UNIVERSITY OF COPENHAGEN
FACULTY OF SCIENCE

NEW RESULTS ON CLASSICAL PROBLEMS IN COMPUTATIONAL GEOMETRY IN THE PLANE

MIKKEL ABRAHAMSEN

PhD Thesis
August 2017

Advisors: Mikkel Thorup (principal) and Christian Wulff-Nilsen

This thesis has been submitted to the PhD School of The Faculty of Science, University of Copenhagen

ABSTRACT

In this thesis, we revisit three classical problems in computational geometry in the plane.

An obstacle that often occurs as a subproblem in more complicated problems is to compute the common tangents of two disjoint, simple polygons. For instance, the common tangents turn up in problems related to visibility, collision avoidance, shortest paths, etc. We provide a remarkably simple algorithm to compute all (at most four) common tangents of two disjoint simple polygons. Given each polygon as a read-only array of its corners in cyclic order, the algorithm runs in linear time and constant workspace and is the first to achieve the two complexity bounds simultaneously. The set of common tangents provides basic information about the convex hulls of the polygons—whether they are nested, overlapping, or disjoint—and our algorithm thus also decides this relationship.

One of the best-known problems in computational geometry is the art gallery problem, which was already studied in the early 70s. Many variations of the problem have been considered, but here we study the classical version where we are given a simple polygon with vertices at rational coordinates and we have to decide whether a given number of guards can be placed in the polygon so that they guard the entire polygon. We give an explicit example of a polygon where three guards are sufficient, but only if they are placed on specific points with irrational coordinates. If the coordinates of the guards are required to be rational, then four guards are needed. We furthermore prove the much more general result that the art gallery problem is complete for the complexity class $\exists\mathbb{R}$, implying that (1) the art gallery problem is equivalent up to polynomial time reductions to the problem of deciding whether a given system of polynomial equations and inequalities with integer coefficients and any number of variables has a solution, and (2) the art gallery problem is not in the complexity class NP unless $\mathsf{NP} = \exists\mathbb{R}$. As a corollary of our construction, we prove that for any real algebraic number α there is an instance of the art gallery problem where one of the coordinates of the guards equals α in any guard set of minimum cardinality. That rules out many geometric approaches to the problem.

A natural clustering problem for points in the plane, which has been studied since the early 90s, is the minimum perimeter sum problem. Here, we are given n points and we want to find the way to partition the points into some number k of clusters so that the sum of perimeters of the convex hulls of the clusters is minimum. For the special case of $k = 2$, the fastest previously known algorithm had quadratic running time and we provide an $O(n \log^4 n)$ time algorithm.

DANSK RESUMÉ (DANISH ABSTRACT)

I denne afhandling undersøger vi tre klassiske problemer fra algoritmisk geometri i planen.

Et ofte forekommende delproblem i mere komplicerede problemer er at beregne fællestangenterne af to disjunkte, simple polygoner. Fællestangenterne dukker for eksempel op i problemer vedrørende synlighed, sammenstødsafværgelse, beregning af kortest veje, etc. Vi beskriver en bemærkelsesværdigt simpel algoritme som beregner alle (højst fire) fællestangenter af to disjunkte simple polygoner. Hvert inputpolygon er givet som en tabel af hjørnerne i cyklisk orden i et skrivebeskyttet lager, og algoritmen bruger lineær tid og en konstant mængde arbejdslager og er den første til at opnå disse to kompleksitetsgrænser samtidigt. Mængden af fællestangenter giver basal information om de konvekse hylstre af polygonerne—om det ene er indeholdt i det andet, om de overlapper eller om de er disjunkte—og vores algoritme bestemmer derfor også dette forhold.

Et af de mest velkendte problemer i algoritmisk geometri er kunstmuseumsproblemet, som er blevet studeret siden først i 70'erne. Mange varianter af problemet er blevet betragtet, men her studerer vi den klassiske version hvor vi er givet et simpelt polygon med hjørner i rationale koordinater og vi skal afgøre hvorvidt et givet antal vagter kan placeres i polygonet sådan at de bevogter hele polygonet. Vi giver et eksplicit eksempel på et polygon hvor tre vagter er tilstrækkelige, men kun hvis de placeres på specifikke punkter med irrationale koordinater. Hvis koordinaterne skal være rationale må fire vagter benyttes. Derudover beviser vi det meget mere generelle resultat at kunstmuseumsproblemet er komplet for kompleksitetsklassen $\exists\mathbb{R}$, hvilket medfører at (1) kunstmuseumsproblemet er ækvivalent under polynomialtidsreduktioner med problemet at afgøre om et givet system af polynomiale ligninger og uligheder med heltallige koefficienter og et vilkårligt antal variable har en løsning, og (2) kunstmuseumsproblemet ikke er i kompleksitetsklassen NP medmindre $\mathsf{NP} = \exists\mathbb{R}$. Det følger som et korollar af vores konstruktion at der for ethvert reelt algebraisk tal α findes en instans af kunstmuseumsproblemet så der blandt ethvert minimalt antal vagter, som tilsammen bevogter polygonet, er en vagt med et koordinat lig α. Dette udelukker mange geometriske tilgange til problemet.

Et naturligt grupperingsproblem for punkter i planen, som er blevet studeret siden begyndelsen af 90'erne, er minimum omkreds-sumproblemet. Givet n punkter vil vi finde en måde at inddele dem på i k grupper sådan at summen af omkredsene af gruppernes konvekse hylstre er minimal. I det specielle tilfælde $k = 2$ havde den hidtil hurtigste kendte algoritme kvadratisk køretid og vi beskriver en algoritme med køretid $O(n \log^4 n)$.

PREFACE

Some of the problems on which I have worked during my PhD studies have been rather distantly related. To get a more coherent thesis, I have chosen to expose a selected subset of my work, namely the results from four papers on classical problems in computational geometry in the plane. The thesis is written "*as a synopsis with manuscripts of papers or already published papers attached*" in accordance with the *General rules and guidelines for the PhD programme* at the Faculty of Science, University of Copenhagen.

For completeness, I here list all the papers I have been working on while being a PhD student. Ten papers have been published at peer-reviewed venues and three are still in submission or preparation. The manuscript [AW17] is a merged and much revised version of the conference papers [Abr15a] and [AW16].

[Abr15a] M. Abrahamsen. "An Optimal Algorithm for the Separating Common Tangents of Two Polygons." In: *31st International Symposium on Computational Geometry (SoCG 2015)*. 2015, pp. 198–208.

[Abr15b] M. Abrahamsen. "Spiral Toolpaths for High-Speed Machining of 2D Pockets With or Without Islands." In: *ASME 2015 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*. 2015.

[Abr+16] M. Abrahamsen, G. Bodwin, E. Rotenberg, and M. Stöckel. "Graph Reconstruction with a Betweenness Oracle." In: *33rd Symposium on Theoretical Aspects of Computer Science (STACS 2016)*. 2016, 5:1–5:14.

[AT16] M. Abrahamsen and M. Thorup. "Finding the Maximum Subset with Bounded Convex Curvature." In: *32nd International Symposium on Computational Geometry (SoCG 2016)*. 2016, 4:1–4:17.

[AW16] M. Abrahamsen and B. Walczak. "Outer Common Tangents and Nesting of Convex Hulls in Linear Time and Constant Workspace." In: *24th Annual European Symposium on Algorithms (ESA 2016)*. 2016, 4:1–4:15.

[Abr+17b] M. Abrahamsen, M. de Berg, K. Buchin, M. Mehr, and A.D. Mehrabi. "Minimum Perimeter-Sum Partitions in the Plane." In: *33rd International Symposium on Computational Geometry (SoCG 2017)*. 2017.

[Abr+17e] M. Abrahamsen, M. de Berg, K. Buchin, M. Mehr, and A.D. Mehrabi. "Range-Clustering Queries." In: *33rd International Symposium on Computational Geometry (SoCG 2017)*. 2017.

[Abr+17a] M. Abrahamsen, J. Holm, E. Rotenberg, and C. Wulff-Nilsen. "Best Laid Plans of Lions and Men." In: *33rd International Symposium on Computational Geometry (SoCG 2017)*. 2017.

[AAM17a] M. Abrahamsen, A. Adamaszek, and T. Miltzow. "Irrational Guards are Sometimes Needed." In: *33rd International Symposium on Computational Geometry (SoCG 2017)*. 2017.

[Abr+17c] M. Abrahamsen, S. Alstrup, J. Holm, M.B.T. Knudsen, and M. Stöckel. "Near-Optimal Induced Universal Graphs for Bounded Degree Graphs." In: *The 44th International Colloquium on Automata, Languages, and Programming (ICALP 2017)*. 2017.

[AAM17b] M. Abrahamsen, A. Adamaszek, and T. Miltzow. *The Art Gallery Problem is* $\exists\mathbf{R}$*-complete*. Preprint, https://arxiv.org/abs/1704.06969. 2017.

[Abr+17d] M. Abrahamsen, S. Alstrup, J. Holm, M.B.T. Knudsen, and M. Stöckel. *Near-Optimal Induced Universal Graphs for Cycles and Paths*. In submission. 2017.

[AW17] M. Abrahamsen and B. Walczak. *Common Tangents of Two Simple Polygons in Linear Time and Constant Workspace*. Manuscript. 2017.

The papers that I have not included in this thesis contain results related to either geometry or graphs, and a brief discussion of the results is included below.

ADDITIONAL PAPERS ON GEOMETRY In the paper [Abr15b], a method is described for computing a spiral toolpath to be used in computer-aided manufacturing. The concept of regions in the plane with bounded convex curvature is introduced in [AT16]. It is described how the problem of finding maximum subsets with bounded convex curvature naturally arises in computer-aided manufacturing, and an efficient algorithm to compute such subsets is described. In the paper [Abr+17e], we present data structures for orthogonal range-clustering queries on a set S of points in the plane, that is, given a query box Q and an integer k, the data structure can quickly return an optimal k-clustering of $S \cap Q$ with respect to various clustering problems. The paper [Abr+17a] is on the classical lion and man game and has two main results. The first answers a question dating back to J.E. Littlewood (1885–1977) by showing that two lions are not always enough to catch a man in a bounded region with obstacles. The second is that a fast man can escape arbitrarily many slightly slower lions in an unbounded region without obstacles. Unfortunately, it has since then come to our attention that the first of these results had essentially already been shown by Bhadauria et al. [Bha+12].

PAPERS ON GRAPHS In the paper [Abr+16], we give an efficient algorithm to reconstruct an unknown graph using only between-

ness queries, that is, for three vertices u, v, w, we may ask an oracle whether there is a shortest path from u to w that contains v, and the complexity of an algorithm is measured as the number of queries that is used. Improved upper and lower bounds are presented in [Abr+17c] on the size of the smallest induced universal graph for the family of graphs with n vertices and bounded degree D. In the related paper [Abr+17d], we study induced universal graphs for families of cycles and paths.

ACKNOWLEDGEMENT I would like to thank my supervisors Mikkel Thorup and Christian Wulff-Nilsen for their guidance and support during my studies. In particular, I am grateful that Mikkel encouraged me to keep working on problems within my primary field of interest, namely computational geometry, although it is neither his own area of expertise, nor within the primary focus of our research group. It made me more motivated and industrious to work on the problems I found the most exciting, and I think it also had a positive effect on my academic maturity and independence that I often had to find my own way.

My thanks also go to all of my collaborators: Anna Adamaszek, Stephen Alstrup, Kevin Buchin, Greg Bodwin, Mark de Berg, Jacob Holm, Mathias Bæk Tejs Knudsen, Mehran Mehr, Ali Mehrabi, Tillmann Miltzow, Eva Rotenberg, Morten Stöckel, Mikkel Thorup, Bartosz Walczak, and Christian Wulff-Nilsen. It has been a pleasure to work with each of these people, and I am proud of the results we have obtained together.

CONTENTS

Part I

SYNOPSIS

1 INTRODUCTION

Computational geometry plays an important role in computer graphics, geographic information systems, and computer aided design and manufacturing, to mention a few areas. This is one reason that it has been an active research area at least since the term "computational geometry" was coined by M.I. Shamos in 1975 [PS85]. Another reason is that the area contains a plethora of problems very appealing from an aesthetical/theoretical perspective.

In this thesis, we present some new results on three classical problems in computational geometry in the plane. Indeed, the youngest of the problems have been studied for more than a quarter of a century at the time of writing. We give an introduction to each problem with references to the relevant literature as well as a summary of our results on the problem. We do not give full proofs, but refer the reader to the papers in the appendix for the details. With the exception of a few details in the "Concluding Remarks" sections 2.4 and 3.4, the chapters do not contain any information which is not also present in the respective papers in the appendix, so the reader might as well just read the papers.

1.1 NOTATION AND DEFINITIONS

For any two points a and b in the plane, the closed line segment with endpoints a and b is denoted by ab. When $a \neq b$, the two-way infinite straight line passing through a and b is denoted by $\mathcal{L}(a, b)$. The segment ab and the line $\mathcal{L}(a, b)$ are considered *oriented* in the direction from a towards b. A *simple polygon* or just a *polygon* with *corners* $a_0, \ldots, a_{n-1}$, denoted by $\mathcal{P}(a_0, \ldots, a_{n-1})$, is a closed curve in the plane composed of n *edges* $a_0a_1, \ldots, a_{n-2}a_{n-1}, a_{n-1}a_0$ that have no common points other than the common endpoints of pairs of edges consecutive in that cyclic order. The polygon $\mathcal{P}(a_0, \ldots, a_{n-1})$ is considered *oriented* so that its *forward* traversal visits corners $a_0, \ldots, a_{n-1}$ in this cyclic order. A *polygonal region* is a closed and bounded region of the plane that is bounded by a polygon.

When the orientations of line segments, lines, and polygons are not used, the objects are just considered as sets of points in the plane.

2

THE COMMON TANGENT PROBLEM

This chapter is based on the paper [AW17] (appendix A), which is a merged and much revised version of the conference papers [Abr15a] and [AW16].

ABSTRACT We provide a remarkably simple algorithm to compute all (at most four) common tangents of two disjoint simple polygons. Given each polygon as a read-only array of its corners in cyclic order, the algorithm runs in linear time and constant workspace and is the first to achieve the two complexity bounds simultaneously. The set of common tangents provides basic information about the convex hulls of the polygons—whether they are nested, overlapping, or disjoint—and our algorithm thus also decides this relationship.

2.1 INTRODUCTION

A tangent of a polygon is a line touching the polygon such that all of the polygon lies on the same side of the line. We consider the problem of computing the common tangents of two disjoint polygons that are simple, that is, they have no self-intersections. The set of common tangents provides basic information about the convex hulls of the polygons, i.e., whether they are disjoint, overlapping, or nested. We call a common tangent outer if the two polygons lie on the same side of it and separating otherwise. Two disjoint polygons have two outer common tangents unless their convex hulls are nested, and if they are properly nested, then there is no outer common tangent. Two polygons have two separating common tangents unless their convex hulls overlap, and if they properly overlap, then there is no separating common tangent. See Figure 1. Common tangents arise in many different contexts, for instance in problems related to convex hulls [PH77], shortest paths [GH89], ray shooting [HS95], and clustering [Abr+17b].

We provide a very simple algorithm to compute all at most four common tangents of two disjoint simple polygons. In view of the above, our algorithm also determines whether the two polygons have (properly) nested, (properly) overlapping, or disjoint convex hulls. Given each of the two polygons as a read-only array of its corners in cyclic order, our algorithm runs in linear time and uses seven variables each storing a boolean value or an index of a corner in one of the arrays. The algorithm is therefore asymptotically optimal with respect to time as well as workspace, and operates in the *constant workspace model* of computation.

The constant workspace model is a restricted version of the RAM model in which the input is read-only, the output is write-only, and only $O(\log n)$ additional bits of *workspace* (with both read and write

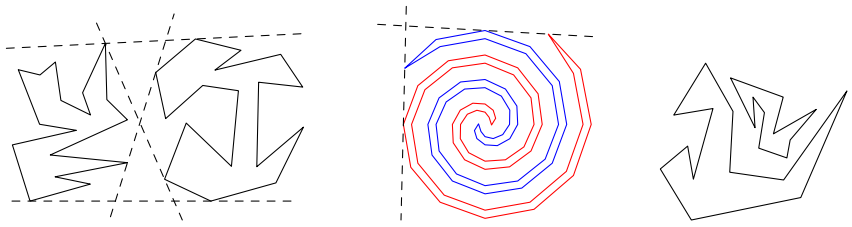


Figure 1: Left: The convex hulls are disjoint—outer and separating common tangents exist. Middle: The convex hulls overlap—only outer common tangents exist. Right: The convex hulls are nested—no common tangents exist.

access) are available, where n denotes the size of the input. Clearly, $\Omega(\log n)$ bits of workspace are necessary to solve any interesting computational problem, because that many bits are required to store an index of or a pointer to an entry in the input. Since blocks of $\Theta(\log n)$ bits are considered to form *words* in the memory, algorithms in the constant workspace model use $O(1)$ words of workspace, which explains the name of the model. The practical relevance of studying problems in the constant workspace model is increasing, as there are many current and emerging memory technologies where writing can be much more expensive than reading in terms of time and energy [Car+16].

The constant workspace model was first considered explicitly for geometric problems by Asano et al. [Asa+11]. Recently, there has been growing interest in algorithms for geometric problems using constant or restricted workspace, see for instance [Asa+13; Bar+15; Bar+14; HP15]. In complexity theory, the class of the decision problems solvable using constant workspace is usually known as L. The constant workspace model has been shown to be surprisingly powerful—for instance, the problem of deciding whether two vertices in an undirected graph are in the same connected component is in L [Rei08].

The problem of computing common tangents of two polygons has received much attention in the special case that the polygons are convex. For instance, computing the outer common tangents of disjoint convex polygons is used as a subroutine in the classical divide-and-conquer algorithm for the convex hull of a set of n points in the plane due to Preparata and Hong [PH77]. They gave a naive linear-time algorithm for outer common tangents, which suffices for an $O(n \log n)$-time convex hull algorithm. The problem is also considered in various dynamic convex hull algorithms [BJ02; HS92; OL81]. Overmars and van Leeuwen [OL81] gave an $O(\log n)$-time algorithm for computing an outer common tangent of two disjoint convex polygons when a separating line is known, where each polygon has at most n corners. Kirkpatrick and Snoeyink [KS95] gave an $O(\log n)$-time algorithm for the same problem but without using a separating line. Guibas et al. [GHS91] gave a lower bound of $\Omega(\log^2 n)$ on the time required to compute an outer common tangent of two intersecting convex polygons even when they are known to intersect in at most two points. They also described an algorithm achieving that bound. Tou-

ssaint [Tou83] considered the problem of computing separating common tangents of convex polygons. He gave a linear-time algorithm using the technique of the "rotating calipers". Guibas et al. [GHS91] gave an $O(\log n)$-time algorithm for the same problem. All the above-mentioned algorithms with sublinear running times make essential use of convexity of the polygons. If the polygons are not convex, a linear-time algorithm due to Melkman [Mel87] can be used to compute the convex hulls before computing the tangents. However, if the polygons are given in read-only memory, then $\Omega(n)$ extra bits are required to store the convex hulls, so this approach does not work in the constant workspace model.

In the following we describe our algorithm, which is presented in full detail using pseudocode in Algorithm 2. (Algorithm 1 is a simplified version of Algorithm 2, which finds the separating common tangents in all cases, but is only guaranteed to find the outer common tangents when the convex hulls of the polygons are disjoint.) In order to find a particular common tangent of two polygons P_0 and P_1, we maintain a pair of corners of support q_0 and q_1, one in each polygon, that together define a candidate $\mathcal{L}(q_0, q_1)$ for the tangent (i.e., $\mathcal{L}(q_0, q_1)$ is the line containing q_0 and q_1). We traverse the polygons one edge at a time, alternating between the polygons. Each polygon is traversed from its corner of support in a cyclic order determined by the type of tangent that we wish to compute—the two separating common tangents correspond to traversing both polygons the same direction whereas the outer common tangents can be found by choosing different directions. We check that each traversed edge is on the correct side of $\mathcal{L}(q_0, q_1)$. When an edge e of, say, P_0 is found that ends at a corner q_0' on the wrong side (i.e., the edge e shows that $\mathcal{L}(q_0, q_1)$ is not the desired tangent), there are two cases. We consider the order of the following three points on $\mathcal{L}(q_0, q_1)$: q_0, q_1, and the intersection point of e and $\mathcal{L}(q_0, q_1)$. If q_1 is not the middle point of these three, we update q_0 to q_0', thus also update the candidate line, and retract the traversal of P_1 to q_1. Otherwise, it can be seen that q_1 must be in the convex hull of P_0 and q_1 can therefore not be a support of the tangent we wish to find. In that case, we block q_0 from being updated and only care to traverse P_1 until an update to q_1 happens. If q_1 has not been updated after a full traversal of P_1, then the convex hulls are nested and no common tangents exist. Otherwise, we unblock q_0 when q_1 is updated. The correctness of the algorithm relies on the surprising fact that if the tangent exists, an update to a corner of support can only happen during the first or second traversal of the respective polygon. If an update happens in the third traversal of a polygon, we conclude that a common tangent of the desired type does not exist. Otherwise, we can after the third traversal of both polygons conclude that the candidate line $\mathcal{L}(q_0, q_1)$ coincides with the tangent that we wish to find. A more detailed description of the algorithm will follow in section 2.3.

2.2 BASIC TERMINOLOGY AND NOTATION

For any two points $a = (a_x, a_y)$ and $b = (b_x, b_y)$ in $\mathbb{R}^2$, we let

$$\det(a, b) = \begin{vmatrix} a_x & b_x \\ a_y & b_y \end{vmatrix} = a_x b_y - b_x a_y.$$

For $a_0, \ldots, a_{n-1} \in \mathbb{R}^2$, we let

$$\det^\star(a_0, \ldots, a_{n-1}) = \det(a_0, a_1) + \cdots + \det(a_{n-2}, a_{n-1}) + \det(a_{n-1}, a_0).$$

In particular, for any three points $a = (a_x, a_y)$, $b = (b_x, b_y)$, and $c = (c_x, c_y)$ in $\mathbb{R}^2$, we have

$$\det^\star(a, b, c) = \begin{vmatrix} a_x & b_x \\ a_y & b_y \end{vmatrix} + \begin{vmatrix} b_x & c_x \\ b_y & c_y \end{vmatrix} + \begin{vmatrix} c_x & a_x \\ c_y & a_y \end{vmatrix} = \begin{vmatrix} a_x & b_x & c_x \\ a_y & b_y & c_y \\ 1 & 1 & 1 \end{vmatrix}.$$

For two distinct points a and b in the plane, the *left side* and the *right side* of an oriented line $\mathcal{L}(a, b)$ are the two closed half-planes $\mathrm{LHP}(a, b)$ and $\mathrm{RHP}(a, b)$, respectively, defined as follows:

$$\mathrm{LHP}(a, b) = \{c \in \mathbb{R}^2 : \det^\star(a, b, c) \geqslant 0\},$$
$$\mathrm{RHP}(a, b) = \{c \in \mathbb{R}^2 : \det^\star(a, b, c) \leqslant 0\}.$$

An oriented polygon $\mathcal{P}(a_0, \ldots, a_{n-1})$ is *counterclockwise* when

$$\det^\star(a_0, \ldots, a_{n-1}) > 0$$

and *clockwise* when

$$\det^\star(a_0, \ldots, a_{n-1}) < 0.$$

We assume for the rest of this chapter that P_0 and P_1 are two disjoint simple polygons with n_0 and n_1 corners, respectively, each defined by a read-only array of its corners:

$$P_0 = \mathcal{P}(p_0[0], \ldots, p_0[n_0 - 1]), \qquad P_1 = \mathcal{P}(p_1[0], \ldots, p_1[n_1 - 1]).$$

We make no assumption (yet) on whether P_0 and P_1 are oriented counterclockwise or clockwise. We further assume that the corners of P_0 and P_1 are in general position, that is, P_0 and P_1 have no corners in common and the combined set of corners $\{p_0[0], \ldots, p_0[n_0 - 1], p_1[0], \ldots, p_1[n_1 - 1]\}$ contains no triple of collinear points. This assumption simplifies the description and the analysis of the algorithm but can be avoided, as we explain in the last section. We do not assume the polygonal regions bounded by P_0 and P_1 to be disjoint—they may be nested. Indices of the corners of each P_k are considered modulo n_k, so that $p_k[i]$ and $p_k[j]$ denote the same corner when $i \equiv j \pmod{n_k}$.

A *tangent* of P_k is a line L such that P_k has a common point with L and is contained in one of the two closed half-planes determined by L. A line L is a *common tangent* of P_0 and P_1 if it is a tangent of both P_0

and P_1; it is an *outer common tangent* if P_0 and P_1 lie on the same side of L and a *separating common tangent* otherwise. The following lemma asserts well-known properties of common tangents of polygons. See Figure 1.

Lemma 1. *A line is a tangent of a polygon* P *if and only if it is a tangent of the convex hull of* P*. Moreover, under the general position assumption, the following holds:*

- P_0 *and* P_1 *have no common tangents if the convex hulls of* P_0 *and* P_1 *are nested;*
- P_0 *and* P_1 *have two outer common tangents and no separating common tangents if the convex hulls of* P_0 *and* P_1 *properly overlap;*
- P_0 *and* P_1 *have two outer common tangents and two separating common tangents if the convex hulls of* P_0 *and* P_1 *are disjoint.*

2.3 ALGORITHMS

We distinguish four particular cases of the common tangent problem: find the pair of indices (s_0, s_1) such that

1. $P_0 \subset \mathrm{RHP}(p_0[s_0], p_1[s_1])$ and $P_1 \subset \mathrm{RHP}(p_0[s_0], p_1[s_1])$,
2. $P_0 \subset \mathrm{LHP}(p_0[s_0], p_1[s_1])$ and $P_1 \subset \mathrm{LHP}(p_0[s_0], p_1[s_1])$,
3. $P_0 \subset \mathrm{RHP}(p_0[s_0], p_1[s_1])$ and $P_1 \subset \mathrm{LHP}(p_0[s_0], p_1[s_1])$,
4. $P_0 \subset \mathrm{LHP}(p_0[s_0], p_1[s_1])$ and $P_1 \subset \mathrm{RHP}(p_0[s_0], p_1[s_1])$.

The line $\mathcal{L}(p_0[s_0], p_1[s_1])$ is an outer common tangent in cases 1–2 and a separating common tangent in cases 3–4. We say that (s_0, s_1) is the *solution* to the particular case of the problem. An algorithm solving each case 1–4 is expected to find and return the solution (s_0, s_1) if it exists (i.e. the convex hulls of P_0 and P_1 are not nested in cases 1–2 and are disjoint in cases 3–4) and to report "no solution" otherwise.

We will describe two general algorithms. Algorithm 1, very simple, fully solves the separating common tangent problem (cases 3–4), finding the separating common tangent if the convex hulls of P_0 and P_1 are disjoint and otherwise reporting that the requested tangent does not exist. Furthermore, Algorithm 1 solves the outer common tangent problem (cases 1–2) provided that the convex hulls of P_0 and P_1 are disjoint. Algorithm 1 also correctly reports that the outer common tangents do not exist if the convex hulls of P_0 and P_1 are nested. However, Algorithm 1 can fail to find the outer common tangents if the convex hulls of P_0 and P_1 properly overlap. Algorithm 2 is an improved version of Algorithm 1 that solves the problem correctly in all cases.

The general idea behind either algorithm is as follows. The algorithm maintains a pair of indices (s_0, s_1) called the *candidate solution*, which determines the line $\mathcal{L}(p_0[s_0], p_1[s_1])$ called the *candidate line*. If each of the two polygons lies on the "correct side" of the candidate line, which is either $\mathrm{RHP}(p_0[s_0], p_1[s_1])$ or $\mathrm{LHP}(p_0[s_0], p_1[s_1])$

depending on the particular case of 1–4 to be solved, then the algorithm returns (s_0, s_1) as the requested solution. Otherwise, for some $u \in \{0, 1\}$, the algorithm finds an index v_u such that $p_u[v_u]$ lies on the "wrong side" of the candidate line, updates s_u by setting $s_u \leftarrow v_u$, and repeats. This general scheme guarantees that if (s_0, s_1) is claimed to be the solution, then it indeed is. However, the algorithm can fall in an infinite loop—when there is no solution or when the existing solution keeps being missed. Detailed implementation of the scheme must guarantee that the solution is found in linearly many steps if it exists. Then, if the solution is not found in the guaranteed number of steps, the algorithm terminates and reports "no solution".

The particular case of 1–4 to be solved is specified to the algorithms by providing two binary parameters $\alpha_0, \alpha_1 \in \{+1, -1\}$ specifying that the final solution (s_0, s_1) should satisfy

$$\begin{aligned}
P_0 &\subset \mathrm{RHP}(p_0[s_0], p_1[s_1]) \quad \text{if } \alpha_0 = +1,\\
P_0 &\subset \mathrm{LHP}(p_0[s_0], p_1[s_1]) \quad \text{if } \alpha_0 = -1,\\
P_1 &\subset \mathrm{RHP}(p_0[s_0], p_1[s_1]) \quad \text{if } \alpha_1 = +1,\\
P_1 &\subset \mathrm{LHP}(p_0[s_0], p_1[s_1]) \quad \text{if } \alpha_1 = -1.
\end{aligned}$$

For clarity, instead of using the parameters α_0 and α_1 explicitly, the pseudocode uses half-planes $\mathcal{H}_0(a, b)$ and $\mathcal{H}_1(a, b)$ defined as follows, for any $k \in \{0, 1\}$ and any distinct $a, b \in \mathbb{R}^2$:

$$\mathcal{H}_k(a, b) = \left\{c \in \mathbb{R}^2 \colon \alpha_k \det^\star(a, b, c) \leqslant 0\right\} = \begin{cases} \mathrm{RHP}(a, b) & \text{if } \alpha_k = +1,\\ \mathrm{LHP}(a, b) & \text{if } \alpha_k = -1. \end{cases}$$

Therefore, a test of the form $c \notin \mathcal{H}_k(a, b)$ in the pseudocode should be understood as testing whether $\alpha_k \det^\star(a, b, c) > 0$. Another assumption that we make when presenting the pseudocode concerns the direction in which each polygon P_k is traversed in order to find an index v_k such that $p_k[v_k] \notin \mathcal{H}_k(p_0[s_0], p_1[s_1])$. For a reason that will become clear later when we analyze correctness of the algorithms, we require that

- P_0 is traversed counterclockwise when $\alpha_1 = +1$ and clockwise when $\alpha_1 = -1$,
- P_1 is traversed clockwise when $\alpha_0 = +1$ and counterclockwise when $\alpha_0 = -1$.

In the pseudocode, the forward orientation of P_k is *assumed* to be the one in which the corners of P_k should be traversed according to the conditions above. When this has not been guaranteed in the problem setup, a reference to a corner of P_k of the form $p_k[i]$ in the pseudocode should be understood as $p_k[\beta_k i]$ for the constant $\beta_k \in \{+1, -1\}$ computed as follows at the very beginning:

$$\begin{aligned}
\beta_0 &= \alpha_1 \operatorname{sgn} \det^\star(p_0[0], \ldots, p_0[n_0 - 1]),\\
\beta_1 &= -\alpha_0 \operatorname{sgn} \det^\star(p_1[0], \ldots, p_1[n_1 - 1]).
\end{aligned}$$

Algorithm 1:

```
1 s_0 ← 0;  v_0 ← 0;  s_1 ← 0;  v_1 ← 0;  u ← 0
2 while s_0 < 2n_0 and s_1 < 2n_1 and (v_0 < s_0 + n_0 or
    v_1 < s_1 + n_1)
3 |  v_u ← v_u + 1
4 |  if p_u[v_u] ∉ H_u(p_0[s_0], p_1[s_1])
5 |  |  s_u ← v_u;  v_{1−u} ← s_{1−u}
6 |  u ← 1 − u
7 if s_0 ⩾ 2n_0 or s_1 ⩾ 2n_1
8 |  return "no solution"
9 return (s_0, s_1)
```

To summarize, we make the following assumptions when presenting the pseudocode of the two algorithms for each particular case of 1–4, respectively:

1. P_0 is counterclockwise, P_1 is clockwise, and $\mathcal{H}_0(a,b) = \mathcal{H}_1(a,b) = \mathrm{RHP}(a,b)$,
2. P_0 is clockwise, P_1 is counterclockwise, and $\mathcal{H}_0(a,b) = \mathcal{H}_1(a,b) = \mathrm{LHP}(a,b)$,
3. P_0 and P_1 are clockwise, $\mathcal{H}_0(a,b) = \mathrm{RHP}(a,b)$, and $\mathcal{H}_1(x,y) = \mathrm{LHP}(a,b)$,
4. P_0 and P_1 are counterclockwise, $\mathcal{H}_0(a,b) = \mathrm{LHP}(a,b)$, and $\mathcal{H}_1(a,b) = \mathrm{RHP}(a,b)$.

Algorithm 1 maintains a candidate solution (s_0, s_1) starting from $(s_0, s_1) = (0,0)$. At the beginning and after each update to (s_0, s_1), the algorithm traverses P_0 and P_1 in parallel with indices (v_0, v_1), starting from $(v_0, v_1) = (s_0, s_1)$ and advancing v_0 and v_1 alternately. The variable $u \in \{0,1\}$ determines the polygon P_u in which the traversal is advanced in the current iteration. If the test in line 4 of Algorithm 1 succeeds, that is, the corner $p_u[v_u]$ lies on the "wrong side" of the candidate line, then the algorithm updates the candidate solution by setting $s_u \leftarrow v_u$ and reverts v_{1-u} back to s_{1-u} in line 5. The algorithm returns (s_0, s_1) when both polygons have been entirely traversed with indices v_0 and v_1 without detecting any corner on the "wrong side" of the candidate line. This can happen only when $P_0 \subset \mathcal{H}_0(p_0[s_0], p_1[s_1])$ and $P_1 \subset \mathcal{H}_1(p_0[s_0], p_1[s_1])$, as required.

See Figure 2 for an example run of Algorithm 1 for the separating common tangent problem (case 4). The following theorem asserts that Algorithm 1 is correct for the separating common tangent problem and "partially correct" for the outer common tangent problem.

Theorem 2. *If Algorithm 1 is to solve the outer common tangent problem (case 1 or 2), then it returns the solution (s_0, s_1) if the convex hulls of P_0 and P_1 are disjoint and reports "no solution" if the convex hulls of P_0 and P_1 are nested. If Algorithm 1 is to solve the separating common tangent*

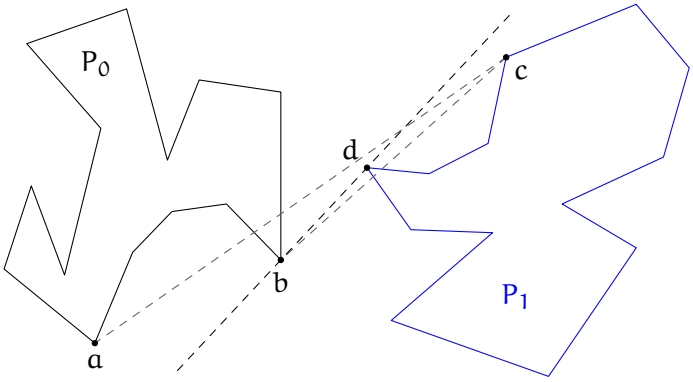


Figure 2: An example of how Algorithm 1 finds the separating common tangent $\mathcal{L}(b, d)$ of P_0 and P_1 starting from $(p_0[0], p_1[0]) = (a, c)$. The segments $p_0[s_0]p_1[s_1]$ on intermediate candidate lines are also shown.

Algorithm 2:

```
1  s_0 ← 0;  v_0 ← 0;  b_0 ← false;  s_1 ← 0;  v_1 ← 0;
     b_1 ← false;  u ← 0
2  while s_0 < 2n_0 and s_1 < 2n_1 and (v_0 < s_0 + n_0 or
     v_1 < s_1 + n_1)
3  |  v_u ← v_u + 1
4  |  if p_u[v_u] ∉ H_u(p_0[s_0], p_1[s_1]) and not b_u
5  |  |  if p_{1−u}[s_{1−u}] ∈ Δ(p_u[s_u], p_u[v_u − 1], p_u[v_u])
6  |  |  |  b_u ← true
7  |  |  else
8  |  |  |  s_u ← v_u;  v_{1−u} ← s_{1−u};  b_{1−u} ← false
9  |  u ← 1 − u
10 if s_0 ⩾ 2n_0 or s_1 ⩾ 2n_1 or b_0 or b_1
11 |  return "no solution"
12 return (s_0, s_1)
```

problem (case 3 or 4), then it returns the solution (s_0, s_1) *if the convex hulls of* P_0 *and* P_1 *are disjoint and reports "no solution" otherwise. Moreover, Algorithm 1 runs in linear time and uses constant workspace.*

If the convex hulls of P_0 and P_1 properly overlap, then Algorithm 1 can fail to find the solution even though it exists. An example of such behavior is presented in Figure 3. Algorithm 2 is an improved version of Algorithm 1 that solves the problem correctly in all cases including the case of properly overlapping convex hulls. In line 5 of Algorithm 2, $\Delta(a, b, c)$ denotes the triangular region spanned by a, b, and c, and a test of the form $z \in \Delta(a, b, c)$ is equivalent to testing whether $\det^\star(z, a, b)$, $\det^\star(z, b, c)$, and $\det^\star(z, c, a)$ are all positive or all negative (they are all non-zero, by the general position assumption). If the test in line 5 succeeds, then $p_{1-u}[s_{1-u}]$ belongs to the convex hull of P_u, and a special boolean variable b_u is set. In later iterations, when $b_k = \texttt{true}$, no update to s_k can occur in line 8 (with $u = k$) until b_k is cleared at an update to s_{1-k} in line 8 (with $u = 1 - k$). As we will

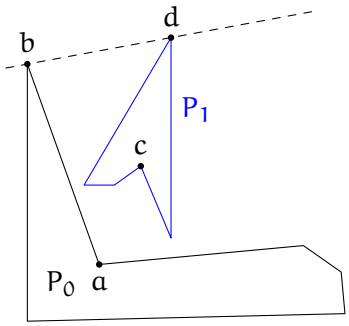


Figure 3: Two polygons P_0 and P_1 for which Algorithm 1 fails to find the outer common tangent $\mathcal{L}(b, d)$ starting from $(p_0[0], p_1[0]) = (a, c)$. If the conditions $s_0 < 2n_0$ and $s_1 < 2n_1$ of the "while" loop are ignored, the algorithm keeps updating (s_0, s_1) indefinitely, always getting back to the initial state where $u = 0$ and $(p_0[s_0], p_1[s_1]) = (p_0[v_0], p_1[v_1]) = (a, c)$.

show, such an update to s_{1-k} must occur unless the convex hull of P_{1-k} is contained in the convex hull of P_k, and preventing updates to s_k when $b_k = \texttt{true}$ suffices to guarantee correctness of the algorithm in all cases.

See Figure 4 for an example run of Algorithm 2 for the outer common tangent problem (case 1), where the convex hulls of P_0 and P_1 properly overlap. If the convex hulls of P_0 and P_1 are disjoint, then the test in line 5 of Algorithm 2 never succeeds, the variables b_0 and b_1 remain unset, and thus Algorithm 2 essentially becomes Algorithm 1.

Theorem 3. *If Algorithm 2 is to solve the outer common tangent problem (case 1 or 2), then it returns the solution (s_0, s_1) unless the convex hulls of P_0 and P_1 are nested, in which case it reports "no solution". If Algorithm 2 is to solve the separating common tangent problem (case 3 or 4), then it returns the solution (s_0, s_1) if the convex hulls of P_0 and P_1 are disjoint and reports "no solution" otherwise. Moreover, Algorithm 2 runs in linear time and uses constant workspace.*

2.4 CONCLUDING REMARKS

It remains open whether an outer common tangent of two polygons that are not disjoint can be found in linear time using constant workspace. There can be a linear number of tangents in that case, but the question is if finding an arbitrary one is possible.

Another natural question is whether the diameter of the convex hull of a simple polygon can be computed in linear time using constant workspace.[1]

1 The diameter of the convex hull is also the farthest pair of corners of the input polygon. The dual question, whether the closest pair of corners of the input polygon can be found in linear time, is already solved, since Aggarwal et al. [Agg+92] proved that finding the closest pair among n points takes $\Omega(n \log n)$ time in the algebraic decision tree model (with no restriction on the workspace), even when the sequence of points defines a simple polygon.

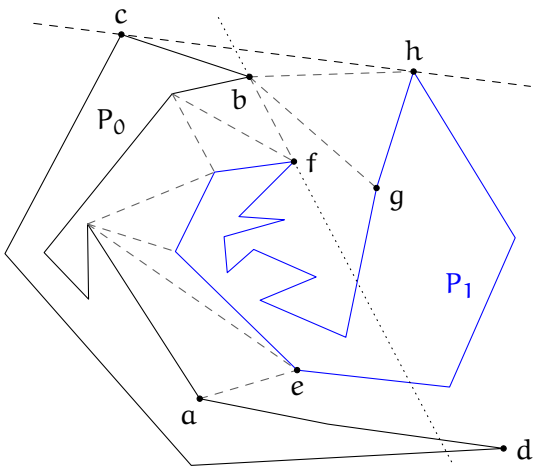


Figure 4: An example of how Algorithm 2 finds the outer common tangent $\mathcal{L}(c, h)$ of P_0 and P_1 starting from $(p_0[0], p_1[0]) = (a, e)$. The segments $p_0[s_0]p_1[s_1]$ on intermediate candidate lines are also shown. In the 11th iteration, an update makes $(p_0[s_0], p_1[s_1]) = (b, f)$ and the dotted line $\mathcal{L}(b, f)$ becomes the candidate line. In the 19th iteration, $u = 0$ and $p_0[v_0] = d$, so b_0 is set to `true`. In the 28th iteration, $u = 1$ and $p_0[v_1] = g$, so b_0 is set back to `false`. In the 31st iteration, an update makes $(p_0[s_0], p_1[s_1]) = (c, h)$, and the outer common tangent is found.

3

THE ART GALLERY PROBLEM

This section is based on the papers [AAM17a] and [AAM17b] (appendix B and C).

ABSTRACT One of the best-known problems in computational geometry is the art gallery problem, which was already studied in the early 70s. Many variations of the problem have been considered, but here we study the classical version where we are given a simple polygon with vertices at rational coordinates and we have to decide whether a given number of guards can be placed in the polygon so that they guard the entire polygon. We give an explicit example of a polygon where three guards are sufficient, but only if they are placed on specific points with irrational coordinates. If the coordinates of the guards are required to be rational, then four guards are needed. We furthermore prove the much more general result that the art gallery problem is complete for the complexity class $\exists\mathbb{R}$, implying that (1) the art gallery problem is equivalent up to polynomial time reductions to the problem of deciding whether a given system of polynomial equations and inequalities with integer coefficients and any number of variables has a solution, and (2) the art gallery problem is not in the complexity class NP unless $\mathsf{NP} = \exists\mathbb{R}$. As a corollary of our construction, we prove that for any real algebraic number α there is an instance of the art gallery problem where one of the coordinates of the guards equals α in any guard set of minimum cardinality. That rules out many geometric approaches to the problem.

3.1 INTRODUCTION

For a polygonal region P and points $x, y \in P$, we say that x *sees* y if the line segment xy is contained in P. A *guard set* S is a set of points in P such that every point in P is seen by some point in S. The points in S are called *guards*. The *art gallery problem* is to decide, given a polygonal region P with n corners and a positive integer k, whether there exists a guard set of size k. A guard set of minimum cardinality is called *optimal*.

This classical version of the art gallery problem has been originally formulated in 1973 by Victor Klee (see the book of O'Rourke [O'R87, page 2]). It is often referred to as the *interior-guard art gallery problem* or the *point-guard art gallery problem*, to distinguish it from other versions that have been introduced over the years.

The art gallery problem has been extensively studied, with some books, surveys, and book chapters dedicated to it [O'R87; She92; Urr00; O'R04; Ber+08; Mat02; DO11; O'R98]. The research is stimulated by a large number of possible variants of the problem and related questions that can be studied. Other versions of the art gallery

problem include restrictions on the positions of the guards, different definitions of visibility, restricted classes of polygonal regions, restricting the part of the polygonal region that has to be guarded, etc. In this chapter we only consider the point-guard art gallery problem.

Despite extensive research, no combinatorial algorithm for the problem problem is known. The only exact algorithm is attributed to Micha Sharir (see [EH06]), who gave an $n^{O(k)}$ time algorithm. This result is obtained using standard tools from real algebraic geometry [BPR06], and it is not known how to solve the problem without using this powerful machinery (see [Bel91] for an analysis of the very restricted case of $k = 2$).

Lee and Lin [LL86] proved, by constructing a reduction from 3SAT, that the art gallery problem is NP-hard when the guards are restricted to the corners of the polygonal region. It has subsequently been shown by Aggarwal ([Agg84], see also [O'R87]) that this argument can be extended to the case with no restrictions on the guards. It is a big open question whether the art gallery problem is in the complexity class NP. A simple way to show NP-membership would be to prove that there always exists an optimal set of guards with *rational* coordinates of polynomially bounded description. In the following section, we show that this is not the case.

3.2 IRRATIONAL GUARDS

Sándor Fekete posed at MIT in 2010 and at Dagstuhl in 2011 an open problem, asking whether there are polygonal regions requiring irrational coordinates in an optimal guard set [Fek; AMTrc]. The question has been raised again by Günter Rote at EuroCG 2011 [Rot11]. It has also been mentioned by Rezende et al. [Rez+16]: "it remains an open question whether there are polygons given by rational coordinates that require optimal guard positions with irrational coordinates". A similar question has been raised by Friedrichs et al. [Fri+16]: "[...] it is a long-standing open problem for the more general Art Gallery Problem (AGP): For the AGP it is not known whether the coordinates of an optimal guard cover can be represented with a polynomial number of bits".

In the paper [AAM17a], we answered the open question of Sándor Fekete by proving the following result. Recall that a polygonal region P is called *monotone* if there exists a line l such that the intersection between any line orthogonal to l and P is either empty or a single line segment.

Theorem 4. *There is a simple monotone polygonal region P with integer corner coordinates such that*

1. *P can be guarded by 3 guards, and*
2. *an optimal guard set of P with guards at points with rational coordinates has size 4.*

The polygonal region from Theorem 4 is shown in Figure 5. An interesting consequence of Theorem 4 is that there is no optimal guard

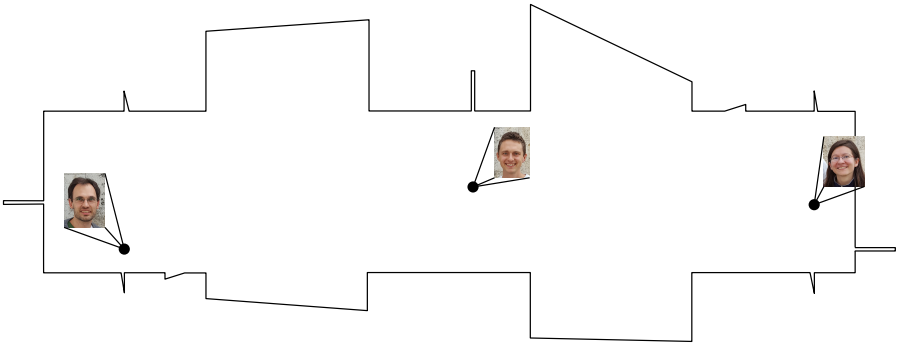


Figure 5: A polygonal region that can be guarded by three guards, but where four guards are needed if the coordinates of the guards are required to be rational. The unique optimal guard set is shown using the faces of the authors of [AAM17a].

set of P among a candidate set of guard positions consisting of intersections between extensions of chords and edges of P. It does not help to expand the candidate set by adding a line through each pair of candidates, thus creating new intersections to be added to the set of candidates, or to repeat this procedure any finite number of iterations, since all candidate points created by such a process must inevitably have rational coordinates. This shows that algorithms based on this procedure, as well as other algorithms for the art gallery problem which consider only rational points as possible guard positions, will in general not find an optimal guard set.

We then extended Theorem 4 by providing a family of polygonal regions for which the ratio between the size of an optimal rational guard set and the size of an optimal set with irrational guards allowed is $4/3$, see Figure 6. One can make an arbitrarily large example by connecting copies of the polygon P from Theorem 4 by thin corridors.

Theorem 5. *There is a family of polygononal regions* $(P_n)_{n\in\mathbb{Z}_+}$ *with integer corner coordinates such that*

1. P_n *can be guarded by* $3n$ *guards, and*
2. *an optimal guard set of* P_n *with guards at points with rational coordinates has size* $4n$.

Moreover, the coordinates of the points defining the polygonal regions P_n *are polynomial in* n.

We showed that the phenomenon with guards at irrational coordinates occurs already in the much simpler class of rectilinear polygons, i.e., polygononal regions where each edge is parallel to the x-axis or to the y-axis, see Figure 7.

Theorem 6. *There is a rectilinear polygonal region* P_R *with corners at integer coordinates satisfying the following properties.*

1. P_R *can be guarded by* 9 *guards.*
2. *An optimal guard set of* P_R *with guards at points with rational coordinates has size* 10.

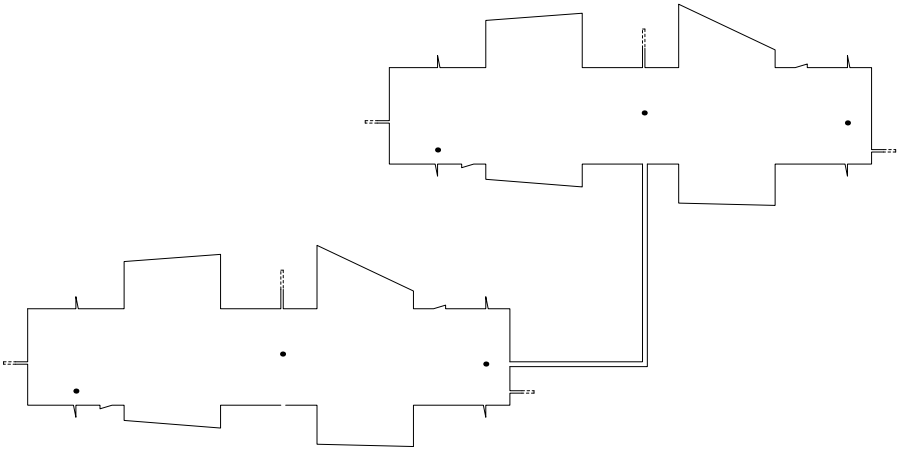


Figure 6: A polygonal region that can be guarded by six guards, but where eight guards are needed if the coordinates of the guards are required to be rational.

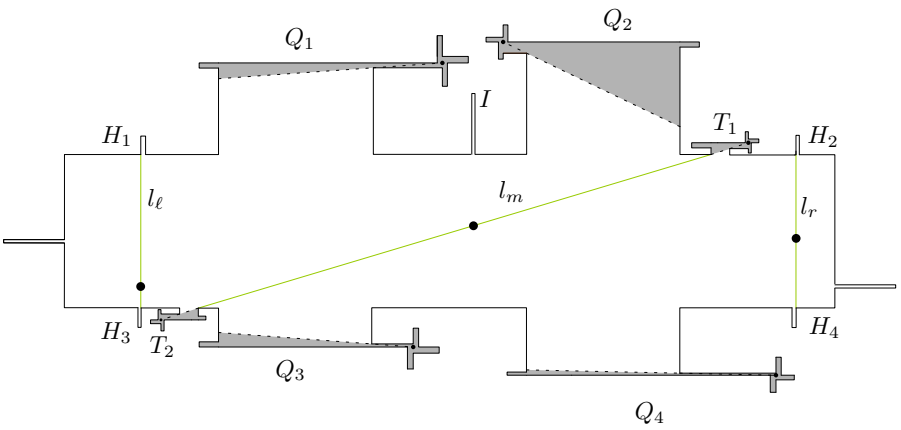


Figure 7: A polygonal region that can be guarded by nine guards, but where ten guards are needed if the coordinates of the guards are required to be rational. With nine guards, a guard has to be placed in each of the six grey regions, seeing exactly that region. The remaining white area is the polygonal region P from Theorem 4.

The optimal guard sets of the polygonal regions from Theorems 4–6 contain guards with coordinates of the form $(a_1 + a_2\sqrt{2}, a_3 + a_4\sqrt{2})$. Hence, each coordinate is a root of a second-degree polynomial with integer coefficients. It is thus natural to ask if there are also polygononal regions where coordinates with higher algebraic degree are needed in an optimal guard set. In the paper [AAM17b], we show that this is indeed the case, which is the topic of the next section.

3.3 $\exists\mathbb{R}$-COMPLETENESS

In the paper [AAM17b], we prove that the art gallery problem is equivalent under polynomial time reductions to deciding whether a system of polynomial equations over the real numbers has a solution.

3.3.1 *The complexity class* $\exists\mathbb{R}$

The *first order theory of the reals* is a set of all true sentences involving real variables, universal and existential quantifiers, boolean and arithmetic operators, constants 0 and 1, parenthesis, equalities and inequalities $(x_1, x_2, \ldots, \forall, \exists, \wedge, \vee, \neg, 0, 1, +, -, \cdot, \ (\ ,\ )\ , =, <, \leqslant)$. A formula is called a *sentence* if it has no free variables, i.e., each variable present in the formula is bound by a quantifier. Note that within such formulas one can easily express integer constants (using binary expansion) and powers. Each formula can be converted to a *prenex form*, i.e., a form where it starts with all the quantifiers and is followed by a quantifier-free formula, by a transformation which changes the length of the formula by at most a constant factor.

The *existential theory of the reals* is a set of all true sentences of the first-order theory of the reals in prenex form with existential quantifiers only, i.e., sentences of the form

$$(\exists X_1 \exists X_2 \ldots \exists X_k)\ \ \Phi(X_1, X_2, \ldots, X_k),$$

where Φ is a quantifier-free formula of the first-order theory of the reals with variables $X_1, \ldots, X_k$. The problem ETR is the problem of deciding whether a given existential formula of the above form is true. The complexity class $\exists\mathbb{R}$ consists of all problems that are reducible to ETR in polynomial time. The most well-known problem in the complexity class $\exists\mathbb{R}$ is deciding whether a system of polynomial equations over the real numbers has a solution.

It is currently known that

$$\mathsf{NP} \subseteq \exists\mathbb{R} \subseteq \mathsf{PSPACE}.$$

It is not hard see that the problem ETR is NP-hard, for instance by the following reduction from 3SAT. For each boolean variable x in an instance of 3SAT, we introduce a real variable x', and require that $x' \cdot (1 - x') = 0$ in order to ensure that $x' \in \{0, 1\}$. For any clause of the 3SAT formula we construct a function which evaluates to 1 if the corresponding clause is satisfied, and to 0 otherwise. For a clause C of the form $x \vee y \vee \neg z$, the corresponding function C' is $1 - (1 - x')(1 -$

$y')z'$. The conjunction of clauses $C_1 \wedge \ldots \wedge C_m$ is then translated to the equation $C'_1 \cdot \ldots \cdot C'_m - 1 = 0$. Clearly, a formula of 3SAT is true if and only if the constructed set of equations has a solution in $\mathbb{R}$. The containment $\exists\mathbb{R} \subseteq \mathsf{PSPACE}$ is highly non-trivial, and it has first been established by Canny [Can88].

By the reduction from 3SAT to ETR sketched above we know that a problem of deciding whether a given polynomial equation over $\{0, 1\}$ with integer coefficients has a solution is NP-hard. The problem is also in NP, as a satisfying assignment clearly serves as a witness. Therefore, NP-complete problems are the problems equivalent (under polynomial time reductions) to deciding whether a given polynomial equation over $\{0, 1\}$ with integer coefficients has a solution. A well-known $\exists\mathbb{R}$-complete problem is the problem of deciding whether a single polynomial equation $Q(x_1, \ldots, x_n) = 0$ with integer coefficients has a solution in $\mathbb{R}$ [Mat14, Proposition 3.2]. Therefore, the $\exists\mathbb{R}$-complete problems are equivalent to deciding whether a given polynomial equation over $\mathbb{R}$ with integer coefficients has a solution.

3.3.2 *Our results*

Theorem 7. *The art gallery problem is $\exists\mathbb{R}$-complete, even the restricted variant where we are given a polygonal region with corners at integer coordinates.*

In our construction, an ETR formula $(\exists X_1 \ldots \exists X_k)\ \Phi(X_1, \ldots, X_k)$ is transformed into an instance (P, g) of the art gallery problem where $g > k$. Let S_Φ denote the solution space of the formula Φ, i.e., $S_\Phi := \{x \in \mathbb{R}^k : \Phi(x)\}$. We will prove the following theorem.

Theorem 8. *Let Φ be an ETR formula with k variables. Then there is an instance (P, g) of the art gallery problem, and constants $c_1, d_1, \ldots, c_k, d_k \in \mathbb{Q}$, such that*

- *if Φ has a solution, then P has a guard set of size g, and*
- *for any guard set G of P of size g, there exists $(x_1, \ldots, x_k) \in S_\Phi$ such that G contains guards at positions $(c_1x_1 + d_1, 0), \ldots, (c_kx_k + d_k, 0)$.*

We get the following corollary.

Corollary 9. *Given any real algebraic number α, there exists a polygonal region P with corners at rational coordinates such that in any optimal guard set of P there is a guard with an x-coordinate equal to α.*

It is a classical result in Galois theory, and has thus been known since the 19th century, that there are polynomial equations of degree five with integer coefficients which have real solutions, but with no solutions expressible by radicals (i.e., solutions that can be expressed using integers, addition, subtraction, multiplication, division, raising to integer powers, and the extraction of n'th roots). One such example is the equation $x^5 - x + 1 = 0$ [Qui17]. It is a peculiar fact that using our reduction, we are able to transform such an equation into

an instance of the art gallery problem where no optimal guard set can be expressed by radicals. Note that our result also rules out any algorithm that only considers guards with coordinates of bounded algebraic degree.

3.3.3 *Other ∃ℝ-complete problems*

A growing class of problems turn out to be equivalent (under polynomial time reductions) to deciding whether polynomial equations and inequalities over the reals have a solution. These problems form the family of ∃ℝ-complete problems as it is currently known. This class includes problems like the stretchability of pseudoline arrangements [Mnë88; Sho91], recognition of intersection graphs of various objects (e.g. segments [Mat14], unit disks [MM13], and general convex sets [Sch09]), recognition of point visibility graphs [CH17], the Steinitz problem for 4-polytopes [RGZ95], deciding whether a graph with given edge lengths can be realized by a straight-line drawing [Sch13; Abe+16], deciding whether a straight line drawing of a graph exists with a given number of edge crossings [Bie91], decision problems related to Nash-equilibria [Gar+15], and positive semidefinite matrix factorization [Shi16]. We refer the reader to the lecture notes by Matoušek [Mat14] and surveys by Schaefer [Sch09] and Cardinal [Car15] for more information on the complexity class ∃ℝ.

3.3.4 *Overall structure of the reductions*

We first show that the art gallery problem is in the complexity class ∃ℝ. For that we present a construction of an ETR-formula Φ for any instance (P, g) of the art gallery problem such that Φ has a solution if and only if P has a guard set of size g. The idea is to encode guards by pairs of variables and compute a set of witnesses (which depend on the positions of the guards) of polynomial size such that the polygonal region is guarded if and only if the witnesses are seen by the guards.

The proof that the art gallery problem is ∃ℝ-hard is the main result of the paper [AAM17b], and it consists of two parts. The first part is of algebraic nature, and in that we introduce a novel ∃ℝ-complete problem which we call ETR-INV.

Definition 10. *In the problem ETR-INV, we are given a set of real variables* $\{x_1, \dots, x_n\}$*, and a set of equations of the form*

$$x = 1, \quad x + y = z, \quad x \cdot y = 1,$$

for $x, y, z \in \{x_1, \dots, x_n\}$*. The goal is to decide whether the system of equations has a solution when each variable is restricted to the range* $[1/2, 2]$*.*

Theorem 11. *The problem ETR-INV is ∃ℝ-complete.*

A common way of making a reduction from ETR to some other problem is to build gadgets corresponding to each of the equations

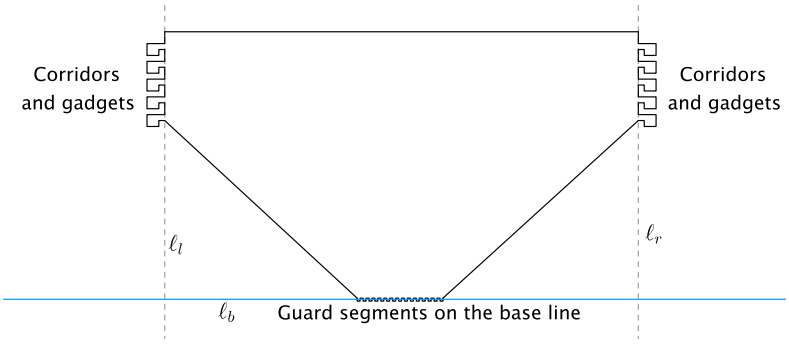


Figure 8: The overall design of an art gallery reduced from an instance of ETR-INV.

$x = 1$, $x + y = z$, and $x \cdot y = z$ for any variables x, y, z. Usually, the multiplication gadget is the most involved one. The reduction from ETR-INV requires building a gadget for *inversion* (i.e., $x \cdot y = 1$) instead of a more general gadget for multiplication. We think that the problem ETR-INV might be of independent interest, and that it will simplify constructing $\exists\mathbb{R}$-hardness proofs.

We then describe a polynomial time reduction from ETR-INV to the art gallery problem, which shows that the art gallery problem is $\exists\mathbb{R}$-hard. This reduction constructs an art gallery instance $(P(\Phi), g(\Phi))$ from an ETR-INV instance Φ, such that $P(\Phi)$ has a guard set of size $g(\Phi)$ if and only if the formula Φ has a solution. See Figure 8 for an overall picture of how $P(\Phi)$ looks. We construct the polygonal region so that it contains $g(\Phi)$ *guard segments* (which are horizontal line segments within P) and *stationary guard positions* (points within P). By introducing *pockets* we enforce that if P has a guard set of size $g(\Phi)$, then there must be exactly one guard at each guard segment and at each stationary guard position. Each guard segment represents a variable of Φ (with multiple segments representing the same variable) in the sense that the position of the guard on the segment specifies the value of the variable, the endpoints of a segment corresponding to the values $1/2$ and 2.

We develop a technique for *copying* guard segments, i.e., enforcing that the guards at two segments correspond to the same variable. We do that by introducing *critical segments* within the polygon, which can be seen by guards from two guard segments (but not from other guard segments). Then the requirement that a critical segment is seen introduces dependency between the guards at the corresponding segments. Different critical segments will enforce different dependencies, and by enforcing that two guards must see together two particular critical segments, we can ensure that the guards represent the same value. The stationary guards are placed to see the remaining areas of the polygon.

With this technique, we are able to copy two or three segments from an area containing guard segments corresponding to all variables into a *gadget*, where we will enforce a dependency between the values of the variables represented by the two or three segments. This is done

by constructing a *corridor* containing two critical segments for each pair of copied segments. The construction is non-trivial, as it requires the critical segments not to be seen from any other segments.

Within the gadgets, we build features that enforce the variables x, y, z represented by the guards to satisfy one of the conditions $x + y \geqslant z$, $x + y \leqslant z$, or $x \cdot y = 1$. The conditions are enforced by a requirement that two or three guards can see together some areas, where for the case of a gadget with three variables the area to be seen is a quadrilateral instead of a line segment.

3.4 CONCLUDING REMARKS

When reducing an instance of ETR-INV to an art gallery, we get many triplets of collinear corners. That is needed for the guard segments representing variables. In that sense the resulting art gallery is degenerate. It is an interesting open question whether the art gallery problem is also $\exists\mathbb{R}$-complete when the art gallery is assumed to have corners in general position. Other variants are interesting as well, for instance the version where only the boundary of the gallery has to be guarded, or when the guards have to be placed on the boundary, or the combination of these.

4

THE MINIMUM PERIMETER SUM PROBLEM

This chapter is based on the paper [Abr+17b] (appendix D).

ABSTRACT A natural clustering problem for points in the plane, which has been studied since the early 90s, is the minimum perimeter sum problem. Here, we are given n points and we want to find the way to partition the points into some number k of clusters so that the sum of perimeters of the convex hulls of the clusters is minimum. For the special case of $k = 2$, the fastest previously known algorithm had quadratic running time and we provide an $O(n \log^4 n)$ time algorithm.

4.1 INTRODUCTION

The clustering problem is to partition a given data set into clusters (that is, subsets) according to some measure of optimality. We are interested in clustering problems where the data set is a set P of points in Euclidean space. Most of these clustering problems fall into one of two categories: problems where the maximum cost of a cluster is given and the goal is to find a clustering consisting of a minimum number of clusters, and problems where the number of clusters is given and the goal is to find a clustering of minimum total cost. In this chapter we consider a basic problem of the latter type, where we wish to find a bipartition (P_1, P_2) of a planar point set P. Bipartition problems are not only interesting in their own right, but also because bipartition algorithms can form the basis of hierarchical clustering methods.

There are many possible variants of the bipartition problem on planar point sets, which differ in how the cost of a clustering is defined. A variant that received a lot of attention is the 2-center problem [Cha99; Dre84; Epp97; JK94; Sha97], where the cost of a partition (P_1, P_2) of the given point set P is defined as the maximum of the radii of the smallest enclosing disks of P_1 and P_2. Other cost functions that have been studied include the maximum diameter of the two point sets [Asa+88] and the sum of the diameters [Her92]; see also the survey by Agarwal and Sharir [AS98] for some more variants.

A natural class of cost function considers the size of the convex hulls $\mathrm{ch}(P_1)$ and $\mathrm{ch}(P_2)$ of the two subsets, where the size of $\mathrm{ch}(P_i)$ can either be defined as the area of $\mathrm{ch}(P_i)$ or as the perimeter $\mathrm{per}(P_i)$ of $\mathrm{ch}(P_i)$. (The perimeter of $\mathrm{ch}(P_i)$ is the length of the boundary $\partial\,\mathrm{ch}(P_i)$.) This class of cost functions was already studied in 1991 by Mitchell and Wynters [MW91]. They studied four problem variants: minimize the sum of the perimeters, the maximum of

the perimeters, the sum of the areas, or the maximum of the areas. In three of the four variants the convex hulls $\mathrm{ch}(P_1)$ and $\mathrm{ch}(P_2)$ in an optimal solution may intersect [MW91, full version]—only in the *minimum perimeter-sum problem* the optimal bipartition is guaranteed to be a so-called *line partition*, that is, a solution with disjoint convex hulls. For each of the four variants they gave an $O(n^3)$ algorithm that uses $O(n)$ space and for all except the minimum-maximum area problem, they also gave an $O(n^2)$ algorithm that uses $O(n^2)$ space; their algorithms only consider line partitions (which in the case of the minimum perimeter-sum problem implies an optimal bipartition). Around the same time, the minimum-perimeter sum problem was studied for partitions into k subsets for $k > 2$; for this variant Capoyleas et al. [CRW91] presented an algorithm with running time $O(n^{6k})$. Mitchell and Wynters mentioned the improvement of the space requirement of the quadratic-time algorithm as an open problem, and they stated the existence of a subquadratic algorithm for any of the four variants as the most prominent open problem.

Rokne et al. [RWW92] made progress on the first question, by presenting an $O(n^2 \log n)$ algorithm that uses only $O(n)$ space for the line-partition version of each of the four problems. Devillers and Katz [DK99] gave algorithms for the min-max variant of the problem, both for area and perimeter, which run in $O((n+k)\log^2 n)$ time. Here k is a parameter that is only known to be in $O(n^2)$, although Devillers and Katz suspected that k is subquadratic. They also gave linear-time algorithms for these problems when the point set P is in convex position and given in cyclic order. Segal [Seg02] proved an $\Omega(n \log n)$ lower bound for the min-max problems. Very recently, and apparently unaware of the earlier work on these problems, Cho et al. [Cho+16] presented an $O(n^2 \log^2 n)$ time algorithm for the minimum-perimeter-sum problem and an $O(n^4 \log^2 n)$ time algorithm for the minimum-area-sum problem (considering all partitions, not only line partitions). Despite these efforts, the main question remained open: is it possible to obtain a subquadratic algorithm for any of the four bipartition problems based on convex-hull size?

4.2 OUR CONTRIBUTION

We answer the question above affirmatively by presenting a subquadratic algorithm for the minimum perimeter-sum bipartition problem in the plane.

As mentioned, an optimal solution (P_1, P_2) to the minimum perimeter-sum bipartition problem must be a line partition. A straightforward algorithm would generate all $\Theta(n^2)$ line partitions and compute the value $\mathrm{per}(P_1) + \mathrm{per}(P_2)$ for each of them. If the latter is done from scratch for each partition, the resulting algorithm runs in $O(n^3 \log n)$ time. The algorithms by Mitchell and Wynters [MW91] and Rokne et al. [RWW92] improve on this by using that the different line bipartitions can be generated in an ordered way, such that subsequent line partitions differ in at most one point. Thus the convex hulls do not

have to be recomputed from scratch, but they can be obtained by updating the convex hulls of the previous bipartition. To obtain a subquadratic algorithm a fundamentally new approach is necessary: we need a strategy that generates a subquadratic number of candidate partitions, instead considering all line partitions. We achieve this as follows.

We start by proving that an optimal bipartition (P_1, P_2) satisfies the following *separation property*:

Theorem 12. *There is a set of* $O(1)$ canonical orientations[1] *such that for any set of points in the plane, the optimal bipartition* (P_1, P_2) *for the minimum-perimeter sum problem satisfies one of the following two conditions:*

- P_1 *can be separated from* P_2 *by a line with a canonical orientation, or*
- *the distance between* $ch(P_1)$ *and* $ch(P_2)$ *is proportional to* $\min(\mathrm{per}(P_1), \mathrm{per}(P_2))$.

There are only $O(n)$ bipartitions of the former type, and finding the best among them is relatively easy. The bipartitions of the second type are much more challenging. We show how to employ a compressed quadtree to generate a collection of $O(n)$ canonical 5-gons—intersections of axis-parallel rectangles and canonical halfplanes—such that the smaller of $\mathrm{ch}(P_1)$ and $\mathrm{ch}(P_2)$ (in a bipartition of the second type) is contained in one of the 5-gons.

It then remains to find the best among the bipartitions of the second type. Even though the number of such bipartitions is linear, we cannot afford to compute their perimeters from scratch. We therefore design a data structure to quickly compute $\mathrm{per}(P \cap Q)$, where Q is a query canonical 5-gon and obtain the following result:

Theorem 13. *Let* P *be a set of* n *points in the plane. Then we can compute a partition* (P_1, P_2) *of* P *that minimizes* $\mathrm{per}(P_1) + \mathrm{per}(P_2)$ *in* $O(n \log^4 n)$ *time using* $O(n \log^3 n)$ *space.*

1 In fact, the seven orientations $\left\{ \begin{pmatrix} \cos \frac{i\pi}{7} \\ \sin \frac{i\pi}{7} \end{pmatrix} \mid i = 0, \ldots, 6 \right\}$ will do.

BIBLIOGRAPHY

[Abe+16] Z. Abel, E.D. Demaine, M.L. Demaine, S. Eisenstat, J. Lynch, and T.B. Schardl. "Who Needs Crossings? Hardness of Plane Graph Rigidity." In: *32nd International Symposium on Computational Geometry (SoCG 2016)*. 2016, 3:1–3:15.

[Abr15a] M. Abrahamsen. "An Optimal Algorithm for the Separating Common Tangents of Two Polygons." In: *31st International Symposium on Computational Geometry (SoCG 2015)*. 2015, pp. 198–208.

[Abr15b] M. Abrahamsen. "Spiral Toolpaths for High-Speed Machining of 2D Pockets With or Without Islands." In: *ASME 2015 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*. 2015.

[AAM17a] M. Abrahamsen, A. Adamaszek, and T. Miltzow. "Irrational Guards are Sometimes Needed." In: *33rd International Symposium on Computational Geometry (SoCG 2017)*. 2017.

[AAM17b] M. Abrahamsen, A. Adamaszek, and T. Miltzow. *The Art Gallery Problem is* $\exists\mathbf{R}$*-complete*. Preprint, https://arxiv.org/abs/1704.06969. 2017.

[AT16] M. Abrahamsen and M. Thorup. "Finding the Maximum Subset with Bounded Convex Curvature." In: *32nd International Symposium on Computational Geometry (SoCG 2016)*. 2016, 4:1–4:17.

[AW16] M. Abrahamsen and B. Walczak. "Outer Common Tangents and Nesting of Convex Hulls in Linear Time and Constant Workspace." In: *24th Annual European Symposium on Algorithms (ESA 2016)*. 2016, 4:1–4:15.

[AW17] M. Abrahamsen and B. Walczak. *Common Tangents of Two Simple Polygons in Linear Time and Constant Workspace*. Manuscript. 2017.

[Abr+16] M. Abrahamsen, G. Bodwin, E. Rotenberg, and M. Stöckel. "Graph Reconstruction with a Betweenness Oracle." In: *33rd Symposium on Theoretical Aspects of Computer Science (STACS 2016)*. 2016, 5:1–5:14.

[Abr+17a] M. Abrahamsen, J. Holm, E. Rotenberg, and C. Wulff-Nilsen. "Best Laid Plans of Lions and Men." In: *33rd International Symposium on Computational Geometry (SoCG 2017)*. 2017.

[Abr+17b] M. Abrahamsen, M. de Berg, K. Buchin, M. Mehr, and A.D. Mehrabi. "Minimum Perimeter-Sum Partitions in the Plane." In: *33rd International Symposium on Computational Geometry (SoCG 2017)*. 2017.

[Abr+17c] M. Abrahamsen, S. Alstrup, J. Holm, M.B.T. Knudsen, and M. Stöckel. "Near-Optimal Induced Universal Graphs for Bounded Degree Graphs." In: *The 44th International Colloquium on Automata, Languages, and Programming (ICALP 2017)*. 2017.

[Abr+17d] M. Abrahamsen, S. Alstrup, J. Holm, M.B.T. Knudsen, and M. Stöckel. *Near-Optimal Induced Universal Graphs for Cycles and Paths*. In submission. 2017.

[Abr+17e] M. Abrahamsen, M. de Berg, K. Buchin, M. Mehr, and A.D. Mehrabi. "Range-Clustering Queries." In: *33rd International Symposium on Computational Geometry (SoCG 2017)*. 2017.

[AMTrc] P.K. Agarwal, K. Mehlhorn, and M. Teillaud. *Dagstuhl Seminar 11111, Computational Geometry*. March 13 – 18 , 2011.

[AS98] P.K. Agarwal and M. Sharir. "Efficient algorithms for geometric optimization." In: *ACM Comput. Surv.* 30 (4 1998), pp. 412–458.

[Agg84] A. Aggarwal. "The art gallery theorem: its variations, applications and algorithmic aspects." PhD thesis. 1984.

[Agg+92] A. Aggarwal, H. Edelsbrunner, P. Raghavan, and P. Tiwari. "Optimal time bounds for some proximity problems in the plane." In: *Information Processing Letters* 42.1 (1992), pp. 55–60.

[Asa+88] T. Asano, B. Bhattacharya, M. Keil, and F. Yao. "Clustering algorithms based on minimum and maximum spanning trees." In: *4th International Symposium on Computational Geometry (SoCG 1988)*. 1988, pp. 252–257.

[Asa+11] T. Asano, W. Mulzer, G. Rote, and Y. Wang. "Constant-work-space algorithms for geometric problems." In: *J. Comput. Geom.* 2.1 (2011), pp. 46–68.

[Asa+13] T. Asano, K. Buchin, M. Buchin, M. Korman, W. Mulzer, G. Rote, and A. Schulz. "Memory-constrained algorithms for simple polygons." In: *Comput. Geom.* 46.8 (2013), pp. 959–969.

[Bar+14] L. Barba, M. Korman, S. Langerman, and R.I. Silveira. "Computing the visibility polygon using few variables." In: *Comput. Geom.* 47.9 (2014), pp. 918–926.

[Bar+15] L. Barba, M. Korman, S. Langerman, K. Sadakane, and R.I. Silveira. "Space–time trade-offs for stack-based algorithms." In: *Algorithmica* 72.4 (2015), pp. 1097–1129.

[BPR06] S. Basu, R. Pollack, and M.-F. Roy. *Algorithms in real algebraic geometry*. Springer-Verlag Berlin Heidelberg, 2006.

[Bel91] P. Belleville. "Computing two-covers of simple polygons." MA thesis. McGill University, 1991.

[Ber+08] M. de Berg, M. van Kreveld, M. Overmars, and O. Cheong. *Computational Geometry: Algorithms and Applications (3rd edition)*. Springer-Verlag, 2008.

[Bha+12] D. Bhadauria, K. Klein, V. Isler, and S. Suri. "Capturing an evader in polygonal environments with obstacles: The full visibility case." In: *The International Journal of Robotics Research* 31.10 (2012), pp. 1176–1189.

[Bie91] D. Bienstock. "Some provably hard crossing number problems." In: *Discrete & Computational Geometry* 6.3 (1991), pp. 443–459.

[BJ02] G.S. Brodal and R. Jacob. "Dynamic planar convex hull." In: *43rd Annual IEEE Symposium on Foundations of Computer Science (FOCS 2002)*. 2002, pp. 617–626.

[Can88] J. Canny. "Some algebraic and geometric computations in PSPACE." In: *Proceedings of the twentieth annual ACM symposium on Theory of computing (STOC 1988)*. ACM. 1988, pp. 460–467.

[CRW91] V. Capoyleas, G. Rote, and G. Woeginger. "Geometric clusterings." In: *J. Alg.* 12 (2 1991), pp. 341–356.

[Car15] J. Cardinal. "Computational Geometry Column 62." In: *SIGACT News* 46.4 (2015), pp. 69–78.

[CH17] J. Cardinal and U. Hoffmann. "Recognition and complexity of point visibility graphs." In: *Discrete & Computational Geometry* 57.1 (2017), pp. 164–178.

[Car+16] E. Carson, J. Demmel, L. Grigori, N. Knight, P. Koanantakool, O. Schwartz, and H.V. Simhadri. "Write-avoiding algorithms." In: *30th IEEE International Parallel and Distributed Processing Symposium (IPDPS 2016)*. 2016, pp. 648–658.

[Cha99] T.M. Chan. "More planar two-center algorithms." In: *Comput. Geom. Theory Appl.* 13 (2 1999), pp. 189–198.

[Cho+16] H.G. Cho, W. Evans, N. Saeedi, and C.S. Shin. "Covering points with convex sets of minimum size." In: *10th Int. Workshop Alg. Comput. (WALCOM 2016)*. 2016, pp. 166–178.

[DO11] S.L. Devadoss and J. O'Rourke. *Discrete and Computational Geometry*. Princeton University Press, 2011.

[DK99] O. Devillers and M.J. Katz. "Optimal line bipartitions of point sets." In: *Int. J. Comput. Geom. Appl.* 9 (1 1999), pp. 39–51.

[Dre84] Z. Drezner. "The planar two-center and two-median problems." In: *Transportation Science* 18 (4 1984), pp. 351–361.

[EH06] A. Efrat and S. Har-Peled. "Guarding galleries and terrains." In: *Information Processing Letters* 100.6 (2006), pp. 238–245.

[Epp97] D. Eppstein. "Faster construction of planar two-centers." In: *8th ACM-SIAM Symp. Discr. Alg. (SODA 1997)*. 1997, pp. 131–138.

[Fek] S. Fekete. private communication.

[Fri+16] S. Friedrichs, M. Hemmer, J. King, and C. Schmidt. "The continuous 1.5D terrain guarding problem: Discretization, optimal solutions, and PTAS." In: *Journal of Computational Geometry* 7.1 (2016), pp. 256–284.

[Gar+15] J. Garg, R. Mehta, V.V. Vazirani, and S. Yazdanbod. "ETR-Completeness for Decision Versions of Multi-player (Symmetric) Nash Equilibria." In: *Proceedings of the 42nd International Colloquium on Automata, Languages, and Programming (ICALP 2015), part 1*. Vol. 9134. Lecture Notes in Computer Science (LNCS). 2015, pp. 554–566.

[GHS91] L. Guibas, J. Hershberger, and J. Snoeyink. "Compact interval trees: a data structure for convex hulls." In: *Int. J. Comput. Geom. Appl.* 1.1 (1991), pp. 1–22.

[GH89] L.J. Guibas and J. Hershberger. "Optimal shortest path queries in a simple polygon." In: *Journal of Computer and System Sciences* 39.2 (1989), pp. 126–152.

[HP15] S. Har-Peled. "Shortest path in a polygon using sublinear space." In: *31st International Symposium on Computational Geometry (SoCG 2015)*. Vol. 34. LIPIcs. 2015, pp. 111–125.

[Her92] J. Hershberger. "Minimizing the sum of diameters efficiently." In: *Comput. Geom. Theory Appl.* 2 (2 1992), pp. 111–118.

[HS92] J. Hershberger and S. Suri. "Applications of a semi-dynamic convex hull algorithm." In: *BIT Numer. Math.* 32.2 (1992), pp. 249–267.

[HS95] J. Hershberger and S. Suri. "A pedestrian approach to ray shooting: Shoot a ray, take a walk." In: *Journal of Algorithms* 18.3 (1995), pp. 403–431.

[JK94] J.W. Jaromczyk and M. Kowaluk. "An efficient algorithm for the Euclidean two-center problem." In: 1994, pp. 303–311.

[KS95] D. Kirkpatrick and J. Snoeyink. "Computing common tangents without a separating line." In: *4th International Workshop on Algorithms and Data Structures (WADS 1995)*. Vol. 955. LNCS. Springer, 1995, pp. 183–193.

[LL86] D.T. Lee and A.K. Lin. "Computational complexity of art gallery problems." In: *IEEE Transactions on Information Theory* 32.2 (1986), pp. 276–282.

[Mato2] J. Matoušek. *Lectures on Discrete Geometry*. Vol. 212. Graduate Texts in Mathematics. Springer-Verlag New York, 2002.

[Mat14] J. Matoušek. *Intersection graphs of segments and* $\exists\mathbb{R}$. Tech. rep. Preprint, https://arxiv.org/abs/1406.2636. 2014.

[MM13] C. McDiarmid and T. Müller. "Integer realizations of disk and segment graphs." In: *Journal of Combinatorial Theory, Series B* 103.1 (2013), pp. 114–143.

[Mel87] A.A. Melkman. "On-line construction of the convex hull of a simple polyline." In: *Inform. Process. Lett.* 25.1 (1987), pp. 11–12.

[MW91] J.S.B. Mitchell and E.L. Wynters. "Finding optimal bipartitions of points and polygons." In: (1991), pp. 202–213.

[Mnë88] N.E. Mnëv. "The universality theorems on the classification problem of configuration varieties and convex polytopes varieties." In: *Topology and geometry – Rohlin seminar*. Ed. by O.Y. Viro. Springer-Verlag Berlin Heidelberg, 1988, pp. 527–543.

[O'R87] J. O'Rourke. *Art Gallery Theorems and Algorithms*. Oxford University Press, 1987.

[O'R98] J. O'Rourke. *Computational Geometry in C*. Cambridge University Press, 1998.

[O'R04] J. O'Rourke. "Visibility." In: *Handbook of Discrete and Computational Geometry*. Ed. by Jacob E. Goodman and Joseph O'Rourke. Second. Chapman & Hall/CRC, 2004. Chap. 28.

[OL81] M.H. Overmars and J. van Leeuwen. "Maintenance of configurations in the plane." In: *J. Comput. System Sci.* 23.2 (1981), pp. 166–204.

[PH77] F.P. Preparata and S.J. Hong. "Convex hulls of finite sets of points in two and three dimensions." In: *Commun. ACM* 20.2 (1977), pp. 87–93.

[PS85] F.P. Preparata and M.I. Shamos. *Computational Geometry – An Introduction*. Springer-Verlag, 1985.

[Qui17] Quintic function – Wikipedia, The Free Encyclopedia. [Online; accessed 14-March-2017]. 2017. URL: https://en.wikipedia.org/wiki/Quintic_function.

[Rei08] O. Reingold. "Undirected connectivity in log-space." In: *Journal of the ACM (JACM)* 55.4 (2008), pp. 1–24.

[Rez+16] P.J. de Rezende, C.C. de Souza, S. Friedrichs, M. Hemmer, A. Kröller, and D.C. Tozoni. "Engineering Art Galleries." In: *Algorithm Engineering – Selected Results and Surveys*. Ed. by L. Kliemann and P. Sanders. Springer International Publishing, 2016, pp. 379–417.

[RGZ95] J. Richter-Gebert and G.M. Ziegler. "Realization spaces of 4-polytopes are universal." In: *Bulletin of the American Mathematical Society* 32.4 (1995), pp. 403–412.

[RWW92] J. Rokne, S. Wang, and X. Wu. "Optimal bipartitions of point sets." In: *4th Canad. Conf. Comput. Geom. (CCCG 1992)*. 1992, pp. 11–16.

[Rot11] G. Rote. *EuroCG Open Problem Session*. See the personal webpage of Günter Rote: http://page.mi.fu-berlin.de/rote/Papers/slides/Open-Problem_artgallery-Morschach-EuroCG-2011.pdf. 2011.

[Sch09] M. Schaefer. "Complexity of some geometric and topological problems." In: *Proceedings of the 17th International Symposium on Graph Drawing (GD 2009)*. Vol. 5849. Lecture Notes in Computer Science (LNCS). Springer. 2009, pp. 334–344.

[Sch13] M. Schaefer. "Realizability of graphs and linkages." In: *Thirty Essays on Geometric Graph Theory*. Ed. by János Pach. Springer-Verlag New York, 2013. Chap. 23, pp. 461–482.

[Seg02] M. Segal. "Lower bounds for covering problems." In: *J. Math. Modelling Alg.* 1 (1 2002), pp. 17–29.

[Sha97] M. Sharir. "A near-linear algorithm for the planar 2-center problem." In: *Discr. Comput. Geom.* 18 (2 1997), pp. 125–134.

[She92] T.C. Shermer. "Recent results in art galleries." In: *Proceedings of the IEEE* 80.9 (1992), pp. 1384–1399.

[Shi16] Y. Shitov. *The complexity of positive semidefinite matrix factorization*. Tech. rep. Preprint, http://arxiv.org/abs/1606.09065. 2016.

[Sho91] P.W. Shor. "Stretchability of pseudolines is NP-hard." In: *Applied Geometry and Discrete Mathematics: The Victor Klee Festschrift*. Ed. by Peter Gritzmann and Bernd Sturmfels. Vol. 4. DIMACS – Series in Discrete Mathematics and Theoretical Computer Science. American Mathematical Society and Association for Computing Machinery, 1991, pp. 531–554.

[Tou83] G.T. Toussaint. "Solving geometric problems with the rotating calipers." In: *IEEE Mediterranean Electrotechnical Conference (MELECON 1983)*. 1983, A10.02/1–4.

[Urr00] J. Urrutia. "Art Gallery and Illumination Problems." In: *Handbook of Computational Geometry*. Ed. by J.-R. Sack and J. Urrutia. Elsevier, 2000. Chap. 22, pp. 973–1027.

Part II

APPENDIX

This appendix contains the papers described in the synopsis.

A

COMMON TANGENTS OF TWO DISJOINT POLYGONS IN LINEAR TIME AND CONSTANT WORKSPACE

Common Tangents of Two Disjoint Polygons in Linear Time and Constant Workspace*

Mikkel Abrahamsen† Bartosz Walczak‡

August 29, 2017

Abstract

We provide a remarkably simple algorithm to compute all (at most four) common tangents of two disjoint simple polygons. Given each polygon as a read-only array of its corners in cyclic order, the algorithm runs in linear time and constant workspace and is the first to achieve the two complexity bounds simultaneously. The set of common tangents provides basic information about the convex hulls of the polygons—whether they are nested, overlapping, or disjoint—and our algorithm thus also decides this relationship.

1 Introduction

A tangent of a polygon is a line touching the polygon such that all of the polygon lies on the same side of the line. We consider the problem of computing the common tangents of two disjoint polygons that are simple, that is, they have no self-intersections. The set of common tangents provides basic information about the convex hulls of the polygons, i.e., whether they are disjoint, overlapping, or nested. We call a common tangent outer if the two polygons lie on the same side of it and separating otherwise. Two disjoint polygons have two outer common tangents unless their convex hulls are nested, and if they are properly nested, then there is no outer common tangent. Two polygons have two separating common tangents unless their convex hulls overlap, and if they properly overlap, then there is no separating common tangent. See Figures 1–3. Common tangents arise in many different contexts, for instance in problems related to convex hulls [18], shortest paths [11], ray shooting [14], and clustering [2].

We provide a very simple algorithm to compute all at most four common tangents of two disjoint simple polygons. In view of the above, our algorithm also determines whether the two polygons have (properly) nested, (properly) overlapping, or disjoint convex hulls. Given each of the two polygons as a read-only array of its corners in cyclic order, our algorithm runs in linear time and uses seven variables each storing a boolean value or an index of a corner in one of the arrays. The algorithm is therefore asymptotically optimal with respect to time as well as workspace, and operates in the *constant workspace model* of computation.

The constant workspace model is a restricted version of the RAM model in which the input is read-only, the output is write-only, and only $O(\log n)$ additional bits of *workspace* (with both read and write access) are available, where n denotes the size of the input. Clearly, $\Omega(\log n)$ bits of workspace are necessary to solve any interesting computational problem, because that many bits are required to store an index of or a pointer to an entry in the input. Since blocks of $\Theta(\log n)$ bits are considered to form *words* in the memory, algorithms in the constant workspace model use $O(1)$ words of workspace, which explains the name of the model. The practical relevance of studying problems in the constant workspace model is increasing, as there are many

*Preliminary versions of this paper appeared at SoCG 2015 [1] and ESA 2016 [3].

†Department of Computer Science, University of Copenhagen, Denmark, miab@di.ku.dk. Partially supported by Mikkel Thorup's Advanced Grant from the Danish Council for Independent Research under the Sapere Aude research career programme.

‡Department of Theoretical Computer Science, Faculty of Mathematics and Computer Science, Jagiellonian University in Kraków, Polandwalczak@tcs.uj.edu.pl. Partially supported by National Science Center of Poland grant 2015/17/D/ST1/00585.

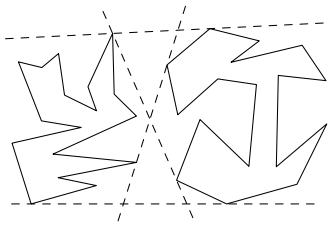


Figure 1: The convex hulls are disjoint—outer and separating common tangents exist.

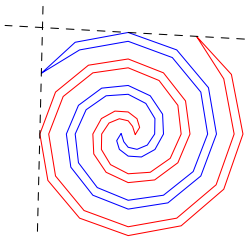


Figure 2: The convex hulls overlap—only outer common tangents exist.

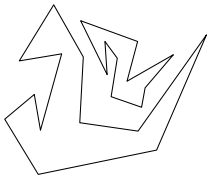


Figure 3: The convex hulls are nested—no common tangents exist.

current and emerging memory technologies where writing can be much more expensive than reading in terms of time and energy [9].

The constant workspace model was first considered explicitly for geometric problems by Asano et al. [5]. Recently, there has been growing interest in algorithms for geometric problems using constant or restricted workspace, see for instance [4,6,7,12]. In complexity theory, the class of the decision problems solvable using constant workspace is usually known as L. The constant workspace model has been shown to be surprisingly powerful—for instance, the problem of deciding whether two vertices in an undirected graph are in the same connected component is in L [19].

The problem of computing common tangents of two polygons has received much attention in the special case that the polygons are convex. For instance, computing the outer common tangents of disjoint convex polygons is used as a subroutine in the classical divide-and-conquer algorithm for the convex hull of a set of n points in the plane due to Preparata and Hong [18]. They gave a naive linear-time algorithm for outer common tangents, which suffices for an $O(n \log n)$-time convex hull algorithm. The problem is also considered in various dynamic convex hull algorithms [8, 13, 17]. Overmars and van Leeuwen [17] gave an $O(\log n)$-time algorithm for computing an outer common tangent of two disjoint convex polygons when a separating line is known, where each polygon has at most n corners. Kirkpatrick and Snoeyink [15] gave an $O(\log n)$-time algorithm for the same problem but without using a separating line. Guibas et al. [10] gave a lower bound of $\Omega(\log^2 n)$ on the time required to compute an outer common tangent of two intersecting convex polygons even when they are known to intersect in at most two points. They also described an algorithm achieving that bound. Toussaint [20] considered the problem of computing separating common tangents of convex polygons. He gave a linear-time algorithm using the technique of the "rotating calipers". Guibas et al. [10] gave an $O(\log n)$-time algorithm for the same problem. All the above-mentioned algorithms with sublinear running times make essential use of convexity of the polygons. If the polygons are not convex, a linear-time algorithm due to Melkman [16] can be used to compute the convex hulls before computing the tangents. However, if the polygons are given in read-only memory, then $\Omega(n)$ extra bits are required to store the convex hulls, so this approach does not work in the constant workspace model.

In the following we describe our algorithm, which is presented in full detail using pseudocode in Algorithm 2. (Algorithm 1 is a simplified version of Algorithm 2, which finds the separating common tangents in all cases, but is only guaranteed to find the outer common tangents when the convex hulls of the polygons are disjoint.) In order to find a particular common tangent of two polygons P_0 and P_1, we maintain a pair of corners of support q_0 and q_1, one in each polygon, that together define a candidate $\mathcal{L}(q_0, q_1)$ for the tangent (i.e., $\mathcal{L}(q_0, q_1)$ is the line containing q_0 and q_1). We traverse the polygons one edge at a time, alternating between the polygons. Each polygon is traversed from its corner of support in a cyclic order determined by the type of tangent that we wish to compute—the two separating common tangents correspond to traversing both polygons the same direction whereas the outer common tangents can be found by choosing different directions. We check that each traversed edge is on the correct side of $\mathcal{L}(q_0, q_1)$. When an edge e of, say, P_0 is found that ends at a corner q_0' on the wrong side (i.e., the edge e shows that $\mathcal{L}(q_0, q_1)$ is not the desired tangent), there are two cases. We consider the order of the following three points on $\mathcal{L}(q_0, q_1)$: q_0, q_1, and

the intersection point of e and $\mathcal{L}(q_0, q_1)$. If q_1 is not the middle point of these three, we update q_0 to q_0', thus also update the candidate line, and retract the traversal of P_1 to q_1. Otherwise, it can be seen that q_1 must be in the convex hull of P_0 and q_1 can therefore not be a support of the tangent we wish to find. In that case, we block q_0 from being updated and only care to traverse P_1 until an update to q_1 happens. If q_1 has not been updated after a full traversal of P_1, then the convex hulls are nested and no common tangents exist. Otherwise, we unblock q_0 when q_1 is updated. The correctness of the algorithm relies on the surprising fact that if the tangent exists, an update to a corner of support can only happen during the first or second traversal of the respective polygon. If an update happens in the third traversal of a polygon, we conclude that a common tangent of the desired type does not exist. Otherwise, we can after the third traversal of both polygons conclude that the candidate line $\mathcal{L}(q_0, q_1)$ coincides with the tangent that we wish to find.

The rest of the paper is structured as follows. In Section 2, we introduce the terminology and conventions used throughout the paper and state well-known facts about the common tangents of two polygons. In Section 3, we describe two algorithms for computing the common tangents of two disjoint simple polygons if they exist or detecting that they do not exist. One is a simplified version of the other and only works in some cases. Section 4 contains proofs that the algorithms work correctly under the assumption of a crucial lemma, which is then proved in Section 5. We conclude in Section 6 by discussing how to avoid the general position assumption of Sections 2–5 and by suggesting some related open problems for future research.

2 Basic terminology and notation

For any two points a and b in the plane, the closed line segment with endpoints a and b is denoted by ab. When $a \neq b$, the two-way infinite straight line passing through a and b is denoted by $\mathcal{L}(a, b)$. The segment ab and the line $\mathcal{L}(a, b)$ are considered *oriented* in the direction from a towards b. A *simple polygon* or just a *polygon* with *corners* $a_0, \ldots, a_{n-1}$, denoted by $\mathcal{P}(a_0, \ldots, a_{n-1})$, is a closed curve in the plane composed of n *edges* $a_0a_1, \ldots, a_{n-2}a_{n-1}, a_{n-1}a_0$ that have no common points other than the common endpoints of pairs of edges consecutive in that cyclic order. The polygon $\mathcal{P}(a_0, \ldots, a_{n-1})$ is considered *oriented* so that its *forward* traversal visits corners $a_0, \ldots, a_{n-1}$ in this cyclic order. A *polygonal region* is a closed and bounded region of the plane that is bounded by a polygon.

For any two points $a = (a_x, a_y)$ and $b = (b_x, b_y)$ in $\mathbb{R}^2$, we let

$$\det(a, b) = \begin{vmatrix} a_x & b_x \\ a_y & b_y \end{vmatrix} = a_x b_y - b_x a_y.$$

For $a_0, \ldots, a_{n-1} \in \mathbb{R}^2$, we let

$$\det^\star(a_0, \ldots, a_{n-1}) = \det(a_0, a_1) + \cdots + \det(a_{n-2}, a_{n-1}) + \det(a_{n-1}, a_0).$$

In particular, for any three points $a = (a_x, a_y)$, $b = (b_x, b_y)$, and $c = (c_x, c_y)$ in $\mathbb{R}^2$, we have

$$\det^\star(a, b, c) = \begin{vmatrix} a_x & b_x \\ a_y & b_y \end{vmatrix} + \begin{vmatrix} b_x & c_x \\ b_y & c_y \end{vmatrix} + \begin{vmatrix} c_x & a_x \\ c_y & a_y \end{vmatrix} = \begin{vmatrix} a_x & b_x & c_x \\ a_y & b_y & c_y \\ 1 & 1 & 1 \end{vmatrix}.$$

For two distinct points a and b in the plane, the *left side* and the *right side* of an oriented line $\mathcal{L}(a, b)$ are the two closed half-planes $\mathrm{LHP}(a, b)$ and $\mathrm{RHP}(a, b)$, respectively, defined as follows:

$$\mathrm{LHP}(a, b) = \{c \in \mathbb{R}^2 \colon \det^\star(a, b, c) \geq 0\},$$
$$\mathrm{RHP}(a, b) = \{c \in \mathbb{R}^2 \colon \det^\star(a, b, c) \leq 0\}.$$

An oriented polygon $\mathcal{P}(a_0, \ldots, a_{n-1})$ is *counterclockwise* when $\det^\star(a_0, \ldots, a_{n-1}) > 0$ and *clockwise* when $\det^\star(a_0, \ldots, a_{n-1}) < 0$.

We assume for the rest of this paper that P_0 and P_1 are two disjoint simple polygons with n_0 and n_1 corners, respectively, each defined by a read-only array of its corners:

$$P_0 = \mathcal{P}(p_0[0], \ldots, p_0[n_0 - 1]), \qquad P_1 = \mathcal{P}(p_1[0], \ldots, p_1[n_1 - 1]).$$

We make no assumption (yet) on whether P_0 and P_1 are oriented counterclockwise or clockwise. We further assume that the corners of P_0 and P_1 are in general position, that is, P_0 and P_1 have no corners in common and the combined set of corners $\{p_0[0], \ldots, p_0[n_0 - 1], p_1[0], \ldots, p_1[n_1 - 1]\}$ contains no triple of collinear points. This assumption simplifies the description and the analysis of the algorithm but can be avoided, as we explain in the last section. We do not assume the polygonal regions bounded by P_0 and P_1 to be disjoint—they may be nested. Indices of the corners of each P_k are considered modulo n_k, so that $p_k[i]$ and $p_k[j]$ denote the same corner when $i \equiv j \pmod{n_k}$.

A *tangent* of P_k is a line L such that P_k has a common point with L and is contained in one of the two closed half-planes determined by L. A line L is a *common tangent* of P_0 and P_1 if it is a tangent of both P_0 and P_1; it is an *outer common tangent* if P_0 and P_1 lie on the same side of L and a *separating common tangent* otherwise. The following lemma asserts well-known properties of common tangents of polygons. See Figures 1–3.

Lemma 2.1. *A line is a tangent of a polygon P if and only if it is a tangent of the convex hull of P. Moreover, under the general position assumption, the following holds:*

- *P_0 and P_1 have no common tangents if the convex hulls of P_0 and P_1 are nested;*
- *P_0 and P_1 have two outer common tangents and no separating common tangents if the convex hulls of P_0 and P_1 properly overlap;*
- *P_0 and P_1 have two outer common tangents and two separating common tangents if the convex hulls of P_0 and P_1 are disjoint.*

3 Algorithms

We distinguish four particular cases of the common tangent problem: find the pair of indices (s_0, s_1) such that

1. $P_0 \subset \mathrm{RHP}(p_0[s_0], p_1[s_1])$ and $P_1 \subset \mathrm{RHP}(p_0[s_0], p_1[s_1])$,
2. $P_0 \subset \mathrm{LHP}(p_0[s_0], p_1[s_1])$ and $P_1 \subset \mathrm{LHP}(p_0[s_0], p_1[s_1])$,
3. $P_0 \subset \mathrm{RHP}(p_0[s_0], p_1[s_1])$ and $P_1 \subset \mathrm{LHP}(p_0[s_0], p_1[s_1])$,
4. $P_0 \subset \mathrm{LHP}(p_0[s_0], p_1[s_1])$ and $P_1 \subset \mathrm{RHP}(p_0[s_0], p_1[s_1])$.

The line $\mathcal{L}(p_0[s_0], p_1[s_1])$ is an outer common tangent in cases 1–2 and a separating common tangent in cases 3–4. We say that (s_0, s_1) is the *solution* to the particular case of the problem. An algorithm solving each case 1–4 is expected to find and return the solution (s_0, s_1) if it exists (i.e. the convex hulls of P_0 and P_1 are not nested in cases 1–2 and are disjoint in cases 3–4) and to report "no solution" otherwise.

We will describe two general algorithms. Algorithm 1, very simple, fully solves the separating common tangent problem (cases 3–4), finding the separating common tangent if the convex hulls of P_0 and P_1 are disjoint and otherwise reporting that the requested tangent does not exist. Furthermore, Algorithm 1 solves the outer common tangent problem (cases 1–2) provided that the convex hulls of P_0 and P_1 are disjoint. Algorithm 1 also correctly reports that the outer common tangents do not exist if the convex hulls of P_0 and P_1 are nested. However, Algorithm 1 can fail to find the outer common tangents if the convex hulls of P_0 and P_1 properly overlap. Algorithm 2 is an improved version of Algorithm 1 that solves the problem correctly in all cases.

The general idea behind either algorithm is as follows. The algorithm maintains a pair of indices (s_0, s_1) called the *candidate solution*, which determines the line $\mathcal{L}(p_0[s_0], p_1[s_1])$ called the *candidate line*. If each of the two polygons lies on the "correct side" of the candidate line, which is either $\mathrm{RHP}(p_0[s_0], p_1[s_1])$ or $\mathrm{LHP}(p_0[s_0], p_1[s_1])$ depending on the particular case of 1–4 to be solved, then the algorithm returns (s_0, s_1) as the requested solution. Otherwise, for some $u \in \{0, 1\}$, the algorithm finds an index v_u such that $p_u[v_u]$ lies on the "wrong side" of the candidate line, updates s_u by setting $s_u \leftarrow v_u$, and repeats. This general scheme guarantees that if (s_0, s_1) is claimed to be the solution, then it indeed is. However, the algorithm can fall in an infinite loop—when there is no solution or when the existing solution keeps being missed. Detailed implementation of the scheme must guarantee that the solution is found in linearly many steps if it exists. Then, if the solution is not found in the guaranteed number of steps, the algorithm terminates and reports "no solution".

The particular case of 1–4 to be solved is specified to the algorithms by providing two binary parameters $\alpha_0, \alpha_1 \in \{+1, -1\}$ specifying that the final solution (s_0, s_1) should satisfy

$$\begin{aligned} P_0 &\subset \mathrm{RHP}(p_0[s_0], p_1[s_1]) \quad \text{if } \alpha_0 = +1, & \qquad P_0 &\subset \mathrm{LHP}(p_0[s_0], p_1[s_1]) \quad \text{if } \alpha_0 = -1, \\ P_1 &\subset \mathrm{RHP}(p_0[s_0], p_1[s_1]) \quad \text{if } \alpha_1 = +1, & \qquad P_1 &\subset \mathrm{LHP}(p_0[s_0], p_1[s_1]) \quad \text{if } \alpha_1 = -1. \end{aligned}$$

For clarity, instead of using the parameters α_0 and α_1 explicitly, the pseudocode uses half-planes $\mathcal{H}_0(a, b)$ and $\mathcal{H}_1(a, b)$ defined as follows, for any $k \in \{0, 1\}$ and any distinct $a, b \in \mathbb{R}^2$:

$$\mathcal{H}_k(a, b) = \big\{c \in \mathbb{R}^2 \colon \alpha_k \det^\star(a, b, c) \le 0\big\} = \begin{cases} \mathrm{RHP}(a, b) & \text{if } \alpha_k = +1, \\ \mathrm{LHP}(a, b) & \text{if } \alpha_k = -1. \end{cases}$$

Therefore, a test of the form $c \notin \mathcal{H}_k(a, b)$ in the pseudocode should be understood as testing whether $\alpha_k \det^\star(a, b, c) > 0$. Another assumption that we make when presenting the pseudocode concerns the direction in which each polygon P_k is traversed in order to find an index v_k such that $p_k[v_k] \notin \mathcal{H}_k(p_0[s_0], p_1[s_1])$. For a reason that will become clear later when we analyze correctness of the algorithms, we require that

- P_0 is traversed counterclockwise when $\alpha_1 = +1$ and clockwise when $\alpha_1 = -1$,
- P_1 is traversed clockwise when $\alpha_0 = +1$ and counterclockwise when $\alpha_0 = -1$.

In the pseudocode, the forward orientation of P_k is *assumed* to be the one in which the corners of P_k should be traversed according to the conditions above. When this has not been guaranteed in the problem setup, a reference to a corner of P_k of the form $p_k[i]$ in the pseudocode should be understood as $p_k[\beta_k i]$ for the constant $\beta_k \in \{+1, -1\}$ computed as follows at the very beginning:

$$\beta_0 = \alpha_1 \operatorname{sgn} \det^\star(p_0[0], \ldots, p_0[n_0 - 1]), \qquad \beta_1 = -\alpha_0 \operatorname{sgn} \det^\star(p_1[0], \ldots, p_1[n_1 - 1]).$$

To summarize, we make the following assumptions when presenting the pseudocode of the two algorithms for each particular case of 1–4, respectively:

1. P_0 is counterclockwise, P_1 is clockwise, and $\mathcal{H}_0(a, b) = \mathcal{H}_1(a, b) = \mathrm{RHP}(a, b)$,
2. P_0 is clockwise, P_1 is counterclockwise, and $\mathcal{H}_0(a, b) = \mathcal{H}_1(a, b) = \mathrm{LHP}(a, b)$,
3. P_0 and P_1 are clockwise, $\mathcal{H}_0(a, b) = \mathrm{RHP}(a, b)$, and $\mathcal{H}_1(x, y) = \mathrm{LHP}(a, b)$,
4. P_0 and P_1 are counterclockwise, $\mathcal{H}_0(a, b) = \mathrm{LHP}(a, b)$, and $\mathcal{H}_1(a, b) = \mathrm{RHP}(a, b)$.

Algorithm 1 maintains a candidate solution (s_0, s_1) starting from $(s_0, s_1) = (0, 0)$. At the beginning and after each update to (s_0, s_1), the algorithm traverses P_0 and P_1 in parallel with indices (v_0, v_1), starting from $(v_0, v_1) = (s_0, s_1)$ and advancing v_0 and v_1 alternately. The variable $u \in \{0, 1\}$ determines the polygon P_u in which the traversal is advanced in the current iteration. If the test in line 4 of Algorithm 1 succeeds, that is, the corner $p_u[v_u]$ lies on the "wrong side" of the candidate line, then the algorithm updates the candidate

Algorithm 1:

1 $s_0 \leftarrow 0$; $v_0 \leftarrow 0$; $s_1 \leftarrow 0$; $v_1 \leftarrow 0$; $u \leftarrow 0$
2 **while** $s_0 < 2n_0$ and $s_1 < 2n_1$ and $(v_0 < s_0 + n_0$ or $v_1 < s_1 + n_1)$
3 $\quad v_u \leftarrow v_u + 1$
4 $\quad$ **if** $p_u[v_u] \notin \mathcal{H}_u(p_0[s_0], p_1[s_1])$
5 $\quad\quad s_u \leftarrow v_u$; $v_{1-u} \leftarrow s_{1-u}$
6 $\quad u \leftarrow 1 - u$
7 **if** $s_0 \geq 2n_0$ or $s_1 \geq 2n_1$
8 $\quad$ **return** "no solution"
9 **return** (s_0, s_1)

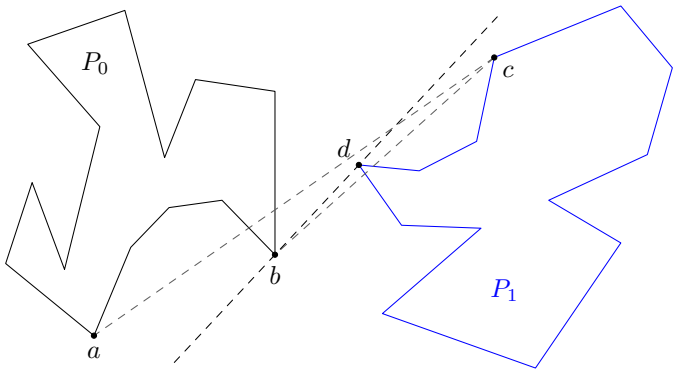


Figure 4: An example of how Algorithm 1 finds the separating common tangent $\mathcal{L}(b, d)$ of P_0 and P_1 starting from $(p_0[0], p_1[0]) = (a, c)$. The segments $p_0[s_0]p_1[s_1]$ on intermediate candidate lines are also shown.

solution by setting $s_u \leftarrow v_u$ and reverts v_{1-u} back to s_{1-u} in line 5. The algorithm returns (s_0, s_1) when both polygons have been entirely traversed with indices v_0 and v_1 without detecting any corner on the "wrong side" of the candidate line. This can happen only when $P_0 \subset \mathcal{H}_0(p_0[s_0], p_1[s_1])$ and $P_1 \subset \mathcal{H}_1(p_0[s_0], p_1[s_1])$, as required.

See Figure 4 for an example run of Algorithm 1 for the separating common tangent problem (case 4). The following theorem asserts that Algorithm 1 is correct for the separating common tangent problem and "partially correct" for the outer common tangent problem.

Theorem 3.1. *If Algorithm 1 is to solve the outer common tangent problem (case 1 or 2), then it returns the solution (s_0, s_1) if the convex hulls of P_0 and P_1 are disjoint and reports "no solution" if the convex hulls of P_0 and P_1 are nested. If Algorithm 1 is to solve the separating common tangent problem (case 3 or 4), then it returns the solution (s_0, s_1) if the convex hulls of P_0 and P_1 are disjoint and reports "no solution" otherwise. Moreover, Algorithm 1 runs in linear time and uses constant workspace.*

If the convex hulls of P_0 and P_1 properly overlap, then Algorithm 1 can fail to find the solution even though it exists. An example of such behavior is presented in Figure 5. Algorithm 2 is an improved version of Algorithm 1 that solves the problem correctly in all cases including the case of properly overlapping convex hulls. In line 5 of Algorithm 2, $\Delta(a, b, c)$ denotes the triangular region spanned by a, b, and c, and a test of the form $z \in \Delta(a, b, c)$ is equivalent to testing whether $\det^\star(z, a, b)$, $\det^\star(z, b, c)$, and $\det^\star(z, c, a)$ are all positive or all negative (they are all non-zero, by the general position assumption). If the test in line 5 succeeds, then $p_{1-u}[s_{1-u}]$ belongs to the convex hull of P_u, and a special boolean variable b_u is set. In later iterations, when $b_k =$ `true`, no update to s_k can occur in line 8 (with $u = k$) until b_k is cleared at an update to s_{1-k} in line 8 (with $u = 1 - k$). As we will show, such an update to s_{1-k} must occur unless the convex

Algorithm 2:

```
1  s_0 ← 0;  v_0 ← 0;  b_0 ← false;  s_1 ← 0;  v_1 ← 0;  b_1 ← false;  u ← 0
2  while s_0 < 2n_0 and s_1 < 2n_1 and (v_0 < s_0 + n_0 or v_1 < s_1 + n_1)
3      v_u ← v_u + 1
4      if p_u[v_u] ∉ H_u(p_0[s_0], p_1[s_1]) and not b_u
5          if p_{1−u}[s_{1−u}] ∈ Δ(p_u[s_u], p_u[v_u − 1], p_u[v_u])
6              b_u ← true
7          else
8              s_u ← v_u;  v_{1−u} ← s_{1−u};  b_{1−u} ← false
9      u ← 1 − u
10 if s_0 ≥ 2n_0 or s_1 ≥ 2n_1 or b_0 or b_1
11     return "no solution"
12 return (s_0, s_1)
```

hull of P_{1-k} is contained in the convex hull of P_k, and preventing updates to s_k when $b_k = \texttt{true}$ suffices to guarantee correctness of the algorithm in all cases.

See Figure 6 for an example run of Algorithm 2 for the outer common tangent problem (case 1), where the convex hulls of P_0 and P_1 properly overlap. If the convex hulls of P_0 and P_1 are disjoint, then the test in line 5 of Algorithm 2 never succeeds, the variables b_0 and b_1 remain unset, and thus Algorithm 2 essentially becomes Algorithm 1.

Theorem 3.2. *If Algorithm 2 is to solve the outer common tangent problem (case 1 or 2), then it returns the solution (s_0, s_1) unless the convex hulls of P_0 and P_1 are nested, in which case it reports "no solution". If Algorithm 2 is to solve the separating common tangent problem (case 3 or 4), then it returns the solution (s_0, s_1) if the convex hulls of P_0 and P_1 are disjoint and reports "no solution" otherwise. Moreover, Algorithm 2 runs in linear time and uses constant workspace.*

4 Correctness of Algorithm 1 and Algorithm 2

In this section, we prove Theorem 3.1 and Theorem 3.2 on correctness and efficiency of Algorithms 1 and 2 while leaving the proof of a key lemma to the next section. First, we prove the claims on running time and workspace usage in Theorems 3.1 and 3.2.

Lemma 4.1. *Algorithms 1 and 2 run in linear time and use constant workspace.*

Proof. It is clear that the algorithms use constant workspace. For the bound on the running time, we prove the following two claims:

1. Before and after every iteration of the "while" loop in Algorithm 1 or 2, we have $(v_u - s_u) - (v_{1-u} - s_{1-u}) \in \{-1, 0\}$.
2. In each iteration, the sum $s_0 + s_1 + v_0 + v_1$ is increased by at least 1.

Initially, we have $s_u = v_u = s_{1-u} = v_{1-u} = 0$, so statement 1 holds before the first iteration. Now, suppose that statement 1 holds before iteration i. After the assignment $v_u \leftarrow v_u + 1$, we have $(v_u - s_u) - (v_{1-u} - s_{1-u}) \in \{0, 1\}$, and the sum $s_0 + s_1 + v_0 + v_1$ has been increased by 1. The former implies that if the assignments $s_u \leftarrow v_u$ and $v_{1-u} \leftarrow s_{1-u}$ are performed in iteration i (in line 5 of Algorithm 1 or line 8 of Algorithm 2), then the sum $s_0 + s_1 + v_0 + v_1$ remains unchanged or is increased by 1 again, and we have $(v_u - s_u) - (v_{1-u} - s_{1-u}) = 0$ afterwards. In total, the sum $s_0 + s_1 + v_0 + v_1$ is increased by 1 or 2 in iteration i. Finally, after the assignment $u \leftarrow 1 - u$, we have $(v_u - s_u) - (v_{1-u} - s_{1-u}) \in \{-1, 0\}$, so statement 1 holds at the end of iteration i.

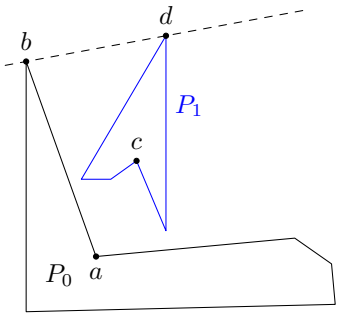


Figure 5: Two polygons P_0 and P_1 for which Algorithm 1 fails to find the outer common tangent $\mathcal{L}(b,d)$ starting from $(p_0[0],p_1[0]) = (a,c)$. If the conditions $s_0 < 2n_0$ and $s_1 < 2n_1$ of the "while" loop are ignored, the algorithm keeps updating (s_0,s_1) indefinitely, always getting back to the initial state where $u = 0$ and $(p_0[s_0],p_1[s_1]) = (p_0[v_0],p_1[v_1]) = (a,c)$.

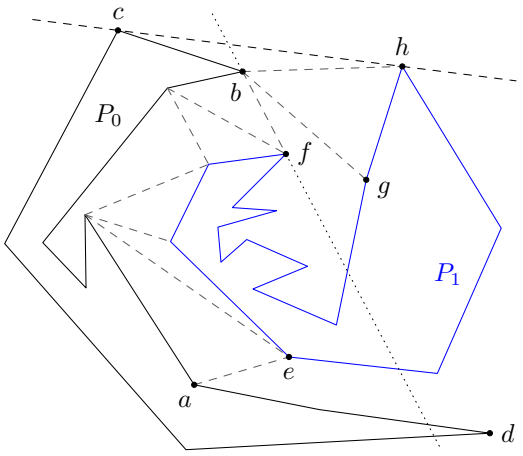


Figure 6: An example of how Algorithm 2 finds the outer common tangent $\mathcal{L}(c,h)$ of P_0 and P_1 starting from $(p_0[0],p_1[0]) = (a,e)$. The segments $p_0[s_0]p_1[s_1]$ on intermediate candidate lines are also shown. In the 11th iteration, an update makes $(p_0[s_0],p_1[s_1]) = (b,f)$ and the dotted line $\mathcal{L}(b,f)$ becomes the candidate line. In the 19th iteration, $u = 0$ and $p_0[v_0] = d$, so b_0 is set to `true`. In the 28th iteration, $u = 1$ and $p_0[v_1] = g$, so b_0 is set back to `false`. In the 31st iteration, an update makes $(p_0[s_0],p_1[s_1]) = (c,h)$, and the outer common tangent is found.

Statement 1 implies that each iteration starts with $s_0 < 2n_0$, $s_1 < 2n_1$, $v_0 - s_0 \le \max(n_0,n_1)$, and $v_1 - s_1 \le \max(n_0,n_1)$ (where at least one of the last two inequalities is strict), otherwise the "while" loop would terminate before that iteration. Therefore, we have $s_0 + s_1 + v_0 + v_1 \le 2(s_0 + s_1) + 2\max(n_0,n_1) < 6(n_0 + n_1)$ before each iteration fully performed by the algorithm, in particular the last one. This, the fact that $s_0 + s_1 + v_0 + v_1 = 0$ before the first iteration, and statement 2 imply that the algorithm makes at most $6(n_0 + n_1)$ iterations, each of which takes constant time. □

Let $k \in \{0,1\}$. We extend the notation $p_k[x]$ to all real numbers x to make the function $\mathbb{R} \ni x \mapsto p_k[x] \in P_k$ a continuous and piecewise linear traversal of P_k:

$$p_k[x] = (\lceil x \rceil - x)p_k[\lfloor x \rfloor] + (x - \lfloor x \rfloor)p_k[\lceil x \rceil] \in p_k[\lfloor x \rfloor]p_k[\lceil x \rceil], \quad \text{for } x \in \mathbb{R} \setminus \mathbb{Z}.$$

When $x, y \in \mathbb{R}$ and $x \le y$, we let $P_k[x,y]$ denote the part of P_k from $p_k[x]$ to $p_k[y]$ in the forward direction of P_k, that is, $P_k[x,y] = \{p_k[z] \colon z \in [x,y]\}$. We say that $P_k[x,y]$ is a *cap* of $\mathcal{H}_k(a,b)$ (for distinct $a, b \in \mathbb{R}^2$) if $P_k[x,y] \subset \mathcal{H}_k(a,b)$ and $P_k[x,y] \cap \mathcal{L}(a,b) = \{x,y\}$; this allows $x = y$.

Lemma 4.2. *For each $k \in \{0,1\}$, Algorithm 1 maintains the following invariant before and after every iteration of the "while" loop: $P_k[s_k,v_k] \subset \mathcal{H}_k(p_0[s_0],p_1[s_1])$.*

Proof. Algorithm 1 starts with $s_0 = v_0 = s_1 = v_1 = 0$, so the invariant holds initially. To show that it is preserved by every iteration, suppose it holds before iteration i. Let u and v_u denote the values in iteration i after the assignment $v_u \leftarrow v_u + 1$ and before the assignment $u \leftarrow 1 - u$. Suppose the test in line 4 of Algorithm 1 succeeds, that is, $p_u[v_u] \notin \mathcal{H}_u(p_0[s_0],p_1[s_1])$; otherwise, clearly, the invariant is preserved by iteration i. The updates in line 5 yield $s_0 = v_0$ and $s_1 = v_1$, which makes the invariant satisfied after iteration i. □

Lemma 4.3. *For each $k \in \{0,1\}$, Algorithm 2 maintains the following invariant before and after every iteration of the "while" loop:*

- *if* $b_k = \texttt{false}$*, then* $P_k[s_k, v_k] \subset \mathcal{H}_k(p_0[s_0], p_1[s_1])$*;*
- *if* $b_k = \texttt{true}$*, then there is* $w_k \in (s_k, v_k)$ *such that* $P_k[s_k, w_k]$ *is a cap of* $\mathcal{H}_k(p_0[s_0], p_1[s_1])$ *and* $p_{1-k}[s_{1-k}] \in p_k[s_k]p_k[w_k]$*.*

Moreover, on every update $s_u \leftarrow v_u$ *in line 8 of Algorithm 2, if* s_u *denotes the value before the update, then* $p_u[v_u] \notin \mathcal{H}_u(p_0[s_0], p_1[s_1])$ *and there is* $w_u \in [v_u - 1, v_u)$ *such that* $P_u[s_u, w_u]$ *is a cap of* $\mathcal{H}_u(p_0[s_0], p_1[s_1])$ *and* $p_{1-u}[s_{1-u}] \notin p_u[s_u]p_u[w_u]$*.*

Proof. Algorithm 2 starts with $s_0 = v_0 = s_1 = v_1 = 0$ and $b_0 = b_1 = \texttt{false}$, so the invariant holds initially. To show that it is preserved by every iteration, suppose it holds before iteration i. Let u and v_u denote the values in iteration i after the assignment $v_u \leftarrow v_u + 1$ and before the assignment $u \leftarrow 1 - u$. Suppose the test in line 4 of Algorithm 2 succeeds, that is, $p_u[v_u] \notin \mathcal{H}_u(p_0[s_0], p_1[s_1])$ and $b_u = \texttt{false}$; otherwise, clearly, the invariant is preserved by iteration i. This and the assumption that the invariant holds before iteration i imply $P_u[s_u, v_u - 1] \subset \mathcal{H}_u(p_0[s_0], p_1[s_1])$ and $P_u[s_u, v_u] \not\subset \mathcal{H}_u(p_0[s_0], p_1[s_1])$. Let $w_u \in [v_u - 1, v_u)$ be maximal such that $P_u[s_u, w_u] \subset \mathcal{H}_u(p_0[s_0], p_1[s_1])$. That is, $p_u[w_u]$ is the intersection point of $p_u[v_u - 1]p_u[v_u]$ and $\mathcal{L}(p_0[s_0], p_1[s_1])$. The general position assumption implies that $P_u[s_u, w_u]$ is a cap of $\mathcal{H}_u(p_0[s_0], p_1[s_1])$. We have $p_{1-u}[s_{1-u}] \in p_u[s_u]p_u[w_u]$ if and only if $p_{1-u}[s_{1-u}] \in \Delta(p_u[s_u], p_u[v_u - 1], p_u[v_u])$. Therefore, if the test in line 5 succeeds, then the assignment $b_u \leftarrow \texttt{true}$ in line 6 makes the invariant satisfied after iteration i. Now, suppose the test in line 5 fails. It follows that $p_{1-u}[s_{1-u}] \notin p_u[s_u]p_u[w_u]$, so the update $s_u \leftarrow v_u$ in line 8 satisfies the second statement of the lemma. Furthermore, the updates in line 8 yield $s_0 = v_0$, $s_1 = v_1$, and $b_{1-u} = \texttt{false}$, which makes the invariant satisfied after iteration i. □

Most effort in proving correctness of the two algorithms lies in the following two lemmas:

Lemma 4.4. *If the convex hulls of* P_0 *and* P_1 *are disjoint, then the "while" loop in Algorithm 1 ends with* $s_0 < 2n_0$ *and* $s_1 < 2n_1$*.*

Lemma 4.5. *If Algorithm 2 is to solve the outer common tangent problem (case 1 or 2) and the convex hulls of* P_0 *and* P_1 *are not nested or Algorithm 2 is to solve the separating common tangent problem (case 3 or 4) and the convex hulls of* P_0 *and* P_1 *are disjoint, then the "while" loop in Algorithm 2 ends with* $s_0 < 2n_0$ *and* $s_1 < 2n_1$*.*

If the convex hulls of P_0 and P_1 are disjoint, then the test in line 5 of Algorithm 2 never succeeds, b_1 and b_2 remain unset all the time, and thus Algorithm 2 becomes equivalent to Algorithm 1. Therefore, Lemma 4.4 is a direct consequence of Lemma 4.5. We prove Lemma 4.5 in the next section. Here, we proceed with the proofs of Theorems 3.1 and 3.2 assuming Lemma 4.5.

Proof of Theorem 3.1. Algorithm 1 returns (s_0, s_1) only when the "while" loop has terminated with $v_0 \geq s_0 + n_0$ and $v_1 \geq s_1 + n_1$. This and Lemma 4.2 imply that $P_0 \subset \mathcal{H}_0(p_0[s_0], p_1[s_1])$ and $P_1 \subset \mathcal{H}_1(p_0[s_0], p_1[s_1])$, that is, (s_0, s_1) is the correct solution. If Algorithm 1 is to solve the outer common tangent problem and the convex hulls of P_0 and P_1 are nested or Algorithm 2 is to solve the separating common tangent problem and the convex hulls of P_0 and P_1 are not disjoint, then the solution does not exist, so the algorithm reports "no solution", as explained above. By Lemma 4.4, if the convex hulls of P_0 and P_1 are disjoint, then the "while" loop ends with $s_0 < 2n_0$ and $s_1 < 2n_1$, so the algorithm returns (s_0, s_1), which is the correct solution, as explained above. By Lemma 4.1, the running time and the workspace usage are as stated. □

For the proof of correctness of Algorithm 2, we need one more lemma.

Lemma 4.6. *If Algorithm 2 is to solve the outer common tangent problem (case 1 or 2) and the convex hulls of* P_0 *and* P_1 *are not nested or Algorithm 2 is to solve the separating common tangent problem (case 3 or 4) and the convex hulls of* P_0 *and* P_1 *are disjoint, then the "while" loop in Algorithm 2 ends with* $b_0 = b_1 = \texttt{false}$*.*

Proof. Consider the final values of s_0, v_0, b_0, s_1, v_1, and b_1 when the "while" loop in Algorithm 2 is terminated. Lemma 4.5 yields $s_0 < 2n_0$ and $s_1 < 2n_1$. This and the termination condition implies $v_0 \geq s_0 + n_0$ and $v_1 \geq s_1 + n_1$. If the algorithm is to solve the separating common tangent problem (case 3 or 4) and we have $b_0 = \texttt{true}$ or $b_1 = \texttt{true}$, then Lemma 4.3 implies that the convex hulls of P_0 and P_1 are not disjoint, contrary to the assumption of the lemma. Now, suppose the algorithm is to solve the outer common tangent problem (case 1 or 2). Thus $\mathcal{H}_0(p_0[s_0], p_1[s_1]) = \mathcal{H}_1(p_0[s_0], p_1[s_1])$. If $b_k = \texttt{true}$ and $b_{1-k} = \texttt{false}$ for some $k \in \{0, 1\}$, then Lemma 4.3 and the fact that $P_{1-k}[s_{1-k}, v_{1-k}] = P_{1-k}$ yield a cap $P_k[s_k, w_k]$ of $\mathcal{H}_k(p_0[s_0], p_1[s_1])$ such that P_{1-k} is contained in the polygonal region bounded by $P_k[s_k, w_k] \cup p_k[s_k]p_k[w_k]$, and therefore the convex hull of P_{1-k} is contained in the convex hull of P_k, contrary to the assumption of the lemma. If $b_0 = b_1 = \texttt{true}$, then Lemma 4.3 yields a cap $P_0[s_0, w_0]$ of $\mathcal{H}_0(p_0[s_0], p_1[s_1])$ and a cap $P_1[s_1, w_1]$ of $\mathcal{H}_1(p_0[s_0], p_1[s_1])$ such that the points $p_1[w_1]$, $p_0[s_0]$, $p_1[s_1]$, and $p_0[w_0]$ occur in this order on $\mathcal{L}(p_0[s_0], p_1[s_1])$, which is impossible when $\mathcal{H}_0(p_0[s_0], p_1[s_1]) = \mathcal{H}_1(p_0[s_0], p_1[s_1])$. □

Proof of Theorem 3.2. Algorithm 2 returns (s_0, s_1) only when the "while" loop has terminated with $v_0 \geq s_0 + n_0$, $v_1 \geq s_1 + n_1$, and $b_0 = b_1 = \texttt{false}$. This and Lemma 4.3 imply that $P_0 \subset \mathcal{H}_0(p_0[s_0], p_1[s_1])$ and $P_1 \subset \mathcal{H}_1(p_0[s_0], p_1[s_1])$, that is, (s_0, s_1) is the correct solution. If Algorithm 2 is to solve the outer common tangent problem (case 1 or 2) and the convex hulls of P_0 and P_1 are nested or Algorithm 1 is to solve the separating common tangent problem (case 3 or 4) and the convex hulls of P_0 and P_1 are not disjoint, then the solution does not exist, so the algorithm reports "no solution", as explained above. Otherwise, the "while" loop ends with $s_0 < 2n_0$ and $s_1 < 2n_1$, by Lemma 4.5, and with $b_0 = b_1 = \texttt{false}$, by Lemma 4.6, and therefore the algorithm returns (s_0, s_1), which is the correct solution, as explained above. By Lemma 4.1, the running time and the workspace usage are as stated. □

5 Proof of Lemma 4.5

To complete the proof of correctness of the two algorithms, it remains to prove Lemma 4.5. If the convex hulls of P_0 and P_1 are disjoint, then the test in line 5 of Algorithm 2 can never succeed, so b_0 and b_1 remain unset, and Algorithm 2 becomes equivalent to Algorithm 1. Therefore, if we prove the second statement of Lemma 4.5 (for Algorithm 2), then the first statement of Lemma 4.5 comes as a direct consequence. For the rest of this section, we adopt the assumptions of Lemma 4.5, in particular that the solution exists, and we show that it is found and returned by Algorithm 2.

5.1 Reduction to one case of the outer common tangent problem

We will reduce Lemma 4.5 for all cases 1–4 of the common tangent problem just to case 1. First, we explain what we mean by such a reduction. An input to Algorithm 2 is a quadruple $(P_0, P_1, \alpha_0, \alpha_1)$, where $P_0 = \mathcal{P}(p_0[0], \ldots, p_0[n_0 - 1])$, $P_1 = \mathcal{P}(p_1[0], \ldots, p_1[n_1 - 1])$, and $\alpha_0, \alpha_1 \in \{+1, -1\}$ are implicit parameters that determine the particular case 1–4 of the common tangent problem to be solved by the algorithm (see Section 3). Consider two inputs $(P_0, P_1, \alpha_0, \alpha_1)$ and $(P'_0, P'_1, \alpha'_0, \alpha'_1)$, where $P_k = \mathcal{P}(p_k[0], \ldots, p_k[n_k - 1])$ and $P'_k = \mathcal{P}(p'_k[0], \ldots, p'_k[n_k - 1])$ for each $k \in \{0, 1\}$. Let $a' = p'_k[i]$ when $a = p_k[i]$ (for $k \in \{0, 1\}$ and $i \in \mathbb{Z}$). The inputs $(P_0, P_1, \alpha_0, \alpha_1)$ and $(P'_0, P'_1, \alpha'_0, \alpha'_1)$ are *equivalent* if the following holds for all $a, b, c \in \{p_0[0], \ldots, p_0[n_0 - 1], p_1[0], \ldots, p_1[n_1 - 1]\}$:

$$\begin{aligned}
&\alpha_0 \operatorname{sgn}\det^\star(a, b, c) = \alpha'_0 \operatorname{sgn}\det^\star(a', b', c') \quad \text{if } |\{a, b, c\} \cap \{p_0[0], \ldots, p_0[n_0 - 1]\}| \in \{0, 2\},\\
&\alpha_1 \operatorname{sgn}\det^\star(a, b, c) = \alpha'_1 \operatorname{sgn}\det^\star(a', b', c') \quad \text{if } |\{a, b, c\} \cap \{p_0[0], \ldots, p_0[n_0 - 1]\}| \in \{1, 3\}.
\end{aligned}$$

In view of the next lemma, if Lemma 4.5 holds for some input $(P_0, P_1, \alpha_0, \alpha_1)$, then it holds for all inputs equivalent to $(P_0, P_1, \alpha_0, \alpha_1)$.

Lemma 5.1. *If inputs $(P_0, P_1, \alpha_0, \alpha_1)$ and $(P'_0, P'_1, \alpha'_0, \alpha'_1)$ are equivalent, then Algorithm 2 applied to $(P_0, P_1, \alpha_0, \alpha_1)$ and $(P'_0, P'_1, \alpha'_0, \alpha'_1)$ produces the same result (i.e., the same solution or the same message "no solution"), which is correct for $(P_0, P_1, \alpha_0, \alpha_1)$ if and only if it is correct for $(P'_0, P'_1, \alpha'_0, \alpha'_1)$.*

Proof. Let $k \in \{0,1\}$. Recall from Section 3 the implicit constant $\beta_k \in \{+1,-1\}$, which determines whether Algorithm 2 traverses P_k (P'_k) forwards or backwards. We show that β_k has the same value for both inputs. It is well known that P_k can be *triangulated*; in particular, there are $n_k - 2$ triangles of the form $\mathcal{P}(a_t, b_t, c_t)$ with $a_t, b_t, c_t \in \{p_k[0], \ldots, p_k[n_k - 1]\}$ $(1 \le t \le n_k - 2)$, all with the same orientation (counterclockwise or clockwise), such that

$$\det^\star(p_k[0], \ldots, p_k[n_k - 1]) = \sum_{t=1}^{n-2} \big(\det(a_t, b_t) + \det(b_t, c_t) + \det(c_t, a_t)\big) = \sum_{t=1}^{n-2} \det^\star(a_t, b_t, c_t).$$

Equivalence of $(P_0, P_1, \alpha_0, \alpha_1)$ and $(P'_0, P'_1, \alpha'_0, \alpha'_1)$ implies

$$\alpha_{1-k} \operatorname{sgn} \det^\star(a_t, b_t, c_t) = \alpha'_{1-k} \operatorname{sgn} \det^\star(a'_t, b'_t, c'_t) \quad \text{for all } t \in \{1, \ldots, n_k - 2\}.$$

We conclude that all triangles $\mathcal{P}(a'_t, b'_t, c'_t)$ $(1 \le t \le n_k - 2)$ have the same orientation and

$$\alpha_{1-k} \operatorname{sgn} \det^\star(p_k[0], \ldots, p_k[n_k - 1]) = \alpha'_{1-k} \operatorname{sgn} \det^\star(p'_k[0], \ldots, p'_k[n_k - 1]).$$

This and the definition of β_k implies that β_k has the same value for both inputs.

We show that Algorithm 2 proceeds in exactly the same way for both inputs, in particular producing the same final result. Specifically, we show the same number of iterations of the "while" loop is performed for both inputs, and for each iteration i, the variables s_0, v_0, b_0, s_1, v_1, and b_1 have the same values for both inputs before and after iteration i (it is clear that u has the same value, because it depends only on i). The initial setup is common for both inputs. Now, suppose s_0, v_0, b_0, s_1, v_1, and b_1 have the same values for both inputs before iteration i. The test in line 4 produces the same outcome for both inputs, because equivalence of $(P_0, P_1, \alpha_0, \alpha_1)$ and $(P'_0, P'_1, \alpha'_0, \alpha'_1)$ implies that $\alpha_u \det^\star(p_0[s_0], p_1[s_1], p_u[v_u]) > 0$ if and only if $\alpha'_u \det^\star(p'_0[s_0], p'_1[s_1], p'_u[v_u]) > 0$. If that outcome is positive, then the test in line 5 produces the same outcome for both inputs, by an analogous argument. It follows that the same assignments are performed in iteration i for both inputs, and therefore s_0, v_0, b_0, s_1, v_1, and b_1 have the same values for both inputs after iteration i.

Finally, we show that the same final result is expected for both inputs. Let $s_0, s_1 \in \mathbb{Z}$. Equivalence of $(P_0, P_1, \alpha_0, \alpha_1)$ and $(P'_0, P'_1, \alpha'_0, \alpha'_1)$ implies that $\alpha_k \det^\star(p_0[s_0], p_1[s_1], p_k[i]) > 0$ if and only if $\alpha'_k \det^\star(p'_0[s_0], p'_1[s_1], p'_k[i]) > 0$ (for $k \in \{0,1\}$ and $i \in \mathbb{Z}$). It follows that the containment $P_k \subset \mathcal{H}_k(p_0[s_0], p_1[s_1])$ is either true for both inputs or false for both inputs (for $k \in \{0,1\}$), so (s_1, s_2) either is the correct solution or is not the correct solution for both inputs. □

First, we reduce Lemma 4.5 for cases 2 and 4 of the common tangent problem to cases 1 and 3 thereof. Consider the transformation $\phi\colon \mathbb{R}^2 \ni (x, y) \mapsto (-x, y) \in \mathbb{R}^2$ (horizontal flip). Let $P'_0 = \mathcal{P}(\phi(p_0[0]), \ldots, \phi(p_0[n_0 - 1]))$ and $P'_1 = \mathcal{P}(\phi(p_1[0]), \ldots, \phi(p_1[n_1 - 1]))$. For any three points $a = (a_x, a_y)$, $b = (b_x, b_y)$, and $c = (c_x, c_y)$ in $\mathbb{R}^2$, we have

$$\det^\star(\phi(a), \phi(b), \phi(c)) = \begin{vmatrix} -a_x & -b_x & -c_x \\ a_y & b_y & c_y \\ 1 & 1 & 1 \end{vmatrix} = -\det^\star(a, b, c).$$

It follows that the input $(P_0, P_1, \alpha_0, \alpha_1)$ is equivalent to $(P'_0, P'_1, -\alpha_0, -\alpha_1)$. If the former is case 2 or 4 of the common tangent problem, then the latter is case 1 or 3 thereof, respectively. By Lemma 5.1, it remains to prove Lemma 4.5 for cases 1 and 3 of the common tangent problem.

Now, we reduce Lemma 4.5 for case 3 of the common tangent problem to case 1 thereof. Suppose an input $(P_0, P_1, \alpha_0, \alpha_1)$ is case 3 of the common tangent problem, that is, $\alpha_0 = +1$ and $\alpha_1 = -1$. Assume that the convex hulls of P_0 and P_1 are disjoint (as in Lemma 4.5). It follows that there is a straight line separating the two convex hulls in the plane. Assume without loss of generality that it is the vertical line $x = 0$ and every corner of P_0 has negative x-coordinate while every corner of P_1 has positive x-coordinate, applying an appropriate rotation or translation of the plane to turn $(P_0, P_1, \alpha_0, \alpha_1)$ into an equivalent input that has these properties. Consider the transformation

$$\phi\colon \big\{(x, y) \in \mathbb{R}^2 \colon x \ne 0\big\} \ni (x, y) \mapsto \left(\frac{y}{x}, \frac{1}{x}\right) \in \big\{(x', y') \in \mathbb{R}^2 \colon x' \ne 0\big\}.$$

For any three points $a = (a_x, a_y)$, $b = (b_x, b_y)$, and $c = (c_x, c_y)$ in $\mathbb{R}^2$ with $a_x \neq 0$, $b_x \neq 0$, and $c_x \neq 0$, we have

$$\det^\star(\phi(a), \phi(b), \phi(c)) = \begin{vmatrix} \frac{a_y}{a_x} & \frac{b_y}{b_x} & \frac{c_y}{c_x} \\ \frac{1}{a_x} & \frac{1}{b_x} & \frac{1}{c_x} \\ 1 & 1 & 1 \end{vmatrix} = \begin{vmatrix} 1 & 1 & 1 \\ \frac{a_y}{a_x} & \frac{b_y}{b_x} & \frac{c_y}{c_x} \\ \frac{1}{a_x} & \frac{1}{b_x} & \frac{1}{c_x} \end{vmatrix} = \frac{\det^\star(a, b, c)}{a_x b_x c_x}. \qquad (\dagger)$$

Since a, b, and c are collinear if and only if $\det^\star(a, b, c) = 0$, it follows from (†) that ϕ preserves collinearity (actually, it is a projective transformation). Let $P_0' = \mathcal{P}(\phi(p_0[0]), \ldots, \phi(p_0[n_0-1]))$ and $P_1' = \mathcal{P}(\phi(p_1[0]), \ldots, \phi(p_1[n_1-1]))$; they are (simple) polygons, because ϕ is a bijection on $\{(x, y) \in \mathbb{R}^2 \colon x \neq 0\}$. For any $a, b, c \in \{p_0[0], \ldots, p_0[n_0 - 1], p_1[0], \ldots, p_1[n_1 - 1]\}$, (†) implies that

$$\begin{aligned} \operatorname{sgn}\det^\star(\phi(a), \phi(b), \phi(c)) &= \operatorname{sgn}\det^\star(a, b, c) && \text{if } |\{a, b, c\} \cap \{p_0[0], \ldots, p_0[n_0 - 1]\}| \in \{0, 2\}, \\ \operatorname{sgn}\det^\star(\phi(a), \phi(b), \phi(c)) &= -\operatorname{sgn}\det^\star(a, b, c) && \text{if } |\{a, b, c\} \cap \{p_0[0], \ldots, p_0[n_0 - 1]\}| \in \{1, 3\}. \end{aligned}$$

Therefore, the input $(P_0, P_1, \alpha_0, \alpha_1)$ is equivalent to $(P_0', P_1', \alpha_0, -\alpha_1)$. Since the former is case 3 of the common tangent problem, the latter is case 1 thereof. By Lemma 5.1, it remains to prove Lemma 4.5 for case 1 of the common tangent problem, and this is what we do in the remainder of Section 5.

5.2 Auxiliary concepts

For the sequel, we assume that the convex hulls of P_0 and P_1 are not nested, P_0 is oriented counterclockwise, P_1 is oriented clockwise, and Algorithm 2 is to solve case 1 of the outer common tangent problem—compute a pair of indices (s_0, s_1) such that $P_0, P_1 \subset \mathrm{RHP}(p_0[s_0], p_1[s_1])$.

Recall that a segment ab in the plane is considered oriented from a to b, so that forward traversal of ab starts at a and ends at b. A *polygonal path* is a curve in the plane composed of n segments $a_0a_1, \ldots, a_{n-1}a_n$ with no common points other than the common endpoints of pairs of consecutive segments in that order. Such a polygonal path is considered oriented from a_0 to a_n, so that forward traversal of it starts at a_0 and ends at a_n. A segment or a polygonal path is *degenerate* if it consists of a single point. When a non-degenerate polygonal path $a_0a_1, \ldots, a_{n-1}a_n$ (a non-degenerate segment if $n = 1$) is contained in the boundary of a polygonal region Q, we say that Q lies *to the left* or *to the right* of $a_0a_1, \ldots, a_{n-1}a_n$ if forward traversal of $a_0a_1, \ldots, a_{n-1}a_n$ agrees with counterclockwise traversal or clockwise traversal, respectively, of the boundary of Q. For $k \in \{0, 1\}$ and $a, b \in P_k$, let $P_k[a, b]$ be the polygonal path from a to b along the boundary of P_k in the forward direction (counterclockwise for P_0 and clockwise for P_1); in particular, $P_k[a, a] = \{a\}$.

Let $\ell_0, r_0 \in P_0$ and $\ell_1, r_1 \in P_1$ be such that $P_0, P_1 \subset \mathrm{LHP}(\ell_0, \ell_1) \cap \mathrm{RHP}(r_0, r_1)$. Thus r_0 and r_1 determine the requested outer common tangent, while ℓ_0 and ℓ_1 determine the other one. It is possible that $\ell_0 = r_0$ or $\ell_1 = r_1$ (but not both). For clarity of presentation, we will ignore this special case and proceed as if $\ell_0 \neq r_0$ and $\ell_1 \neq r_1$. Our arguments remain correct when $\ell_k = r_k$ ($k \in \{0, 1\}$) after adding the following exceptions to the definitions of a polygon, a polygonal path, and $P_k[a, b]$:

- the point $\ell_k = r_k$ is allowed to occur twice on a polygon (as two corners) or a polygonal path (as both endpoints of the path), where one occurrence is denoted by ℓ_k and the other by r_k;
- $P_k[\ell_k, r_k] = P_k$; forward traversal of $P_k[\ell_k, r_k]$ makes one full traversal of P_k from ℓ_k to r_k in the forward direction of P_k (counterclockwise for P_0 and clockwise for P_1).

A *door* is a segment xy such that $xy \cap P_0 = \{x\}$ and $xy \cap P_1 = \{y\}$ (always oriented from the endpoint on P_0 to the endpoint on P_1). A *zone* is a polygonal region Z such that the interior of Z is disjoint from $P_0 \cup P_1$ and the boundary of Z is the union of some non-empty part of P_0 (not necessarily connected), some non-empty part of P_1 (likewise), and some segments with both endpoints on $P_0 \cup P_1$. Let E be the polygonal region bounded by $\ell_0\ell_1 \cup P_0[\ell_0, r_0] \cup P_1[\ell_1, r_1] \cup r_0r_1$. Since $E \subset \mathrm{LHP}(\ell_0, \ell_1) \cap \mathrm{RHP}(r_0, r_1)$, it follows that E lies to the left of $\ell_0\ell_1$, to the right of $P_0[\ell_0, r_0]$, to the left of $P_1[\ell_1, r_1]$, and to the right of r_0r_1.

Observation 5.2. *The polygonal region E is a zone and satisfies $E \cap P_0 = P_0[\ell_0, r_0]$ and $E \cap P_1 = P_1[\ell_1, r_1]$. Moreover, every door or zone is contained in E. In particular, every door has one endpoint on $P_0[\ell_0, r_0]$ and the other on $P_1[\ell_1, r_1]$.*

Proof. Let $k \in \{0,1\}$ and $S_k = (\mathrm{LHP}(\ell_0,\ell_1) \cap \mathrm{RHP}(r_0,r_1)) \setminus P_k[\ell_k,r_k]$. Since E and the polygonal region bounded by P_k lie to the opposite sides of $P_k[\ell_k,r_k]$, the sets $P_k \setminus P_k[\ell_k,r_k]$ and $E \setminus P_k[\ell_k,r_k]$ are contained in different connected components of S_k. A consequence of this property is that $E \cap P_k = P_k[\ell_k,r_k]$. This, for both $k \in \{0,1\}$, proves the first statement. Another consequence is that $P_k \setminus P_k[\ell_k,r_k]$ and P_{1-k} belong to different connected components of S_k, because $E \setminus P_k[\ell_k,r_k]$ and P_{1-k} intersect. Therefore, $P_k[\ell_k,r_k]$ intersects the interior of every segment or polygonal region that is contained in $\mathrm{LHP}(\ell_0,\ell_1) \cap \mathrm{RHP}(r_0,r_1)$ and intersects both $P_k \setminus P_k[\ell_k,r_k]$ and P_{1-k}. However, every door or zone is contained in $\mathrm{LHP}(\ell_0,\ell_1) \cap \mathrm{RHP}(r_0,r_1)$ (because P_0 and P_1 are) and internally disjoint from $P_k[\ell_k,r_k]$. This, for both $k \in \{0,1\}$, proves the last two statements. □

For two doors xy and $x'y'$, let $xy \preceq x'y'$ denote that $P_0[x,x'] \subseteq P_0[\ell_0,r_0]$ and $P_1[y,y'] \subseteq P_1[\ell_1,r_1]$ (that is, forward traversal of $P_0[\ell_0,r_0]$ encounters x no later than x' and forward traversal of $P_1[\ell_1,r_1]$ encounters y no later than y'), and let $xy \prec x'y'$ denote that $xy \preceq x'y'$ and $xy \neq x'y'$. Two doors are *non-crossing* if they are disjoint or they intersect only at a common endpoint. The following observation implies that $\prec$ is a total order on any set of pairwise non-crossing doors:

Observation 5.3. *Any two non-crossing doors xy and $x'y'$ satisfy $xy \prec x'y'$ or $x'y' \prec xy$.*

Proof. Observation 5.2 yields $xy, x'y' \subset E$. If neither $xy \prec x'y'$ nor $x'y' \prec xy$, then the points x, x', y, and y' are distinct and occur on the boundary of E in this cyclic order (clockwise or counterclockwise), which contradicts the assumption that xy and $x'y'$ do not cross. □

Observation 5.4. *The boundary of every zone Z contains exactly two doors. Moreover, if these doors are denoted by xy and $x'y'$ so that $xy \prec x'y'$, then*

1. *Z lies to the left of xy and to the right of $x'y'$,*
2. *a door $x''y''$ is disjoint from the interior of Z if and only if $x''y'' \preceq xy$ or $x'y' \preceq x''y''$.*

Proof. The boundary of Z intersects both P_0 and P_1, so it contains at least two doors. By Observation 5.3, since the doors on the boundary of Z are pairwise non-crossing, they are totally ordered by $\prec$. Let xy be the minimum door and $x'y'$ be the maximum door on the boundary of Z with respect to the order $\prec$. We will show statements 1 and 2 for xy and $x'y'$. Statement 2, minimality of xy, and maximality of $x'y'$ imply that the boundary of Z contains no other doors.

For every door $x''y''$, since $Z \subseteq E$ and $x''y'' \subset E$ (by Observation 5.2), the following holds: if the set $E \setminus x''y''$ has two connected components intersecting Z, then $x''y''$ intersects the interior of Z. If neither $x''y'' \preceq xy$ nor $x'y' \preceq x''y''$, then the set $E \setminus x''y''$ has two connected components intersecting Z (by the definition of $\preceq$), so $x''y''$ intersects the interior of Z, which is one implication in statement 2. It also follows that either of the sets $E \setminus xy$ and $E \setminus x'y'$ has only one connected component intersecting Z, so Z is contained in the polygonal region $Z^\star$ bounded by $xy \cup P_0[x,x'] \cup P_1[y,y'] \cup x'y'$, which is contained in E. If $x \neq x'$, then $Z^\star$ lies to the right of $P_0[x,x']$ (because E does), and if $y \neq y'$, then $Z^\star$ lies to the left of $P_1[y,y']$ (because E does). Therefore, $Z^\star$ and thus Z lie to the left of xy and to the right of $x'y'$, which is statement 1. Moreover, if $x''y'' \preceq xy$ or $x'y' \preceq x''y''$, then $x''y''$ is disjoint from the interior of $Z^\star$ and therefore is disjoint from the interior of Z, which is the converse implication in statement 2. □

For the rest of this subsection, fix points $q_0 \in P_0$ and $q_1 \in P_1$, and consider doors contained in q_0q_1 (*doors on q_0q_1* in short). The *sign* of such a door xy on q_0q_1 is

- $+1$ if forward traversal of q_0q_1 encounters x first and y second (then xy is *positive* on q_0q_1),
- -1 if forward traversal of q_0q_1 encounters y first and x second (then xy is *negative* on q_0q_1).

See Figure 7 for an illustration.

Observation 5.5. *The signs of all doors on q_0q_1 sum up to 1.*

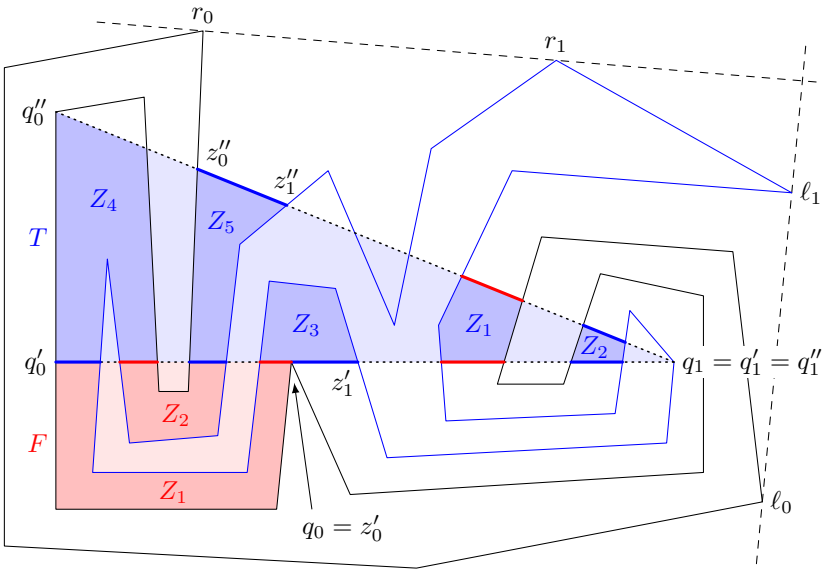


Figure 7: Thick blue segments are positive doors and thick red segments are negative doors on $q_0'q_1'$ and $q_0''q_1''$. The primary door on $q_0'q_1'$ is $z_0'z_1'$, and that on $q_0''q_1''$ is $z_0''z_1''$. The red region F (considered in Lemma 5.7) determines red zones $Z_1 \prec Z_2$. The blue region T (considered in Lemma 5.8) determines blue zones $Z_1 \prec Z_2 \prec Z_3 \prec Z_4 \prec Z_5$, of which Z_1, Z_2, and Z_5 are two-sided while Z_3 and Z_4 are one-sided.

Proof. Let $x_1y_1, \ldots, x_dy_d$ be all the doors on q_0q_1 enumerated in the order they are encountered by forward traversal of q_0q_1. The *first endpoint* of such a door is the one closer to q_0, and the *last endpoint* is the one closer to q_1. Every subsegment of q_0q_1 connecting a point on P_0 with a point on P_1 contains at least one of the doors. Since $q_0 \in P_0$, the subsegment of q_0q_1 from q_0 to the first endpoint of x_1y_1 contains no points of P_1, so x_1y_1 is positive on q_0q_1. For $i \in \{1, \ldots, d-1\}$, the subsegment of q_0q_1 from the last endpoint of x_iy_i to the first endpoint of $x_{i+1}y_{i+1}$ contains no points of P_0 or no points of P_1, so the sign of $x_{i+1}y_{i+1}$ is opposite to the sign of x_iy_i on q_0q_1. Finally, since $q_1 \in P_1$, the subsegment of q_0q_1 from the last endpoint of x_dy_d to q_1 contains no points of P_0, so x_dy_d is positive on q_0q_1. This implies that x_iy_i is positive on q_0q_1 for i odd, x_iy_i is negative on q_0q_1 for i is even, and d is odd. Therefore, the signs of $x_1y_1, \ldots, x_dy_d$ on q_0q_1 sum up to 1. □

The doors on q_0q_1 are pairwise non-crossing, so they are totally ordered by the relation $\prec$, by Observation 5.3. Let $x_1y_1, \ldots, x_dy_d$ be all the doors on q_0q_1 ordered so that $x_1y_1 \prec \cdots \prec x_dy_d$. The *primary door* on q_0q_1 is the door x_jy_j with minimum $j \in \{1, \ldots, d\}$ such that the signs of $x_1y_1, \ldots, x_jy_j$ on q_0q_1 sum up to 1. Such an index j exists, because d is a candidate, by Observation 5.5. Minimality of j in the definition of the primary door directly implies the following:

Observation 5.6. *The primary door x_iy_i is positive on q_0q_1. Moreover, if $i \geq 2$, then the door $x_{i-1}y_{i-1}$ is also positive on q_0q_1.*

5.3 Proof of Lemma 4.5 for the remaining case

We go back to the proof of Lemma 4.5. Having reduced Lemma 4.5 to case 1 of the outer common tangent problem, we have assumed the setup of that case: the convex hulls of P_0 and P_1 are not nested, P_0 is oriented counterclockwise, P_1 is oriented clockwise, and Algorithm 2 is to compute a pair of indices (s_0, s_1) such that $P_0, P_1 \subset \mathrm{RHP}(p_0[s_0], p_1[s_1])$, that is, $p_0[s_0] = r_0$ and $p_1[s_1] = r_1$.

Algorithm 2 starts with $(s_0, s_1) = (0, 0)$ and then makes some updates to the candidate solution (s_0, s_1) in line 8 until the end of the "while" loop. The second part of Lemma 4.3 explains what these updates

look like: on every update $s_u \leftarrow v_u$ in line 8 of Algorithm 2, if s_u denotes the value before the update, then $p_u[v_u] \notin \mathrm{RHP}(p_0[s_0], p_1[s_1])$ and there is $w_u \in [v_u - 1, v_u)$ such that $P_u[s_u, w_u]$ is a cap of $\mathrm{RHP}(p_0[s_0], p_1[s_1])$, $P_u[w_u, v_u]$ is the segment $p_u[w_u]p_u[v_u]$, and $p_{1-u}[s_{1-u}] \notin p_u[s_u]p_u[w_u]$.

First, we present informally the general proof idea. Imagine that an update like above happens in continuous time, as follows. Let $q_0 = p_0[s_0]$ and $q_1 = p_1[s_1]$. As the update begins, the point q_u jumps over $P_u[s_u, w_u]$ "right behind" $p_u[w_u]$ (if $s_u < w_u$), and then it moves continuously to $p_u[v_u]$ along the segment $p_u[w_u]p_u[v_u]$ of P_u. Thus $q_u = p_u[s_u]$ again after the assignment $s_u \leftarrow v_u$. As q_u is moving during the update, we track the primary door z_0z_1 on q_0q_1 and show that

1. the door z_0z_1 is only moving (piecewise continuously) forward in the order $\prec$,
2. the point q_u never passes or jumps over z_u.

To see how this implies Lemma 4.5, consider the overall move of q_0, q_1, and z_0z_1 during all updates to the candidate solution (s_0, s_1), starting from $q_0 = p_0[0]$ and $q_1 = p_1[0]$. For each $k \in \{0, 1\}$, statement 1 implies that z_k is only moving forward on $P_k[\ell_k, r_k]$, never passing or jumping over r_k, and statement 2 asserts that q_k never passes or jumps over z_k, whence it follows that q_k passes or jumps over r_k at most once.

Now, we proceed with the proof of Lemma 4.5. The next two lemmas formalize statement 1 above—Lemma 5.7 for the initial jump over $P_u[s_u, w_u]$, and Lemma 5.8 for the continuous move along $p[w_u]p[v_u]$. See Figure 7 for an illustration of Lemmas 5.7 and 5.8. Statement 2 above is formalized by the invariant that we formulate after proving Lemmas 5.7 and 5.8.

Lemma 5.7. *Let $u \in \{0, 1\}$, $q_u, q'_u \in P_u$, and $q_{1-u} = q'_{1-u} \in P_{1-u}$. If $P_u[q_u, q'_u]$ is a cap of $\mathrm{RHP}(q_0, q_1)$ and $q_{1-u} \notin q_uq'_u$, then the same door is primary on q_0q_1 and on $q'_0q'_1$.*

Proof. The lemma is trivial when $q_u = q'_u$, so assume $q_u \neq q'_u$. Thus $q_0q_1 \subset q'_0q'_1$ or $q'_0q'_1 \subset q_0q_1$ (because $q_{1-u} \notin q_uq'_u$). Let $q''_0q''_1$ be the longer of q_0q_1 and $q'_0q'_1$ ($q'_0q'_1$ in the former and q_0q_1 in the latter case). Let F be the polygonal region bounded by $P_u[q_u, q'_u] \cup q_uq'_u$. Thus $F \subset \mathrm{RHP}(q''_0, q''_1)$. Let $\mathcal{Z}$ be the set of zones contained in F with boundaries contained in $(P_0 \cap F) \cup (P_1 \cap F) \cup q_uq'_u$. For every door $xy \subset q_uq'_u$, there is a zone in $\mathcal{Z}$ to the right of xy (if F is to the right of xy) or to the left of xy (if F is to the left of xy). By Observation 5.4, the zones in $\mathcal{Z}$ can be ordered as $Z_1, \ldots, Z_d$ and the doors contained in $q_uq'_u$ can be ordered as $x_1y_1, x^1y^1, \ldots, x_dy_d, x^dy^d$ so that

- every zone $Z_i \in \mathcal{Z}$ has exactly two doors on the boundary, namely, x_iy_i and x^iy^i,
- $x_1y_1 \prec x^1y^1 \prec \cdots \prec x_dy_d \prec x^dy^d$.

For each $i \in \{1, \ldots, d\}$, since $Z_i \subset \mathrm{RHP}(q''_0, q''_1)$, Observation 5.4 (1 and 2) implies that

- x_iy_i is negative and x^iy^i is positive on $q''_0q''_1$,
- x_iy_i and x^iy^i are consecutive in the order $\prec$ of the doors on $q''_0q''_1$.

By Observation 5.6, none of $x_1y_1, x^1y^1, \ldots, x_dy_d, x^dy^d$ is primary on $q''_0q''_1$. Moreover, for each door xy on the shorter of q_0q_1 and $q'_0q'_1$, the following two sums are equal:

- the sum of the signs of all doors on $q''_0q''_1$ up to xy in the order $\prec$,
- the sum of the signs of all doors on the shorter of q_0q_1 and $q'_0q'_1$ up to xy in the order $\prec$.

We conclude that the same door is primary on $q''_0q''_1$ and on the shorter of q_0q_1 and $q'_0q'_1$. □

Lemma 5.8. *Let $u \in \{0, 1\}$, $q'_uq''_u \subset P_u$, and $q'_{1-u} = q''_{1-u} \in P_{1-u}$. Let $z'_0z'_1$ be the primary door on $q'_0q'_1$ and $z''_0z''_1$ be the primary door on $q''_0q''_1$. If $q''_u \notin \mathrm{RHP}(q'_0, q'_1)$, then $z'_0z'_1 \prec z''_0z''_1$.*

Proof. Let T be the triangular region bounded by $q_0'q_1' \cup q_0'q_0'' \cup q_1'q_1'' \cup q_0''q_1''$, where either $q_0'q_0''$ or $q_1'q_1''$ is a degenerate segment. Thus $T \subset \mathrm{LHP}(q_0', q_1') \cap \mathrm{RHP}(q_0'', q_1'')$. Let $\mathcal{Z}$ be the set of zones contained in T with boundaries contained in $q_0'q_1' \cup (P_0 \cap T) \cup (P_1 \cap T) \cup q_0''q_1''$. For every door $xy \subset q_0'q_1' \cup q_0''q_1''$, there is a zone in $\mathcal{Z}$ to the right of xy (if T lies to the right of xy) or to the left of xy (if T lies to the left of xy). By Observation 5.4, the zones in $\mathcal{Z}$ can be ordered as $Z_1, \ldots, Z_d$ and the doors contained in $q_0'q_1' \cup q_0''q_1''$ can be ordered as $x_1y_1, x^1y^1, \ldots, x_dy_d, x^dy^d$ so that

- every zone $Z_i \in \mathcal{Z}$ has exactly two doors on the boundary, namely, x_iy_i and x^iy^i,
- $x_1y_1 \prec x^1y^1 \prec \cdots \prec x_dy_d \prec x^dy^d$.

For every $i \in \{1, \ldots, d\}$, since $Z_i \subset \mathrm{LHP}(q_0', q_1') \cap \mathrm{RHP}(q_0'', q_1'')$, Observation 5.4 (1) implies that

- x_iy_i is a positive door on $q_0'q_1'$ or a negative door on $q_0''q_1''$,
- x^iy^i is a negative door on $q_0'q_1'$ or a positive door on $q_0''q_1''$.

We will use these two properties extensively without explicit reference.

We say that a zone $Z_i \in \mathcal{Z}$ is *one-sided* if x_iy_i and x^iy^i lie both on $q_0'q_1'$ or both on $q_0''q_1''$, otherwise we say that Z_i is *two-sided*. For each one-sided zone $Z_i \in \mathcal{Z}$, the doors x_iy_i and x^iy^i have opposite signs on $q_0'q_1'$ or $q_0''q_1''$ (whichever they lie on). For each two-sided zone $Z_i \in \mathcal{Z}$, if $x_i'y_i'$ and $x_i''y_i''$ denote the two doors on the boundary of Z_i so that $x_i'y_i' \subseteq q_0'q_1'$ and $x_i''y_i'' \subseteq q_0''q_1''$, then the sign of $x_i'y_i'$ on $q_0'q_1'$ is equal to the sign of $x_i''y_i''$ on $q_0''q_1''$. Let I be the set of indices $i \in \{1, \ldots, d\}$ such that Z_i is a two-sided zone in $\mathcal{Z}$. The above and Observation 5.5 implies that

- the signs of the doors $x_i'y_i'$ on $q_0'q_1'$ over all $i \in I$ sum up to 1,
- the signs of the doors $x_i''y_i''$ on $q_0''q_1''$ over all $i \in I$ sum up to 1,

and the following four sums are equal, for each $j \in I$:

- the sum of the signs of all doors on $q_0'q_1'$ up to $x_j'y_j'$ in the order $\prec$,
- the sum of the signs of the doors $x_i'y_i'$ on $q_0'q_1'$ over all $i \in I \cap \{1, \ldots, j\}$,
- the sum of the signs of the doors $x_i''y_i''$ on $q_0''q_1''$ over all $i \in I \cap \{1, \ldots, j\}$,
- the sum of the signs of all doors on $q_0''q_1''$ up to $x_j''y_j''$ in the order $\prec$.

Let $j \in I$ be minimum such that the four sums above are equal to 1. Since x_iy_i is positive and x^iy^i is negative on $q_0'q_1'$ for every (one-sided) zone $Z_i \in \mathcal{Z}$ such that $x_iy_i, x^iy^i \subseteq q_0'q_1'$, it follows that the primary door on $q_0'q_1'$ is either $x_j'y_j'$ or x_iy_i for some (one-sided) zone $Z_i \in \mathcal{Z}$ with $x_iy_i, x^iy^i \subseteq q_0'q_1'$ and $i < j$, and thus $x_iy_i \prec x_j'y_j'$. For every (one-sided) zone $Z_i \in \mathcal{Z}$ such that $x_iy_i, x^iy^i \subseteq q_0''q_1''$, since x_iy_i is negative and x^iy^i is positive on $q_0''q_1''$, neither x_iy_i nor x^iy^i is primary on $q_0''q_1''$, by Observation 5.6. It follows that $x_j''y_j''$ is the primary door on $q_0''q_1''$. Since the primary door is always positive, we have $x_j'y_j' = x_jy_j$ and $x_j''y_j'' = x^jy^j$, and thus $x_j'y_j' \prec x_j''y_j''$. We conclude that $z_0'z_1' \preceq x_j'y_j' \prec x_j''y_j'' = z_0''z_1''$. □

Let $a_0 \in [0, n_0)$ and $a_1 \in [0, n_1)$ be such that $p_0[a_0] = \ell_0$ and $p_1[a_1] = \ell_1$. To prove Lemma 4.5, we show that Algorithm 2 maintains the following invariant: if $c_0 \in [a_0, a_0 + n_0)$ and $c_1 \in [a_1, a_1 + n_1)$ are such that $p_0[c_0]p_1[c_1]$ is the primary door on $p_0[s_0]p_1[s_1]$, then $s_0 \le c_0$ and $s_1 \le c_1$. The invariant implies that $s_0 \le c_0 < a_0 + n_0 < 2n_0$ and $s_1 \le c_1 < a_1 + n_1 < 2n_1$ at the end of the "while" loop in Algorithm 2, which is the assertion of Lemma 4.5.

Algorithm 2 starts with $s_0 = 0 \le a_0 \le c_0$ and $s_1 = 0 \le a_1 \le c_1$, so the invariant holds initially. Consider an update $s_u \leftarrow v_u$ in line 8 of Algorithm 2, where $u \in \{0, 1\}$, assuming that the invariant holds before the update. Let s_u denote the value before the update and w_u be as claimed by the second part of Lemma 4.3. Let $q_u = p_u[s_u]$, $q_u' = p_u[w_u]$, $q_u'' = p_u[v_u]$, and $q_{1-u} = q_{1-u}' = q_{1-u}'' = p_{1-u}[s_{1-u}]$. The conclusions of Lemma 4.3 imply that q_uq_u' is a segment of $\mathcal{L}(q_0, q_1)$ not containing q_{1-u}, $P_u[q_u, q_u']$ is a cap of $\mathrm{RHP}(q_0, q_1)$,

$P_u[q'_u, q''_u]$ is the single segment $q'_u q''_u$, and $q''_u \notin \mathrm{RHP}(q_0, q_1) = \mathrm{RHP}(q'_0, q'_1)$, where the last equality follows from $q'_{1-u} = q_{1-u} \in \mathcal{L}(q_0, q_1) \setminus q_u q'_u$. These conditions are what we need to apply Lemma 5.7 (to q_0, q_1, q'_0, and q'_1) and Lemma 5.8 (to q'_0, q'_1, q''_0, and q''_1). Let $z_0 z_1$, $z'_0 z'_1$, and $z''_0 z''_1$ be the primary doors on $q_0 q_1$, $q'_0 q'_1$, and $q''_0 q''_1$, respectively. Lemma 5.7 and Lemma 5.8 yield $z_0 z_1 = z'_0 z'_1 \prec z''_0 z''_1$. Let $c_0, c''_0 \in [a_0, a_0 + n_0)$ and $c_1, c''_1 \in [a_1, a_1 + n_1)$ be such that $p_0[c_0] = z_0$, $p_0[c''_0] = z''_0$, $p_1[c_1] = z_1$, and $p_1[c''_1] = z''_1$. This and $z_0 z_1 \prec z''_0 z''_1$ imply $c_0 < c''_0$ and $c_1 < c''_1$. This and the assumption that $s_0 \le c_0$ and $s_1 \le c_1$ (which is the invariant before the update) imply $s_0 < c''_0$ and $s_1 < c''_1$. This already gives one inequality of the invariant after the update, namely, $s_{1-u} \le c''_{1-u}$. It remains to prove $v_u \le c''_u$, which is the other inequality of the invariant after the update. Since $q''_u \notin \mathrm{RHP}(q_0, q_1)$ and $q''_{1-u} = q_{1-u}$, we have $\mathrm{RHP}(q_0, q_1) \cap q''_0 q''_1 = \{q''_{1-u}\}$. This and $P_u[q_u, q'_u] \subset \mathrm{RHP}(q_0, q_1)$ imply $P_u[q_u, q'_u] \cap q''_0 q''_1 = \emptyset$. Since $P_u[q'_u, q''_u] = q'_u q''_u$ and $q'_u \notin q''_0 q''_1$, we have $P_u[q'_u, q''_u] \cap q''_0 q''_1 = \{q''_u\}$. Thus $P_u[q_u, q''_u] \cap q''_0 q''_1 = \{q''_u\}$. This and the fact that $p_u[c''_u] = z''_u \in P_u \cap q''_0 q''_1$ imply $c''_u \notin [s_u, v_u)$. This and $s_u < c''_u$ give the requested inequality $v_u \le c''_u$. We conclude that the invariant is preserved at the considered update to s_u, which completes the proof of Lemma 4.5.

6 Concluding remarks

So far, we were assuming that the combined set $\hat{P}$ of corners of P_0 and P_1 contains no triple of collinear points, which guarantees that expressions of the form $\operatorname{sgn}\det^\star(a, b, c)$ with $a, b, c \in \hat{P}$ are non-zero. Such expressions are evaluated in line 4 of Algorithm 1 and lines 4 and 5 of Algorithm 2. Now, we adapt the algorithms to handle the case that $\hat{P}$ may contain triples of collinear points. Consider the following transformation, where ϵ is a (very small) positive constant:

$$\phi_\epsilon \colon \mathbb{R}^2 \ni (x, y) \mapsto \big(x + \epsilon y, y + \epsilon(x + \epsilon y)^2\big) \in \mathbb{R}^2.$$

It has the property that for any distinct points $a, b, c \in \mathbb{R}^2$, the limit of $\operatorname{sgn}\det^\star(\phi_\epsilon(a), \phi_\epsilon(b), \phi_\epsilon(c))$ as $\epsilon \to 0$ exists, is non-zero, and can be easily computed from the coordinates of a, b, and c. We can modify Algorithms 1 and 2 so that they evaluate that limit instead of $\operatorname{sgn}\det^\star(a, b, c)$ for any distinct points $a, b, c \in \hat{P}$. Conceptually, this is the same as invoking the algorithms on the polygons P_0 and P_1 transformed by ϕ_ϵ, where ϵ is small enough so that the value of $\operatorname{sgn}\det^\star(\phi_\epsilon(a), \phi_\epsilon(b), \phi_\epsilon(c))$ is equal to the limit for any distinct points $a, b, c \in \hat{P}$. Consequently, if either algorithm claims to find a solution, the fact that it is correct on the transformed input implies that it is correct on the original input. The same reasoning (and the same modification of the algorithms) can be applied with the following transformation ψ_ϵ instead of ϕ_ϵ:

$$\psi_\epsilon \colon \mathbb{R}^2 \ni (x, y) \mapsto \big(x + \epsilon y, y - \epsilon(x + \epsilon y)^2\big) \in \mathbb{R}^2.$$

Any straight line is transformed by ϕ_ϵ and ψ_ϵ (with ϵ small enough) into two lines that curve in the opposite directions. This property is important when we want to find "degenerate" common tangents. Namely, if the convex hulls of P_0 and P_1 touch, then one of ϕ_ϵ and ψ_ϵ makes them overlap properly while the other makes them disjoint; therefore, one modification of Algorithm 1 or 2 finds the "degenerate" separating common tangent while the other does not. Similarly, if the convex hulls of P_0 and P_1 are nested and their boundaries touch, then one of ϕ_ϵ and ψ_ϵ makes them overlap properly while the other moves the smaller convex hull to the interior of the larger; therefore, one modification of Algorithm 2 finds the "degenerate" outer common tangent while the other does not. We recognize that there is no solution when both modifications report no solution.

It remains open whether an outer common tangent of two polygons that are not disjoint can be found in linear time and constant workspace. Another natural question is whether the diameter of the convex hull of a simple polygon can be computed in linear time and constant workspace.

References

[1] M. Abrahamsen. An optimal algorithm for the separating common tangents of two polygons. In *31st International Symposium on Computational Geometry (SoCG 2015)*, volume 34 of *LIPIcs*, pages 198–208, 2015. arXiv:1511.04036 (corrected version).

[2] M. Abrahamsen, M. de Berg, K. Buchin, M. Mehr, and A.D. Mehrabi. Minimum perimeter-sum partitions in the plane. In *33rd International Symposium on Computational Geometry (SoCG 2017)*, volume 77 of *LIPIcs*, pages 4:1–4:15, 2017.

[3] M. Abrahamsen and B. Walczak. Outer common tangents and nesting of convex hulls in linear time and constant workspace. In *24th Annual European Symposium on Algorithms (ESA 2016)*, volume 57 of *LIPIcs*, pages 4:1–4:15, 2016.

[4] T. Asano, K. Buchin, M. Buchin, M. Korman, W. Mulzer, G. Rote, and A. Schulz. Memory-constrained algorithms for simple polygons. *Comput. Geom.*, 46(8):959–969, 2013.

[5] T. Asano, W. Mulzer, G. Rote, and Y. Wang. Constant-work-space algorithms for geometric problems. *J. Comput. Geom.*, 2(1):46–68, 2011.

[6] L. Barba, M. Korman, S. Langerman, K. Sadakane, and R.I. Silveira. Space–time trade-offs for stack-based algorithms. *Algorithmica*, 72(4):1097–1129, 2015.

[7] L. Barba, M. Korman, S. Langerman, and R.I. Silveira. Computing the visibility polygon using few variables. *Comput. Geom.*, 47(9):918–926, 2014.

[8] G.S. Brodal and R. Jacob. Dynamic planar convex hull. In *43rd Annual IEEE Symposium on Foundations of Computer Science (FOCS 2002)*, pages 617–626, 2002.

[9] E. Carson, J. Demmel, L. Grigori, N. Knight, P. Koanantakool, O. Schwartz, and H.V. Simhadri. Write-avoiding algorithms. In *30th IEEE International Parallel and Distributed Processing Symposium (IPDPS 2016)*, pages 648–658, 2016.

[10] L. Guibas, J. Hershberger, and J. Snoeyink. Compact interval trees: a data structure for convex hulls. *Int. J. Comput. Geom. Appl.*, 1(1):1–22, 1991.

[11] L.J. Guibas and J. Hershberger. Optimal shortest path queries in a simple polygon. *J. Comput. System Sci.*, 39(2):126–152, 1989.

[12] S. Har-Peled. Shortest path in a polygon using sublinear space. In *31st International Symposium on Computational Geometry (SoCG 2015)*, volume 34 of *LIPIcs*, pages 111–125, 2015.

[13] J. Hershberger and S. Suri. Applications of a semi-dynamic convex hull algorithm. *BIT Numer. Math.*, 32(2):249–267, 1992.

[14] J. Hershberger and S. Suri. A pedestrian approach to ray shooting: shoot a ray, take a walk. *J. Algorithms*, 18(3):403–431, 1995.

[15] D. Kirkpatrick and J. Snoeyink. Computing common tangents without a separating line. In *4th International Workshop on Algorithms and Data Structures (WADS 1995)*, volume 955 of *LNCS*, pages 183–193. Springer, 1995.

[16] A.A. Melkman. On-line construction of the convex hull of a simple polyline. *Inform. Process. Lett.*, 25(1):11–12, 1987.

[17] M.H. Overmars and J. van Leeuwen. Maintenance of configurations in the plane. *J. Comput. System Sci.*, 23(2):166–204, 1981.

[18] F.P. Preparata and S.J. Hong. Convex hulls of finite sets of points in two and three dimensions. *Commun. ACM*, 20(2):87–93, 1977.

[19] O. Reingold. Undirected connectivity in log-space. *Journal of the ACM (JACM)*, 55(4):1–24, 2008.

[20] G.T. Toussaint. Solving geometric problems with the rotating calipers. In *IEEE Mediterranean Electrotechnical Conference (MELECON 1983)*, pages A10.02/1–4, 1983.

B

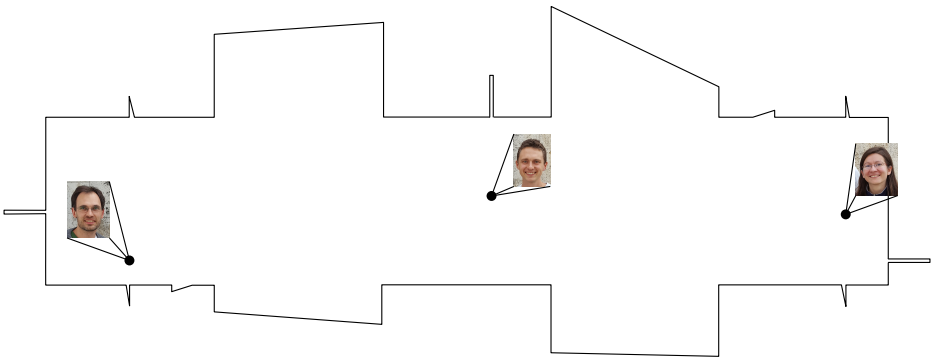
IRRATIONAL GUARDS ARE SOMETIMES NEEDED

B

Irrational Guards are Sometimes Needed

Mikkel Abrahamsen[1], Anna Adamaszek[1], and Tillmann Miltzow[2]

[1]University of Copenhagen, Denmark. {miab,anad}@di.ku.dk
[2]Institute for Computer Science and Control, Hungarian Academy of Sciences (MTA SZTAKI), t.miltzow@gmail.com



Till, Mikkel, Anna meticulously guarding the polygon: a little irrational, but pretty optimal.

Abstract

In this paper we study the *art gallery problem*, which is one of the fundamental problems in computational geometry. The objective is to place a minimum number of guards inside a simple polygon such that the guards together can see the whole polygon. We say that a guard at position x sees a point y if the line segment xy is fully contained in the polygon.

Despite an extensive study of the art gallery problem, it remained an open question whether there are polygons given by integer coordinates that require guard positions with irrational coordinates in any optimal solution. We give a positive answer to this question by constructing a *monotone* polygon with integer coordinates that can be guarded by three guards only when we allow to place the guards at points with irrational coordinates. Otherwise, four guards are needed. By extending this example, we show that for every n, there is polygon which can be guarded by $3n$ guards with irrational coordinates but need $4n$ guards if the coordinates have to be rational. Subsequently, we show that there are rectilinear polygons given by integer coordinates that require guards with irrational coordinates in any optimal solution.

1 Introduction

For a polygon $\mathcal{P}$ and points $x, y \in \mathcal{P}$, we say that x *sees* y if the interval xy is contained in $\mathcal{P}$. A *guard set* S is a set of points in $\mathcal{P}$ such that every point in $\mathcal{P}$ is seen by some point in S. The points in S are called *guards*. The *art gallery problem* is to find a minimum cardinality guard set for a simple polygon $\mathcal{P}$ on n vertices. The polygon $\mathcal{P}$ is considered to be filled, i.e., it consists of a closed polygonal curve in the plane and the bounded region enclosed by this curve.

This classical version of the art gallery problem has been originally formulated in 1973 by Victor Klee (see the book of O'Rourke [24, page 2]). It is often referred to as the *interior-guard art gallery problem* or the *point-guard art gallery problem*, to distinguish it from other versions that have been introduced over the years.

In 1978, Steve Fisk provided an elegant proof that $\lfloor n/3 \rfloor$ guards are always sufficient and sometimes necessary to guard a polygon with n vertices [18]. Five years earlier, Victor Klee had posed this question to Václav Chvátal, who soon gave a more complicated solution [12]. Since then, the art gallery problem has been extensively studied, both from the combinatorial and the algorithmic perspective. Most of this research, however, is not focused directly on the classical art gallery problem, but on its numerous versions, including different definitions of visibility, restricted classes of polygons, different shapes of the guards, restrictions on the positions of the guards, etc. For more detailed information we refer the reader to the following surveys [20, 24, 29, 31].

Despite extensive research on the art gallery problem, no combinatorial algorithm for finding an optimal solution, or even for deciding whether a guard set of a given size k exists, is known. The only exact algorithm is attributed to Micha Sharir (see [15]), who has shown that in $n^{O(k)}$ time one can find a guard set consisting of k guards, if such a guard set exists. This result is obtained by using standard tools from real algebraic geometry [3], and it is not known how to find an optimal solution without using this powerful machinery (see [4] for an analysis of the very restricted case of $k = 2$). To stress this even more: Without the tools from algebraic geometry, we would not know if it is decidable whether a guard set of size k exists or not! Some recent lower bounds [6] based on the exponential time hypothesis suggest that there might be no better exact algorithms than the one by Sharir.

To explain the difficulty in constructing exact algorithms, we want to emphasize that it is *not* known whether the decision version of the art gallery problem (i.e., the problem of deciding whether there is a guard set consisting of k guards, where k is a parameter) lies in the complexity class NP, even with the algorithm by Sharir. While NP-hardness and APX-hardness of the art gallery problem have been shown for different classes of polygons [7, 16, 21, 22, 25, 28, 30], the question of whether the point-guard art gallery problem is in NP remains open. A simple way to show NP-membership would be to prove that there always exists an optimal set of guards with *rational* coordinates of polynomially bounded description.

Indeed, Sándor Fekete posed at MIT in 2010 and at Dagstuhl in 2011 an open problem, asking whether there are polygons requiring irrational coordinates in an optimal guard set [1, 17]. The question has been raised again by Günter Rote at EuroCG 2011 [26]. It has also been mentioned by Rezende *et al.* [13]: "it remains an open question whether there are polygons given by rational coordinates that require optimal guard positions with irrational coordinates". A similar question has been raised by Friedrichs *et al.* [19]: "[...] it is a long-standing open problem for the more general Art Gallery Problem (AGP): For the AGP it is not known whether the coordinates of an optimal guard cover can be represented with a polynomial number of bits".

Our results. We answer the open question of Sándor Fekete, by proving the following main result of our paper. Recall that a polygon $\mathcal{P}$ is called *monotone* if there exists a line l such that every line orthogonal to l intersects $\mathcal{P}$ at most twice.

Theorem 1. *There is a simple monotone polygon $\mathcal{P}$ with integer coordinates of the vertices such that*

(i) $\mathcal{P}$ can be guarded by 3 guards placed at points with irrational coordinates, and

(ii) an optimal guard set of $\mathcal{P}$ with guards at points with rational coordinates has size 4.

We then extend this result, by providing a family of polygons for which the ratio between the number of guards in an optimal solution restricted to guards at rational positions, to the number of guards in an optimal solution allowing irrational guards, is 4/3.

Theorem 2. *There is a family of simple polygons $(\mathcal{P}_n)_{n\in\mathbb{Z}_+}$ with integer coordinates of the vertices such that*

(i) $\mathcal{P}_n$ can be guarded by $3n$ guards placed at points with irrational coordinates, and

(ii) an optimal guard set of $\mathcal{P}_n$ with guards at points with rational coordinates has size $4n$.

Moreover, the coordinates of the points defining the polygons $\mathcal{P}_n$ are polynomial in n.

We show that the phenomenon with guards at irrational coordinates occurs also in the important class of rectilinear polygons.

Theorem 3. *There is a rectilinear polygon $\mathcal{P}_R$ with vertices at integer coordinates satisfying the following properties.*

(i) $\mathcal{P}_R$ can be guarded by 9 guards if we allow placing guards at points with irrational coordinates.

(ii) An optimal guard set of $\mathcal{P}_R$ with guards at points with rational coordinates has size 10.

Other related work. The art gallery problem has been studied from the perspective of approximation algorithms. Efrat and Har-Peled [15] gave a randomized polynomial time algorithm for finding a guard set S where the guards are restricted to a very fine grid Γ. To be more precise, if coordinates of the vertices of the input polygon $\mathcal{P}$ are given by positive integers and L is the largest such integer, then Γ can be defined as the points in $L^{-20}\cdot\mathbb{Z}^2\cap\mathcal{P}$. Let $OPT_{\text{grid}}\subset\Gamma$ be a guard set with a minimum number of guards under this restriction. The algorithm of Efrat and Har-Peled yields an $O(\log|OPT_{\text{grid}}|)$-approximation for this problem. However, it remained open whether OPT_{grid} is an approximation of an optimal unrestricted guard set OPT. Bonnet and Miltzow [5] filled this gap by showing that under a general position assumption $|OPT_{\text{grid}}|=O(|OPT|)$, which yields the first polynomial time approximation algorithm for simple polygons under this assumption. It is easy to construct a polygon with integer coordinates that forces a guard on the point $(1/3,1/3)$, which might not lie on the grid, in case that L is not divisible by 3. This implies that OPT_{grid} is not optimal. But this does not rule out that there is a slightly more clever choice of Γ so that OPT_{grid} is indeed optimal. It follows from our Theorem 2 that there are polygons (requiring arbitrarily many guards in an optimal guard set) such that for any choice of $\Gamma\subset\mathbb{Q}^2$, it holds that $|OPT_{\text{grid}}|\geq 4/3\cdot|OPT|$. No lower bound of this kind has been known before. More general, our result shows that no algorithm which considers only rational points as possible guard positions can achieve an approximation ratio better than $4/3$.

A new line of research focuses on implementing algorithms that are capable of solving instances of the art gallery problem with thousands of vertices, giving a solution which is close to the optimal one, see the recent survey by Rezende *et al.* [13]. They explain that many practical algorithms rely on "routines to find candidates for discrete guard and witness locations." We show that this technique inevitably leads to sub-optimal solutions unless irrational candidate locations are also considered. We believe that our example and techniques are a good starting point to construct benchmark instances for implementations of art gallery algorithms. Benchmark instances serve to validate the quality of algorithms. Using the same instances when comparing different algorithms makes the results comparable.

A problem related to the art-gallery problem is the *terrain guarding problem*. In this problem, an x-monotone polygonal curve c (i.e., the terrain) is given. The region R above the curve c has to be guarded, and the guards are restricted to lie on c. Similarly as in our problem, a guard x sees a point y if xy is contained in the region R. Although the solution space of the terrain guarding instance is the continuous polygonal curve c, a discretization of the solution space has been recently described by Friedrichs *et al.* [19]. Given a terrain with n vertices at integer position, they describe a set $S\subset\mathbb{Q}$ of size $O(n^3)$, computable in polynomial time, such that there is an optimal guard placement restricted to S. It follows that for the terrain guarding problem the phenomenon with irrational numbers does not appear, and also the decision version of the terrain guarding problem is in NP.

Irrational numbers turn up surprisingly in other areas of computational geometry. One such example is the *nested polytopes problem*. Here, we are given two nested polytopes $S\subseteq P$ and want to find a polytope T with a minimum number of corners such that T is nested between S and P, i.e., $S\subseteq T\subseteq P$. Christikov *et al.* [11] recently gave an example of two nested polytopes $S\subseteq P$ in $\mathbb{R}^3$, with all corners at rational coordinates, such that there is a unique polytope T with 5 corners nested between S and P, and T has corners with irrational coordinates. The nested polytopes problem is closely related to *nonnegative*

matrix multiplication, where similar phenomena have been discovered, that a problem defined entirely by rational numbers has an optimal solution requiring irrational numbers [10, 11].

The Structure of the Paper. Section 2 contains the description of a monotone polygon $\mathcal{P}$ with vertices at points with rational coordinates that can be guarded by three guards only if the guards are placed at points with irrational coordinates. In Section 3, we describe the intuition behind our construction, and explain how we have found the polygon $\mathcal{P}$. The formal proof of Theorems 1 and 2 is then provided in Section 4. In Section 5, we present the rectilinear polygon $\mathcal{P}_R$ from Theorem 3 requiring guards with irrational coordinates in an optimal guard set. Finally, in Section 6 we suggest some open problems for future research.

2 The Polygon

In Figure 1 we present the polygon $\mathcal{P}$. In Section 4 we will prove that $\mathcal{P}$ can be guarded by three guards only when we allow the guards to be placed at points with irrational coordinates.

The polygon $\mathcal{P}$ is constructed as follows. We start with a *basic rectangle* $[0, 20] \times [0, 4] \subset \mathbb{R}^2$. Then, we append to it six *triangular pockets* (colored with green in the figure), which are triangles defined by the following coordinates

$T_t^\ell : \{(2, 4), (2, 4.5), (2.1, 4)\}$, $T_b^\ell : \{(2, 0), (2, -0.5), (1.9, 0)\}$,
$T_t^m : \{(16\frac{5}{6}, 4), (17\frac{2}{6}, 4.15), (17\frac{2}{6}, 4)\}$, $T_b^m : \{(3.5, 0), (3, -0.15), (3, 0)\}$,
$T_t^r : \{(19, 4), (19, 4.5), (19.1, 4)\}$, and $T_b^r : \{(19, 0), (19, -0.5), (18.9, 0)\}$.

Next, we append three *rectangular pockets* (colored with blue in the figure, for practical reasons these pockets are drawn in the figure shorter than they actually are), which are rectangles defined in the following way.

R_ℓ: $[-10, 0] \times [1.7, 1.8]$, R_r: $[20, 30] \times [0.5, 0.6]$, and R_m: $[10.5, 10.6] \times [4, 8]$.

Last, we append four *quadrilateral pockets* (colored with red in the figure), which are defined by points with the following coordinates

Top-left pocket P_t^ℓ	$\{(4, 4),$	$(4, \frac{280}{47}),$	$(8, \frac{294}{47}),$	$(8, 4)\}$
Top-right pocket P_t^r	$\{(12, 4),$	$(12, \frac{2486}{375}),$	$(16, \frac{1776}{375}),$	$(16, 4)\}$
Bottom-left pocket P_b^ℓ	$\{(4, 0),$	$(4, -\frac{12}{19}),$	$(8, -\frac{18}{19}),$	$(8, 0)\}$
Bottom-right pocket P_b^r	$\{(12, 0),$	$(12, -\frac{34}{21}),$	$(16, -\frac{36}{21}),$	$(16, 0)\}$.

The polygon $\mathcal{P}$ is clearly monotone. We will denote by e_t^ℓ, e_t^r, e_b^ℓ, and e_b^r the non-axis-parallel edge within each of the four quadrilateral pockets, respectively.

3 Intuition

In this section, we explain the key ideas behind the construction of the polygon $\mathcal{P}$. Our presentation is informal, but it resembles the work process that lead to the construction of $\mathcal{P}$ more than the formal proof of Theorem 1 in Section 4 does. Here we omit all "scary" computations and focus on conveying the big picture. In the end of this section, we also explain how we actually constructed the polygon $\mathcal{P}$.

Define a *rational point* to be a point with two rational coordinates. An *irrational point* is a point that is not rational. A *rational line* is a line that contains two rational points. An *irrational line* is a line that is not rational.

Forcing a Guard on a Line Segment. Consider the drawing of the polygon $\mathcal{P}$ in Figure 1. We will now explain an idea of how three pairs of triangular pockets, (T_t^ℓ, T_b^ℓ), (T_t^m, T_b^m), and (T_t^r, T_b^r), can enforce three guards on three line segments within $\mathcal{P}$.

Consider the two triangular pockets in Figure 2a. The blue line segment contains one edge of each of these pockets, and the interiors of the pockets are at different sides of the line segment. A guard which sees the point t must be placed within the orange triangular region, and a guard which sees b must be placed within the yellow triangular region. Thus, a single guard can see both t and b only if it is on the blue line segment tb, which is the intersection of the two regions.

Consider now the case that we have k pairs of triangular pockets, and no two regions corresponding to different pairs of pockets intersect. In order to guard the polygon with k guards, there must be one guard on the line segment corresponding to each pair. Our polygon $\mathcal{P}$ has three such pairs of pockets (see Figure 2b), and it can be checked that the corresponding regions do not intersect.

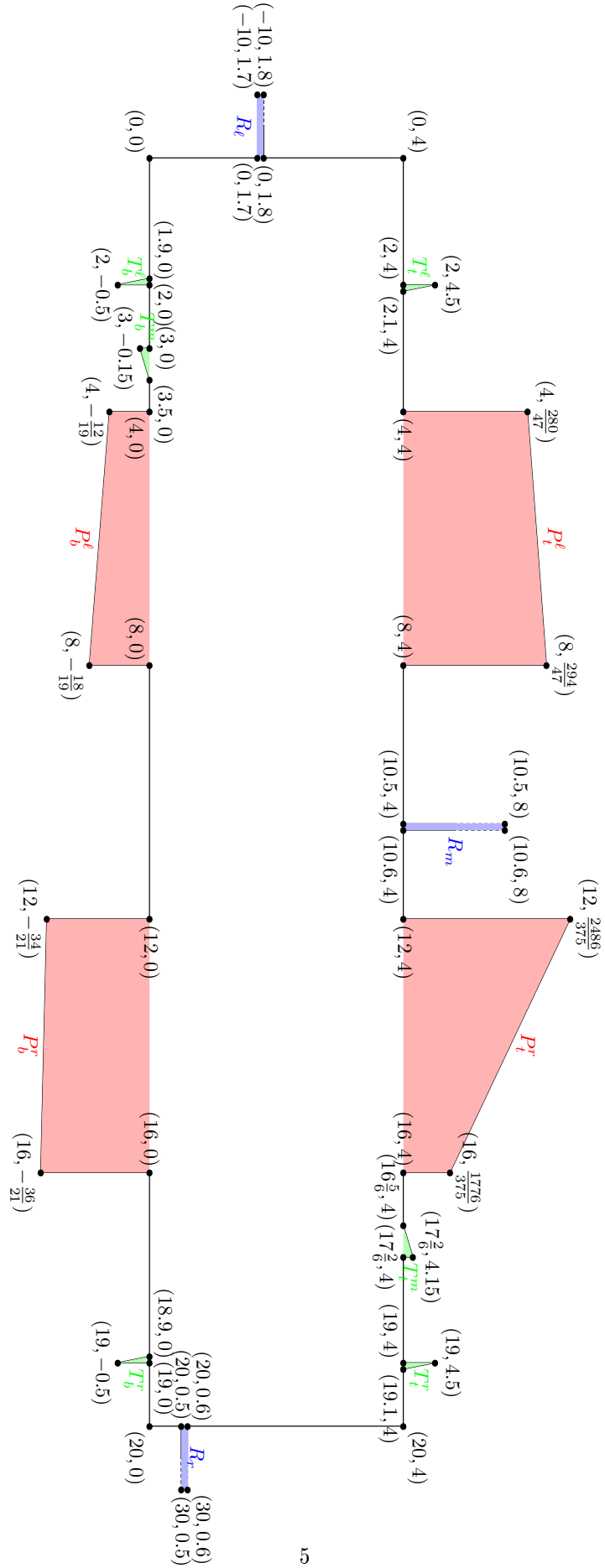
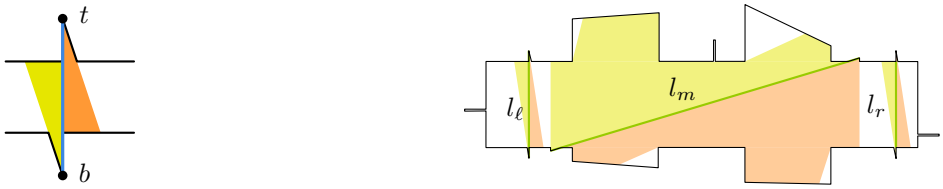


Figure 1: The polygon $\mathcal{P}$. We will show that $\mathcal{P}$ can be guarded by three guards only when we allow the guards to be placed at points with irrational coordinates. For practical reasons, the blue rectagular pockets are drawn shorter than they actually are.



(a) The only way that one guard can see both t and b is when the guard is on the blue line.

(b) The only way to guard the polygon with three guards requires one guard on each of the green lines l_ℓ, l_m, l_r.

Figure 2: Forcing guards to lie on specific line segments.

Notice that in this way we can only enforce a guard to be on a rational line, as the line contains vertices of the polygon, which are rational points.

Restricting a Guard to a Region Bounded by a Curve. For the following discussion, see the Figure 3 and notation therein. We want to guard the polygon from Figure 3 using two guards, g_1 and g_2. We assume that g_1 is forced to be on the blue vertical line l.

Consider some position of g_1 on l, such that g_1 can see at least one point of the top edge e_t of the top quadrilateral pocket, and at least one point of the bottom edge e_b of the bottom quadrilateral pocket. Let p_t and p_b denote the leftmost points seen by g_1 on e_t and e_b, respectively. Observe that p_t moves to the right if g_1 moves up, and to the left if g_1 moves down. The point p_b behaves in the opposite way when g_1 is moved. Consider some fixed position of g_1 on the blue line, and the corresponding positions of p_t and p_b. Let b be the bottom right corner of the top pocket, and d the top right corner of the bottom pocket. Let i be the intersection point of the line containing p_t and b, with the line containing p_b and d. The points b, d, i define a triangular region Δ. It is clear that if we place the guard g_2 anywhere inside Δ, then g_1 and g_2 will together see the entire polygon. On the other hand, if we place g_2 to the right of Δ, then g_1 and g_2 will not see the entire polygon, as some part of the top or the bottom pocket will not be seen.

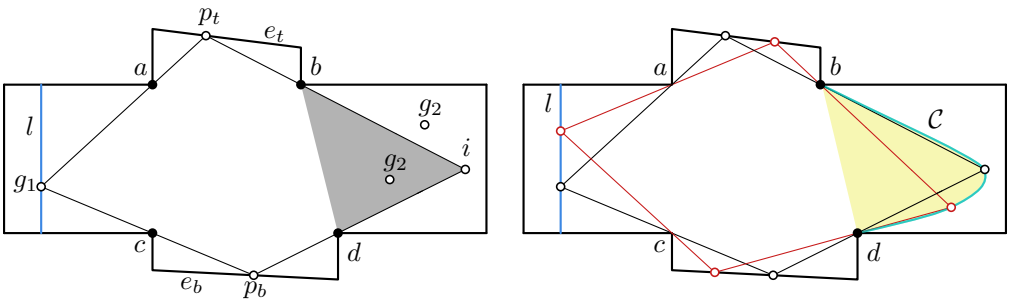


Figure 3: Left: The guard g_2 must be inside the triangular region (or to the left of it) in order to guard the entire part of the polygon that is not seen by g_1. Right: All possible positions of the point i define a simple curve $\mathcal{C}$.

Now, let us move the guard g_1 along the blue line. Each position of g_1 yields some intersection point i. We denote the union of all these intersection points by $\mathcal{C}$ (see the right picture in Figure 3). It is easy to see that $\mathcal{C}$ is a simple curve. We can compute a parameterization of $\mathcal{C}$ since we have described how to construct the point i as a function of the position of g_1.

Note that g_2 sees a larger part of *both* pockets if it is moved horizontally to the left and a smaller part of *both* pockets if it is moved horizontally to the right. Consider a fixed position of g_2 on or to the right of the segment bd. Let g_2' be the horizontal projection of g_2 on $\mathcal{C}$. Let g_1 be the unique position on the blue line such that g_1 and g_2' see all of the polygon. If g_2 is to the left of $\mathcal{C}$, g_2' sees less of the pockets than g_2, so g_1 and g_2 can together see everything. If g_2 is to the right of $\mathcal{C}$, g_2 sees less of the pockets than g_2' and neither the top nor the bottom pocket are completely guarded by g_1 and g_2. For any higher placement of g_1 even less of the top pocket is guarded and for any lower placement of g_1 even less of the bottom pocket is guarded. Thus, there exists no placement of g_1 such that both pockets are completely guarded by g_1 and g_2. We summarize our reasoning in the following observation.

Observation 1. *Consider a fixed position of g_2 on or to the right of the segment bd. There exists a position of g_1 on l such that the entire polygon is seen by g_1 and g_2 if and only if g_2 lies on or to the left of the curve $\mathcal{C}$.*

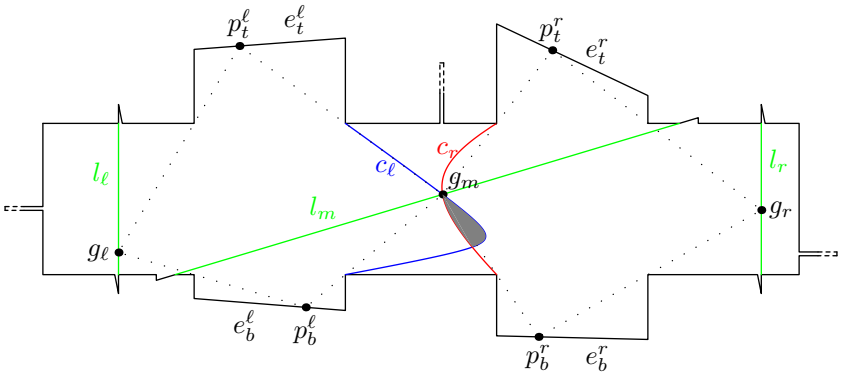


Figure 4: The polygon $\mathcal{P}$.

Restricting a Guard to a Single (Irrational) Point. For this paragraph, let us consider the polygon $\mathcal{P}$ introduced in Section 2, and consider a guard set for $\mathcal{P}$ consisting of three guards. The polygon $\mathcal{P}$ is drawn again in Figure 4, together with additional labels and information. The three guards g_ℓ, g_m, g_r are forced by the triangular pockets to lie on the three green lines l_ℓ, l_m, l_r, respectively. Additionally, the three rectangular pockets R_ℓ, R_m, R_r force the guards to lie within one of three short intervals within each line. (These properties of our construction will be discussed in more detail in Section 4.) With these restrictions, we will show that for the three guards to see the whole polygon, it must hold that the guards g_ℓ and g_m can together see the left pockets P_t^ℓ and P_b^ℓ, and the guards g_m and g_r can together see the right pockets P_t^r and P_b^r.

Then, the curve c_ℓ bounds from the right the feasible region for the guard g_m, such that g_ℓ and g_m can together see the left pockets P_t^ℓ and P_b^ℓ. Similarly, the curve c_r bounds from the left the feasible region for the guard g_m, such that g_r and g_m can together see the right pockets P_t^r and P_b^r. Thus, the only way that g_ℓ, g_m, and g_r can see the whole polygon is when g_m is within the grey region, between c_ℓ and c_r. Our idea is to define the line l_m so that it contains an intersection point of c_ℓ and c_r, and it does not enter the interior of the grey region. A simple computation with sage [14] outputs equations defining the two curves:

$$c_\ell : 138x^2 - 568xy - 1071y^2 - 3018x + 8828y + 15312 = 0 \ ,$$

$$c_r : 138x^2 - 156xy - 356y^2 - 1791x + 3296y + 1620 = 0 \ .$$

See Appendix A for the sage code for this computation. It can be checked, even by hand, that the point

$$p = (3.5 + 5\sqrt{2}, 1.5\sqrt{2}) \approx (10.57, 2.12)$$

lies on both curves, and also on the line $l_m = \{\,(x, y) \,:\, y = 0.3x - 1.05\,\}$. Therefore, p is a feasible (and at the same time irrational) position for the guard g_m. Moreover, by plotting c_ℓ, c_r, and l_m in $\mathcal{P}$ as in Figure 4, we get an indication that as we traverse l_m from left to right, at the point p we exit the area where g_m and g_l can guard together the two left pockets, and at the same time we enter the area where g_m and g_r can guard together the two right pockets. Thus, the only feasible position for the guard g_m is the irrational point p. A formal proof will be given in Section 4.

Searching for the Polygon. The simplicity of the ideas behind our construction does not reflect the difficulty of finding the exact coordinates for the polygon $\mathcal{P}$. The reader might for instance presume that most other choices of horizontal pockets would work, if the line l_m is changed accordingly. However, this is not the case.

It is easy to construct the pockets so that the corresponding curves c_ℓ and c_r intersect at some point p. We expect p to be an irrational point in general, since the curves c_ℓ and c_r are defined by two second

degree polynomials, as indicated above. In our construction, we need to force g_m to be on a line l_m containing p, but we can only force g_m to be on a rational line. Hence, we require the existence of a rational line l_m that contains p.

As any two rational lines intersect in a rational point, there can be at most one rational line containing the irrational point p. Moreover, there exists a rational line containing p if and only if $p = (r_1 + r_2\alpha, r_3 + r_4\alpha)$ for some $r_1, r_2, r_3, r_4 \in \mathbb{Q}$, where $\alpha \in \mathbb{R} \setminus \mathbb{Q}$ is an irrational number. The equation of the rational line containing p is then $y = \frac{r_4}{r_2} \cdot x + (r_3 - r_1 \cdot \frac{r_4}{r_2})$. We say that this line *supports* p. Therefore, we should not hope that the intersection point of the curves c_ℓ and c_r defined by arbitrarily chosen pockets will have a supporting line. Our main idea to overcome this problem has been to reverse-engineer the polygon, after having chosen the positions of the guards. We chose three irrational guards, all with supporting rational lines, and then defined the pockets so that g_m automatically became the intersection point between the curves c_ℓ and c_r associated with the pockets.

We chose all three guards to have coordinates of the form $(r_1 + r_2\sqrt{2}, r_3 + r_4\sqrt{2})$ for $r_1, r_2, r_3, r_4 \in \mathbb{Q}$. Assume, for the ease of presentation, that we already know that we can end up with a polygon described as follows. (In our initial attempts, our polygons were much less regular.) The polygon should consist of the rectangle $R = [0, 20] \times [0, 4]$ with some pockets added. We would like the pockets to extrude vertically from the horizontal edges of R such that the pockets meet R along the segments $(4, 0)(8, 0)$, $(12, 0)(16, 0)$, $(4, 4)(8, 4)$, and $(12, 4)(16, 4)$, respectively.

We now explain the technique for constructing the bottom pocket to the left which should extrude from R vertically downwards from the corners $(4, 0)$ and $(8, 0)$. We have to define the edge e_b^ℓ, which is the bottom edge in the pocket. We want p_b^ℓ to be a point on e_b^ℓ such that g_ℓ can only see the part of e_b^ℓ from p_b^ℓ and to the right, whereas g_m can only see the part of e_b^ℓ from p_b^ℓ and to the left. Therefore, we define p_b^ℓ to be the intersection point between the line containing g_ℓ and $(4, 0)$, and the line containing g_m and $(8, 0)$. It follows that p_b^ℓ is of the form $(r_1 + r_2\sqrt{2}, r_3 + r_4\sqrt{2})$ for some $r_1, r_2, r_3, r_4 \in \mathbb{Q}$. Hence, there is a unique rational line l supporting p_b^ℓ, and e_b^ℓ must be a segment on l. We therefore need that both of the points $(4, 0)$ and $(8, 0)$ are above l, since otherwise we do not get a meaningful polygon. However, this is not the case for arbitrary choices of the guards g_ℓ and g_m. The other pockets add similar restrictions to the positions of the guards.

In the construction we had to take care of other issues as well. In particular, the line l_m which supports the guard g_m cannot enter the grey region between the two curves c_ℓ and c_r, as otherwise the position of g_m would not be unique, and the guard could be moved to a rational point. Also, the three lines l_ℓ, l_m, l_r supporting the three guards g_ℓ, g_m, g_r cannot intersect within the polygon.

We experimented with the construction in GeoGebra [2], where we had the possibility to draw the lines supporting $p_t^\ell, p_b^\ell, p_t^r, p_b^r$ and see how they were changing in an intricate way as we changed the coordinates of the guards. For most choices of the guards and other parts of the polygon, we did not get meaningful results. The great advantage of GeoGebra is that we could continuously vary all parts of the polygon and play around with all parameters, thus gaining an intuitive understanding of various dependencies. After experimenting for a while, we were able to produce feasible examples and then find more appealing examples with simpler coordinates etc. In particular, it was important to us that many edges of the polygon are axis-parallel, so that we could easier derive from our example a rectilinear polygon with the same property, i.e., that the optimal guard set requires points with irrational coordinates.

4 Proof of Theorems 1 and 2

Basic observations. Recall the construction of the polygon $\mathcal{P}$ as defined in Section 2, and consider a guard set of $\mathcal{P}$ of cardinality at most 3. Let l_ℓ, l_m, l_r be, respectively, the restrictions of the following lines to $\mathcal{P}$:

$$x = 2, \quad y = 0.3x - 1.05, \quad \text{and} \quad x = 19.$$

As argued in Section 3, the triangular pockets enforce a guard onto each of these lines.

Lemma 4. *Consider any guard set S for $\mathcal{P}$ consisting of at most 3 guards. Then (i) $|S| = 3$, and (ii) there is one guard on each of the lines l_ℓ, l_m, l_r.*

Proof. Each triangular pocket $T_t^\ell, T_b^\ell, T_t^m, T_b^m, T_t^r, T_b^r$ has one vertex which is not on the basic rectangle $[0, 20] \times [0, 4]$. For each triangular pocket, we consider the points in $\mathcal{P}$ that can see that vertex. These positions correspond to the areas pictured in yellow and orange in Figure 2b.

It is straightforward to check that the only positions of guards that can see two such vertices are on the segments l_ℓ, l_m, l_r. Since these segments are non-intersecting, at least three guards are needed to see the whole polygon $\mathcal{P}$. If there are three guards, then there must be one guard on each of the segments l_ℓ, l_m, l_r. □

Now, consider the intervals $i_1 = [0.5, 0.6]$ and $i_2 = [1.7, 1.8]$. Similarly as for the case of triangular pockets, we can show that rectangular pockets R_ℓ, R_m, R_r enforce a guard with an x-coordinate in $[10.5, 10.6]$, and two remaining guards with y-coordinates in i_1 and i_2.

Lemma 5. *Consider any guard set S for $\mathcal{P}$ consisting of 3 guards. Then one of the guards has an x-coordinate in $[10.5, 10.6]$. For the remaining two guards, one has a y-coordinate in i_1 and the other one in i_2.*

Proof. From Lemma 4, there must be one guard g_ℓ on l_ℓ, one guard g_m on l_m, and the last guard g_r on l_r. Recall that the rectangular pockets are as follows R_ℓ: $[-10, 0] \times [1.7, 1.8]$, R_r: $[20, 30] \times [0.5, 0.6]$, and R_m: $[10.5, 10.6] \times [4, 8]$. It is straightforward to check that none of the guards g_ℓ, g_r can see the two top vertices of the pocket R_m. Therefore, the middle guard g_m has to see both these vertices and it must have an x-coordinate in $[10.5, 10.6]$.

Then, as $g_m \in l_m$, the y-coordinate of g_m is in $[2.1, 2.13]$. Therefore, g_m cannot see any of the left vertices of R_ℓ, or any of the right vertices of R_r. These four vertices must be seen by the guards g_ℓ and g_r.

As some guard must see the bottom-left corner of the pocket R_ℓ, it must be placed at a height of at least 1.7. Then, this guard cannot see any of the right vertices of R_r. Therefore, the last guard must see both right vertices of R_r, and its height must be within $i_1 = [0.5, 0.6]$. Then, this guard cannot see any left vertex of the pocket R_ℓ, and the second guard must see both left vertices of the pocket, and its height must be within $i_2 = [1.7, 1.8]$. □

Dependencies between guard positions. Let $\{g_\ell, g_m, g_r\}$ be a guard set of $\mathcal{P}$, with $g_\ell \in l_\ell, g_m \in l_m$, and $g_r \in l_r$. We will now analyze dependencies between the positions of the guards that are caused by the horizontal pockets of $\mathcal{P}$. Recall that the non-axis-parallel edges of these pockets are denoted by e_t^ℓ, e_t^r, e_b^ℓ, and e_b^r.

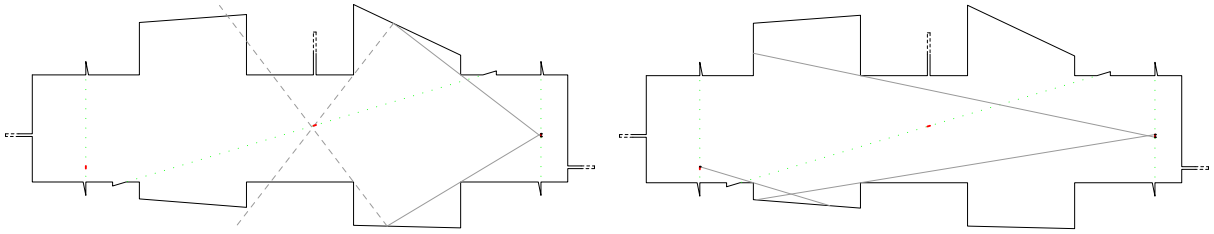
We start with two technical lemmas that are needed for Lemma 8.

Lemma 6. *Let $h \in [0, 4]$ be the height of the guard g_ℓ. If $h > \frac{135}{47} \approx 2.87$ then g_ℓ cannot see any point on e_t^ℓ, and otherwise it can see a part of e_t^ℓ starting from the x-coordinate $\frac{908-188h}{181-47h}$ and to the right of it. If $h < \frac{9}{19} \approx 0.47$ then g_ℓ cannot see any point on e_b^ℓ, and otherwise it can see a part of e_b^ℓ starting from the x-coordinate $\frac{76h+12}{19h-3}$ and to the right of it.*

Proof. Consider the guard g_ℓ and the top-left pocket. The left-most point on e_t^ℓ that g_ℓ can see is at the intersection of the following two lines: the line containing g_ℓ and the bottom-left corner of the pocket (i.e., the point $(4, 4)$), and the line containing e_t^ℓ. If $g_\ell = (2, h)$, then the equation of the first line is $y = \frac{4-h}{2} + (2h - 4)$. The second contains points $(4, \frac{280}{47})$ and $(8, \frac{294}{47})$, and its equation is $y = \frac{7}{94}x + \frac{266}{47}$. The x-coordinate of the intersection is $\frac{908-188h}{181-47h}$. It reaches a value of 8 (i.e., the point coincides with the right endpoint of e_t^ℓ) when $h = \frac{135}{47}$.

Now, consider the guard g_ℓ and the bottom-left pocket. The leftmost point on e_b^ℓ that g_ℓ can see is at the intersection of the following two lines: the line containing g_ℓ and the top-left corner of the pocket (i.e., the point $(4, 0)$), and the line containing e_b^ℓ. The first of these lines has equation $y = -\frac{h}{2}x + 2h$. The second line contains points $(4, -\frac{12}{19}), (8, -\frac{18}{19})$, and its equation is $y = -\frac{3}{38}x - \frac{6}{19}$. The x-coordinate of the intersection is $\frac{76h+12}{19h-3}$, which reaches 8 when $h = \frac{9}{19}$. □

Lemma 7. *Let $h \in [0, 4]$ be the height of the guard g_r. If $h > \frac{507}{250} = 2.028$ then g_ℓ cannot see any point on e_t^r, and otherwise it can see a part of e_t^r starting from the x-coordinate $\frac{4000h-9768}{250h-645}$ and to the left of it. If $h < \frac{17}{14} \approx 1.21$ then g_ℓ cannot see any point on e_b^r, and otherwise it can see a part of e_b^r starting from the x-coordinate $\frac{224h-56}{14h+1}$ and to the left of it.*



(a) g_ℓ and g_r cannot see any of the right pockets.

(b) g_ℓ and g_r cannot see any of the left pockets.

Figure 5: Showing that g_ℓ and g_r cannot see together a whole pocket. Possible positions for the guards are pictured in red.

Proof. Consider the guard g_r and the top-right pocket. The right-most point on e_t^r that g_r can see is at the intersection of the following two lines: the line containing g_r and the bottom-right corner of the pocket (i.e., the point $(16, 4)$), and the line containing e_t^r. If $g_r = (19, h)$, then the equation of the first line is $y = \frac{h-4}{3}x + \frac{76-16h}{3}$. The second contains points $(12, \frac{2486}{375})$ and $(16, \frac{1776}{375})$, and its equation is $y = -\frac{71}{150}x + \frac{4616}{375}$. The x-coordinate of the intersection is $\frac{4000h-9768}{250h-645}$. It reaches a value of 12 (i.e., the point coincides with the left endpoint of e_t^r) when $h = \frac{507}{250} = 2.028$.

Now, consider the guard g_r and the bottom-right pocket. The rightmost point on e_b^r that g_r can see is at the intersection of the following two lines: the line containing g_r and the top-right corner of the pocket (i.e., the point $(16, 0)$), and the line containing e_b^r. The first of these lines has equation $y = \frac{h}{3}x - \frac{16h}{3}$. The second line contains points $(12, -\frac{34}{21}), (16, -\frac{36}{21})$, and its equation is $y = -\frac{1}{42}x - \frac{4}{3}$. The x-coordinate of the intersection is $\frac{224h-56}{14h+1}$, which reaches 12 when $h = \frac{17}{14} \approx 1.21$. □

Lemma 8. *The guards g_ℓ and g_m together see all of e_t^ℓ and e_b^ℓ, and the guards g_m and g_r together can see all of e_t^r and e_b^r. Also, the y-coordinate of g_ℓ is in i_1, and the y-coordinate of g_r is in i_2.*

Proof. By the construction of $\mathcal{P}$, it holds that if a guard sees a point on one of the edges e_t^ℓ, e_t^r, e_b^ℓ, and e_b^r, then the guard sees an interval of the edge containing an endpoint of the edge. It now follows that if three guards together see one of these edges, then two do as well. In order to prove the lemma, it thus suffices to prove that

- g_ℓ and g_r cannot together see any of the edges e_t^ℓ, e_b^ℓ, e_t^r, and e_b^r,
- g_ℓ and g_m cannot together see any of the right edges e_t^r and e_b^r, and
- g_m and g_r cannot together see any of the left edges e_t^ℓ and e_b^ℓ.

We first show that the height h of the right guard g_r must be within i_2. From Lemma 5, the guard g_m has an x-coordinate in $[10.5, 10.6]$, and for the remaining two guards one has an y-coordinate in i_1, and the other one in i_2. If $h \notin i_2$, we would have $h \leq 0.6$, and from Lemma 7, the edge e_b^r would be completely invisible to g_r. Then, the middle guard g_m would have to see that edge entirely, i.e., it would have an x-coordinate of at least 12. As the x-coordinate of g_m is in $[10.5, 10.6]$, we get that $h \in i_2$. Also, the y-coordinate of g_ℓ must be in i_1.

We now prove that g_ℓ and g_r cannot together see any of the right edges e_t^r and e_b^r (see Figure 5a). Since $h \in i_2$, Lemma 7, gives that g_r cannot see e_t^r to the right of the point $(\frac{742}{55}, \frac{1629}{275})$, and e_b^r to the right of the point $(\frac{1736}{131}, -\frac{216}{131})$. It is now easy to verify that no point on l_ℓ can see any of these two points. Hence, g_ℓ and g_r cannot together see any of the edges e_t^r and e_b^r.

We now prove that g_ℓ and g_r cannot together see e_t^ℓ (see Figure 5b). Since the y-coordinate of g_r is in i_2, it follows that g_r does not see any point on e_t^ℓ. Since the x-coordinate of g_ℓ is less than 4, neither g_ℓ nor g_r can see the left endpoint of e_t^ℓ.

To show that g_ℓ and g_r cannot together see the edge e_b^ℓ, we argue as follows (see Figure 5b). The guard g_ℓ is placed at a height of at most 0.6, and g_r at a height of at most 1.8. It follows from Lemma 6 and from elementary computations that neither of the guards can see the interval of e_b^ℓ with x-coordinates between $\frac{2076}{507} < 4.1$ and $\frac{48}{7} > 6.8$.

As the x-coordinate of both g_ℓ and g_m is smaller than 12, none of these guards can see the left endpoint of the edges e_t^r, e_b^r. Therefore, g_ℓ and g_m cannot together see any of the edges e_t^r, e_b^r. Similarly, as the x-coordinates of g_m and g_r are greater than 8, g_m and g_r cannot together see e_t^ℓ or e_b^ℓ. This completes our proof. □

Lemma 9. *The maximum x-coordinate of g_m such that g_ℓ and g_m can together see e_t^ℓ and e_b^ℓ is $x = 3.5 + 5\sqrt{2}$. The corresponding position of g_ℓ is $(2, 2-\sqrt{2})$.*

Proof. Consider the guard g_ℓ at position $(2, h)$. From Lemma 8, we know that $h \in i_1 = [0.5, 0.6]$. If g_m and g_ℓ together see e_t^ℓ, we know from Lemma 6 in the appendix that g_m has to be on or below the line containing the vertices $(8, 4)$ and $(\frac{908-188h}{181-47h}, \frac{7}{94} \cdot \frac{908-188h}{181-47h} + \frac{266}{47})$, i.e., the line with equation $y = \frac{92-23h}{-135+47h}x + \frac{-1276+372h}{-135+47h}$. As g_m is at the line $y = 0.3x - 1.05$, its x-coordinate satisfies $0.3x - 1.05 \leq \frac{92-23h}{-135+47h}x + \frac{-1276+372h}{-135+47h}$, i.e., $x \leq \frac{28355-8427h}{2650-742h}$.

If g_m and g_ℓ together see e_b^ℓ, then g_m has to be on or above the line containing the vertices $(8, 0)$ and $(\frac{76h+12}{19h-3}, -\frac{3}{38} \cdot \frac{76h+12}{19h-3} - \frac{6}{19})$, i.e., the line with equation $y = \frac{3h}{19h-9}x - \frac{24h}{19h-9}$. Hence, the x-coordinate of g_ℓ must satisfy $0.3x - 1.05 \geq \frac{3h}{19h-9}x - \frac{24h}{19h-9}$, i.e., $x(1-h) \leq \frac{81h+189}{54}$. Therefore, since $h < 1$, we must have $x \leq \frac{81h+189}{54-54h}$.

We now know that $x \leq \min\{\frac{28355-8427h}{2650-742h}, \frac{81h+189}{54-54h}\}$. The first of the two values decreases with h, and the second one increases with h. Therefore the maximum is obtained when $\frac{28355-8427h}{2650-742h} = \frac{81h+189}{54-54h}$, i.e., for $h = 2 - \sqrt{2}$. The value of x is then $3.5 + 5\sqrt{2}$. The corresponding position of the guard g_ℓ is $(2, h) = (2, 2-\sqrt{2})$. □

Similarly, we can compute the left-most possible position of g_m such that g_m and g_r can see together both right pockets. The proof is in the appendix.

Computing the unique solution. We can now show that there is only one guard set for $\mathcal{P}$ consisting of three guards. Let us start by computing the right-most possible position of g_m such that g_ℓ and g_m can see together both left pockets.

Lemma 10. *The minimum x-coordinate of g_m such that g_r and g_m can see both e_t^r and e_b^r is $x = 3.5 + 5\sqrt{2}$. The corresponding position of g_r is $(19, 1 + \frac{\sqrt{2}}{2})$.*

Proof. Consider the guard g_r at position $(19, h)$. From Lemma 8, we know that $h \in i_2 = [1.7, 1.8]$. If g_m and g_r together see e_t^r, we know from Lemma 7 in the appendix that g_m has to be on or below the line containing the vertices $(12, 4)$ and $(\frac{4000h-9768}{250h-645}, -\frac{71}{150}\frac{4000h-9768}{250h-645} + \frac{4616}{375})$, i.e., the line with equation $y = \frac{46h-184}{250h-507}x + \frac{448h+180}{250h-507}$. As g_m is at the line $y = 0.3x - 1.05$, its x coordinate satisfies: $0.3x - 1.05 \leq \frac{46h-184}{250h-507}x + \frac{448h+180}{250h-507}$, i.e., $x \geq \frac{490h-243}{20h+22}$.

If g_m and g_r together see e_b^r, then g_m has to be on or above the line containing the vertices $(12, 0)$ and $(\frac{224h-56}{14h+1}, -\frac{1}{42}\frac{224h-56}{14h+1} - \frac{4}{3})$, i.e., the line with equation $y = \frac{6h}{17-14h}x - \frac{72h}{17-14h}$. Hence, the x-coordinate of g_r must satisfy $0.3x - 1.05 \geq \frac{6h}{17-14h}x - \frac{72h}{17-14h}$, i.e., $x \geq \frac{34h-7}{4h-2}$.

We have to minimize the value of $\max\{\frac{490h-243}{20h+22}, \frac{34h-7}{4h-2}\}$. When the value of h increases, the first of these two values increases, and the second one decreases. The minimum value is therefore obtained when $\frac{490h-243}{20h+22} = \frac{34h-7}{4h-2}$, i.e., for $h = 1 + \frac{\sqrt{2}}{2}$. The value of x is then $3.5 + 5\sqrt{2}$. □

We are now ready to prove our main theorems.

Proof of Theorem 1. Let $\mathcal{P}$ be the polygon constructed as in Section 2, and let S be a guard set for $\mathcal{P}$ consisting of at most 3 guards. From Lemma 4 we have $|S| = 3$, and there is one guard at each of the lines l_ℓ, l_m, l_r. Denote these guards by g_ℓ, g_m, g_r, respectively. From Lemma 8 we know that if g_ℓ, g_m, and g_r together see all of $\mathcal{P}$, then g_ℓ and g_m must see all of e_t^ℓ and e_b^ℓ, and g_m and g_r must see all of e_t^r and e_b^r. It then follows from Lemmas 9 and 10 that g_m must have coordinates $(3.5+5\sqrt{2}, 1.5\sqrt{2}) \approx (10.57, 2.12)$, $g_\ell = (2, 2-\sqrt{2}) \approx (2, 0.59)$, and $g_r = (19, 1+\frac{\sqrt{2}}{2}) \approx (19, 1.71)$. Thus, indeed, the guards g_ℓ, g_m, and g_r see the entire polygon $\mathcal{P}$ and are the only three guards doing so.

By scaling $\mathcal{P}$ up by the least common multiple of the denominators in the coordinates of the corners of $\mathcal{P}$, we obtain a polygon with integer coordinates. This does not affect the number of guards required to see all of $\mathcal{P}$.

In order to guard $\mathcal{P}$ using four guards with rational coordinates, we choose two rational guards $g'_{m,1}$ and $g'_{m,2}$ on l_m a little bit to the left and to the right of g_m, respectively. The guard $g'_{m,1}$ sees a little more of both of the edges e_t^ℓ and e_b^ℓ than does g_m, whereas $g'_{m,2}$ sees a little more of e_t^r and e_b^r. Therefore, we can choose a rational guard g'_ℓ on l_ℓ close to g_ℓ such that g'_ℓ and $g'_{m,1}$ together see e_t^ℓ and e_b^ℓ, and a rational guard g'_r on l_r with analogous properties. Thus, $g'_\ell, g'_{m,1}, g'_{m,2}, g'_r$ guard $\mathcal{P}$. □

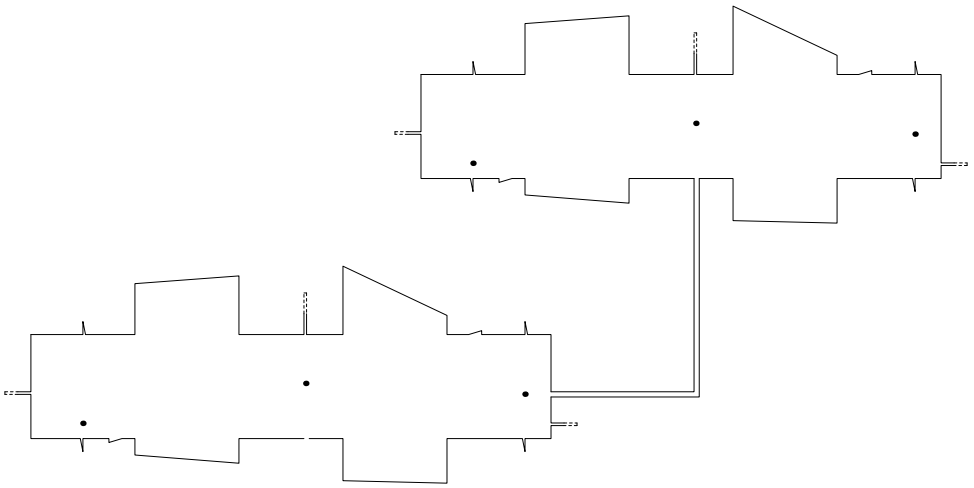


Figure 6: A sketch of a polygon that can be guarded by 6 guards when irrational coordinates are allowed, but needs 8 guards when only rational coordinates are allowed.

Proof of Theorem 2. We will now construct a polygon $\mathcal{P}_n$ that can be guarded by $3n$ guards placed at points with irrational coordinates, but such that when we restrict guard positions to points with rational coordinates, the minimum number of guards becomes $4n$. We start by making n copies of the polygon $\mathcal{P}$ described above, which we denote by $\mathcal{P}^{(1)}, \ldots, \mathcal{P}^{(n)}$. We connect the copies into one polygon $\mathcal{P}_n$ as follows. Each consecutive pair $\mathcal{P}^{(i)}, \mathcal{P}^{(i+1)}$ is connected by a thin corridor consisting of a horizontal piece $H^{(i)}$ visible by the rightmost guard in $\mathcal{P}^{(i)}$, and a vertical piece $V^{(i)}$ visible to the middle guard in $\mathcal{P}^{(i+1)}$ (see Figure 6 for the case $n = 2$). We can then guard $\mathcal{P}_n$ using $3n$ guards, by placing three guards within each polygon $\mathcal{P}^{(i)}$ in the same way as for $\mathcal{P}$, i.e., at irrational points.

Now, assume that $\mathcal{P}_n$ can be guarded by at most $4n-1$ guards. We will show that at least one guard must be irrational. For formal reasons, we define $H^{(0)} = V^{(0)} = H^{(n)} = V^{(n)} = \emptyset$. The horizontal and vertical corridors $H^{(i)}$ and $V^{(i)}$, for $i \in \{0, \ldots, n\}$, intersect at a rectangular area $B^{(i)} = H^{(i)} \cap V^{(i)}$ which we call a *bend*. For $i \in \{1, \ldots, n-1\}$, the bend $B^{(i)}$ is non-empty and visible from both polygons $\mathcal{P}^{(i)}$ and $\mathcal{P}^{(i+1)}$. Define the *extension* of $\mathcal{P}^{(i)}$, denoted by $E(\mathcal{P}^{(i)})$, to be the union of $\mathcal{P}^{(i)}$ and the adjacent corridors excluding the bends, i.e., $E(\mathcal{P}^{(i)}) = \mathcal{P}^{(i)} \cup (V^{(i-1)} \setminus B^{(i-1)}) \cup (H^{(i)} \setminus B^{(i)})$. Since the extensions are pairwise disjoint, there is an extension $E(\mathcal{P}^{(i)})$ containing at most three guards. If there are no guards in any of the bends $B^{(i-1)}, B^{(i)}$ it follows from Theorem 1 that three guards must be placed inside $\mathcal{P}^{(i)}$ at irrational coordinates, so assume that there is a guard in one or both of the bends. If the adjacent corridors $V^{(i-1)}$ and $H^{(i)}$ are long enough and thin enough, a guard in the bends $B^{(i-1)}$ and $B^{(i)}$ cannot see any left corner of any of the vertical pockets of $\mathcal{P}^{(i)}$, any point in a triangular pocket, or any point in a horizontal pocket. Hence, all the features of $\mathcal{P}^{(i)}$ that enforce the irrationality of the guards are unseen by the guards in the bends and it follows that there must be irrational guards in $\mathcal{P}^{(i)}$. Therefore, at least $4n$ guards are needed if we require them to be rational. Similarly as in the proof of Theorem 1, we can show that $4n$ rational guards are enough to guard $\mathcal{P}_n$. □

5 Rectilinear Polygon

Figure 7 depicts a rectilinear polygon $\mathcal{P}_R$ with corners at rational coordinates that can be guarded by 9 guards, but requires 10 guards if we restrict the guards to points with rational coordinates. Before the formal proof, we want to give the reader a short overview. The construction of $\mathcal{P}_R$ starts with the

polygon $\mathcal{P}$ from Theorem 1. We will extend the non-rectilinear parts by "equivalent" rectilinear parts, colored gray in the figure. The rectilinear pockets will be constructed in such a way, that each of them will require at least one guard in the interior. Additionally, if the interior of each pocket contains only one guard, then these guards must be placed at specific positions, making the area not seen by these six additional guards exactly the polygon $\mathcal{P}$ described in Section 2 (the white area in Figure 7). Thus, the remaining 3 guards must be placed at three irrational points by Theorem 1.

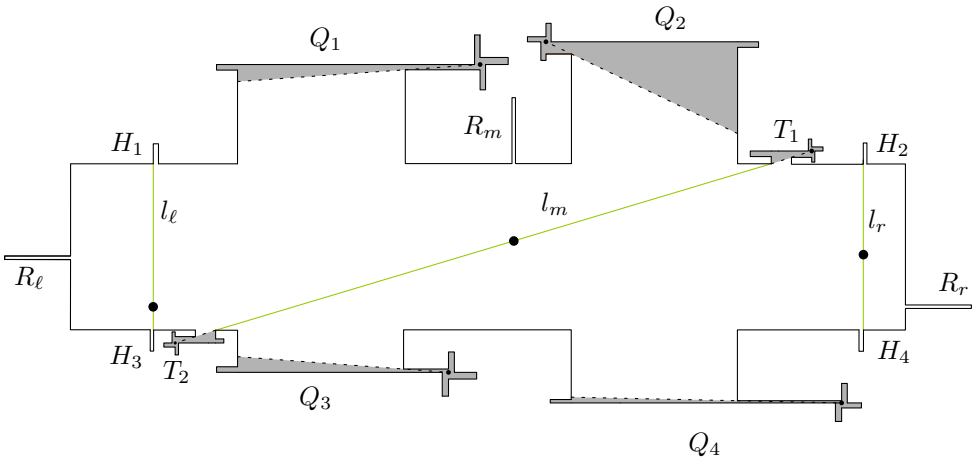


Figure 7: The rectilinear polygon $\mathcal{P}_R$ can be guarded with 9 guards only when we allow placing guards at irrational points.

Proof of Theorem 3. We describe a polygon $\mathcal{P}_R$ with vertices at integer coordinates that can be guarded by 9 guards with irrational coordinates, but needs 10 guards if only rational coordinates are allowed.

The construction of $\mathcal{P}_R$ starts with the polygon $\mathcal{P}$ from Theorem 1. We will replace the non-rectilinear parts by "equivalent" rectilinear parts, see Figure 7 for an illustration of the complete polygon $\mathcal{P}_R$ and the notation therein. The additional areas need to be guarded by additional guards, as will be described later.

First, consider the triangular pockets of the polygon $\mathcal{P}$. These pockets have been added to enforce the guards to be on the lines l_ℓ, l_m, l_r. Four of these pockets, the ones corresponding to l_ℓ and l_r, can be easily replaced by corresponding rectilinear pockets denoted by H_1, H_2, H_3, H_4, where three vertices of the new rectilinear pockets are the same as the vertices of the original triangular pockets. This does not work for the pockets corresponding to the line l_m, as this line is not axis-parallel, in particular, a guard on the line l_m would not see all of the interior of such rectangular pockets.

The two triangular pockets corresponding to l_m and the four quadrilateral pockets will be extended to new, more complicated pockets. Note that there are only two different kinds of pockets that need to be extended, *triangular pockets* and *quadrilateral pockets*, as pictured on the left of Figure 8 and Figure 9. Each triangular pocket is defined by three vertices and one of the sides of each pocket is not axis-parallel. Similarly, each quadrilateral pocket is defined by four vertices and one of the sides of each pocket is not axis-parallel.

Consider a pocket P, which needs to be extended in order to become rectilinear. Our extensions are pictured in the middle of Figure 8 and 9. The green area in the middle of Figure 8 and 9 is a newly-created pocket Q. For now, let us assume that Q does not intersect other parts of the polygon. The pocket Q satisfies the following properties (see the right pictures in Figure 8 and 9).

(a) There are four points $p_1(Q), p_2(Q), p_3(Q), p_4(Q)$ within Q, such that each of them can only be seen by a guard, which is inside Q.

(b) There exists exactly one point $q(Q)$ that can see all four points $p_1(Q), p_2(Q), p_3(Q), p_4(Q)$.

(c) The point $q(Q)$ sees exactly the interior of Q.

(d) All vertices of Q are rational.

We now show that a pocket Q satisfying all these properties can be constructed. First, we extend the non-axis-parallel edge of the pocket P in the direction outside the polygon and place a point $q = q(Q)$, with rational coordinates, on it. We let p_1, p_2, p_3, p_4 be points with rational coordinates directly above, to the right, below, and to the left of q, respectively. Then, we construct four rectilinear sub-pockets each with a vertex at one of the points p_1, p_2, p_3, p_4, so that all these can be seen by q. These pockets can also be constructed with rational coordinates because q has rational coordinates. Clearly, we can choose the point q close enough to P so that the resulting pocket Q does not intersect the rest of the polygon.

Let $\mathcal{P}_R$ be the constructed rectilinear polygon as pictured in Figure 7, where all triangular and horizontal pockets have been extended by rectilinear pockets. We have to show that $\mathcal{P}_R$ can be guarded by 9 guards, but that we need 10 guards if we require the guards to be at rational coordinates. The underlying idea is that after an optimal placement of one guard in each of the six pockets that have been extended in order to become rectilinear, the remaining area that must be seen by the remaining guards is exactly the same as in the original polygon $\mathcal{P}$.

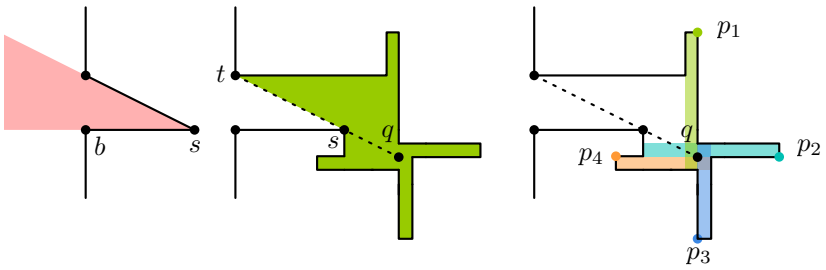


Figure 8: A triangular pocket is extended into a new rectilinear pocket.

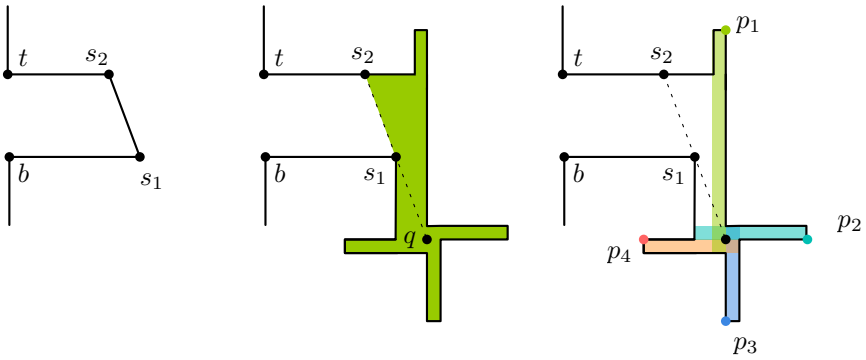


Figure 9: A quadrangular pocket is extended into a new rectilinear pocket.

We first present a solution with 9 guards when we are allowed to place guards at points with irrational coordinates. For this we place guards at the points $q(T_1)$, $q(T_2)$, $q(Q_1)$, $q(Q_2)$, $q(Q_3)$, $q(Q_4)$ so that the interior of each of the pockets $T_1, T_2, Q_1, Q_2, Q_3, Q_4$ is seen, see Property (b) and (c). Then we cover the remaining part of the polygon with three irrational guards as described in the proof of Theorem 1.

It remains to show that 10 guards are required when we restrict the guards to have rational coordinates. Suppose for the purpose of contradiction that there is a solution with 9 rational guards. Note that there must be at least one guard in each pocket $Q \in \{T_1, T_2, Q_1, Q_2, Q_3, Q_4\}$ because of Property (a). We will now show that there must be at least three guards placed outside of $T_1 \cup T_2 \cup Q_1 \cup Q_2 \cup Q_3 \cup Q_4$. First, notice that no guard placed in any of the pockets T_1, T_2, Q_1, Q_2, Q_3, Q_4 can see any of the following points: the top-left vertex of H_1 and H_2, and the bottom-right vertex of H_3 and H_4. To see these four points, at least two guards are needed. If there are only two guards, one of them must lie on l_ℓ, and the other on l_r. But then none of the guards placed on $l_\ell \cup l_r \cup T_1 \cup T_2 \cup Q_1 \cup Q_2 \cup Q_3 \cup Q_4$ can see the top edge of the pocket R_m, and one more guard is needed. Therefore, at least three guards must be placed outside of $T_1 \cup T_2 \cup Q_1 \cup Q_2 \cup Q_3 \cup Q_4$.

When only 9 guards are available, there must be exactly 3 guards outside the pockets T_1, T_2, Q_1,

Q_2, Q_3, Q_4, and exactly one guard inside each pocket $Q \in \{T_1, T_2, Q_1, Q_2, Q_3, Q_4\}$. As each pocket Q contains exactly one guard, then this guard must be the point $q(Q)$ because of Property (a) and (b). Let $\mathcal{P}_R^*$ be the area unseen by the guards within the six pockets. This polygon is exactly $\mathcal{P}$, and by Theorem 1 the unique solution with three guards is irrational. A polygon with integer coordinates can then be obtained by multiplying all coordinates with the least common multiple of all denominators of the coordinates. □

6 Future Work

One of the most prominent open questions related to the art gallery problem is whether the problem is in NP. Recently, some researchers popularized an interesting complexity class, called $\exists\mathbb{R}$, being somewhere between NP and PSPACE [8, 9, 23, 27]. Many geometric problems for which membership in NP is uncertain have been shown to be complete for the complexity class $\exists\mathbb{R}$. Famous examples are: order type realizability, pseudoline stretchability, recognition of segment intersection graphs, recognition of unit disk intersection graphs, recognition of point visibility graphs, minimizing rectilinear crossing number, linkage realizability. This suggests that there might indeed be no polynomial sized witness for any of these problems as this would imply NP = $\exists\mathbb{R}$. It is an interesting open problem whether the art gallery problem is $\exists\mathbb{R}$-complete or not.

The irrational coordinates of the guards in our examples are all of degree 2, i.e., they are roots in second-degree polynomials with integer coefficients. We would like to know if polygons exist where irrational numbers of higher degree are needed in the coordinates of an optimal solution.

We have constructed a simple polygon requiring three guards placed at points with irrational coordinates. It is a natural question whether there exists a polygon which can be guarded by two guards only if they are placed at points with irrational coordinates.

We show that there exists polygons for which $|OPT_{\mathbb{Q}}| \geq \frac{4}{3}|OPT|$. It follows from the work by Bonnet and Miltzow [5] that it always holds that $|OPT_{\mathbb{Q}}| \leq 9|OPT|$. It is interesting to see if any of these bounds can be improved.

As previously mentioned, there is an algorithm running in time $n^{O(k)}$ that finds a guard set consisting of k guards if such guard set exists. We would like to know if an algorithm exists for the same problem when we restrict ourselves to guards with coordinates in $\mathbb{Q}$.

Acknowledgement. We want to thank Sándor Fekete, Frank Hoffmann, Udo Hoffmann, Linda Kleist, Péter Kutas, Günter Rote and Andrew Winslow for discussions on the problem and links to the literature. Special thanks goes to Michał Adamaszek for providing the sage code.

Mikkel Abrahamsen is partially supported by Mikkel Thorup's Advanced Grant from the Danish Council for Independent Research under the Sapere Aude research career programme. Anna Adamszek is supported by the Danish Council for Independent Research DFF-MOBILEX mobility grant. Tillmann Miltzow is supported by the ERC grant "PARAMTIGHT: Parameterized complexity and the search for tight complexity results", no. 280152. We want to further thank the developers of the software GeoGebra. Being able to do computations and visualize parameter changes in real time facilitated our search tremendously.

References

[1] Pankaj Kumar Agarwal, Kurt Mehlhorn, and Monique Teillaud. Dagstuhl Seminar 11111, Computational Geometry, March 13 – 18 , 2011.

[2] GeoGebra 5.0. `http://www.geogebra.org`, 2016.

[3] Saugata Basu, Richard Pollack, and Marie-Françoise Roy. *Algorithms in real algebraic geometry.* Springer-Verlag Berlin Heidelberg.

[4] Patrice Belleville. Computing two-covers of simple polygons. Master's thesis, McGill University, 1991.

[5] Édouard Bonnet and Tillmann Miltzow. An approximation algorithm for the art gallery problem. *CoRR*, abs/1607.05527, 2016.

[6] Édouard Bonnet and Tillmann Miltzow. Parameterized hardness of art gallery problems. In *24th Annual European Symposium on Algorithms (ESA)*, pages 19:1–19:17, 2016. Full version available at `https://arxiv.org/abs/1603.08116`.

[7] Björn Brodén, Mikael Hammar, and Bengt J. Nilsson. Guarding lines and 2-link polygons is APX-hard. In *Proceedings of the 13th Canadian Conference on Computational Geometry (CCCG)*, pages 45–48, 2001.

[8] John Canny. Some algebraic and geometric computations in PSPACE. In *Proceedings of the twentieth annual ACM symposium on Theory of computing (STOC)*, pages 460–467. ACM, 1988.

[9] Jean Cardinal. Computational geometry column 62. *SIGACT News*, 46(4):69–78, December 2015.

[10] Dmitry Chistikov, Stefan Kiefer, Ines Marušić, Mahsa Shirmohammadi, and James Worrell. Nonnegative matrix factorization requires irrationality. *CoRR*, abs/1605.06848, 2016.

[11] Dmitry Chistikov, Stefan Kiefer, Ines Marusic, Mahsa Shirmohammadi, and James Worrell. On Restricted Nonnegative Matrix Factorization. In *43rd International Colloquium on Automata, Languages, and Programming (ICALP 2016)*, volume 55 of *LIPIcs*, pages 103:1–103:14, 2016.

[12] Václav Chvátal. A combinatorial theorem in plane geometry. *Journal of Combinatorial Theory, Series B*, 18(1):39–41, 1975.

[13] Pedro Jussieu de Rezende, Cid C. de Souza, Stephan Friedrichs, Michael Hemmer, Alexander Kröller, and Davi C. Tozoni. Engineering art galleries. In *Algorithm Engineering: Selected Results and Surveys*, LNCS, pages 379–417. Springer, 2016.

[14] The Sage Developers. *SageMath, the Sage Mathematics Software System (Version 7.4)*, 2016. `http://www.sagemath.org`.

[15] Alon Efrat and Sariel Har-Peled. Guarding galleries and terrains. *Inf. Process. Lett.*, 100(6):238–245, 2006.

[16] Stephan Eidenbenz, Christoph Stamm, and Peter Widmayer. Inapproximability results for guarding polygons and terrains. *Algorithmica*, 31(1):79–113, 2001.

[17] Sándor Fekete. private communication.

[18] Steve Fisk. A short proof of Chvátal's watchman theorem. *J. Comb. Theory, Ser. B*, 24(3):374, 1978.

[19] Stephan Friedrichs, Michael Hemmer, James King, and Christiane Schmidt. The continuous 1.5D terrain guarding problem: Discretization, optimal solutions, and PTAS. *Journal of Computational Geometry*, 7(1):256–284, 2016.

[20] Subir Kumar Ghosh. *Visibility algorithms in the plane*. Cambridge University Press, 2007.

[21] Erik Krohn and Bengt J. Nilsson. Approximate guarding of monotone and rectilinear polygons. *Algorithmica*, 66(3):564–594, 2013.

[22] D. T. Lee and Arthur K. Lin. Computational complexity of art gallery problems. *IEEE Transactions on Information Theory*, 32(2):276–282, 1986.

[23] Jirí Matousek. Intersection graphs of segments and $\exists\mathbb{R}$. *CoRR*, abs/1406.2636, 2014.

[24] Joseph O'Rourke. *Art Gallery Theorems and Algorithms*. Oxford University Press, 1987.

[25] Joseph O'Rourke and Kenneth Supowit. Some NP-hard polygon decomposition problems. *IEEE Transactions on Information Theory*, 29(2):181–190, 1983.

[26] Günter Rote. EuroCG open problem session, 2011. See the personal webpage of Günter Rote: `http://page.mi.fu-berlin.de/rote/Papers/slides/Open-Problem_artgallery-Morschach-EuroCG-2011.pdf`.

[27] Marcus Schaefer. Complexity of some geometric and topological problems. In *International Symposium on Graph Drawing*, pages 334–344. Springer, 2009.

[28] Dietmar Schuchardt and Hans-Dietrich Hecker. Two NP-hard art-gallery problems for ortho-polygons. *Math. Log. Q.*, 41:261–267, 1995.

[29] Thomas C. Shermer. Recent results in art galleries. *Proceedings of the IEEE*, 80(9):1384–1399, 1992.

[30] Ana Paula Tomás. Guarding thin orthogonal polygons is hard. In *Fundamentals of Computation Theory*, pages 305–316. Springer, 2013.

[31] Jorge Urrutia. Art gallery and illumination problems. In J.-R. Sack and J. Urrutia, editors, *Handbook of Computational Geometry*, chapter 22, pages 973–1027. Elsevier, 2000.

A Computations

```
def colinear(A,B,C):
    return Matrix([[A[0],A[1],1],[B[0],B[1],1],[C[0],C[1],1]]).determinant()

R.<t,p1,q1,p2,q2,x,y> = QQ[]

eq1 = ideal(
colinear((2,t),(4,4),(p1,q1)),
colinear((2,t),(4,0),(p2,q2)),
colinear((4,280/47),(p1,q1),(8,294/47)),
colinear((4,-12/19),(p2,q2),(8,-18/19)),
colinear((p1,q1),(8,4),(x,y)),
colinear((p2,q2),(8,0),(x,y))
    ).elimination_ideal([t,p1,q1,p2,q2]).gens()[0]

eq2 = ideal(
colinear((19,t),(16,4),(p1,q1)),
colinear((19,t),(16,0),(p2,q2)),
colinear((16,1776/375),(p1,q1),(12,2486/375)),
colinear((16,-36/21),(p2,q2),(12,-34/21)),
colinear((p1,q1),(12,4),(x,y)),
colinear((p2,q2),(12,0),(x,y))
    ).elimination_ideal([t,p1,q1,p2,q2]).gens()[0]

print eq1
print eq2
```

C

THE ART GALLERY PROBLEM IS $\exists\mathbf{R}$-COMPLETE

The Art Gallery Problem is $\exists\mathbb{R}$-complete

Mikkel Abrahamsen[1], Anna Adamaszek[1], and Tillmann Miltzow[2]

[1]University of Copenhagen, Denmark. {miab,anad}@di.ku.dk
[2] Université libre de Bruxelles (ULB), Brussels, Belgium. t.miltzow@gmail.com

Abstract

We prove that the *art gallery problem* is equivalent under polynomial time reductions to deciding whether a system of polynomial equations over the real numbers has a solution. The art gallery problem is a classical problem in computational geometry, introduced in 1973 by Viktor Klee. Given a simple polygon $\mathcal{P}$ and an integer k, the goal is to decide if there exists a set G of k *guards* within $\mathcal{P}$ such that every point $p \in \mathcal{P}$ is seen by at least one guard $g \in G$. Each guard corresponds to a point in the polygon $\mathcal{P}$, and we say that a guard g *sees* a point p if the line segment pg is contained in $\mathcal{P}$.

The art gallery problem has stimulated a myriad of research in geometry and in algorithms. However, despite extensive research, the complexity status of the art gallery problem has not been resolved. It has long been known that the problem is NP-hard, but no one has been able to show that it lies in NP. Recently, the computational geometry community became more aware of the complexity class $\exists\mathbb{R}$. The class $\exists\mathbb{R}$ consists of problems that can be reduced in polynomial time to the problem of deciding whether a system of polynomial equations with integer coefficients and any number of real variables has a solution. It can be easily seen that NP $\subseteq \exists\mathbb{R}$. We prove that the art gallery problem is $\exists\mathbb{R}$-complete, implying that (1) any system of polynomial equations over the real numbers can be encoded as an instance of the art gallery problem, and (2) the art gallery problem is not in the complexity class NP unless NP $= \exists\mathbb{R}$. As a corollary of our construction, we prove that for any real algebraic number α there is an instance of the art gallery problem where one of the coordinates of the guards equals α in any guard set of minimum cardinality. That rules out many geometric approaches to the problem.

1 Introduction

We prove that the art gallery problem is equivalent under polynomial time reductions to deciding whether a system of polynomial equations over the real numbers has a solution.

The art gallery problem. Given a simple polygon $\mathcal{P}$, we say that two points $p, q \in \mathcal{P}$ *see each other* if the line segment pq is contained in $\mathcal{P}$. A set of points $G \subseteq \mathcal{P}$ is said to *guard* the polygon $\mathcal{P}$ if every point $p \in \mathcal{P}$ is seen by at least one guard $g \in G$. Such a set G is called a *guard set* of $\mathcal{P}$, and the points of G are called *guards*. A guard set of $\mathcal{P}$ is *optimal* if it is a minimum cardinality guard set of $\mathcal{P}$.

In the *art gallery problem* we are given an integer g and a polygon $\mathcal{P}$ with corners at rational coordinates, and the goal is to decide if $\mathcal{P}$ has a guard set of cardinality g. We consider a polygon as a Jordan curve consisting of finitely many line segments *and* the region that it encloses. The art gallery problem has been introduced in 1973 by Viktor Klee, and it has stimulated a myriad of research in geometry and in algorithms. However, the complexity status of the art gallery problem has stayed unresolved. We are going to prove that the problem is $\exists\mathbb{R}$-complete. Below, we give a formal definition of the complexity class $\exists\mathbb{R}$.

The complexity class $\exists\mathbb{R}$. The *first order theory of the reals* is a set of all true sentences involving real variables, universal and existential quantifiers, boolean and arithmetic operators, constants 0 and 1, parenthesis, equalities and inequalities $(x_1, x_2, \ldots, \forall, \exists, \wedge, \vee, \neg, 0, 1, +, -, \cdot, (\ ,\ )\ , =, <, \leq)$. A formula is called a *sentence* if it has no free variables, i.e., each variable present in the formula is bound by a quantifier. Note that within such formulas one can easily express integer constants (using binary expansion) and powers. Each formula can be converted to a *prenex form*, i.e., a form where it starts with all the quantifiers and is followed by a quantifier-free formula, by a transformation which changes the length of the formula by at most a constant factor.

The *existential theory of the reals* is a set of all true sentences of the first-order theory of the reals in prenex form with existential quantifiers only, i.e., sentences of the form

$$(\exists X_1 \exists X_2 \ldots \exists X_k)\ \ \Phi(X_1, X_2, \ldots, X_k),$$

where Φ is a quantifier-free formula of the first-order theory of the reals with variables $X_1, \ldots, X_k$. The problem ETR is the problem of deciding whether a given existential formula of the above form is true. The complexity class $\exists\mathbb{R}$ consists of all problems that are reducible to ETR in polynomial time. The most well-known problem in the complexity class $\exists\mathbb{R}$ is deciding whether a system of polynomial equations over the real numbers has a solution.

It is currently known that

$$\text{NP} \subseteq \exists\mathbb{R} \subseteq \text{PSPACE}.$$

It is not hard see that the problem ETR is NP-hard, for instance by the following reduction from 3SAT. For each boolean variable x in an instance of 3SAT, we introduce a real variable x', and require that $x' \cdot (1 - x') = 0$ in order to ensure that $x' \in \{0, 1\}$. For any clause of the 3SAT formula we construct a function which evaluates to 1 if the corresponding clause is satisfied, and to 0 otherwise. For a clause C of the form $x \vee y \vee \neg z$, the corresponding function C' is $1 - (1 - x')(1 - y')z'$. The conjunction of clauses $C_1 \wedge \ldots \wedge C_m$ is then translated to the equation $C'_1 \cdot \ldots \cdot C'_m - 1 = 0$. Clearly, a formula of 3SAT is true if and only if the constructed set of equations has a solution in $\mathbb{R}$. The containment $\exists\mathbb{R} \subseteq$ PSPACE is highly non-trivial, and it has first been established by Canny [11].

By the reduction from 3SAT to ETR sketched above we know that a problem of deciding whether a given polynomial equation over $\{0, 1\}$ with integer coefficients has a solution is NP-hard. The problem is also in NP, as a satisfying assignment clearly serves as a witness.

Therefore, NP-complete problems are the problems equivalent (under polynomial time reductions) to deciding whether a given polynomial equation over $\{0, 1\}$ with integer coefficients has a solution. A well-known $\exists\mathbb{R}$-complete problem is the problem of deciding whether a single polynomial equation $Q(x_1, \ldots, x_n) = 0$ with integer coefficients has a solution in $\mathbb{R}$ [28, Proposition 3.2]. Therefore, the $\exists\mathbb{R}$-complete problems are equivalent to deciding whether a given polynomial equation over $\mathbb{R}$ with integer coefficients has a solution.

Our results and their implications. We prove that solving the art gallery problem is, up to a polynomial time reduction, as hard as deciding whether a system of polynomial equations and inequalities over the real numbers has a solution.

Theorem 1. *The art gallery problem is $\exists\mathbb{R}$-complete, even the restricted variant where we are given a polygon with vertices at integer coordinates.*

In our construction, an ETR formula $(\exists X_1 \ldots \exists X_k)\ \Phi(X_1, \ldots, X_k)$ is transformed into an instance $(\mathcal{P}, g)$ of the art gallery problem where $g > k$. Let S_Φ denote the solution space of the formula Φ, i.e., $S_\Phi := \{x \in \mathbb{R}^k : \Phi(x)\}$. We will prove the following theorem.

Theorem 2. *Let Φ be an ETR formula with k variables. Then there is an instance $(\mathcal{P}, g)$ of the art gallery problem, and constants $c_1, d_1, \ldots, c_k, d_k \in \mathbb{Q}$, such that*

- *if Φ has a solution, then $\mathcal{P}$ has a guard set of size g, and*
- *for any guard set G of $\mathcal{P}$ of size g, there exists $(x_1, \ldots, x_k) \in S_\Phi$ such that G contains guards at positions $(c_1x_1 + d_1, 0), \ldots, (c_kx_k + d_k, 0)$.*

We get the following corollary.

Corollary 3. *Given any real algebraic number α, there exists a polygon $\mathcal{P}$ with vertices at rational coordinates such that in any optimal guard set of $\mathcal{P}$ there is a guard with an x-coordinate equal to α.*

It is a classical result in Galois theory, and has thus been known since the 19th century, that there are polynomial equations of degree five with integer coefficients which have real solutions, but with no solutions expressible by radicals (i.e., solutions that can be expressed using integers, addition, subtraction, multiplication, division, raising to integer powers, and the extraction of n'th roots). One such example is the equation $x^5 - x + 1 = 0$ [35]. It is a peculiar fact that using the reduction described in this paper, we are able to transform such an equation into an instance of the art gallery problem where no optimal guard set can be expressed by radicals.

Our result rules out many approaches to solving the art gallery problem. A natural approach to finding a guard set for a given polygon $\mathcal{P}$ is to create a candidate set for the guards, and select a guard set as a subset of the candidate set. For instance, a candidate set can consist of the vertices of $\mathcal{P}$. The candidate set can then be expanded by considering all lines containing two candidates and adding all intersection points of these lines to the candidate set. This process can be repeated any finite number of times, but only candidates with rational coordinates can be obtained that way, and the candidate set will thus not contain an optimal guard set in general. Algorithms of this kind are for instance discussed by de Rezende *et al.* [15]. One can get a more refined set of candidates by also considering certain quadratic curves [6], or more complicated curves. Our results imply that if the algebraic degree of the considered curves is bounded by a constant, a such approach cannot lead to an optimal solution in general, since the coordinates of the candidates will also have algebraic degree bounded by a constant.

Related work. The art gallery problem has been extensively studied, with some books, surveys, and book chapters dedicated to it [31, 41, 45, 33, 14, 27, 16, 32]. The research is stimulated by a large number of possible variants of the problem and related questions that can be studied. The version of the art gallery problem considered in this paper is the classical one, and it has been originally formulated by Victor Klee (see O'Rourke [31]). Other versions of the art gallery problem include restrictions on the positions of the guards, different definitions of visibility, restricted classes of polygons, restricting the part of the polygon that has to be guarded, etc.

The art gallery problem has been studied both from combinatorial and from the algorithmic perspective. Studies have been made on algorithms performing well in practice on real-world and simulated instances of the problem [9, 15]. Another branch of research investigates approximation algorithms for the art gallery problem and its variants [18, 8, 22].

The first exact algorithm for solving the art gallery problem was published in 2002 in the conference version of a paper by Efrat and Har-Peled [17]. They attribute the result to Micha Sharir. Before this time, the problem was not even known to be decidable. The approach is to reduce the art gallery problem to a formula in the first order theory of the reals and use standard algebraic methods to decide if that formula is true, such as the techniques provided by Basu *et al.* [5]. No algorithm is known that avoids the use of this powerful machinery.

Lee and Lin [25] proved, by constructing a reduction from 3SAT, that the art gallery problem is NP-hard when the guards are restricted to the vertices of the polygon. It has subsequently been shown by Aggarwal ([3], see also [31]) that this argument can be extended to the case with no restrictions on the guards. Various papers showed other hardness results or conditional lower bounds for the art gallery problem and its variations [25, 40, 44, 10, 18, 34, 24, 8, 21].

Note that the version of the art gallery problem where the guards are restricted to the vertices of the polygon is obviously in NP. Another related problem is the *terrain guarding problem*. Here, the area above an x-monotone polygonal curve c has to be guarded by guards restricted to c. Friedrichs *et al.* [19] recently showed that terrain guarding is in NP. These two problems are both NP-complete. The authors of the present paper [2] gave a simple example of a polygon with a unique optimal guard set consisting of three guards at irrational coordinates. Four guards are needed if they have to have rational coordinates. This could be an indication that the original version of the art gallery problem with no restrictions on the guards is actually more difficult than these related problems. Friedrichs *et al.* [19] stated that "[...] it is a long-standing open problem for the more general Art Gallery Problem (AGP): For the AGP it is not known whether the coordinates of an optimal guard cover can be represented with a polynomial number of bits". In the present paper, we show that such a representation does not exist and that the art gallery problem is indeed not in NP under the assumption NP $\neq \exists\mathbb{R}$.

A growing class of problems turn out to be equivalent (under polynomial time reductions) to deciding whether polynomial equations and inequalities over the reals have a solution. These problems form the family of $\exists\mathbb{R}$-complete problems as it is currently known. This class includes problems like the stretchability of pseudoline arrangements [30, 43], recognition of intersection graphs of various objects (e.g. segments [28], unit disks [29], and general convex sets [37]), recognition of point visibility graphs [13], the Steinitz problem for 4-polytopes [36], deciding whether a graph with given edge lengths can be realized by a straight-line drawing [38, 1], deciding whether a straight line drawing of a graph exists with a given number of edge crossings [7], decision problems related to Nash-equilibria [20], and positive semidefinite matrix factorization [42]. We refer the reader to the lecture notes by Matoušek [28] and surveys by Schaefer [37] and Cardinal [12] for more information on the complexity class $\exists\mathbb{R}$.

Overview of the paper and techniques. In Section 2 we show that the art gallery problem is in the complexity class $\exists\mathbb{R}$. For that we present a construction of an ETR-formula Φ for any instance $(\mathcal{P}, g)$ of the art gallery problem such that Φ has a solution if and only if $\mathcal{P}$ has a

guard set of size g. The details of the construction are in Appendix A. The idea is to encode guards by pairs of variables and compute a set of witnesses (which depend on the positions of the guards) of polynomial size such that the polygon is guarded if and only if the witnesses are seen by the guards.

The proof that the art gallery problem is $\exists\mathbb{R}$-hard is the main result of the paper, and it consists of two parts. The first part is of algebraic nature, and in that we introduce a novel $\exists\mathbb{R}$-complete problem which we call ETR-INV. A common way of making a reduction from ETR to some other problem is to build gadgets corresponding to each of the equations $x = 1$, $x + y = z$, and $x \cdot y = z$ for any variables x, y, z. Usually, the multiplication gadget is the most involved one. An instance of ETR-INV is a conjunction of formulas of the form $x = 1$, $x + y = z$, and $x \cdot y = 1$, with the requirement that each variable must be in the interval $[1/2, 2]$. In particular, the reduction from ETR-INV requires building a gadget for *inversion* (i.e., $x \cdot y = 1$) instead of a more general gadget for multiplication. The formal definition of ETR-INV and the proof that it is $\exists\mathbb{R}$-complete is presented in Section 3 (with details in Appendix B). We think that the problem ETR-INV might be of independent interest, and that it will simplify constructing $\exists\mathbb{R}$-hardness proofs. The problem ETR-INV has already been used to prove $\exists\mathbb{R}$-completeness of a geometric graph drawing problem with prescribed face areas [23].

In Section 4 (with details in Appendix C) we describe a polynomial time reduction from ETR-INV to the art gallery problem, which shows that the art gallery problem is $\exists\mathbb{R}$-hard. This reduction constructs an art gallery instance $(\mathcal{P}(\Phi), g(\Phi))$ from an ETR-INV instance Φ, such that $\mathcal{P}(\Phi)$ has a guard set of size $g(\Phi)$ if and only if the formula Φ has a solution. We construct the polygon so that it contains $g(\Phi)$ *guard segments* (which are horizontal line segments within $\mathcal{P}$) and *stationary guard positions* (points within $\mathcal{P}$). By introducing *pockets* we enforce that if $\mathcal{P}$ has a guard set of size $g(\Phi)$, then there must be exactly one guard at each guard segment and at each stationary guard position. Each guard segment represents a variable of Φ (with multiple segments representing the same variable) in the sense that the position of the guard on the segment specifies the value of the variable, the endpoints of a segment corresponding to the values $1/2$ and 2.

We develop a technique for *copying* guard segments, i.e., enforcing that the guards at two segments correspond to the same variable. We do that by introducing *critical segments* within the polygon, which can be seen by guards from two guard segments (but not from other guard segments). Then the requirement that a critical segment is seen introduces dependency between the guards at the corresponding segments. Different critical segments will enforce different dependencies, and by enforcing that two guards must see together two particular critical segments, we can ensure that the guards represent the same value. The stationary guards are placed to see the remaining areas of the polygon.

With this technique, we are able to copy two or three segments from an area containing guard segments corresponding to all variables into a *gadget*, where we will enforce a dependency between the values of the variables represented by the two or three segments. This is done by constructing a *corridor* containing two critical segments for each pair of copied segments. The construction is non-trivial, as it requires the critical segments not to be seen from any other segments.

Within the gadgets, we build features that enforce the variables x, y, z represented by the guards to satisfy one of the conditions $x + y \geq z$, $x + y \leq z$, or $x \cdot y = 1$. The conditions are enforced by a requirement that two or three guards can see together some areas, where for the case of a gadget with three variables the area to be seen is a quadrilateral instead of a line segment.

2 The art gallery problem is in $\exists\mathbb{R}$

In this section we sketch a proof that the art gallery problem is in the complexity class $\exists\mathbb{R}$. Our proof also works for the more general version of the art gallery problem where the input polygon can have polygonal holes.

Theorem 4. *The art gallery problem is in the complexity class* $\exists\mathbb{R}$.

Sketch of proof. Let $(\mathcal{P}, g)$ be an instance of the art gallery problem where the polygon $\mathcal{P}$ has n vertices, each of which has rational coordinates represented by at most B bits. Assume that $g < n$, as it is otherwise trivial that $\mathcal{P}$ can be guarded by g guards. We show how to construct in polynomial time a quantifier-free formula $\Phi := \Phi(\mathcal{P}, g)$ of the first-order theory of the reals such that Φ is satisfiable if and only if $\mathcal{P}$ has a guard set of cardinality g. It has been our priority to define the formula Φ so that it is as simple as possible to describe. It might be possible to construct an equivalent but shorter formula.

The description of Φ resembles the formula Ψ that Micha Sharir described to Efrat and Har-Peled [17]:

$$\Psi \;:=\; \left[\exists x_1, y_1, \ldots, x_g, y_g \; \forall p_x, p_y : \text{INSIDE-POLYGON}(p_x, p_y) \Longrightarrow \bigvee_{i=1}^{g} \text{SEES}(x_i, y_i, p_x, p_y)\right].$$

For each $i \in \{1, \ldots, g\}$, the variables x_i, y_i represent the position of a guard $g_i := (x_i, y_i)$, and $p := (p_x, p_y)$ represents an arbitrary point. The predicate INSIDE-POLYGON(p_x, p_y) tests if the point p is contained in the polygon $\mathcal{P}$ and SEES(x_i, y_i, p_x, p_y) checks if the guard g_i can see the point p. Thus, the formula is satisfiable if and only if $\mathcal{P}$ has a guard set of cardinality g. Note that although the implication "$\Longrightarrow$" is not allowed in the first order theory of the reals, we can always substitute "$A \Longrightarrow B$" by "$\neg A \vee B$".

The formula Ψ is not an existential formula. Our main idea behind obtaining a formula with no universal quantifier is finding a polynomial number of points inside $\mathcal{P}$ which, if all seen by the guards, ensure that all of $\mathcal{P}$ is seen. We denote such a set of points as a *witness set*.

Creating a witness set. We are now ready to describe the witness set that replaces the universal quantifier. Let $\mathcal{L} := \{\ell_1, \ldots, \ell_m\}$ be the set of lines containing either an edge of $\mathcal{P}$, or a guard g_i for $i \in \{1, \ldots, g\}$ and a vertex v of $\mathcal{P}$.* The well-defined lines in $\mathcal{L}$ partition the plane into a collection of open *regions* $\mathcal{A}$, which are connected components of $\mathbb{R}^2 \setminus \bigcup_{\ell \in \mathcal{L}} \ell$ (see Figure 1 for an example).

The set $\mathcal{A}$ has the following properties.

- Each region in $\mathcal{A}$ is an open convex polygon.
- Each region in $\mathcal{A}$ is either contained in $\mathcal{P}$ or contained in the complement of $\mathcal{P}$.
- The closure of the union of the regions that are contained in $\mathcal{P}$ equals $\mathcal{P}$.
- For each region $R \in \mathcal{A}$, each guard g_i either sees all points of R, or sees no point of R. In particular, if a guard g_i sees one point in R, it sees all of R and its closure.

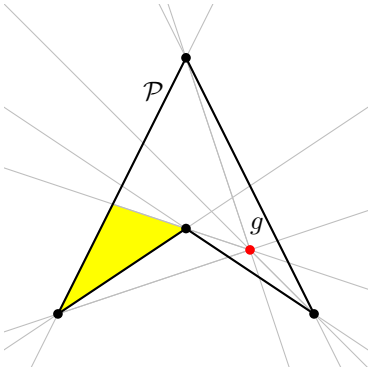


Figure 1: A polygon $\mathcal{P}$ together with a guard g and the resulting line arrangement $\mathcal{L}$. The yellow region is not seen.

*Note that if a line is defined as passing through a guard g_i and a vertex v such that g_i and v are coincident, the line is not well-defined. Such lines are not considered in the partition into regions described below, but they are included in the set $\mathcal{L}$. However, in this sketch of the proof, we will ignore them.

Thus, it is sufficient to test that for each region $R \in \mathcal{A}$ which is in $\mathcal{P}$, at least one point in R is seen by a guard. For three points a, b, c, define the *centroid* of a, b, c to be the point $C(a, b, c) := (a + b + c)/3$. If a, b, c are three different vertices of the same region $R \in \mathcal{A}$, then the centroid $C(a, b, c)$ must lie in the interior of R. Note that each region has at least three vertices and thus contains at least one such centroid. Let X be the set of all intersection points of two non-parallel, well-defined lines in $\mathcal{L}$, i.e., X consists of all vertices of all regions in $\mathcal{A}$. In the formula Φ, we generate all points in X. For any three points a, b, c in X, we also generate the centroid $C(a, b, c)$. If the centroid is in $\mathcal{P}$, we check that it is seen by a guard. Since there are $O(((gn)^2)^3)$ centroids of three points in X and each is tested by a formula of size $O(gnB)$, we get a formula of length $O(g^7 n^7 B^2) = O(n^{14} B^2)$. Clearly, the formula can be computed in polynomial time. □

3 The problem ETR-INV

To show that the art gallery problem is $\exists\mathbb{R}$-hard, we will provide a reduction from the problem ETR-INV, which we introduce below. In this section, we sketch how to show that ETR-INV is $\exists\mathbb{R}$-complete.

Definition 5 (ETR-INV)**.** *In the problem ETR-INV, we are given a set of real variables* $\{x_1, \ldots, x_n\}$*, and a set of equations of the form*

$$x = 1, \quad x + y = z, \quad x \cdot y = 1,$$

for $x, y, z \in \{x_1, \ldots, x_n\}$*. The goal is to decide whether the system of equations has a solution when each variable is restricted to the range* $[1/2, 2]$*.*

Theorem 6. *The problem ETR-INV is* $\exists\mathbb{R}$*-complete.*

Sketch of proof. To show that ETR-INV is $\exists\mathbb{R}$-hard, we perform a series of polynomial time reductions, starting from a formula $(\exists X_1 \ldots \exists X_k)\ \Phi(X_1, \ldots, X_k)$ which is an instance of the problem ETR. We use a lemma by Schaefer and Štefankovič [39] to reduce this formula to a single polynomial F with integer coefficients, so that the equation $F = 0$ has a solution if and only if Φ has a solution. Using a standard technique, we transform the equation $F = 0$ into a formula Φ' which is a conjunction of equations of the forms $x = 1$, $x + y = z$, and $x \cdot y = z$. The idea is to replace each addition or multiplication of two variables x, y in F by a new variable z, and to add to Φ' the corresponding condition $x + y = z$ or $x \cdot y = z$. The coefficients of F are decomposed in a similar way. Another standard technique is used to get an equivalent problem where the variables have been scaled down to the range $[-1/8, 1/8]$. A third standard technique is used to shift the variables to the range $[1/2, 2]$. We finally show how we can substitute each equation of the form $x \cdot y = z$ by an equivalent set of equations using only addition and inversion, i.e., equations of the forms $x + y = z$ and $x \cdot y = 1$. We explain the main ideas in the following while ignoring that the variables should stay in the range $[1/2, 2]$. In this simplified case, the result in fact follows from the proof by Aho *et al.* [4, Section 8.2] that squaring and taking reciprocals is equivalent to multiplication.

We first show how to define a new variable V_{x^2} satisfying $V_{x^2} = x^2$. The technique is based on the observation that for $x \notin \{0, 1\}$, we have $\frac{1}{x-1} - \frac{1}{x} = \frac{1}{x^2-x}$. With the equations allowed in ETR-INV we can easily construct variables $V_{\frac{1}{x-1}}$ and $V_{\frac{1}{x}}$ satisfying $V_{\frac{1}{x-1}} = \frac{1}{x-1}$ and $V_{\frac{1}{x}} = \frac{1}{x}$. The equation $V_{\frac{1}{x}} + V_{\frac{1}{x^2-x}} = V_{\frac{1}{x-1}}$ ensures that for a new variable $V_{\frac{1}{x^2-x}}$, we have $V_{\frac{1}{x^2-x}} = \frac{1}{x^2-x}$. Now, the equations $V_{x^2-x} \cdot V_{\frac{1}{x^2-x}} = 1$ and $V_{x^2-x} + x = V_{x^2}$ ensure that $V_{x^2} = x^2$.

In order to substitute equations of the form $x \cdot y = z$ with the equations allowed in ETR-INV, we observe that $(x + y)^2 - x^2 - y^2 = 2xy$. We can create variables satisfying $V_{(x+y)^2} = (x + y)^2$,

$V_{x^2} = x^2$, and $V_{y^2} = y^2$ using the technique for squaring a variable described before. It is then easy to construct a variable V_{2xy} satisfying $V_{2xy} = 2xy$.

The constructed instance of ETR-INV can be computed in polynomial time, and it has a solution if and only if the formula Φ has a solution. Therefore, ETR-INV is $\exists\mathbb{R}$-hard. As the conjunction of the equations of ETR-INV, together with the inequalities describing the restricted range $[1/2, 2]$ of the variables, is a quantifier-free formula of the first-order theory of the reals, ETR-INV is in $\exists\mathbb{R}$, which yields that ETR-INV is $\exists\mathbb{R}$-complete. □

4 Reduction from ETR-INV to the art gallery problem

Overview of the construction. Let Φ be an instance of ETR-INV consisting of k equations with n variables $X := \{x_0, \ldots, x_{n-1}\}$. We show that there exists a polygon $\mathcal{P} := \mathcal{P}(\Phi)$ with corners at rational coordinates which can be computed in polynomial time such that Φ has a solution if and only if $\mathcal{P}$ can be guarded by some number $g := g(\Phi)$ of guards. The number g will follow from the construction. A sketch of the polygon $\mathcal{P}$ is shown in Figure 2.

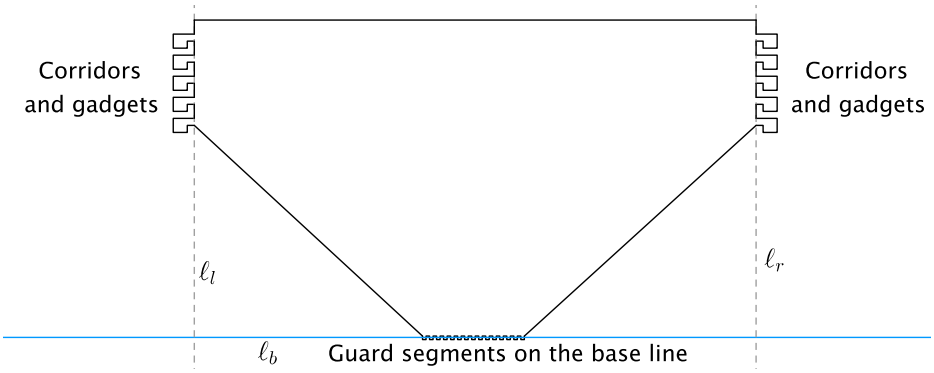


Figure 2: A high-level sketch of the construction of the polygon $\mathcal{P}$.

Each variable $x_i \in X$ is represented by a collection of *guard segments*, which are horizontal line segments contained in the interior of $\mathcal{P}$. Consider one guard segment $s := ab$, where a is to the left of b, and assume that s represents the variable x_i and that there is exactly one guard p placed on s. The guard segment s can be oriented to the right or to the left. The guard p on s specifies the value of the variable x_i as $\frac{1}{2} + \frac{3\|ap\|}{2\|ab\|}$ if s is oriented to the right, and $\frac{1}{2} + \frac{3\|bp\|}{2\|ab\|}$ if s is oriented to the left. Here, the additive term $\frac{1}{2}$ and the factor $\frac{3}{2}$ stem from the fact that all the variables in X are contained in the interval $[\frac{1}{2}, 2]$.

Suppose that there exists a solution to Φ. We will show that in that case any optimal guard set G of $\mathcal{P}$ has size $g(\Phi)$ and *specifies* a solution to Φ in the sense that it satisfies the following two properties.

- Each variable $x_i \in X$ is specified *consistently* by G, i.e., there is exactly one guard on each guard segment representing x_i, and all these guards specify the same value of x_i.
- The guard set G is *feasible*, i.e., the values of X thus specified is a solution to Φ.

Moreover, if there is no solution to Φ, each guard set of $\mathcal{P}$ consists of more than $g(\Phi)$ guards.

The polygon $\mathcal{P}$ is constructed in the following way. The bottom part of the polygon consists of a collection of *pockets*, containing in total $4n$ collinear and equidistant guard segments $s_0, \ldots, s_{4n-1}$, three right-oriented and one left-oriented segment corresponding to each variable of Φ. We denote the horizontal line containing these guard segments as the *base line* or ℓ_b. At the left and at the right side of $\mathcal{P}$, there are some *corridors* attached, each of which leads into a *gadget*. The *entrances* to the corridors at the right side of $\mathcal{P}$ are line segments contained in a vertical line ℓ_r. Likewise, the entrances to the corridors at the left side of $\mathcal{P}$ are contained

in a vertical line ℓ_l. The gadgets also contain guard segments, and they are used to impose dependencies between the guards in order to ensure that if there is a solution to Φ, then any minimum guard set of $\mathcal{P}$ consists of $g(\Phi)$ guards and specifies a solution to Φ in the sense defined above. The corridors are used to copy the positions of guards on guard segments on the base line to guards on guard segments inside the gadgets. Each gadget corresponds to a constraint of one of the types $x + y \geq z$, $x + y \leq z$, $x \cdot y = 1$, $x + y \geq 5/2$, and $x + y \leq 5/2$. The first three types of constraints are used to encode the dependencies between the variables in X as specified by Φ, whereas the latter two constraints are used to encode the dependencies between the right-oriented and left-oriented guard segments representing a single variable in X. The constraints of type $x = 1$ are enforced by modifying the pocket containing a guard segment corresponding to x.

Creating stationary guard positions and guard segments. We denote some points in $\mathcal{P}$ as *stationary guard positions*. A guard placed at a stationary guard position is called a *stationary guard*. A stationary guard position is the unique point $p \in \mathcal{P}$ such that a guard placed at p can see some set of vertices (usually two vertices) of $\mathcal{P}$. We use stationary guards for the purpose of seeing some region on one side of a line segment ℓ, but no points on the other side of ℓ. See Figure 3 (left) for an explanation of how such a construction can be made.

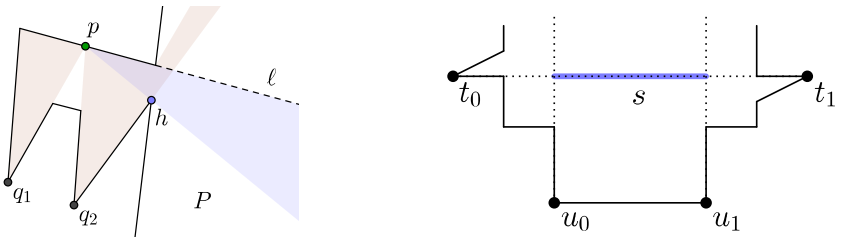


Figure 3: Left: The construction of a stationary guard position p that sees an area in P below a line segment ℓ. The brown areas are the regions of points that see q_1 and q_2, and p is the only point that sees both q_1 and q_2. The point p sees the points in the blue wedge, and the angle of the wedge can be adjusted by choosing the point h accordingly. Right: The construction of a guard segment s (the blue segment). In order to see the points t_0, t_1, a guard must be on the horizontal dotted segment. Furthermore, in order to see u_0, u_1, the guard must be between the vertical dotted segments that contain the endpoints of s. Thus, a guard sees t_0, t_1, u_0, u_1 if and only if the guard is at s.

We likewise denote some horizontal line segments of $\mathcal{P}$ as *guard segments*. A guard segment s consists of all points from which a guard can see some set of four vertices of $\mathcal{P}$. See Figure 3 (right) for an example of such a construction.

In the full version of the reduction, we will prove that for any guard set of size of at most $g(\Phi)$, there is one guard placed at each stationary guard position and at each guard segment, and there are no guards except of these positions. As explained earlier, guards placed at the guard segments will be used to encode the values of the variables of Φ.

Imposing inequalities by nooks and umbras. The nooks and umbras, which we introduce below, are our basic tools used to impose dependency between guards placed on two different guard segments. For the following definitions, see Figure 4.

Definition 7 (nook and umbra)**.** *Let $\mathcal{P}$ be a polygon with guard segments $r_0 := a_0b_0$ and $r_1 := a_1b_1$, where r_0 is to the left of r_1. Let c_0, c_1 be two vertices of $\mathcal{P}$, such that c_0 is to the left of c_1. Suppose that the rays $\overrightarrow{b_0c_0}$ and $\overrightarrow{b_1c_1}$ intersect at a point f_0, the lines $\overrightarrow{a_0c_0}$ and $\overrightarrow{a_1c_1}$ intersect at a point f_1, and that $Q := c_0c_1f_1f_0$ is a convex quadrilateral contained in $\mathcal{P}$. For each $i \in \{0, 1\}$ define the function $\pi_i \colon r_i \longrightarrow f_0f_1$ such that $\pi_i(p)$ is the intersection of the ray $\overrightarrow{pc_i}$ with the line segment f_0f_1, and suppose that π_i is bijective.*

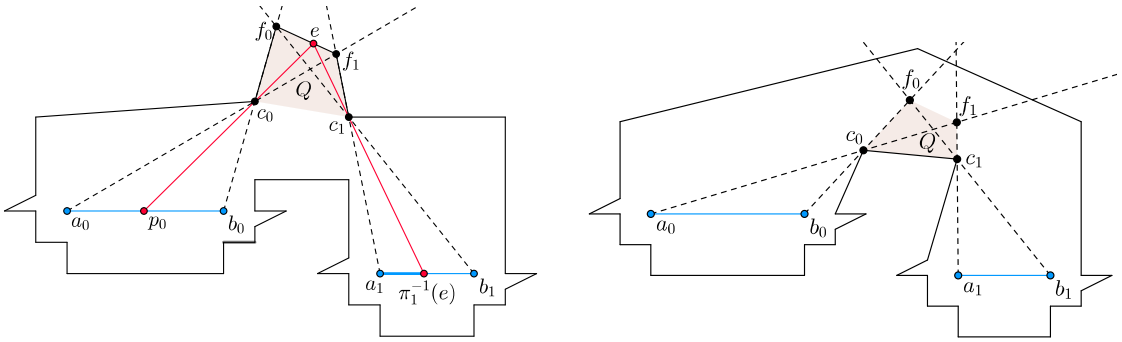


Figure 4: The brown area Q representing a nook (left), and an umbra (right). In the left figure, note that if a guard p_1 placed at the segment a_1b_1 has to see the whole line segment f_0f_1 together with p_0, then p_1 must be on or to the left of the point $\pi_1^{-1}(e)$, where $e := \pi_0(p_0)$.

We say that Q is a nook *of the pair of guard segments r_0, r_1 if for each $i \in \{0,1\}$ and every $p \in r_i$, a guard at p can see all of the segment $\pi_i(p)f_{1-i}$ but nothing else of f_0f_1. We say that Q is an* umbra *of the segments r_0, r_1 if for each $i \in \{0,1\}$ and every $p \in r_i$, a guard at p can see all of the segment $\pi_i(p)f_i$ but nothing else of f_0f_1. The functions π_0, π_1 are called* projections *of the nook or the umbra.*

Definition 8 (critical segment and shadow corners)**.** *Consider a nook or an umbra $Q := c_0c_1f_1f_0$ of a pair of guard segments r_0, r_1. The line segment f_0f_1 is called the* critical segment *of Q, and the vertices c_0, c_1 are called the* shadow corners *of Q.*

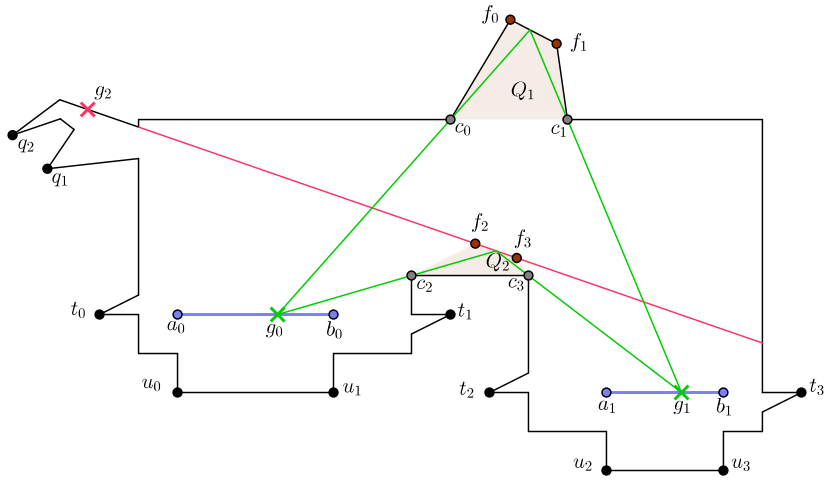


Figure 5: Q_1 is a copy-nook of the segments $r_0 := a_0b_0$ and $r_1 := a_1b_1$ with a critical segment f_0f_1, and Q_2 is a copy-umbra for the same pair with a critical segment f_2f_3. It can be seen that this polygon cannot be guarded by fewer than 3 guards, and any guard set with 3 guards must contain a guard g_0 on r_0, a guard g_1 on r_1, and a stationary guard at the point g_2. The guards g_0 and g_1 must specify the same value on r_0 and r_1, respectively.

We will construct nooks and umbras for pairs of guard segments where we want to enforce dependency between the values of the corresponding variables. When making use of an umbra, we will also create a stationary guard position from which a guard sees the whole quadrilateral Q, but nothing on the other side of the critical segment f_0f_1. In this way we can enforce the guards on r_0 and r_1 to see all of f_0f_1 together. For the case of a nook, the segment f_0f_1 will always be on the polygon boundary, and then there will be no stationary guard needed. See Figure 5 for an example of a construction of both a nook and an umbra for a pair of guard segments.

Definition 9. *Let Q be a nook or an umbra of a pair of guard segments $r_0 := a_0b_0$ and $r_1 := a_1b_1$ with the same orientation, such that the shadow corners c_0 and c_1 have the same y-coordinate. We then call Q a* copy-nook *or a* copy-umbra*, respectively.*

Consider a pair of guard segments r_0, r_1 oriented in the same way, for which there is both a copy-nook and a copy-umbra. We can show that if there is one guard placed at each segment, and the guards together see both critical segments, then the two guards specify the same value. We use this observation to make one guard segment a *copy* of another. See Figure 5, where the guard segment a_1b_1 is a copy of a_0b_0.

Corridors. Inside each gadget there are two or three guard segments r_i, r_j, r_l (or r_i, r_j) corresponding to two or three pairwise different guard segments from the base line s_i, s_j, s_l (or s_i, s_j). A corridor ensures that the segments r_i, r_j, r_l are copies of the segments s_i, s_j, s_l, respectively. To obtain that, we need to construct a copy-nook and a copy-umbra within the corridor for each pair of corresponding segments, see Figure 6 for a simplified illustration. The lower wall of the corridor of the gadget is a horizontal edge c_0, c_1. The vertices c_0, c_1 of the corridor act as shadow corners in three overlapping copy-umbras for the pairs (s_i, r_i), (s_j, r_j), and (s_l, r_l), respectively. We construct the top wall of the corridor so that it creates three copy-nooks for the same pairs. To enforce that for any guard set of size $g(\Phi)$ and each $\sigma \in \{i, j, l\}$, the guard segments s_σ and r_σ specify the same value, we have to ensure that no guards on guard segments other than s_σ and r_σ can see the critical segments of the copy-umbra and the copy-nook of the pair s_σ, r_σ. The precise construction of the corridor that ensures this property is non-trivial. The high-level idea behind the construction is as follows. By placing the corridor sufficiently far away from the segments on the base line, and by making the corridor *entrance* sufficiently small, we obtain that the visibility lines from the guard segment endpoints through the points at the corridor entrance are almost parallel and can be described by a simple pattern. Moreover, each point in the corridor which is far enough from the corridor entrances can be seen by points from at most one guard segment placed to the left of the left entrance, and from at most one guard segment placed to the right of the right entrance. Stationary guards within the corridor ensure that the remaining area of the corridor is seen. This allows us to construct the corridor with the desired properties.

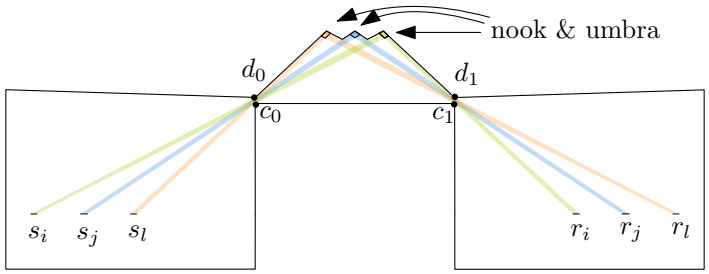


Figure 6: In this figure, we display a simplified corridor construction. The vertices c_0, c_1 serve as shadow corners for three copy-umbras simultaneously for the pairs $(s_i, r_i), (s_j, r_j), (s_l, r_l)$. Each of these pairs also have a small copy-nook in the top of corridor. The entrances c_0d_0 and c_1d_1 to the corridor are sufficiently small so that the critical segments of the nook and umbra of each pair of segments s_σ, r_σ (contained in the small boxes at the top of the figure) are not seen by other guard segments.

Addition gadget. For any equation of the form $x_i + x_j = x_l$ in Φ, where $i, j, l \in \{0, \ldots, n-1\}$, we construct a $\geq$-addition gadget which represents the inequality $x_i + x_j \geq x_l$, and a $\leq$-addition gadget for the inequality $x_i + x_j \leq x_l$.

The general idea behind the construction of the gadget imposing the $\geq$-inequality is as follows (see Figure 7). We can place three guard segments r'_i, r_j, r_l in such a way that a certain

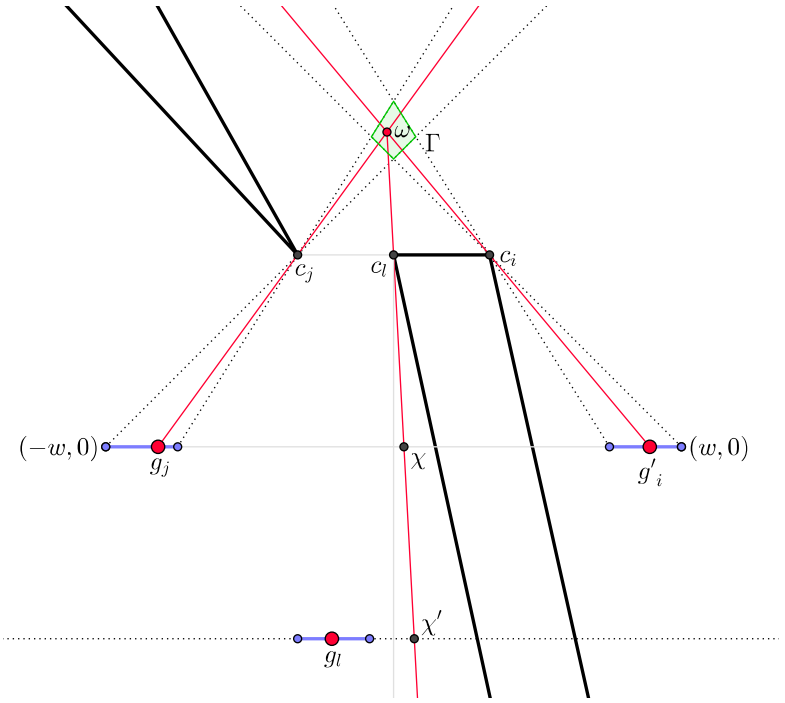


Figure 7: The thick black segments are edges of the polygon. The green quadrilateral is Γ. In order to see Γ together with g'_i and g_j, the guard g_l must be on or to the left of the point χ'.

quadrilateral Γ is seen by the guards g'_i, g_j, g_l placed at r'_i, r_j, r_l, respectively, if and only if the values x'_i, x_j, x_l specified by the corresponding segments satisfy the inequality $x'_i + x_j \geq x_l$. The area of the gadget except of Γ can be seen with the help of stationary guards. The actual gadget is more complicated, as the guard segments r'_i, r_j, r_l need to be copies (where copying is performed via corridor, as described earlier) of the base line guard segments. To make it possible, we introduce an additional guard segment r_i within the gadget, and we copy appropriately chosen segments from the base line into r_i, r_j, r_l from the left (i.e., the gadget is placed in the right side of $\mathcal{P}$). Then, by introducing a copy-nook within the construction sketched in Figure 7, we ensure that the value x'_i corresponding to r'_i is not greater than the value x_i corresponding to r_i. Such a gadget enforces the desired inequality $x_i + x_j \geq x_l$. The gadget enforcing the $\leq$-inequality is obtained in a similar way, but here we need to place the gadget in the left side of $\mathcal{P}$.

Orientation gadget. In the addition gadget, we need to copy three pairwise different line segments from the base line into the gadget. Additionally, we require the segments corresponding to the values of x_i, x_j, x_l to be copied into the gadget in a particular order. To make it possible, we create three right-oriented segments corresponding to each variable and one left-oriented.

To ensure that two base line guard segments specify the same value, we use the orientation gadget. This gadget is a slight modification of the addition gadget, where instead of a guard segment r_l, we place a stationary guard corresponding to a value of 2.5. This allows us to ensure that two line segments from the base line, one left-oriented and one right-oriented, specify the same value. Using orientation gadgets we can enforce consistency between all base line guard segments corresponding to the same variable.

Inversion gadget. In the inversion gadget, we have a right-oriented guard segment r_i and a left-oriented guard segment r_j. We construct an umbra Q_u such that guards at r_i, r_j see the whole critical segment of the umbra if and only if the values x_i, x_j specified by the segments satisfy $x_i \cdot x_j \leq 1$. We also construct a nook Q_n which enforces the inequality $x_i \cdot x_j \geq 1$.

A The art gallery problem is in $\exists\mathbb{R}$

In this section we will prove that the art gallery problem is in the complexity class $\exists\mathbb{R}$. Our proof works also for a more general version of the art gallery problem, where the input polygon can have polygonal holes.

Theorem 4. *The art gallery problem is in the complexity class* $\exists\mathbb{R}$.

Proof. Let $(\mathcal{P}, g)$ be an instance of the art gallery problem where the polygon $\mathcal{P}$ has n vertices, each of which has rational coordinates represented by at most B bits. We show how to construct a quantifier-free formula $\Phi := \Phi(\mathcal{P}, g)$ of the first-order theory of the reals such that Φ is satisfiable if and only if $\mathcal{P}$ has a guard set of cardinality g. The formula Φ has length $O(g^7 n^7 B^2) = O(n^{14} B^2)$ and can be computed in polynomial time. It has been our priority to define the formula Φ so that it is as simple as possible to describe. It might be possible to construct an equivalent but shorter formula.

The description of Φ is similar to the formula Ψ that Micha Sharir described to Efrat and Har-Peled [17]

$$\Psi \;:=\; \left[\exists x_1, y_1, \ldots, x_k, y_k \; \forall p_x, p_y : \text{INSIDE-POLYGON}(p_x, p_y) \Longrightarrow \bigvee_{i=1}^{k} \text{SEES}(x_i, y_i, p_x, p_y)\right].$$

For each $i \in \{1, \ldots, k\}$, the variables x_i, y_i represent the position of guard $g_i := (x_i, y_i)$, and $p := (p_x, p_y)$ represents an arbitrary point. The predicate INSIDE-POLYGON(p_x, p_y) tests if the point p is contained in the polygon $\mathcal{P}$, and SEES(x_i, y_i, p_x, p_y) checks if the guard g_i can see the point p. Thus, the formula is satisfiable if and only if there is a guard set of cardinality g. Note that although the implication "$\Longrightarrow$" is not allowed in the first order theory of the reals, we can always substitute "$A \Longrightarrow B$" by "$\neg A \vee B$".

For the purpose of self-containment, we will briefly repeat the construction of the predicates INSIDE-POLYGON(p_x, p_y) and SEES(x_i, y_i, p_x, p_y). The elementary tool is evaluation of the sign of the determinant $\det(\vec{u}, \vec{v})$ of two vectors $\vec{u}, \vec{v}$. Recall that the sign of the expression $\det(\vec{u}, \vec{v})$ determines whether $\vec{v}$ points to the left of $\vec{u}$ (if $\det(\vec{u}, \vec{v}) > 0$), is parallel to $\vec{u}$ (if $\det(\vec{u}, \vec{v}) = 0$), or points to the right of $\vec{u}$ (if $\det(\vec{u}, \vec{v}) < 0$).

We compute a triangulation $\mathcal{T}$ of the polygon $\mathcal{P}$, e.g., using an algorithm from [14], order the vertices of each triangle of $\mathcal{T}$ in the counter-clockwise order, and orient each edge of the triangle accordingly. A point is contained inside the polygon if and only if it is contained in one of the triangles of $\mathcal{T}$. A point is contained in a triangle if and only if it is on one of the edges or to the left of each edge. Thus the predicate INSIDE-POLYGON(p_x, p_y) has length $O(nB)$.

A guard g_i sees a point p if and only if no two consecutive edges of $\mathcal{P}$ block the visibility. See Figure 8 on why it is not sufficient to check each edge individually. Given a guard g_i, a point p, and two consecutive edges e_1, e_2 of $\mathcal{P}$, it can be checked by evaluating a constant number of determinants whether e_1, e_2 block the visibility between g_i and p. Thus SEES(x_i, y_i, p_x, p_y) has length $O(nB)$ and consequently $\bigvee_{i=1}^{k}$ SEES(x_i, y_i, p_x, p_y) has length $O(knB)$.

Note that the formula Ψ is not a formula in ETR because of the universal quantifier. The main idea to get an equivalent formula with no universal quantifier is to find a polynomial number of points inside $\mathcal{P}$ which, if all seen by the guards, ensure that all of $\mathcal{P}$ is seen. We denote such a set of points as a *witness set*.

Creating a witness set. We are now ready to describe the witness set that replaces the universal quantifier. Let $\mathcal{L} := \{\ell_1, \ldots, \ell_m\}$ be the set of lines containing either an edge of $\mathcal{P}$, or a guard $g \in G$ and a vertex $v \in P$.[†] The well-defined lines in $\mathcal{L}$ partition the plane into *regions*, which are connected components of $\mathbb{R}^2 \setminus \bigcup_{\ell \in \mathcal{L}} \ell$ (see Figure 8 for an example).

[†]Note that if a line is defined as passing through a guard $g \in G$ and a vertex $v \in P$ such that g and v are coincident, the line is not well-defined. Such lines are not considered in the partition into regions described below, but they are included in the set $\mathcal{L}$. Later we will show how to ignore these lines in our formula.

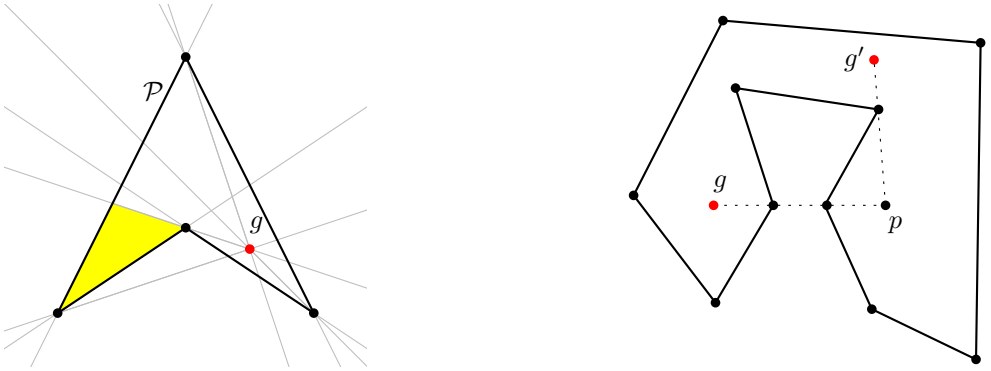


Figure 8: Illustrations for the proof that the art gallery problem is in $\exists\mathbb{R}$. Left: A polygon $\mathcal{P}$ together with a guard g and the resulting line arrangement $\mathcal{L}$. The yellow region is not seen. Right: A line segment gp going through the vertices of $\mathcal{P}$. This illustrates that it is not sufficient to check each edge individually for crossing. A vertex of $\mathcal{P}$ is on the line segment $g'p$, but the visibility between p and g' is not blocked in that case.

Let $\mathcal{A}$ be the set of these regions. It is easy to verify that $\mathcal{A}$ has the following properties:

- Each region in $\mathcal{A}$ is an open convex polygon.
- Each region in $\mathcal{A}$ is either contained in $\mathcal{P}$ or contained in the complement of $\mathcal{P}$.
- The closure of the union of the regions that are contained in $\mathcal{P}$ equals $\mathcal{P}$.
- For each region $R \in \mathcal{A}$, each guard $g \in G$ either sees all points of R or sees no point in R. In particular, if a guard g sees one point in R, it sees all of R and its closure.

Thus it is sufficient to test that for each region $R \in \mathcal{A}$ which is in $\mathcal{P}$, at least one point in R is seen by a guard. For three points a, b, c, define the *centroid* of a, b, c to be the point $C(a, b, c) := (a + b + c)/3$. If a, b, c are three different vertices of the same region $R \in \mathcal{A}$, then the centroid $C(a, b, c)$ must lie in the interior of R. Note that each region has at least three vertices and thus contains at least one such centroid. Let X be the set of all intersection points between two non-parallel well-defined lines in $\mathcal{L}$, i.e., X consists of all vertices of all regions in $\mathcal{A}$. In the formula Φ, we generate all points in X. For any three points a, b, c in X, we also generate the centroid $C(a, b, c)$. If the centroid is in $\mathcal{P}$, we check that it is seen by a guard. Since there are $O(((kn)^2)^3)$ centroids of three points in X and each is tested by a formula of size $O(knB)$, we get a formula of the aforementioned size.

Constructing the formula Φ. Each line ℓ_i is defined by a pair of points $\{(p_i, q_i), (p'_i, q'_i)\}$. Let $\vec{\ell_i} := (p'_i - p_i, q'_i - q_i)$ be a direction vector corresponding to the line. A line ℓ is well-defined if and only if the corresponding vector $\vec{\ell}$ is non-zero.

The lines ℓ_i, ℓ_j are well-defined and non-parallel if and only if $\det(\vec{\ell_i}, \vec{\ell_j}) \neq 0$. If two lines ℓ_i, ℓ_j are well-defined and non-parallel, their intersection point X^{ij} is well-defined and it has coordinates

$$\left(\frac{(p_j q'_j - q_j p'_j)(p'_i - p_i) - (p_i q'_i - q_i p'_i)(p'_j - p_j)}{\det(\vec{\ell_i}, \vec{\ell_j})}, \frac{(p_j q'_j - q_j p'_j)(q'_i - q_i) - (p_i q'_i - q_i p'_i)(q'_j - q_j)}{\det(\vec{\ell_i}, \vec{\ell_j})} \right).$$

For each pair $(i, j) \in \{1, \ldots, m\}^2$, we add the variables x_{ij}, y_{ij} to the formula Φ and we define INTERSECT(i, j) to be the formula

$$\det(\vec{\ell_i}, \vec{\ell_j}) \neq 0 \implies \Big[\det(\vec{\ell_i}, \vec{\ell_j}) \cdot x_{ij} = (p_j q'_j - q_j p'_j)(p'_i - p_i) - (p_i q'_i - q_i p'_i)(p'_j - p_j) \wedge \det(\vec{\ell_i}, \vec{\ell_j}) \cdot y_{ij} = (p_j q'_j - q_j p'_j)(q'_i - q_i) - (p_i q'_i - q_i p'_i)(q'_j - q_j) \Big].$$

It follows that if the formula INTERSECT(i, j) is true then either

- ℓ_i or ℓ_j is not well-defined or they are both well-defined, but parallel, or
- ℓ_i and ℓ_j are well-defined and non-parallel and the variables x_{ij} and y_{ij} are the coordinates of the intersection point X^{ij} of the lines.

Let $\Lambda := \{\lambda_1, \ldots, \lambda_{m^6}\} = \{1, \ldots, m\}^6$ be all the tuples of six elements from the set $\{1, \ldots, m\}$. Each tuple $\lambda := (a, b, c, d, e, f) \in \Lambda$ corresponds to a centroid of the following three points: the intersection point of the lines ℓ_a, ℓ_b, the intersection point of the lines ℓ_c, ℓ_d, and the intersection point of the lines ℓ_e, ℓ_f. For each tuple λ, we proceed as follows. We define the formula CENTROID-DEFINED(λ) to be

$$\det(\vec{\ell_a}, \vec{\ell_b}) \neq 0 \ \wedge \ \det(\vec{\ell_c}, \vec{\ell_d}) \neq 0 \ \wedge \ \det(\vec{\ell_e}, \vec{\ell_f}) \neq 0.$$

We add the variables u_λ, v_λ to the formula Φ, and define the formula CENTROID(λ) as

$$3u_\lambda = x_{ab} + x_{cd} + x_{ef} \ \wedge \ 3v_\lambda = y_{ab} + y_{cd} + y_{ef}.$$

It follows that if the formulas CENTROID-DEFINED(λ) and CENTROID(λ) are both true, then the lines in each of the pairs $(\ell_a, \ell_b), (\ell_c, \ell_d), (\ell_e, \ell_f)$ are well-defined and non-parallel, and the variables u_λ and v_λ are the coordinates of the centroid $C(X^{ab}, X^{cd}, X^{ef})$.

We are now ready to write up our existential formula as

$$\exists x_1, y_1, \ldots, x_k, y_k \ \ \exists x_{11}, y_{11}, x_{12}, y_{12}, \ldots, x_{mm}, y_{mm} \ \ \exists u_{\lambda_1}, v_{\lambda_1}, \ldots, u_{\lambda_{m^6}}, v_{\lambda_{m^6}} : \Phi,$$

where

$$\begin{aligned}
\Phi := & \left[\bigwedge_{(i,j) \in \{1,\ldots,m\}^2} \text{INTERSECT}(i, j) \right] \wedge \\
& \bigwedge_{\lambda \in \Lambda} \Bigg[\text{CENTROID-DEFINED}(\lambda) \implies \\
& \left[\text{CENTROID}(\lambda) \ \wedge \ \left[\text{INSIDE-POLYGON}(u_\lambda, v_\lambda) \implies \bigvee_{i=1}^{k} \text{SEES}(x_i, y_i, u_\lambda, v_\lambda) \right] \right] \Bigg].
\end{aligned}$$

□

B The problem ETR-INV

To show that the art gallery problem is $\exists\mathbb{R}$-hard, we will provide a reduction from the problem ETR-INV, which we introduce below. In this section, we will show that ETR-INV is $\exists\mathbb{R}$-complete.

Definition 5 (ETR-INV)**.** *In the problem ETR-INV, we are given a set of real variables* $\{x_1, \ldots, x_n\}$*, and a set of equations of the form*

$$x = 1, \quad x + y = z, \quad x \cdot y = 1,$$

for $x, y, z \in \{x_1, \ldots, x_n\}$*. The goal is to decide whether the system of equations has a solution when each variable is restricted to the range* $[1/2, 2]$*.*

Thus, if $\Phi(x)$ is an instance of ETR-INV with variables $x := (x_1, \ldots, x_n)$, the space of solutions $S_\Phi := \{x \in [1/2, 2]^n : \Phi(x)\}$ consists of the vectors in $[1/2, 2]^n$ that satisfy all the equations of Φ.

In order to show that ETR-INV is $\exists\mathbb{R}$-complete, we make use of the following problem.

Definition 10. *In the problem* $ETR^{c,+,\cdot}$*, where* $c \in \mathbb{R}$*, we are given a set of real variables* $\{x_1, \ldots, x_n\}$*, and a set of equations of the form*

$$x = c, \quad x + y = z, \quad x \cdot y = z,$$

for $x, y, z \in \{x_1, \ldots, x_n\}$*. The goal is to decide whether the system of equations has a solution.*

A modified version of the problem, where we additionally require that $x_1, \ldots, x_n \in [a, b]$ *for some* $a, b \in \mathbb{R}$*, is denoted by* $ETR^{c,+,\cdot}_{[a,b]}$*.*

We are now ready to prove that ETR-INV is $\exists\mathbb{R}$-complete.

Theorem 6. *The problem ETR-INV is* $\exists\mathbb{R}$*-complete.*

Proof. To show that ETR-INV is $\exists\mathbb{R}$-hard, we will perform a series of polynomial time reductions, starting from ETR and subsequently reducing it to the problems $\mathrm{ETR}^{1,+,\cdot}$, $\mathrm{ETR}^{1/8,+,\cdot}_{[-1/8,1/8]}$, and $\mathrm{ETR}^{1,+,\cdot}_{[1/2,2]}$, and ending with ETR-INV.

To simplify the notation, while considering a problem $\mathrm{ETR}^{c,+,\cdot}$ or $\mathrm{ETR}^{c,+,\cdot}_{[a,b]}$, we might substitute any variable in an equation by the constant c. For instance, $x + c = z$ is a shorthand for the equations $x + y = z$ and $y = c$, where y is an additional variable.

Reduction to $\mathrm{ETR}^{1,+,\cdot}$. We will first argue that $\mathrm{ETR}^{1,+,\cdot}$ is $\exists\mathbb{R}$-hard. This seems to be folklore, but we did not find a formal statement. For the sake of self-containment and rigorousness, we present here a short proof based on the following lemma.

Lemma 11 (Schaefer, Štefankovič [39])**.** *Let* $\Phi(x)$ *be a quantifier-free formula of the first order theory of the reals, where* $x := (x_1, x_2, \ldots, x_n)$ *is a vector of variables. We can construct in polynomial time a polynomial* $F : \mathbb{R}^{n+m} \to \mathbb{R}$ *of degree 4, for some* $m = O(|\Phi|)$*, so that*

$$\{x \in \mathbb{R}^n : \Phi(x)\} = \{x \in \mathbb{R}^n : (\exists y \in \mathbb{R}^m)\ F(x, y) = 0\}.$$

The coefficients of F *have bitlength* $O(|\Phi|)$*.*

Thus it is $\exists\mathbb{R}$-hard to decide if a polynomial has a real root. We reduce this problem to $\mathrm{ETR}^{1,+,\cdot}$. Consider a polynomial equation $Q = 0$. First, we generate all variables corresponding to all the coefficients of Q, by using only the constant 1, addition and multiplication. For example, a variable corresponding to 20 can be obtained as follows: $V_1 = 1, V_{2^1} = 1 + 1, V_{2^2} = V_{2^1} \cdot V_{2^1}, V_{2^4} = V_{2^2} \cdot V_{2^2}, V_{20} = V_{2^4} + V_{2^2}$. We are now left with a polynomial Q' consisting entirely of sums of products of variables, and we keep simplifying Q' as described in the following.

Whenever there is an occurrence of a sum $x + y$ or a product $x \cdot y$ of two variables in Q', we introduce a new variable z. In the first case, we add the equation $x + y = z$ to Φ and substitute the term $x + y$ by z in Q'. In the latter case, we add the equation $x \cdot y = z$ to Φ and substitute $x \cdot y$ by z in Q'. We finish the construction when Q' has been simplified to consist of a single variable, i.e., $Q' = x$, in which case we add the equation $x + V_1 = V_1$ (corresponding to the equation $Q' = 0$) to Φ. When the process finishes, Φ yields an instance of $\mathrm{ETR}^{1,+,\cdot}$, and the solutions to Φ are in one-to-one correspondence with the solutions to the original polynomial equation $Q = 0$.

Reduction to $\mathbf{ETR}^{1/8,+,\cdot}_{[-1/8,1/8]}$. We will now present a reduction from the problem $\mathrm{ETR}^{1,+,\cdot}$ to $\mathrm{ETR}^{1/8,+,\cdot}_{[-1/8,1/8]}$. We use the following result from algebraic geometry, which was stated by Schaefer and Štefankovič [39] in a simplified form. Given an instance $\Phi(x)$ of ETR over the vector of variables $x := (x_1, \ldots, x_n)$, we define the semi-algebraic set S_Φ as the solution space

$$S_\Phi := \{x \in \mathbb{R}^n : \Phi(x)\}.$$

The complexity L of a semi-algebraic set S_Φ is defined as the number of symbols appearing in the formula Φ defining S_Φ (see [28]).

Corollary 12 (Schaefer and Štefankovič [39])**.** *Let B be the set of points in $\mathbb{R}^n$ at distance at most $2^{L^{8n}} = 2^{2^{8n \log L}}$ from the origin. Every non-empty semi-algebraic set S in $\mathbb{R}^n$ of complexity at most $L \geq 4$ contains a point in B.*

Let Φ be an instance of $\mathrm{ETR}^{1,+,\cdot}$ with n variables $x_1, \ldots, x_n$. We construct an instance Φ' of $\mathrm{ETR}^{1/8,+,\cdot}_{[-1/8,1/8]}$ such that Φ has a solution if and only if Φ' has a solution. Let us fix $k := (8n \cdot \log L + 3)$ and $\varepsilon := 2^{-2^k}$. In Φ', we first define a variable V_ε satisfying $V_\varepsilon = \varepsilon$, using $\Theta(k)$ new variables $V_{1/2^{2^2}}, V_{1/2^{2^3}}, \ldots, V_{1/2^{2^k}}$ and equations

$$\begin{aligned} V_{1/2^{2^2}} + V_{1/2^{2^2}} &= 1/8, \\ V_{1/2^{2^2}} \cdot V_{1/2^{2^2}} &= V_{1/2^{2^3}}, \\ V_{1/2^{2^3}} \cdot V_{1/2^{2^3}} &= V_{1/2^{2^4}}, \\ &\vdots \\ V_{1/2^{2^{k-1}}} \cdot V_{1/2^{2^{k-1}}} &= V_\varepsilon. \end{aligned}$$

In Φ', we use the variables $V_{\varepsilon x_1}, \ldots, V_{\varepsilon x_n}$ instead of $x_1, \ldots, x_n$. An equation of Φ of the form $x = 1$ is transformed to the equation $V_{\varepsilon x} = V_\varepsilon$ in Φ'. An equation of Φ of the form $x + y = z$ is transformed to the equation $V_{\varepsilon x} + V_{\varepsilon y} = V_{\varepsilon z}$ of Φ'. An equation of Φ of the form $x \cdot y = z$ is transformed to the following equations of Φ', where $V_{\varepsilon^2 z}$ is a new variable satisfying

$$\begin{aligned} V_{\varepsilon x} \cdot V_{\varepsilon y} &= V_{\varepsilon^2 z}, \\ V_\varepsilon \cdot V_{\varepsilon z} &= V_{\varepsilon^2 z}. \end{aligned}$$

Assume that Φ is true. Then there exists an assignment of values to the variables $x_1, \ldots, x_n$ of Φ that satisfies all the equations and where each variable x_i satisfies $|x_i| \in \left[0, 2^{2^{8n \log L}}\right]$. Then the assignment $V_{\varepsilon x_i} = \varepsilon x_i$ and (when $V_{\varepsilon^2 x_i}$ appears in Φ') $V_{\varepsilon^2 x_i} = \varepsilon^2 x_i$ yields a solution to Φ' with all variables in the range $[-1/8, 1/8]$. On the other hand, if there is a solution to Φ', an analogous argument yields a corresponding solution to Φ. We have given a reduction from $\mathrm{ETR}^{1,+,\cdot}$ to $\mathrm{ETR}^{1/8,+,\cdot}_{[-1/8,1/8]}$. The length of the formula increases by at most a polylogarithmic factor.

Reduction to $\mathbf{ETR}^{1,+,\cdot}_{[1/2,2]}$. We will now show a reduction from $\mathrm{ETR}^{1/8,+,\cdot}_{[-1/8,1/8]}$ to $\mathrm{ETR}^{1,+,\cdot}_{[1/2,2]}$. The reduction is similar as in [43]. We substitute each variable $x_i \in [-1/8, 1/8]$ by $V_{x_i+7/8}$

which will be assumed to have a value of $x_i + 7/8$. Instead of an equation $x = 1/8$ we now have $V_{x+7/8} = 1$. Using addition and the variable equal 1, we can easily get the variables $V_{1/2}, V_{3/2}, V_{3/4}, V_{7/4}, V_{7/8}$ with corresponding values of $1/2, 3/2, 3/4, 7/4$, and $7/8$. Instead of each equation $x + y = z$ we now have equations:

$$\begin{aligned} V_{x+7/8} + V_{y+7/8} &= V_{(z+7/8)+7/8}, \\ V_{z+7/8} + V_{7/8} &= V_{(z+7/8)+7/8}. \end{aligned}$$

As the original variables x, y, z have values in the interval $[-1/8, 1/8]$, the added variables $V_{(z+7/8)+7/8}$ have a value in $[13/8, 15/8]$.

Instead of each equation $x \cdot y = z$ we have the following set of equations

$$\begin{aligned} V_{x+7/8} + V_{y+7/8} &= V_{x+y+14/8}, \quad (V_{x+y+14/8} \in [12/8, 2]) \\ V_{x+y+7/8} + V_{7/8} &= V_{x+y+14/8}, \quad (V_{x+y+7/8} \in [5/8, 9/8]) \\ V_{x+7/8} + V_1 &= V_{x+15/8}, \quad (V_{x+15/8} \in [14/8, 2]) \\ V_{x+1} + V_{7/8} &= V_{x+15/8}, \quad (V_{x+1} \in [7/8, 9/8]) \\ V_{y+7/8} + V_1 &= V_{y+15/8}, \quad (V_{y+15/8} \in [14/8, 2]) \\ V_{y+1} + V_{7/8} &= V_{y+15/8}, \quad (V_{y+1} \in [7/8, 9/8]) \\ V_{x+1} \cdot V_{y+1} &= V_{xy+x+y+1}, \quad (V_{xy+x+y+1} \in [49/64, 81/64]) \\ V_{xy+x+y+1} + V_{1/2} &= V_{xy+x+y+3/2}, \quad (V_{xy+x+y+3/2} \in [81/64, 113/64]) \\ V_{xy+5/8} + V_{x+y+7/8} &= V_{xy+x+y+3/2}, \quad (V_{xy+5/8} \in [39/64, 41/64]) \\ V_{xy+5/8} + 1 &= V_{xy+13/8}, \quad (V_{xy+13/8} \in [103/64, 105/64]) \\ V_{z+7/8} + V_{3/4} &= V_{xy+13/8}. \end{aligned}$$

Each formula Φ of $\mathrm{ETR}^{1/8,+,\cdot}_{[-1/8,1/8]}$ is transformed to a formula Φ' of $\mathrm{ETR}^{1,+,\cdot}_{[1/2,2]}$, as explained above. If there is a solution for Φ, there clearly is a solution for Φ', as all the newly introduced variables have a value within the intervals claimed above. If there is a solution for Φ', there is also a solution for Φ, as the newly introduced variables $V_{2x+7/4}, V_{x+5/8} \in [1/2, 2]$ ensure that $x \in [-1/8, 1/8]$. The increase in the length of the formula is linear.

Note that the only place where we use multiplication is in the formula $V_{x+1} \cdot V_{y+1} = V_{xy+x+y+1}$, where $V_{x+1}, V_{y+1} \in [7/8, 9/8]$. We will use this fact in the next step of the reduction.

Reduction to ETR-INV. We will now show that $\mathrm{ETR}^{1/8,+,\cdot}_{[-1/8,1/8]}$ reduces to ETR-INV. In the first step, we reduce a formula Φ of $\mathrm{ETR}^{1/8,+,\cdot}_{[-1/8,1/8]}$ to a formula Φ' of $\mathrm{ETR}^{1,+,\cdot}_{[1/2,2]}$, as described in the step above. We now have to show how to express each equation $x \cdot y = z$ of Φ' using only the equations allowed in ETR-INV. Note that, as explained above, multiplication is used only for variables $x, y \in [7/8, 9/8]$. Some of the steps in this reduction rely on techniques also used in the proof by Aho *et al.* [4, Section 8.2] that squaring and taking reciprocals is equivalent to multiplication.

We first show how to define a new variable V_{x^2} satisfying $V_{x^2} = x^2$, where $x \in [7/8, 9/8]$.

$$\begin{aligned} x + V_{3/4} &= V_{x+3/4}, \quad (V_{x+3/4} \in [13/8, 15/8]) \\ V_{1/(x+3/4)} \cdot V_{x+3/4} &= 1, \quad (V_{1/(x+3/4)} \in [8/15, 8/13]) \\ V_{x-1/4} + 1 &= V_{x+3/4}, \quad (V_{x-1/4} \in [5/8, 7/8]) \\ V_{1/(x-1/4)} \cdot V_{x-1/4} &= 1, \quad (V_{1/(x-1/4)} \in [8/7, 8/5]) \\ V_{1/(x^2+x/2-3/16)} + V_{1/(x+3/4)} &= V_{1/(x-1/4)}, \quad (V_{1/(x^2+x/2-3/16)} \in [64/105, 64/65]) \\ V_{1/(x^2+x/2-3/16)} \cdot V_{x^2+x/2-3/16} &= 1, \quad (V_{x^2+x/2-3/16} \in [65/64, 105/64]) \end{aligned}$$

$$
\begin{aligned}
x + V_{7/8} &= V_{x+7/8}, \quad (V_{x+7/8} \in [14/8, 2]) \\
V_{x+1/8} + V_{3/4} &= V_{x+7/8}, \quad (V_{x+1/8} \in [1, 10/8]) \\
V_{x/2+1/16} + V_{x/2+1/16} &= V_{x+1/8}, \quad (V_{x/2+1/16} \in [1/2, 10/16]) \\
V_{x^2-1/4} + V_{x/2+1/16} &= V_{x^2+x/2-3/16}, \quad (V_{x^2-1/4} \in [33/64, 65/64]) \\
V_{x^2-1/4} + V_{3/4} &= V_{x^2+1/2}, \quad (V_{x^2+1/2} \in [81/64, 113/64]) \\
V_{x^2} + V_{1/2} &= V_{x^2+1/2}.
\end{aligned}
$$

Note that the constructed variables are in the range $[\frac{1}{2}, 2]$. In the following, as shorthand for the construction given above, we allow to use equations of the form $x^2 = y$, for a variable x with a value in $[7/8, 9/8]$. We now describe how to express an equation $x \cdot y = z$, where $x, y \in [7/8, 9/8]$.

$$
\begin{aligned}
x + V_{7/8} &= V_{x+7/8}, \quad (V_{x+7/8} \in [14/8, 2]) \\
V_{(x+7/8)/2} + V_{(x+7/8)/2} &= V_{x+7/8}, \quad (V_{(x+7/8)/2} \in [14/16, 1]) \\
y + V_{7/8} &= V_{y+7/8}, \quad (V_{y+7/8} \in [14/8, 2]) \\
V_{(y+7/8)/2} + V_{(y+7/8)/2} &= V_{y+7/8}, \quad (V_{(y+7/8)/2} \in [14/16, 1]) \\
V_{(x+7/8)/2} + V_{(y+7/8)/2} &= V_{(x+y)/2+7/8}, \quad (V_{(x+y)/2+7/8} \in [14/8, 2]) \\
V_{(x+y)/2} + V_{7/8} &= V_{(x+y)/2+7/8}, \quad (V_{(x+y)/2} \in [7/8, 9/8]) \\
V^2_{(x+y)/2} &= V_{((x+y)/2)^2}, \quad (V_{((x+y)/2)^2} \in [49/64, 81/64]) \\
V_{((x+y)/2)^2} + V_{1/2} &= V_{((x+y)/2)^2+1/2}, \quad (V_{((x+y)/2)^2+1/2} \in [81/64, 113/64]) \\
x^2 &= V_{x^2}, \quad (V_{x^2} \in [49/64, 81/64]) \\
y^2 &= V_{y^2}, \quad (V_{y^2} \in [49/64, 81/64]) \\
V_{x^2} + V_{1/2} &= V_{x^2+1/2}, \quad (V_{x^2+1/2} \in [81/64, 113/64]) \\
V_{x^2/2+1/4} + V_{x^2/2+1/4} &= V_{x^2+1/2}, \quad (V_{x^2/2+1/4} \in [81/128, 113/128]) \\
V_{y^2} + V_{1/2} &= V_{y^2+1/2}, \quad (V_{y^2+1/2} \in [81/64, 113/64]) \\
V_{y^2/2+1/4} + V_{y^2/2+1/4} &= V_{y^2+1/2}, \quad (V_{y^2/2+1/4} \in [81/128, 113/128]) \\
V_{x^2/2+1/4} + V_{y^2/2+1/4} &= V_{(x^2+y^2)/2+1/2}, \quad (V_{(x^2+y^2)/2+1/2} \in [81/64, 113/64]) \\
V_{(x^2+y^2)/2} + V_{1/2} &= V_{(x^2+y^2)/2+1/2}, \quad (V_{(x^2+y^2)/2} \in [49/64, 81/64]) \\
V_{(x^2+y^2)/4+1/4} + V_{(x^2+y^2)/4+1/4} &= V_{(x^2+y^2)/2+1/2}, \quad (V_{(x^2+y^2)/4+1/4} \in [81/128, 113/128]) \\
V_{(x^2+y^2)/4+1/4} + V_{xy/2+1/4} &= V_{((x+y)/2)^2+1/2}, \quad (V_{xy/2+1/4} \in [81/128, 113/128]) \\
V_{xy/2+1/4} + V_{xy/2+1/4} &= V_{xy+1/2}, \quad (V_{xy+1/2} \in [81/64, 113/64]) \\
z + V_{1/2} &= V_{xy+1/2}.
\end{aligned}
$$

The constructed variables are in a range $[1/2, 2]$.

A formula Φ of $\mathrm{ETR}^{1/8,+,\cdot}_{[-1/8,1/8]}$ has been first transformed into a formula Φ' of $\mathrm{ETR}^{1,+,\cdot}_{[1/2,2]}$, and subsequently into a formula Φ'' of ETR-INV. If Φ is satisfiable, then both Φ' and Φ'' are satisfiable. If Φ'' is satisfiable, then both Φ' and Φ are satisfiable. We get that ETR-INV is $\exists\mathbb{R}$-hard.

As the conjunction of the equations of ETR-INV, together with the inequalities describing the allowed range of the variables within ETR-INV, is a quantifier-free formula of the first-order theory of the reals, ETR-INV is in $\exists\mathbb{R}$, which yields that ETR-INV is $\exists\mathbb{R}$-complete. □

Lemma 11 and Corollary 12 together with the reductions explained in this section imply the following lemma.

Lemma 13. *Let Φ be an instance of ETR with variables $x_1, \ldots, x_n$. Then there exists an instance Ψ of ETR-INV with variables $y_1, \ldots, y_m$, $m \geq n$, and constants $c_1, \ldots, c_n, d_1, \ldots, d_n \in \mathbb{Q}$, such that*

- *there is a solution to Φ if and only if there is a solution to Ψ, and*
- *for any solution $(y_1, \ldots, y_m)$ to Ψ, there exists a solution $(x_1, \ldots, x_n)$ to Φ where $y_1 = c_1 x_1 + d_1, \ldots, y_n = c_n x_n + d_n$.*

C Reduction from ETR-INV to the art gallery problem

C.1 Notation

Given two different points p, q, the line containing p and q is denoted as $\overleftrightarrow{pq}$, the ray with the origin at p and passing through q is denoted as $\overrightarrow{pq}$, and the line segment from p to q is denoted as pq. For a point p, we let $x(p)$ and $y(p)$ denote the x- and y-coordinate of p, respectively. Table 1 shows the definitions of some objects and distances frequently used in the description of the construction.

C.2 Overview of the construction

Let Φ be an instance of the problem ETR-INV with n variables $X := \{x_0, \ldots, x_{n-1}\}$ and consisting of k equations. We show that there exists a polygon $\mathcal{P} := \mathcal{P}(\Phi)$ with corners at rational coordinates which can be computed in polynomial time such that Φ has a solution if and only if $\mathcal{P}$ can be guarded by some number $g := g(\Phi)$ of guards. The number g will follow from the construction. A sketch of the polygon $\mathcal{P}$ is shown in Figure 9.

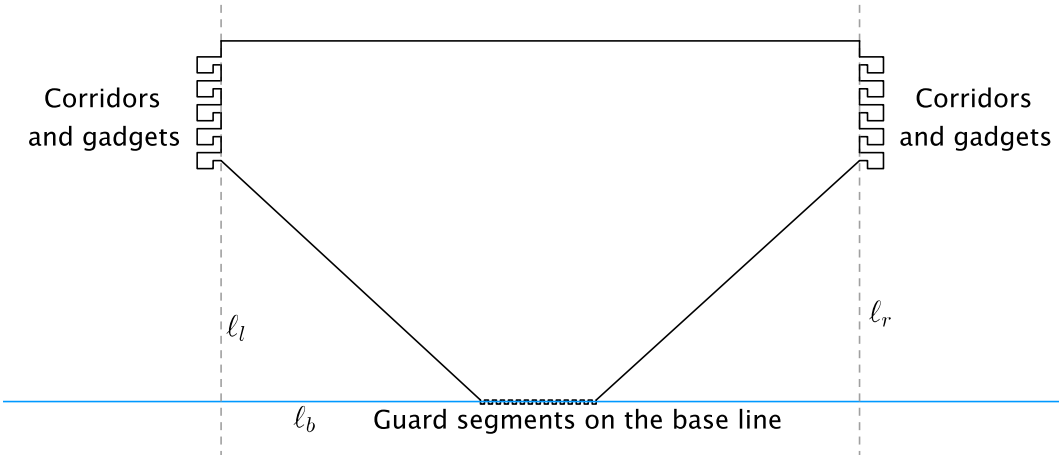


Figure 9: A high-level sketch of the construction of the polygon $\mathcal{P}$.

Each variable $x_i \in X$ is represented by a collection of *guard segments*, which are horizontal line segments contained in the interior of $\mathcal{P}$. Consider one guard segment $s := ab$, where a is to the left of b, and assume that s represents the variable x_i and that there is exactly one guard p placed on s. The guard segment s can be oriented to the right or to the left. The guard p on s specifies the value of the variable x_i as $\frac{1}{2} + \frac{3\|ap\|}{2\|ab\|}$ if s is oriented to the right, and $\frac{1}{2} + \frac{3\|bp\|}{2\|ab\|}$ if s is oriented to the left. Here, the additive term $\frac{1}{2}$ and the factor $\frac{3}{2}$ stem from the fact that all the variables in X are contained in the interval $[\frac{1}{2}, 2]$.

Suppose that there is a solution to Φ. We will show that in that case any minimum guard set G of $\mathcal{P}$ has size $g(\Phi)$ and *specifies* a solution to Φ in the sense that it satisfies the following two properties.

- Each variable $x_i \in X$ is specified *consistently* by G, i.e., there is exactly one guard on each guard segment representing x_i, and all these guards specify the same value of x_i.
- The guard set G is *feasible*, i.e., the values of X thus specified is a solution to Φ.

Moreover, if there is no solution to Φ, each guard set of $\mathcal{P}$ consists of more than $g(\Phi)$ guards.

The polygon $\mathcal{P}$ is constructed in the following way. The bottom part of the polygon consists of a collection of *pockets*, containing in total $4n$ collinear and equidistant guard segments. We

Name	Description/value
Φ	ETR-INV formula that we reduce from
$X := \{x_0, \ldots, x_{n-1}\}$	set of variables of Φ
$\mathcal{P} := \mathcal{P}(\Phi)$	final polygon to be constructed from Φ
$g := g(\Phi)$	number of guards needed to guard $\mathcal{P}$ if and only if Φ has a solution
k	number of equations in Φ
n	number of variables in Φ
k'	$3n$ plus number of equations $x \cdot y = 1$, $x + y = z$ in Φ
N	$\max\{4n, k'\}$
C	200000
ℓ_b (base line)	line that contains $4n$ guard segments at the bottom of $\mathcal{P}$
$s_i = a_i b_i$	guard segment on ℓ_b, $i \in \{0, \ldots, 4n-1\}, \|a_i b_i\| = 3/2$
$\mathcal{P}_M$ (main part of $\mathcal{P}$)	middle part of $\mathcal{P}$, without corridors and gadgets
ℓ_r, ℓ_l	vertical lines bounding $\mathcal{P}_M$
corridor	connection between the main part of $\mathcal{P}$ and a gadget
c_0d_0, c_1d_1	corridor entrances, $d_0 := c_0 + (0, \frac{3}{CN^2})$ and $d_1 := c_1 + (0, \frac{1.5}{CN^2})$
r_i, r_j, r_l, r'_i	guard segments within gadgets, of length $\frac{1.5}{CN^2}$
a'_σ, b'_σ, $\sigma \in \{i, j, l\}$	left and right endpoint of r_σ
ℓ_c	vertical line through $\frac{c_1+c_2}{2}$
o,o'	intersections of rays $\overrightarrow{a_0c_0}$ and $\overrightarrow{b'_lc_1}$ with the line ℓ_c
δ, ρ , ε	$\delta := \frac{13.5}{CN^2}$, $\rho := \frac{\delta}{9} = \frac{1.5}{CN^2}$, $\varepsilon := \frac{\rho}{12} = \frac{1}{8CN^2}$
V	$\frac{c_0+c_1}{2} + (0,1) + [-38N\rho, +38N\rho] \times [-38N\rho, +38N\rho]$
slab $S(q, v, r)$	region of all points with distance at most r to the line through q with direction v
center of slab	line in the middle of a slab
L-slabs, R-slabs	uncertainty regions for visibility rays, see page 31

Table 1: Parameters, variables, and certain distances that are frequently used are summarized in this table for easy access. Some descriptions are much simplified.

denote the horizontal line containing these guard segments as the *base line* or ℓ_b. In order from left to right, we denote the guard segments as $s_0, \ldots, s_{4n-1}$. The segments $s_0, \ldots, s_{n-1}$ are right-oriented segments representing the variables $x_0, \ldots, x_{n-1}$, as are the segments $s_n, \ldots, s_{2n-1}$, and $s_{2n}, \ldots, s_{3n-1}$. The segments $s_{3n}, \ldots, s_{4n-1}$ are left-oriented ant they also represent the variables $x_0, \ldots, x_{n-1}$. At the left and at the right side of $\mathcal{P}$, there are some *corridors* attached, each of which leads into a *gadget*. The *entrances* to the corridors at the right side of $\mathcal{P}$ are line segments contained in a vertical line ℓ_r. Likewise, the entrances to the corridors at the left side of $\mathcal{P}$ are contained in a vertical line ℓ_l. The gadgets also contain guard segments, and they are used to impose dependencies between the guards in order to ensure that if there is a solution to Φ, then any minimum guard set of $\mathcal{P}$ consists of $g(\Phi)$ guards and specifies a solution to Φ in the sense defined above. The corridors are used to copy the positions of guards on guard segments on the base line to guards on guard segments inside the gadgets. Each gadget corresponds to a constraint of one of the types $x + y \geq z$, $x + y \leq z$, $x \cdot y = 1$, $x + y \geq 5/2$, and $x + y \leq 5/2$. The first three types of constraints are used to encode the dependencies between the variables in X as specified by Φ, whereas the latter two constraints are used to encode the dependencies between the right-oriented and left-oriented guard segments representing a single variable in X.

C.3 Creating a stationary guard position

We denote some points of $\mathcal{P}$ as *stationary guard positions*. A guard placed at a stationary guard position is called a *stationary guard*. We will often define a stationary guard position as the unique point $p \in \mathcal{P}$ such that a guard placed at p can see some two vertices q_1, q_2 of the polygon $\mathcal{P}$.

We will later prove that for any guard set of size of at most $g(\Phi)$, there is a guard placed at each stationary guard position. For that, we will need the lemma stated below. For an example of the application of the lemma, see Figure 12. The stationary guard position g_2 is the only point from which a guard can see both vertices q_1 and q_2. Applying Lemma 14 with $p := g_2$, $W := \{q_1, q_2\}$, $A := P$ and $M := \{t_1, t_2\}$, we get that there must be a guard placed at g_2 in any guard set of size 3. The purpose of the area A is so that we can restrict our arguments to a small area of the polygon.

Lemma 14. *Let P be a polygon, $A \subseteq P$, and M a set of points in A such that no point in M can be seen from a point in $P \setminus A$, and no two points in M can be seen from the same point in P. Suppose that there is a point $p \in A$ and a set of points $W \subset A$ such that*

1. *no point in W can be seen from a point in $P \setminus A$,*
2. *the only point in P that sees all points in W is p, and*
3. *no point in P can see a point in M and a point in W simultaneously.*

Then any guard set of P has at least $|M| + 1$ guards placed within A, and if a guard set with $|M| + 1$ guards placed within A exists, one of its guards is placed at p.

Proof. Let q be a point in W. Since no two points in the set $\{q\} \cup M$ can be seen from the same point in P, and no point from $P \setminus A$ can see a point in $\{q\} \cup M$, at least $|M| + 1$ guards are needed within A. Suppose that a guard set with exactly $|M| + 1$ guards placed within A exists. There must be $|M|$ guards in A such that each of them can see one point in M and no point in W. The last guard in A has to be at the point p in order to see all points in W. □

In the polygon $\mathcal{P}$ we often use stationary guards for the purpose of seeing some region on one side of a line segment ℓ, but no points on the other side of ℓ. Other guards have the responsibility to see the remaining area. See Figure 10 (left) for an explanation of how a stationary guard position can be constructed.

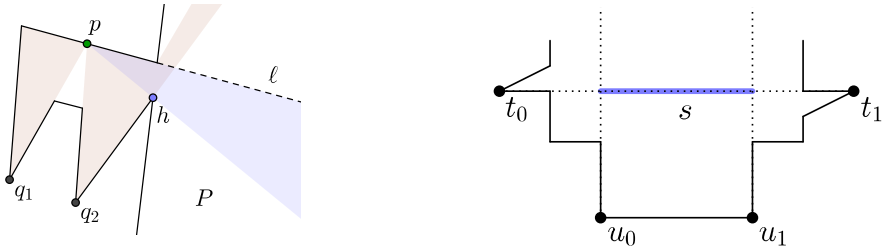


Figure 10: Left: The construction of a stationary guard position p that sees an area in P below a line segment ℓ. The brown areas are the regions of points that see q_1 and q_2, and p is the only point that sees both q_1 and q_2. The point p sees the points in the blue wedge, and the angle of the wedge can be adjusted by choosing the point h accordingly. Right: The construction of a guard segment s (the blue segment). In order to see the points t_0, t_1, a guard must be on the horizontal dotted segment. Furthermore, in order to see u_0, u_1, the guard must be between the vertical dotted segments that contain the endpoints of s. Thus, a guard sees t_0, t_1, u_0, u_1 if and only if the guard is at s.

C.4 Creating a guard segment

In the construction of $\mathcal{P}$ we will denote some horizontal line segments of $\mathcal{P}$ as *guard segments*. We will later prove that for any guard set of size of at most $g(\Phi)$, there is exactly one guard placed at each guard segment.

We will always define a guard segment s by providing a collection of four vertices of $\mathcal{P}$ such that a guard within $\mathcal{P}$ can see all these four vertices if and only if it is placed on the line segment s. See Figure 10 (right) for an example of such a construction. To show that there is a guard placed on a guard segment, we will use the following lemma.

Lemma 15. *Let P be a polygon, $A \subseteq P$, and M a set of points in A such that no point in M can be seen from a point in $P \setminus A$, and no two points in M can be seen from the same point in P. Suppose that there is a line segment s in A, and points $t_0, t_1, u_0, u_1 \in A$ such that*

1. *no point in $\{t_0, t_1, u_0, u_1\}$ can be seen from a point in $P \setminus A$,*
2. *a guard in P sees all of the points t_0, t_1, u_0, u_1 if and only if the guard is at s, and*
3. *no point in P can see a point in M and one of the points t_0, t_1, u_0, u_1.*

Then any guard set of P has at least $|M| + 1$ guards placed within A, and if a guard set with $|M| + 1$ guards placed within A exists, one of its guards is placed on the line segment s.

Proof. Similar to the proof of Lemma 14. □

Consider once more the example pictured in Figure 12, where we want to guard the polygon with only three guards. We define two guard segments, a_0b_0 and a_1b_1. The first one is defined by the vertices t_0, t_1, u_0, u_1, and the second one by the vertices t_2, t_3, u_2, u_3. Applying Lemma 15 we get that there must be a guard placed at a_0b_0 (we set $A := P$ and $M := \{t_2, q_2\}$) and at a_1b_1 (we set $A := P$ and $M := \{t_1, q_2\}$) in any guard set of size 3.

As already explained in Section C.2, guards placed at the guard segments will be used to encode the values of the variables of Φ.

C.5 Imposing inequalities by nooks and umbras

In this section we introduce nooks and umbras, which are our basic tools used to impose dependency between guards placed on two different guard segments. For the following definitions, see Figure 11.

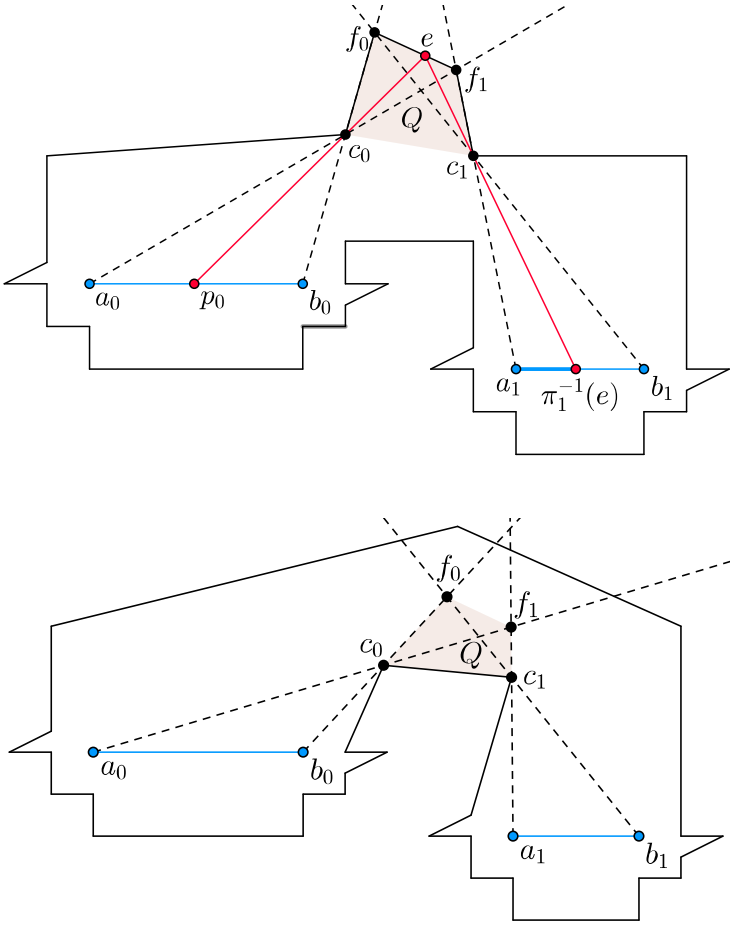


Figure 11: The brown area Q representing a nook (top), and an umbra (bottom). In the left figure, note that if a guard p_1 placed at the segment a_1b_1 has to see the whole line segment f_0f_1 together with p_0, then p_1 must be on or to the left of the point $\pi_1^{-1}(e)$, where $e := \pi_0(p_0)$.

Definition 7 (nook and umbra)**.** *Let $\mathcal{P}$ be a polygon with guard segments $r_0 := a_0b_0$ and $r_1 := a_1b_1$, where r_0 is to the left of r_1. Let c_0, c_1 be two vertices of $\mathcal{P}$, such that c_0 is to the left of c_1. Suppose that the rays $\overrightarrow{b_0c_0}$ and $\overrightarrow{b_1c_1}$ intersect at a point f_0, the lines $\overrightarrow{a_0c_0}$ and $\overrightarrow{a_1c_1}$ intersect at a point f_1, and that $Q := c_0c_1f_1f_0$ is a convex quadrilateral contained in $\mathcal{P}$. For each $i \in \{0,1\}$ define the function $\pi_i \colon r_i \longrightarrow f_0f_1$ such that $\pi_i(p)$ is the intersection of the ray $\overrightarrow{pc_i}$ with the line segment f_0f_1, and suppose that π_i is bijective.*

We say that Q is a nook *of the pair of guard segments r_0, r_1 if for each $i \in \{0,1\}$ and every $p \in r_i$, a guard at p can see all of the segment $\pi_i(p)f_{1-i}$ but nothing else of f_0f_1. We say that Q is an* umbra *of the segments r_0, r_1 if for each $i \in \{0,1\}$ and every $p \in r_i$, a guard at p can see all of the segment $\pi_i(p)f_i$ but nothing else of f_0f_1. The functions π_0, π_1 are called* projections *of the nook or the umbra.*

We will construct nooks and umbras for pairs of guard segments where we want to enforce dependency between the values of the corresponding variables. When making use of an umbra, we will also create a stationary guard position from which a guard sees the whole quadrilateral Q, but nothing on the other side of the line segment f_0f_1. In this way we can enforce the guards on r_0 and r_1 to see all of f_0f_1 together. For the case of a nook, the segment f_0f_1 will always be on the polygon boundary, and then there will be no stationary guard needed. See Figure 12 for an example of a construction of both a nook and an umbra for a pair of guard segments.

Definition 8 (critical segment and shadow corners)**.** *Consider a nook or an umbra $Q :=$*

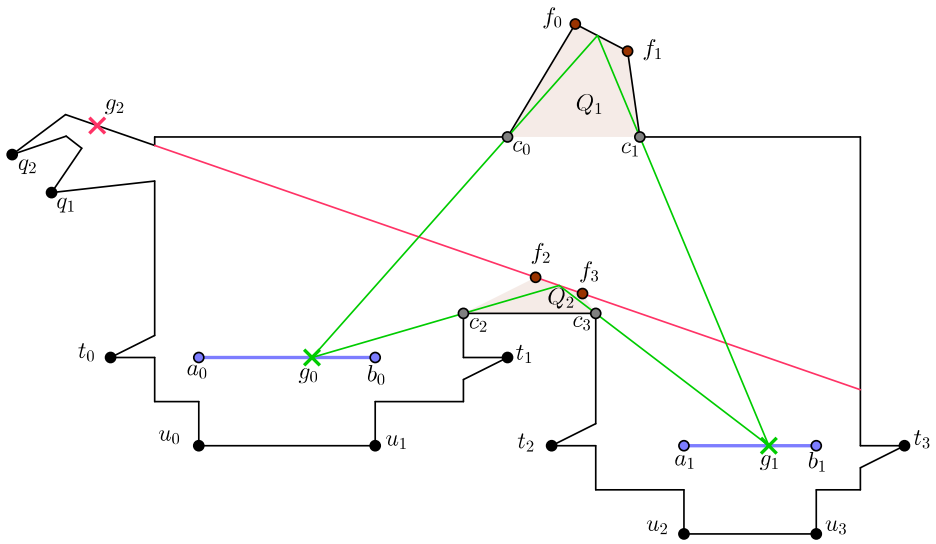


Figure 12: Q_1 is a copy-nook of the segments $r_0 := a_0b_0$ and $r_1 := a_1b_1$ with a critical segment f_0f_1, and Q_2 is a copy-umbra for the same pair with a critical segment f_2f_3. Lemmas 14 and 15 imply that this polygon cannot be guarded by fewer than 3 guards, and any guard set with 3 guards must contain a guard g_0 on r_0, a guard g_1 on r_1, and a stationary guard at the point g_2. The guards g_0 and g_1 must specify the same value on r_0 and r_1, respectively.

$c_0c_1f_1f_0$ of a pair of guard segments r_0, r_1. The line segment f_0f_1 is called the critical segment *of Q, and the vertices c_0, c_1 are called the* shadow corners *of Q.*

Consider a nook or an umbra of a pair of guard segments r_0, r_1. Let p_0, p_1 be the guards placed on the guard segments r_0 and r_1, respectively, and assume that p_0 and p_1 together see all of the critical segment f_0f_1. Let $e := \pi_0(p_0)$. The condition that p_0, p_1 together see all of f_0f_1 enforces dependency between the position of the guard p_1 and the point $\pi_1^{-1}(e)$. If Q is a nook, p_1 must be in the closed wedge W between the rays $\overrightarrow{ec_0}$ and $\overrightarrow{ec_1}$. If Q is an umbra, p_1 must be in $\overline{\mathcal{P} \setminus W}$ (the closure of the complement of W), i.e., either on or to the right of $\pi_1^{-1}(e)$. This observation will allow us to impose an inequality on the x-coordinates of p_0, p_1, and thus on the variables corresponding to the guard segments r_0, r_1.

C.6 Copying one variable

Definition 9. *Let Q be a nook or an umbra of a pair of guard segments $r_0 := a_0b_0$ and $r_1 := a_1b_1$ with the same orientation, such that the shadow corners c_0 and c_1 have the same y-coordinate. We then call Q a* copy-nook *or a* copy-umbra*, respectively.*

We can show the following result.

Lemma 16. *Let Q be a copy-nook or a copy-umbra for a pair of guard segments $r_0 := a_0b_0$ and $r_1 := a_1b_1$. Then for every point $e \in f_0f_1$ we have $\frac{\|a_0\pi_0^{-1}(e)\|}{\|a_0b_0\|} = \frac{\|a_1\pi_1^{-1}(e)\|}{\|a_1b_1\|}$, i.e., the points $\pi_0^{-1}(e)$ and $\pi_1^{-1}(e)$ on the corresponding guard segments r_0 and r_1 represent the same value.*

Proof. See Figure 13. Let $\ell := \overleftrightarrow{c_0c_1}$ be the horizontal line containing the line segment c_0c_1, and ℓ' a horizontal line passing trough e. Let f_2 be an intersection point of the line $\overleftrightarrow{f_0f_1}$ with the line ℓ. Let a_0', a_1', b_0', b_1' be the intersection points of the rays $\overrightarrow{a_0c_0}, \overrightarrow{a_1c_1}, \overrightarrow{b_0c_0}, \overrightarrow{b_1c_1}$, respectively, with the line ℓ'.

We obtain

$$\frac{\|a_0\pi_0^{-1}(e)\|}{\|\pi_0^{-1}(e)b_0\|} \cdot \frac{\|\pi_1^{-1}(e)b_1\|}{\|a_1\pi_1^{-1}(e)\|} = \frac{\|ea_0'\|}{\|b_0'e\|} \cdot \frac{\|b_1'e\|}{\|ea_1'\|} = \frac{\|ea_0'\|}{\|ea_1'\|} \cdot \frac{\|b_1'e\|}{\|b_0'e\|} = \frac{\|c_0f_2\|}{\|c_1f_2\|} \cdot \frac{\|c_1f_2\|}{\|c_0f_2\|} = 1.$$

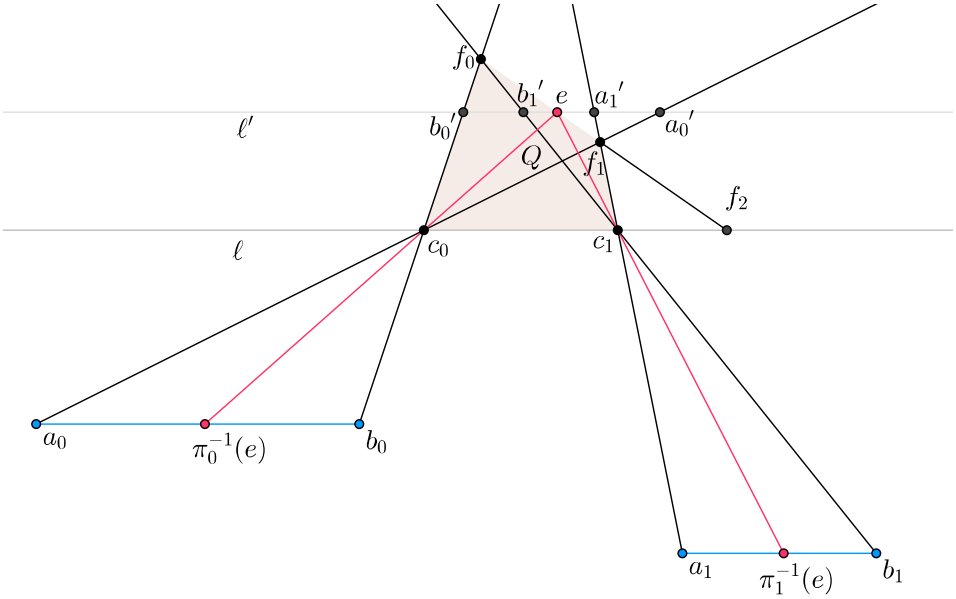


Figure 13: A copy-nook or a copy-umbra Q for a pair of guard segments $r_0 := a_0b_0$ and $r_1 := a_1b_1$. The points $\pi_0^{-1}(e)$ and $\pi_1^{-1}(e)$ represent the same value.

The first equality holds as the following pairs of triangles are similar: $a_0\pi_0^{-1}(e)c_0$ and $a_0'ec_0$, $\pi_0^{-1}(e)b_0c_0$ and $eb_0'c_0$, $a_1\pi_1^{-1}(e)c_1$ and $a_1'ec_1$, $\pi_1^{-1}(e)b_1c_1$ and $eb_1'c_1$. The third equality holds as the following pairs of triangles are similar: $ea_0'f_1$ and $f_2c_0f_1$, $ea_1'f_1$ and $f_2c_1f_1$, $b_1'ef_0$ and $c_1f_2f_0$, and $b_0'ef_0$ and $c_0f_2f_0$. □

The following lemma is a direct consequence of Lemma 16. See Figure 12 for an example of how a construction as described in the lemma can be made.

Lemma 17. *Let r_0, r_1 be a pair of guard segments oriented in the same way for which there is both a copy-nook and a copy-umbra. Suppose that there is exactly one guard p_0 placed at r_0 and one guard p_1 placed at r_1, and that the guards p_0 and p_1 see together both critical segments. Then the guards p_0 and p_1 specify the same value.*

Definition 18. *Let r_0, r_1 be a pair of guard segments for which there is both a copy-nook and a copy-umbra. We say that r_1 is a* copy *of r_0. If there is only a copy-nook* or *a copy-umbra of the pair r_0, r_1, we say that r_1 is a* weak copy *of r_0.*

It will follow from the construction of the polygon $\mathcal{P}$ that if there is a solution to Φ, then for any optimal guard set of $\mathcal{P}$ and any pair of guard segments r_0, r_1 such that r_1 is a copy of r_0, there is exactly one guard on each segment r_0, r_1, and the guards together see the whole critical segment of both the copy-nook and the copy-umbra.

C.7 The overall design of the polygon $\mathcal{P}$

Recall the high-level sketch of the polygon $\mathcal{P}$ in Figure 9. The bottom part of the polygon consists of *pockets* containing $4n$ guard segments $s_0, \ldots, s_{4n-1}$. The guard segments are placed on the base line ℓ_b, each segment having a width of $3/2$ and contained within a pocket of width 13.5. Therefore the horizontal space used for the $4n$ guard segments on the base line ℓ_b is $54n$. The wall of $\mathcal{P}$ forming the $4n$ pockets is denoted the *bottom wall*. The detailed description of the pockets and of the bottom wall is presented in Section C.8.

Let k' be equal to $3n$ plus the number of equations in Φ of the form $x_i + x_j = x_k$ or $x_i \cdot x_j = 1$. At the right side of $\mathcal{P}$ and at the left side of $\mathcal{P}$ there will be at most k' *corridors* attached, each of which leads into a *gadget*. The *entrances* to the corridors are contained in the vertical lines

ℓ_r and ℓ_l. The corridors are described in Section C.9, and they are placed equidistantly, with a vertical distance of 3 between the entrances of two consecutive corridors along the lines ℓ_r and ℓ_l. The gadgets are described in Sections C.10–C.13. The total vertical space occupied by the corridors and the gadgets at each side of $\mathcal{P}$ is at most $3k'$.

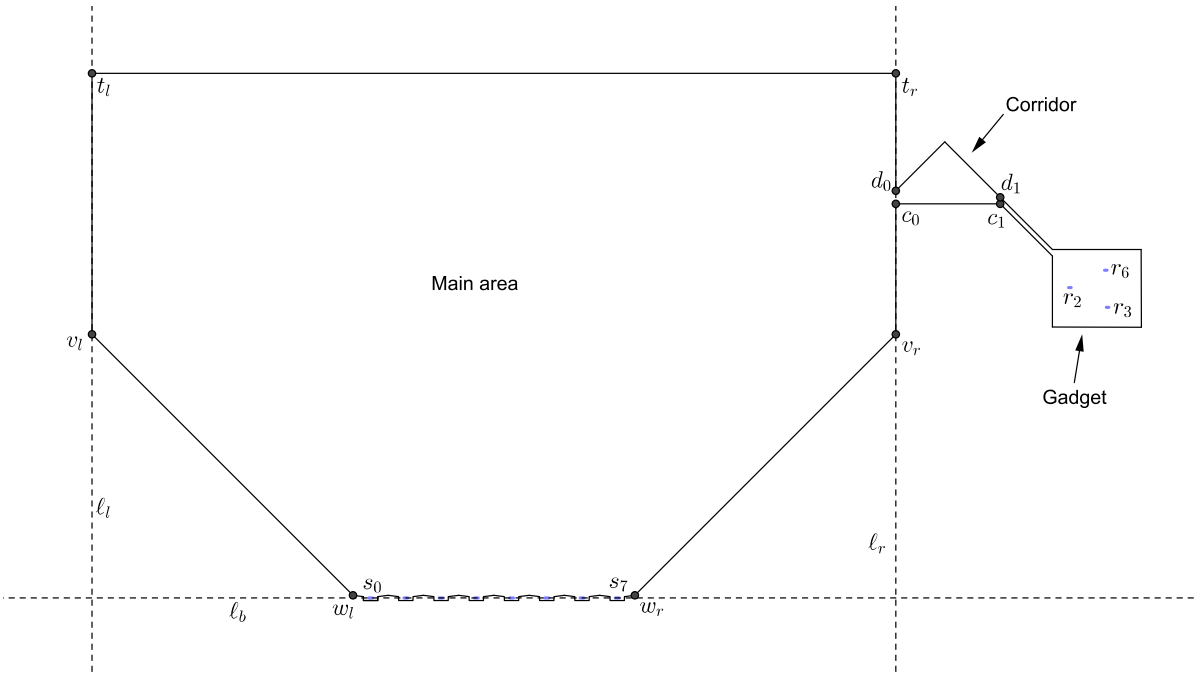


Figure 14: A sketch of the construction of $\mathcal{P}$ with 8 guard segments $s_0, \ldots, s_7$ on the base line, and one gadget at the right side of $\mathcal{P}$. The gadget contains the guard segments r_2, r_3, r_6, which are copies of the segments s_2, s_3, s_6, respectively. The gadget is the part of the polygon to the right of the line segment c_1d_1. The proportions in the drawing are not correct.

Consider the sketch of the polygon $\mathcal{P}$ in Figure 14. Let $N := \max\{4n, k'\}$, and let us define a constant $C := 200000$. Let w_l and w_r denote the left and right endpoint of the bottom wall, respectively. The horizontal distance from w_l to the line ℓ_l is $CN^2 - 54n + 6$, as is the horizontal distance from w_r to ℓ_r. The horizontal distance from the left endpoint a_0 of the leftmost segment to ℓ_r, as well as the horizontal distance from the right endpoint b_{4n-1} of the rightmost segment to ℓ_l, is CN^2. The vertical distance from ℓ_b up to the entrance of the first corridor is CN^2. The boundary of $\mathcal{P}$ contains an edge connecting w_l to the point $v_l := w_l + (-(CN^2 - 54n + 6), CN^2 - 1)$ on ℓ_l, and an edge connecting w_r to the point $v_r := w_r + (CN^2 - 54n + 6, CN^2 - 1)$ on ℓ_r. Let $t_l := v_l + (0, 3k')$ and $t_r := v_r + (0, 3k')$. The *main area* $\mathcal{P}_M$ of $\mathcal{P}$ is the area bounded by the bottom wall of $\mathcal{P}$ (to be defined in Section C.8) and a polygonal curve defined by the points $w_l v_l t_l t_r v_r w_r$. The entrances to the corridors are on the segment $v_l t_l$ in the left side and on the segment $v_r t_r$ in the right side. The set $\mathcal{P} \setminus \mathcal{P}_M$ outside of the main area consists of corridors and gadgets.

The reason why we need the distances from the guard segments on the base line ℓ_b to the gadgets to be so large is that we want all the rays from the guard segments on the base line through the corridor entrances on ℓ_r (ℓ_l) to have nearly the same slopes. That will allow us to describe a general method for copying guard segments from the base line into the gadgets.

Each gadget corresponds to a constraint involving either two or three variables, where each variable corresponds to a guard segment on the base line. Gadgets are connected with the main area $\mathcal{P}_M$ via corridors. A corridor does not contain any guard segments, and its aim is enforcing consistency between (two or three) pairs of guard segments, where one segment from each pair is in $\mathcal{P}_M$ and the other one is in the gadget. Each corridor has two vertical *entrances*, the entrance c_0d_0 of height $\frac{3}{CN^2}$ connecting it with $\mathcal{P}_M$, and the entrance c_1d_1 of height $\frac{1.5}{CN^2}$

connecting it with a gadget. The bottom wall of a corridor is a horizontal line segment c_0c_1 of length 2. The shape of the upper wall is more complicated, and it depends on the indices of the guard segments involved in the corresponding constraint, and on the height of where the corridor is placed with respect to the base line ℓ_b.

A gadget can be thought of as a room which is connected with the main area $\mathcal{P}_M$ of $\mathcal{P}$ via a corridor, i.e., attached to the vertices c_1 and d_1 of the corridor. There are five different kinds of gadgets, each corresponding to a different kind of inequality or equation, and, unlike for the case of corridors, all gadgets of the same type are identical. Each gadget contains one or two guard segments for each variable present in the corresponding formula. All guard segments within a gadget are of length $\frac{1.5}{CN^2}$, and are placed very close to the *middle point* of the gadget, defined as $m := c_1 + (1, -1)$ for gadgets at the right side of $\mathcal{P}$, and $m := c_1 + (-1, -1)$ for gadgets at the left side of $\mathcal{P}$.

C.8 Construction of the bottom wall

In this section we present the construction of the bottom wall of $\mathcal{P}$. We first describe the overall construction, as shown in Figure 15, and later we introduce small features corresponding to each equation of the type $x_i = 1$ in Φ.

The bottom wall forms $4n$ *pockets*, each pocket containing one guard segment on the base line ℓ_b. Each pocket has a width of 13.5. Each guard segment has a length of 3/2, and the distance between two consecutive segments is 12.

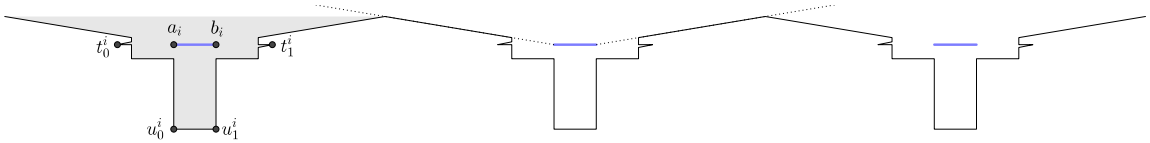


Figure 15: The construction of three consecutive guard segments (blue) on the base line. A pocket corresponding to a single guard segment $s_i := a_ib_i$ is marked in grey.

Let $s_0 := a_0b_0, \ldots, s_{4n-1} := a_{4n-1}b_{4n-1}$ be the guard segments in order from left to right. A pocket for a guard segment $s_i = a_ib_i$ is constructed as shown in Figure 15 (the grey area in the figure). The left endpoint of the pocket is at the point $a_i + (-6, 1)$, and the right endpoint of the pocket is at the point $a_i + (7.5, 1)$. The guard segment is defined by the following points, which are vertices of the pocket: $t_0^i := a_i + (-2, 0)$, $t_1^i := a_i + (3.5, 0)$, $u_0^i := a_i + (0, -5)$, $u_1^i := a_i + (1.5, -5)$. The vertical edges of the pocket are contained in lines $x = x(a_i) - 1.5$, $x = x(a_i)$, $x = x(a_i) + 1.5$, and $x = x(a_i) + 3$. The horizontal edges of the pocket are contained in lines $y = 0, y = -0.5$, and $y = -5$. The remaining edges are constructed so that the points t_0^i, t_1^i can be seen only from within the pocket, and that any point on s_i sees the whole area of the pocket.

Consider an equation of the form $x_i = 1$ in Φ. There are four guard segments representing x_i, i.e., the guard segments s_i, s_{i+n}, s_{i+2n}, and s_{i+3n}, where the first three are right-oriented and the last one is left-oriented. We add two spikes in the construction of the leftmost of these guard segments, i.e., the segment s_i, as shown in Figure 16. The dashed lines in the figure intersect at the point $g_i \in s_i$, where $g_i := a_i + (1/2, 0)$. The spike containing q_1^i enforces the guard to be at the point g_i or to the right of it, while the spike containing q_2^i enforces the guard to be at g_i or to the left of it. Also, the points q_1^i and q_2^i are chosen so that they cannot be seen by any points from within the corridors or gadgets. The guard segment is thus reduced to a stationary guard position g_i corresponding to the value $x_i = 1$.

Note that we only need to add such spikes to the pocket containing s_i, since the construction described in Section C.12 will enforce the guards on all the segments s_i, s_{i+n}, s_{i+2n}, and s_{i+3n} to specify the value of x_i consistently. We have now specified all the details of the main area $\mathcal{P}_M$ of $\mathcal{P}$ (recall the definition of $\mathcal{P}_M$ from Section C.7). The following lemma holds.

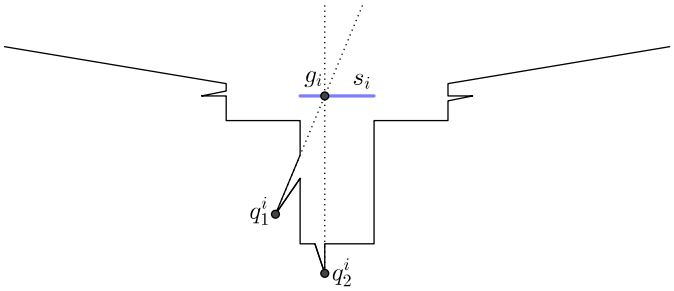


Figure 16: The spikes with vertices at q_1^i and q_2^i enforce the guard from the guard segment s_i to be at the point g_i corresponding to the value of 1.

Lemma 19. *For any ETR-INV formula Φ, we can construct in polynomial time the bottom wall corresponding to Φ such that every vertex has rational coordinates, with the enumerator upper-bounded by ζN and denominator upper-bounded by ζ, where ζ is some universal constant.*

We can prove the following lemma.

Lemma 20. *Any guard set G of the polygon $\mathcal{P}$ satisfies the following properties.*

- *G has at least $4n$ guards placed in $\mathcal{P}_M$.*
- *If G has exactly $4n$ guards placed in $\mathcal{P}_M$, then it has one guard within each of the $4n$ guard segments $s_0, \ldots, s_{4n-1}$.*
- *If G has exactly $4n$ guards placed in the main area of $\mathcal{P}$, then for each variable $x_i \in X$ such that there is an equation $x_i = 1$ in Φ, the guard at the segment s_i is at the position corresponding to the value 1.*

Moreover, any set G' of points such that (i) G' has a point within each of the $4n$ guard segments of $\mathcal{P}_M$, and (ii) for each variable $x_i \in X$ such that there is an equation $x_i = 1$ in Φ there is a guard at s_i at the position corresponding to the value of 1, can see together the whole area of $\mathcal{P}_M$.

Proof. Recall that each point t_1^i, for $i \in \{0, \ldots, 4n-1\}$, can be seen only from within the pocket corresponding to the guard segment s_i. Therefore any guard set requires at least one guard within each of these pockets, i.e., it contains at least $4n$ guards in $\mathcal{P}_M$.

Now, consider any guard segment s_i, for $i \in \{0, \ldots, 4n-1\}$. Let $M := \{t_1^j : j \in \{0, 1, \ldots, 4n-1\}, j \neq i\}$. Note also that none of the points $t_0^i, t_1^i, u_0^i, u_1^i$, and also no point from M can be seen by a guard placed within a corridor or a gadget of $\mathcal{P}$, i.e., outside of the main area of $\mathcal{P}$. By Lemma 15, by taking $t_0^i, t_1^i, u_0^i, u_1^i$ as t_0, t_1, u_0, u_1 and $A := \mathcal{P}_M$, we obtain that if a guard set G has exactly $4n$ guards placed in $\mathcal{P}_M$, then it has a guard within s_i.

For the third property, consider any variable $x_i \in X$ such that there is an equation $x_i = 1$ in Φ. As none of the points q_1^i and q_2^i can be seen from guards within the corridors or gadgets, or from the guards within the other guard segments on the base line, both of them must be seen by the only guard g_i placed on s_i. Then, g_i must be placed at the position corresponding to the value 1.

At last, consider any set G' of points such that (i) G' has a point within each of the $4n$ guard segments of $\mathcal{P}_M$, and (ii) for each variable $x_i \in X$ such that there is an equation $x_i = 1$ in Φ there is a guard at s_i at the position corresponding to the value of 1. We will show that G' can see together the whole area of $\mathcal{P}_M$. From the construction of the pockets, a guard within a pocket containing s_i can see the whole area of the pocket (in particular, the guards at positions corresponding to the value of 1 can see the whole area of the added spikes). The guard at s_i also

sees the whole area of $\mathcal{P}_M$ which is above the pocket, i.e., all points of $\mathcal{P}_M$ with x-coordinates in $[x(a_i) - 6, x(a_i) + 7.5]$. The part of $\mathcal{P}_M$ to the left of the leftmost pocket can be seen by the guard on the leftmost guard segment, and the part of $\mathcal{P}_M$ to the right of the rightmost pocket can be seen by the guard on the rightmost guard segment. □

C.9 Construction of a corridor

In this section, we describe the construction of a corridor. Inside each gadget there are two or three guard segments r_i, r_j, r_l (or r_i, r_j) corresponding to two or three pairwise different guard segments from the base line s_i, s_j, s_l (or s_i, s_j). We require that for each $\sigma \in \{i, j, l\}$ the guard segments s_σ, r_σ have the same orientation. For the corridors attached at the right side of $\mathcal{P}$ we assume $i < j < l$, and for the corridors attached at the left side we assume $i > j > l$. We describe here how to construct a corridor that ensures that that the segments r_i, r_j, r_l are copies of the segments s_i, s_j, s_l, respectively. This construction requires that the guard segments within the gadget satisfy the conditions of some technical lemmas (see Lemma 21 and 26).

Note that this construction can be generalized for copying an arbitrary subset of guard segments, but since we only need to copy two or three segments, we explain the construction in the setting of three segments. The construction for two segments is analogous but simpler. We first describe how to copy into a gadget at the right side of the polygon $\mathcal{P}$ – copying into gadgets at the left side of $\mathcal{P}$ can be done in a symmetric way and is described shortly in Section C.9.4.

As described briefly in Section C.7, the lower wall of the corridor of the gadget is a horizontal edge c_0c_1 of length 2, where c_0 is on the line ℓ_r and c_1 is to the right of c_0. The upper wall of the corridor is more complicated, and it will be described later. It has the left endpoint at $d_0 := c_0 + (0, \frac{3}{CN^2})$, and the right endpoint $d_1 := c_1 + (0, \frac{1.5}{CN^2})$. The vertical line segments c_0d_0 and c_1d_1 are called the *entrances* of the corridor.

C.9.1 Idea of the copying construction

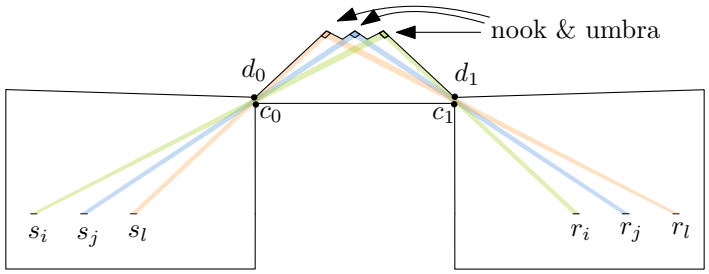


Figure 17: In this figure, we display a simplified corridor construction. The vertices c_0, c_1 serve as shadow corners for three copy-umbras simultaneously for the pairs $(s_i, r_i), (s_j, r_j), (s_l, r_l)$. Each of these pairs also have a small copy-nook in the top of corridor. The entrances c_0d_0 and c_1d_1 to the corridor are sufficiently small so that the critical segments of the nook and umbra of each pair of segments s_σ, r_σ (contained in the small boxes at the top of the figure) are not seen by other guard segments.

To ensure that the segments r_i, r_j, r_l are copies of the segments s_i, s_j, s_l, we need to construct within the corridor copy-nooks and copy-umbras for the pairs of corresponding segments, see Figure 17 for a simplified illustration. The vertices c_0, c_1 of the corridor act as shadow corners in three overlapping copy-umbras for the pairs (s_i, r_i), (s_j, r_j), and (s_l, r_l), respectively. We construct the chain of $\mathcal{P}$ from d_0 to d_1 bounding the corridor from above so that it creates three copy-nooks for the same pairs. To enforce that for any guard set of size $g(\Phi)$, for each $\sigma \in \{i, j, l\}$ the guard segments s_σ and r_σ specify the same value, we have to ensure that no guards on guard segments other than s_σ and r_σ can see the critical segments of the copy-umbra and the copy-nook of the pair s_σ, r_σ. For each $\sigma \in \{i, j, l\}$ we also introduce a stationary

guard position, so that guards placed at these positions together see all the copy-umbras, but nothing on the other sides of the critical segments of the copy-umbras. We also need to ensure that the guards placed at the stationary guard positions cannot see the critical segments of the copy-umbras and the copy-nooks of other pairs. We then obtain (see Lemma 24) that for any guard set with one guard at each guard segment, and with no guards placed outside of the guard segments and stationary guard positions, the segments s_σ, r_σ specify the value of x_σ consistently.

Our construction will ensure that for any $\sigma \in \{i, j, l\}$, only the guard segment s_σ from the base line, and only the guard segment r_σ from within the gadget can see the critical segments of the corresponding copy-nook and copy-umbra. In particular, we will ensure that the vertical edge of $\mathcal{P}$ directly above the entrance c_0d_0 blocks visibility from all guard segments $s_{\sigma'}$ for $\sigma' \in \{0, \ldots, \sigma - 1\}$, whereas the vertical edge of $\mathcal{P}$ directly below c_0d_0 blocks visibility from all guard segments $s_{\sigma'}$ for $\sigma' \in \{\sigma + 1, \ldots, 4n - 1\}$. An analogous property will be ensured for the gadget guard segments.

The main idea to achieve the above property is to make the entrances c_id_i of the corridor sufficiently small. However, we cannot place the point d_0 arbitrarily close to c_0, since both endpoints a_σ and b_σ of the segment s_σ have to see the left endpoint of the critical segment of the copy-umbra, and the right endpoint of the critical segment of the copy-nook for the pair s_σ, r_σ (the points f_0 and f_1, respectively, in the context of Section C.5). By placing the corridor sufficiently far away from the segments on the base line, we obtain that the visibility lines from the guard segment endpoints through the points c_0, d_0 are almost parallel and can be described by a simple pattern. The same holds for the pair of points d_1 and c_1 and the guard segment r_σ. The pattern enables us to construct the corridor with the desired properties.

In the following, we introduce objects that make it possible to describe the upper corridor wall and prove that the construction works as intended.

C.9.2 Introducing slabs

In a small area around the point $\frac{c_0+c_1}{2} + (0, 1)$, every ray from an endpoint of a base line guard segment through one of the points c_0, d_0 intersects every ray from an endpoint of a gadget guard segment through one of the points c_1, d_1. These rays intersect at angles close to $\pi/2$, and they form an arrangement consisting of quadrilaterals, creating a nearly-regular pattern. However, the arrangement of rays is not completely regular. We therefore introduce a collection of thin slabs, where each slab contains one of the rays in a small neighbourhood around $\frac{c_0+c_1}{2} + (0, 1)$, and such that the slabs form an orthogonal grid with axis $(1, 1)$ and $(-1, 1)$. Thus, the slabs are introduced in order to handle the "uncertainty" and irregularity of the rays.

Given a point q and a vector v, the *slab* $S(q, v, r)$ consists of all points at a distance of at most r from the line through q parallel to v. The *center* of the slab $S(q, v, r)$ is the line through q parallel to v.

Let $r_i := a'_ib'_i, r_j := a'_jb'_j, r_l := a'_lb'_l$. Let ℓ_c be a vertical line passing through the middle of the segment c_0c_1. Recall that here we describe the construction of a corridor to be attached at the right side of $\mathcal{P}$. Let o be the intersection point of the ray $\overrightarrow{a_0c_0}$ with ℓ_c, and o' the intersection point of the ray $\overrightarrow{b'_lc_1}$ with ℓ_c. All the points $o, o', \frac{c_0+c_1}{2} + (0, 1)$ lie on the vertical line ℓ_c.

Let us define vectors $\alpha := (1, 1)$, $\beta := (-1, 1)$, and introduce a grid of slabs parallel to α and β. Let us fix $\delta := \frac{13.5}{CN^2}$, $\rho := \frac{\delta}{9} = \frac{1.5}{CN^2}$, and $\varepsilon := \frac{\rho}{12} = \frac{1}{8CN^2}$. For each $\sigma \in \{0, \ldots, 4n - 1\}$ and $\gamma \in \{0, 1, 2, 3\}$ we define a slab

$$L^\gamma_\sigma := S(o + (0, \sigma\delta + \gamma\rho), \alpha, \varepsilon),$$

which we denote as an *L-slab*. Let $\tau(i) := 2$, $\tau(j) := 1$, and $\tau(l) := 0$. For each $\sigma \in \{i, j, l\}$ and $\gamma \in \{0, 1, 2, 3\}$ we define a slab

$$R^\gamma_\sigma := S(o' + (0, \tau(\sigma)\delta + \gamma\rho), \beta, \varepsilon),$$

which we denote as an *R-slab*.

In the case of gadgets with just two guard segments r_i, r_j, we define the point o' as the intersection point of the ray $\overrightarrow{b'_j c_1}$ with ℓ_c, and we define $\tau(i) := 1$ and $\tau(j) := 0$. Then, the R-slabs R^γ_σ are defined as above for $\sigma \in \{i, j\}$.

See Figure 18 for an illustration of the area where the L-slabs intersect the R-slabs.

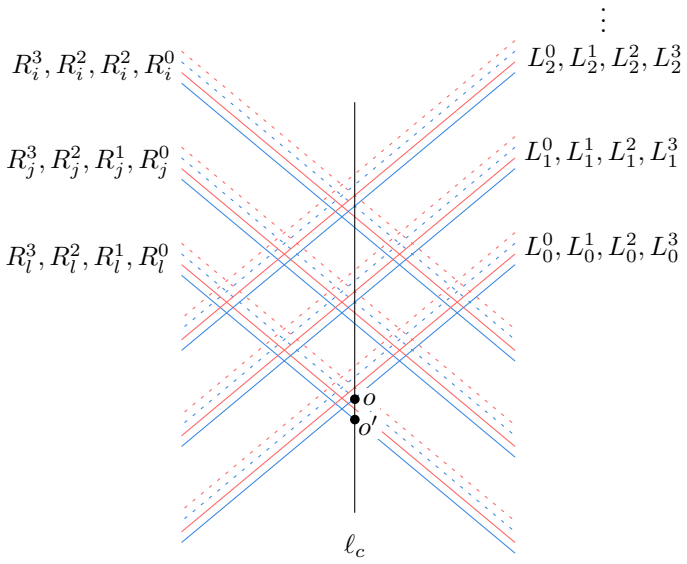


Figure 18: The L-slabs have slope 1 and the R-slabs have slope -1. For each guard segment we get 4 equidistant slabs. The width of each slab is 2ε. The distance between two slabs from the same group is ρ and the distance between two groups is δ. All intersections are contained in the region denoted by V.

Let V be the square $\frac{c_0+c_1}{2} + (0,1) + [-38N\rho, 38N\rho] \times [-38N\rho, 38N\rho]$. We now prove that the area where the L-slabs and the R-slabs intersect is contained in V. Let us denote by $\mathcal{R}$ all the rays with endpoint at one of the guard segments in the main area, going through c_0 and d_0 and all the rays from the endpoints of gadget guard segments through the points c_1, d_1. We also ensure that all rays in $\mathcal{R}$ are inside a predefined slab within the area V.

In sections specific to the particular gadgets, we will prove the following lemma. We state the lemma for gadgets with three guard segments r_i, r_j, r_l, but it has a natural analogy for gadgets with just two guard segments r_i, r_j.

Lemma 21. *For any gadget to be attached at the right side of the polygon $\mathcal{P}$ and containing the guard segments $r_i = a'_i b'_i, r_j = a'_j b'_j, r_l = a'_l b'_l$ the following holds, where c_1 is the bottom-right endpoint of the corridor corresponding to the gadget.*

1. *The intersection of any R-slab with the line ℓ_c is contained in V.*
2. *For each $\sigma \in \{i, j, l\}$, it holds that $\overrightarrow{b'_\sigma c_1} \cap V \subset R^0_\sigma, \overrightarrow{a'_\sigma c_1} \cap V \subset R^1_\sigma, \overrightarrow{b'_\sigma d_1} \cap V \subset R^2_\sigma$, and $\overrightarrow{a'_\sigma d_1} \cap V \subset R^3_\sigma$.*
3. *There are no stationary guard positions or guard segments different from r_i, r_j, r_l within the gadget from which any point of the corridor can be seen.*

Assuming that the above lemma holds, we will prove the following.

Lemma 22. *For any corridor to be attached at the right side of the polygon $\mathcal{P}$, the following properties are satisfied.*

1. *The intersection of any L-slab with any R-slab is contained in V.*

2. *For each* $\sigma \in \{0, \dots, 4n-1\}$*, it holds that* $\overrightarrow{a_\sigma c_0} \cap V \subset L^0_\sigma, \overrightarrow{b_\sigma c_0} \cap V \subset L^1_\sigma, \overrightarrow{a_\sigma d_0} \cap V \subset L^2_\sigma$*, and* $\overrightarrow{b_\sigma d_0} \cap V \subset L^3_\sigma$*.*

Proof. We will first show that the intersection of any L-slab with the line ℓ_c is contained in V. Consider Figure 19. Recall that o is the intersection point of the ray $\overrightarrow{a_0 c_0}$ with ℓ_c. The horizontal distance between ℓ_c and c_0 is 1. Let u be the intersection point of the lines ℓ_b and ℓ_r. From the polygon description in Section C.7 we know that $\|a_0 u\| = CN^2$, and $\|c_0 u\| \in [CN^2, CN^2 + 3k'] \subseteq [CN^2, CN^2 + 3N]$. The distance between the point $\frac{c_0+c_1}{2}$ and the point o is $\frac{\|c_0 u\|}{\|a_0 u\|} \in [1, 1 + \frac{3N}{CN^2}] = [1, 1 + 2N\rho]$. From the definition of the L-slabs, the intersection of the L-slabs with the vertical line ℓ_c is contained in the line segment $(o - (0, 2\varepsilon), o + (0, 4N\delta)) \subseteq (\frac{c_0+c_1}{2} + (0, 1 - 2\varepsilon), \frac{c_0+c_1}{2} + (0, 1 + 2N\rho + 4N\delta))$, which is contained in V as $\delta = 9\rho$.

By Property 1 of Lemma 21, any point in the intersection of an L-slab and an R-slab must have a y-coordinate within the range of V. As the angles of the slabs are exactly $\pi/4$ and $3\pi/4$, we get that also the x-coordinates of the intersection must be within the range of V, see also to the right of Figure 19. That gives us Property 1.

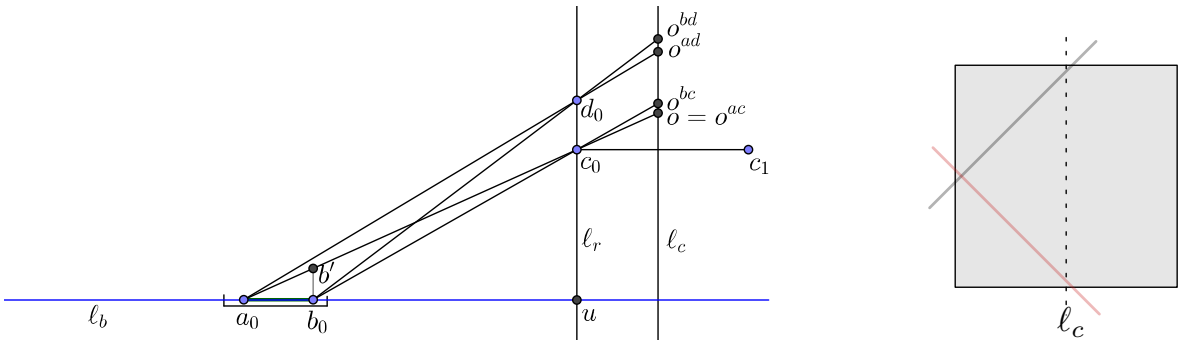


Figure 19: Left: Structure of rays with origins a_0 and b_0, containing points c_0 and d_0. Right: Even in the case that a left ray intersects ℓ_c at the very top of V and a right ray intersects at the very bottom of V, they still have to intersect within V.

For Property 2, let us first consider $\sigma = 0$. Let us define $o^{ac}, o^{ad}, o^{bc}, o^{bd}$ as the intersection points of the rays $\overrightarrow{a_0 c_0}, \overrightarrow{a_0 d_0}, \overrightarrow{b_0 c_0}, \overrightarrow{b_0 d_0}$ with the line ℓ_c (see Figure 19). We have $o = o^{ac}$. The points o^{ad}, o^{bc}, o^{bd} lie above o, and we will now estimate the distance between each of them and o. We do that as follows.

First, consider the distance $\|oo^{ad}\|$. From below, we have a trivial bound

$$\|oo^{ad}\| \geq \|c_0 d_0\| = \frac{3}{CN^2} = 2\rho.$$

From the similarity of triangles $a_0 c_0 d_0$ and $a_0 o o^{ad}$, and as the distance between the line ℓ_c and the line ℓ_r equals 1, we obtain the following upper bound for $\|oo^{ad}\|$

$$\|oo^{ad}\| = \|c_0 d_0\| \cdot \frac{\|a_0 u\| + 1}{\|a_0 u\|} = \frac{3}{CN^2} \cdot \frac{CN^2 + 1}{CN^2} = \frac{3}{CN^2} \cdot \left(1 + \frac{1}{CN^2}\right) \leq 2\rho + \frac{\varepsilon}{6}.$$

Let b' be a vertical projection of b_0 on the ray $\overrightarrow{a_0 c_0}$. From similarity of triangles $c_0 o o^{bc}$ and $c_0 b' b_0$, and triangles $a_0 b_0 b'$ and $a_0 u c_0$, we get the following equality

$$\|oo^{bc}\| = \|b_0 b'\| \cdot \frac{1}{\|b_0 u\|} = \|c_0 u\| \cdot \frac{\|a_0 b_0\|}{\|a_0 u\|} \cdot \frac{1}{\|b_0 u\|} = 3/2 \cdot \frac{\|c_0 u\|}{\|a_0 u\| \|b_0 u\|}.$$

We instantly get

$$\|oo^{bc}\| \geq 3/2 \cdot \frac{CN^2}{(CN^2)^2} = \rho.$$

For an upper bound, we compute

$$\frac{\|a_0u\|\|b_0u\|}{\|c_0u\|} \geq \frac{CN^2(CN^2-3/2)}{CN^2+3N} \geq \frac{(CN^2+3N)(CN^2-4N)}{CN^2+3N} = CN^2-4N \geq \frac{CN^2}{1+1/72},$$

where the last inequality follows since $C \geq 73 \cdot 4 = 292$. That gives us

$$\|oo^{bc}\| \leq 3/2 \cdot \frac{1+1/72}{CN^2} = \frac{3/2}{CN^2} + \frac{1/48}{CN^2} = \rho + \frac{\varepsilon}{6}.$$

In the same way as for $\|oo^{bc}\|$ we obtain the following bounds

$$\rho \leq \|o^{ad}o^{bd}\| \leq \rho + \varepsilon/6.$$

As $\|oo^{bd}\| = \|oo^{ad}\| + \|o^{ad}o^{bd}\|$, we instantly get

$$3\rho \leq \|oo^{bd}\| \leq 3\rho + \varepsilon/3.$$

Summarizing this part, we have the following bounds:

$$\rho \leq \|oo^{bc}\| \leq \rho + \varepsilon/3,\ 2\rho \leq \|oo^{ad}\| \leq 2\rho + \varepsilon/3,\ 3\rho \leq \|oo^{bd}\| \leq 3\rho + \varepsilon/3.$$

Therefore, the intersection points of the rays $\overrightarrow{a_0c_0}, \overrightarrow{b_0c_0}, \overrightarrow{a_0d_0}, \overrightarrow{b_0d_0}$ are contained in the required slabs, at a vertical distance of at most $\varepsilon/3$ from the centers of the slabs.

Now, consider any $\sigma \in \{0, \ldots, 4n-1\}$, $\tau \in \{a, b\}$ and $\eta \in \{c, d\}$. Let us denote by $\mathcal{R}$ all the rays with endpoint at one of the guard segments in the main area, going through c_0 and d_0 and all the rays from the endpoints of gadget guard segments through the points c_1, d_1.

Let $\widehat{o}$ be the intersection point of the ray $\overrightarrow{\tau_\sigma\eta_0}$ with the line ℓ_c (see Figure 20). We first

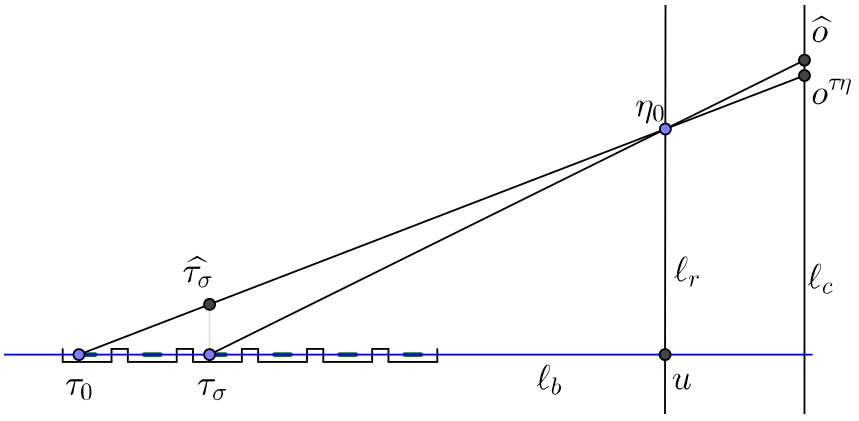


Figure 20: Bounding $\|o\widehat{o}\|$.

bound the distance $\|o^{\tau\eta}\widehat{o}\|$. Recall that u is the intersection point of ℓ_b and ℓ_r. Let $\widehat{\tau_\sigma}$ be the point on the ray $\overrightarrow{\tau_0\eta_0}$ vertically above τ_σ.

As the triangles $\eta_0 o^{\tau\eta}\widehat{o}$ and $\eta_0\widehat{\tau_\sigma}\tau_\sigma$ are similar, the triangles $\tau_0\tau_\sigma\widehat{\tau_\sigma}$ and $\tau_0 u\eta_0$ are similar, and the distance between the lines ℓ_c and ℓ_r is 1, we get the following equality

$$\|o^{\tau\eta}\widehat{o}\| = \|\tau_\sigma\widehat{\tau_\sigma}\| \cdot \frac{1}{\|\tau_\sigma u\|} = \|u\eta_0\| \cdot \frac{\|\tau_0\tau_\sigma\|}{\|\tau_0 u\|} \cdot \frac{1}{\|\tau_\sigma u\|} = 13.5\sigma \cdot \frac{\|u\eta_0\|}{\|\tau_0 u\|\|\tau_\sigma u\|}.$$

We first bound $\|o^{\tau\eta}\widehat{o}\|$ from above. As $C \geq 1297 \cdot 58 = 75226$, we get

$$\begin{aligned}\frac{\|\tau_0 u\|\|\tau_\sigma u\|}{\|u\eta_0\|} &\geq \frac{(CN^2 - 3/2)(CN^2 - 54N)}{CN^2 + 3N} \geq \frac{(CN^2 + 3N)(CN^2 - 58N)}{CN^2 + 3N} \\ &= CN^2 - 58N \geq \frac{CN^2}{1 + \frac{1}{1296N}},\end{aligned}$$

and, as $\sigma \leq 4N$, we can bound

$$\|o^{\tau\eta}\widehat{o}\| \leq 13.5\sigma \cdot \frac{1 + \frac{1}{1296N}}{CN^2} \leq \sigma \cdot \frac{13.5}{CN^2} + \frac{1/24}{CN^2} = \sigma\delta + \frac{\varepsilon}{3}.$$

To bound $\|o^{\tau\eta}\widehat{o}\|$ from below, we compute

$$\|o^{\tau\eta}\widehat{o}\| = 13.5\sigma \cdot \frac{\|u\eta_0\|}{\|\tau_0 u\|\|\tau_\sigma u\|} \geq 13.5\sigma \cdot \frac{CN^2}{(CN^2)^2} = \sigma\delta.$$

Therefore, as $\|o\widehat{o}\| = \|o^{\gamma\eta}\widehat{o}\| + \|oo^{\gamma\eta}\|$, the intersection points of the rays $\overrightarrow{a_\sigma c_0}, \overrightarrow{b_\sigma c_0}, \overrightarrow{a_\sigma d_0}, \overrightarrow{b_\sigma d_0}$ are contained in the required slabs, at a vertical distance of at most $2\varepsilon/3$ from the centers of the slabs.

We now need to verify that the rays stay in their respective slabs within the range of the x-coordinates of V. We therefore bound the slope of a ray $\overrightarrow{\tau_\sigma\eta_0}$ for any $\sigma \in \{0, \ldots, 4n-1\}$, $\tau \in \{a, b\}$ and $\eta \in \{c, d\}$. A bound from below is

$$\frac{\|u\eta_0\|}{\|\tau_\sigma u\|} \geq \frac{CN^2}{CN^2} \geq 1.$$

To bound the slope of $\overrightarrow{\tau_\sigma\eta_0}$ from above, we compute (as $C \geq 57 \cdot 1368 + 54 = 78030$)

$$\frac{\|u\eta_0\|}{\|\tau_\sigma u\|} \leq \frac{CN^2 + 3N}{CN^2 - 54N} \leq 1 + \frac{57N}{CN^2 - 54N} \leq 1 + \frac{1}{1368N}.$$

The center of each L-slab has the slope equal to 1. As the vertical distance between $\widehat{o}$ and the center of the slab is at most $2\varepsilon/3$, and $\frac{1}{1368N} \cdot 38N\rho = \frac{\rho}{36} = \frac{\varepsilon}{3}$, we get that $\overrightarrow{\tau_\sigma\eta_0} \cap V$ is contained in the corresponding slab. □

C.9.3 Constructing the corridor using slabs

We are now ready to describe the exact construction of the corridor. As mentioned before, the bottom wall is simply the line segment c_0c_1. We first describe the approximate shape of the upper wall, defined by a polygonal curve Λ connecting the points d_1 and d_0. Later we will present how to modify Λ into a final polygonal curve Λ', which is exactly the upper wall of the corridor.

Note that in the corridor construction here we assume that $i < j < l$. In particular, the L-slabs L_l^γ are above the L-slabs L_j^γ, which are above L_i^γ. For the R-slabs it is the opposite, i.e., the R-slabs R_l^γ are below the R-slabs R_j^γ, which are below R_i^γ.

Figure 21 shows the grid of slabs and a sketch of the curve Λ approximating the upper wall (excluding most of the leftmost and rightmost edge of Λ, with endpoints at d_0 and d_1,

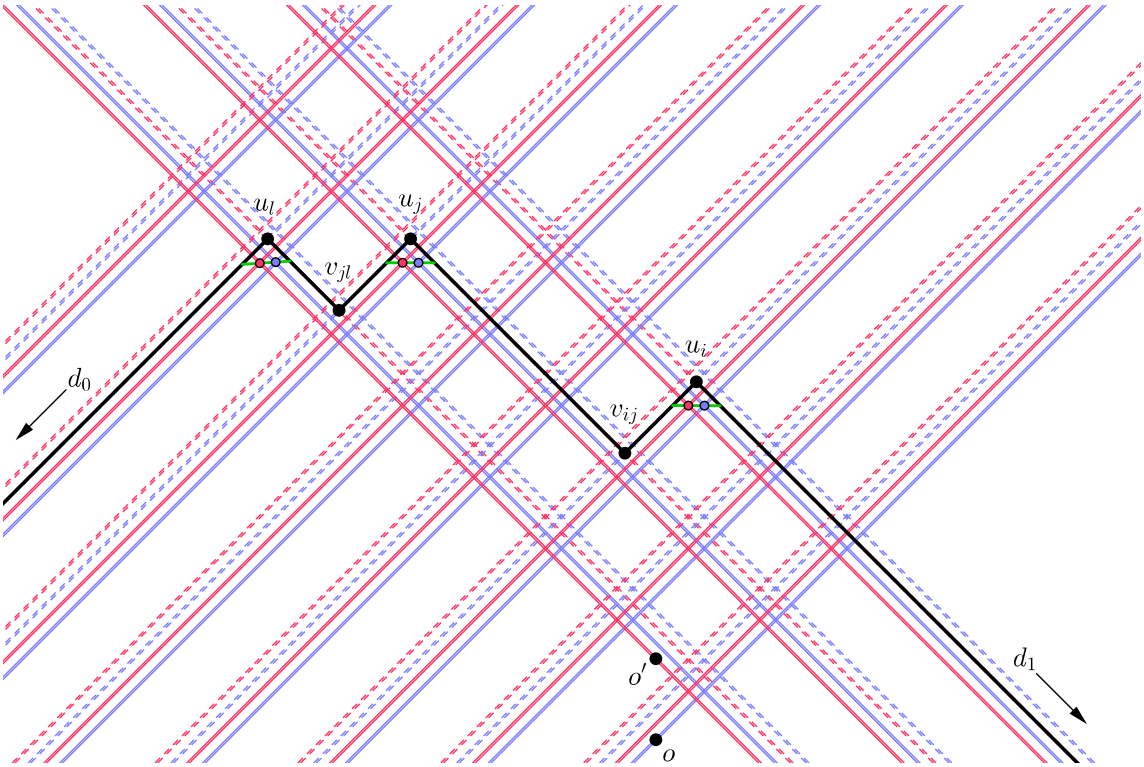


Figure 21: The grid of L- and R-slabs (bounded by blue and red lines) and an approximate shape Λ (black) of the upper wall of the corridor copying the guard segments s_i, s_j, s_l to the guard segments r_i, r_j, r_l in the gadget. Here, there are 8 guard segments $s_0, \ldots, s_7$ on the base line, and we have $i = 2$, $j = 5$, and $l = 6$. The blue lines bound the slabs corresponding to the rays originating at the left endpoints of the guard segments (i.e., slabs $L^0_\sigma, L^2_\sigma, R^1_\sigma, R^3_\sigma$), and the red lines bound the slabs corresponding to the rays originating at the right endpoints. The full lines bound the slabs corresponding to the rays passing through c_0 or c_1 (i.e., slabs $L^0_\sigma, L^1_\sigma, R^0_\sigma, R^1_\sigma$), and the dashed lines bound the slabs corresponding to the rays passing through d_0 or d_1. The blue points and the red points are the intersections of the rays $\overrightarrow{a_\sigma c_0} \cap \overrightarrow{a'_\sigma c_1}$ and $\overrightarrow{b_\sigma c_0} \cap \overrightarrow{b'_\sigma c_1}$, respectively, for $\sigma \in \{i, j, l\}$. The green segments each contains a critical segment (the part between the blue and the red point) of a copy-umbra for s_σ and r_σ with shadow corners c_0, c_1.

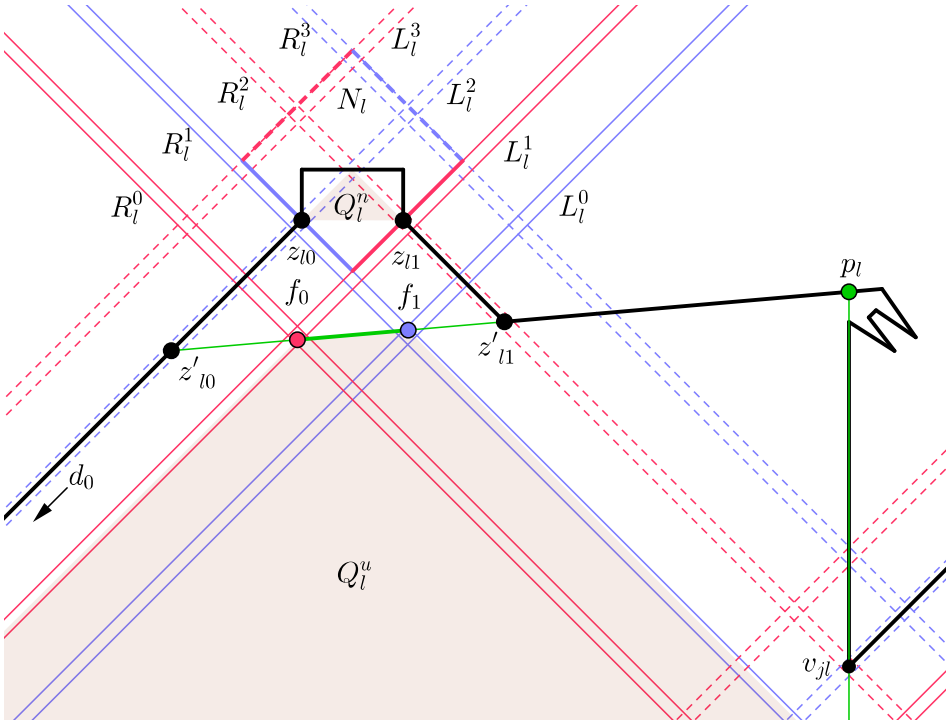


Figure 22: A closeup of the upper wall of the corridor from Figure 21 with features for copying the leftmost pair s_l, r_l of guard segments. The curve Λ' is drawn in black. The boundary of the square N_l is drawn with thick line segments. The points z_{l0}, z_{l1} are the shadow corners of the copy-nook Q_l^n (brown area) of s_l, r_l. The critical segment of Q_l^n is on the topmost black segment and it is so short that it appears as if it was just a point. The green line segment f_0f_1 is the critical segment of a copy-umbra Q_l^u of s_l, r_l with shadow corners c_0, c_1. The point p_l is a stationary guard position, from which a guard can see the area below the segment $z_{l0}z_{l1}$ containing f_0f_1. Furthermore, p_l sees the area to the left of the vertical ray emitting downwards from p_l.

respectively, since they are too long to be pictured together with the middle segments). For $\sigma \in \{i, j, l\}$, let u_σ be the intersection point of the rays $\overrightarrow{a_\sigma d_0}$ and $\overrightarrow{b'_\sigma d_1}$. Let v_{ij} be the intersection point of the rays $\overrightarrow{a_i d_0}$ and $\overrightarrow{b'_j d_1}$, and v_{jl} the intersection point of the rays $\overrightarrow{a_j d_0}$ and $\overrightarrow{b'_l d_1}$. The curve Λ is then a path defined by the points $d_1u_iv_{ij}u_jv_{jl}u_ld_0$. By Lemma 22, $\Lambda \cap V$ is contained in the union of the L-slabs and the R-slabs, as shown in Figure 21. Due to the relative position of the slabs $L_l^\gamma, L_j^\gamma, L_i^\gamma$ and $R_l^\gamma, R_j^\gamma, R_i^\gamma$ as discussed above, the curve Λ is x-monotone, and the point v_{ij} (resp. v_{jl}) has smaller y-coordinate than the neighbouring points u_i, u_j (resp. u_j, u_l), i.e., the curve Λ always has a zig-zag shape resembling the one from Figure 21.

We will now show how to modify Λ by adding to the curve some features. The first modification is in order to construct copy-nooks Q_i^n, Q_j^n, Q_l^n for each of the pairs (s_i, r_i), (s_j, r_j), and (s_l, r_l), respectively. Note that the area above c_0c_1 and below Λ already contains a copy-umbra Q_σ^u for each pair s_σ, r_σ for $\sigma \in \{i, j, l\}$ with shadow corners c_0 and c_1 (as Q_σ^u is contained in the triangular area bounded above c_0c_1 and below the rays $\overrightarrow{b_\sigma c_0}, \overrightarrow{a'_\sigma c_1}$, which, due to Lemmas 21 and 22, is below Λ). The second reason why we need to modify Λ is in order to create stationary guard positions p_i, p_j, p_l that see the areas of the copy-umbras, but nothing above their critical segments. In the following, we explain how to modify the fragment of Λ consisting of the leftmost two edges, i.e., the path $v_{jl}u_ld_0$. The construction is presented in Figure 22. We then perform similar modifications for the fragments of Λ consisting of the paths $v_{ij}u_jv_{jl}$ and $d_1u_iv_{ij}$.

First, we show how to construct a copy-nook Q_l^n of s_l and r_l with shadow corners at Λ. The curve Λ will then be modified so that Q_l^n is contained within the corridor. Let N_l be the square

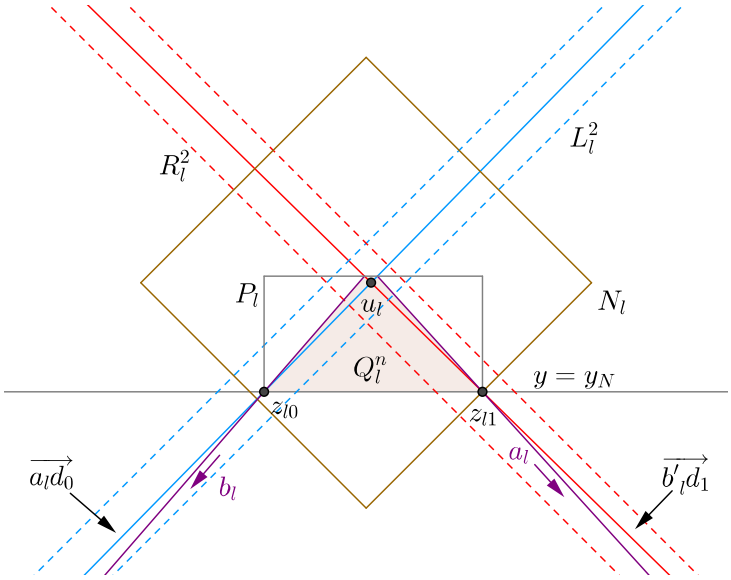


Figure 23: Construction of the nook Q_l^n.

consisting of points which are above the slabs L_l^1 and R_l^1, but not above L_l^3 or R_l^3. The square N_l is approximately the area which is seen both from the right endpoint b_l of s_l, and the left endpoint a'_l of r_l. Note that N_l contains the point u_l (as $u_l \in L_l^2 \cap R_l^2$, and thus u_l is above L_l^1 and R_l^1 and below L_l^3 and R_l^3). The copy-nook Q_l^n for the pair s_l, r_l will be created inside N_l (see Figure 23). Consider the two intersection points of the boundary of N_l with the line segments $v_{jl}u_l$ and $u_l d_0$. Let y_N be the larger of the y-coordinates of these two intersection points. The shadow corners of the nook Q_l^n are chosen as intersection points of the horizontal line $y = y_N$ with the line segments $v_{jl}u_l$ and $u_l d_0$, and they are denoted by z_{l1} and z_{l0}, respectively. In this way we ensure that both shadow corners are visible from any point within the segments s_l and r_l, and that they define a copy-nook Q_l^n for the pair of segments s_l, r_l. Note that the entire nook Q_l^n is contained in N_l, since by Lemmas 21 and 22 no point on s_l or r_l can see a point above the slabs L_l^3 or R_l^3. We now modify the curve Λ as follows. Let P_l be a quadrilateral with two vertices at z_{l0} and z_{l1}, and such that it contains vertical edges incident to z_{l0} and z_{l1}, and an edge containing the critical segment for Q_l^n. We modify Λ so that between the points z_{l0} and z_{l1}, it consists of the vertical edges and the topmost edge of P_l.

Now, consider the copy-umbra Q_l^u for the pair of segments s_l, r_l with shadow corners c_0 and c_1. Let $f_0 := \overrightarrow{b_l c_0} \cap \overrightarrow{b'_l c_1}$ and $f_1 := \overrightarrow{a_l c_0} \cap \overrightarrow{a'_l c_1}$. Note that the points f_0, f_1 correspond to the red and blue points in Figures 21 and 22. The segment $f_0 f_1$ is the critical segment for Q_l^u. By Lemmas 21 and 22, $f_0 \in L_l^1 \cap R_l^0$ and $f_1 \in L_l^0 \cap R_l^1$, and the squares $L_l^1 \cap R_l^0, L_l^0 \cap R_l^1$ have a sidelength of 2ε. Therefore the slope of the line $\overleftrightarrow{f_0 f_1}$ is in the interval $\left[-\frac{2\sqrt{2}\varepsilon}{\sqrt{2}\rho}, \frac{2\sqrt{2}\varepsilon}{\sqrt{2}\rho}\right] = [-1/6, 1/6]$, and this line intersects both line segments $v_{jl}u_l$ and $u_l d_0$. Let z'_{l0} and z'_{l1} be the intersection points of the line $\overleftrightarrow{f_0 f_1}$ with the line segments $u_l d_0$ and $u_l v_{jl}$, respectively. (We similarly define points z'_{i0}, z'_{i1} on $d_1 u_i v_{ij}$ as the intersection points with the line containing the critical segment of the umbra Q_i^u, and z'_{j0}, z'_{j1} on $v_{ij}u_j v_{jl}$ as the intersection points with the line containing the critical segment of the umbra Q_j^u.) We introduce a stationary guard position p_l by creating a pocket which will require modifying the curve Λ again. The pocket is extruding to the right from $v_{jl}u_l$, following the line $\overleftrightarrow{f_0 f_1}$, as pictured in Figure 22. Likewise, it is extruding vertically up from v_{jl}. The pocket contains a stationary guard position p_l on the line $\overleftrightarrow{f_0 f_1}$. Clearly, a guard placed at p_l sees nothing above the line segment $f_0 f_1$. Note that it sees the part of Q_l^u to the left of the

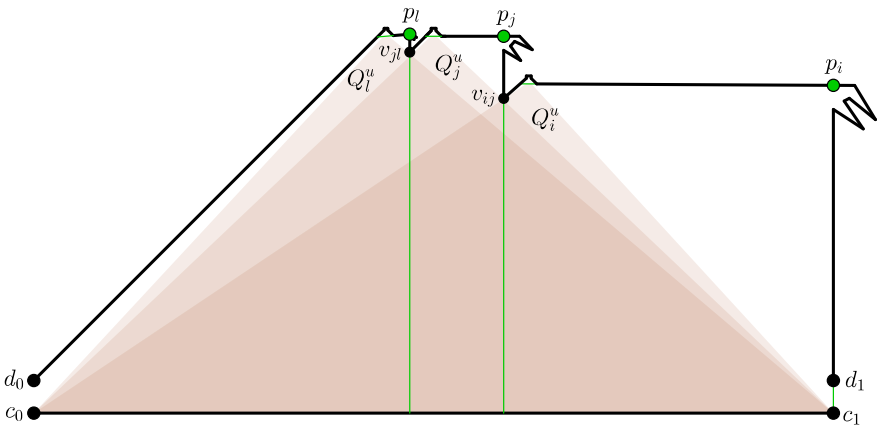


Figure 24: The complete construction of the corridor.

vertical line through v_{jl}.

For the middle two edges $v_{ij}u_jv_{jl}$ of Λ, we place the stationary guard position p_j vertically above v_{ij} so that it sees an area below the critical segment of the umbra Q_j^n and to the left of the vertical line through v_{ij}. For the rightmost edges $d_1u_iv_{jl}$, we place the stationary guard position p_i vertically above d_1 so that it sees an area below the critical segment of the umbra Q_i^u to the left of the vertical line through d_1. Let Λ' be the wall obtained by doing the modifications to Λ described here, and let C be the corridor, that is, the area bounded by the lower wall c_0c_1, the upper wall Λ' between d_0 and d_1, and by the vertical entrance segments c_0d_0 and c_1d_1. See Figure 24 for a picture of the complete corridor.

Lemma 23. *The stationary guard positions p_i, p_j, p_l have the following three properties.*

- *The three stationary guard positions p_i, p_j, p_l see together all of the corridor except the points above the segments $z'_{i0}z'_{i1}$, $z'_{j0}z'_{j1}$, $z'_{l0}z'_{l1}$.*
- *None of the guards can see anything to the right of the right entrance c_1d_1.*
- *None of the stationary guard positions p_i, p_j, p_l for the pairs $(s_i, r_i), (s_j, r_j), (s_l, r_l)$, respectively, can see any point on the critical segment of the nook or umbra of one of the other pairs.*

Proof. See Figure 24. For the first claim, note that the vertical lines through v_{jl} and v_{ij} divide the corridor into three parts. It is now clear that all points in the leftmost part below $z'_{l0}z'_{l1}$ are seen by p_l, all points in the middle part below $z'_{j0}z'_{j1}$ are seen by p_j, and all points in the rightmost part below $z'_{i0}z'_{i1}$ are seen by p_i.

For the second part, observe that the point p_i cannot see any point to the right of the vertical line through d_1, and the visibility of p_j and p_l is bounded by vertical lines more to the left.

For the last part, we note that the curve Λ' passes through points v_{jl} and v_{ij}, blocking visibility between stationary guard positions and critical segments corresponding to different pairs. □

Lemma 24. *Suppose that in each of the pairs (s_i, r_i), (s_j, r_j), (s_l, r_l) of guard segments corresponding to a corridor C, the two segments have the same orientation. Then C satisfies the following properties.*

1. *In any guard set G of $\mathcal{P}$ there are at least 3 guards placed within the corridor C, and if there are exactly 3 then they are placed at the stationary guard positions p_i, p_j, p_l. (The number is 2 instead of 3 if we construct C to copy only two segments.)*

2. *Let G be any set of points with exactly one guard on each guard segment and each stationary guard position, and with no guards outside of stationary guard positions and guard segments. If the whole area of C is seen by G, then for each of the pairs (s_i, r_i), (s_j, r_j), (s_l, r_l) the two guards on the segments specify the same value.*

3. *For any set of points G which satisfies the properties: (i) there is a guard at each point p_i, p_j, p_l and at each guard segment s_i, s_j, s_l and r_i, r_j, r_j, and (ii) the values specified by the pairs of segments s_i, s_j, s_l and r_i, r_j, r_j are consistent, G sees the whole area of C.*

4. *No guard at a stationary guard position or a guard segment outside the gadget can see any point inside the gadget below the line $\overleftrightarrow{d_0 c_1}$.*

5. *The vertical distance from $c_0 c_1$ to the topmost point of the corridor is at most* 1.4.

Proof. For Property 1, note that the points defining the stationary guards within C can be seen only from within C. We can now use Lemma 14, setting A as the corridor area and choosing the points defining the stationary guards to construct the set M, to prove the desired property.

For Property 2, consider the set G as described, and let $\sigma \in \{i, j, l\}$. The stationary guard positions p_i, p_j, p_l cannot see any points above the line containing the critical segments of the umbra Q^u_σ. Lemma 23 and Property 3 of Lemma 21 give us that guards at the guard segments $s_0, s_1, \ldots, s_{4n-1}, r_i, r_j, r_l$ must see the critical lines of the nook and umbra Q^n_σ, Q^u_σ. We will now show that among all these segments, only guards placed at the segments s_σ, r_σ are able to do so. From Lemma 22 we get that for any $\sigma' > \sigma$, no guard on the guard segment $s_{\sigma'}$ can see a point in the square V below $L^0_{\sigma'}$. As $L^0_{\sigma'}$ is above L^3_σ, it is also above the critical segments of the nook Q^n_σ and the umbra Q^u_σ of the pair s_σ, r_σ. Likewise, for any $\sigma' < \sigma$, no guard on the guard segment $s_{\sigma'}$ can see a point in V above $L^3_{\sigma'}$. As $L^3_{\sigma'}$ is below L^0_σ, it is also below the critical segments of the pair s_σ, r_σ. A similar argument shows that among the guard segments r_i, r_j, r_l, only guards on r_σ can see any points on the critical segments for the pair s_σ, r_σ. Therefore the two guards on s_σ, r_σ must see together both critical segments of that pair, and by Lemma 17 the guards must specify the same value.

For Property 3, consider a set G satisfying (i) and (ii). By Lemma 23, the stationary guards can see the whole area of C except of the points which are above the line segments $z'_{i0}z'_{i1}, z'_{j0}z'_{j1}, z'_{l0}z'_{l1}$ containing the critical segments of the umbras. Consider any $\sigma \in \{i, j, l\}$, and the guard segments s_σ, r_σ. As these guards can see the complete critical segment for the umbra Q^u_σ, they can see the whole area contained above the critical segment and below Λ. As they can see the complete critical interval for the nook Q^n_σ, they can see the whole area of the polygon P_σ. Therefore, they can see the whole area above the critical interval and below Λ'.

Property 4 is a clear consequence of Lemma 23.

For Property 5, note that the top wall of the corridor can only extend beyond the square V due to a part of the wall Λ' creating a stationary guard position. Recall that the slope of the critical segments of the umbras Q^u_i, Q^u_j, Q^u_l is in the range $[-1/6, 1/6]$. Since $\|c_0 c_1\| = 2$, it follows that any point on Λ' is at height at most $1 + 38N\rho + 2 \cdot 1/6 < 1.4$ above $c_0 c_1$.

□

Lemma 25. *Assume that the endpoints of guard segments corresponding to a corridor C are at rational points, with the enumerators and the denominators upper-bounded by $(\zeta C N^2)^{O(1)}$. Then, we can construct the corridor C in such a way that each vertex of C has rational coordinates, with the enumerator and the denominator upper-bounded by $(\zeta C N^2)^{O(1)}$. The corridor construction can be done in polynomial time.*

Proof. Note that the entrances to the corridor are also at rational points, with enumerators and denominators upper-bounded by $(\zeta C N^2)^{O(1)}$. Therefore, each of the lines defining the polygonal curve Λ is defined by two rational points with this property. The same holds for the lines bounding the L-slabs and the R-slabs.

Consider the construction of a copy-nook Q^n_σ within the corridor. The vertices of Q^n_σ are then at points which are defined as intersection of two lines, where each line is defined by two rational points with enumerators and denominators upper-bounded by $(\zeta CN^2)^{O(1)}$. Therefore, the vertices of the nook, and therefore also the vertices of the quadrilateral P_σ are also of this form.

The stationary guard positions p_i, p_j, p_l are the intersection points of two lines, again each line defined by two points with the above property. Therefore, the enumerators and denominators of the stationary guard positions are also upper-bounded by $(\zeta CN^2)^{O(1)}$. The vertices of the pockets corresponding to p_i, p_j, p_l can be chosen with much freedom, and therefore they can also be at points satisfying the lemma statement. □

C.9.4 Corridor construction for gadgets at the left side of $\mathcal{P}$

For the gadgets attached at the left side of the polygon $\mathcal{P}$, the construction of the corridor is analogous. Now the points c_0, d_0 lie on the line ℓ_l instead of ℓ_r, and the points c_1, d_1 are to the left of c_0, d_0. As we want the points o (o') to correspond to the lowest intersection point of a ray from an endpoint of a guard segment in the base line (in the gadget, respectively) containing the point c_0 (c_1, respectively) with ℓ_c, we redefine these points in the following way. The point o is the intersection point of the ray $\overrightarrow{b_{4n-1}c_0}$ with ℓ_c, and o' is the intersection point of the ray $\overrightarrow{a'_l c_1}$ with ℓ_c. As we want the slabs L^γ_σ to contain fragments of rays from the endpoints of the segment s_σ, we redefine

$$L^\gamma_\sigma := S(o + (0, (4n - 1 - \sigma)\delta + \gamma\rho), \beta, \varepsilon).$$

Similarly, we redefine

$$R^\gamma_\sigma := S(o' + (0, \tau(\sigma)\delta + \gamma\rho), \alpha, \varepsilon).$$

As now the left endpoints of the gadget guard segments are further away from the line ℓ_c than the right endpoints, each gadget attached to the left side of $\mathcal{P}$ has to satisfy the following (instead of Lemma 21).

Lemma 26. *For any gadget to be attached to the left side of the polygon $\mathcal{P}$ and containing the guard segments $r_i = a'_i b'_i, r_j = a'_j b'_j, r_l = a'_l b'_l$ the following holds, where c_1 is the bottom-left endpoint of the corridor corresponding to the gadget.*

1. *The intersection of any R-slab with the line ℓ_c is contained in V.*
2. *For each $\sigma \in \{i, j, l\}$, it holds that $\overrightarrow{a'_\sigma c_1} \cap V \subset R^0_\sigma, \overrightarrow{b'_\sigma c_1} \cap V \subset R^1_\sigma, \overrightarrow{a'_\sigma d_1} \cap V \subset R^2_\sigma$, and $\overrightarrow{b'_\sigma d_1} \cap V \subset R^3_\sigma$.*
3. *There are no stationary guard positions or guard segments different from r_i, r_j, r_l within the gadget, from which any point of the corridor can be seen.*

For the same reason, instead of Lemma 22 we get the following lemma.

Lemma 27. *Within any corridor to be attached to the left side of the polygon $\mathcal{P}$ the following properties are satisfied.*

1. *The intersection of any L-slab with any R-slab is contained in V.*
2. *For each $\sigma \in \{0, \ldots, 4n-1\}$, it holds that $\overrightarrow{b_\sigma c_0} \cap V \subset L^0_\sigma, \overrightarrow{a_\sigma c_0} \cap V \subset L^1_\sigma, \overrightarrow{b_\sigma d_0} \cap V \subset L^2_\sigma$, and $\overrightarrow{a_\sigma d_0} \cap V \subset L^3_\sigma$.*

Due to symmetry of $\mathcal{P}_M$, the proof of Lemma 27 is the same as the proof of Lemma 22. The corridor construction and the proof of Lemma 24 is then the same as for the corridor attached at the right side of $\mathcal{P}$. The only difference is that in order to ensure that the L-slabs L_l^γ are above the L-slabs L_j^γ, which are above L_i^γ (which is required to get a meaningful zig-zag shape of the upper wall of the corridor) we now have to assume that $i > j > l$ (as the definition of the L-slabs has changed).

C.10 The $\geq$-addition gadget

In this section we present construction of the $\geq$-addition gadget which represents an inequality $x_i + x_j \geq x_l$, where $i, j, l \in \{0, \ldots, n-1\}$. In Section C.11 we will show how to modify the construction to obtain the $\leq$-addition gadget for the inequality $x_i + x_j \leq x_l$. For any equation of the form $x_i + x_j = x_l$ in Φ we will then add both gadgets into our polygon $\mathcal{P}$.

C.10.1 Idea behind the gadget construction

We first describe the general idea behind the construction of a gadget imposing an inequality $x'_i + x_j \geq x_l$ for three variables x'_i, x_j, x_l. See Figure 25 for a sketch of the construction. Let $w, v, h > 0$ be rational values such that $w > v + 3/2$. Let r'_i, r_j, r_l be right-oriented guard segments[‡] of length $3/2$ such that r_j has its left endpoint at the point $(-w, 0)$, r'_i has its right endpoint at $(w, 0)$, and r_l has its left endpoint at $(-2, -h)$. Let $g'_i := (w - 2 + x'_i, 0)$, $g_j := (-w - 1/2 + x_j, 0)$, and $g_l := (-5/2 + x_l, -h)$ be three guards on r'_i, r_j, r_l, respectively, representing the values $x'_i, x_j, x_l \in [1/2, 2]$.

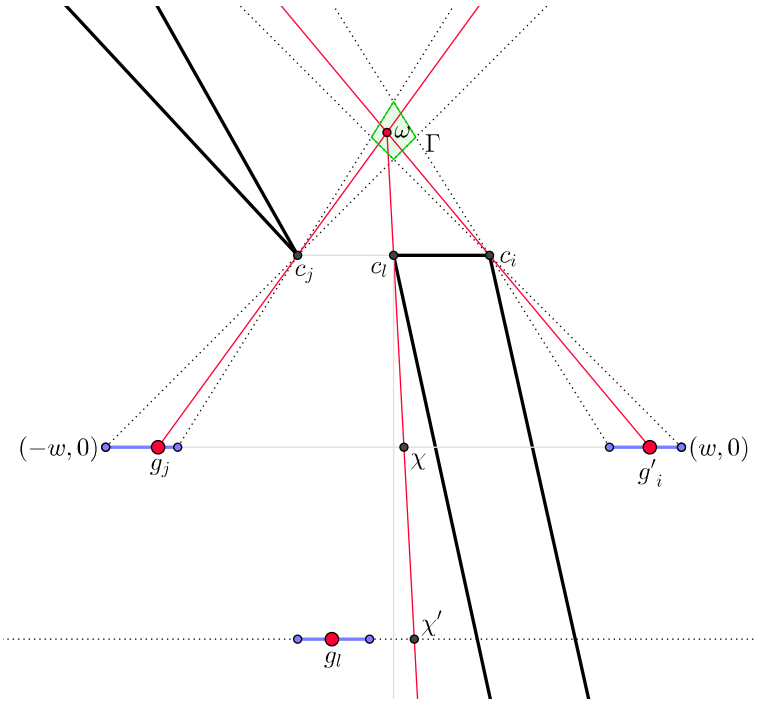


Figure 25: The thick black segments are edges of the polygon. The green quadrilateral is Γ. In order to see Γ together with g'_i and g_j, the guard g_l must be on or to the left of the point χ'.

Let $c_i := (v, h), c_j := (-v, h), c_l := (0, h)$. Let Γ be a collection of points ω such that the ray $\overrightarrow{\omega c_i}$ intersects r'_i, and the ray $\overrightarrow{\omega c_j}$ intersects r_j. Then Γ is a quadrilateral, bounded by the

[‡]We use r'_i instead of r_i here (and x'_i instead of x_i), as the guard segment r'_i specifying the value x'_i will only be a weak copy of the segment from the base line with a value x_i, i.e., it will hold that $x'_i \leq x_i$. More details will be provided later.

following rays: the rays with origin at the endpoints of r'_i and containing c_i, and the rays with origin at the endpoints of r_j and containing c_j. Suppose that

- for every point g'_i on r'_i and ω in Γ, the points ω and g'_i can see each other if and only if ω is on or to the right of the line $\overleftrightarrow{g'_i c_i}$,
- for every point g_j on r_j and ω in Γ, the points ω and g_j can see each other if and only if ω is on or to the right of the line $\overleftrightarrow{g_j c_j}$,
- for every point g_l on r_l and ω in Γ, the points ω and g_l can see each other if and only if ω is on or to the left of the line $\overleftrightarrow{g_l c_l}$.

Then, we can show the following result.

Lemma 28. *The guards g'_i, g_j, g_l can see together the whole quadrilateral Γ if and only if $x'_i + x_j \geq x_l$.*

Proof. Let $\omega \in \Gamma$ be the intersection point of the rays $\overrightarrow{g'_i c_i}$ and $\overrightarrow{g_j c_j}$.

Suppose that the guards g'_i, g_j, g_l together see the whole quadrilateral Γ. Since g'_i cannot see the area to the left of the line $\overleftrightarrow{\omega g'_i}$, and g_j cannot see the area to the left of the line $\overleftrightarrow{\omega g_j}$, there are points arbitrarily close to ω which are not seen by any of the guards g'_i, g_j. Therefore, g_l has to see ω.

Consider the rays $\overrightarrow{\omega c_i}$, $\overrightarrow{\omega c_j}$, and $\overrightarrow{\omega c_l}$. Let χ be the intersection point of the ray $\overrightarrow{\omega c_l}$ with a horizontal line $y = 0$, and χ' the intersection point of the ray $\overrightarrow{\omega c_l}$ with a horizontal line $y = -h$. Note that the guard g_l can see the point ω if and only if g_l is coincident with χ' or to the left of χ'.

From the similarity of triangles $g_j g'_i \omega$ and $c_j c_i \omega$ we get that $\frac{y(\omega)}{y(\omega)-h} = \frac{\|g'_i - g_j\|}{2v} = \frac{2w + x'_i - x_j - 3/2}{2v}$. From the similarity of triangles $g_j \chi \omega$ and $c_j c_\ell \omega$ we get that $\frac{y(\omega)}{y(\omega)-h} = \frac{\|\chi - g_j\|}{v}$, and therefore $\|\chi - g_j\| = w + x'_i/2 - x_j/2 - 3/4$, and $\chi = (x'_i/2 + x_j/2 - 5/4, 0)$. From the similarity of triangles $(0,0)\chi c_l$ and $(0,-h)\chi' c_l$ we get that $\chi' = (x'_i + x_j - 5/2, 0)$. The condition that the guard g_l is coincident with χ' or to the left of χ' is equivalent to $-5/2 + x_l \leq x'_i + x_j - 5/2$, i.e., $x'_i + x_j \geq x_l$.

On the other hand, if $x'_i + x_j \geq x_l$ then the guard g_l is coincident with χ' or to the left of χ', and therefore g_l can see ω. Then the guards g'_i, g_j, g_l can together see the whole Γ. □

C.10.2 Fragment of the gadget for testing the inequality

We now present construction of a polygon $\mathcal{P}_{\text{ineq}}$ containing three guard segments r'_i, r_j, r_l and vertices c_i, c_j, c_l with coordinates as described above, where we set $w := 26$, $v := 10$, and $h := 10.5$, enforcing an inequality on the values corresponding to the guard segments. The main part of the polygon $\mathcal{P}_{\text{ineq}}$ is pictured in Figure 26. The three blue line segments correspond to the guard segments, and the three green dots to stationary guard positions. The stationary guard positions g_t, g_m, g_b have been chosen as follows. The point g_t is at the ray with origin at the right endpoint of r_j and containing c_j, g_b is at the ray with origin at the right endpoint of r_j and containing c_l, and g_m is at the ray with origin at the right endpoint of r'_i and containing c_i. For each guard segment and each stationary guard position we introduce pockets of the polygon, such that the points defining each guard segment and stationary guard position cannot be seen from any other guard segment or stationary guard position. Two edges in the left of the figure are only shown partially. They end to the left at vertices $c_1 := (-CN^2, CN^2)$ and $d_1 := (-CN^2, CN^2 + 1.5)$, respectively. These vertices are connected by an edge $c_1 d_1$ which closes the polygon. We can show the following result.

Lemma 29. *Consider the polygon $\mathcal{P}_{ineq}$ from Figure 26. A set of points $G \subset \mathcal{P}_{ineq}$ of cardinality at most 6 guards $\mathcal{P}_{ineq}$ if and only if*

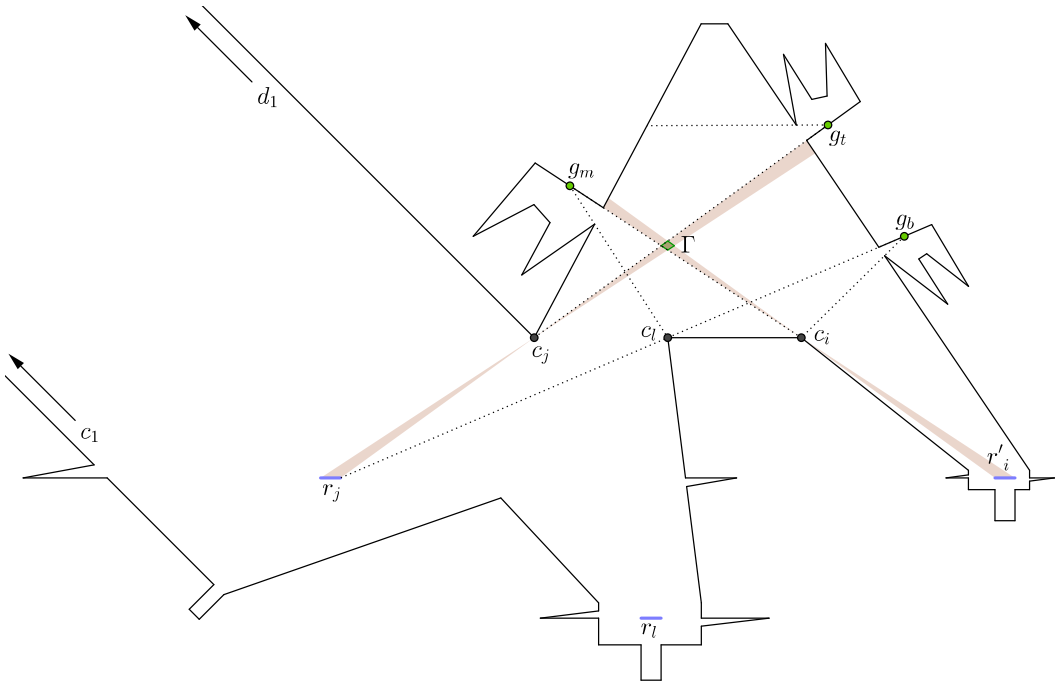


Figure 26: The main part of a polygon $\mathcal{P}_{\text{ineq}}$ for which all guard sets G of size 6 have the following form: (i) there is one guard at each stationary guard position (denoted by the green dots), (ii) there is one guard at each guard segment (blue line segments), and (iii) the values x'_i, x_j, x_l specified by the guards at the corresponding guard segments r'_i, r_j, r_l satisfy the inequality $x'_i + x_j \geq x_l$. Moreover, all sets G satisfying conditions (i)-(iii) are guard sets. The brown regions represent points of $\mathcal{P}_{\text{ineq}}$ included in some ray originating at r_j and containing c_j, or originating at r'_i and containing c_i. The quadrilateral Γ represents the intersection of these two regions.

- *there is exactly one guard placed at each guard segment r'_i, r_j, r_l and at each stationary guard position g_t, g_m, g_b, and*
- *the variables x'_i, x_j, x_l corresponding to the guard segments r'_i, r_j, r_l, respectively, satisfy the inequality $x'_i + x_j \geq x_l$.*

Proof. Assume first that the two conditions are satisfied. Observe that the stationary guard positions g_t, g_m, g_b have been chosen so that guards placed at them cannot see any point in the interior of Γ, but together with the guards g'_i, g_j, g_l placed at r'_i, r_j, r_l can see the whole area of $\mathcal{P}_{\text{ineq}} \setminus \Gamma$. (For this property to hold, the position of g'_i, g_j, g_l within the corresponding guard segments does not matter.) Since additionally the inequality $x'_i + x_j \geq x_l$ is satisfied, then Lemma 28 yields that the whole area of Γ is seen, and hence G guards $\mathcal{P}_{\text{ineq}}$.

Assume now that a set G of at most 6 guards sees all of $\mathcal{P}_{\text{ineq}}$. Using Lemmas 14 and 15 (where we set $A = \mathcal{P}_{\text{ineq}}$ and choose the points in M among the following: one point t_1 defining each stationary guard position, and one point q_2 defining each guard segment) we can show that the polygon requires at least 6 guards, and if there are 6 guards then there must be one guard at each stationary guard position, and at each guard segment. Then, as Γ is seen by the guards, by Lemma 28 we get that the inequality $x'_i + x_j \geq x_l$ holds. □

In the actual $\geq$-addition gadget we modify $\mathcal{P}_{\text{ineq}}$ in a way described later and scale it down by a factor of $\frac{1}{CN^2}$. Then we connect it to the main part of the polygon $\mathcal{P}$. The polygon $\mathcal{P}_{\text{ineq}}$ is attached at the right side of $\mathcal{P}$, and the connection between $\mathcal{P}_{\text{ineq}}$ and the main area of $\mathcal{P}$ is via a corridor. The point which corresponds to $(0, 0)$ in the polygon $\mathcal{P}_{\text{ineq}}$ is then at the position $m := c_1 + (1, -1)$ in $\mathcal{P}$, where c_1 is the bottom-right vertex of the corridor.

C.10.3 Copying guard segments to their final position

Let r'_i, r_j, r_l denote the guard segments of $\mathcal{P}_{\text{ineq}}$. We now show how to enforce dependency between appropriate guard segments from the base line of the polygon $\mathcal{P}$ and the guard segments r'_i, r_j, r_l.

We ensure that the guard segments r_j and r_l are copies of segments from the base line by connecting the (scaled) polygon $\mathcal{P}_{\text{ineq}}$ to the polygon $\mathcal{P}$ via a corridor. However, the segment r'_i cannot be copied in this way, as the edges of $\mathcal{P}_{\text{ineq}}$ are blocking visibility. Instead, we introduce an additional guard segment r_i within the gadget, and we copy appropriately chosen segments s_i, s_j, s_l from the base line into r_i, r_j, r_l. This part is explained in detail in Section C.10.5. Then, by introducing a copy-nook within the construction of $\mathcal{P}_{\text{ineq}}$, we ensure that r'_i is a weak copy of r_i. This is explained in detail in Section C.10.4. In Section C.10.6 we summarize the properties of the constructed gadget.

The gadget is scaled by a factor of $\frac{1}{CN^2}$ before it is attached to the corridor, and the points c_1, d_1 of the gadget are coincident with the points defining the right entrance of the corridor, which have the same names. After this operation, the middle point of the gadget (i.e., the point corresponding to $(0, 0)$ in the coordinate system of the gadget) satisfies the equality $m = c_1 + (1, -1)$, as stated in Section C.7.

For the picture of the complete gadget see Figure 27.

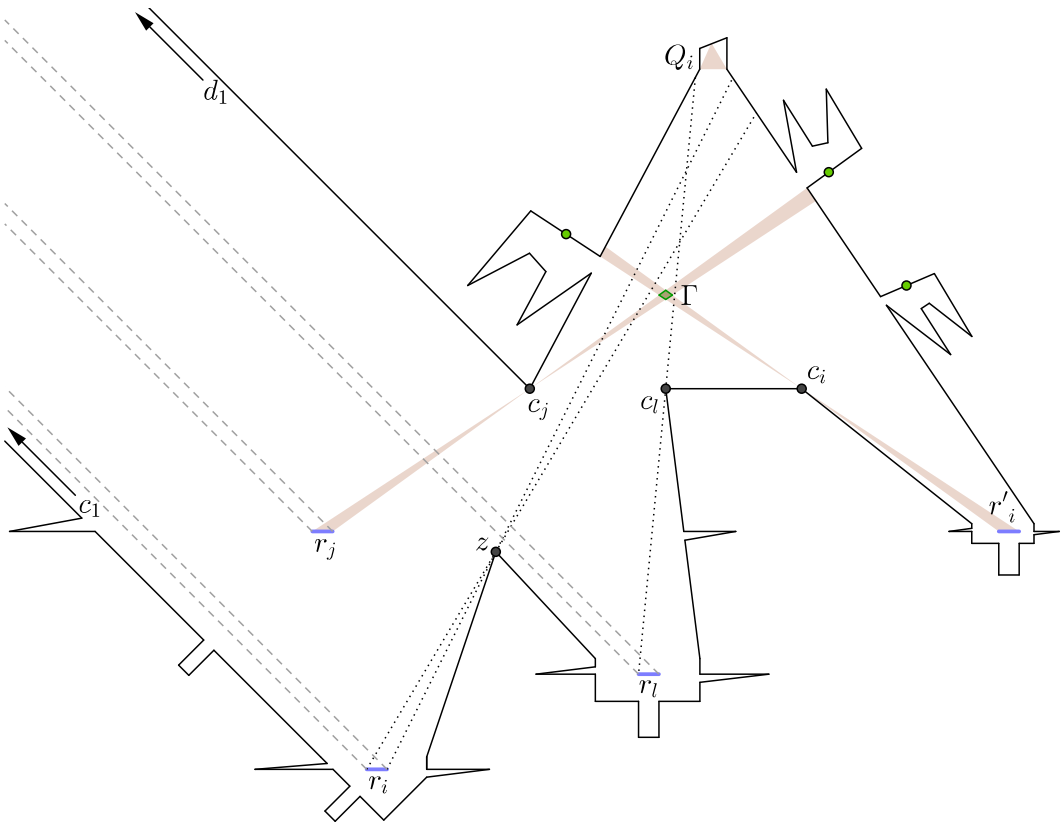


Figure 27: Detailed construction of the $\geq$-addition gadget. Note that the dotted lines show that no guard on r_i can see any point in Γ because of the corner z, a guard on r_i can always see both shadow corners of the copy-nook Q_i, and no point on r_l sees any point of Q_i because of the corner c_l. For each of the segments r_σ, $\sigma \in \{i, j, l\}$, the rays from points on r_σ through the corridor entrance $c_1 d_1$ are between the two grey dashed rays emitting from the endpoints of r_σ.

C.10.4 Introducing a new guard segment r_i

Consider the setting as described in Section C.10.2, and the polygon $\mathcal{P}_{\text{ineq}}$ from Figure 26. We explain how to modify $\mathcal{P}_{\text{ineq}}$ into a polygon $\mathcal{P}'_{\text{ineq}}$, which is a scaled version of our gadget. The main part of $\mathcal{P}'_{\text{ineq}}$ is shown in Figure 27, where again the edges to the left with endpoints at c_1 and d_1 are not fully shown.

The polygon $\mathcal{P}'_{\text{ineq}}$ is obtained from $\mathcal{P}_{\text{ineq}}$ in the following way. First, we add an additional guard segment r_i of length $3/2$ (i.e., the same as the length of other guard segments in the construction), with its left endpoint at the point $(-20.5, -17)$. This requires introducing a pocket corresponding to the added guard segment. We ensure that r'_i is a *weak copy* of r_i by creating a copy-nook Q_i the for pair of guard segments r_i, r'_i, which cannot be seen from any other guard segment or stationary guard position. The shadow corners of Q_i are $(2.5, 34)$ and $(4.5, 34)$. We have to ensure that after this modification, the gadget still enforces the desired inequality. In particular, we have to ensure that a guard placed at r_i cannot see any point in the interior of Γ. We can do that by introducing a new corner $z = (-12.5, -1.5)$ of the polygon that blocks r_i from seeing Γ. Note that z does not block r_i from seeing the segment c_1d_1 (as shown by the dashed grey lines in Figure 27).

Lemma 30. *A set of points $G \subset \mathcal{P}'_{ineq}$ of cardinality at most 7 guards $\mathcal{P}'_{ineq}$ if and only if*

- *there is exactly one guard placed at each guard segment r'_i, r_i, r_j, r_l and at each stationary guard position,*
- *the variables x_i, x'_i corresponding to the guard segments r_i, r'_i, respectively, satisfy the inequality $x_i \geq x'_i$, and*
- *the variables x'_i, x_j, x_l corresponding to the guard segments r'_i, r_j, r_l, respectively, satisfy the inequality $x'_i + x_j \geq x_l$.*

Proof. Assume that the polygon is guarded by a set G of at most 7 guards. Similarly as in Lemma 29 we can show that there must be exactly one guard at each guard segment and each stationary guard position. As the copy-nook Q_i can be seen only by guards placed at r_i and r'_i, it follows from Lemma 16 that the polygon is guarded by G only if the variables x_i, x'_i corresponding to r_i, r'_i, respectively, satisfy the inequality $x_i \geq x'_i$. As the quadrilateral Γ cannot be seen by a guard from r_i, we get from Lemma 29 that the variables x'_i, x_j, x_l corresponding to the guard segments r'_i, r_j, r_l, respectively, must satisfy the inequality $x'_i + x_j \geq x_l$.

Now assume that there is exactly one guard placed at each guard segment r'_i, r_i, r_j, r_l and at each stationary guard position, the variables x_i, x'_i satisfy the inequality $x_i \geq x'_i$, and the variables x'_i, x_j, x_l satisfy the inequality $x'_i + x_j \geq x_l$. Then the whole area of Γ is seen by the guards, as is the whole area of the nook Q_i. The remaining area is also seen by the guards, which we can show in the same way as in Lemma 29. □

C.10.5 Copying three guard segments via a corridor

We are given a formula from Φ of the form $x_i + x_j = x_l$, where $i, j, l \in \{0, \ldots, n-1\}$ and want to construct a gadget imposing an inequality $x_i + x_j \geq x_l$. We need to show that the values of the required variables can be copied into the guard segments r_i, r_j, r_l in the gadget described above. First, we will explain how to choose the segments from the base line to be copied into the gadget. Then, we will show that the $\geq$-addition gadget satisfies properties required by Lemma 21, which will ensure that the gadget can be connected to the main area by a corridor.

In order to apply a corridor construction as described in Section C.9 to copy three guard segments from the base line into the gadget, we require the segments in order from left to right on the base line to represent the variables x_i, x_j, x_l. Recall that there are n variables $x_0, \ldots, x_{n-1}$ in the formula Φ, but that for any $\sigma \in \{0, \ldots, n-1\}$, we use $x_{\sigma+n}, x_{\sigma+2n}$ and $x_{\sigma+3n}$ as synonyms

for x_σ. Therefore, the inequality $x_i + x_j \geq x_l$ is equivalent to $x_i + x_{n+j} \geq x_{2n+l}$. The guard segments on the base line are $s_0, \ldots, s_{4n-1}$, where each s_σ represents the variable x_σ. The segments $s_0, \ldots, s_{3n-1}$ are right-oriented whereas $s_{3n}, \ldots, s_{4n-1}$ are left-oriented. (In Section C.12 we explain how to obtain these dependencies between the guard segments.) Therefore, with slight abuse of notation, we redefine $j := j + n$ and $l := l + 2n$ so that $i < j < l < 3n$. Then, the guard segments s_i, s_j, s_l satisfy our requirements.

We now need to show that our gadget construction satisfies the conditions of Lemma 21. Recall that our gadget is the polygon $\mathcal{P}'_{\text{ineq}}$ scaled by a factor of $\frac{1}{CN^2}$. First, we will prove an auxiliary lemma.

Lemma 31. *Let G be any gadget to be attached at the right side of the polygon $\mathcal{P}$ such that the guard segments r_i, r_j, r_l have a length of $\frac{3/2}{CN^2}$ and are contained in the box $m+[-\Delta,\Delta]\times[-\Delta,\Delta]$, where $\Delta := \frac{50}{CN^2}$. Suppose furthermore that the left endpoint of r_j is on the line through $a'_i+(\delta,\delta)$ parallel to the vector $(-1,1)$ and the left endpoint of r_l is on the line through $a'_i+(2\delta,2\delta)$ parallel to the same vector, where a'_i is the left endpoint of r_i. Then properties 1 and 2 of Lemma 21 both hold for G.*

Proof. Assume in this proof without loss of generality that $c_1 = (0,0)$. Then $m = (1,-1)$. Define $(\omega_{xi}, \omega_{yi}) := CN^2 \cdot (b'_i - m)$, $(\omega_{xj}, \omega_{yj}) := CN^2 \cdot (b'_j - m)$, and $(\omega_{xl}, \omega_{yl}) := CN^2 \cdot (b'_l - m)$. It follows from the conditions in the lemma that each of these values is in $[-50, 50]$.

Define o'_σ for each $\sigma \in \{i,j,l\}$ to be the intersection point of the ray $\overrightarrow{b'_\sigma c_1}$ with ℓ_c. Recall that the point o' is defined as o'_l. We first verify that Property 1 holds, that is, each point o'_σ is in the rectangle V. We thus need to ensure that $y(o'_\sigma) \in [1 - 38N\rho, 1 + 38N\rho]$. The horizontal distance between ℓ_c and c_1 is 1. Therefore, the vertical distance between c_0c_1 and the point o'_σ is $y(o'_\sigma) = \frac{-y(b'_\sigma)}{x(b'_\sigma)} = \frac{CN^2-\omega_{y\sigma}}{CN^2+\omega_{x\sigma}}$. We have $y(o'_\sigma) \geq \frac{CN^2-50}{CN^2+50}$. Note that the lower edge of the square V has y-coordinate $1 - 38N\rho = 1 - \frac{57}{CN}$. Now, the inequality $\frac{CN^2-50}{CN^2+50} \geq 1 - \frac{57}{CN}$ is equivalent to $57CN^3 + 2850N \geq 100CN^2$, which is true for all $N \geq 2$.

Similarly, we have that $y(o'_\sigma) \leq \frac{CN^2+50}{CN^2-50}$. The upper edge of V has y-coordinate $1 + 38N\rho = 1 + \frac{57}{CN}$. The inequality $\frac{CN^2+50}{CN^2-50} \leq 1 + \frac{57}{CN}$ is equivalent to $100CN^2 \leq 57CN^3 - 2850N$, which is also true for all $N \geq 2$. Hence, Property 1 holds.

Fix any $\sigma \in \{i,j,l\}$. Recall that the vertical distance between the centers of two consecutive rays from $R^0_\sigma, R^1_\sigma, R^2_\sigma, R^3_\sigma$ is ρ. We now show that the following four properties imply that Property 2 holds. Afterwards, we will prove that the four properties are satisfied.

- **a** The vertical distance from o'_σ to the center of the slab R^0_σ is at most $\varepsilon/4$,
- **b** the distance d^0_σ between the intersection points of $\overrightarrow{b'_\sigma c_1}$ and $\overrightarrow{b'_\sigma d_1}$ with ℓ_c is in the interval $[2\rho - \frac{\varepsilon}{4}, 2\rho + \frac{\varepsilon}{4}]$,
- **c** for any symbol $\mu \in \{c,d\}$ the distance d^μ_σ between the intersection points of $\overrightarrow{a'_\sigma \mu_1}$ and $\overrightarrow{b'_\sigma \mu_1}$ with ℓ_c is in $\left[\rho - \frac{\varepsilon}{4}, \rho + \frac{\varepsilon}{4}\right]$, and
- **d** the absolute value of the slope of any ray with origin at an endpoint of r_σ and passing through a point in c_1d_1 is in $\left[1 - \frac{1}{38N\rho} \cdot \frac{\varepsilon}{4}, 1 + \frac{1}{38N\rho} \cdot \frac{\varepsilon}{4}\right]$.

The first three properties yield that all rays $\overrightarrow{b'_\sigma c_1}, \overrightarrow{a'_\sigma c_1}, \overrightarrow{b'_\sigma d_1}$, and $\overrightarrow{a'_\sigma d_1}$ intersect the line ℓ_c within their corresponding slabs at the vertical distance of at most $\frac{3\varepsilon}{4}$ from the center of the slab. The last property yields that the rays are contained in the corresponding slabs throughout the square V.

We now prove Property **a**. For $\sigma = l$, the distance is 0 by definition. We thus have to bound the distances $\|o'o'_i\|$ and $\|o'o'_j\|$.

Note that the conditions in the lemma gives that $y(b'_j) = y(b'_l) + x(b'_l) - x(b'_j) - \delta$ and $y(b'_i) = y(b'_l) + x(b'_l) - x(b'_i) - 2\delta$. We have

$$\begin{aligned}
\|o'o'_j\| = y(o'_j) - y(o') &= \frac{-y(b'_j)}{x(b'_j)} - \frac{-y(b'_l)}{x(b'_l)} = \frac{-y(b'_l) - x(b'_l) + x(b'_j) + \delta}{x(b'_j)} + \frac{y(b'_l)}{x(b'_l)} \\
&= \frac{-y(b'_l) - x(b'_l) + \delta}{x(b'_j)} + \frac{y(b'_l) + x(b'_l)}{x(b'_l)} \\
&= \frac{2CN^2 - \omega_{yl} - \omega_{xl} + CN^2\delta}{CN^2 + \omega_{xj}} + \frac{-2CN^2 + \omega_{yl} + \omega_{xl}}{CN^2 + \omega_{xl}} \\
&\in \left[\frac{CN^2\delta}{CN^2 + 50}, \frac{CN^2\delta}{CN^2 - 50}\right] \subset \left[\frac{CN^2\delta}{CN^2 + 50N^2}, \frac{CN^2\delta}{CN^2 - 50N^2}\right] \\
&= \left[\frac{4000}{4001} \cdot \delta, \frac{4000}{3999} \cdot \delta\right] \subset [\delta - \varepsilon/4, \delta + \varepsilon/4]\,.
\end{aligned}$$

A similar argument gives $\|o'o'_i\| \in [2\delta - \varepsilon/4, 2\delta + \varepsilon/4]$.

We will prove Property **b** as follows. We have $|x(b'_\sigma) - x(m)| \le \Delta \le \frac{1}{1500}$ (as $C > 75000$), and

$$\begin{aligned}
d^0_\sigma &= \|c_1 d_1\| \cdot \frac{2 + x(b'_\sigma) - x(m)}{1 + x(b'_\sigma) - x(m)} \\
&\in \left[\rho \cdot \frac{2 - \frac{1}{1500}}{1 + \frac{1}{1500}}, \rho \cdot \frac{2 + \frac{1}{1500}}{1 - \frac{1}{1500}}\right] = \left[\rho \cdot \left(2 - \frac{2}{1501}\right), \rho \cdot \left(2 + \frac{3}{1499}\right)\right] \subset \left[2\rho - \frac{\varepsilon}{4}, 2\rho + \frac{\varepsilon}{4}\right].
\end{aligned}$$

For Property **c**, denote H as the distance between the point a'_σ and its vertical projection on the ray $\overrightarrow{b'_\sigma \mu_1}$. We have $\frac{d^\mu_\sigma}{H} = \frac{1}{1 + x(a'_\sigma) - x(m)} = \frac{1}{1 + x(b'_\sigma) - x(m) - \rho}$, $\frac{H}{\rho} = \frac{1 + y(m) - y(b'_\sigma) + \|\mu_1 c_1\|}{1 + x(b'_\sigma) - x(m)}$ and $\rho \le \frac{1}{50000}$ (as $C \ge 75000$), and therefore

$$\begin{aligned}
d^\mu_\sigma &= \rho \cdot \frac{1 + y(m) - y(b'_\sigma) + \|\mu_1 c_1\|}{1 + x(b'_\sigma) - x(m)} \cdot \frac{1}{1 + x(b'_\sigma) - x(m) - \rho} \\
&\in \left[\rho \cdot \frac{1 - \frac{1}{1500}}{1 + \frac{1}{1500}} \cdot \frac{1}{1 + \frac{1}{1500}}, \rho \cdot \frac{1 + \frac{1}{1500} + \frac{1}{50000}}{1 - \frac{1}{1500}} \cdot \frac{1}{1 - \frac{1}{1500} - \frac{1}{50000}}\right] \\
&= \left[\rho \cdot \left(1 - \frac{4501}{2253001}\right), \rho \cdot \left(1 + \frac{458897}{224695603}\right)\right] \\
&\subset \left[\delta - \frac{\varepsilon}{4}, \delta + \frac{\varepsilon}{4}\right].
\end{aligned}$$

For Property **d**, we have to bound the slope of the rays. Note that $\frac{1}{38N\rho} \cdot \frac{\varepsilon}{4} = \frac{1}{1824N}$. Since $C \ge 50 \cdot 1824 = 91200$, we get that the slope is at least

$$\frac{y(c_1) - y(b'_\sigma)}{x(b'_\sigma) - x(c_1)} = \frac{1 + y(m) - y(b'_\sigma)}{1 + x(b'_\sigma) - x(m)} \ge \frac{1 - \Delta}{1 + \Delta} = 1 - \frac{50}{CN^2 + 50} \ge 1 - \frac{50}{CN} \ge 1 - \frac{1}{1824N}.$$

On the other hand, since $C \ge 101.5 \cdot 1824 + 50 = 185186$, we get that the slope is at most

$$\frac{y(d_1) - y(b'_\sigma)}{x(b'_\sigma) - x(d_1)} \le \frac{1 + \Delta + \rho}{1 - \Delta} = 1 + \frac{101.5}{CN^2 - 50} \le 1 + \frac{101.5}{(C - 50)N} \le 1 + \frac{1}{1824N}.$$

Since all four properties hold, so does Property 2. □

Proof of Lemma 21 for the ≥-addition gadget. Note that in the ≥-addition gadget, the segments r_i, r_j, r_l have lengths of $\frac{3/2}{CN^2}$ and their right endpoints are placed at positions $m + (\frac{-20.5}{CN^2}, \frac{-17.5}{CN^2}), m + (\frac{-24.5}{CN^2}, 0), m + (\frac{-0.5}{CN^2}, \frac{-10.5}{CN^2})$, respectively, where $m := c_1 + (1, -1)$. As the conditions of Lemma 31 are satisfied, Properties 1 and 2 hold.

Property 3 also holds, as the edge $c_j d_1$ blocks all points at stationary guard positions and at the guard segment r'_i from seeing $c_1 d_1$. □

C.10.6 Summary

Lemma 32. *Consider the addition gadget together with the corresponding corridor representing an inequality $x_i + x_j \geq x_l$, as described below. The following properties hold.*

- *The gadget and the corridor fit into a rectangular box of height* 3*.*
- *For any guard set of $\mathcal{P}$, at least* 10 *guards have to be placed in the corridor and the gadget.*
- *Assume that in the main area $\mathcal{P}_M$, there is exactly one guard at each guard segment, and there are no guards outside of the guard segments. Then* 10 *guards can be placed in the corridor and the gadget so that the whole corridor and gadget is seen if and only if the values x_i, x_j, x_l specified by the guards at the guard segments s_i, s_j, s_l satisfy the inequality $x_i + x_j \geq x_l$.*

Proof. Recall that by Lemma 24, the distance from c_0c_1 to the topmost point in the corridor is at most 1.4. The main part of the gadget is centered around the point $c_0 + (1, -1)$, and as it is of size $\Theta(\frac{1}{CN^2})$, the vertical space of at most 1.1 below the line segment c_0c_1 is enough to fit the gadget.

From Lemma 24, there are at least 3 guards placed within the corridor. From Lemma 30, there are at least 7 guards placed within the gadget. That gives us at least 10 guards needed.

Assume that there are exactly 10 guards within the corridor and gadget and that the corridor is completely seen. Then, from Lemma 24 and 30, there is exactly one guard at each guard segment and each stationary guard position. Then, by Lemma 24, the values x_i, x_j, x_l specified by s_i, s_j, s_l, and the values specified by r_i, r_j, r_l, are the same. By Lemma 30, the values x_i, x'_i corresponding to r'_i, r_i satisfy $x_i \geq x'_i$, and we also have $x'_i + x_j \geq x_l$. That enforces inequality $x_i + x_j \geq x_l$.

On the other hand, assume that $x_i + x_j \geq x_l$. We first place a guard at every of the 6 stationary guard positions in the corridor and gadget. By Lemmas 24 and 30, if we set guards at the 4 guard segments so that the values specified by guards at r_i, r_j, r_l are x_i, x_j, x_l, and the value x'_i specified by the guard at r'_i is the same as x_i, then the whole area of the gadget is guarded. □

C.11 The $\leq$-addition gadget

In this section we present construction of a gadget representing an inequality $x_i + x_j \leq x_l$, where $i, j, l \in \{0, \ldots, n-1\}$. The idea of the construction of this gadget is analogous as for the $\geq$-addition gadget presented in Section C.10, and the basic principle is presented in Section C.11.1. The principle underlying the $\geq$-addition gadget, as explained in Section C.10.1, required the polygon to have edges blocking the visibility between the segment r'_i (placed at the right side) and the segments r_j, r_l (placed at the left side). We managed to get around that by making r'_i a weak copy of a segment r_i, which in turn was a copy of a segment s_i on the base line. In contrast to this, the principle underlying the $\leq$-addition gadget presented here requires the polygon to have an edge separating r'_i placed at the left side from the segments r_j, r_l at the right side. If we were to build a gadget to be placed at the right side of $\mathcal{P}$, we would have to copy the variables corresponding to both of the segments r_j, r_l weakly, and the gadget would not enforce the desired inequality. To avoid this problem, we will place the gadget at the left side of $\mathcal{P}$. Then, we introduce an additional guard segment r_i, and we make r'_i a weak copy of r_i using a copy-nook Q_i. As r'_i is to the left of r_i, the copy-nook Q_i enforces the inequality $x'_i \geq x_i$, where r'_i, r_i represent x'_i, x_i, respectively. The result is that the gadget enforces the desired inequality. The relative placement of the segments r_i, r_j, r_l, r'_i (in particular, the value of w from Section C.10.2) has to be slightly different than in the construction of $\geq$-addition gadget, as it does not seem to be possible to make the gadgets completely symmetrical.

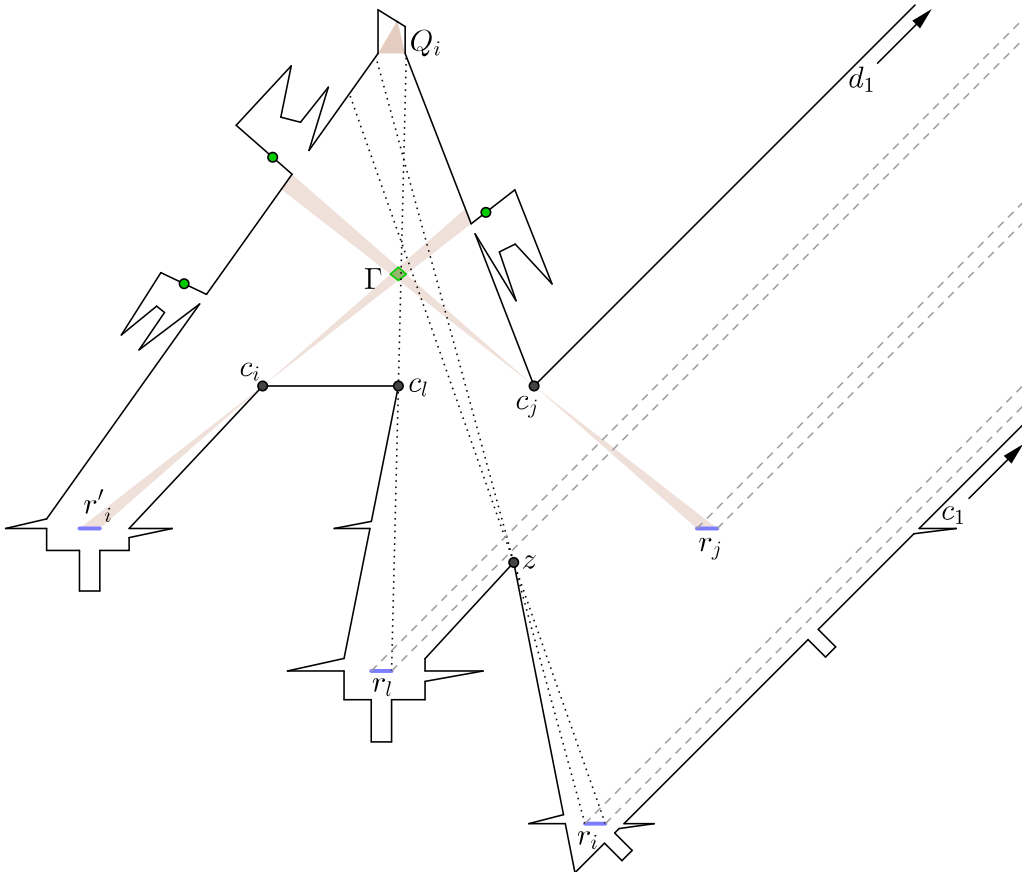


Figure 28: Detailed construction of the $\leq$-addition gadget. As previously, the dotted line shows that no guard on r_i can see any point in Γ because of the corner z, a guard on r_i can always see both shadow corners of the copy-nook Q_i, and no point on r_l sees any point of Q_i because of the corner c_l. For each of the segments r_σ, $\sigma \in \{i, j, l\}$, the rays from points on r_σ through the corridor entrance $c_1 d_1$ are between the two grey dashed rays emitting from the endpoints of r_σ.

C.11.1 Idea behind the gadget construction

The idea behind a gadget imposing an inequality $x'_i + x_j \leq x_l$ is similar as for the $\geq$-addition gadget described above. As before, consider rational values $w, v, h > 0$, where $w > v + 3/2$, and let r'_i, r_j, r_l be right-oriented guard segments of length $3/2$ such that r'_i has its left endpoint at the point $(-w, 0)$, r_j has its right endpoint at $(w, 0)$, and r_l has its left endpoint at $(-2, -h)$. Let $g'_i := (-w - 1/2 + x_i, 0)$, $g_j := (w - 2 + x_j, 0)$, and $g_l := (-5/2 + x_l, -h)$ be three guards on r'_i, r_j, r_l, respectively, representing the values $x'_i, x_j, x_l \in [1/2, 2]$.

Suppose that there are vertices $c_i := (-v, h), c_j := (v, h), c_l := (0, h)$ of $\mathcal{P}$. As before, let Γ be a collection of points ω such that the ray $\overrightarrow{\omega c_i}$ intersects r'_i, and the ray $\overrightarrow{\omega c_j}$ intersects r_j. Then Γ is a quadrilateral, bounded by the following rays: the rays with origin at the endpoints of r'_i and containing c_i, and the rays with origin at the endpoints of r_j and containing c_j. Suppose that

- for every point g'_i on r'_i and ω in Γ, the points ω and g'_i can see each other if and only if ω is on or to the left of the line $\overleftrightarrow{g'_i c_i}$,
- for every point g_j on r_j and ω in Γ, the points ω and g_j can see each other if and only if ω is on or to the left of the line $\overleftrightarrow{g_j c_j}$,
- for every point g_l on r_l and ω in Γ, the points ω and g_l can see each other if and only if ω is on or to the right of the line $\overleftrightarrow{g_l c_l}$.

Then, we can show the following result in an analogous way as we proved Lemma 28.

Lemma 33. *The guards g'_i, g_j, g_l can see together the whole quadrilateral Γ if and only if $x'_i + x_j \leq x_l$.*

C.11.2 A specification of the gadget

We will present the construction of a gadget with four guard segments r_i, r_j, r_l, r'_i, where the segments r'_i, r_j, r_l correspond to the segments in the idea described in Section C.11.1, where this time we set $w := 23.5$, $v := 10$, and $h := 10.5$. The gadget is shown in Figure 28 and should be attached to the left side of the main are $\mathcal{P}_M$ using a left corridor as described in Section C.9.4.

As for the case of the $\geq$-addition gadget, there are three stationary guards which do not see any point within Γ, but which enforce that the whole area except of Γ is seen whenever a guard is placed at each of the guard segments r_i, r_j, r_l, r'_i. The guard segment r_i has length $3/2$ (as do r'_i, r_j, r_l) and is placed with its left endpoint at the point $(13.75, -21.75)$. We will ensure that r'_i is a *weak copy* of r_i by creating a copy-nook Q_i the for pair of guard segments r_i, r'_i. The nook Q_i has shadow corners $(-1.5, 35)$ and $(0.5, 35)$. To ensure that a guard placed at r_i cannot see any point in the interior of Γ, we introduce a new corner $z := (8.5, -2.5)$ of the polygon that blocks r_i from seeing Γ. Two edges to the right in the figure are not fully shown. They end at vertices $c_1 := (CN^2, CN^2)$ and $d_1 := (CN^2, CN^2 + 1.5)$.

In order to attach the gadget to the main are $\mathcal{P}_M$, we scale down the construction described here by the factor $\frac{1}{CN^2}$ and translate it so that the point which corresponds to $(0, 0)$ in the gadget will be placed at position $m := c_1 + (-1, -1)$. (Recall that the left entrance to the corridor, at which we attach the gadget, is the segment $c_1 d_1$.)

The following lemma is proved in the same way as Lemma 30.

Lemma 34. *Let $\mathcal{P}'_{\text{rev-ineq}}$ be the polygon obtained from the gadget described above by closing it by adding the edge $c_1 d_1$. A set of points $G \subset \mathcal{P}'_{\text{rev-ineq}}$ of cardinality at most 7 guards $\mathcal{P}'_{\text{rev-ineq}}$ if and only if*

- *there is exactly one guard placed at each guard segment r'_i, r_i, r_j, r_l and at each stationary guard position,*
- *the variables x_i, x'_i corresponding to the guard segments r_i, r'_i, respectively, satisfy the inequality $x_i \leq x'_i$, and*
- *the variables x'_i, x_j, x_l corresponding to the guard segments r'_i, r_j, r_l, respectively, satisfy the inequality $x'_i + x_j \leq x_l$.*

C.11.3 Copying three guard segments via a corridor

Here we proceed as in section C.10.5. We need to show is that the variables x_i, x_j, x_l can be copied into guard segments r_i, r_j, r_l from three guard segments on the base line. Due to the gadget construction, we now require the segment corresponding to the variable x_l to be the leftmost one, and all guard segments have to be right-oriented. With slight abuse of notation, we redefine $i := i + n$ and $j := j + 2n$. Then, the segments s_l, s_i, s_j satisfy our requirements.

As before, to prove that the corridor construction enforces the required dependency between the guards on the base line and guards within the gadget, i.e., for Lemma 24 to work, we need to show that our gadget construction satisfies the conditions of Lemma 26. In particular, we use the following symmetric version of Lemma 31 for gadgets attached to the left side of $\mathcal{P}$.

Lemma 35. *Let G be any gadget to be attached at the left side of the polygon $\mathcal{P}$ such that the guard segments r_l, r_j, r_i have a length of $\frac{3/2}{CN^2}$ and are contained in the box $m + [-\Delta, \Delta] \times [-\Delta, \Delta]$, where $\Delta := \frac{50}{CN^2}$. Suppose furthermore that the left endpoint of r_j is on the line*

through $a'_i + (-\delta, \delta)$ parallel to the vector $(1,1)$ and the left endpoint of r_l is on the line through $a'_i + (-2\delta, 2\delta)$ parallel to the same vector, where a'_i is the left endpoint of r_i. Then properties 1 and 2 of Lemma 26 both hold for G.

Proof of Lemma 26 for the $\leq$-addition gadget. Note that in the $\leq$-addition gadget, the segments r_i, r_j, r_l have lengths of $\frac{3/2}{CN^2}$ and their left endpoints are placed at positions $m + (\frac{13.75}{CN^2}, \frac{-21.75}{CN^2}), m + (\frac{22}{CN^2}, 0), m + (\frac{-2}{CN^2}, \frac{-10.5}{CN^2})$, respectively, where $m := c_1 + (-1,-1)$. As the conditions of Lemma 35 are satisfied, Properties 1 and 2 hold.

Property 3 also holds, as the edge $c_j d_1$ blocks all points at stationary guard positions and at the guard segment r'_i from seeing $c_1 d_1$. □

C.11.4 Summary

In the same way as in Lemma 32, we get the following result.

Lemma 36. *Consider the $\leq$-addition gadget together with the corridor, corresponding to the inequality $x_i + x_j \leq x_l$. The following properties hold.*

- *The gadget and the corridor fit into a rectangular box of height 3.*
- *For any guard set of $\mathcal{P}$, at least 10 guards have to be placed in the corridor and the gadget.*
- *Assume that in the main area $\mathcal{P}_M$, there is exactly one guard at each guard segment, and there are no guards outside of the guard segments. Then 10 guards can be placed in the corridor and the gadget so that the whole corridor and gadget is seen if and only if the values x_i, x_j, x_l specified by the guards at the guard segments s_i, s_j, s_l satisfy the inequality $x_i + x_j \leq x_l$.*

C.12 The $\geq$- and $\leq$-orientation gadgets

In this section we explain how to enforce consistency between the guard segments on the base line which represent the same variable x_i, for $i \in \{0, \ldots, n-1\}$. Recall that there are four guard segments $s_i, s_{n+i}, s_{2n+i}, s_{3n+i}$ representing the variable x_i, and that the first three ones are right-oriented, and the last one is left-oriented.

We will present a gadget enforcing that two guard segments corresponding to the same variable x_i and oriented in different directions specify the variable consistently. We will then use this gadget for the following pairs of guard segments: (s_i, s_{3n+i}), (s_{n+i}, s_{3n+i}), and (s_{2n+i}, s_{3n+i}).

Consider two guard segments s_i, s_j on the base line, where s_i is right-oriented and s_j is left-oriented, and assume that there is one guard placed on each of these segments. Let x_i and x_j be the values represented by s_i and s_j, respectively. Let x_j^r be the value that would be specified by s_j if s_j was right-oriented instead of left-oriented. We have $x_j + x_j^r = 2.5$. Therefore s_i and s_j specify the same value if and only if $x_i + x_j^r = 2.5$.

Performing a simple modification of the $\geq$- and $\leq$-addition gadgets, we obtain the $\geq$- and $\leq$-orientation gadgets, which together enforce the equality $x_i + x_j^r = 2.5$. See Figure 29 for a detailed picture of the main part of the $\geq$-orientation gadget, which enforces that $x_i + x_j^r \geq 2.5$, or, equivalently, $x_i \geq x_j$. In the $\geq$-addition gadget, we copy three values from the base line into the gadget. Here, we copy only the value of the two segments s_i, s_j. Instead of the guard segment r_l inside the gadget, we create a stationary guard position p at the line containing r_l at distance $\frac{1}{CN^2}$ to the right of the right endpoint of r_l. Then p corresponds to the value of $5/2$ on r_l (ignoring that p lies outside r_l).

The $\leq$-orientation gadget, which corresponds to the inequality $x_i + x_j^r \leq 5/2$, is obtained by an analogous modification of the $\leq$-addition gadget. Note that in both of these orientation gadgets, we create 4 stationary guard positions in the corridor instead of 6 for the addition

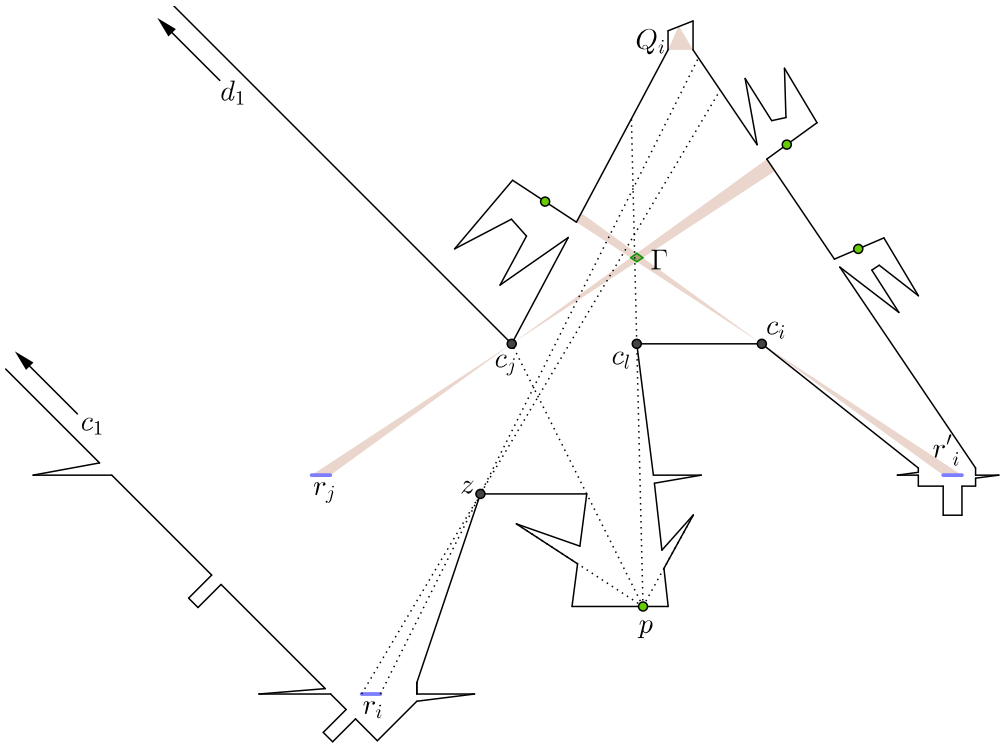


Figure 29: Detailed construction of the $\geq$-orientation gadget for $x_i^r + x_j^r \geq 2.5$, which is a modified version of the $\geq$-addition gadget.

gadgets, and the gadget itself contains 4 stationary guards and 3 guard segments, instead of 3 and 4 in the addition gadgets, respectively.

We summarize the properties of the orientation gadgets by the following Lemma, which can be proven in a way analogous to Lemmas 32 and 36.

Lemma 37. *Consider the $\geq$-orientation gadget (resp. $\leq$-orientation gadget) together with the corresponding corridor for making r_i, r_j copies of guard segments s_i, s_j on the base line, where s_i is right-oriented and s_j is left-oriented. The following properties hold.*

- *The gadget and the corridor fit into a rectangular box of height 3.*
- *For any guard set of $\mathcal{P}$, at least 9 guards have to be placed in the corridor and the gadget.*
- *Assume that in the main area $\mathcal{P}_M$, there is exactly one guard at each guard segment, and there are no guards outside of the guard segments. Then 9 guards can be placed in the corridor and the gadget so that the whole corridor and gadget is seen if and only if the values x_i, x_j specified by the guards at the guard segments s_i, s_j satisfy the inequality $x_i \geq x_j$ (resp. $x_i \leq x_j$).*

C.13 The inversion gadget

In this section we present the construction of the inversion gadget which represent an inequality $x_i \cdot x_j = 1$, where $i, j \in \{0, \ldots, n-1\}$. We made use of Maple [26] for the construction and verification of this gadget.

C.13.1 Idea behind the gadget construction

We first describe the principle underlying the inversion gadget. See Figure 30. Let $a'_i := (1/2, 0)$, $b'_i := (2, 0)$, $a'_j := (13.9, 0.1)$, $b'_j := (15.4, 0.1)$, and suppose that $r_i := a'_i b'_i$ is a right-oriented

guard segment representing the variable x_i and $r_j := a'_j b'_j$ is a left-oriented guard segment representing the variable x_j.

Let $\xi_0 := (7, \frac{541}{184}) \approx (7, 2.94)$ and $\xi_1 := (9, \frac{259139}{112792}) \approx (9, 2.30)$ and suppose that ξ_0, ξ_1 are shadow corners of an umbra Q_u with corners $\xi_0\xi_1 f_1 f_0$ of r_i, r_j. Then $f_0 := (\frac{499811}{70923}, \frac{38731813}{13049832}) \approx (7.05, 2.97)$ and $f_1 := (\frac{112379}{15432}, \frac{4355591}{1419744}) \approx (7.28, 3.07)$.

Lemma 38. *If guards p_i, p_j on r_i, r_j, respectively, see $f_0 f_1$ together, then $x_i x_j \leq 1$.*

Proof. Let π_i and π_j be the associated projections from r_i and r_j to $f_0 f_1$, respectively. Note that since p_i represents the variable x_i, we must have $p_i := (x_i, 0)$. Let

$$e := \pi_i(p_i) = \left(\frac{258288\, x_i - 16765}{36994\, x_i - 3065}, \frac{20013754\, x_i - 1295695}{6806896\, x_i - 563960} \right).$$

Now, $\pi_j^{-1}(e) = (15.9 - 1/x_i, 1/10)$, which represents the value $15.9 - (15.9 - 1/x_i) = 1/x_i$ on r_j. In order to see $f_0 f_1$ together with p_i, the guard p_j has to stand on $\pi_j^{-1}(e)$ or to the right. This corresponds to x_j being at most $1/x_i$. In other words, if a guard p_j on r_j sees $f_0 f_1$ together with p_i, then $x_i x_j \leq 1$. □

We now construct an umbra that impose the guards to satisfy the opposite inequality $x_i x_j \geq 1$: Let $\xi_2 := (7, \frac{8865}{752}) \approx (7, 11.79)$ and $\xi_3 := (9, \frac{4214815}{460976}) \approx (9, 9.14)$ and suppose that ξ_2, ξ_3 are shadow corners of a nook Q_n with corners $\xi_2\xi_3 f_3 f_2$ of r_i, r_j. Then $f_2 := (\frac{182083}{25835}, \frac{231222249}{19427920}) \approx (7.05, 11.90)$ and $f_3 := (\frac{205139}{28156}, \frac{130288905}{10586656}) \approx (7.29, 12.31)$.

Lemma 39. *If guards p_i, p_j on r_i, r_j, respectively, see $f_2 f_3$ together, then $x_i x_j \geq 1$.*

Proof. Let $\hat{\pi}_0$ and $\hat{\pi}_1$ be the associated projections from r_i and r_j to $f_2 f_3$, respectively. Let

$$\hat{e} := \hat{\pi}_0(p_0) = \left(\frac{470184\, x_i - 29953}{67346\, x_i - 5517}, \frac{597022290\, x_i - 37933335}{50644192\, x_i - 4148784} \right).$$

Now, $\hat{\pi}_j^{-1}(\hat{e}) = (15.9 - 1/x_i, 1/10)$, which represents the value $15.9 - (15.9 - 1/x_i) = 1/x_i$ on r_j. In order to see $f_2 f_3$ together with p_i, the guard p_j has to stand on $\hat{\pi}_1^{-1}(\hat{e})$ or to the left. This corresponds to x_j being at least $1/x_i$ so that $x_i x_j \geq 1$. □

We thus have the following lemma:

Lemma 40. *If guards p_i and p_j placed at guard segments r_i and r_i, respectively, see both critical segments $f_0 f_1$ and $f_2 f_3$, then the corresponding values specified by p_i and p_j satisfy $x_i x_j = 1$.*

C.13.2 The construction of the gadget

We now explain how to make the complete gadget for the equation $x_i x_j = 1$, as shown in Figure 30. We make the wall in the gadget so that it creates the umbra Q_u and the nook Q_n as described before. We also create a stationary guard position at the green point in the figure which sees the umbra Q_u, but nothing above the line containing the critical segment of Q_u. Two edges at the left side of the gadget are not fully shown. They end at vertices $c_1 := (-CN^2, CN^2)$ and $d_1 := (-CN^2, CN^2 + 1.5)$, respectively.

The gadget contains two guard segments r_i and r_j representing x_i and x_j, respectively, and it is required that r_i is right-oriented and r_j is left-oriented. Therefore, with slight abuse of notation, we redefine $j := j + 3n$, so that s_i, s_j are guard segments on the base line representing x_i, x_j, respectively, where s_i is right-oriented and s_j is left-oriented. We then use a corridor as described in Section C.9 to make r_i, r_j copies of s_i, s_j, respectively. Recall that the endpoints of the right entrance of the corridor are denoted c_1, d_1. In order to attach the gadget to the

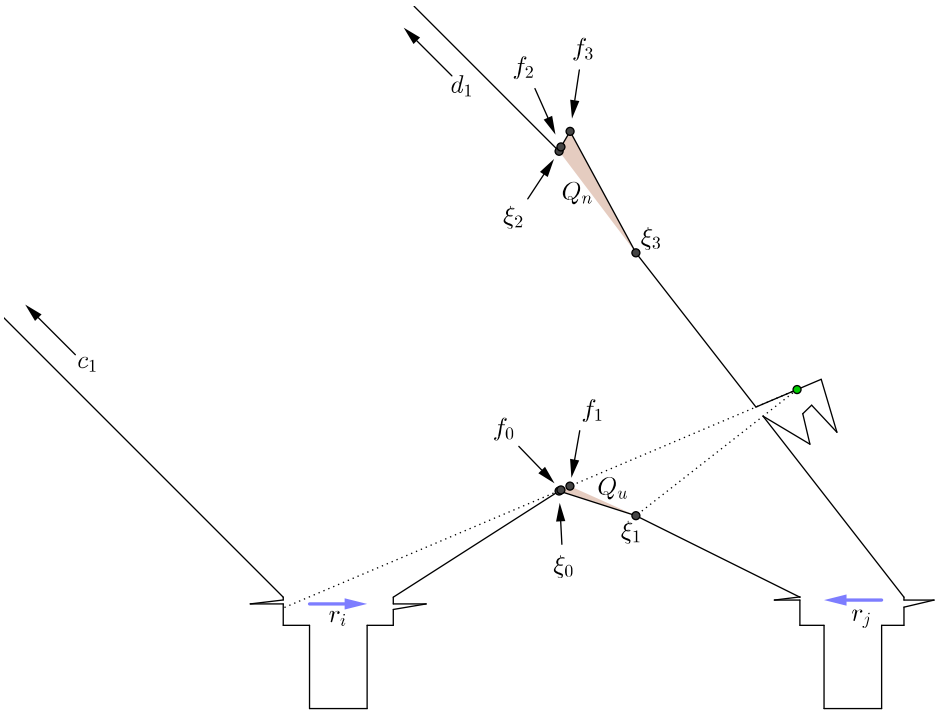


Figure 30: The inversion gadget. The nook and umbra (the brown areas) for the pair of guard segments r_i, r_j impose the inequality $x_i x_j = 1$ on the variables x_i and x_j represented by r_i and r_j. The green point is a stationary guard position which sees the umbra but nothing above the line $\overleftrightarrow{f_0 f_1}$.

corridor, we first scale it down by the factor $\frac{1}{CN^2}$ and then translate it so that the points c_1, d_1 of the gadget coincides with the endpoints of the right entrance of the corridor with the same names. We thus obtain that the point $(0,0)$ in the gadget becomes the point $m := c_1 + (1,-1)$ in $\mathcal{P}$.

Lemma 41. *Let $\mathcal{P}'_{inv}$ be the polygon obtained from the inversion gadget by closing it by adding the edge $c_1 d_1$. A set of points $G \subset \mathcal{P}'_{inv}$ of cardinality at most 3 guards $\mathcal{P}'_{inv}$ if and only if*

- *there is exactly one guard placed at each guard segment r_i, r_j and at the stationary guard position,*
- *the variables x_i, x_j corresponding to the guard segments r_i, r_j, respectively, satisfy the equation $x_i \cdot x_j = 1$.*

Proof. Assume that the polygon is guarded by a set G of at most 3 guards. Similarly as in Lemma 29 we can show that there must be exactly one guard at each guard segment and at the stationary guard position. It then follows from Lemma 40 that $x_i \cdot x_j = 1$.

Now assume that there is exactly one guard placed at each guard segment r_i, r_j and at the stationary guard position, and that the variables x_i, x_j represented by the guards at r_i, r_j satisfy $x_i \cdot x_j = 1$. Then the whole area of Q_u and Q_n is seen by the guards. The remaining area is clearly also seen by the guards. □

C.13.3 Connecting the gadget with a corridor

We now need to show that our gadget construction satisfies the conditions of Lemma 21.

Proof of Lemma 21 for the inversion gadget. Note that in the inversion gadget, the segments r_i, r_j have lengths of $\frac{3/2}{CN^2}$ and their right endpoints are placed at positions $m + (\frac{2}{CN^2}, 0)$ and $m + (\frac{15.4}{CN^2}, \frac{0.1}{CN^2})$, respectively, where $m := c_1 + (1,-1)$. As the conditions of Lemma 31 (here

used in a simplified version for a gadget with only two guard segments r_i and r_j) are satisfied, Properties 1 and 2 hold.

Property 3 also holds, as the stationary guard position in the gadget cannot see the edge c_1d_1. □

C.13.4 Summary

Lemma 42. *Consider the inversion gadget together with the corresponding corridor representing an inequality $x_i \cdot x_j = 1$, as described below. The following properties hold.*

- *The gadget and the corridor fit into a rectangular box of height 3.*
- *For any guard set of $\mathcal{P}$, at least 5 guards have to be placed in the corridor and the gadget in total.*
- *Assume that in the main area $\mathcal{P}_M$, there is exactly one guard at each guard segment, and there are no guards outside of the guard segments. Then the whole area of the corridor and the gadget can be guarded by 5 guards (together with the guards on the base line) if and only if the values x_i, x_j specified by the guards at the guard segments s_i, s_j satisfy the inequality $x_i \cdot x_j = 1$.*

Proof. The proof for the first property is similar to that for the addition gadget in Lemma 32.

From Lemma 24, there are at least 2 guards placed within the corridor. Furthermore, there must be at least 3 guards placed within the gadget by Lemma 41. That gives us at least 5 guards needed.

Assume that there are exactly 5 guards within the corridor and gadget and that the corridor is completely seen. Then, from Lemma 24 and 41, there is exactly one guard at each guard segment and each stationary guard position. Then, by Lemma 24, the values x_i, x_j specified by s_i, s_j, and the values specified by r_i, r_j, are the same. Lemma 24 gives that the guards at r_i, r_j must see the critical segments of both Q_n and Q_u. Then, by Lemma 41, the values x_i, x_j thus satisfy $x_i \cdot x_j = 1$.

On the other hand, assume that $x_i \cdot x_j = 1$. We first place a guard at every of the 3 stationary guard positions in the corridor and gadget. By Lemmas 24 and 41, if we set guards at the 2 guard segments so that the values specified by guards at r_i, r_j are x_i, x_j, then the whole area of the gadget is guarded. □

C.14 Putting it all together

Let Φ be an ETR-INV formula with n variables, k_1 equations of the form $x_i + x_j = x_l$, and k_2 equations of the form $x_i \cdot x_j = 1$. We have already explained how to construct the polygon $\mathcal{P}(\Phi)$, but we shall here give a brief summary of the process. We start by constructing the main area with $4n$ guard segments. We modify the pockets corresponding to the variables x_i for which Φ contains equation $x_i = 1$, as described in Section C.8. To enforce dependency between the base line guard segments corresponding to the same variable, we construct $3n$ $\geq$-orientation gadgets (attached at the right side of the polygon) and $3n$ $\leq$-orientation gadgets (attached at the left side), as described in Section C.12. For each equality of the form $x_i + x_j = x_l$ in Φ, we construct a corresponding $\geq$-addition gadget (attached at the right side), and a $\leq$-addition gadget (attached at the left side), as described in Sections C.10 and C.11, respectively. For each equality of the form $x_i \cdot x_j = 1$ in Φ we construct a corresponding inversion gadget (attached at the right side), as described in Section C.13. The total number of gadgets at each side of $\mathcal{P}$ is therefore at most $3n + k_1 + k_2 = k'$, as stated in Section C.7.

Without loss of generality, we assume that the y-coordinate of the base line of $\mathcal{P}(\Phi)$ is 0. We set $g(\Phi) := 58n + 20k_1 + 5k_2$. Then $(\mathcal{P}(\Phi), g(\Phi))$ is an instance of the art gallery problem, and the following theorem holds.

Theorem 43. *The polygon $\mathcal{P}(\Phi)$ has vertices at rational coordinates, which can be computed in polynomial time. Moreover, there exist constants $d_0, \ldots, d_{n-1} \in \mathbb{Q}$ such that for any $x := (x_0, \ldots, x_{n-1}) \in \mathbb{R}^n$, x is a solution to Φ if and only if there exists a guard set G of cardinality $g(\Phi)$ containing guards at all the positions $(x_0 + d_0, 0) \ldots, (x_{n-1} + d_{n-1}, 0)$.*

Proof. Consider a guard set G of the polygon $\mathcal{P} := \mathcal{P}(\Phi)$. By Lemma 20, G has at least $4n$ guards placed in $\mathcal{P}_M$, and if the number of guards within $\mathcal{P}_M$ equals $4n$, then there must be exactly one guard at each guard segment. Lemma 37 implies that within each of the $6n$ orientation gadgets of $\mathcal{P}$ together with the corresponding corridors, there are at least 9 guards, giving at least $54n$ guards in total. Similarly, from Lemmas 32 and 36 we obtain that there must be at least 10 guards placed within each $\geq$-addition gadget and each $\leq$-addition gadget plus the corresponding corridors, giving at least $20k_1$ guards. By Lemma 42, there are at least 5 guards within each inversion gadget and the corresponding corridor, giving at least $5k_2$ guards in total. Therefore, G has at least $58n + 20k_1 + 5k_2$ guards, which is equal to $g(\Phi)$.

If a guard set of size $g(\Phi)$ exists, then there are exactly $4n$ guards in $\mathcal{P}_M$, 9 guards within each orientation gadget, 10 guards within each $\geq$-addition gadget and each $\leq$-addition gadget, and 5 guards within each inversion gadget. The same lemmas give us then that there is exactly one guard at each guard segment and each stationary guard position, and no guards away from the guard segments or the stationary guard positions. Also, the variables $x_0, \ldots, x_{n-1}$ specified by the guard segments $s_0, \ldots, s_{n-1}$ are a solution to Φ.

On the other hand, if there exists a solution to Φ, then we get a guard set of size $g(\Phi)$ by placing the guards accordingly. It is thus clear that the solutions to Φ correspond to the optimal guard sets of $\mathcal{P}$, as stated in the theorem.

Due to Lemmas 19 and 25, we get that the vertices of $\mathcal{P}_M$ and the corridor vertices are all rational, with the enumerators and denominators polynomially bounded. Next, consider all the vertices of the gadgets. Each gadget is first described as a polygon with coordinates where enumerators and denominators are both of size $\Theta(1)$ (as this construction is fixed and it does not depend on the formula Φ; and we can easily choose the vertices so that they are all at rational coordinates), and this polygon is subsequently scaled down by a factor of $\frac{1}{CN^2}$ and attached at a corridor entrance, which also has polynomially bounded enumerators and denominators. Thus, the coordinates of the vertices in the gadgets have polynomially bounded complexity and can be computed in polynomial time. □

We can now prove the main theorem of the paper.

Theorem 1. *The art gallery problem is $\exists\mathbb{R}$-complete, even the restricted variant where we are given a polygon with vertices at integer coordinates.*

Proof. By Theorem 4, the art gallery problem is in the complexity class $\exists\mathbb{R}$. From Theorem 6 we know that the problem ETR-INV is $\exists\mathbb{R}$-complete. We presented a polynomial time construction of an instance $(\mathcal{P}(\Phi), g(\Phi))$ of the art gallery problem from an instance Φ of ETR. Theorem 43 gives that it is $\exists\mathbb{R}$-hard to solve the art gallery problem when the coordinates of the polygon vertices are given by rational numbers. Note that the number of vertices of $\mathcal{P}$ is proportional to the input length $|\Phi|$. By Theorem 43, there is a polynomial $|\Phi|^m$ which is a bound on every denominator of a coordinate of a vertex in $\mathcal{P}$. The product Π of denominators of all coordinates of vertices of $\mathcal{P}$ thus has size at most $|\Phi|^{m\,O(|\Phi|)}$. It follows that we can express Π by $O(m|\Phi| \log |\Phi|)$ bits. By multiplying every coordinate of $\mathcal{P}$ by Π, we get a polygon $\mathcal{P}'$ with integer coordinates and the theorem follows. □

Theorem 2, restated below, likewise easily follows from Lemma 13 in Appendix B and Theorem 43.

Theorem 2. *Let Φ be an ETR formula with k variables. Then there is an instance $(\mathcal{P}, g)$ of the art gallery problem, and constants $c_1, d_1, \ldots, c_k, d_k \in \mathbb{Q}$, such that*

- *if Φ has a solution, then $\mathcal{P}$ has a guard set of size g, and*
- *for any guard set G of $\mathcal{P}$ of size g, there exists $(x_1, \ldots, x_k) \in S_\Phi$ such that G contains guards at positions $(c_1x_1 + d_1, 0), \ldots, (c_kx_k + d_k, 0)$.*

We can now prove Corollary 3, restated below.

Corollary 3. *Given any real algebraic number α, there exists a polygon $\mathcal{P}$ with vertices at rational coordinates such that in any optimal guard set of $\mathcal{P}$ there is a guard with an x-coordinate equal to α.*

Proof. Let $P(x)$ be a polynomial of degree more than 0 in one variable x such that the equation $P(x) = 0$ has α as a solution. The equation might have other solutions as well, but we can choose integers p_1, p_2, q_1, q_2 such that α is the only solution in the interval $[p_1/q_1, p_2/q_2]$. Then the formula $P(x) = 0 \wedge p_1 \leq q_1x \wedge q_2x \leq p_2$ is an instance of the problem ETR with a unique solution $x = \alpha$. Now, by Theorem 2, there exists a polygon $\mathcal{P}$ and rational constants c, d such that in any optimal guard set of $\mathcal{P}$, one guard has coordinates $(c\alpha+d, 0)$. By subtracting d from the x-coordinate of all vertices of $\mathcal{P}$ and then dividing all coordinates by c, we get a polygon $\mathcal{P}'$ such that any optimal guard set of $\mathcal{P}'$ has a guard at the point $(\alpha, 0)$. □

References

[1] Zachary Abel, Erik D. Demaine, Martin L. Demaine, Sarah Eisenstat, Jayson Lynch, and Tao B. Schardl. Who needs crossings? Hardness of plane graph rigidity. In *32nd International Symposium on Computational Geometry (SoCG 2016)*, pages 3:1–3:15, 2016.

[2] Mikkel Abrahamsen, Anna Adamaszek, and Tillmann Miltzow. Irrational guards are sometimes needed. In *33rd International Symposium on Computational Geometry (SoCG 2017)*, 2017. To appear.

[3] Alok Aggarwal. *The art gallery theorem: its variations, applications and algorithmic aspects*. PhD thesis, 1984.

[4] Alfred V. Aho, John E. Hopcroft, and Jeffrey D. Ullman. *The design and analysis of computer algorithms*. Addison-Wesley, 1975.

[5] Saugata Basu, Richard Pollack, and Marie-Françoise Roy. On the combinatorial and algebraic complexity of quantifier elimination. *Journal of the ACM*, 43(6):1002–1045, 1996.

[6] Patrice Belleville. Computing two-covers of simple polygons. Master's thesis, McGill University, 1991.

[7] Daniel Bienstock. Some provably hard crossing number problems. *Discrete & Computational Geometry*, 6(3):443–459, 1991.

[8] Édouard Bonnet and Tillmann Miltzow. Parameterized hardness of art gallery problems. In *24th Annual European Symposium on Algorithms (ESA)*, pages 19:1–19:17, 2016.

[9] Dorit Borrmann, Pedro J. de Rezende, Cid C. de Souza, Sándor P. Fekete, Stephan Friedrichs, Alexander Kröller, Andreas Nüchter, Christiane Schmidt, and Davi C. Tozoni. Point guards and point clouds: Solving general art gallery problems. In *Symposuim on Computational Geometry 2013 (SoCG 2013)*, pages 347–348, 2013.

[10] Björn Brodén, Mikael Hammar, and Bengt J. Nilsson. Guarding lines and 2-link polygons is APX-hard. In *Proceedings of the 13th Canadian Conference on Computational Geometry (CCCG 2001)*, pages 45–48, 2001.

[11] John Canny. Some algebraic and geometric computations in PSPACE. In *Proceedings of the twentieth annual ACM symposium on Theory of computing (STOC 1988)*, pages 460–467. ACM, 1988.

[12] Jean Cardinal. Computational geometry column 62. *SIGACT News*, 46(4):69–78, 2015.

[13] Jean Cardinal and Udo Hoffmann. Recognition and complexity of point visibility graphs. *Discrete & Computational Geometry*, 57(1):164–178, 2017.

[14] Mark De Berg, Marc Van Kreveld, Mark Overmars, and Otfried Cheong Schwarzkopf. *Computational Geometry: Algorithms and Applications (3rd edition)*. Springer-Verlag, 2008.

[15] Pedro J. de Rezende, Cid C. de Souza, Stephan Friedrichs, Michael Hemmer, Alexander Kröller, and Davi C. Tozoni. Engineering art galleries. In Peter Sanders Lasse Kliemann, editor, *Algorithm Engineering – Selected Results and Surveys*, pages 379–417. Springer International Publishing, 2016.

[16] Satyan L. Devadoss and Joseph O'Rourke. *Discrete and Computational Geometry*. Princeton University Press, 2011.

[17] Alon Efrat and Sariel Har-Peled. Guarding galleries and terrains. *Information Processing Letters*, 100(6):238–245, 2006.

[18] Stephan Eidenbenz, Christoph Stamm, and Peter Widmayer. Inapproximability results for guarding polygons and terrains. *Algorithmica*, 31(1):79–113, 2001.

[19] Stephan Friedrichs, Michael Hemmer, James King, and Christiane Schmidt. The continuous 1.5D terrain guarding problem: Discretization, optimal solutions, and PTAS. *Journal of Computational Geometry*, 7(1):256–284, 2016.

[20] Jugal Garg, Ruta Mehta, Vijay V. Vazirani, and Sadra Yazdanbod. ETR-completeness for decision versions of multi-player (symmetric) Nash equilibria. In *Proceedings of the 42nd International Colloquium on Automata, Languages, and Programming (ICALP 2015), part 1*, volume 9134 of *Lecture Notes in Computer Science (LNCS)*, pages 554–566, 2015.

[21] James King and Erik Krohn. Terrain guarding is NP-hard. In *Proceedings of the Twenty-First Annual ACM-SIAM Symposium on Discrete Algorithms (SODA 2010)*, pages 1580–1593, 2010.

[22] David G. Kirkpatrick. An $O(\lg \lg \mathrm{OPT})$-approximation algorithm for multi-guarding galleries. *Discrete & Computational Geometry*, 53(2):327–343, 2015.

[23] Linda Kleist, Tillmann Miltzow, and Paweł Rzążewski. Is area univesality $\forall\exists\mathbb{R}$-complete? In *The European Workshop on Computational Geometry (EuroCG 2017)*, pages 181–184, 2017. Full version in preparation.

[24] Erik Krohn and Bengt J. Nilsson. Approximate guarding of monotone and rectilinear polygons. *Algorithmica*, 66(3):564–594, 2013.

[25] D. T. Lee and Arthur K. Lin. Computational complexity of art gallery problems. *IEEE Transactions on Information Theory*, 32(2):276–282, 1986.

[26] Maple 2016.1, Maplesoft, a division of Waterloo Maple Inc., Waterloo, Ontario. Maple is a trademark of Waterloo Maple Inc.

[27] Jiří Matoušek. *Lectures on Discrete Geometry*, volume 212 of *Graduate Texts in Mathematics*. Springer-Verlag New York, 2002.

[28] Jiří Matoušek. Intersection graphs of segments and $\exists\mathbb{R}$. 2014. Preprint, `https://arxiv.org/abs/1406.2636`.

[29] Colin McDiarmid and Tobias Müller. Integer realizations of disk and segment graphs. *Journal of Combinatorial Theory, Series B*, 103(1):114–143, 2013.

[30] Nicolai E Mnëv. The universality theorems on the classification problem of configuration varieties and convex polytopes varieties. In Oleg Y. Viro, editor, *Topology and geometry – Rohlin seminar*, pages 527–543. Springer-Verlag Berlin Heidelberg, 1988.

[31] Joseph O'Rourke. *Art Gallery Theorems and Algorithms*. Oxford University Press, 1987.

[32] Joseph O'Rourke. *Computational Geometry in C*. Cambridge University Press, 1998.

[33] Joseph O'Rourke. Visibility. In Jacob E. Goodman and Joseph O'Rourke, editors, *Handbook of Discrete and Computational Geometry*, chapter 28. Chapman & Hall/CRC, second edition, 2004.

[34] Joseph O'Rourke and Kenneth Supowit. Some NP-hard polygon decomposition problems. *IEEE Transactions on Information Theory*, 29(2):181–190, 1983.

[35] Quintic function – Wikipedia, The Free Encyclopedia, 2017. [Online; accessed 14-March-2017].

[36] Jürgen Richter-Gebert and Günter M. Ziegler. Realization spaces of 4-polytopes are universal. *Bulletin of the American Mathematical Society*, 32(4):403–412, 1995.

[37] Marcus Schaefer. Complexity of some geometric and topological problems. In *Proceedings of the 17th International Symposium on Graph Drawing (GD 2009)*, volume 5849 of *Lecture Notes in Computer Science (LNCS)*, pages 334–344. Springer, 2009.

[38] Marcus Schaefer. Realizability of graphs and linkages. In János Pach, editor, *Thirty Essays on Geometric Graph Theory*, chapter 23, pages 461–482. Springer-Verlag New York, 2013.

[39] Marcus Schaefer and Daniel Štefankovič. Fixed points, Nash equilibria, and the existential theory of the reals. *Theory of Computing Systems*, 60(2):172–193, 2017.

[40] Dietmar Schuchardt and Hans-Dietrich Hecker. Two NP-hard art-gallery problems for ortho-polygons. *Mathematical Logic Quarterly*, 41:261–267, 1995.

[41] Thomas C. Shermer. Recent results in art galleries. *Proceedings of the IEEE*, 80(9):1384–1399, 1992.

[42] Yaroslav Shitov. The complexity of positive semidefinite matrix factorization. 2016. Preprint, `http://arxiv.org/abs/1606.09065`.

[43] Peter W. Shor. Stretchability of pseudolines is NP-hard. In Peter Gritzmann and Bernd Sturmfels, editors, *Applied Geometry and Discrete Mathematics: The Victor Klee Festschrift*, volume 4 of *DIMACS – Series in Discrete Mathematics and Theoretical Computer Science*, pages 531–554. American Mathematical Society and Association for Computing Machinery, 1991.

[44] Ana Paula Tomás. Guarding thin orthogonal polygons is hard. In *Fundamentals of Computation Theory*, pages 305–316. Springer, 2013.

[45] Jorge Urrutia. Art gallery and illumination problems. In J.-R. Sack and J. Urrutia, editors, *Handbook of Computational Geometry*, chapter 22, pages 973–1027. Elsevier, 2000.

D

MINIMUM PERIMETER-SUM PARTITIONS IN THE PLANE

Minimum Perimeter-Sum Partitions in the Plane*

Mikkel Abrahamsen

Department of Computer Science, University of Copenhagen, Denmark
miab@di.ku.dk

Mark de Berg, Kevin Buchin, Mehran Mehr, and Ali D. Mehrabi

Department of Computer Science, TU Eindhoven, the Netherlands
mdberg@win.tue.nl,k.a.buchin@tue.nl,m.mehr@tue.nl,amehrabi@win.tue.nl

22nd November 2016

Abstract

Let P be a set of n points in the plane. We consider the problem of partitioning P into two subsets P_1 and P_2 such that the sum of the perimeters of $\text{CH}(P_1)$ and $\text{CH}(P_2)$ is minimized, where $\text{CH}(P_i)$ denotes the convex hull of P_i. The problem was first studied by Mitchell and Wynters in 1991 who gave an $O(n^2)$ time algorithm. Despite considerable progress on related problems, no subquadratic time algorithm for this problem was found so far. We present an algorithm solving the problem in $O(n \log^4 n)$ time and a $(1+\varepsilon)$-approximation algorithm for the same problem running in $O(n + 1/\varepsilon^2 \cdot \log^4(1/\varepsilon))$ time.

*MA is partly supported by Mikkel Thorup's Advanced Grant from the Danish Council for Independent Research under the Sapere Aude research career programme. MdB, KB, MM, and AM are supported by the Netherlands' Organisation for Scientific Research (NWO) under project no. 024.002.003, 612.001.207, 022.005025, and 612.001.118 respectively.

1 Introduction

The clustering problem is to partition a given data set into clusters (that is, subsets) according to some measure of optimality. We are interested in clustering problems where the data set is a set P of points in Euclidean space. Most of these clustering problems fall into one of two categories: problems where the maximum cost of a cluster is given and the goal is to find a clustering consisting of a minimum number of clusters, and problems where the number of clusters is given and the goal is to find a clustering of minimum total cost. In this paper we consider a basic problem of the latter type, where we wish to find a bipartition (P_1, P_2) of a planar point set P. Bipartition problems are not only interesting in their own right, but also because bipartition algorithms can form the basis of hierarchical clustering methods.

There are many possible variants of the bipartition problem on planar point sets, which differ in how the cost of a clustering is defined. A variant that received a lot of attention is the 2-center problem [6, 10, 11, 14, 19], where the cost of a partition (P_1, P_2) of the given point set P is defined as the maximum of the radii of the smallest enclosing disks of P_1 and P_2. Other cost functions that have been studied include the maximum diameter of the two point sets [2] and the sum of the diameters [13]; see also the survey by Agarwal and Sharir [1] for some more variants.

A natural class of cost function considers the size of the convex hulls $\textsc{ch}(P_1)$ and $\textsc{ch}(P_2)$ of the two subsets, where the size of $\textsc{ch}(P_i)$ can either be defined as the area of $\textsc{ch}(P_i)$ or as the perimeter $\mathrm{per}(P_i)$ of $\textsc{ch}(P_i)$. (The perimeter of $\textsc{ch}(P_i)$ is the length of the boundary $\partial\,\textsc{ch}(P_i)$.) This class of cost functions was already studied in 1991 by Mitchell and Wynters [16]. They studied four problem variants: minimize the sum of the perimeters, the maximum of the perimeters, the sum of the areas, or the maximum of the areas. In three of the four variants the convex hulls $\textsc{ch}(P_1)$ and $\textsc{ch}(P_2)$ in an optimal solution may intersect [16, full version]—only in the *minimum perimeter-sum problem* the optimal bipartition is guaranteed to be a so-called *line partition*, that is, a solution with disjoint convex hulls. For each of the four variants they gave an $O(n^3)$ algorithm that uses $O(n)$ space and for all except the minimum-maximum area problem, they also gave an $O(n^2)$ algorithm that uses $O(n^2)$ space; their algorithms only consider line partitions (which in the case of the minimum perimeter-sum problem implies an optimal bipartition). Around the same time, the minimum-perimeter sum problem was studied for partitions into k subsets for $k > 2$; for this variant Capoyleas *et al.* [5] presented an algorithm with running time $O(n^{6k})$. Mitchell and Wynters mentioned the improvement of the space requirement of the quadratic-time algorithm as an open problem, and they stated the existence of a subquadratic algorithm for any of the four variants as the most prominent open problem.

Rokne *et al.* [17] made progress on the first question, by presenting an $O(n^2 \log n)$ algorithm that uses only $O(n)$ space for the line-partition version of each of the four problems. Devillers and Katz [9] gave algorithms for the min-max variant of the problem, both for area and perimeter, which run in $O((n+k)\log^2 n)$ time. Here k is a parameter that is only known to be in $O(n^2)$, although Devillers and Katz suspected that k is subquadratic. They also gave linear-time algorithms for these problems when the point set P is in convex position and given in cyclic order. Segal [18] proved an $\Omega(n \log n)$ lower bound for the min-max problems. Very recently, and apparently unaware of the earlier work on these problems, Cho *et al.* [7] presented an $O(n^2 \log^2 n)$ time algorithm for the minimum-perimeter-sum problem and an $O(n^4 \log^2 n)$ time algorithm for the minimum-area-sum problem (considering all partitions, not only line partitions). Despite these efforts, the main question is still open: is it possible to obtain a subquadratic algorithm for any of the four bipartition problems based on convex-hull size?

Our contribution. We answer the question above affirmatively by presenting a subquadratic algorithm for the minimum perimeter-sum bipartition problem in the plane.

As mentioned, an optimal solution (P_1, P_2) to the minimum perimeter-sum bipartition problem must be a line partition. A straightforward algorithm would generate all $\Theta(n^2)$ line partitions and compute the value $\text{per}(P_1) + \text{per}(P_2)$ for each of them. If the latter is done from scratch for each partition, the resulting algorithm runs in $O(n^3 \log n)$ time. The algorithms by Mitchell and Wynters [16] and Rokne *et al.* [17] improve on this by using that the different line bipartitions can be generated in an ordered way, such that subsequent line partitions differ in at most one point. Thus the convex hulls do not have to be recomputed from scratch, but they can be obtained by updating the convex hulls of the previous bipartition. To obtain a subquadratic algorithm a fundamentally new approach is necessary: we need a strategy that generates a subquadratic number of candidate partitions, instead considering all line partitions. We achieve this as follows.

We start by proving that an optimal bipartition (P_1, P_2) has the following property: there is a set of $O(1)$ canonical orientations such that P_1 can be separated from P_2 by a line with a canonical orientation, or the distance between $\text{CH}(P_1)$ and $\text{CH}(P_2)$ is $\Omega(\min(\text{per}(P_1), \text{per}(P_2))$. There are only $O(1)$ bipartitions of the former type, and finding the best among them is relatively easy. The bipartitions of the second type are much more challenging. We show how to employ a compressed quadtree to generate a collection of $O(n)$ canonical 5-gons—intersections of axis-parallel rectangles and canonical halfplanes—such that the smaller of $\text{CH}(P_1)$ and $\text{CH}(P_2)$ (in a bipartition of the second type) is contained in one of the 5-gons.

It then remains to find the best among the bipartitions of the second type. Even though the number of such bipartitions is linear, we cannot afford to compute their perimeters from scratch. We therefore design a data structure to quickly compute $\text{per}(P \cap Q)$, where Q is a query canonical 5-gon. Brass *et al.* [4] presented such a data structure for the case where Q is an axis-parallel rectangle. Their structure uses $O(n \log^2 n)$ space and has $O(\log^5 n)$ query time; it can be extended to handle canonical 5-gons as queries, at the cost of increasing the space usage to $O(n \log^3 n)$ and the query time to $O(\log^7 n)$. Our data structure improves upon this: it has $O(\log^4 n)$ query time for canonical 5-gons (and $O(\log^3 n)$ for rectangles) while using the same amount of space. Using this data structure to find the best bipartition of the second type we obtain our main result: an exact algorithm for the minimum perimeter-sum bipartition problem that runs in $O(n \log^4 n)$ time.

Besides our exact algorithm, we present a linear-time $(1 + \varepsilon)$-approximation algorithm. Its running time is $O(n + T(n_\varepsilon))$, where $T(n_\varepsilon)$ is the running time of an exact algorithm on an instance of size $n_\varepsilon = O(1/\varepsilon^2)$. When we plug in our exact algorithm as a subroutine, we thus obtain $O(n + 1/\varepsilon^2 \cdot \log^4(1/\varepsilon))$ running time.

2 The exact algorithm

In this section we present an exact algorithm for the minimum-perimeter-sum partition problem. We first prove a separation property that an optimal solution must satisfy, and then we show how to use this property to develop a fast algorithm.

Let P be the set of n points in the plane for which we want to solve the minimum-perimeter-sum partition problem. An optimal partition (P_1, P_2) of P has the following two basic properties: P_1 and P_2 are non-empty, and the convex hulls $\text{CH}(P_1)$ and $\text{CH}(P_2)$ are disjoint. In the remainder, whenever we talk about a partition of P, we refer to a partition with these two properties.

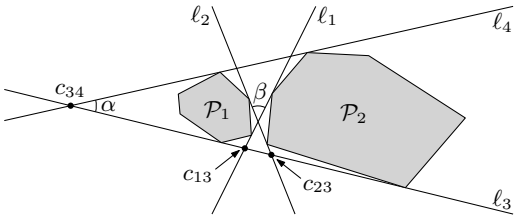


Figure 1: The angles α and β.

2.1 Geometric properties of an optimal partition

Consider a partition (P_1, P_2) of P. Define $\mathcal{P}_1 := \text{CH}(P_1)$ and $\mathcal{P}_2 := \text{CH}(P_2)$ to be the convex hulls of P_1 and P_2, respectively, and let ℓ_1 and ℓ_2 be the two inner common tangents of $\mathcal{P}_1$ and $\mathcal{P}_2$. The lines ℓ_1 and ℓ_2 define four wedges: one containing P_1, one containing P_2, and two empty wedges. We call the opening angle of the empty wedges the *separation angle* of P_1 and P_2. Furthermore, we call the distance between $\text{CH}(P_1)$ and $\text{CH}(P_2)$ the *separation distance* of P_1 and P_2.

Theorem 1. *Let P be a set of n points in the plane, and let (P_1, P_2) be a partition of P that minimizes $\text{per}(P_1) + \text{per}(P_2)$. Then the separation angle of P_1 and P_2 is at least $\pi/6$ or the separation distance is at least $c_{\text{sep}} \cdot \min(\text{per}(P_1), \text{per}(P_2))$, where $c_{\text{sep}} := 1/250$.*

The remainder of this section is devoted to proving Theorem 1. To this end let (P_1, P_2) be a partition of P that minimizes $\text{per}(P_1) + \text{per}(P_2)$. Let ℓ_3 and ℓ_4 be the outer common tangents of $\mathcal{P}_1$ and $\mathcal{P}_2$. We define α to be the angle between ℓ_3 and ℓ_4. More precisely, if ℓ_3 and ℓ_4 are parallel we define $\alpha := 0$, otherwise we define α as the opening angle of the wedge defined by ℓ_3 and ℓ_4 containing $\mathcal{P}_1$ and $\mathcal{P}_2$. We denote the separation angle of P_1 and P_2 by β; see Fig. 1. Let c_{ij} be the intersection point between ℓ_i and ℓ_j, where $i < j$. If ℓ_3 and ℓ_4 are parallel, we choose c_{34} as a point at infinity on ℓ_3. Assume without loss of generality that neither ℓ_1 nor ℓ_2 separate $\mathcal{P}_1$ from c_{34} and that ℓ_3 is the outer common tangent such that $\mathcal{P}_1$ and $\mathcal{P}_2$ are to the left of ℓ_3 when traversing ℓ_3 from c_{34} to an intersection point in $\ell_3 \cap \mathcal{P}_1$. Assume furthermore that c_{13} is closer to c_{34} than c_{23}.

For two lines, rays, or segments r_1, r_2, let $\angle(r_1, r_2)$ be the angle we need to rotate r_1 in counterclockwise direction until r_1 and r_2 are parallel. For three points a, b, c, let $\angle(a, b, c) := \angle(ba, bc)$. For $i = 1, 2$ and $j = 1, 2, 3, 4$, let s_{ij} be a point in $P_i \cap \ell_j$. Let $\partial\mathcal{P}_i$ denote the boundary of $\mathcal{P}_i$ and $\text{per}(\mathcal{P}_i)$ the perimeter of $\mathcal{P}_i$. Furthermore, let $\partial\mathcal{P}_i(x, y)$ denote the portion of $\partial\mathcal{P}_i$ from $x \in \partial\mathcal{P}_i$ counterclockwise to $y \in \partial\mathcal{P}_i$, and $\text{length}(\partial\mathcal{P}_i(x, y))$ denote the length of $\partial\mathcal{P}_i(x, y)$.

Lemma 2. *We have $\alpha + 3\beta \geqslant \pi$.*

Proof. Since $\text{per}(\mathcal{P}_1) + \text{per}(\mathcal{P}_2)$ is minimum, we know that

$$\text{length}(\partial\mathcal{P}_1(s_{13}, s_{14})) + \text{length}(\partial\mathcal{P}_2(s_{24}, s_{23})) \;\leqslant\; \Psi,$$

where $\Psi := |s_{13}s_{23}| + |s_{14}s_{24}|$. Furthermore, we know that $s_{11}, s_{12} \in \partial\mathcal{P}_1(s_{13}, s_{14})$ and $s_{21}, s_{22} \in \partial\mathcal{P}_1(s_{24}, s_{23})$. We thus have

$$\text{length}(\partial\mathcal{P}_1(s_{13}, s_{14})) + \text{length}(\partial\mathcal{P}_2(s_{24}, s_{23})) \;\geqslant\; \Phi,$$

where $\Phi := |s_{13}s_{11}| + |s_{11}s_{12}| + |s_{12}s_{14}| + |s_{24}s_{21}| + |s_{21}s_{22}| + |s_{22}s_{23}|$. Hence, we must have

$$\Phi \;\leqslant\; \Psi. \tag{1}$$

Now assume that $\alpha + 3\beta < \pi$. We will show that this assumption, together with inequality (1), leads to a contradiction, thus proving the lemma. To this end we will argue that if (1) holds, then it must also hold when (i) s_{21} or s_{22} coincides with c_{12}, and (ii) s_{11} or s_{12} coincides with c_{12}. To finish the proof it then suffices to observe that that if (i) and (ii) hold, then $\mathcal{P}_1$ and $\mathcal{P}_2$ touch in c_{12} and so (1) contradicts the triangle inequality.

It remains to argue that if (1) holds, then we can create a situation where (1) holds and (i) and (ii) hold as well. To this end we ignore that the points s_{ij} are specific points in the set P and allow the point s_{ij} to move on the tangent ℓ_j, as long as the movement preserves (1). Moving s_{13} along ℓ_3 away from s_{23} increases Ψ more than it increases Φ, so (1) is preserved. Similarly, we can move s_{14} away from s_{24}, s_{23} away from s_{13}, and s_{24} away from s_{13}.

We first show how to create a situation where (i) holds, and (1) still holds as well. Let $\gamma_{ij} := \angle(\ell_i, \ell_j)$. We consider two cases.

- *Case (A):* $\gamma_{32} < \pi - \beta$.

  Note that $\angle(xs_{23}, \ell_2) \geqslant \gamma_{32}$ for any $x \in s_{22}c_{12}$. However, by moving s_{23} sufficiently far away we can make $\angle(xs_{23}, \ell_2)$ arbitrarily close to γ_{32}, and we can ensure that $\angle(xs_{23}, \ell_2) < \pi - \beta$ for any point $x \in s_{22}c_{12}$. We now let the point x move at unit speed from s_{22} towards c_{12}. To be more precise, let $T := |s_{22}c_{12}|$, let $\mathbf{v}$ be the unit vector with direction from c_{23} to c_{12}, and for any $t \in [0, T]$ define $x(t) := s_{22} + t \cdot \mathbf{v}$. Note that $x(0) = s_{22}$ and $x(T) = c_{12}$.

  Let $a(t) := |x(t)s_{23}|$ and $b(t) := |x(t)s_{21}|$.

  Lemma 11 in the appendix gives that

  $$a'(t) = -\cos(\angle(x(t)s_{23}, \ell_2)) \text{ and } b'(t) = \cos(\angle(\ell_2, x(t)s_{21})).$$

  Since $\angle(x(t)s_{23}, \ell_2) < \pi - \beta$ for any value $t \in [0, T]$, we get $a'(t) < -\cos(\pi - \beta)$. Furthermore, we have $\angle(\ell_2, x(t)s_{21}) \geqslant \pi - \beta$ and hence $b'(t) \leqslant \cos(\pi - \beta)$. Therefore, $a'(t) + b'(t) < 0$ for any t and we conclude that $a(T) + b(T) \leqslant a(0) + b(0)$. This is the same as $|s_{21}c_{12}| + |c_{12}s_{23}| \leqslant |s_{21}s_{22}| + |s_{22}s_{23}|$, so (1) still holds when we substitute s_{22} by c_{12}.

- *Case (B):* $\gamma_{32} \geqslant \pi - \beta$.

  Using our assumption $\alpha + 3\beta < \pi$ we get $\gamma_{32} > \alpha + 2\beta$. Note that $\gamma_{14} = \pi - \gamma_{32} + \alpha + \beta$. Hence, $\gamma_{14} < \pi - \beta$. By moving s_{24} and s_{21}, we can in a similar way as in Case (A) argue that (1) still holds when we substitute s_{21} by c_{12}.

We conclude that in both cases we can ensure (i) without violating (1).

Since $\gamma_{42} \leqslant \gamma_{32}$ and $\gamma_{13} \leqslant \gamma_{14}$, we likewise have $\gamma_{42} < \pi - \beta$ or $\gamma_{13} < \pi - \beta$. Hence, we can substitute s_{11} or s_{12} by c_{12} without violating (1), thus finishing the proof. □

Let $\text{dist}(\mathcal{P}_1, \mathcal{P}_2) := \min_{(p,q) \in \mathcal{P}_1 \times \mathcal{P}_2} |pq|$ denote the separation distance between $\mathcal{P}_1$ and $\mathcal{P}_2$. Recall that α denotes the angle between the two common outer tangents of $\mathcal{P}_1$ and $\mathcal{P}_2$; see Fig. 1

Lemma 3. *We have*

$$\textit{dist}(\mathcal{P}_1, \mathcal{P}_2) \;\geqslant\; f(\alpha) \cdot \text{per}(\mathcal{P}_1), \tag{2}$$

where $f\colon [0, \pi] \to \mathbb{R}$ *is the increasing function*

$$f(\varphi) \;:=\; \frac{\sin(\varphi/4)}{1 + \sin(\varphi/4)} \cdot \frac{\sin(\varphi/2)}{1 + \sin(\varphi/2)} \cdot \frac{1 - \cos(\varphi/4)}{2}.$$

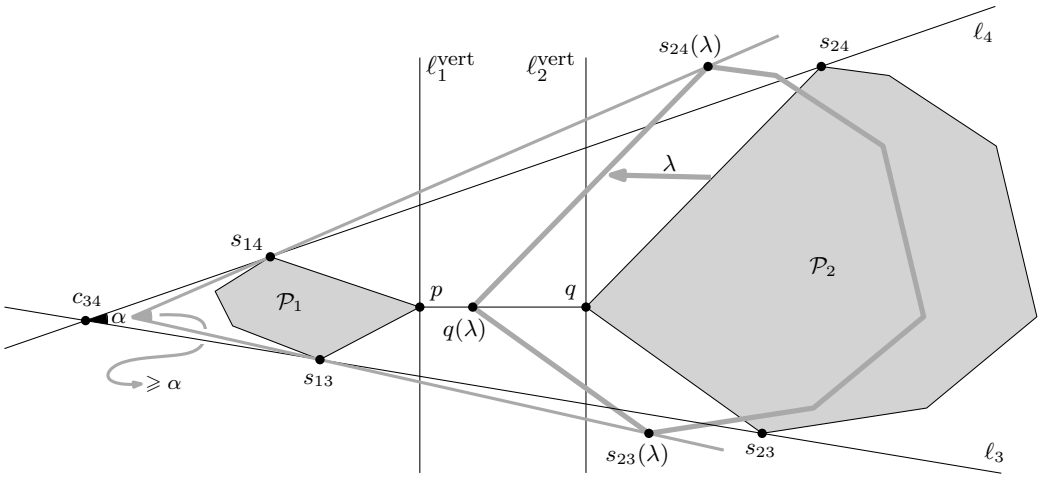


Figure 2: Illustration for the proof of Lemma 3.

Proof. The statement is trivial if $\alpha = 0$ so assume $\alpha > 0$. Let $p \in \mathcal{P}_1$ and $q \in \mathcal{P}_2$ be points so that $|pq| = \text{dist}(\mathcal{P}_1, \mathcal{P}_2)$ and assume without loss of generality that pq is a horizontal segment with p being its left endpoint. Let ℓ_1^{vert} and ℓ_2^{vert} be vertical lines containing p and q, respectively. Note that $\mathcal{P}_1$ is in the closed half-plane to the left of ℓ_1^{vert} and $\mathcal{P}_2$ is in the closed half-plane to the right of ℓ_2^{vert}. Recall that s_{ij} denotes a point on $\partial\mathcal{P}_i \cap \ell_j$.

We prove the following claim in the appendix.

Claim: There exist two convex polygons $\mathcal{P}_1'$ and $\mathcal{P}_2'$ satisfying the following conditions:

1. $\mathcal{P}_1'$ and $\mathcal{P}_2'$ have the same outer common tangents as $\mathcal{P}_1$ and $\mathcal{P}_2$, namely ℓ_3 and ℓ_4.
2. $\mathcal{P}_1'$ is to the left of ℓ_1^{vert} and $p \in \partial\mathcal{P}_1'$; and $\mathcal{P}_2'$ is to right of ℓ_2^{vert} and $q \in \partial\mathcal{P}_2'$.
3. $\text{per}(\mathcal{P}_1') = \text{per}(\mathcal{P}_1)$.
4. $\text{per}(\mathcal{P}_1') + \text{per}(\mathcal{P}_2') \leqslant \text{per}(\text{CH}(\mathcal{P}_1' \cup \mathcal{P}_2'))$.
5. There are points $s_{ij}' \in \mathcal{P}_i' \cap \ell_j$ for all $i \in \{1,2\}$ and $j \in \{3,4\}$ such that $\partial\mathcal{P}_1'(s_{13}', p)$, $\partial\mathcal{P}_1'(p, s_{14}')$, $\partial\mathcal{P}_2'(s_{24}', q)$, and $\partial\mathcal{P}_2'(q, s_{23}')$ each consist of a single line segment.
6. Let $s_{2j}'(\lambda) := s_{2j}' - (\lambda, 0)$ and let $\ell_j'(\lambda)$ be the line through s_{1j}' and $s_{2j}'(\lambda)$ for $j \in \{3,4\}$. Then $\angle(\ell_3'(|pq|), \ell_4'(|pq|)) \geqslant \alpha/2$.

Note that condition 2 implies that $\text{dist}(\mathcal{P}_1', \mathcal{P}_2') = \text{dist}(\mathcal{P}_1, \mathcal{P}_2) = |pq|$, and hence inequality (2) follows from condition 3 if we manage to prove $\text{dist}(\mathcal{P}_1', \mathcal{P}_2') \geqslant f(\alpha) \cdot \text{per}(\mathcal{P}_1')$. Therefore, with a slight abuse of notation, we assume from now on that $\mathcal{P}_1$ and $\mathcal{P}_2$ satisfy the conditions in the claim, where the points s_{ij} play the role as s_{ij}' in conditions 5 and 6.

We now consider a copy of $\mathcal{P}_2$ that is translated horizontally to the left over a distance λ; see Fig. 2. Let $s_{24}(\lambda)$, $s_{23}(\lambda)$, and $q(\lambda)$ be the translated copies of s_{24}, s_{23}, and q, respectively, and let $\ell_j(\lambda)$ be the line through s_{1j} and $s_{2j}(\lambda)$ for $j \in \{3,4\}$. Furthermore, define

$$\Phi(\lambda) \;:=\; |s_{13}p| + |s_{14}p| + |s_{23}(\lambda)q(\lambda)| + |s_{24}(\lambda)q(\lambda)|$$

and

$$\Psi(\lambda) \;:=\; |s_{13}s_{23}(\lambda)| + |s_{14}s_{24}(\lambda)|.$$

Note that $\Phi(\lambda) = \Phi$ is constant. By conditions 4 and 5, we know that

$$\Phi \;\leqslant\; \Psi(0). \tag{3}$$

Note that $q(|pq|) = p$. We now apply Lemma 12 from the appendix to get

$$\Phi - \Psi(|pq|) \;\geqslant\; \sin(\delta/2) \cdot \frac{1-\cos(\delta/2)}{1+\sin(\delta/2)} \cdot (|s_{13}p| + |s_{14}p|), \tag{4}$$

where $\delta := \angle(\ell_3(|pq|), \ell_4(|pq|))$. By condition 6, we know that $\delta \geqslant \alpha/2$. The function $\delta \longmapsto \sin(\delta/2) \cdot \frac{1-\cos(\delta/2)}{1+\sin(\delta/2)}$ is increasing for $\delta \in [0, \pi]$ and hence inequality (4) also holds for $\delta = \alpha/2$.

When λ increases from 0 to $|pq|$ with unit speed, the value $\Psi(\lambda)$ decreases with speed at most 2, i.e., $\Psi(\lambda) \geqslant \Psi(0) - 2\lambda$. Using this and inequalities (3) and (4), we get

$$2|pq| \;\geqslant\; \Psi(0) - \Psi(|pq|) \;\geqslant\; \Phi - \Phi + \sin(\alpha/4) \cdot \frac{1-\cos(\alpha/4)}{1+\sin(\alpha/4)} \cdot (|s_{13}p| + |s_{14}p|),$$

and we conclude that

$$|pq| \;\geqslant\; \frac{1}{2} \cdot \sin(\alpha/4) \cdot \frac{1-\cos(\alpha/4)}{1+\sin(\alpha/4)} \cdot (|s_{13}p| + |s_{14}p|). \tag{5}$$

By the triangle inequality, $|s_{13}p| + |s_{14}p| \geqslant |s_{13}s_{14}|$. Furthermore, for a given length of $s_{13}s_{14}$, the fraction $|s_{13}s_{14}|/(|s_{14}c_{34}| + |c_{34}s_{13}|)$ is minimized when $s_{13}s_{14}$ is perpendicular to the angular bisector of ℓ_3 and ℓ_4. (Recall that c_{34} is the intersection point of the outer common tangents ℓ_3 and ℓ_4; see Fig. 2.) Hence

$$|s_{13}s_{14}| \;\geqslant\; \sin(\alpha/2) \cdot (|s_{14}c_{34}| + |c_{34}s_{13}|)\,. \tag{6}$$

We now conclude

$$\begin{aligned}
|s_{13}p| + |s_{14}p| &= \tfrac{\sin(\alpha/2)}{1+\sin(\alpha/2)} \cdot \Big(\tfrac{|s_{13}p|+|s_{14}p|}{\sin(\alpha/2)} + |s_{13}p| + |s_{14}p| \Big) \\
&\geqslant \tfrac{\sin(\alpha/2)}{1+\sin(\alpha/2)} \cdot \Big(\tfrac{|s_{13}s_{14}|}{\sin(\alpha/2)} + |s_{13}p| + |s_{14}p| \Big) && \text{by the triangle inequality} \\
&\geqslant \tfrac{\sin(\alpha/2)}{1+\sin(\alpha/2)} \cdot \Big(|s_{14}c_{34}| + |c_{34}s_{13}| + |s_{13}p| + |s_{14}p| \Big) && \text{by (6)} \\
&\geqslant \tfrac{\sin(\alpha/2)}{1+\sin(\alpha/2)} \cdot \mathrm{per}(\mathcal{P}_1),
\end{aligned}$$

where the last inequality follows because $\mathcal{P}_1$ is fully contained in the quadrilateral $s_{14}, c_{34}, x_{13}, p$. The statement (2) in the lemma now follows from (5). □

We are now ready to prove Theorem 1.

Proof of Theorem 1. If the separation angle of P_1 and P_2 is at least $\pi/6$, we are done. Otherwise, Lemma 2 gives that $\alpha > \pi/2$, and Lemma 3 gives that $\mathrm{dist}(\mathcal{P}_1, \mathcal{P}_2) \;\geqslant\; f(\pi/2) \cdot \mathrm{per}(\mathcal{P}_1) \geqslant (1/250) \cdot \min(\mathrm{per}(\mathcal{P}_1), \mathrm{per}(\mathcal{P}_2))$. □

2.2 The algorithm

Theorem 1 suggests to distinguish two cases when computing an optimal partition: the case where the separation angle is large (namely at least $\pi/6$) and the case where the separation distance is large (namely at least $c_{\mathrm{sep}} \cdot \min(\mathrm{per}(P_1), \mathrm{per}(P_2))$). As we will see, the first case can be handled in $O(n \log n)$ time and the second case in $O(n \log^4 n)$ time, leading to the following theorem.

Theorem 4. *Let P be a set of n points in the plane. Then we can compute a partition (P_1, P_2) of P that minimizes $\mathrm{per}(P_1) + \mathrm{per}(P_2)$ in $O(n \log^4 n)$ time using $O(n \log^3 n)$ space.*

To find the best partition when the separation angle is at least $\pi/6$, we observe that in this case there is a separating line whose orientation is $j \cdot \pi/7$ for some $0 \leqslant j < 7$. For each of these orientations we can scan over the points with a line ℓ of the given orientation, and maintain the perimeters of the convex hulls on both sides. This takes $O(n \log n)$ time in total; see Appendix B.

Next we show how to compute the best partition with large separation distance. We assume without loss of generality that $\text{per}(P_2) \leqslant \text{per}(P_1)$. It will be convenient to treat the case where P_2 is a singleton separately. We prove the following lemma in the appendix.

Lemma 5. *The point $p \in P$ minimizing $\text{per}(P \setminus \{p\})$ can be computed in $O(n \log n)$ time.*

It remains to compute the best partition (P_1, P_2) with $\text{per}(P_2) \leqslant \text{per}(P_1)$ whose separation distance is at least $c_{\text{sep}} \cdot \text{per}(P_2)$ and where P_2 is not a singleton. Let (P_1^*, P_2^*) denote this partition. Define the *size* of a square[1] σ to be its edge length. A square σ is a *good square* if (i) $P_2^* \subset \sigma$, and (ii) $\text{size}(\sigma) \leqslant c^* \cdot \text{per}(P_2^*)$, where $c^* := 18$. Our algorithm globally works as follows.

1. Compute a set S of $O(n)$ squares such that S contains a good square.
2. For each square $\sigma \in S$, construct a set H_σ of $O(1)$ halfplanes such that the following holds: if $\sigma \in S$ is a good square then there is a halfplane $h \in H_\sigma$ such that $P_2^* = P(\sigma \cap h)$, where $P(\sigma \cap h) := P \cap (\sigma \cap h)$.
3. For each pair (σ, h) with $\sigma \in S$ and $h \in H_\sigma$, compute $\text{per}(P \setminus P(\sigma \cap h)) + \text{per}(P(\sigma \cap h))$, and report the partition $(P \setminus P(\sigma \cap h), P(\sigma \cap h))$ that gives the smallest sum.

Step 1: Finding a good square. To find a set S that contains a good square, we first construct a set S_{base} of so-called *base squares*. The set S will then be obtained by expanding the base squares appropriately.

We define a base square σ to be *good* if (i) σ contains at least one point from P_2^*, and (ii) $c_1 \cdot \text{diam}(P_2^*) \leqslant \text{size}(\sigma) \leqslant c_2 \cdot \text{diam}(P_2^*)$, where $c_1 := 1/4$ and $c_2 := 4$ and $\text{diam}(P_2^*)$ denotes the diameter of P_2^*. Note that $2 \cdot \text{diam}(P_2^*) \leqslant \text{per}(P_2^*) \leqslant 4 \cdot \text{diam}(P_2^*)$. For a square σ, define $\overline{\sigma}$ to be the square with the same center as σ and whose size is $(1 + 2/c_1) \cdot \text{size}(\sigma)$. The following lemma is proved in the appendix.

Lemma 6. *If σ is a good base square then $\overline{\sigma}$ is a good square.*

To obtain S it thus suffices to construct a set S_{base} that contains a good base square. To this end we first build a compressed quadtree for P. For completeness we briefly review the definition of compressed quadtrees; see also Fig. 3 (left).

Assume without loss of generality that P lies in the interior of the unit square $U := [0, 1]^2$. Define a *canonical square* to be any square that can be obtained by subdividing U recursively into quadrants. A *compressed quadtree* [12] for P is a hierarchical subdivision of U, defined as follows. In a generic step of the recursive process we are given a canonical square σ and the set $P(\sigma) := P \cap \sigma$ of points inside σ. (Initially $\sigma = U$ and $P(\sigma) = P$.)

- If $|P(\sigma)| \leqslant 1$ then the recursive process stops and σ is a square in the final subdivision.
- Otherwise there are two cases. Consider the four quadrants of σ. The first case is that at least two of these quadrants contain points from $P(\sigma)$. (We consider the quadrants to be closed on the left and bottom side, and open on the right and top side, so a point is contained in a unique quadrant.) In this case we partition σ into its four quadrants—we

[1] Whenever we speak of squares, we always mean axis-parallel squares.

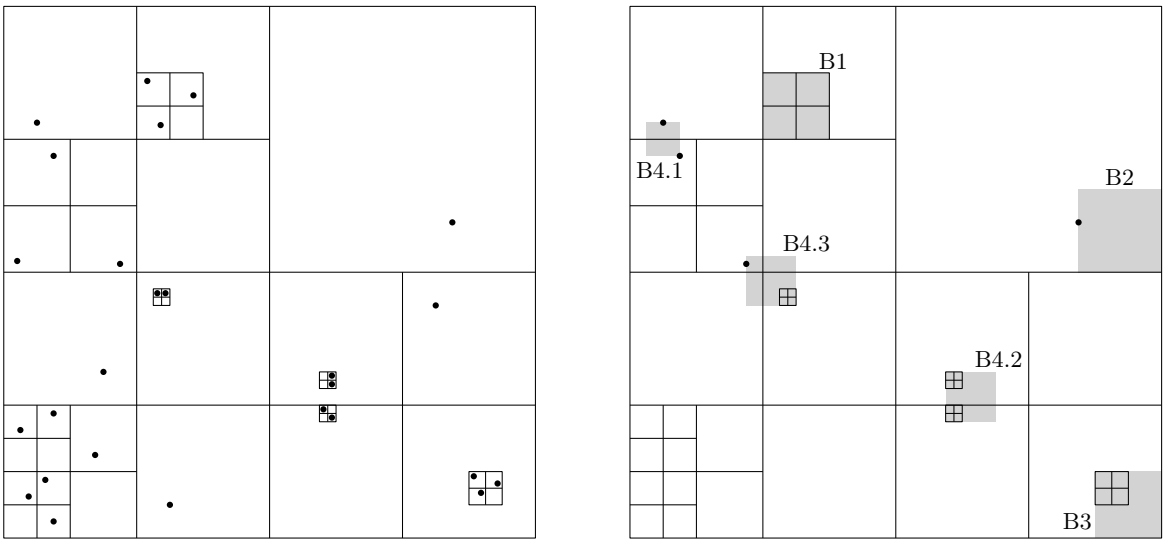


Figure 3: A compressed quadtree and some of the base squares generated from it. In the right figure, only the points are shown that are relevant for the shown base squares.

call this a *quadtree split*—and recurse on each quadrant. The second case is that all points from $P(\sigma)$ lie inside the same quadrant. In this case we compute the smallest canonical square, σ', that contains $P(\sigma)$ and we partition σ into two regions: the square σ' and the so-called *donut region* $\sigma \setminus \sigma'$. We call this a *shrinking step*. After a shrinking step we only recurse on the square σ', not on the donut region.

A compressed quadtree for a set of n points can be computed in $O(n \log n)$ time in the appropriate model of computation[2] [12]. The idea is now as follows. Let $p, p' \in P_2^*$ be a pair of points defining $\mathrm{diam}(P_2^*)$. The compressed quadtree hopefully allows us to zoom in until we have a square in the compressed quadtree that contains p or p' and whose size is roughly equal to $|pp'|$. Such a square will be then a good base square. Unfortunately this does not always work since p and p' can be separated too early. We therefore have to proceed more carefully: we need to add five types of base squares to S_{base}, as explained next and illustrated in Fig. 3 (right).

(B1) Any square σ that is generated during the recursive construction—note that this not only refers to squares in the final subdivision—is put into S_{base}.

(B2) For each point $p \in P$ we add a square σ_p to S_{base}, as follows. Let σ be the square of the final subdivision that contains p. Then σ_p is a smallest square that contains p and that shares a corner with σ.

(B3) For each square σ that results from a shrinking step we add an extra square σ' to S_{base}, where σ' is the smallest square that contains σ and that shares a corner with the parent square of σ.

[2]In particular we need to be able to compute the smallest canonical square containing two given points in $O(1)$ time. See the book by Har-Peled [12] for a discussion.

(B4) For any two regions in the final subdivision that touch each other—we also consider two regions to touch if they only share a vertex—we add at most one square to S_{base}, as follows. If one of the regions is an empty square, we do not add anything for this pair. Otherwise we have three cases.

- **(B4.1)** If both regions are non-empty squares containing points p and p', respectively, then we add a smallest enclosing square for the pair of points p, p' to S_{base}.
- **(B4.2)** If both regions are donut regions, say $\sigma_1 \setminus \sigma_1'$ and $\sigma_2 \setminus \sigma_2'$, then we add a smallest enclosing square for the pair σ_1', σ_2' to S_{base}.
- **(B4.3)** If one region is a non-empty square containing a point p and the other is a donut region $\sigma \setminus \sigma'$, then we add a smallest enclosing square for the pair p, σ' to S_{base}.

Lemma 7. *The set S_{base} has size $O(n)$ and contains a good base square. Furthermore, S_{base} can be computed in $O(n \log n)$ time.*

Proof. A compressed quadtree has size $O(n)$ so we have $O(n)$ base squares of type (B1) and (B3). Obviously there are $O(n)$ base squares of type (B2). Finally, the number of pairs of final regions that touch is $O(n)$—this follows because we have a planar rectilinear subdivision of total complexity $O(n)$—and so the number of base squares of type (B4) is $O(n)$ as well. The fact that we can compute S_{base} in $O(n \log n)$ time follows directly from the fact that we can compute the compressed quadtree in $O(n \log n)$ time [12].

It remains to prove that S_{base} contains a good base square. We call a square σ *too small* when $\text{size}(\sigma) < c_1 \cdot \text{diam}(P_2^*)$ and *too large* when $\text{size}(\sigma) > c_2 \cdot \text{diam}(P_2^*)$; otherwise we say that σ has the *correct size*. Let $p, p' \in P_2^*$ be two points with $|pp'| = \text{diam}(P_2^*)$, and consider a smallest square $\sigma_{p,p'}$, in the compressed quadtree that contains both p and p'. Note that $\sigma_{p,p'}$ cannot be too small, since $c_1 = 1/4 < 1/\sqrt{2}$. If $\sigma_{p,p'}$ has the correct size, then we are done since it is a good base square of type (B1). So now suppose $\sigma_{p,p'}$ is too large.

Let $\sigma_0, \sigma_1, \ldots, \sigma_k$ be the sequence of squares in the recursive subdivision of $\sigma_{p,p'}$ that contain p; thus $\sigma_0 = \sigma_{p,p'}$ and σ_k is a square in the final subdivision. Define $\sigma_0', \sigma_1', \ldots, \sigma_{k'}'$ similarly, but now for p' instead of p. Suppose that none of these squares has the correct size—otherwise we have a good base square of type (B1). There are three cases.

- *Case (i): σ_k and $\sigma_{k'}'$ are too large.*

  We claim that σ_k touches $\sigma_{k'}'$. To see this, assume without loss of generality that $\text{size}(\sigma_k) \leqslant \text{size}(\sigma_{k'}')$. If σ_k does not touch $\sigma_{k'}'$ then $|pp'| \geqslant \text{size}(\sigma_k)$, which contradicts that σ_k is too large. Hence, σ_k indeed touches $\sigma_{k'}'$. But then we have a base square of type (B4.1) for the pair p, p' and since $|pp'| = \text{diam}(P_2^*)$ this is a good base square.

- *Case (ii): σ_k and $\sigma_{k'}'$ are too small.*

  In this case there are indices $0 < j \leqslant k$ and $0 < j' \leqslant k'$ such that σ_{j-1} and $\sigma_{j'-1}'$ are too large and σ_j and $\sigma_{j'}'$ are too small. Note that this implies that both σ_j and $\sigma_{j'}'$ result from a shrinking step, because $c_1 < c_2/2$ and so the quadrants of a too-large square cannot be too small. We claim that σ_{j-1} touches $\sigma_{j'-1}'$. Indeed, similarly to Case (i), if σ_{j-1} and $\sigma_{j'-1}'$ do not touch then $|pp'| > \min(\text{size}(\sigma_{j-1}), \text{size}(\sigma_{j'-1}'))$, contradicting that both σ_{j-1} and $\sigma_{j'-1}'$ are too large. We now have two subcases.

  - The first subcase is that the donut region $\sigma_{j-1} \setminus \sigma_j$ touches the donut region $\sigma_{j'-1}' \setminus \sigma_{j'}$. Thus a smallest enclosing square for σ_j and $\sigma_{j'}'$ has been put into S_{base} as a base

square of type (B4.2). Let σ^* denote this square. Since the segment pp' is contained in σ^* we have

$$c_1 \cdot \text{diam}(P_2^*) \;<\; \text{diam}(P_2^*)/\sqrt{2} \;=\; |pp'|/\sqrt{2} \;\leqslant\; \text{size}(\sigma^*).$$

Furthermore, since σ_j and $\sigma'_{j'}$ are too small we have

$$\text{size}(\sigma^*) \;\leqslant\; \text{size}(\sigma_j) + \text{size}(\sigma'_{j'}) + |pp'| \;\leqslant\; 3 \cdot \text{diam}(P_2^*) \;<\; c_2 \cdot \text{diam}(P_2^*), \qquad (7)$$

and so σ^* is a good base square.

– The second subcase is that $\sigma_{j-1} \setminus \sigma_j$ does not touch $\sigma'_{j'-1} \setminus \sigma_{j'}$. This can only happen if σ_{j-1} and $\sigma'_{j'-1}$ just share a single corner, v. Observe that σ_j must lie in the quadrant of σ_{j-1} that has v as a corner, otherwise $|pp'| \geqslant \text{size}(\sigma_{j-1})/2$ and σ_{j-1} would not be too large. Similarly, $\sigma'_{j'}$ must lie in the quadrant of $\sigma'_{j'-1}$ that has v as a corner. Thus the base squares of type (B3) for σ_j and $\sigma'_{j'}$ both have v as a corner. Take the largest of these two base squares, say σ_j. For this square σ^* we have

$$c_1 \cdot \text{diam}(P_2^*) \;<\; \text{diam}(P_2^*)/2\sqrt{2} \;=\; |pp'|/2\sqrt{2} \;\leqslant\; \text{size}(\sigma^*),$$

since $|pp'|$ is contained in a square of twice the size of σ^*. Furthermore, since σ_j is too small and $|pv| < |pp'|$ we have

$$\text{size}(\sigma^*) \;\leqslant\; \text{size}(\sigma_j) + |pv| \;\leqslant\; (c_1 + 1) \cdot \text{diam}(P_2^*) \;<\; c_2 \cdot \text{diam}(P_2^*). \qquad (8)$$

Hence, σ^* is a good base square.

- *Case (iii): neither (i) nor (ii) applies.*

  In this case σ_k is too small and $\sigma'_{k'}$ is too large (or vice versa). Thus there must be an index $0 < j \leqslant k$ such that σ_{j-1} is too large and σ_j is too small. We can now follow a similar reasoning as in Case (ii): First we argue that σ_j must have resulted from a shrinking step and that σ_{j-1} touches $\sigma'_{k'}$. Then we distinguish two subcases, namely where the donut region $\sigma_j \setminus \sigma_{j-1}$ touches $\sigma'_{k'}$ and where it does not touch $\sigma'_{k'}$. The arguments for the two subcases are similar to the subcases in Case (ii), with the following modifications. In the first subcase we use base squares of type (B4.3) and in (7) the term $\text{size}(\sigma'_{j'})$ disappears; in the second subcase we use a type (B3) base square for σ_j and a type (B2) base square for p', and when the base square for p' is larger than the base square for σ_j then (8) becomes $\text{size}(\sigma^*) \leqslant 2\,|p'v| < c_2 \cdot \text{diam}(P_2^*)$.

□

Step 2: Generating halfplanes. Consider a good square $\sigma \in S$. Let Q_σ be a set of $4 \cdot c^*/c_{\text{sep}} + 1 = 18001$ points placed equidistantly around the boundary of σ. Note that the distance between two neighbouring points in Q_σ is less than $c_{\text{sep}}/c^* \cdot \text{size}(\sigma)$. For each pair q_1, q_2 of points in Q_σ, add to H_σ the two halfplanes defined by the line through q_1 and q_2. The following lemma is proved in the appendix.

Lemma 8. *For any good square $\sigma \in S$, there is a halfplane $h \in H_\sigma$ such that $P_2^* = P(\sigma \cap h)$.*

Step 3: Evaluating candidate solutions. In this step we need to compute for each pair (σ, h) with $\sigma \in S$ and $h \in H_\sigma$, the value $\text{per}(P \setminus P(\sigma \cap h)) + \text{per}(P(\sigma \cap h))$. We do this by preprocessing P into a data structure that allows us to quickly compute $\text{per}(P \setminus P(\sigma \cap h))$ and $\text{per}(P(\sigma \cap h))$ for a given pair (σ, h). Recall that the bounding lines of the halfplanes h we must process have $O(1)$ different orientations. We construct a separate data structure for each orientation.

Consider a fixed orientation ϕ. We build a data structure $\mathcal{D}_\phi$ for range searching on P with ranges of the form $\sigma \cap h$, where σ is a square and h is halfplane whose bounding line has orientation ϕ. Since the edges of σ are axis-parallel and the bounding line of the halfplanes h have a fixed orientation, we can use a standard three-level range tree [3] for this. Constructing this tree takes $O(n \log^2 n)$ time and the tree has $O(n \log^2 n)$ nodes.

Each node ν of the third-level trees in $\mathcal{D}_\phi$ is associated with a *canonical subset* $P(\nu)$, which contains the points stored in the subtree rooted at ν. We preprocess each canonical subset $P(\nu)$ as follows. First we compute the convex hull $\text{CH}(P(\nu))$. Let $v_1, \ldots, v_k$ denote the convex-hull vertices in counterclockwise order. We store these vertices in order in an array, and we store for each vertex v_i the value $\text{length}(\partial P(v_1, v_i))$, that is, the length of the part of $\partial \,\text{CH}(P(\nu))$ from v_1 to v_i in counterclockwise order. Note that the convex hull $\text{CH}(P(\nu))$ can be computed in $O(|P(\nu)|)$ from the convex hulls at the two children of ν. Hence, the convex hulls $\text{CH}(P(\nu))$ (and the values $\text{length}(\partial P(v_1, v_i))$) can be computed in $\sum_{\nu \in \mathcal{D}_\phi} O(|P(\nu)|) = O(n \log^3 n)$ time in total, in a bottom-up manner.

Now suppose we want to compute $\text{per}(P(\sigma \cap h))$, where the orientation of the bounding line of h is ϕ. We perform a range query in $\mathcal{D}_\phi$ to find a set $N(\sigma \cap h)$ of $O(\log^3 n)$ nodes such that $P(\sigma \cap h)$ is equal to the union of the canonical subsets of the nodes in $N(\sigma \cap h)$. Standard range-tree properties guarantee that the convex hulls $\text{CH}(P(\nu))$ and $\text{CH}(P(\mu))$ of any two nodes $\nu, \mu \in N(\sigma \cap h)$ are disjoint. Note that $\text{CH}(P(\sigma \cap h))$ is equal to the convex hull of the set of convex hulls $\text{CH}(P(\nu))$ with $\nu \in N(\sigma \cap h)$. Lemma 13 in the appendix thus implies that we can compute $\text{per}(P(\sigma \cap h))$ in $O(\log^4 n)$ time.

Observe that $P \setminus P(\sigma \cap h)$ can also be expressed as the union of $O(\log^3 n)$ canonical subsets with disjoint convex hulls, since $\mathbb{R}^2 \setminus (\sigma \cap h)$ is the disjoint union of $O(1)$ ranges of the right type. Hence, we can compute $\text{per}(P \setminus P(\sigma \cap h))$ in $O(\log^4 n)$ time. We thus obtain the following result, which finishes the proof of Theorem 4.

Lemma 9. *Step 3 can be performed in $O(n \log^4 n)$ time and using $O(n \log^3 n)$ space.*

3 The approximation algorithm

Theorem 10. *Let P be a set of n points in the plane and let (P_1^*, P_2^*) be a partition of P minimizing $\text{per}(P_1^*) + \text{per}(P_2^*)$. Suppose we have an exact algorithm for the minimum perimeter-sum problem running in $T(k)$ time for instances with k points. Then for any given $\varepsilon > 0$ we can compute a partition (P_1, P_2) of P such that $\text{per}(P_1) + \text{per}(P_2) \leqslant (1+\varepsilon) \cdot \big(\text{per}(P_1^*) + \text{per}(P_2^*)\big)$ in $O(n + T(n_\varepsilon))$ time, where $n_\varepsilon = O(1/\varepsilon^2)$.*

For the proof, we consider two cases and give an algorithm for each. When $\text{per}(P_1^*) + \text{per}(P_2^*)$ is small compared to the bounding box of P, we show that the exact solution can be found in linear time. Otherwise, we construct a subset $\widehat{P} \subseteq P$ of size $O(1/\varepsilon^2)$ in linear time that approximates the P with respect to the minimum perimeter-sum problem, and apply the exact algorithm to get an optimal clustering of $\widehat{P}$. Each point $p \in P \setminus \widehat{P}$ is then added to the cluster containing the representative of p in $\widehat{P}$. Let (P_1, P_2) be the resulting clustering. Our analysis yields that $\text{per}(P_1) + \text{per}(P_2) \leqslant (1+\varepsilon) \cdot \big(\text{per}(P_1^*) + \text{per}(P_2^*)\big)$. See Appendix D for more details.

References

[1] P.K. Agarwal, M. Sharir Efficient algorithms for geometric optimization. *ACM Comput. Surv.* 30(4): 412–458 (1998).

[2] T. Asano, B. Bhattacharya, M. Keil, and F. Yao. Clustering algorithms based on minimum and maximum spanning trees. In *Proc. 4th ACM Symp. Comput. Geom. (SoCG)*, pages 252–257, 1988.

[3] M. de Berg, O. Cheong, M. van Kreveld, and M. Overmars. *Computational Geometry: Algorithms and Applications (3rd edition)*. Springer-Verlag, 2008.

[4] P. Brass, C. Knauer, C.-S. Shin, M. Smid, and I. Vigan. Range-aggregate queries for geometric extent problems. In *Proc. 19th Computing: Australasian Theory Symp. (CATS)*, pages 3–10, 2013.

[5] V. Capoyleas, G. Rote, G. Woeginger. Geometric clusterings. *J. Alg.* 12(2): 341–356 (1991).

[6] T.M. Chan. More planar two-center algorithms. *Comput. Geom. Theory Appl.* 13(2): 189–198 (1999).

[7] H.G. Cho, W. Evans, N. Saeedi, and C.S. Shin. Covering points with convex sets of minimum size. In *Proc. 10th Int. Workshop Alg. Comput. (WALCOM)*, LNCS 9627, pages 166–178, 2016.

[8] T.H. Cormen, C.E. Leiserson, R.L. Rivest, and C. Stein. *Introduction to Algorithms (3rd edition)*. MIT Press, 2009.

[9] O. Devillers and M.J. Katz. Optimal line bipartitions of point sets. *Int. J. Comput. Geom. Appl.* 9(1): 39–51 (1999).

[10] Z. Drezner. The planar two-center and two-median problems. *Transportation Science* 18(4): 351–361 (1984).

[11] D. Eppstein. Faster construction of planar two-centers. In *Proc. 8th ACM-SIAM Symp. Discr. Alg. (SODA)*, pages 131–138 (1997).

[12] S. Har-Peled. *Geometric approximation algorithms*. Mathematical surveys and monographs, Vol. 173. American Mathematical Society, 2011.

[13] J. Hershberger. Minimizing the sum of diameters efficiently. *Comput. Geom. Theory Appl.* 2(2): 111–118 (1992).

[14] J.W. Jaromczyk and M. Kowaluk. An efficient algorithm for the Euclidean two-center problem. In *Proc. 10th ACM Symp. Comput. Geom. (SoCG)*, pages 303–311 (1994).

[15] D. Kirkpatrick and J. Snoeyink. Computing common tangents without a separating line. In *Proc. 4th Workshop Alg. Data Struct. (WADS)*, LNCS 955, pages 183–193, 1995.

[16] J.S.B. Mitchell and E.L. Wynters. Finding optimal bipartitions of points and polygons. In In *Proc. 2nd Workshop Alg. Data Struct. (WADS)*, LNCS 519, pages 202–213, 1991. Full version available at `http://www.ams.sunysb.edu/~jsbm/`.

[17] J. Rokne, S. Wang, and X. Wu. Optimal bipartitions of point sets. In *Proc. 4th Canad. Conf. Comput. Geom. (CCCG)*, pages 11–16, 1992.

[18] M. Segal. Lower bounds for covering problems. *J. Math. Modelling Alg.* 1(1): 17–29 (2002).

[19] M. Sharir. A near-linear algorithm for the planar 2-center problem. *Discr. Comput. Geom.* 18(2): 125–134 (1997).

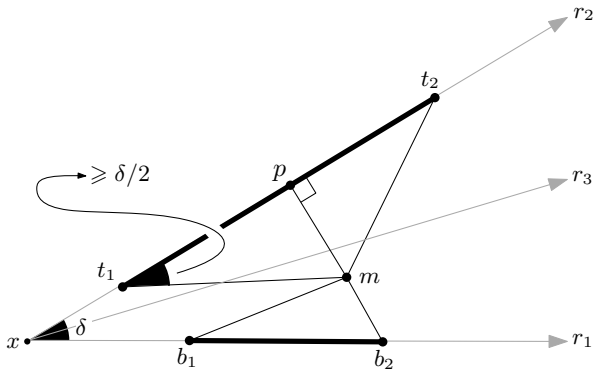


Figure 4: Illustration for Lemma 12. Φ is the total length of the four segments t_1m, t_2m, b_1m, b_2m, and Ψ is equal to the total length of the two fat segments.

A Omitted lemma and proofs in Section 2.1

Lemma 11. *Let p_0 and q be points and $\mathbf{v}$ be a unit vector. Let $p(t) := p_0 + t \cdot \mathbf{v}$ and $d(t) := |p(t)q|$ and assume that $p(t) \neq q$ for all $t \in \mathbb{R}$. Then $d'(t) = \cos(\angle(q, p(t), p(t) + \mathbf{v}))$ if the points $q, p(t), p(t) + \mathbf{v}$ make a left-turn and $d'(t) = -\cos(\angle(q, p(t), p(t) + \mathbf{v}))$ otherwise.*[3]

Proof. We prove the lemma for an arbitrary value $t = t_0$. By reparameterizing p, we may assume that $t_0 = 0$. Furthermore, by changing the coordinate system, we can without loss of generality assume that $p_0 = (0, 0)$ and $q = (x, 0)$ for some value $x > 0$.

Let $\phi := \angle((x, 0), (0, 0), \mathbf{v})$. Assume that $\mathbf{v}$ has positive y-coordinate—the case that $\mathbf{v}$ has negative y-coordinate can be handled analogously. We have proven the lemma if we manage to show that $d'(0) = -\cos\phi$. Note that since $\mathbf{v}$ has positive y-coordinate, we have $p(t) = (t\cos\phi, t\sin\phi)$ for every $t \in \mathbb{R}$. Hence

$$d(t) \;=\; \sqrt{(t\cos\phi - x)^2 + t^2\sin^2\phi}.$$

and

$$d'(t) \;=\; \frac{t - x\cos\phi}{\sqrt{t^2 - 2tx\cos\phi + x^2}}.$$

Evaluating in $t = 0$, we get

$$d'(0) \;=\; -\frac{x\cos\phi}{|x|} = -\cos\phi,$$

where the last equality follows since $x > 0$. □

The following lemma is illustrated in Fig. 4.

Lemma 12. *Let x be a point and r_1 and r_2 be two rays starting at x such that $\angle(r_1, r_2) = \delta$, and assume that $\delta \leqslant \pi$. Let $b_1, b_2 \in r_1$ and $t_1, t_2 \in r_2$ be such that $b_1 \in xb_2$ and $t_1 \in xt_2$, and let m be a point in the quadrilateral $b_1b_2t_2t_1$. Then*

$$\Phi - \Psi \;\geqslant\; \frac{(1 - \cos(\delta/2)) \cdot \sin(\delta/2)}{1 + \sin(\delta/2)} \cdot (|b_1m| + |t_1m|),$$

[3]Note that $\angle(q, p(t), p(t) + \mathbf{v}) = \angle(q, p(t), p(t) - \mathbf{v})$ by the definition of $\angle(\cdot, \cdot, \cdot)$ which is the reason that there are two cases in the lemma.

where $\Phi := |b_1m| + |t_1m| + |b_2m| + |t_2m|$ *and* $\Psi := |b_1b_2| + |t_1t_2|$.

Proof. First note that

$$|b_1m| + |b_2m| \;\geqslant\; |b_1b_2| \tag{9}$$

and

$$|t_1m| + |t_2m| \;\geqslant\; |t_1t_2|. \tag{10}$$

Let r_3 be the angular bisector of r_1 and r_2. Assume without loss of generality that m lies in the wedge defined by r_1 and r_3. Then $\angle(m, t_1, t_2) \geqslant \delta/2$.

We now consider two cases.

- *Case (A):* $|t_1m| \geqslant \frac{\sin(\delta/2)}{1+\sin(\delta/2)} \cdot (|b_1m| + |t_1m|)$.

  Our first step is to prove that

  $$|t_1m| + |t_2m| - |t_1t_2| \geqslant (1 - \cos(\delta/2)) \cdot |t_1m|. \tag{11}$$

  Let p be the orthogonal projection of m on r_2. Note that $|t_2m| \;\geqslant\; |t_2p|$. Consider first the case that p is on the same side of t_1 as x. In this case $|t_2p| \geqslant |t_1t_2|$ and therefore

  $$|t_1m| + |t_2m| - |t_1t_2| \geqslant |t_1m| \geqslant (1 - \cos(\delta/2)) \cdot |t_1m|,$$

  which proves (11).

  Assume now that p is on the same side of t_1 as t_2. In this case, we have $\angle(m, t_1, t_2) \leqslant \pi/2$ and thus $|t_1p| = \cos(\angle(m, t_1, t_2)) \cdot |t_1m| \leqslant \cos(\delta/2) \cdot |t_1m|$. Hence we have

  $$\begin{aligned} |t_1m| + |t_2m| - |t_1t_2| &\geqslant |t_1m| + |t_2p| - (|t_1p| + |t_2p|) \\ &\geqslant (1 - \cos(\delta/2)) \cdot |t_1m|, \end{aligned}$$

  and we have proved (11).

  We now have

  $$\begin{aligned} \Phi - \Psi &= |b_1m| + |t_1m| + |b_2m| + |t_2m| - |b_1b_2| - |t_1t_2| \\ &\geqslant |b_1m| + |b_2m| - |b_1b_2| + (1 - \cos(\delta/2)) \cdot |t_1m| && \text{by (11)} \\ &\geqslant (1 - \cos(\delta/2)) \cdot \tfrac{\sin(\delta/2)}{1+\sin(\delta/2)} \cdot (|b_1m| + |t_1m|) && \text{by (9)} \end{aligned}$$

  where the last step uses that we are in Case (A). Thus the lemma holds in Case (A).

- *Case (B):* $|t_1m| < \frac{\sin(\delta/2)}{1+\sin(\delta/2)} \cdot (|b_1m| + |t_1m|)$.

  The condition for this case can be rewritten as

  $$|b_1m| \;>\; \frac{1}{1 + \sin\delta/2} \cdot (|b_1m| + |t_1m|). \tag{12}$$

  To prove the lemma in this case we first argue that $\angle(b_2, b_1, m) > \pi/2$. To this end, assume for a contradiction that $\angle(b_2, b_1, m) \leqslant \pi/2$. It is easy to verify that for a given length of t_1m (and assuming $\angle(b_2, b_1, m) \leqslant \pi/2$), the fraction $|b_1m|/(|b_1m| + |t_1m|)$ is maximized when segment t_1m is perpendicular to r_2, and $m \in r_3$, and $b_1 = x$. But then

  $$\frac{|b_1m|}{|b_1m| + |t_1m|} \leqslant \frac{1}{1 + \sin\delta/2},$$

which would contradict (12). Thus we indeed have $\angle(b_2, b_1, m) > \pi/2$. Hence, $|b_2m| \geqslant |b_1b_2|$, and so $|b_1m| + |b_2m| - |b_1b_2| \geqslant |b_1m|$. We can now derive

$$\begin{array}{rcll} \Phi - \Psi & = & |b_1m| + |t_1m| + |b_2m| + |t_2m| - |b_1b_2| - |t_1t_2| & \\ & \geqslant & |b_1m| + |t_1m| + |t_2m| - |t_1t_2| & \text{by the above} \\ & \geqslant & \frac{1}{1+\sin\delta/2} \cdot \big(|b_1m| + |t_1m|\big) & \text{by (10) and (12)} \\ & \geqslant & \big(\sin(\delta/2) \cdot (1 - \cos(\delta/2))\big) \cdot \frac{1}{1+\sin\delta/2} \cdot \big(|b_1m| + |t_1m|\big) & \end{array}$$

Thus the lemma also holds in Case (B).

□

Proof of the claim in the proof of Lemma 3. We show how to modify $\mathcal{P}_1$ and $\mathcal{P}_2$ until they have all the required conditions. Of course, they already satisfy conditions 1–4. We first show how to obtain condition 5, namely that $\partial\mathcal{P}_1(s_{13}, p)$ and $\partial\mathcal{P}_1(p, s_{14})$—and similarly $\partial\mathcal{P}_2(s_{24}, q)$ and $\partial\mathcal{P}_1(q, s_{23})$—each consist of a single line segment, as depicted in Fig. 2. To this end, let v_{ij} be the intersection point $\ell_i^{\text{vert}} \cap \ell_j$ for $i \in \{1, 2\}$ and $j \in \{3, 4\}$. Let $s' \in s_{14}v_{14}$ be the point such that $\text{length}(\partial\mathcal{P}_1(p, s_{14})) = |ps'| + |s's_{14}|$. Such a point exists since

$$|ps_{14}| \;\leqslant\; \text{length}(\partial\mathcal{P}_1(p, s_{14})) \;\leqslant\; |pv_{14}| + |v_{14}s_{14}|.$$

We modify $\mathcal{P}_1$ by substituting $\partial\mathcal{P}_1(p, s_{14})$ with the segments ps' and $s's_{14}$. We can now redefine $s_{14} := s'$ so that $\partial\mathcal{P}_1(p, s_{14}) = ps_{14}$ is a line segment. We can modify $\mathcal{P}_1$ in a similar way to ensure that $\partial\mathcal{P}_1(s_{13}, p) = s_{13}p$, and we can modify $\mathcal{P}_2$ to ensure $\partial\mathcal{P}_2(s_{24}, q) = s_{24}q$ and $\partial\mathcal{P}_2(q, s_{23}) = qs_{23}$. Note that these modifications preserve conditions 1–4 and that condition 5 is now satisfied.

The only condition that $(\mathcal{P}_1, \mathcal{P}_2)$ might not satisfy is condition 6. Let $s_{2j}(\lambda) := s_{2j} - (\lambda, 0)$ and let $\ell_j(\lambda)$ be the line through $s_{2j}(\lambda)$ and s_{1j} for $j \in \{3, 4\}$. Clearly, if the slopes of ℓ_3 and ℓ_4 have different signs (as in Fig. 2), the angle $\angle(\ell_3(\lambda), \ell_4(\lambda))$ is increasing for $\lambda \in [0, |pq|]$, and condition 6 is satisfied. However, if the slopes of ℓ_3 and ℓ_4 have the same sign, the angle might decrease.

Consider the case where both slopes are positive—the other case is analogous. Changing $\mathcal{P}_2$ by substituting $\partial\mathcal{P}_2(s_{23}, s_{24})$ with the line segment $s_{23}s_{24}$ makes $\text{per}(\mathcal{P}_1) + \text{per}(\mathcal{P}_2)$ and $\text{per}(\text{CH}(\mathcal{P}_1 \cup \mathcal{P}_2))$ decrease equally much and hence condition 4 is preserved. This clearly has no influence on the other conditions. We thus assume that $\mathcal{P}_2$ is the triangle $qs_{23}s_{24}$. Consider what happens if we move s_{23} along the line ℓ_3 away from c_{34} with unit speed. Then $|s_{13}s_{23}|$ grows with speed exactly 1 whereas $|qs_{23}|$ grows with speed at most 1. We therefore preserve condition 4, and the other conditions are likewise not affected.

We now move s_{23} sufficiently far away so that $\angle(\ell_3, \ell_3(|pq|)) \leqslant \alpha/4$. Similarly, we move s_{24} sufficiently far away from c_{34} along ℓ_4 to ensure that $\angle(\ell_4, \ell_4(|pq|)) \leqslant \alpha/4$. It then follows that $\angle(\ell_3(|pq|), \ell_4(|pq|)) \geqslant \angle(\ell_3, \ell_4) - \alpha/2 = \alpha/2$, and condition 6 is satisfied. □

B The best partition with large separation angle

Define the *orientation* of a line ℓ, denoted by $\phi(\ell)$, to be the counterclockwise angle that ℓ makes with the positive y-axis. If the separation angle of P_1 and P_2 is at least $\pi/6$, then there must be a line ℓ separating P_1 from P_2 that does not contain any point from P and such that $\phi(\ell) = j \cdot \pi/7$ for some $j \in \{0, 1, \ldots, 6\}$. For each of these seven orientations we can compute the best partition in $O(n \log n)$ time, as explained next.

Without loss of generality, consider separating lines ℓ with $\phi(\ell) = 0$, that is, vertical separating lines. Let X be the set of all x-coordinates of the points in P. For any x-value $x \in X$ define $P_1(x) := \{p \in P \mid p_x \leqslant x\}$, where p_x denotes the x-coordinate of a point p, and define $P_2(x) := P \setminus P_1(x)$. Our task is to find the best partition of the form $(P_1(x), P_2(x))$ over all $x \in X$. To this end we first compute the values $\mathrm{per}(P_1(x))$ for all $x \in X$ in $O(n \log n)$ time in total, as follows. We compute the lengths of the upper hulls of the point sets $P_1(x)$, for all $x \in X$, using Graham's scan [3], and we compute the lengths of the lower hulls in a second scan. (Graham's scan goes over the points from left to right and maintains the upper (or lower) hull of the encountered points; it is trivial to extend the algorithm so that it also maintains the length of the hull.) By combining the lengths of the upper and lower hulls, we get the values $\mathrm{per}(P_1(x))$.

Computing the values $\mathrm{per}(P_2(x))$ can be done similarly, after which we can easily find the best partition of the form $(P_1(x), P_2(x))$ in $O(n)$ time. Thus the best partition with large separation angle can be found in $O(n \log n)$ time.

C Omitted lemma and proofs in Section 2.2

Proof of Lemma 5. The point p we are looking for must be a vertex of $\mathrm{CH}(P)$. First we compute $\mathrm{CH}(P)$ in $O(n \log n)$ time [3]. Let $v_0, v_1, \ldots, v_{m-1}$ denote the vertices of $\mathrm{CH}(P)$ in counterclockwise order. Let Δ_i be the triangle with vertices $v_{i-1}v_iv_{i+1}$ (with indices taken modulo m) and let P_i denote the set of points lying inside Δ_i, excluding v_i but including v_{i-1} and v_{i+1}. Note that any point $p \in P$ is present in at most two sets P_i. Hence, $\sum_{i=0}^{m} |P_i| = O(n)$. It is not hard to compute the sets P_i in $O(n \log n)$ time in total. After doing so, we compute all convex hulls $\mathrm{CH}(P_i)$ in $O(n \log n)$ time in total. Since

$$\mathrm{per}(P \setminus \{v_i\}) = \mathrm{per}(P) - |v_{i-1}v_i| - |v_iv_{i+1}| + \mathrm{per}(P_i) - |v_{i-1}v_{i+1}|,$$

we can now find the point p minimizing $\mathrm{per}(P \setminus \{p\})$ in $O(n)$ time. □

Proof of Lemma 6. The distance from any point in σ to the boundary of $\overline{\sigma}$ is at least

$$\frac{\mathrm{size}(\overline{\sigma}) - \mathrm{size}(\sigma)}{2} \;\geqslant\; \mathrm{diam}(P_2^*).$$

Since σ contains a point from P_2^*, it follows that $P_2^* \subset \overline{\sigma}$. Since $\mathrm{size}(\sigma) \leqslant c_2 \cdot \mathrm{diam}(P_2^*)$, we have

$$\mathrm{size}(\overline{\sigma}) \;\leqslant\; (2/c_1 + 1) \cdot c_2 \cdot \mathrm{diam}(P_2^*) \;=\; 36 \cdot \mathrm{diam}(P_2^*) \;\leqslant\; c^* \cdot \mathrm{per}(P_2^*).$$

□

Proof of Lemma 8. In the case where $\sigma \cap P_1^* = \emptyset$, two points in Q_σ from the same edge of σ define a half-plane h such that $P_2^* = P(\sigma \cap h)$, so assume that σ contains one or more points from P_1^*.

We know that the separation distance between P_1^* and P_2^* is at least $c_{\mathrm{sep}} \cdot \mathrm{per}(P_2^*)$, where $c_{\mathrm{sep}} = 1/250$. Moreover, $\mathrm{size}(\sigma) \leqslant c^* \cdot \mathrm{per}(P_2^*)$, where $c^* = 18$. Hence, there is an empty open stripe O with a width of at least $c_{\mathrm{sep}}/c^* \cdot \mathrm{size}(\sigma)$ separating P_2^* from P_1^*. Since σ contains a point from P_1^*, we know that $\sigma \setminus O$ consists of two pieces and that the part of the boundary of σ inside O consists of two disjoint portions B_1 and B_2 each of length at least $c_{\mathrm{sep}}/c^* \cdot \mathrm{size}(\sigma)$. Hence the sets $B_1 \cap Q_\sigma$ and $B_2 \cap Q_\sigma$ contain points q_1 and q_2, respectively, that define a half-plane h as desired. □

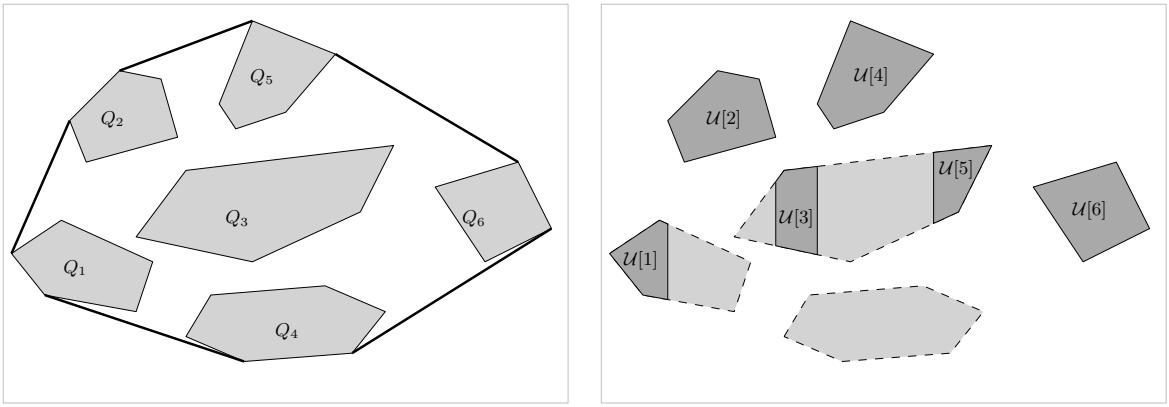


Figure 5: A collection of disjoint polygons $\mathcal{Q}$ (left) and the vertical slices in the corresponding list $\mathcal{U}$ which appear on the upper envelope (right). Note that polygon Q_3 defines two slices that contribute to the upper envelope.

Lemma 13. *Let $\mathcal{Q}$ be a set of k pairwise disjoint convex polygons with m vertices in total. Suppose each $Q \in \mathcal{Q}$ is represented by an array storing its vertices in counterclockwise order, and suppose for each vertex v_i of Q the value* $\text{length}(\partial Q(v_1, v_i))$ *is known. Let $\mathbf{Q} := \bigcup_{Q\in\mathcal{Q}} Q$. Then we can compute the perimeter of* $\textsc{ch}(\mathbf{Q})$ *in $O(k \log m)$ time.*

Proof. Any ordered pair (Q_i, Q_j) of disjoint convex polygons has two outer common tangents: the *left outer tangent*, which is the one having Q_i and Q_j on its right when directed from Q_i to Q_j, and the *right outer tangent*. The *bridge* $B(Q_i, Q_j)$ from Q_i to Q_j is the minimum-length segment q_iq_j contained in the left outer tangent of Q_i and Q_j and connecting points in Q_i and Q_j. The boundary $\partial\,\textsc{ch}(\mathbf{Q})$ consists of portions of boundaries ∂Q, where $Q \in \mathcal{Q}$, that are connected by bridges.

The *upper convex hull* of a set of points S, denoted by $\textsc{uh}(S)$, is the part of $\partial\,\textsc{ch}(S)$ from the rightmost to the leftmost point in S in counterclockwise direction. We compute a list $\mathcal{L}$ that represents $\textsc{uh}(\mathbf{Q})$. $\mathcal{L}$ consists of the polygons in $\mathcal{Q}$ having corners on $\textsc{uh}(\mathbf{Q})$ in the order they are encountered as we traverse $\textsc{uh}(\mathbf{Q})$ from left to right. We denote the length of $\mathcal{L}$ as $|\mathcal{L}|$ and the entries as $\mathcal{L}[1], \ldots, \mathcal{L}[|\mathcal{L}|]$, and do similarly for other lists. Consecutive polygons $\mathcal{L}[i], \mathcal{L}[i+1]$ should always be different, but the same polygon $Q \in \mathcal{Q}$ can appear in $\mathcal{L}$ multiple times, since several portions of ∂Q can appear on $\textsc{uh}(\mathbf{Q})$ interrupted by portions of boundaries of other polygons.

The *upper envelope* of a set of points S, denoted $\textsc{env}(S)$, is the subset $\{(x, y) \in S \mid \forall (x, y') \in S\colon y' \leqslant y\}$. In order to compute $\mathcal{L}$, we first compute $\textsc{env}(\mathbf{Q})$. Clearly, if a portion of the boundary of a polygon $Q \in \mathcal{Q}$ is on $\textsc{uh}(\mathbf{Q})$, then the same portion is also on $\textsc{env}(\mathbf{Q})$. We thus have $\textsc{uh}(\mathbf{Q}) = \textsc{uh}(\textsc{env}(\mathbf{Q}))$. The envelope $\textsc{env}(\mathbf{Q})$ can be computed with a simple sweep-line algorithm, as described next.

Define the x-range of a polygon $Q \in \mathcal{Q}$ to be the interval $I_x(Q) := [x_{\min}(Q), x_{\max}(Q)]$, where $x_{\min}(Q)$ and $x_{\max}(Q)$ denote the minimum and maximum x-coordinate of Q, respectively. For an interval $I \subseteq I_x(Q)$, define $Q[I]$ to be the intersection of Q with the vertical slab $I \times (-\infty, +\infty)$. We call $Q[I]$ a *vertical slice* of Q. Our representation of Q allows us to do the following using the algorithm described by Kirkpatrick and Snoeyink [15]: given vertical slices $Q[I]$ and $Q'[I']$, compute the bridge $B(Q[I], Q'[I'])$.

Consider the upper envelope ENV($\mathbf{Q}$). It consists of portions of the upper boundaries of the polygons in $\mathbf{Q}$. Each maximal boundary portion of some polygon Q that shows up on ENV($\mathbf{Q}$) defines a vertical slice of Q, namely the slice whose top boundary is exactly the envelope portion. We create a list $\mathcal{U}$ that stores these vertical slices in left-to-right order; see Fig. 5. Consecutive slices $\mathcal{U}[i],\mathcal{U}[i+1]$ are always from different polygons, but multiple slices from the same polygon $Q \in \mathcal{Q}$ can appear in $\mathcal{U}$, since several portions of ∂Q can appear on ENV($\mathbf{Q}$) interrupted by portions of boundaries of other polygons.

As mentioned, we will compute ENV($\mathbf{Q}$) using a sweep-line algorithm. As the sweep line ℓ moves from left to right, we maintain a data structure Σ containing all the polygons intersecting ℓ from top to bottom. Let Σ^{top} be the topmost polygon in Σ. In case Σ is empty, so is Σ^{top}. We implement Σ as a red-black tree [8]. Note that since the polygons are disjoint, the vertical order of any two polygons in Σ is invariant, and so Σ only needs to be updated when ℓ starts or stops intersecting a polygon in $\mathcal{Q}$. Thus, to find the sorted set of *events* we simply find the leftmost point L_i and the rightmost point R_i of each polygon $Q_i \in \mathcal{Q}$ and sort these points from left to right.

An event $e_j \in E$ is now handled as follows.

- If $e_j = L_i$, we insert Q_i to Σ. This requires $O(\log k)$ comparisons between Q_i and polygons currently stored in Σ, to find the position where Q should be inserted. Each such comparison can be done in $O(1)$ time since Q_i is above Q_j if and only if L_iR_i is above L_jR_j.

  If Σ^{top} changes from some polygon Q_h to Q_i, then we add the appropriate vertical slice of Q_h to $\mathcal{U}$. (This slice ends at the current position of the sweep line ℓ, and it starts at the most recent position of ℓ at which Q_h became Σ^{top}.)

- If $e_j = R_i$ then we delete Q_i from Σ in $O(\log k)$ time. If Σ^{top} was equal to Q_i before the event, we add the appropriate vertical slice of Q_i to $\mathcal{U}$.

There are $2k$ events to handle, each taking $O(\log k)$ time, so the total time used to compute $\mathcal{U}$ is $O(k \log k)$.

We now proceed to the algorithm computing the list $\mathcal{L}$ representing the upper convex hull of the vertical slices in $\mathcal{U}$. In the sequel, we think of $\mathcal{U}$ as a list of polygons with disjoint x-ranges sorted from left to right. Let $\mathcal{M}$ be a subsequence of $\mathcal{U}$, and let b_i be the bridge between $\mathcal{M}[i]$ and $\mathcal{M}[i+1]$. We say that a triple $\mathcal{M}[i-1], \mathcal{M}[i], \mathcal{M}[i+1]$ is a *valid triple* if either

(a) the right endpoint of b_{i-1} lies strictly to the left of the left endpoint of b_i, or

(b) the right endpoint of b_{i-1} coincides with the left endpoint of b_i, and b_{i-1} and b_i make a right turn.

We need the following claim.

Claim: Suppose $\mathcal{M}$ satisfies the following conditions:

(i) All triples $\mathcal{M}[i-1], \mathcal{M}[i], \mathcal{M}[i+1]$ in $\mathcal{M}$ are valid triples.

(ii) Every polygon $\mathcal{U}[i]$ that is not in $\mathcal{M}$ lies completely below one bridge b_i between consecutive polygons in $\mathcal{M}$. (Note that this condition implies that the first element in $\mathcal{M}$ is $\mathcal{U}[1]$ and the last element is $\mathcal{U}[|\mathcal{U}|]$.)

Then $\mathcal{M}$ correctly represent UH($\mathcal{U}$).

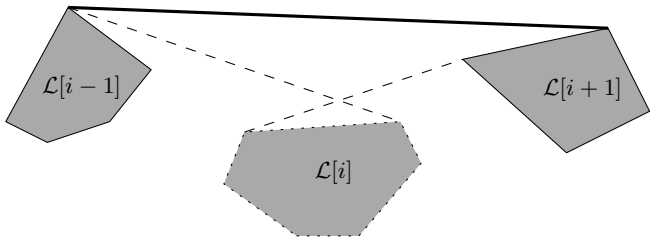


Figure 6: An invalid triple of polygons.

Proof of the Claim. Observe that condition (i), together with the definition of a valid triple, implies that the bridges between consecutive polygons in $\mathcal{U}$ together with the relevant boundary pieces—namely, for each polygon in $\mathcal{U}$ the piece of its upper boundary in between the bridges to the previous and the next polygon in $\mathcal{U}$—form a convex x-monotone chain. Hence, $\mathcal{M}$ represents the upper hull of all polygons that appear in $\mathcal{M}$. On the other hand, a polygon that does not appear in $\mathcal{M}$ cannot contribute to $\textsc{uh}(\mathcal{U})$ by condition (ii). We conclude that $\mathcal{M}$ correctly represents $\textsc{uh}(\mathcal{U})$. □

We now describe the algorithm computing $\mathcal{L}$, and we prove its correctness by showing that it satisfies the conditions from the claim.

The algorithm is essentially the same as Andrew's version of Graham's scan [3] for point sets, except that the standard right-turn check for points is replaced by a valid-triple check for polygons. Thus it works as follows. We handle the polygons from $\mathcal{U}$ to $\mathcal{L}$ one by one in order from $\mathcal{U}[1]$ to $\mathcal{U}[|\mathcal{U}|]$. To handle $\mathcal{U}[i]$ we first append $\mathcal{U}[i]$ to $\mathcal{L}$. Next, we check if the last three polygons in $\mathcal{L}$ defines a valid triple. If not, we remove the middle of the three polygons, and check if the new triple at the end of $\mathcal{L}$ is valid, remove the middle polygon if the triple is invalid, and so on. This continues until either the last triple in the list is valid, or we have only two polygons left in $\mathcal{L}$. We have then proceed to handle the next polygon, $\mathcal{U}[i+1]$.

We claim that the algorithm satisfies the following invariant: When we have added $\mathcal{U}[1], \ldots, \mathcal{U}[i]$ to $\mathcal{L}$, then $\mathcal{L}$ defines the upper convex hull $\textsc{ch}(\mathcal{U}[1, \ldots, i])$. It clearly follows from this invariant that when we have handled the last polygon in $\mathcal{U}$, then $\mathcal{L}$ correctly defines $\textsc{uh}(\mathcal{U})$.

We prove the invariant by induction. Assume therefore that it holds when we have added the polygons $\mathcal{U}[1, \ldots, i]$ to $\mathcal{L}$ and consider what happens when we add $\mathcal{U}[i+1]$ to $\mathcal{L}$. By our invalid-triple removal procedure, after we have handled $\mathcal{U}[i+1]$ all triples $\mathcal{U}[j-1], \mathcal{U}[j], \mathcal{U}[j+1]$ that remain in $\mathcal{L}$ must be valid, either because the triple was already in the list before the addition of $\mathcal{U}[i+1]$, or because it is a triple involving $\mathcal{U}[i+1]$ (in which case it was explicitly checked). Thus condition (i) is satisfied. To establish condition (ii) we only need to argue that every polygon that is removed from $\mathcal{L}$ is completely below some bridge. This is true because the middle polygon of an invalid triple lies below the bridge between the first and last polygon of the triple—see Fig. 6. Hence, the resulting list $\mathcal{L}$ satisfies conditions (ii) as well. This completes the proof of the correctness of the algorithm.

Since $\mathcal{U}$ has size $O(k)$, we need to do $O(k)$ checks for invalid triples. Each such check involves the computation of two bridges, which takes $O(\log m)$ time. Thus the whole procedure takes $O(k \log m)$ time. It is easy to compute the length of $\textsc{uh}(\mathbf{Q})$ within the same time bounds. Similarly, we can compute the lower convex hull of $\mathbf{Q}$ and its length in $O(k \log m)$ time. This finishes the proof of the lemma. □

D Omitted proof in Section 3

Proof of Theorem 10. Consider the axis-parallel bounding box B of P. Let w be the width of B and let h be its height. Assume without loss of generality that $w \geqslant h$. Our algorithm works in two steps.

- *Step 1: Check if* $\mathrm{per}(P_1^*) + \mathrm{per}(P_2^*) \leqslant w/16$. *If so, compute the exact solution.*

  We partition B vertically into four strips with width $w/4$, denoted B_1, B_2, B_3, and B_4 from left to right. If B_2 or B_3 contains a point from P, we have $\mathrm{per}(P_1^*)+\mathrm{per}(P_2^*) \geqslant w/2 > w/16$ and we go to Step 2. If B_2 and B_3 are both empty, we consider two cases.

  - *Case (i):* $h \leqslant w/8$.

    In this case we simply return the partition $(P \cap B_1, P \cap B_4)$. To see that this is optimal, we first note that any subset $P' \subset P$ that contains a point from B_1 as well as a point from B_4 has $\mathrm{per}(P') \geqslant 2 \cdot (3w/4) = 3w/2$. On the other hand, $\mathrm{per}(P \cap B_1) + \mathrm{per}(P \cap B_4) \leqslant 2 \cdot (w/2 + 2h) \leqslant 3w/2$.
  - *Case (ii):* $h > w/8$.

    We partition B horizontally into four rows with height $h/4$, numbered R_1, R_2, R_3, and R_4 from bottom to top. If R_2 or R_3 contains a point from P, we have $\mathrm{per}(P_1^*) + \mathrm{per}(P_2^*) \geqslant h/2 > w/16$, and we go the Step 2. If R_2 and R_3 are both empty, we overlay the vertical and the horizontal partitioning of B to get a 4×4 grid of cells $C_{ij} := B_i \cap R_j$ for $i, j \in \{1, \ldots, 4\}$. We know that only the corner cells $C_{11}, C_{14}, C_{41}, C_{44}$ contain points from P. If three or four corner cells are non-empty, $\mathrm{per}(P_1^*) + \mathrm{per}(P_2^*) \geqslant 6h/4 > w/16$. Hence, we may without loss of generality assume that any point of P is in C_{11} or C_{44}. We now return the partition $(P \cap C_{11}, P \cap C_{44})$, which is easily seen to be optimal.

- *Step 2: Handle the case where* $\mathrm{per}(P_1^*) + \mathrm{per}(P_2^*) > w/16$.

  The idea is to compute a subset $\widehat{P} \subset P$ of size $O(1/\varepsilon^2)$ such that an exact solution to the minimum perimeter-sum problem on $\widehat{P}$ can be used to obtain a $(1+\varepsilon)$-approximation for the problem on P.

  We subdivide B into $O(1/\varepsilon^2)$ rectangular cells of width and height at most $c := \varepsilon w/(64\sqrt{2}\pi)$. For each cell C where $P \cap C$ is non-empty we pick an arbitrary point in $P \cap C$, and we let $\widehat{P}$ be the set of selected points. For a point $p \in \widehat{P}$, let $C(p)$ be the cell containing p. Intuitively, each point $p \in \widehat{P}$ represents all the points $P \cap C(p)$. Let $(\widehat{P}_1, \widehat{P}_2)$ be a partition of $\widehat{P}$ that minimizes $\mathrm{per}(\widehat{P}_1) + \mathrm{per}(\widehat{P}_2)$. We assume we have an algorithm that can compute such an optimal partition in $T(|\widehat{P}|)$ time. For $i = 1, 2$, define

  $$P_i := \bigcup_{p \in \widehat{P}_i} P \cap C(p).$$

  Our approximation algorithm returns the partition (P_1, P_2). (Note that the convex hulls of P_1 and P_2 are not necessarily disjoint.) It remains to prove the approximation ratio.

  First, note that $\mathrm{per}(\widehat{P}_1) + \mathrm{per}(\widehat{P}_2) \leqslant \mathrm{per}(P_1^*) + \mathrm{per}(P_2^*)$ since $\widehat{P} \subseteq P$. For $i = 1, 2$, let $\widetilde{P}_i$ consist of all points in the plane (not only points in P) within a distance of at most $\sqrt{2}c$ from $\textsc{ch}(\widehat{P}_i)$. In other words, $\widetilde{P}_i$ is the Minkowksi sum of $\textsc{ch}(\widehat{P}_i)$ with a disk D of radius $\sqrt{2}c$ centered at the origin; see Fig. 7. Note that if $p \in \widehat{P}_i$, then $q \in \widetilde{P}_i$ for any $q \in P \cap C(p)$,

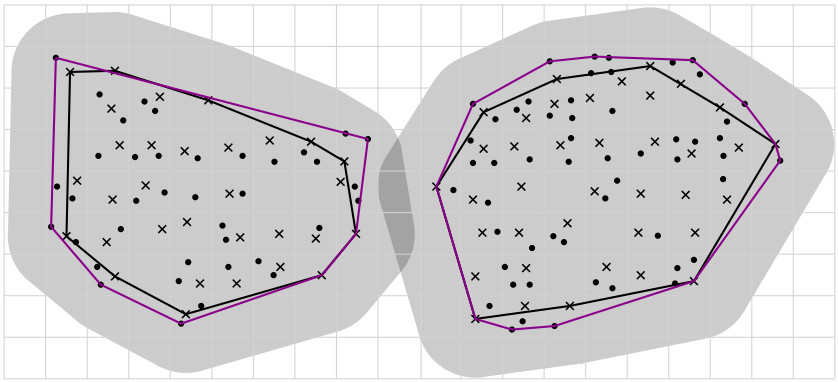


Figure 7: The crossed points are the points of $\widehat{P}$. The left gray region is $\widetilde{P}_1$ and the right gray region is $\widetilde{P}_2$. The left purple-colored polygon is the convex hull of P_1 and the right purple-colored polygon is the convex hull of P_2.

since any two points in $C(p)$ are at most $\sqrt{2}c$ apart from each other. Therefore $P_i \subset \widetilde{P}_i$ and hence $\mathrm{per}(P_i) \leqslant \mathrm{per}(\widetilde{P}_i)$. Note also that $\mathrm{per}(\widetilde{P}_i) = \mathrm{per}(\widehat{P}_i) + 2\sqrt{2}c\pi$. These observations yield

$$\begin{aligned}
\mathrm{per}(P_1) + \mathrm{per}(P_2) &\leqslant \mathrm{per}(\widetilde{P}_1) + \mathrm{per}(\widetilde{P}_2) \\
&= \mathrm{per}(\widehat{P}_1) + \mathrm{per}(\widehat{P}_2) + 4\sqrt{2}c\pi \\
&\leqslant \mathrm{per}(P_1^*) + \mathrm{per}(P_2^*) + 4\sqrt{2}c\pi \\
&= \mathrm{per}(P_1^*) + \mathrm{per}(P_2^*) + 4\sqrt{2}\pi \cdot \big(\varepsilon w/(64\sqrt{2}\pi)\big) \\
&\leqslant \mathrm{per}(P_1^*) + \mathrm{per}(P_2^*) + \varepsilon w/16 \\
&\leqslant (1+\varepsilon) \cdot (\mathrm{per}(P_1^*) + \mathrm{per}(P_2^*)).
\end{aligned}$$

As all the steps can be done in linear time, the time complexity of the algorithm is $O(n + T(n_\varepsilon))$ for some $n_\varepsilon = O(1/\varepsilon^2)$.

□