

An Improved Bound for Plane Covering Paths

Hugo A. Akitaya ✉️🏠^{id}

Miner School of Computer & Information Sciences, University of Massachusetts, Lowell, MA, USA

Greg Aloupis ✉️🏠

Khoury College of Computer Sciences, Northeastern University, Boston, MA, USA

Ahmad Biniaz ✉️🏠^{id}

School of Computer Science, University of Windsor, Canada

Prosenjit Bose ✉️🏠^{id}

School of Computer Science, Carleton University, Ottawa, Canada

Jean-Lou De Carufel ✉️🏠^{id}

School of Electrical Engineering and Computer Science, University of Ottawa, Canada

Cyril Gavoille ✉️🏠^{id}

LaBRI, University of Bordeaux, France

John Iacono 🏠^{id}

Department of Computer Science, Université libre de Bruxelles, Belgium

Linda Kleist ✉️🏠^{id}

Department of Computer Science, Universität Potsdam, Germany

Michiel Smid ✉️🏠^{id}

School of Computer Science, Carleton University, Ottawa, Canada

Diane Souvaine ✉️🏠^{id}

Department of Computer Science, Tufts University, Medford, MA, USA

Leonidas Theocharous ✉️🏠^{id}

School of Electrical Engineering and Computer Science, University of Ottawa, Canada

Abstract

A *covering path* for a finite set P of points in the plane is a polygonal path such that every point of P lies on a segment of the path. The vertices of the path need not be at points of P . A covering path is *plane* if its segments do not cross each other. Let $\pi(n)$ be the minimum number such that every set of n points in the plane admits a plane covering path with at most $\pi(n)$ segments. We prove that $\pi(n) \leq \lceil 6n/7 \rceil$. This improves the previous best-known upper bound of $\lceil 21n/22 \rceil$, due to Biniaz (SoCG 2023). Our proof is constructive and yields a simple $O(n \log n)$ -time algorithm for computing a plane covering path.

2012 ACM Subject Classification Theory of computation → Computational geometry

Keywords and phrases Covering Path, Upper Bound, Simple Algorithm

Digital Object Identifier 10.4230/LIPIcs.ESA.2025.75

Funding *Hugo A. Akitaya*: Supported by the NSF award CCF-2348067.

Ahmad Biniaz: Supported in part by NSERC.

Prosenjit Bose: Supported in part by NSERC.

Jean-Lou De Carufel: Supported in part by NSERC.

Cyril Gavoille: Supported by the French ANR projects ANR-22-CE48-0001 (TEMPOGRAL) and ANR-24-CE48-7768-01 (ENEDISC).

John Iacono: This work was supported by the Fonds de la Recherche Scientifique-FNRS.

Michiel Smid: Supported in part by NSERC.

Leonidas Theocharous: Supported in part by NSERC.

Acknowledgements This work was initiated at the 12th Annual Workshop on Geometry and Graphs, held at the Bellairs Research Institute in Holetown, Barbados, in February 2025. The authors thank the organizers and participants.



© Hugo A. Akitaya, Greg Aloupis, Ahmad Biniaz, Prosenjit Bose, Jean-Lou De Carufel, Cyril Gavoille, John Iacono, Linda Kleist, Michiel Smid, Diane Souvaine, and Leonidas Theocharous; licensed under Creative Commons License CC-BY 4.0

33rd Annual European Symposium on Algorithms (ESA 2025).

Editors: Anne Benoit, Haim Kaplan, Sebastian Wild, and Grzegorz Herman; Article No. 75; pp. 75:1–75:10

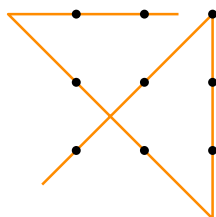


Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

1 Introduction

The problems of traversing points in the plane by polygonal paths are fundamental to discrete and computational geometry, and have been studied from both combinatorial and computational points of view. For instance one can refer to traversing a set of points by a non-crossing path [3, 14], or by a path that minimizes the total edge length [7, 28, 30, 31],¹ minimizes the longest edge length [10, 11], maximizes the shortest edge length [4, 5], minimizes the number of turns (or bends) [18, 32],² or minimizes the total or the largest turning angle [1, 2, 20].

In this paper we focus on traversing a set of points by a non-crossing polygonal path with a small number of segments. This problem has been studied before [12, 13, 18, 32], and is related to the classic nine-dot puzzle that asks to cover the vertices of a 3×3 grid by a polygonal path with no more than 4 segments [29], as in Figure 1.



■ **Figure 1** A non-plane covering path with four segments for the nine-dot puzzle.

Let P be a finite set of points in the plane. A *covering path* for P is a polygonal path (i.e., a path with straight-line segments) such that every point of P lies on some segment of the path. A vertex of a covering path can be any point in the plane (i.e., not necessarily a point in P). The segments of a covering path are also referred to as *links* [6] or *bends* [32] in the literature. A covering path with the smallest number of links is called a *minimum-link* or a *minimum-bend* covering path. A covering path is called *plane* or *non-crossing* if its segments do not cross each other. Figure 1 illustrates a covering path with 4 segments that is not plane.

In addition to their puzzling nature, dating back to 1914 [29], covering paths with a small number of segments are of interest in robotics and heavy machinery, where turning is an expensive operation [32]. Covering paths have also received considerable attention in recent years [6, 13, 18, 25]. Morić (in CCCG 2010) [17] and Welzl (in GWOP 2011) [17], and later Dumitrescu, Gerbner, Keszegh, and Tóth [18] raised challenging questions about covering paths, including the following.

► **Question 1.** *What is the minimum number $\pi(n)$ such that every set of n points in the plane can be covered by a non-crossing path with at most $\pi(n)$ straight-line segments?*

It is not hard to see that $n/2 \leq \pi(n) \leq n-1$, because any set of points can be covered by a plane path of $n-1$ segments (say by connecting the points from left to right), and each straight-line segment can cover at most two points (if no three points are collinear). Dumitrescu et al. [18] presented the first non-trivial lower and upper bounds:

$$\frac{5n}{9} - O(1) < \pi(n) < \left(1 - \frac{1}{601\,080\,391}\right)n.$$

¹ Known as the traveling salesperson path problem.

² Known as the minimum bend traveling salesperson problem.

The upper bound has been improved to $\lceil 21n/22 \rceil$ by Biniarz [12, 13]. The upper bounds are universal (i.e., any point set admits these bounds), while the lower bounds are existential (i.e., there exist point sets that achieve these bounds).

1.1 Our Contributions

We present an algorithm that constructs a plane covering path of at most $\lceil 6n/7 \rceil$ segments for any set of n points in the plane, which implies that $\pi(n) \leq \lceil 6n/7 \rceil$. The algorithm runs in $O(n \log n)$ time. This is optimal, because the problem of computing a plane covering path (even without the minimality constraint) has an $\Omega(n \log n)$ lower bound in the worst case in the algebraic decision tree model of computation – by a reduction from sorting [18].

Simplicity and differences to previous algorithms. Our algorithm shares similarities with the algorithms of Biniarz [12, 13] and Dumitrescu et al. [18] in terms of iteratively scanning the points from left to right. One limitation of the previous approaches is the use of *caps* and *cups* to save a segment in each iteration. A *k-cap* is a concave x -monotone chain of k points, and a *k-cup* is a convex x -monotone chain of k points. Dumitrescu et al. construct an 18-cap or 18-cup in each iteration of their algorithm [18], while Biniarz uses 5-caps or 5-cups [12, 13]. To ensure the existence of such caps or cups, at least 601080391 and 21 points are required, respectively, as implied by the Erdős-Szekeres theorem [19]. In contrast, our algorithm uses convex chains of points that are not necessarily x -monotone (hence neither a cap nor a cup).

Besides using a cap or a cup in each iteration, the previous algorithms [13, 18] partition the remaining scanned points (within the iteration) into certain subsets such that each subset is contained in a convex region, and the convex regions of all subsets are disjoint. The algorithms compute a covering path within each region, and then concatenate the paths and the cap or cup. The partitioning and the concatenation are nicely crafted so that the resulting path is plane. Our iterations are more straightforward and do not use complex partitioning – we consider three main cases based on the size of the convex hull of 5 points.

1.2 Related Problems and Results

If we allow crossings, one can obtain better upper bounds on the number of segments of covering paths. For example by repeatedly applying the Erdős-Szekeres theorem [19], one can obtain a covering path with at most $n/2 + O(n/\log n)$ segments [17, 18]. The optimization problem, which is to compute a minimum-link covering path for a set of points, is shown NP-hard by Arkin et al. [6]. The best known approximation algorithm, due to Stein and Wagner [32], has a ratio of $O(\log z)$ where z is the maximum number of collinear points.

A *covering tree* for a point set P is a tree with straight-line edges that cover every point of P . Covering trees are useful in red-blue separation [22], and in the construction of rainbow polygons [13, 21]. Let $\tau(n)$ be the minimum number such that every set of n points in the plane can be covered by a plane tree with at most $\tau(n)$ edges. It is known that $9n/17 - O(1) \leq \tau(n) \leq \lceil 4n/5 \rceil$ where the lower and upper bounds come from [18] and [13], respectively. The exact values of $\pi(n)$ and $\tau(n)$ for the vertices of the square grid have been determined by Keszegh [25].

Two other well-studied related problems to mention are covering points with minimum-link³ axis-aligned paths [8, 9, 16, 24] or with a minimum number of lines [15, 23, 26, 27].

³ For axis-aligned paths this is usually referred to as minimum-turn.

2 The Algorithm

In this section we prove the following theorem.

► **Theorem 1.** *Every set of n points in the plane admits a non-crossing covering path with at most $\lceil 6n/7 \rceil$ segments. Thus, $\pi(n) \leq \lceil 6n/7 \rceil$. Such a path can be constructed in $O(n \log n)$ time.*

Our proof is constructive. We present an algorithm that computes, for any set P of n points in the plane, a non-crossing covering path with at most $\lceil 6n/7 \rceil$ segments.

Preliminaries. We denote the convex hull of a point set S by $\text{CH}(S)$. For two points p and q we denote by $\ell(p, q)$ the line through p and q , by pq the line segment with endpoints p and q , and by \overrightarrow{pq} the ray that emanates from p and passes through q . The concatenation of two paths, δ_1 and δ_2 , that share an endpoint is denoted by $\delta_1 \oplus \delta_2$.

Our algorithm is iterative and scans the points from left to right. Note that via a suitable rotation we may assume that no two points of P have the same x -coordinate. In the first iteration we scan only one point. In each subsequent iteration except for the last, we scan 6 or 7 new points and cover them with 5 or 6 new segments, respectively. Thus the ratio of the number of new segments to the number of covered points in these *intermediate* iterations is at most $6/7$. In the last iteration we scan the (at most 6) remaining points.

We assume that, among the scanned points in each iteration, no three are collinear. Collinear points are typically easier to handle because one can cover three points by one segment – such cases can be handled by an argument similar to that of [13]. Let m represent the number of points that have been scanned so far and let l' represent the rightmost scanned point. We maintain the following invariant at the beginning of every iteration after the first.

Invariant. All m points that have been scanned so far are covered by a non-crossing path δ with at most $\lfloor 6m/7 \rfloor$ segments. The path δ lies to the left of the vertical line through l' . The degree of l' in δ is one if $m \geq 2$ and zero if $m = 1$.

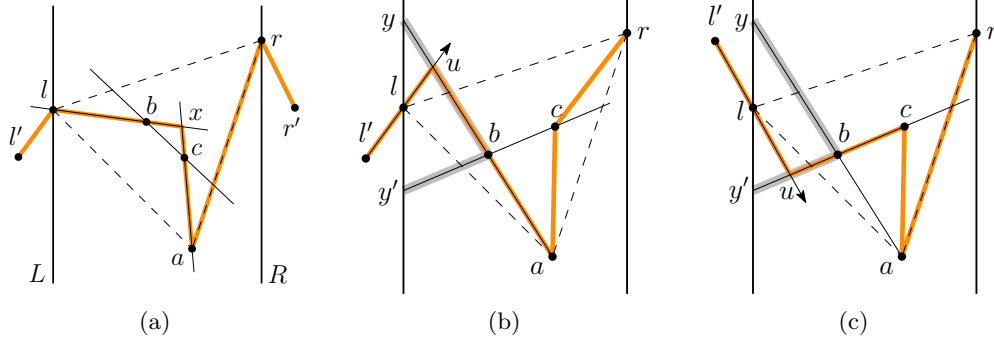
The invariant holds after the first iteration, trivially. We now focus on how to handle each intermediate iteration; this is the main part of our algorithm and proof. The last iteration is described at the end. Note that the invariant has the floor-function, whereas Theorem 1 has the ceiling-function. As we will see, at the beginning of each iteration, the m points scanned so far are covered by at most $\lfloor 6m/7 \rfloor$ segments. After the last iteration, we have $m = n$ and this upper bound becomes $\lceil 6n/7 \rceil$.

In each intermediate iteration the new points are appended to the current path δ in a suitable way, without altering the structure of δ . In our descriptions we will use Δ to denote the resulting path.

We start by scanning 6 new points. Denote these points by l, a, b, c, r, r' , where l is the leftmost, r' is the rightmost and r is the second rightmost point, as in Figure 2. Let $S = \{l, a, b, c, r\}$. Thus, S has all newly scanned points except for r' . Let L and R be the vertical lines through l and r , respectively. We refer to the region between L and R as *the slab*. Observe that l and r are both vertices of the boundary of $\text{CH}(S)$. We consider three cases depending on the number of vertices on $\text{CH}(S)$.

Case 1. $\text{CH}(S)$ has 3 vertices. Without loss of generality, assume that a is the third vertex of $\text{CH}(S)$, located below $\ell(l, r)$. Thus b and c are inside triangle $\triangle(a, l, r)$. The line $\ell(b, c)$ intersects exactly two edges of $\triangle(a, l, r)$. We consider three subcases.

- $\ell(b, c)$ intersects $\ell(l, r)$ and ar , as shown in Figure 2(a). The points l, b, c , and a form the vertices of a convex quadrilateral. After a suitable relabeling, we may assume that they appear in this order along the boundary of the quadrilateral. Then the intersection x of $\ell(a, c)$ and $\ell(l, b)$ lies in $\triangle(a, l, r)$. We form the path (l', l, x, a, r, r') and append it to δ . Thus we cover the six new vertices (l, b, c, a, r, r') with five segments. The savings come from avoiding using segment bc .



■ **Figure 2** $\text{CH}(S)$ has three vertices. (a) $\ell(b, c)$ intersects lr and ar . (b) $\ell(b, c)$ intersects al and ar , and $\vec{l'l}$ intersects by . (c) $\ell(b, c)$ intersects al and ar , and $\vec{l'l}$ intersects by' .

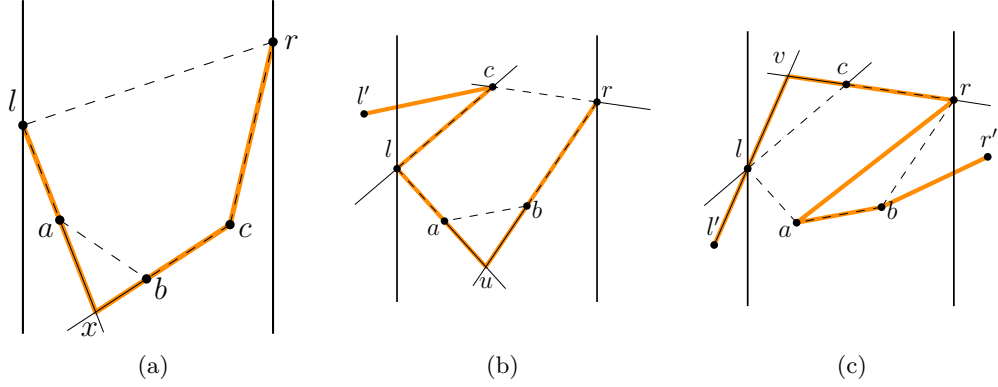
- $\ell(b, c)$ intersects lr and al . This is symmetric to the previous case. We let l' and l play the role of r' and r . Points a, b, c, r form a convex quadrilateral, and we create the path (l', l, a, x, r, r') .
- $\ell(b, c)$ intersects al and ar . We may assume, without loss of generality, that b is to the left of c . Then \vec{ab} intersects L or \vec{ac} intersects R . Because of symmetry, assume that \vec{ab} intersects L . Their intersection, y , must be above l . Denote by y' the intersection point of $\ell(b, c)$ with L , as in Figure 2(b). Observe that y' is below l . In this setting $\vec{l'l}$ intersects either by or by' , at a point u . If $\vec{l'l}$ intersects by , we set $\Delta = \delta \oplus (l', u, a, c, r, r')$ as in Figure 2(b), otherwise we set $\Delta = \delta \oplus (l', u, c, a, r, r')$ as in Figure 2(c). Either way we cover 6 vertices with 5 segments, by avoiding lb on our path.

Case 2. $\text{CH}(S)$ has 5 vertices (i.e., l, a, b, c, r). By reflection and relabeling, there are two subcases to consider.

- The three vertices a, b , and c lie on the same side of lr (*wlog* below). We may assume that they appear in the order a, b, c from left to right; see Figure 3(a). Then the intersection point x of $\ell(l, a)$ and $\ell(b, c)$ lies in the slab. We set $\Delta = \delta \oplus (l', l, x, c, r, r')$.
- Two vertices, say a and b , lie on one side of lr (*wlog* below), and c lies on the other side. Assume that a is to the left of b . The intersection, u , of $\ell(l, a)$ and $\ell(r, b)$ must be in the slab, by convexity. If l' is above $\ell(l, c)$ then we set $\Delta = \delta \oplus (l', c, l, u, r, r')$, as shown in Figure 3(b). Symmetrically, if r' is above $\ell(r, c)$, we set $\Delta = \delta \oplus (l', l, u, r, c, r')$. Finally if l' is below $\ell(l, c)$ and r' is below $\ell(r, c)$, let v be the intersection point of $\ell(l', l)$ and $\ell(r, c)$; see Figure 3(c). By convexity, v is in the slab, so we can set $\Delta = \delta \oplus (l', v, r, a, b, r')$.

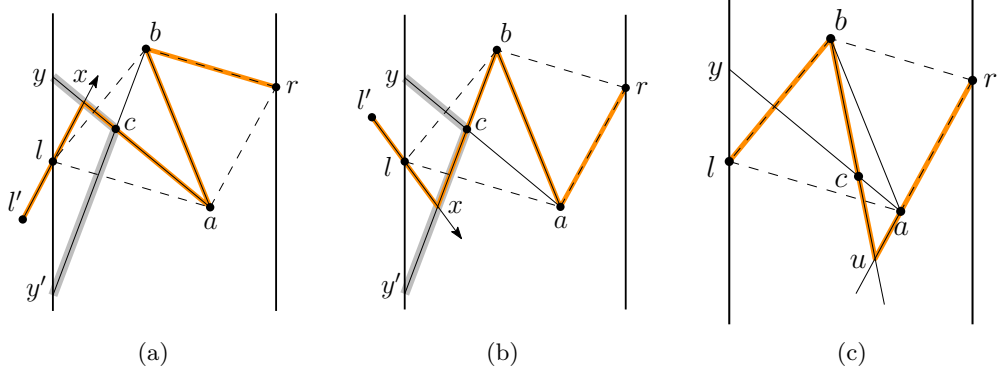
Case 3. $\text{CH}(S)$ has 4 vertices: l, r, a, b . We consider two subcases.

Case 3a. a and b lie on different sides of lr . *Wlog*, a is below. The point c lies either in triangle $\triangle(a, b, l)$ or in triangle $\triangle(a, b, r)$. The configurations are symmetric so we show how



■ **Figure 3** $\text{CH}(S)$ has five vertices. (a) a, b, c lie on the same side of lr . (b) l' is above $\ell(l, c)$. (c) l' is below $\ell(l, c)$ and r' is below $\ell(r, c)$.

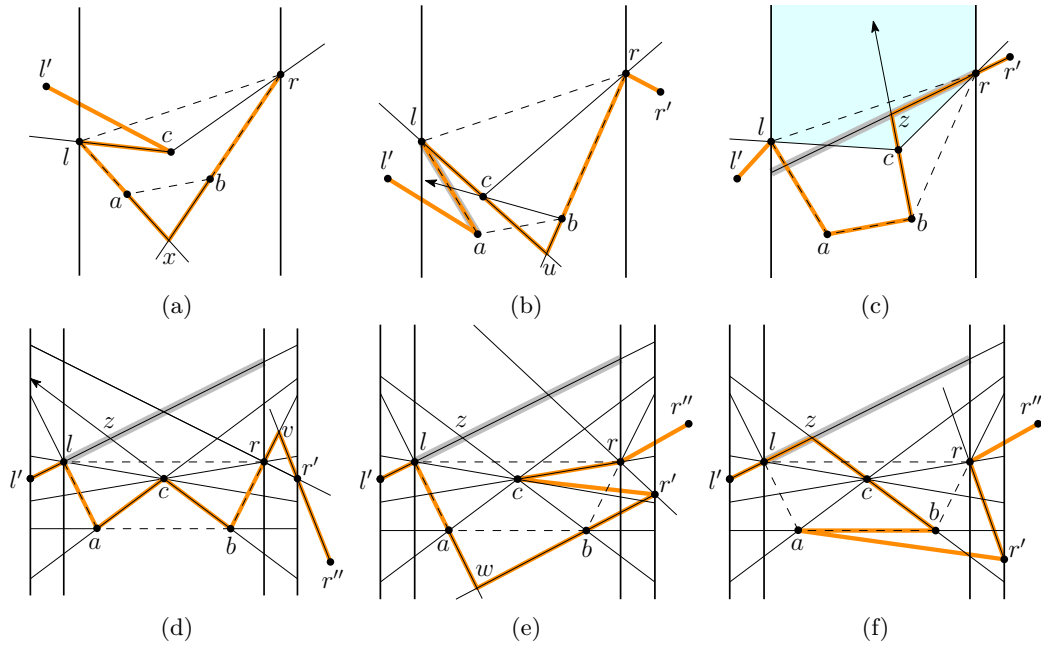
to handle the former. It is implied that c is to the left of \overrightarrow{ab} , which in turn implies that at least one of \overrightarrow{ac} or \overrightarrow{bc} intersects L . Wlog, assume that \overrightarrow{ac} intersects L . The intersection point, y , must be above l . There are two subcases to consider:



■ **Figure 4** $\text{CH}(S)$ has four vertices, the points a and b lie on different sides of lr , the point c is in $\triangle(a, b, l)$, and \overrightarrow{ac} intersects L . (a) \overrightarrow{bc} intersects L , and $\overrightarrow{l'l}$ intersects cy . (b) \overrightarrow{bc} intersects L , and $\overrightarrow{l'l}$ intersects cy' . (c) \overrightarrow{bc} intersects R .

- \overrightarrow{bc} intersects L , at point y' , which must be below l . Then either $\overrightarrow{l'l}$ intersects cy as in Figure 4(a), or it intersects cy' as in Figure 4(b). Either way, let the intersection point be x . Respectively, we set $\Delta = \delta \oplus (l', x, a, b, r, r')$ or we set $\Delta = \delta \oplus (l', x, b, a, r, r')$.
- \overrightarrow{bc} intersects R , as in Figure 4(c). In this case the intersection point u of $\ell(b, c)$ and $\ell(r, a)$ lies in the slab. We set $\Delta = \delta \oplus (l', l, b, u, r, r')$.

Case 3b. a and b are on the same side of lr , (wlog, below; and also a is to the left of b). By convexity, the intersection point, x , of $\ell(l, a)$ and $\ell(r, b)$ is in the slab. If l' lies above $\ell(l, c)$, as in Figure 5(a), we set $\Delta = \delta \oplus (l', c, l, x, r, r')$. Symmetrically, if r' lies above $\ell(r, c)$, then we set $\Delta = \delta \oplus (l', l, x, r, c, r')$. It remains to consider the case when l' is below $\ell(l, c)$ and r' is below $\ell(r, c)$. We consider two subcases.



■ **Figure 5** $\text{CH}(S)$ has four vertices, a and b lie on the same side of lr . (a) l' is above $\ell(l, c)$. In (b)-(e) we have l' below $\ell(l, c)$ and r' below $\ell(r, c)$. (b) \vec{bc} intersects al . (c) \vec{bc} intersects $r'r$ within the slab. (d) r'' is below $\ell(r, r')$. (e) r'' is above $\ell(r, r')$, and r' is above $\ell(a, b)$. (f) r'' is above $\ell(r, r')$, and r' is below $\ell(a, b)$.

- \vec{bc} intersects al , as in Figure 5(b). In this case, the intersection point u of \vec{bc} and \vec{al} is in the slab because c is to the right of l , and b is to the left of r . Moreover, since l is above \vec{bc} , u does not lie on either of the segments lc and rb . We set $\Delta = \delta \oplus (l', a, l, u, r, r')$. Symmetrically, if \vec{ac} intersects br we can set $\Delta = \delta \oplus (l', l, u, r, b, r')$, but this does not seem to help or simplify the next case.
 - \vec{bc} does not intersect al . In this case \vec{bc} lies in the wedge with counterclockwise boundary ray \vec{bl} and clockwise boundary ray \vec{br} . Because l' is below $\ell(l, c)$ and r' is below $\ell(r, c)$, it follows that \vec{bc} intersects at least one of the rays $\vec{l'l}$ and $\vec{r'r}$ within the slab. If the intersection is with $\vec{r'r}$ then the intersection point z must be inside the shaded region in Figure 5(c), i.e., above lr or in the triangle $\triangle(l, c, r)$. Accordingly, we set $\Delta = \delta \oplus (l', l, a, b, z, r')$. Otherwise, let z be the intersection of \vec{bc} with $\vec{l'l}$. This is the case when we need to scan a new point.
- Let r'' be the next point, after r' . If r'' is below $\ell(r, r')$, then the intersection point v of $\ell(b, r)$ and $\ell(r'', r')$ is in the vertical slab between r and r' ; we set $\Delta = \delta \oplus (l', l, a, c, b, v, r'')$ as in Figure 5(d). Otherwise, r'' is above $\ell(r, r')$. Now if r' is above $\ell(a, b)$, then the intersection point w of $\ell(l, a)$ and $\ell(r', b)$ lies in the slab, below $\ell(a, b)$; we set $\Delta = \delta \oplus (l', l, w, r', c, r, r'')$ as in Figure 5(e). Because r' is below $\ell(c, r)$, cr' does not intersect rr'' . If r' is below $\ell(a, b)$, we set $\Delta = \delta \oplus (l', z, b, a, r', r, r'')$ as in Figure 5(f). In each case we cover seven new points (including r'') with six edges.

This is the end of an intermediate iteration. In each case, we append to δ a non-crossing path that lies in the vertical slab between l and the rightmost scanned point. Thus Δ is non-crossing. Moreover the degree of the rightmost point in Δ (which is r' or r'') is one.

In each case we have scanned 6 or 7 new points and covered them by 5 or 6 new segments, respectively. If we have scanned 6 points, then the total number M of scanned points is $M = m + 6$, and the number of segments of Δ is

$$|\Delta| = |\delta| + 5 \leq \left\lfloor \frac{6m}{7} \right\rfloor + 5 = \left\lfloor \frac{6m + 35}{7} \right\rfloor \leq \left\lfloor \frac{6M}{7} \right\rfloor.$$

If we have scanned 7 points, then $M = m + 7$, and we have

$$|\Delta| = |\delta| + 6 \leq \left\lfloor \frac{6m}{7} \right\rfloor + 6 = \left\lfloor \frac{6m + 42}{7} \right\rfloor = \left\lfloor \frac{6M}{7} \right\rfloor.$$

Therefore, the invariant holds after every intermediate iteration.

In the last iteration of the algorithm we are left with n' points where $1 \leq n' \leq 6$. We cover these points by an x -monotone path with $n' - 1$ segments and connect it to the rightmost scanned point (of the path that has been constructed so far) by an additional segment. This gives a non-crossing covering path for all points. The total number of segments in this path is at most

$$\left\lfloor \frac{6(n - n')}{7} \right\rfloor + (n' - 1) + 1 = \left\lfloor \frac{6(n - n') + 7n'}{7} \right\rfloor = \left\lfloor \frac{6n + n'}{7} \right\rfloor \leq \left\lfloor \frac{6n + 6}{7} \right\rfloor \leq \left\lceil \frac{6n}{7} \right\rceil.$$

Each iteration takes $O(1)$ time. Therefore, after rotating and sorting the points in $O(n \log n)$ time, the rest of the algorithm takes $O(n)$ time. This concludes the proof of Theorem 1.

► **Remark 2.** One could scan exactly 7 points in every intermediate iteration, and obtain the same upper bound. Our choice to scan 6 points whenever possible serves as a demonstration for the complexity of producing a non-crossing covering path with at most $5n/6$ segments.

► **Remark 3.** In contrast to the algorithms of [13] and [18], our algorithm does not rely on the existence of caps or cups to reduce the number of segments used. This has been illustrated in Figure 2(a) when c is to the right of a ; in Figure 5(c) where z is to the left of c ; and in Figure 5(f) when z is to the right of c .

► **Remark 4.** For the 7 points l', l, a, b, c, d, r, r' in Figure 5(d) we have not found a non-crossing path that starts at l' , ends at r' , covers all the points, and stays within the slab defined by l' and r' . This suggests the necessity of considering r'' in our algorithm.

3 Open Problems

A natural open problem is to narrow the gap between the lower bound $5n/9$ and the upper bound $6n/7$ for $\pi(n)$. It is not hard to see that a covering path with a minimum number of segments is a subset of the arrangement of the lines through the $\binom{n}{2}$ pairs of points. Thus one can find a covering path with a minimum number of segments in exponential time. It would be interesting to present a sub-exponential algorithm for computing such a path.

References

- 1 Alok Aggarwal, Don Coppersmith, Sanjeev Khanna, Rajeev Motwani, and Baruch Schieber. The angular-metric traveling salesman problem. In *Proceedings of the Eighth Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 221–229, 1997. URL: <http://dl.acm.org/citation.cfm?id=314161.314259>.

- 2 Alok Aggarwal, Don Coppersmith, Sanjeev Khanna, Rajeev Motwani, and Baruch Schieber. The angular-metric traveling salesman problem. *SIAM Journal on Computing*, 29(3):697–711, 1999. doi:10.1137/S0097539796312721.
- 3 Oswin Aichholzer, Sergio Cabello, Ruy Fabila Monroy, David Flores-Peñaloza, Thomas Hackl, Clemens Huemer, Ferran Hurtado, and David R. Wood. Edge-removal and non-crossing configurations in geometric graphs. *Discrete Mathematics & Theoretical Computer Science*, 12(1):75–86, 2010. doi:10.46298/DMTCS.525.
- 4 Esther M. Arkin, Yi-Jen Chiang, Joseph S. B. Mitchell, Steven Skiena, and Tae-Cheon Yang. On the maximum scatter TSP (extended abstract). In *Proceedings of the Eighth Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 211–220, 1997. URL: <http://dl.acm.org/citation.cfm?id=314161.314258>.
- 5 Esther M. Arkin, Yi-Jen Chiang, Joseph S. B. Mitchell, Steven Skiena, and Tae-Cheon Yang. On the maximum scatter traveling salesperson problem. *SIAM Journal on Computing*, 29(2):515–544, 1999. doi:10.1137/S0097539797320281.
- 6 Esther M. Arkin, Joseph S. B. Mitchell, and Christine D. Piatko. Minimum-link watchman tours. *Information Processing Letters*, 86(4):203–207, 2003. doi:10.1016/S0020-0190(02)00502-1.
- 7 Sanjeev Arora. Polynomial time approximation schemes for Euclidean traveling salesman and other geometric problems. *Journal of the ACM*, 45(5):753–782, 1998. doi:10.1145/290179.290180.
- 8 Sergey Bereg, Prosenjit Bose, Adrian Dumitrescu, Ferran Hurtado, and Pavel Valtr. Traversing a set of points with a minimum number of turns. In *Proceedings of the 23rd ACM Symposium on Computational Geometry (SCG)*, pages 46–55, 2007. doi:10.1145/1247069.1247077.
- 9 Sergey Bereg, Prosenjit Bose, Adrian Dumitrescu, Ferran Hurtado, and Pavel Valtr. Traversing a set of points with a minimum number of turns. *Discrete & Computational Geometry*, 41(4):513–532, 2009. doi:10.1007/S00454-008-9127-1.
- 10 Ahmad Biniaz. Euclidean bottleneck bounded-degree spanning tree ratios. In *Proceedings of the 2020 ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 826–836, 2020. doi:10.1137/1.9781611975994.50.
- 11 Ahmad Biniaz. Euclidean bottleneck bounded-degree spanning tree ratios. *Discrete & Computational Geometry*, 67(1):311–327, 2022. doi:10.1007/S00454-021-00286-4.
- 12 Ahmad Biniaz. Improved bounds for covering paths and trees in the plane. In *39th International Symposium on Computational Geometry (SoCG)*, volume 258, pages 19:1–19:15, 2023. doi:10.4230/LIPICS.SOCG.2023.19.
- 13 Ahmad Biniaz. Improved bounds for covering paths and trees in the plane. *J. Comput. Geom.*, 15(1):88–108, 2024. doi:10.20382/JOCG.V15I1A4.
- 14 Jakub Cerný, Zdenek Dvorák, Vít Jelínek, and Jan Kára. Noncrossing Hamiltonian paths in geometric graphs. *Discrete Applied Mathematics*, 155(9):1096–1105, 2007. doi:10.1016/J.DAM.2005.12.010.
- 15 Jianer Chen, Qin Huang, Iyad Kanj, and Ge Xia. Near-optimal algorithms for point-line covering problems. *CoRR*, abs/2012.02363, 2020. arXiv:2012.02363.
- 16 Michael J. Collins. Covering a set of points with a minimum number of turns. *International Journal of Computational Geometry & Applications*, 14(1-2):105–114, 2004. doi:10.1142/S021819590400138X.
- 17 Erik D. Demaine and Joseph O’Rourke. Open problems from CCCG 2010. In *Proceedings of the 22nd Canadian Conference on Computational Geometry*, 2011.
- 18 Adrian Dumitrescu, Dániel Gerbner, Balázs Keszegh, and Csaba D. Tóth. Covering paths for planar point sets. *Discrete & Computational Geometry*, 51(2):462–484, 2014. doi:10.1007/S00454-013-9563-4.
- 19 P. Erdős and G. Szekeres. A combinatorial problem in geometry. *Compositio Mathematica*, 2:463–470, 1935.
- 20 Sándor P. Fekete and Gerhard J. Woeginger. Angle-restricted tours in the plane. *Computational Geometry: Theory and Applications*, 8:195–218, 1997. doi:10.1016/S0925-7721(96)00012-0.

- 21 David Flores-Peñaloza, Mikio Kano, Leonardo Martínez-Sandoval, David Orden, Javier Tejel, Csaba D. Tóth, Jorge Urrutia, and Birgit Vogtenhuber. Rainbow polygons for colored point sets in the plane. *Discrete Mathematics*, 344(7):112406, 2021. doi:10.1016/J.DISC.2021.112406.
- 22 Radoslav Fulek, Balázs Keszegh, Filip Morić, and Igor Uljarević. On polygons excluding point sets. *Graphs and Combinatorics*, 29(6):1741–1753, 2013. doi:10.1007/S00373-012-1221-8.
- 23 Magdalene Grantson and Christos Levcopoulos. Covering a set of points with a minimum number of lines. In *Proceedings of the 6th International Conference on Algorithms and Complexity (CIAC)*, pages 6–17, 2006. doi:10.1007/11758471_4.
- 24 Minghui Jiang. On covering points with minimum turns. *International Journal of Computational Geometry & Applications*, 25(1):1–10, 2015. doi:10.1142/S0218195915500016.
- 25 Balázs Keszegh. Covering paths and trees for planar grids. *Geombinatorics Quarterly*, 24, 2014.
- 26 Stefan Langerman and Pat Morin. Covering things with things. In *Proceedings of the 10th Annual European Symposium on Algorithms (ESA)*, volume 2461 of *Lecture Notes in Computer Science*, pages 662–673. Springer, 2002. doi:10.1007/3-540-45749-6_58.
- 27 Stefan Langerman and Pat Morin. Covering things with things. *Discrete & Computational Geometry*, 33(4):717–729, 2005. doi:10.1007/S00454-004-1108-4.
- 28 E. L. Lawler, J. K. Lenstra, A. H. G. Rinnooy Kan, and D. B. Shmoys. *The Traveling Salesman Problem*. John Wiley & Sons, New York, NY, 1985.
- 29 Samuel Loyd. *Cyclopedia of 5000 Puzzles, Tricks & Conundrums*. The Lamb Publishing Company, 1914.
- 30 Joseph S. B. Mitchell. Geometric shortest paths and network optimization. In Jörg-Rüdiger Sack and Jorge Urrutia, editors, *Handbook of Computational Geometry*, pages 633–701. North Holland / Elsevier, 2000. doi:10.1016/B978-044482537-7/50016-4.
- 31 Christos H. Papadimitriou. The Euclidean traveling salesman problem is NP-complete. *Theoretical Computer Science*, 4(3):237–244, 1977. doi:10.1016/0304-3975(77)90012-3.
- 32 Clifford Stein and David P. Wagner. Approximation algorithms for the minimum bends traveling salesman problem. In *Proceedings of the 8th International Conference on Integer Programming and Combinatorial Optimization (IPCO)*, pages 406–422, 2001. doi:10.1007/3-540-45535-3_32.