the *unit interval graphs* [19], and the most general class of intersection graphs of connected shapes in $\mathbb{R}^2$ are the *string graphs* [13]. Between these two, a hierarchy of classes exist; part of it is illustrated in Fig. 2.

Hartmann *et al.* [16] introduce *grid intersection graphs* where the shapes are aligned to an orthogonal grid; Mustata [27] gives an overview of the state of the art and also discusses the complexity of computational problems on such classes. Cardinal *et al.* [7] prove several relations between segment intersection graphs and ray intersection graphs; in particular they introduce *downward ray graphs*: intersection graphs of rays that all point into a common half-plane. Their main result is that *recognition* of several classes is complete for the existential theory of the reals. *Circle graphs* are intersection graphs of chords of a circle; equivalently, they may be defined as interval graphs where there is an edge between two intervals on a line when they intersect but are not nested [15]. Circle graphs are known to be contained in 1-string graphs [8]. We are not aware of any published statements of stricter containment; we show in this paper that they are in fact contained in the downward ray graphs.

When utilizing graph embedding algorithms in hardness proofs, one important issue is the representation of the embedding. The class of graphs which can be represented as intersection graphs of a given set of shapes is not necessarily the same as the class of graphs which can be represented as intersection graphs of shapes which each can be represented with coordinates of bounded complexity. For instance, McDiarmid and Müller [25] show that not all realizable unit disk graphs can be realized with coordinates of logarithmic complexity, and the same is true for *segment* graphs [23].

2 Preliminaries, Overview and Challenges

2.1 Polygonal Curves and the Hausdorff Distance

A *polygonal curve* (also called a *polyline*) $P = \{p_1, p_2, \ldots, p_n\}$ is defined by an ordered sequence of n vertices. We can treat P as a continuous map $P : [1, n] \to \mathbb{R}^d$ that maps real values in the interval $[1 \ldots n]$ to points on the polyline by linearly interpolating between the vertices, which allows us to visualize a polyline as $n - 1$ line segments linked one after the other. We will refer to these segments as the polyline's *links*. For integer i, $P(i)$ will return the vertex p_i. Points on P's i'th link are parametrized as $P(i + \lambda) = (1 - \lambda)p_i + \lambda p_{i+1}$. The *directed Hausdorff distance* from curve P to curve Q, which have n and m vertices respectively, is given by $\overrightarrow{H}(P, Q) = \max\limits_{i \in [1 \ldots n]} \min\limits_{j \in [1 \ldots m]} ||P(i) - Q(j)||$. I.e. it is equal to the Euclidean distance from the point on P furthest from Q to the point on Q closest to that point. The *(undirected) Hausdorff distance* is the maximum over both directions, i.e. $H(P, Q) = \max\{\overrightarrow{H}(P, Q), \overrightarrow{H}(Q, P)\}$.

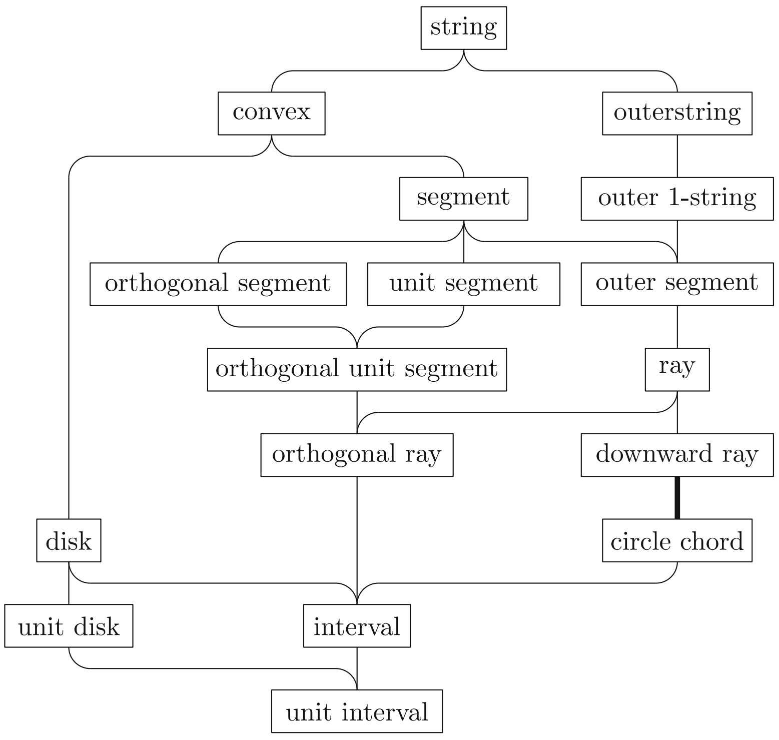

Fig. 2. Intersection graph classes and their inclusion relations. The thickened edge indicates our contribution. In order to keep the figure readable many classes and refinements have been omitted; for an extensive overview we refer the reader to e.g. [6,7,27] or to graphclasses.org.

2.2 Problem

The problem we wish to tackle (and which we will prove NP-hard) is:

Problem 1. Directed Curve Simplification. Given a polyline P, integer k and a value δ, find another polyline P' such that the directed Hausdorff distance from P to P' is at most δ and the number of links in P' is at most k.

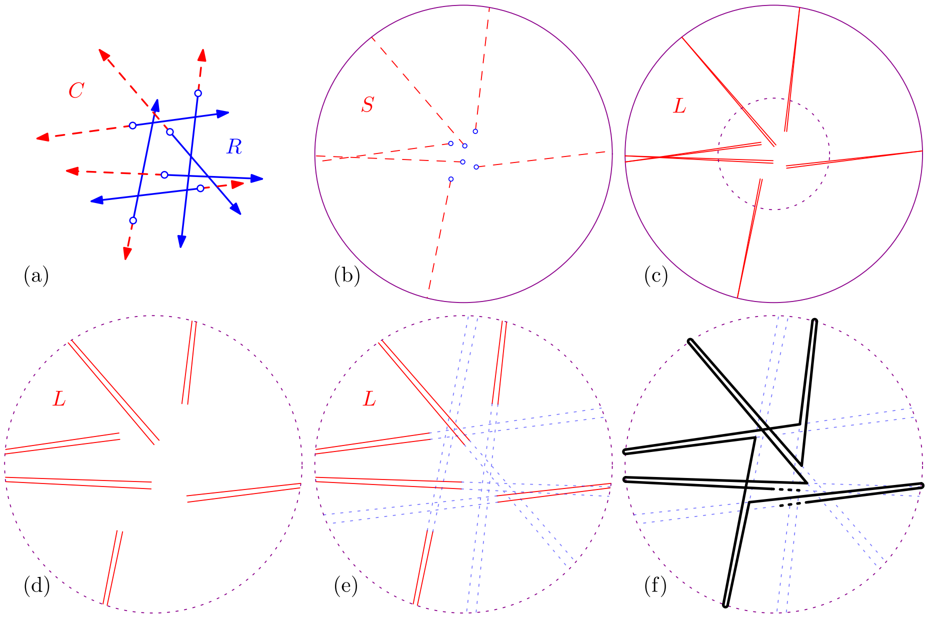

Fig. 3. The idea for a small example (which does not admit a Hamiltonian cycle). (a) A set of rays R (blue) whose intersection graph is G, and the complement C (red, dashed). (b) Zooming out until we can draw a circle that contains all intersections among rays in C. (c) Replacing each ray in C by a *needle*. (d) Zooming back in. (e) The extensions of the needles (blue, dotted) correspond to the original rays. (f) A polygon covering all needles must correspond to a Hamiltonian cycle in G (here, there is no solution). (Color figure online)

Note that van de Kerkhof *et al.* [20] call this problem the *non-restricted global curve simplification problem* to distinguish it from other variants; in the remainder of the present paper we use the shorter name for convenience. We will find that the key difficulty in solving Directed Curve Simplification lies in the following similar problem:

Problem 2. Segment Polyline Cover. Given a set L of line segments in the plane and integer k, find a polyline P such that every segment in L is *covered* by P (contained in at least one segment of P), and P has at most k links.

2.3 Proof Idea

Our approach is to show that Segment Polyline Cover is hard by a reduction from Hamiltonian Path on ray intersection graphs. Specifically, we use the following idea.

Observation 1. *Let G be a ray intersection graph with n vertices. There exists a set L of $2n$ segments such that G has a Hamiltonian cycle if and only if there is a polygon covering L with $2n$ vertices.*

We can use Observation 1 to prove Segment Polyline Cover is NP-hard and then reduce Segment Polyline Cover to Directed Curve Simplification, proving it NP-hard as well.

We sketch the proof of Observation 1 here; the rest of the paper is devoted to making it precise. The high level proof idea is illustrated in Fig. 3. Let R be a set of rays in $\mathbb{R}^2$, and G its intersection graph. The *complement* of a ray r is the ray with the same origin and the same supporting line as r which points in the opposite direction. Let C be the complement of R. We cut the rays in C to a set of segments S in such a way that C and S have the same intersection graph. Then we replace each segment $s \in S$ by a *needle*: a pair of segments both very close to s that share one endpoint (different from the corresponding ray's origin). Let L be the resulting set of $2n$ segments. Now, any polygon with $2n$ segments covering L must use the two edges of one needle consecutively (since, by construction, the extension of these segments does not intersect the supporting line of any other segment), and it can connect an edge from one needle to an edge of another needle exactly when the corresponding original rays in R intersect.

2.4 Challenges

Though the idea is conceptually simple, there are several difficulties in turning Observation 1 into a proof that Directed Curve Simplification is NP-hard.

- The simple idea above is phrased in terms of a Hamiltonian Cycle and covering segments by a polygon; for our proof we need to use a polyline. We need to be careful in how to handle the endpoints.
- We need to establish that Hamiltonian Path is indeed NP-hard on ray intersection graphs.
- We need to know how to embed a ray intersection graph as an actual set of rays with limited bit complexity.
- We need to model the input to Directed Curve Simplification as an instance of Segment Polyline Cover. Specifically, the complement of a set of rays is not necessarily connected; but the input to Directed Curve Simplification must be connected.
- The Segment Polyline Cover problem closely resembles Directed Curve Simplification for $\delta = 0$; to extend it to the case $\delta > 0$ we (again) need to carefully consider the complexity of the embedding.

Most of these challenges can be overcome, as we show in the remainder of this paper. However, since the problem of recognizing if a graph can be embedded as a set of intersecting rays is complete for the existential theory of the reals [7], we know that there are ray intersection graphs that cannot be embedded by a set of rays with subexponential bit complexity, unless $\mathbf{NP} = \exists\mathbb{R}$. In this paper, we work around this problem by considering a smaller class of graphs, and allowing a superpolynomial grid for our embeddings, which we show is sufficient for the proof of Theorem 4.

3 Hamiltonian Cycles in Ray Intersection Graphs

3.1 Embedding Circle Graphs as Ray Graphs

We will show that each circle graph can be embedded as a ray intersection graph. To show this, we construct a set of n points that lie on a convex, increasing curve such that all chords connecting a pair of points can be extended to a ray to the right, and none of these rays will intersect below the curve. This requires the curve to grow very fast. We use the points $(x, x!)$ for $x \in [1..n]$, where $x! = \Pi_{i=1}^{x} i$ is the factorial function. Indeed, these points have the following property.

Lemma 1. *Let $a, b, c, d \in [1..n]$ be four numbers such that $a < b$ and $c < d$. Let A be the ray starting at $(a, a!)$ and containing $(b, b!)$, and let B be the ray starting at $(b, b!)$ such that $B \subset A$. Similarly, let C be the ray starting at $(c, c!)$ and containing $(d, d!)$, and let D be the ray starting at $(d, d!)$ such that $D \subset C$. Then B and D do not intersect; hence, A and C intersect if and only if $A \backslash B$ and $C \backslash D$ intersect.*

Proof. Since every ray is drawn between two points on the curve of the function $x!$, we know that it intersects this curve only at these points. The distance between y-coordinates of successive points keeps rapidly increasing as x increases, but the distance between x-coordinates of successive points is constant. Thus the slope of a ray r_1 whose intersection points with the curve lie to the right of those of ray r_2 will be greater than the slope of r_2. Without loss of generality we assume $a < c$. There are three possible cases, see Fig. 4:

- $c < b < d$: Here it is clear that A and C will intersect at an x-coordinate somewhere between c and b, and so B and D will not intersect.
- $b < c$: Here we can easily see that B and D do not intersect, as B starts below D and has a lower slope.
- $d < b$: Whereas the first two cases only require the curve to be convex and increasing, this case also requires the function to grow quick enough: Since D starts to the left of B it could possibly intersect B if its slope was higher. We will now show, however, that the factorial function grows quick enough so that this cannot happen. For a fixed b, the lowest slope that B can have is when $a = 1$. The highest slope that D can have occurs when $c = b - 2$ and $d = b - 1$. The slope of B is equal to the slope of A, which would be $\frac{b!-1!}{b-1} = b \cdot (b-2)! - \frac{1}{b-1}$. The slope of D (and C) in this scenario would be $\frac{(b-1)!-(b-2)!}{1} = (b-2) \cdot (b-2)!$. We can see that the slope of B is higher than D if $2 \cdot (b-2)! \geq \frac{1}{b-1}$ which obviously holds for all $b > 2$. So since B starts above D and has higher slope, B and D will not intersect.

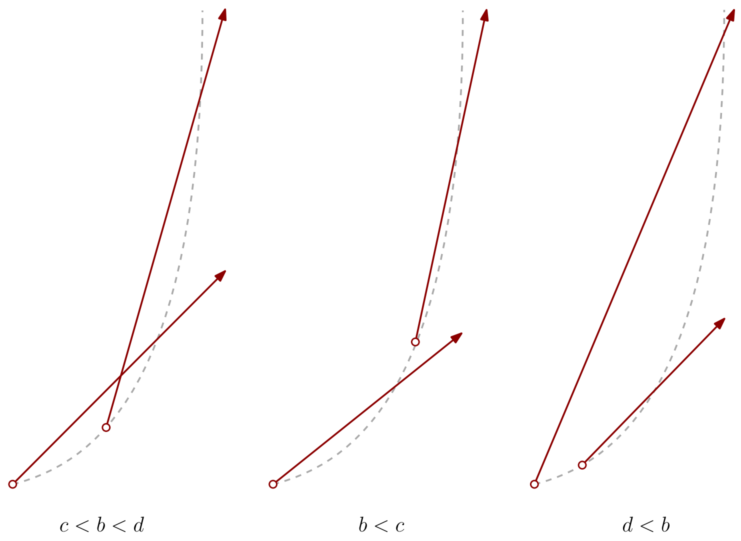

Fig. 4. The three cases for two rays. a and b are the x-coordinates of the points where the first ray intersects with the curve $y = x!$, and c and d are those values for the second ray. No matter the case, the rays will not intersect below the curve.

Once we have constructed these points we can "unroll" any circle graph by picking one chord endpoint on the circle to be the first point and then traversing the circle in clockwise order and assigning each chord endpoint we encounter the next point of our set. See Fig. 5 for a sketch. Because the y-coordinate for a point will not grow bigger than $O(n^n)$ we can represent the points using polynomial bit complexity. At this point, we have shown that circle graphs are contained in ray graphs. In fact, our construction gives a bit more:

Theorem 1. *The class of circle graphs is contained in the class of ray intersection graphs. Furthermore, every circle graph can be embedded as the intersection graph of a set of rays such that:*

- *every ray is grounded on a common curve* (grounded ray graph [7])*;*
- *every ray points towards the upper right quadrant* (downward ray graph [7])*;*
- *every ray is described by a point and a vector with polynomial bit complexity.*

3.2 Hamiltonian Paths and Cycles

Next, we show that Hamiltonian Path problem is NP-hard on ray graphs, and in particular, on ray graphs with polynomial bit complexity.

We reduce from the Hamiltonian Path problem on circle graphs. We make use of the proof from Damaschke [11]. He shows that Hamiltonian cycle is NP-hard on circle graphs, by reducing from Hamiltonian Cycle in cubic bipartite graphs. He also claims that there is an easy adaptation that shows the Hamiltonian path problem is also NP-hard for circle graphs. We will start by making

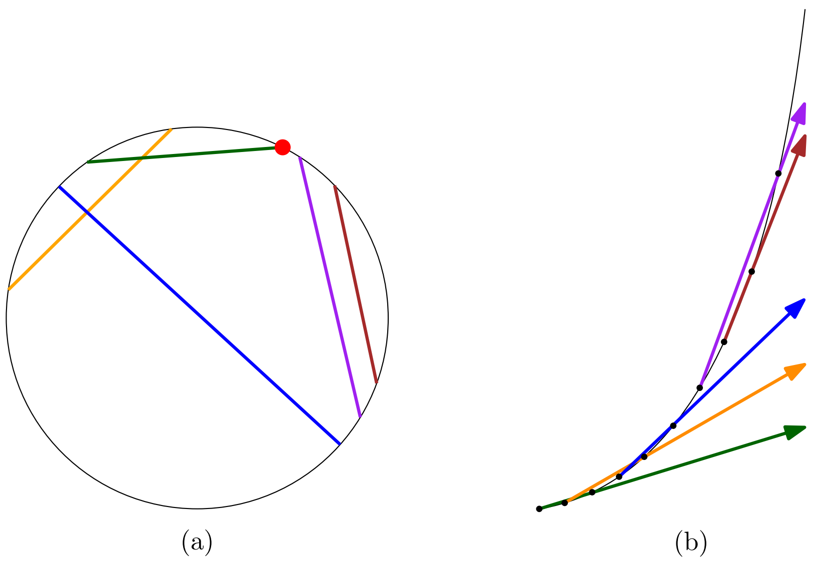

Fig. 5. (a) A circle graph with colors assigned to the chords. The chosen starting point is marked with a red dot. (b) Unrolled version of (a), by assigning chord endpoints in counterclockwise order to points on the convex curve they can be extended into rays without intersecting. (Color figure online)

this adaptation explicit: We construct an instance of the circle graph problem as described in [11], but then we replace one of the X-chords with two parallel chords close to where the X-chord was, so that they both intersect the same chords that were intersected by the X-chord. For both of the new chords we then add one new chord that only intersects that chord and no others. Now we know that the circle graph will have a Hamiltonian path if and only if the bipartite graph has a Hamiltonian cycle. From Theorem 1 we now immediately have:

Corollary 1. *Hamiltonian Path is NP-hard on intersection graphs of rays that have a polynomial bit complexity.*

4 Connected Segment Polyline Cover

Next, we introduce the Connected Segment Polyline Cover problem, and show that it is NP-hard by a reduction from Hamiltonian Path problem on circle graphs through the construction outlined above.

Problem 3. Connected Segment Polyline Cover. Given a set L of n line segments whose union is connected, and an integer k, decide if there exists a polyline of k links that fully covers all segments in L.

We start by embedding the circle graph as a ray intersection graph in the manner outlined above. Then, we compute all intersection points between supporting lines of the rays. One of these intersection points will have the lowest y-coordinate. We will then choose a value that is lower than this lowest y-coordinate, which we will denote as y_ℓ. For each ray r, let p_r be its starting point. Let $\bar{r}$ be r's complement: the part of the supporting line that is not covered by r. Let $\tilde{r}$ be the part of $\bar{r}$ that has $y \geq y_\ell$. Now we construct a *needle* for each ray's complement: Two line segments that share one endpoint at the point where $\bar{r}$ has y-coordinate y_ℓ. The other endpoint for both segments lies very close to p_r. The endpoints are on opposite sides of the ray starting point so we get a wedge-like shape that runs nearly parallel to $\tilde{r}$. In addition to these $2n$ segments, which we will refer to as *needle segments*, we create three more segments which we will refer to as the *leading segments*: We create one horizontal segment we call s_h with y-coordinate between y_ℓ and the lowest intersection point between ray supporting lines, that starts far to the right of the needle segments and ends to the left of them, intersecting all of the needles. Attached to s_h is a large vertical segment we call s_v, running up to a point above the highest starting point of a ray. Attached to that is another horizontal segment we call s_t, this one being short and ending to the left of any ray starting point. See Fig. 6 for a sketch.

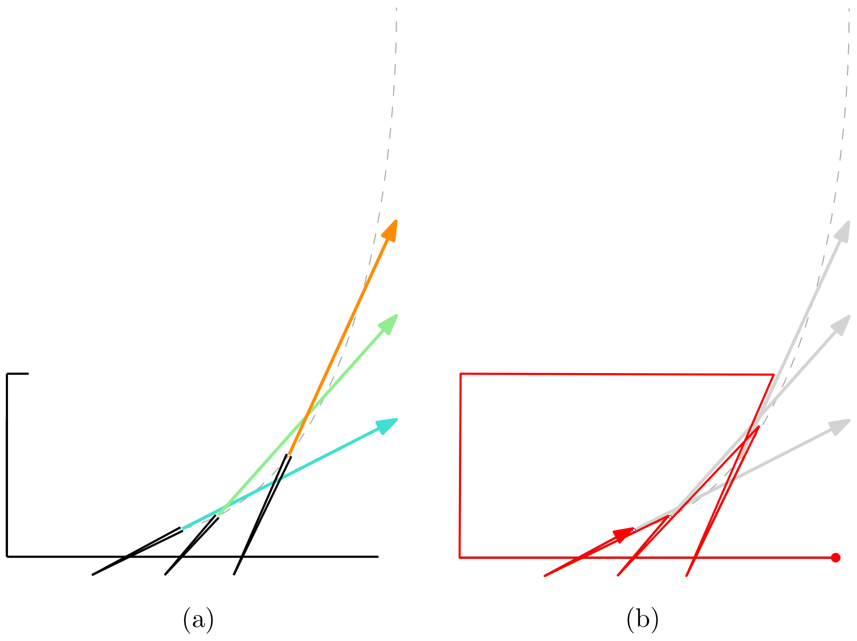

Fig. 6. (a) Sketch of a reduction of a circle graph with three chords. Segments shown in black. (b) Polyline of $2n + 3$ links covering the constructed segments, corresponding to a Hamiltonian Path traversing the rays, starting with the ray with the largest slope and ending with the ray with the smallest slope.

Now we have $2n + 3$ segments in total, where n is the number of chords in the original graph.

Lemma 2. *We can cover all segments using a polyline of* $2n + 3$ *links if and only if the circle graph has a Hamiltonian Path.*

Proof. To see why this is true, consider that since none of the segments are collinear and no three segments intersect in the same point, a suitable polyline must fully cover one segment with each link. For a polyline to be able to bend from fully covering one segment to fully covering another, either the segments must have a shared endpoint, or the supporting lines of the segments must intersect in a point not contained in either segment. Segment s_h intersects all needle segments in their interior and is parallel to s_t, so we know that a suitable polyline must start[1] by covering s_h, and it must bend at the common endpoint with s_v and then fully cover s_v. All of the intersection points between s_v and the supporting lines of needle segments lie below the endpoint it shares with s_h, so to be able to cover s_v with the second link the polyline must next connect to s_t, meaning it bends at the shared endpoint of s_v and s_t. The third link is horizontal, covering s_t. Since the supporting line of s_t intersects all of the rays, the polyline can bend to any needle segment for its next link.

Since we have covered our additional segments s_h, s_v, and s_t, the rest of the $2n$ links must cover one needle segment each. Observe that the needle segments all extend downward to below the lowest intersection point between supporting lines. This means that when a link covers a needle segment when travelling downward, the next link must then travel upward on the other half of the needle, as all intersection points with supporting lines of other segments lie in the segment's interior. When the next link then covers a needle segment when travelling upward, the only places the polyline can viably bend next are near places where the ray associated with the previous needle intersects another ray. So we can cover the $2n$ needle segments using a polyline of $2n$ links if and only if there is a Hamiltonian Path in the ray intersection graph and thus a Hamiltonian Path in the original circle graph.

Since the transformation is polynomial, we know the problem is NP-hard. We can also see that the problem is in NP, since for any instance we can expect that if a polyline of k links exists covering a set L, one must also exist where each vertex has coordinates of polynomial complexity, since the vertices could all lie on the intersection points of the supporting lines of the segments, or otherwise on points with rational coordinates on those supporting lines. This polyline could serve as a certificate for the verification algorithm. This gives the following theorem:

Theorem 2. Connected Segment Polyline Cover *is NP-complete.*

[1] A suitable polyline could also end with s_h, but we will define the polyline to be in this direction for ease of notation.

5 Directed Curve Simplification

Finally, we reduce Connected Segment Polyline Cover to Directed Curve Simplification. As a problem instance, we are given a set L of n non-collinear line segments in the plane whose union is connected. We construct an input polyline of polynomial size that completely covers the set of segments and no other points. We could do this, for example, by treating the segment end-points and intersection points as vertices of a graph connected by edges, and have our polyline be the path of a breadth-first search through the graph. We set δ to 0. Now we know that, since a simplification must cover the union of L, any simplification of our input polyline that has n links must cover each segment in L completely with one link. This means such a simplification would be a solution to our instance of Connected Segment Polyline Cover. Since the reduction is polynomial in size, we know that this variant of the GCS problem is NP-hard, and using a similar argument to the one for Connected Segment Polyline Cover it is easy to see that it is in NP as well.

Theorem 3. Directed Curve Simplification, *restricted to instances where* $\delta = 0$, *is NP-complete.*

5.1 Non-zero δ

We can also extend this reduction to non-zero δ by picking $\delta > 0$ but still small enough such that it would not change the combinatorial structure of the space the polyline can lie in, so each link of the polyline must still correspond to exactly one segment in L. For the segments we have constructed in the earlier reduction, we will show that setting $\delta < \frac{3}{4n!}$ will guarantee the structure of the space will not change. So a simplified polyline of $2n + 3$ links with $0 < \delta < \frac{3}{4n!}$ will only exist if and only if it also exists for $\delta = 0$.

For space reasons, we only sketch the ideas of the proof here; details can be found in the full version [21].

The main idea is to choose δ sufficiently small so that there are no additional intersections between extensions of segments that are not supposed to intersect. When $\delta = 0$, the space the simplified polyline can occupy is exactly the input segments, but for non-zero δ, the polyline does not have to exactly cover the original segment. If we center two circles with radius δ on the endpoints of a segment, the two inner tangents of these circles will form the bounding lines of a cone that covers all possible polyline links that are able to "cover" a segment. We will call the part of the cone that is within δ of the segment the *tip* of the cone, and the rest of the cone the *tail* of the cone. For $\delta = 0$, the supporting line for the segment forms a degenerate cone of width 0. To preserve the combinatorial structure, fattening the cones cannot introduce intersections between cone tails, as these correspond to two segments' supporting lines intersecting in the exterior of the segments. To simplify the algebra, we consider a slightly larger cone, between the lines connecting points created by going 2δ to the left and right of the original endpoints of the needle segments. It is easy to see that these larger cones contain the true cones. We reach the bound on δ given above by case distinction of different configurations of potentially intersecting cones, and taking the minimum.

Since we can have a small enough δ of polynomial bit complexity, this means the Directed Curve Simplifiction problem is NP-hard in general, as for larger values of δ the construction could be scaled up.

If the general problem is in NP is hard to say, since our approach for showing this for the previous problems does not extend, and it might be possible that inputs exist where the only possible simplifications of k links have vertex coordinates of exponential bit complexity. This remains an open problem.

Theorem 4. Directed Curve Simplification *is NP-hard.*

6 Conclusion

We have shown that Directed Curve Simplification is NP-hard, which completes the results in Table 1 and completely settles the complexity of global curve simplification under the Hausdorff distance.

As the main tool in our reduction, we have shown that every circle graph can be embedded as an intersection graph of rays with coordinates of polynomial complexity. It is still an open question if it is possible to embed every circle graph as rays with coordinates of logarithmic complexity. Whether Directed Curve Simplification is in NP is another open problem that remains.

Acknowledgements. The authors would like to thank Birgit Vogtenhuber, Tillman Miltzow, and anonymous reviewers for valuable discussions and suggestions. Mees van de Kerkhof and Maarten Löffler are (partially) supported by the Dutch Research Council under grant 628.011.00. Maarten Löffler is supported by the Dutch Research Council under grant 614.001.50.

References

1. Abam, M.A., de Berg, M., Hachenberger, P., Zarei, A.: Streaming algorithms for line simplification. Discret. Comput. Geom. **43**(3), 497–515 (2010)
2. Agarwal, P.K., Har-Peled, S., Mustafa, N., Wang, Y.: Near linear time approximation algorithm for curve simplification. Algorithmica **42**(3–4), 203–219 (2005)
3. Barequet, G., Chen, D.Z., Daescu, O., Goodrich, M.T., Snoeyink, J.: Efficiently approximating polygonal paths in three and higher dimensions. Algorithmica **33**(2), 150–167 (2002)
4. de Berg, M., van Kreveld, M., Schirra, S.: Topologically correct subdivision simplification using the bandwidth criterion. Cartogr. Geogr. Inf. Syst. **25**(4), 243–257 (1998)
5. Buzer, L.: Optimal simplification of polygonal chain for rendering. In: Proceedings of 23rd Annual ACM Symposium on Computational Geometry (SoCG), pp. 168–174 (2007)
6. Cabello, S., Jejčič, M.: Refining the hierarchies of classes of geometric intersection graphs. Electron. J. Combin. **24**(1), 1–33 (2017)

7. Cardinal, J., Felsner, S., Miltzow, T., Tompkins, C., Vogtenhuber, B.: Intersection graphs of rays and grounded segments. J. Graph Algorithms Appl. **22**(2), 273–295 (2018)
8. Chalopin, J., Gonçalves, D., Ochem, P.: Planar graphs are in 1-string. In: Proceedings of the Eighteenth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2007, pp. 609–617 (2007)
9. Chan, S., Chin, F.: Approximation of polygonal curves with minimum number of line segments or minimum error. Int. J. Comput. Geom. Appl. **6**(1), 59–77 (1996)
10. Chen, D.Z., Daescu, O., Hershberger, J., Kogge, P.M., Mi, N., Snoeyink, J.: Polygonal path simplification with angle constraints. Comput. Geom. **32**(3), 173–187 (2005)
11. Damaschke, P.: The Hamiltonian circuit problem for circle graphs is NP-complete. Inf. Process. Lett. **32**(1), 1–2 (1989)
12. Douglas, D.H., Peucker, T.K.: Algorithms for the reduction of the number of points required to represent a digitized line or its caricature. Cartographica **10**(2), 112–122 (1973)
13. Ehrlich, G., Even, S., Tarjan, R.E.: Intersection graphs of curves in the plane. J. Comb. Theory. Ser. B **21**, 8–20 (1976)
14. Godau, M.: A natural metric for curves - computing the distance for polygonal chains and approximation algorithms. In: Proceedings of the 8th Annual Symposium on Theoretical Aspects of Computer Science (STACS), pp. 127–136 (1991)
15. Golumbic, M.C.: Algorithmic graph theory and perfect graphs (2004)
16. Hartman, I.A., Newman, I., Ziv, R.: On grid intersection graphs. Discret. Math. **87**, 41–52 (1991)
17. Imai, H., Iri, M.: An optimal algorithm for approximating a piecewise linear function. J. Inf. Process. **9**(3), 159–162 (1986)
18. Imai, H., Iri, M.: Polygonal approximations of a curve - formulations and algorithms. In: Computational Morphology: A Computational Geometric Approach to the Analysis of Form (1988)
19. Jinjiang, Y., Sanming, Z.: Optimal labelling of unit interval graphs. Appl. Math. **10**(3), 337–344 (1995)
20. van de Kerkhof, M., Kostitsyna, I., Löffler, M., Mirzanezhad, M., Wenk, C.: Global curve simplification. In: Proceedings of the 27th European Symposium on Algorithms (ESA) (2019)
21. van de Kerkhof, M., Kostitsyna, I., Löffler, M.: Embedding ray intersection graphs and global curve simplification (2021). https://arxiv.org/abs/2109.00042
22. Kostitsyna, I., Löffler, M., Polishchuk, V., Staals, F.: On the complexity of minimum-link path problems. J. Comput. Geom. **8**(2), 80–108 (2017)
23. Kratochvíl, J., Matoušek, J.: Intersection graphs of segments. J. Comb. Theory, Ser. B **62**(2), 289–315 (1994)
24. van Kreveld, M., Löffler, M., Wiratma, L.: On optimal polyline simplification using the Hausdorff and Fréchet distance. In: Proceedings of the 34th International Symposium on Computational Geometry (SoCG), vol. 56, pp. 1–14 (2018)
25. McDiarmid, C., Müller, T.: Integer realizations of disk and segment graphs. J. Comb. Theory, Ser. B **103**(1), 114–143 (2013)
26. McKee, T.A., McMorris, F.: Topics in Intersection Graph Theory. Society for Industrial and Applied Mathematics, USA (1999)
27. Mustata, I.: On subclasses of grid intersection graphs. Doctoral thesis, Technische Universität Berlin, Berlin (2014)