Geometric Optimization Problems on Orthogonal Polygons: Hardness Results and Approximation Algorithms

by

Saeed MehrabiDavoodabadi

A Thesis submitted to The Faculty of Graduate Studies of
The University of Manitoba
in partial fulfilment of the requirements of the degree of

DOCTOR OF PHILOSOPHY

Department of Computer Science
University of Manitoba
Winnipeg

Thesis advisor
Professor Stephane Durocher

Author
Saeed MehrabiDavoodabadi

Geometric Optimization Problems on Orthogonal Polygons: Hardness Results and Approximation Algorithms

Abstract

In this thesis, we design and develop new approximation algorithms and complexity results for three guarding and partitioning problems on orthogonal polygons; namely, guarding orthogonal polygons using sliding cameras, partitioning orthogonal polygons so as to minimize the stabbing number and guarding orthogonal terrains using vertex guards.

We first study a variant of the well-known art gallery problem in which sliding cameras are used to guard the polygon. We consider two versions of this problem: the Minimum-Cardinality Sliding Cameras (MCSC) problem in which we want to guard P with the minimum number of sliding cameras, and the Minimum-Length Sliding Cameras (MLSC) problem in which the goal is to compute a set S of sliding cameras for guarding P so as to minimize the total length of trajectories along which the cameras in S travel. We answer questions posed by Katz and Morgenstern (2011) by presenting the following results: (i) the MLSC problem is polynomially tractable even for orthogonal polygons with holes, (ii) the MCSC problem is NP-complete when P is allowed to have holes, and (iii) an $O(n)$-time exact algorithm for the MCSC problem on monotone polygons.

We then study a conforming variant of the problem of computing a partition of an orthogonal polygon P into rectangles whose stabbing number is minimum over all such partitions of P. The stabbing number of such a partition is the maximum number of rectangles intersected by any orthogonal line segment inside the polygon. In this thesis, we first give an $O(n \log n)$-time algorithm that solves this problem exactly on histograms. We then show that the problem is NP-hard for orthogonal polygons with holes, providing the first hardness result for this problem. To complement the NP-hardness result, we give a 2-approximation algorithm for the problem on both polygons with and without holes.

Finally, we study a variant of the terrain guarding problem on orthogonal terrains in which the objective is to guard the vertices of an orthogonal terrain with the minimum number of vertex guards. We give a linear-time algorithm for this problem under a directed visibility constraint.

Saeed Mehrabi
University of Manitoba
August 2015

Contents

List of Figures

Acknowledgments

First and foremost, I must thank my advisor, Dr. Stephane Durocher, without whom I could have not been able to achieve this stage of my life. Steph's brilliant ideas and suggestions during our weekly meetings over the past three years has made a lot of contributions to this thesis. He spent a lot of time helping me in writing papers and research proposals, preparing conference presentations, and completing my thesis. Steph was always available for advice and help whenever I needed it. Thank you very much, Steph!

I thank Dr. Mark Keil (University of Saskatchewan), Dr. Pak Ching Li (University of Manitoba) and Dr. Jason Morrison (University of Manitoba) very much for accepting to serve on my thesis committee, reading my thesis, and providing useful comments and suggestions on my thesis. I would also like to thank Dr. Mark de Berg for hosting me at TU Eindhoven during Summer 2013.

I would like to thank the Faculty of Graduate Studies at the University of Manitoba and the Government of Manitoba for financially supporting my research by a University of Manitoba Graduate Fellowship (UMGF) and a Manitoba Graduate Scholarship (MGS); their generous support is acknowledged.

Last but not the least, I would like to thank my family. My love and gratitude to my parents for their endless love, support and encouragement throughout my life. Thanks to my brothers, Mohammad, Ali and Abbas, for their countless helps and support during my undergraduate and graduate studies.

To my parents, my brothers, my nephew, Ali, and my niece, Ayda.

Chapter 1

Introduction

Geometric optimization problems arise in a wide range of fields such as robotics, wireless networks, image processing and engineering. Guarding and partitioning problems are among the oldest and best-known geometric problems that have motivated many theoretical studies in different areas of computational geometry.

The main focus of this thesis is to study the computational complexity of three geometric optimization problems related to guarding and partitioning of orthogonal polygons and to design efficient approximation algorithms for these problems. Our motivation for designing approximation algorithms for these problems stems from the fact that many of them are NP-hard. We say that a polynomial-time algorithm is an α-approximation algorithm for an optimization problem if the output of the algorithm is within a factor α of the optimal solution in the worst case [50].

The first problem studied in this thesis is a variant of the well-known art gallery problem on orthogonal polygon in which *sliding cameras* are used to guard the gallery. In the second problem, we consider a partitioning problem on orthogonal polygons such that the objective is to minimize the *stabbing number* of the partition. Finally, in the third problem, we study a discrete version of the art gallery problem on *orthogonal terrains*. This thesis is organized as three parts each of which studies one of the three problems. In the following, we give an overview of each of the three parts of this thesis, summarizing our contributions to each problem.

1.1 Orthogonal Art Galleries and Sliding Cameras

In Part I chapters of this thesis, we study a variant of the well-known Art Gallery Problem on orthogonal polygons. A polygon P is called orthogonal if every edge of P is either horizontal or vertical. In the standard art gallery problem, which was introduced by Victor Klee in 1973 [41], we are given the polygon P in the plane and the objective is to guard P by the minimum number of point guards. In other words, we need to find a set of point guards such that every point in P is seen by at least one of the guards, where a guard g sees a point p if and only if the segment gp is contained in P. The art gallery problem is well-known to be NP-hard [34]. It remains NP-hard even for orthogonal polygons [43]. The art gallery problem and its many variants have been studied extensively for different types of polygons (e.g., monotone polygons and polyominoes), different types of guards (e.g., points, line segments and edges) and different visibility types. See Chapter 2 for a detailed discussion of related work on the art gallery problem.

Recently, Katz and Morgenstern [27] introduced a variant of the art gallery problem in which sliding cameras are used to guard an orthogonal polygon. Let P be an orthogonal polygon with n vertices. A sliding camera travels back and forth along an orthogonal line segment s inside P. The camera can see a point $p \in P$ if there is a point $q \in s$ such that pq is a line segment normal to s that is completely inside P. In the *Minimum-Cardinality Sliding Cameras (MCSC) problem*, the objective is to guard P with the minimum number of sliding cameras, while in the *Minimum-Length Sliding Cameras (MLSC) problem* the goal is to find such a set S so as to minimize the total length of trajectories along which the cameras in S travel.

1.1.1 Our Contributions

In this thesis, we answer several questions posed by Katz and Morgenstern [27] by presenting the following results. In Chapter 3, we show that the MCSC problem is NP-hard on orthogonal polygons with holes. In Chapter 4, we show that the MLSC problem is polynomial tractable even on orthogonal polygons with holes. To this end, we give an $O(n^{2.3727})$-time exact algorithm for the MLSC problem on any orthogonal polygon P with n vertices. In Chapter 5, we give an exact $O(n)$-time dynamic programming algorithm for the MCSC problem on monotone polygons, where n is the number of the vertices of the polygon. We also show that our algorithm can be used to solve the MCSC problem on any orthogonal polygon for which the dual graph induced by the vertical decomposition of the polygon is a path.

1.2 Optimal Partitions of Orthogonal Polygons

In Part II chapters of this thesis, we study a partitioning problem on orthogonal polygons; that is, partitioning a polygonal domain into subdivisions so as to optimize a given objective, for example minimizing the number of subdivisions or the total length of the partition segments. Many variants of the problem have been studied depending on their applications and objective functions. Abam et al. [1] studied the following version of the problem: given an orthogonal polygon P, we want to compute a partition of the polygon P into rectangles such that the *stabbing number* of the resulting partition is minimum over that of all such partitions of P. A line segment inside P is orthogonal if it is horizontal or vertical. The stabbing number of a partition of P into rectangles is the maximum number of rectangles stabbed by any orthogonal line segment inside P. A partition of an orthogonal polygon P into rectangles is called *conforming* if the both endpoints of every segment of the partition lie on the boundary of P. In this thesis, we study the *Minimum Stabbing Number (MSN) problem* on orthogonal polygons, where the objective of the MSN problem is to compute a conforming partition of P whose stabbing number is minimum over that of all such partitions of P. Our motivation for studying the MSN problem is the recent work of Abam et al. [1] who gave a polynomial-time 3-approximation algorithm for the problem of computing a (not necessarily conforming) partition of orthogonal polygons into rectangles whose stabbing number is minimum over all such partitions of P.

1.2.1 Our Contributions

In Chapter 7, we show that the MSN problem is NP-hard when the polygon P is allowed to have holes, providing the first complexity result for this problem. To show NP-hardness, we describe a reduction from a variant of the 3SAT problem, called *planar variable restricted 3SAT*, to the MSN problem. The planar variable restricted 3SAT problem is a constrained version of 3SAT in which each variable can appear in at most three clauses and the corresponding *variable-clause graph* must be planar. In Chapter 8, we first give a polynomial-time 2-approximation algorithm for the MSN problem on any orthogonal polygon P using a linear programming (LP) formulation of the problem. A *histogram* H is a simple orthogonal polygon whose boundary contains some edge, called the *base*, whose length is equal to the sum of the lengths of the edges of H that are parallel to the base; the base edge is not necessarily on the boundary of the union. We give an $O(n \log n)$-time algorithm that solves the MSN problem exactly when P is a histogram, where n is the number of the vertices of P. The algorithm relies on constructing a tree structure T associated with polygon P in such a way that the stabbing number of a partition of P corresponds to some properties of T.

1.3 Guarding Orthogonal Terrains

In Part III chapters of this thesis, we study a discrete variant of the terrain guarding problem on orthogonal terrains. A *terrain* is an x-monotone polygonal chain in the plane; a terrain T is orthogonal if every edge of T is either horizontal of vertical. In the continuous terrain guarding problem, the objective is to find the minimum number of point guards on T such that every point on T is guarded. In the discrete terrain guarding problem, on the other hand, we are given two sets G and S of points on T and the objective is to guard every point in S with the minimum number of point guards in G. King and Krohn [30] showed that both continuous and discrete versions of the terrain guarding problem are NP-hard on arbitrary terrains. Their proof does not immediately imply the hardness of the problem on orthogonal terrains. Gibson et al. [20] and Friedrichs et al. [18] gave a polynomial-time approximation scheme (PTAS) for the discrete and continuous versions of the terrain guarding problem, respectively. In this thesis, we study a variant of the discrete terrain guarding problem on *orthogonal* terrains, called the *Orthogonal Terrain Guarding (OTG) problem*, in which the objective is to guard the vertices of an orthogonal terrain with the minimum number of vertex guards.

1.3.1 Our Contributions

In this thesis, we show that the OTG problem can be solved in polynomial time on orthogonal terrains if we restrict the visibility among the adjacent vertices of the terrain. In Chapter 10, we consider several instances of the OTG problem, depending on the type of the visibility constraints between the adjacent vertices, and show several are polynomial tractable. For instance, considering three vertices u, v and w of an orthogonal terrain T such that u is a convex vertex with neighbours v and w, we give a linear-time algorithm that solves the OTG problem under the assumption that u can only see v and neither v nor w can see u. We call this variant of the problem, the *Directed Terrain Guarding (DTG) problem*. Our algorithms are based on reducing the DTG problem to two independent subproblems and then solving each subproblem using a greedy algorithm.

Remark

Part of the work presented in this thesis has already been published in the following conference proceedings:

1. Stephane Durocher, Pak Ching Li, and Saeed Mehrabi. "Guarding orthogonal terrains." To appear in *proceedings of the 27th Canadian Conference on Computational Geometry (CCCG 2015), Kingston, Canada*. 2015.

2. Mark de Berg, Stephane Durocher, and Saeed Mehrabi. "Guarding monotone art galleries with sliding cameras in linear time". In *proceedings of the 8th Annual International Conference on Combinatorial Optimization and Applications (COCOA 2014), Maui, USA*. Springer Lecture Notes in Computer Science. 8881:113–125 2014.

3. Stephane Durocher, and Saeed Mehrabi. "A 3-approximation algorithm for guarding orthogonal art galleries with sliding cameras". In *proceedings of the 25th International Workshop on Combinatorial Algorithms (IWOCA 2014), Duluth, USA*. Springer Lecture Notes in Computer Science. 8986:1–13. 2014.

4. Stephane Durocher, Omrit Filtser, Robert Fraser, Ali Mehrabi, and Saeed Mehrabi. "A (7/2)-approximation algorithm for guarding orthogonal art gal-

leries with sliding cameras". In *proceedings of the 11th Latin American Theoretical Informatics Symposium (LATIN 2014), Montevideo, Uruguay*. Springer Lecture Notes in Computer Science. 8392: 294–305. 2014.

5. Stephane Durocher and Saeed Mehrabi. "Guarding orthogonal art galleries using sliding ameras: algorithmic and hardness results". In *proceedings of the 38th International Symposium on Mathematical Foundations of Computer Science (MFCS 2013), Vienna, Austria*. Springer Lecture Notes in Computer Science. 8087:314–324. 2013.

6. Stephane Durocher and Saeed Mehrabi. "Computing partitions of rectilinear polygons with minimum stabbing number". In *proceedings of the 18th International Computing and Combinatorics Conference (COCOON 2012), Sydney, Australia*. Springer Lecture Notes in Computer Science. 7434:228–239. 2012.

Part I

Orthogonal Art Galleries and Sliding Cameras

Chapter 2

Background

Let P be a polygonal region in the plane. Two points p and q in P see each other if the line segment connecting p and q lies entirely inside or on the boundary of P. The classic *Art Gallery Problem* asks for computing the minimum number $G(n)$ such that, for any polygon P with n vertices, there exists a set of $G(n)$ points that collectively see polygon P entirely; we refer to the points in $G(n)$ as *guards* and say that they *guard* P. The art gallery problem has been studied in two main streams: one stream focuses on finding lower and upper bounds on $G(n)$, while in the other stream the optimization version of the problem has been studied. In this chapter, we give an overview of the related work on the art gallery problem that is most relevant to this thesis. See the surveys by O'Rourke [41] or Urrutia [48] for a history of the art gallery problem.

2.1 Lower and Upper Bounds

In this section, we describe the related work on computing lower and upper bounds on $G(n)$, the number of point guards that is sufficient and necessary to guard P.

A polygon is called *simple* if it has no holes and its boundary has no self-intersecting point. The art gallery problem was first posed on simple polygons by Victor Klee in 1973 [39]. Chvátal [7] was the first to answer Klee's art gallery question by giving an upper bound proving that $\lfloor n/3 \rfloor$ point guards are always sufficient and sometimes necessary to guard a simple polygon with n vertices. Later, Chvátal's proof was simplified by Fisk [17]. This upper bound on the guard number is also tight, because occasionally $\lfloor n/3 \rfloor$ guards are necessary [7]: see Figure 2.1(a). For polygons with holes Fisk's proof no longer holds because the dual graph of the triangulation is not a tree. Hoffmann et al. [24] proved that $\lfloor (n+h)/3 \rfloor$ guards are always sufficient and occasionally necessary, where h is the number of holes.

If we restrict the art gallery problem to *orthogonal polygons* (i.e., polygons with only vertical and horizontal edges), then we can obtain better bounds. The structure of orthogonal polygons allows us to apply better partitions than triangulations. By partitioning the polygon into convex quadrilaterals, Kahn et al. [25] proved that $\lfloor n/4 \rfloor$ guards are always sufficient and sometimes necessary to guard the interior of a simple orthogonal polygon with n vertices. The same bound was also reported independently by O'Rourke [39] using L-shape decomposition of orthogonal polygons. Figure 2.1(b) shows that the upper bound $\lfloor n/4 \rfloor$ is tight.

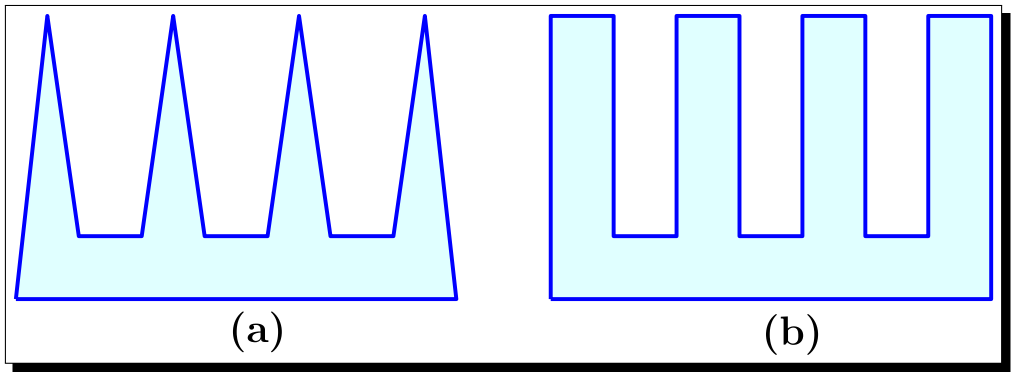

Figure 2.1: (a) A simple polygon with $n = 3k$ vertices that requires k guards [7]. (b) An orthogonal polygon with $n = 4k$ vertices that requires k guards [25].

If the orthogonal polygon P is allowed to have holes, then Hoffmann [23] showed that the same upper bound of $\lfloor n/4 \rfloor$ guards holds, assuming that guards can be positioned any where inside the polygon. However, if guards are required to lie on the vertices of the polygon, then there is no tight bound known. O'Rourke [41] proved that $\lfloor (n + 2h)/4 \rfloor$ vertex guards are suffiecient. Moreover, the Shermer's conjecture [45, 41] states that $\lfloor (n + h)/4 \rfloor$ guards are always sufficient and sometimes necessary to guard an orthogonal polygon with n vertices and h holes. Zylinski [52] proved that if there exists a quadrilateralization of P whose dual graph is a *cactus*, then $\lfloor (n + h)/4 \rfloor$ guards are always sufficient to guard P. A cactus is a connected graph with the property that any two of its cycles share at most one vertex and any such vertex is a cut vertex.

Avis and Toussaint [2] introduced the notion of *mobile guards* in art galleries: a mobile guard is a point guard that travels back and forth either along an edge of the polygon or along a straight line segment that is contained in the polygon. In the mobile guard problem, it is not required for every point in the polygon to be *constantly* guarded; instead, we only need every point to be guarded by at least one guard during its walk. They conjectured that except for a small number of polygons, $\lfloor n/4 \rfloor$ edge guards are sufficient to guard a polygon; this conjecture is still open. O'Rourke [40] proved that $\lfloor n/4 \rfloor$ diagonal guards (i.e., guards that are allowed to move along a diagonal) are sufficient for guarding general polygons. For orthogonal polygons, O'Rourke [41] showed that $\lfloor (3n+4)/16 \rfloor$ mobile guards are sufficient. Finally, Bjorling-Sachs and Souvaine [5] proved that $\lfloor (n-2)/5 \rfloor$ edge guards are sufficient for monotone polygons.

2.2 Minimizing Guards

The other stream studying the art gallery problem focuses on the complexity of the minimization version of this problem, where the objective is to guard the interior of a polygonal domain with the minimum number of guards.. Lee and Lin [34] showed that the art gallery problem is NP-hard on simple polygons. Moreover, the problem is also NP-hard on simple orthogonal polygons [43] and it remains NP-hard even for monotone polygons [33]. Eidenbenz et al. [13] proved that the art gallery problems is APX-hard on simple polygons. Moreover, they showed that if the polygon is allowed to have holes, then the problem cannot be approximated by a polynomial-time algorithm with factor $((1-\epsilon)/12)\ln n$, for any $\epsilon > 0$, where n is the number of the vertices of the polygon.

King [29] gave an $O(n^3)$-time $(\log\log OPT)$-approximation algorithm for the art gallery problem on arbitrary simple polygons, where OPT denotes the size of an optimal solution. Ghosh [19] gave an $O(n^5)$-time $(\log n)$-approximation algorithm for the problem on polygons with holes. Moreover, King [29] gave an $O((h+1)^2 n^3)$-time approximation algorithm with approximation factor $((1+\log(h+1))\log OPT)$ for the problem on any arbitrary polygon with h holes, where OPT is the size of an optimal solution; note that this is a faster algorithm than that of Ghosh for $h = o(n)$. Krohn and Nilsson [33] gave a constant-factor approximation algorithm for the problem on monotone polygons. They also gave a polynomial-time algorithm for the art gallery problem on orthogonal polygons that computes a solution of size $O(OPT^2)$, where OPT is the cardinality of an optimal solution.

Due to inapproximability results of the standard art gallery problem, many variants of the problem have been studied over the past decades, where limited visibility allows the problem to be solved in polynomial time [38, 3, 36, 51]. Motwani et al. [36] studied the art gallery problem under s-visibility, where a guard point $p \in P$ can see all points in P that can be connected to p by an orthogonal staircase path contained in P. They use a perfect graph approach to solve the problem in polynomial time. Worman and Keil [51] defined r-visibility, in which a guard point $p \in P$ can see all points $q \in P$ such that the bounding rectangle of p and q (i.e., the axis-parallel rectangle with diagonal pq) is contained in P. Given that P has n vertices, they use a similar approach to Motwani et al. [36] to solve this problem in $\widetilde{O}(n^{17})$ time, where the notation $\widetilde{O}()$ omits poly-logarithmic factors. Moreover, Lingas et al. [35] presented a linear-time 3-approximation algorithm for this problem.

2.3 Sliding Cameras

Katz and Morgenstern [27] introduced *sliding cameras* for guarding orthogonal polygons. A sliding camera is a point guard that travels along a horizontal or vertical line segment s inside an orthogonal polygon P, and it can see a point $p \in P$ if the there is a point $q \in s$ such that pq is a line segment normal to s that is completely inside P. Recall that in the MCSC problem, the objective is to guard P with the minimum number of sliding cameras, while in the MLSC problem we wish to guard P so as to minimize the total length of line segments along which cameras travel. We assume that in both variants of the problem, polygon P and sliding cameras are all constrained to be orthogonal. Note that in both variants every point inside P must be visible to at least one camera at some point along its trajectory. See Figure 2.2 for an illustration of these variants.

Katz and Morgenstern [27] first considered a restricted version of the MCSC problem, where cameras are constrained to travel only vertically inside the polygon. By a similar approach to Motwani et al. [36], they construct a graph G corresponding to a polygon P and then show that (i) solving this problem on P is equivalent to solving the minimum clique cover problem on G, and that (ii) G is chordal. Since the minimum clique cover problem is polynomial-time solvable on chordal graphs, they solve the guarding problem in polynomial time for vertical cameras. They also generalized the problem such that both vertical and horizontal cameras are allowed; they gave a 2-approximation algorithm for the MCSC problem under the assumption that the given input is an x-monotone orthogonal polygon. They leave open the complexity of the MCSC and the MLSC problems as

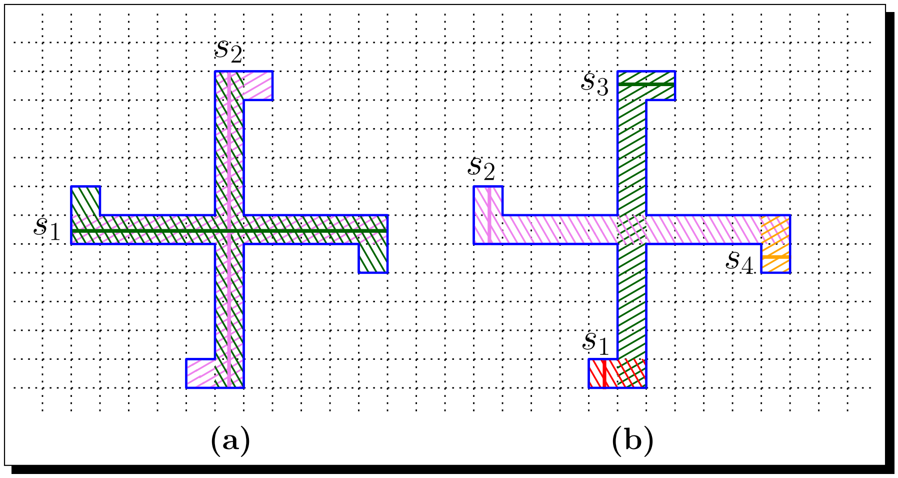

Figure 2.2: An illustration of the MCSC and MLSC problems. Each grid cell has size 1×1. (a) The set of two sliding cameras s_1 and s_2 as an optimal solution for the MCSC problem on P; each shaded region indicates the visibility region of the corresponding camera. (b) A set of four sliding cameras whose total length is 6, which is an optimal solution for the MLSC problem on P.

well as giving approximation algorithms for these problems (in case they are NP-hard). Seddighin [44] considered the sliding cameras problem under k-visibility, where a camera's line of sight can pass through k edges of the polygon. He first proved that the MLSC problem is NP-hard under k-visibility, for any fixed $k \geq 2$, and then gave a 2-approximation algorithm for this problem.

A *double-sided histogram* is the union of two histograms that share a common base edge and that are located on opposite sides of the base. The MCSC problem is equivalent to the problem of covering P with the minimum number of double-sided histograms, since the visibility region of a sliding camera is exactly a double-sided histogram. Fekete and Mitchell [16] proved that partitioning an orthogonal polygon (possibly with holes) into a minimum number of histograms is NP-hard. Note that in general, the fact that an orthogonal polygon can be covered by k histograms does not imply that it can be partitioned by k histograms. Therefore, their proof does not directly imply that covering an orthogonal polygon with minimum number of double-sided histograms is NP-hard, leaving open the question of whether the MCSC problem is also NP-hard for orthogonal polygons with holes.

Chapter 3

Hardness Result

In this chapter, we show that the MCSC problem is NP-complete when P is allowed to have holes. We first define the problem formally.

MCSC WITH HOLES

INPUT: An orthogonal polygon P, possibly with holes and an integer k.

OUTPUT: YES, if there exist k orthogonal line segments inside P that guard P entirely; NO, otherwise.

Given a candidate solution for the MCSC WITH HOLES problem, we can verify the solution in polynomial time by checking whether the union of the visibility regions of cameras in the solution is P. Therefore, the problem is in NP. We show NP-hardness by a reduction from the *minimum hitting of horizontal unit segments* problem, which we call the MIN SEGMENT HITTING problem, defined as follows [21]:

MIN SEGMENT HITTING

INPUT: n pairs (a_i, b_i), $i = 1, \ldots, n$, of integers and an integer k.

OUTPUT: YES, if there exist k orthogonal lines $l_1, \ldots, l_k$ in the plane, i.e., for each i, l_i is horizontal or vertical, such that each line segment $[(a_i, b_i), (a_i + 1, b_i)]$ is hit by at least one of the lines; NO, otherwise.

Hassin and Megiddo [21] proved that the MIN SEGMENT HITTING problem is NP-complete. Let I be an instance of the MIN SEGMENT HITTING problem, where I is a set of n horizontal unit-length segments with integer coordinates. We construct an orthogonal polygon P (with holes) such that there exists a set of k orthogonal lines that hit the segments in I if and only if there exists a set C of $k + 4$ orthogonal line segments inside P that collectively guard P. Throughout this section, we refer to the segments in I as *unit segments* and to the segments in C as *line segments*.

3.1 Gadgets

We first observe that any two unit segments in I can share at most one point, which must be a common endpoint of the two unit segments. For each unit segment $s_i \in I$, $1 \leq i \leq n$, we denote the left endpoint of s_i by (a_i, b_i) and, therefore, the right endpoint of s_i is $(a_i + 1, b_i)$. Let $L(s_i)$ denote the set of unit segments in I for which the x-coordinate of their left endpoints is equal to a_i. Moreover, let $N(s_i)$ denote the set of unit segments in I that have at least one endpoint with x-

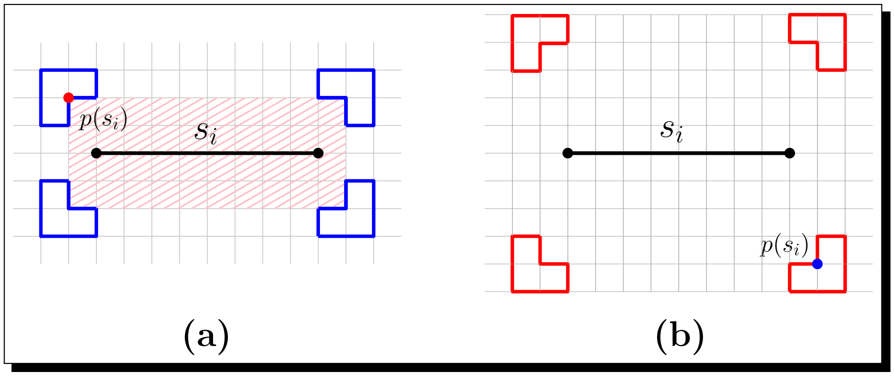

Figure 3.1: The L-holes associated with a line segment $s_i \in I$, where (a) a_i is even, and (b) a_i is odd.

coordinate equal to a_i or $a_i + 1$. Our reduction refers to an L-hole, which we define as an orthogonal polygon with six vertices such that exactly one of them is reflex. We constrain each grid cell to have size $\frac{1}{12} \times \frac{1}{12}$. The L-holes have variable size; in order to specify the size of L-holes, we first need the following notation.

Let $I = \{O_1, O_2, \ldots, O_r\}$ be a partition of I such that the left endpoints of all the unit segments in O_m, for each $1 \leq m \leq r$, have the same x-coordinate. Consider the set O_m, for some $1 \leq m \leq r$, and let $|O_m| = t$. The idea is to associate exactly four L-holes for each unit segment $s \in O_m$ depending on t and the parity of the x-coordinate of the left endpoint of s.

Case 1. $t = 1$. Let s_i be the only unit segment in O_m. If a_i is even, then Figure 3.1(a) shows the L-holes associated with s_i. If a_i is odd, then Figure 3.1(b) shows the L-holes associated with s_i. In both cases, each L-hole has height and width of $1/6$. In the case a_i is odd, the L-holes are located such that the vertical distance between any point on an L-hole and s_i is at least $1/3$. Note the red vertex on the upper left L-hole of s_i in Figure 3.1(a) and the blue vertex on the lower right L-hole of s_i in

Figure 3.1(b); we call this vertex the *visibility vertex* of s_i, which we denote $p(s_i)$. The L-holes associated with s_i do not interfere with the L-holes associated with the line segments in $N(s_i)$ because for any unit segment $s_j \in N(s_i)$ the vertical distance d between s_i and s_j is either zero or at least one. If $d \geq 1$, then it is trivial that the L-holes of s_i do not interfere with those of s_j. Now, suppose that s_i and s_j share a common endpoint; that is $d = 0$. Since s_i and s_j have unit lengths a_i and a_j have different parities and, therefore, the L-holes associated with s_i and s_j do not interfere with each other. Figure 3.2 shows an example of such two unit segments and their corresponding L-holes.

Case 2. $t \geq 2$. Let $\{s_{i_1}, s_{i_2}, \ldots, s_{i_t}\}$ be the unit segments in O_m ordered from top to bottom. We associate each unit segment in O_m with four L-holes similar to Case 1: two left L-holes around its left endpoint and two right L-holes around its right endpoint. Here, each L-hole has height and width equal to $\frac{1}{6t}$; note that the size of L-holes remain polynomial in the size of the input. Informally, the idea is to locate the L-holes in such a way that the L-holes of every two unit segments in O_m do not block the vertical visibility of their horizontal edges. To this end, consider the vertical slab of size two grid units to the left of the unit segments in O_m; see Figure 3.3(a) for an illustration. We first partition the slab into t vertical subslabs. Consider the subslabs from left to right. Then, we consider the unit segments in O_m from top to bottom and locate their left L-holes in separate subslabs from left to right; see Figure 3.3(a). Note that the left L-holes of *each* unit segment in O_m lie in the same slab (i.e., they are aligned with each other). Next, we locate the right L-holes of the unit segments in O_m analogously by considering the vertical

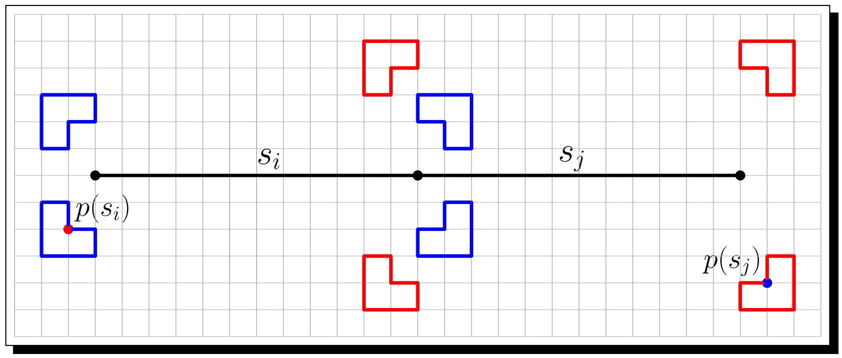

Figure 3.2: An illustration of the L-holes associated with two line segments s_i and s_j in I that share a common endpoint such that a_i is even and $L(s_i) = \emptyset$.

slab of size two grid units to the right of the unit segments in O_m, and then partitioning the slab into t vertical subslabs. Then, consider the subslabs from right to left: we locate the right L-holes of the unit segments in O_m from top to bottom in separate subslabs from *right to left* (as opposed to the left L-holes that were located in separate subslabs from *left to right*). See Figure 3.3(b) for an illustration. Our construction ensures the following observation:

Observation 3.1.1. *Let s be a unit segment in O_m. Then, the visibility region of the topmost horizontal edge of the upper left L-hole (resp., the lowest horizontal edge of the lower left L-hole) of s is unbounded from above (resp., from below); i.e., it is not blocked by any other L-hole. Similarly, the visibility region of the topmost horizontal edge of the upper right L-hole (resp., the lowest horizontal edge of the lower right L-hole) of s is unbounded from above (resp., from below). See the shaded regions in Figure 3.3 for an example.*

Observation 3.1.1 holds even if the unit segments in Case 2 share a common endpoint; this is illustrated in Figure 3.4. We now describe the reduction.

3.2 Reduction

Given an instance I of the MIN SEGMENT HITTING problem, we first associate each unit segment in $s_i \in I$ with four L-holes as described above. After adding the corresponding L-holes, we enclose I in a rectangle such that all unit segments and the L-holes associated with them lie in its interior. Finally, we create four small rectangles each of which is located on one corner of the bigger rectangle (as shown in Figure 3.5) such that any orthogonal line that passes through one of the smaller rectangles cannot intersect any of the unit segments in I; note the *visibility vertex* of each smaller rectangle shown in red. Moreover, there exists a dent on the enterance to each smaller rectangle to ensure that no orthogonal line segment in P can see more than one visibility vertex of the smaller rectangles. See Figure 3.5 for a complete example of the reduction. Let P be the resulting orthogonal polygon. We now show the following lemma.

Lemma 3.2.1. *There exist k orthogonal lines such that each unit segment in I is hit by one of the lines if and only if there exists $k + 4$ orthogonal line segments inside P that collectively guard P.*

Proof. $(\Rightarrow)$ Suppose that there exists a set S of k lines such that each unit segment in I is hit by at least one line in S. We can assume that every line $L \in S$ hits at least one unit segment in I; otherwise, we can remove L from S without affecting the feasibility of S. Let $L \in S$ and let $L_P = L \cap P$. If L is horizontal, then L, and therefore L_P, do not cross the interior of any L-hole inside P. Similarly, if L is vertical and passes through an endpoint of some unit segment(s) in I, then neither

L nor L_P passes through the interior of any L-hole in P.[1] Now, suppose that L is vertical and passes through the interior of some unit segment $s \in I$. Translate L_P horizontally such that it passes through the midpoint of s. Since unit segments have endpoints on adjacent integer grid point, L_P still crosses the same set of unit segments of I as it did before this move. Moreover, this ensures that L_P does not cross any L-hole inside P. Consider the set $S' = \{L_P \mid L \in S\}$.

The line segments in S' cannot guard the visibility vertex of any of the smaller rectangles. Moreover, if all line segments in S' are vertical or all are horizontal, then they cannot collectively guard the bigger rectangle entirely. Next, we add four orthogonal line segments into S' each of which guards one of the smaller rectangles entirely. We choose each of these line segments in such way that it guards one smaller rectangle and hits the dent that is on the other corner of the bigger rectangle; see the red dashed line segments in Figure 3.5. We call these four line segments the *boundary guards*. Let S'' be the union of S' and the boundary guards. Clearly, $|S''| = k + 4$. We now show that S'' is a feasible solution for the MCSC problem on P. Let p be a point in P. If p is inside one of the smaller rectangles, then it must be guarded by one of the boundary guards. Recall the vertical subslabs obtained by dividing the vertical slabs when locating the L-holes; note that each of such subslabs contains exactly two L-holes.

- Suppose that p lies in one of the subslabs. If it is between the L-holes, then p is guarded by the line segment in S' whose corresponding line intersects s. Otherwise, by Observation 3.1.1, one of the horizontal boundary guards

[1] Note that it is possible for L to pass through the boundary of some L-hole.

must see p.

- Now, suppose that p does not lie on a vertical subslab. Then, if p lies inside the rectangle induced by the reflex vertices of the L-holes associated with s (i.e., the shaded rectangle in Figure 3.1(a) for instance), for some unit segment $s \in I$, then p is guarded by the line segment in S' whose corresponding line intersects s. This is true even for two unit segments with a common endpoint because then the L-holes associated with such two unit segments have different heights and so they do not block the visibility. If p is not interior to any of such rectangles, then it is seen by the boundary of the polygon in at least one orthogonal direction and, therefore, one of the boundary guards sees p.

Therefore, the set S'' is a feasible solution for the MCSC problem on P such that $|S''| = k + 4$.

($\Leftarrow$) Now, suppose that there exists a set M of $k + 4$ orthogonal line segments contained in P that collectively guard P. For each line segment $c \in M$, let L_c denote the line induced by c. We now describe how to find k lines that form a solution to instance I by moving the line segments in M accordingly such that each unit segment in I is hit by at least one of the corresponding lines. From the construction of polygon P, no line segment in M can see more than one visibility vertex of the smaller rectangles. Thus, let M' be the set of the four line segments in M each of which guards a visibility vertex of a smaller rectangle. We know that no line segment in M' can see $p(s)$, for all $s \in I$. Therefore, for each unit segment $s \in I$ in order, consider a line segment $\ell \in M \setminus M'$ that guards $p(s)$; let ℓ' be the maximal line segment inside P that is aligned with ℓ. Note that ℓ' must intersect

R, the rectangle induced by the reflex vertices of the L-holes associated with unit segment s (see the shaded rectangle in Figure 3.1 for an example). If ℓ' is horizontal and $L_{\ell'}$ does not align s, then move ℓ' accordingly up or down until it aligns with s. Thus, $L_{\ell'}$ is a line that hits s. Now, suppose that ℓ' is vertical. If ℓ' intersects s, then $L_{\ell'}$ also intersects s. If ℓ' does not intersect s, then the endpoints of ℓ' must be on the boundary of two of the L-holes associated with s; this is because the only way for a maximal line segment to see $p(s)$ is to intersect R and, therefore, either intersect s or be bounded by the L-holes of s. Thus, we move ℓ' horizontally to the left or to the right until it hits s. Therefore, $L_{\ell'}$ is a line that hits s after this move.

Therefore, we have obtained exactly one line from each line segment in $M \setminus M'$ such that each unit segment in I is hit by at least one of the lines. This completes the proof of the lemma. □

By Lemma 3.2.1 we obtain the main result of this section:

Theorem 3.2.2. *The* MCSC WITH HOLES *is* NP-*complete.*

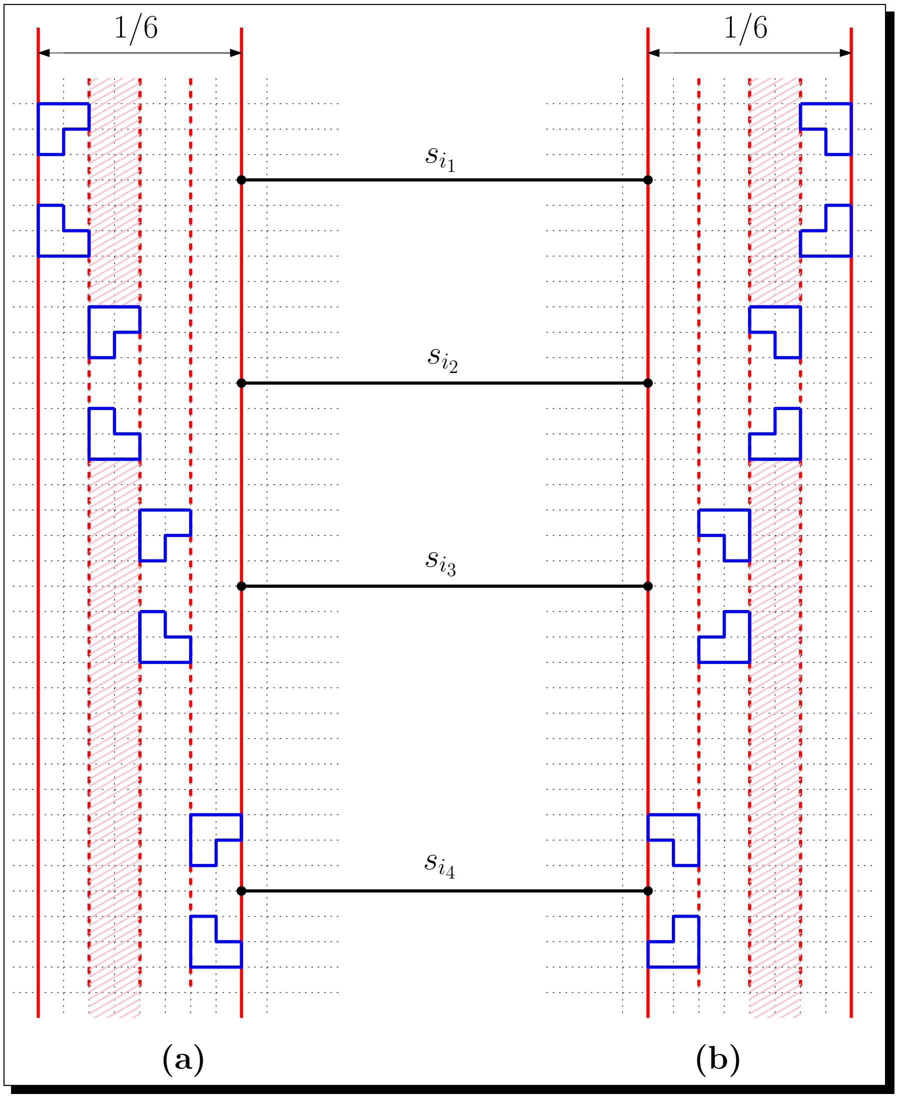

Figure 3.3: The unit segments in O_m, where $|O_m| = 4$, and the associated L-holes. (a) The left slab (shown as the two solid vertical red lines) is divided into four subslabs (separated by the three dashed vertical red lines) and the left L-holes of the unit segments in O_m (ordered from top to bottom) are located from left to right in separate subslabs. (b) The right slab is divided into four subslabs and the right L-holes of the unit segments in O_m (ordered from top to bottom) are located from right to left in separate subslabs. Note the *unbounded* shaded regions below and above the L-holes associated with unit segment s_{i_2}.

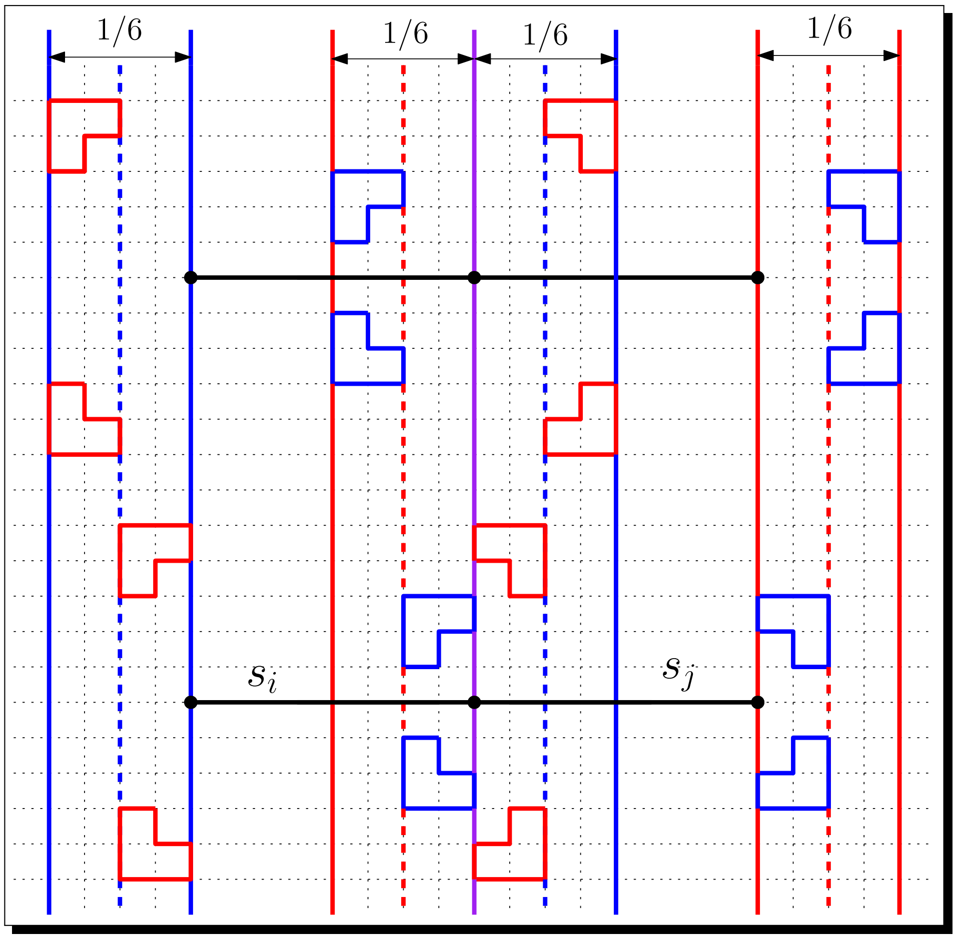

Figure 3.4: An illustration of the L-holes associated with two line segments s_i and s_j in I that share a common endpoint such that a_i is odd and $|L(s_i)| = 1$ (i.e., the only unit segment in $L(s_i)$ is the one that is directly above s_i).

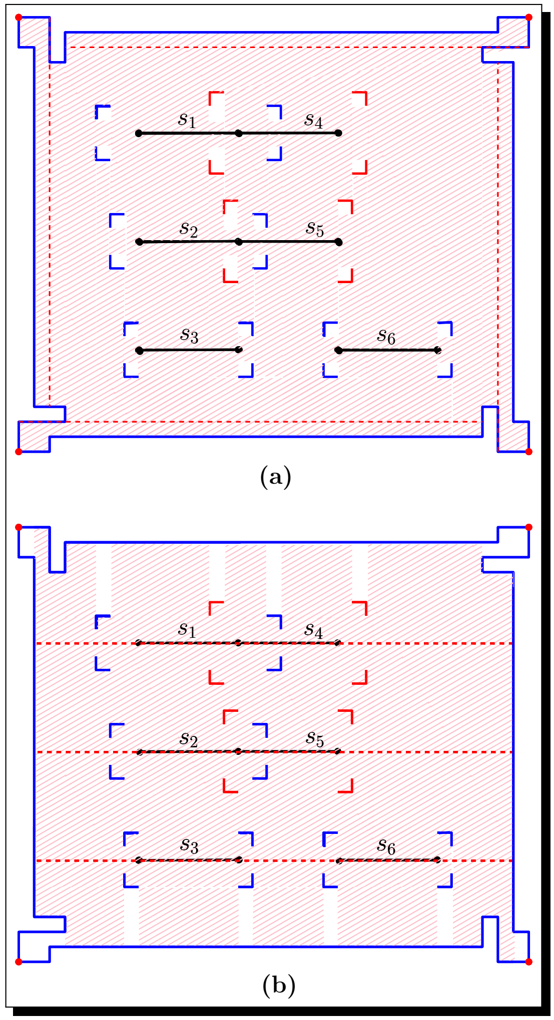

Figure 3.5: A complete example of the reduction, where $I = \{s_1, s_2, \ldots, s_6\}$, with the assumption that a_1 is even. Each line segment that has a bend represents an L-hole associated with a unit segment. Note the red vertex inside each smaller rectangle. This vertex, which we call the *visibility vertex* of the smaller rectangle, is only visible to the line segments that pass through the interior of the smaller rectangle, which in turn cannot intersect any unit segment in I. The shaded regions indicate (a) the visibility region of the boundary guards, and (b) the visibility region of three horizontal sliding cameras induced by a solution to the MIN SEGMENT HITTING problem.

Chapter 4

The MLSC Problem: An $O(n^{2.3727})$-Time Exact Algorithm

In this chapter we show that the MLSC problem is polynomially tractable even for orthogonal polygons with holes. To this end, we give an $O(n^{2.3727})$-time exact algorithm for the MLSC problem on any orthogonal polygon P, where n denotes the number of the vertices of P.

Throughout the chapter, we denote an orthogonal polygon with n vertices by P. A vertex u of P is *reflex*, if the angle at u that is interior to P is $270°$. We denote the set of reflex vertices and the set of edges of P by $V(P)$ and $E(P)$, respectively. We consider P to be a closed region; therefore, the trajectory of a sliding camera may include an edge of P. We also assume that a camera can see all points on its trajectory. We say that a set T of orthogonal line segments contained in P is a *cover* of P, if the corresponding sliding cameras can collectively see all points in P; equivalently, we say that the line segments in T *guard* P entirely.

Let T be a cover of P. We say that T is an *optimal cover for P* if the total length of line segments along which the cameras in T travel is minimum over that of all covers of P. Our algorithm relies on reducing the MLSC problem to the *minimum-weight vertex cover problem* in bipartite graphs. We remind the reader of the definition of the minimum-weight vertex cover problem:

Definition 1. *Given a graph $G = (V, E)$ with positive vertex weights, the* minimum-weight vertex cover *problem is to find a subset $V' \subseteq V$ that is a vertex cover of G (i.e., every edge in E has at least one endpoint in V') such that the sum of the weights of vertices in V' is minimized.*

The minimum-weight vertex cover problem is NP-hard in general [26]. However, König's theorem [32] that describes the equivalence between maximum matching and vertex cover in bipartite graphs implies that the minimum-weight vertex cover problem in bipartite graphs is solvable in polynomial time. Given P, we first construct a vertex-weighted graph G_P and then we show (i) that the MLSC problem on P is equivalent to the minimum-weight vertex cover problem on G_P, and (ii) that graph G_P is bipartite.

Similar to Katz and Morgenstern [27], we define a partition of an orthogonal polygon P into rectangles as follows. Extend the two edges of P incident to every reflex vertex in $V(P)$ inward until they hit the boundary of P. We define a set $S(P)$ as follows. For every edge e of P, if both endpoints of e are convex vertex, then add e to $S(P)$. If at least one of the endpoints of e is a reflex vertex, then add into $S(P)$ the line segment induced by extending e. We refer to elements of $S(P)$ simply as *edges*. The edges in $S(P)$ partition P into a set of rectangles; let $R(P)$ denote the set

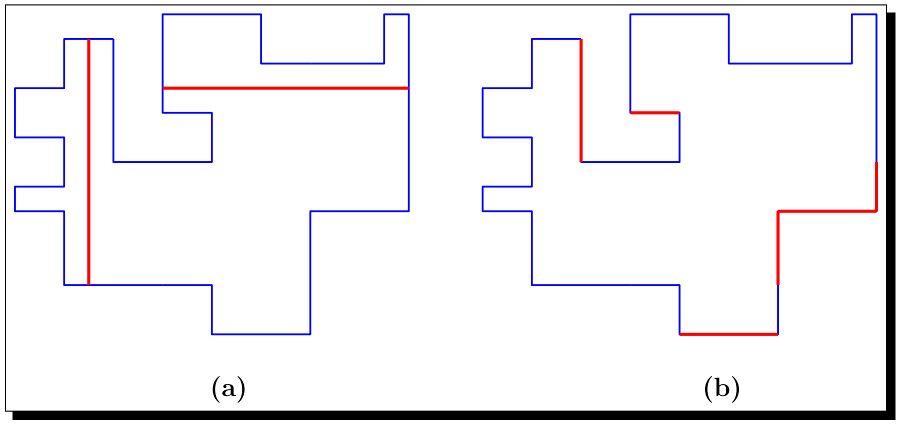

Figure 4.1: An illustration of a regular cover. (a) A simple orthogonal polygon P and two sliding cameras as a cover C of P. (b) A regular cover of P induced by C.

of resulting rectangles. We observe that in order to guard P entirely, it suffices to guard all rectangles in $R(P)$. The following observations are straightforward:

Observation 4.0.1. *Let T be a cover of P and let s be an orthogonal line segment in T. Then, for any partition of s into line segments $s_1, s_2, \ldots, s_k$ the set $T' = (T \setminus \{s\}) \cup \{s_1, \ldots, s_k\}$ is also a cover of P and the respective sums of the lengths of segments in T and T' are equal.*

Observation 4.0.2. *Let T be a cover of P. Moreover, let T' be the set of line segments obtained from T by translating every vertical line segment in T horizontally to the nearest boundary of P to its right and every horizontal line segment in T vertically to the nearest boundary of P below it. Then, T' is also a cover of P and the respective sums of the lengths of line segments in T and T' are equal. We call T' a* regular cover *of P. See Figure 4.1 for an example.*

Given P, let $H(P)$ denote the subset of the boundary of P consisting of line segments that are immediately to the right of or below P; in other words, for each edge $e \in H(P)$, the region of the plane immediately to the right of or below e does not belong to the interior of P. Let $B(P)$ denote the partition of $H(P)$ into line segments induced by the edges in $S(P)$. The following lemma follows from Observations 4.0.1 and 4.0.2:

Lemma 4.0.3. *Every orthogonal polygon P has an optimal cover $T \subseteq B(P)$.*

Observation 4.0.4. *Let P be an orthogonal polygon and consider its corresponding set $R(P)$ of rectangles induced by edges in $S(P)$. Every rectangle $R \in R(P)$ is seen by exactly one vertical line segment in $B(P)$ and exactly one horizontal line segment in $B(P)$. Furthermore, if $T \subseteq B(P)$ is a cover of P, then every rectangle in $R(P)$ must be seen by at least one horizontal or one vertical line segment in T.*

We denote the horizontal and vertical line segments in $B(P)$ that can see a rectangle $R \in R(P)$ by R_V and R_H, respectively. Using Observation 4.0.4, we now describe a reduction of the MLSC problem to the minimum-weight vertex cover problem. We construct an undirected weighted graph $G_P = (V, E)$ associated with P as follows: each line segment $s \in B(P)$ corresponds to a vertex $v_s \in V$ such that the weight of v_s is the length of s. We denote the vertex in V that corresponds to the line segment $s \in B(P)$ by v_s. Two vertices $v_s, v_{s'} \in V$ are adjacent in G_P if and only if the line segments s and s' see a common rectangle $R \in R(P)$. See Figure 4.2. By Observation 4.0.4 the following result is straightforward:

Observation 4.0.5. *There is a bijection between rectangles in $R(P)$ and edges in G_P.*

Next we show equivalence between the two problems and then prove that graph G_P is bipartite. To this end, we first need the following result.

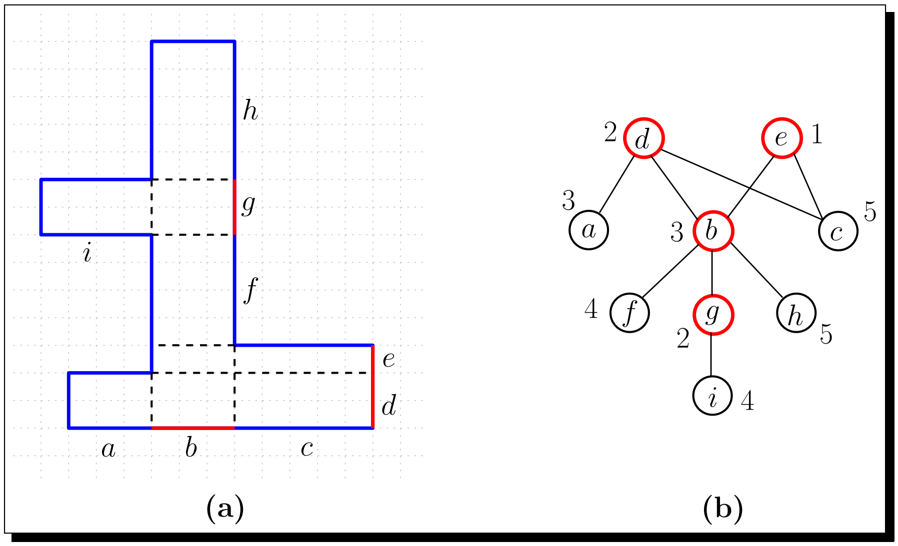

Figure 4.2: An illustration of the reduction; each grid cell has size 1×1. (a) An orthogonal polygon P along with the elements of $B(P)$ labelled as $a, b, c, \ldots, i$. (b) The graph G_P associated with P; the integer value besides each vertex indicates the weight of the vertex. The vertices of a vertex cover on G_P and their corresponding guarding line segments for P are shown in red.

Lemma 4.0.6. *Let $R \in R(P)$ be a rectangle and let T be a cover of P. Then, there exists a set $T' \subseteq T$ such that all line segments in T' have the same orientation (i.e., they are all vertical or they are all horizontal) and they collectively guard R entirely.*

Proof. Suppose no such set T' exists. Let R_v (resp., R_h) be the subregion of R that is guarded by the union of the vertical (resp., horizontal) line segments in T and let $R_v^c = R \setminus R_v$ (resp., $R_h^c = R \setminus R_h$). Since R cannot be guarded exclusively by vertical line segments (resp., horizontal line segments), we have $R_v^c \neq \emptyset$ (resp., $R_h^c \neq \emptyset$). Choose any point $p \in R_v^c$ and let L_h be the maximal horizontal line segment inside R that crosses p. Since no vertical line segment in T can guard p, we conclude that no point on L_h is guarded by a vertical line segment in T. Similarly, choose any point $q \in R_h^c$ and let L_v be the maximal vertical line segment inside R that contains q. By an analogous argument, we conclude that no point on L_v is guarded by a horizontal line segment. Since L_h and L_v are maximal and have perpendicular orientations, L_h and L_v intersect inside R. Therefore, no orthogonal line segment in T can guard the intersection point of L_h and L_v, which is a contradiction. □

Theorem 4.0.7. *The MLSC problem on P reduces to the minimum-weight vertex cover problem on G_P.*

Proof. Let S_0 be a vertex cover of G_P and let C_0 be a cover of P defined in terms of S_0; the mapping from S_0 to C_0 will be defined later. Moreover, for each vertex v of G_P let $w(v)$ denote the weight of v and for each line segment $s \in C_0$ let $len(s)$ denote the length of s. We need to prove that S_0 is a minimum-weight vertex cover of G_P if and only if C_0 is an optimal cover of P. We show the following stronger statements: (i) for any vertex cover S of G_P, there exists a cover C of P such that

$$\sum_{s \in C} len(s) = \sum_{v \in S} w(v),$$

and (ii) for any cover C of P, there exists a vertex cover S of G_P such that

$$\sum_{v \in S} w(v) = \sum_{s \in C} len(s).$$

Part 1. Choose any vertex cover S of G_P. We find a cover C for P as follows: for each edge $(v_s, v_{s'}) \in E$, if $v_s \in S$ we locate a guarding line segment on the boundary of P that is aligned with the line segment $s \in B(P)$. Otherwise, we locate a guarding line segment on the boundary of P that is aligned with the line segment $s' \in B(P)$.

Since at least one of v_s and $v_{s'}$ is in S, we conclude by Observation 4.0.5 that every rectangle in $R(P)$ is guarded by at least one line segment located on the boundary of P and so C is a cover of P. Moreover, for each vertex in S we locate exactly one guarding line segment on the boundary of P whose length is the same as the weight of the vertex. Therefore,

$$\sum_{s \in C} len(s) = \sum_{v \in S} w(v).$$

Part 2. Choose any cover C of P. We construct a vertex cover S for G_P as follows. By Observation 4.0.2, let T' be the regular cover obtained from C. Moreover, let M be the partition of T' into line segments induced by the edges in $S(P)$. By Observation 4.0.2, M is also a cover of P. Now, let S be the subset of the vertices of G_P such that $v_s \in S$ if and only if $s \in M$. By Lemma 4.0.6, for any rectangle $R \in R(P)$, there exists a set $C'_R \subseteq C$ such that all line segments in C'_R have the same orientation and collectively guard R. Therefore, by Observation 4.0.5 and the fact that M is a cover of G_P, we conclude that S is a vertex cover of G_P. Moreover, we observe that

$$\sum_{v \in S} w(v) = \sum_{s \in M} len(s) = \sum_{s \in C} len(s).$$

□

We now show that graph G_P is bipartite.

Lemma 4.0.8. *Graph G_P is bipartite.*

Proof. The proof follows from the facts that (i) we have two types of vertices in G_P; those that correspond to the vertical line segments in $B(P)$ and those that correspond to the horizontal line segments in $B(P)$, and that (ii) no two vertical line segments in $B(P)$ nor any two horizontal line segments in $B(P)$ can see a fixed rectangle in $R(P)$. □

We now examine the running time of the algorithm. Graph G_P can be constructed in $O(n^2 \log n)$ time as follows (recall that n is the number of the vertices of P). For each line segment $s \in B(P)$, let $r(s)$ be a ray that is normal to s and starts from a point on s. For every pair $s_1, s_2 \in B(P)$, where s_1 is horizontal and s_2 is vertical, we check to see if $r(s_1)$ and $r(s_2)$ intersect each other inside P. If so, then the line segments s_1 and s_2 can see a common rectangle and, therefore, we add an edge between the corresponding vertices in G_P. Considering all pairs $s_1, s_2 \in B(P)$ takes $O(n^2)$ time and the ray shooting quesries can be answered in $O(\log n)$ time [6]. Therefore, graph G_P is constructed in $O(n^2 \log n)$ time. The procedure described in Part 1 of the proof of Theorem 4.0.7 can be completed in $O(n)$ time. We now examine the running time for the construction described in Part 2 of the proof of Theorem 4.0.7. We first note that by locating a guard on every edge of the polygon P, we obtain a feasible solution for the MLSC problem; hence, we can assume that $|C| \in O(n)$, where C is a cover of P. So, we can compute a regular cover of P from C in $O(n)$ time, by moving every line segment in C down or to the right until it hits the boundary of P. The set M can also be obtained in linear time. Thus, the construction in the second part of the proof can be also completed in $O(n)$ time. A minimum vertex cover of G_P can be found by solving the maximum matching problem on G_P since these two problems are equivalent on bipartite graphs by Konig's theorem [32]. The maximum matching on G_P can be sovled in $O(T(n))$ time, where $T(n)$ is the time required to multiply two $n \times n$ matrices [37]. Since the best known algorithm for multiplying two $n \times n$ matrix runs in $O(n^{2.3727})$ time [49], our algorithm runs in $O(n^{2.3727})$ overall time. Therefore, by Theorem 4.0.7, Lemma 4.0.8 and the fact that minimum-weight vertex cover is solvable in $O(n^{2.3727})$ time on bipartite graphs, we have the main result of this chapter:

Theorem 4.0.9. *Given an orthogonal polygon P, there exists an $O(n^{2.3727})$-time algorithm that finds an optimal cover of P, where n is the number of the vertices of P.*

Chapter 5

The MCSC Problem on Monotone Art Galleries

In this chapter, we give a linear-time dynamic programming algorithm for the MCSC problem on orthogonal x-monotone polygons. This not only improves the 2-approximation algorithm of Katz and Morgenstern [27], but also provides, to the best of our knowledge, the first polynomial-time algorithm for the MCSC problem on a non-trivial subclass of orthogonal polygons. We also show how to extend this result to so-called *orthogonal path polygons*. These are orthogonal polygons for which the dual graph induced by the vertical decomposition of P is a path. (The vertical decomposition of an orthogonal polygon P is the decomposition of P into rectangles obtained by extending the vertical edge incident to every reflex vertex of P inward until it hits the boundary of P. The dual graph of the vertical decomposition is the graph that has a node for each rectangle in the decomposition and an edge between two nodes if and only if their corresponding rectangles are adjacent.) Observe that the class of orthogonal monotone polygons is a subclass of orthogonal path polygons.

5.1 Preliminaries

For a simple orthogonal and x-monotone polygon P, the leftmost and rightmost vertical edges of P are unique and we denote them by leftEdge(P) and rightEdge(P), respectively. For a sliding camera s in P, we define the *visibility polygon* of s as the maximal subpolygon $P(s)$ of P such that every point in $P(s)$ is guarded by s.

Let $V_P = \{e_1 = \text{leftEdge}(P), e_2, \ldots, e_m = \text{rightEdge}(P)\}$, for some $m > 0$, be the set of vertical edges of P ordered from left to right. For simplicity we assume that every two vertical edges in V_P have distinct x-coordinates, but it is easy to adapt the algorithm to handle degenerate cases. Let P_i^+ (resp., P_i^-), for some $1 \leq i \leq m$, denote the subpolygon of P that lies to the right (resp., to the left) of the vertical line through e_i.

For an orthogonal line segment s in P, we denote the left endpoint and the right endpoint of s by left(s) and right(s), respectively. If s is vertical, we define its left and right endpoints to be its upper and lower endpoints, respectively. We denote the x-coordinate of a point p by $x(p)$. Let s_i and s_j be two horizontal line segments in P. We define the *overlap region of* s_i *and* s_j as the set of points in P that are visible to both s_i and s_j; if $P(s_i) \cap P(s_j)$ is a line or a point (i.e., it has measure zero), then we consider the overlap region of s_i and s_j to be empty. We first show that we can restrict our attention to solutions that are in some suitable canonical form.

Canonical Form.

A *feasible solution* to the MCSC problem is a set M of sliding cameras that guards the entire polygon P. We say that a feasible solution M is in *canonical form* if and only if the following properties hold:

(i) Every vertical line segment in M is *vertically maximal*, meaning that it extends as far upwards and downwards as possible.

(ii) No vertical line segment in M intersects the interior or passes through the right endpoint of any horizontal line segment in M.

(iii) The overlap region of s_i and s_j is empty, for every two horizontal line segments $s_i, s_j \in M$ such that $s_i \neq s_j$.

(iv) Every horizontal line segment $s \in M$ is *rightward maximal*, meaning that s extends at least as far to the right as any horizontal line segment $s' \subset P$ starting at the same x-coordinate, that is, with $x(\text{left}(s)) = x(\text{left}(s'))$.

(v) Let $s_1, \ldots, s_k$ be the sequence of line segments in M ordered from left to right according to their left endpoint, where in case of ties vertical line segments come before horizontal line segments, and let $M_i := \{s_1, \ldots, s_i\}$. Then, M_i guards every point of P that is to the left of the vertical line $x = x(right(s_i))$.

Lemma 5.1.1. *For any x-monotone orthogonal polygon P, there exists an optimal solution M for the MCSC problem on P that is in canonical form.*

Proof. Consider the sequence $s_1, \ldots, s_k$ of line segments in M ordered from left to right according to their left endpoint, where in case of ties vertical line segments come before horizontal line segments. This ordering is well defined, because an optimal solution will never have two vertical line segments with the same x-coordinates or two horizontal line segments whose left endpoints have the same x-coordinates. We now show how to modify the line segments in M to get an optimal solution in canonical form. Without loss of generality, we assume that all vertical segments in M are already vertically maximal.

We first modify M so that if s_1 is horizontal, then $\text{left}(s_1)$ lies on $\text{leftEdge}(P)$, the leftmost vertical edge of P. Assume this is not the case. Then, $\text{leftEdge}(P)$ is seen by a vertical line segment s_j, for some $j > 1$. We now replace s_1 and s_j by two horizontal line segments, as follows. The first line segment is a rightward maximal line segment s starting on $\text{leftEdge}(P)$—note that s must intersect s_j—and the second horizontal line segment is a rightward maximal line segment s' with $x(\text{left}(s')) = x(\text{right}(s))$. Clearly replacing s_1, s_j by s, s' gives another optimal solution. With a slight abuse of notation we let M denote this new optimal solution, and we let $s_1, \ldots, s_k$ denote the ordered set of line segments in the new solution. Note that we now have that if s_1 is horizontal, then it starts at $\text{leftEdge}(P)$ and it is rightward maximal.

Next we turn M into an optimal solution in canonical form. To this end we go over the line segments in order. When we handle line segment s_i we will replace s_i by a line segment s'_i, but we will not modify any other line segment. Let $M_i := \{s'_1, \ldots, s'_i\}$. We maintain the following invariant:

> *Invariant:* After handling s_i, the modified set M is still an optimal solution. Moreover, M_i has all the required properties: (i) all vertical line segments in M_i are vertically maximal, (ii) no vertical line segment in M_i intersects the interior or passes through the right endpoint of any

horizontal line segment in M_i, (iii) the overlap region of any two horizontal line segments in M_i is empty, (iv) every horizontal line segment in M_i is rightward maximal, and (v) M_i guards everything to the left of the vertical line $x = x(\text{right}(s_i))$.

Handling s_1 is trivial: we simply set $s'_1 := s_1$. If s_1 is vertical then this clearly establishes the invariant—note that no line segment s_j with $j > 1$ can see anything to the left of s_1 that is not also seen by s_1 (since s_1 is vertically maximal), which implies that s_1 must see everything to its left. If s_1 is horizontal then the invariant holds as well, since we already made sure that s_1 is rightward maximal if it is horizontal. Now suppose the invariant holds after we have handled s_{i-1}, and consider s_i. There are two cases.

- If s_i is vertical, then we proceed as follows. Observe that M_i must guard everything to the left of s_i, since M is feasible and no line segment s_j with $j > i$ can see a point to the left of s_i that is not seen by s_i. Since s_i is vertical, the fact that M_{i-1} satisfies the invariant immediately implies that M_i has properties (iii) and (iv). So the only problem is that s_i may intersect the interior or may pass through the right endpoint of some line segment s'_j with $j < i$. If this is not the case we simply set $s'_i := s_i$, otherwise we replace s_i by a rightward maximal line segment s'_i with $x(\text{left}(s'_i)) = x(\text{right}(s'_j))$.

  After the replacement, M is still a feasible (and, hence, optimal) solution. Indeed, everything to the left of the vertical line $x = x(\text{right}(s'_j))$ is guarded by M_j and to the right of the vertical line $x = x(\text{right}(s'_j))$, the new line segment s'_i sees at least as much as s_i. By the same argument, M_i guards everything to the left of the vertical line $x = x(\text{right}(s'_i))$ and, therefore, the property (v) holds. Finally, M_i has properties (i) and (ii) because M_{i-1} had those properties and the new line segment s'_i is horizontal, and M_i has properties (iii) and (iv) by construction.

There is one subtlety that we must address. Namely, we have to show that after replacing s_i by s'_i the order of the line segments does not change. In other words, we must show that s'_i is still the i-th line segment in the order. (Otherwise we would have to argue about a different set M_i.) Obviously $\text{left}(s'_i)$ lies to the right of $\text{left}(s'_j)$ for all $j < i$. Moreover, there cannot be any line segment s_k with $k > i$ such that $\text{left}(s_k)$ lies in between s_i and $\text{left}(s'_i)$. Indeed, such a line segment could be omitted, contradicting the optimality of M.

- If s_i is horizontal, we proceed as follows. Obviously, the only properties that may be violated are properties (iii) and (iv). It might be the case that a vertical line segment $s_j \in M$ intersects the interior or passes through the right endpoint of s_i (thus violating the property (ii)), but this may happen only if $j > i$ and, therefore, the invariant is still maintained for M_i; if such line segment s_j exists, then the set M will be modified when we later handle s_j. If M_i only violates property (iv), then we replace s_i by a rightward maximal line segment s'_i with $x(\text{left}(s'_i)) = x(\text{left}(s_i))$. If s_i violates property (iii), then let $s'_j \in M_{i-1}$ be the horizontal line segment that has an overlap with s_i. We now replace s_i by a rightward maximal line segment s'_i with $x(\text{left}(s'_i)) = x(\text{right}(s'_j))$.

  Since s'_i sees at least as much as s_i (except possibly for points that were already seen by s'_j), the new solution is still feasible. Moreover, M_i sees everything to the left of the vertical line $x = x(\text{right}(s'_i))$. Therefore, since M_{i-1} satisfies the invariant and because of the way s'_i is constructed, M_i has all the properties (i)–(v). Finally, the new line segment s'_i is still the i-th line segment in the order, as can be verified in the same way as before.

After handling the last line segment s_k in M, the set M_k is an optimal solution in canonical form, thus proving the lemma. □

5.2 A Dynamic Programming Algorithm

In this section, we present the linear-time exact algorithm for the MCSC problem on orthogonal and x-monotone polygons. Our algorithm is based on a dynamic programming approach.

5.2.1 The Recursive Structure

Let P be an orthogonal x-monotone polygon with n vertices. Below we discuss the recursive structure of the MCSC problem on P and we define the subproblems we use in our dynamic programming algorithm.

Let $M_{\text{OPT}} = \{s_1, \ldots, s_k\}$ be an optimal solution for the MCSC problem on P that is in canonical form, where the segments are numbered from left to right. Consider a segment $s_j \in M_{\text{OPT}}$. By property (v) of the canonical form, no segment $s_{j'} \in M_{\text{OPT}}$ with $j > j$ is needed to guard anything to the left of $\text{right}(s_j)$. Hence, after having selected $s_1, \ldots, s_j$, the subproblem we are left with is to guard P_i^+, where i is such that $\text{right}(s_j)$ lies on the line containing the vertical edge e_i. Note that when s_j is vertical, we already guarded a part of P_i^+, and we have to take this into account in our subproblem. Hence, we define two types of subproblems.

Type A. Given $1 \leq i \leq k$, guard P_i^+ with the minimum number of sliding cameras.

Type B. Given $1 \leq i \leq k$, guard P_i^+ with the minimum number of sliding cameras, under the assumption that the subregion of P_i^+ that is visible from leftEdge(P_i^+) has already been guarded.

We denote the number of guards needed in an optimal solution of Type A on the polygon P_i^+ by $A[i]$ and the number of guards needed in an optimal solution of Type B on the polygon P_i^+ by $B[i]$. Note that the minimum number of cameras needed to guard the entire polygon P is $A[1]$. In the sequel we show how to compute the values $A[i]$ and $B[i]$; computing the actual solution can then be done in a standard manner.

5.2.2 Solving the Subproblems

We now give the recursive formulas on which our dynamic programming algorithm is based. Recall that the vertical edges of P are number $e_1, \ldots, e_k$ from left to right. We denote the vertical line containing e_i by $\ell(e_i)$. The following lemma gives the recursive formula for solving the subproblem of Type A on P_i^+.

Lemma 5.2.1. *Let s be a rightward maximal line segment whose left endpoint lies on $\ell(e_i)$, and let e_{i_1} be the vertical edge of P on which* right(s) *lies. Furthermore, let e_{i_2} be the rightmost vertical edge of P such that s', the vertically maximal segment aligned with e_{i_2},*

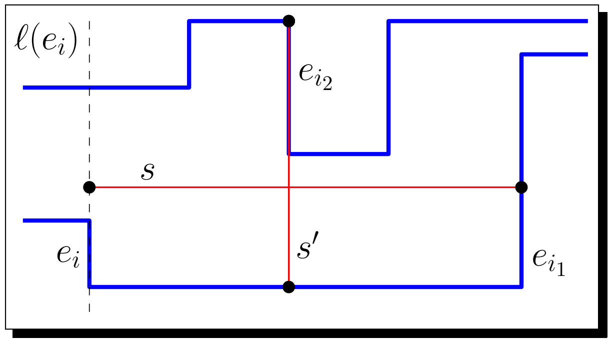

Figure 5.1: An illustration of the two cases in solving subproblem of Type A on P_i^+.

guards everything of P_i^+ lying to the left of e_{i_2}. See Figure 5.1 for an illustration. Then,

$$A[i] = \begin{cases} 0 & \textit{if } i = k \\ \min\left(A[i_1], B[i_2]\right) + 1 & \textit{if } i < k \end{cases}$$

Proof. Trivially $A[i] = 0$ for $i = k$, so assume $i < k$.

Consider the first segment s^* of an optimal solution for P_i^+ that is in canonical form. By property (v), we know that s^* must guard leftEdge(P_i^+). Hence, if it is horizontal, it must start at leftEdge(P_i^+). By property (iv), segment s^* is rightward maximal. Hence, if the first segment is horizontal then the segment s is the correct choice. After choosing s, we have to guard everything to the right of s. Note that properties (ii) and (iii) imply that the next segment to be chosen lies in $P_{i_1}^+$ (see Figure 5.1). Hence, if we decide to pick segment s then we are indeed left with solving the subproblem of Type A on $P_{i_1}^+$. Thus in this case $A[i] = A[i_1] + 1$.

The other option is that the first segment s^* is vertical. Again, by property (v) we know that s^* must guard everything between leftEdge(P_i^+) and s^*. But then it is obviously best to choose s^* as far to the right as possible. Hence, s' is the correct choice. Now the subproblem we are left with is of Type B and on $P_{i_2}^+$ (see

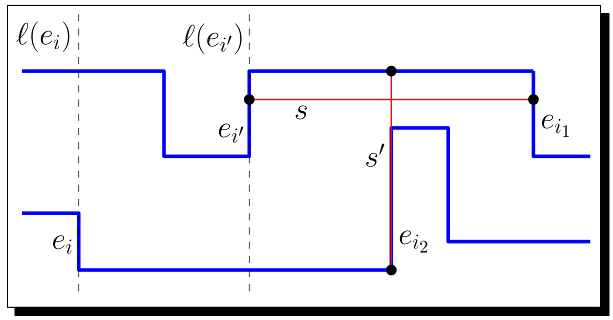

Figure 5.2: An illustration of the two cases in solving subproblem of Type B on P_i^+.

Figure 5.1), so we have $A[i] = B[i_2] + 1$.

The best way to solve subproblem of Type A on P_i^+ is the best of these two options, which proves the lemma. □

For the subproblems of Type B we have a similar lemma.

Lemma 5.2.2. *Let $e_{i'}$ be the leftmost vertical edge in P_i^+ that is not seen by* leftEdge(P_i^+)*, let s be a rightward maximal line segment whose left endpoint lies on $\ell(e_{i'})$, and let e_{i_1} be the vertical edge of P on which* right(s) *lies. Furthermore, let e_{i_2} be the rightmost vertical edge of P such that s', the vertically maximal segment aligned with e_{i_2}, together with* leftEdge(P_i^+) *guards everything of P_i^+ lying to the left of e_{i_2}. See Figure 5.2 for an illustration. Then,*

$$B[i] = \begin{cases} 0 & \text{if } i = k \\ \min\left(A[i_1], B[i_2]\right) + 1 & \text{if } i < k \end{cases}$$

Proof. Trivially $B[i] = 0$ for $i = k$, so assume $i < k$.

Consider the first segment s^* of an optimal solution for P_i^+ that is in canonical form. First, suppose that s^* is horizontal. Obviously it is best to make s^* extend to the right as much as possible, which means left(s^*) should be to the right as far as possible. However, left(s^*) cannot go beyond $e_{i'}$ by property (v). By property (iv), segment s^* is rightward maximal. Hence, if the first segment is horizontal then the segment s is the correct choice. After choosing s, we have to guard everything to the right of s. Note that properties (ii) and (iii) imply that the next segment to be chosen lies in $P_{i_1}^+$. Moreover, since s is rightward maximal and starts to the right of leftEdge(P_i^+), the edge leftEdge(P_i^+) cannot see anything to the right of right(s). Hence, if we decide to pick segment s then we are indeed left with solving the subproblem of Type A on $P_{i_1}^+$ (see Figure 5.2). Thus, in this case $B[i] = A[i_1] + 1$.

The other option is that the first segment s^* is vertical. Again, by property (v) we know that s^*, together with leftEdge(P_i^+), must guard everything between leftEdge(P_i^+) and s^*. But then it is best to choose s^* as far to the right as possible. Hence, s' is the correct choice. Now the subproblem we are left with is of Type B and on $P_{i_2}^+$ (see Figure 5.2), so we have $B[i] = B[i_2] + 1$.

The best way to solve subproblem of Type B on P_i^+ is the best of these two options, which proves the lemma. □

5.2.3 Algorithmic Details

In this section, we analyze the algorithm and describe how it can be implemented in linear time. To compute the optimal solution for guarding P_i^+, we need to solve two subproblems; that is, we need to solve a subproblem of Type A and a subproblem of Type B for P_i^+. To solve the subproblem of Type A for P_i^+, we need to solve two subproblems: one is of Type A for which we need to find the vertical edge e_{i_1} described in Lemma 5.2.1, and the other one is of Type B for which we need to find the vertical edge e_{i_2} described in Lemma 5.2.1. Similarly, to solve the subproblem of Type B for P_i^+, we need to solve two subproblems: one is of Type A for which we need to find the vertical edge e_{i_1} described in Lemma 5.2.2, and the other one is of Type B for which we need to find the vertical edge e_{i_2} described in Lemma 5.2.2. Therefore, each vertical edge $e_i \in V_P$ is associated with at most four other vertical edges of P; we call these four edges the *associated edges of* e_i. In the following, we show how the associated edges can be computed in $O(n)$ time for all the vertical edges in V_P.

Lemma 5.2.3. *The associated edges of all the vertical edges in V_P can be computed in $O(n)$ time.*

Proof. We show that each type of associated edge can be computed in linear time for all the vertical edges in V_P. The lemma then follows from the fact that there are four types of associated edges. We first give some definitions. A reflex vertex v of P is called *right* reflex (resp. *left* reflex) if the interior of P lies to the right (resp., to the left) of the vertical edge incident to v. Moreover, for a reflex vertex v_i of P, we denote the vertical edge incident to v_i by e_i and the maximal vertical line segment in P aligned with e_i by L_i. In the following, we assume that the sequence of the reflex vertices of P ordered from right to left is given.

Step 1: the associated edge e_{i_1} described in Lemma 5.2.1. To compute this associated edge, we use a vertical line sweeping P from right to left; the sweep line halts at each reflex vertex of P. Let `UQ` and `LQ` be two double-ended queues that store the reflex vertices, respectively, on the upper chain and lower chain of P. By one exception, we assume that `UQ` (resp., `LQ`) contains initially the upper vertex (resp., the lower vertex) of rightEdge(P). Reflex vertices are added to the end of the queues, but they might be removed from either the front or the end of the queues. The vertices are removed from a queue depending on whether the vertex v_i at which the sweep line is currently halted lies on the upper chain or on the lower chain of P and also depending on where v_i lies on the chain relative to the previously visited vertices. We maintain the following invariant:

> *Invariant:* When the sweep line halts at the reflex vertex v_i, then (i) the queue `UQ` stores a reflex vertex v_j of the upper chain if and only if v_j lies to the right of L_i and L_i can see at least one point on L_j; the part of L_j that is visible to L_i is also stored. The vertices in `UQ` are sorted from right to left by their x-coordinate, and (ii) the queue `LQ` stores a reflex vertex $v_{j'}$ of the lower chain if and only if $v_{j'}$ lies to the right of L_i and L_i can see at least one point on $L_{j'}$; the part of L_j that is visible to L_i is also stored. The vertices in `LQ` are sorted from right to left by their x-coordinate.

Consider v_i, the vertex at which the sweep line is currently halted, and suppose that v_x and v_y are the vertices at the front of the two queues. First, we maintain the invariant. To this end, if the part of L_j that is visible to L_i is empty, then we remove v_j from `UQ` for all v_j in `UQ`. Similarly, if the part of $L_{j'}$ that is visible to L_i is empty, then we remove $v_{j'}$ from `LQ` for all $v_{j'}$ in `LQ`. Next, we set the associated edge for e_i to e_x or e_y whichever is further to the right from e_i. The vertex v_i is then added to the appropriate queue. See Figure 5.3 for an example. Since every reflex vertex of P is added to a queue at most once, this step can be completed in $O(n)$ time.

Step 2: the associated edge e_{i_2} described in Lemma 5.2.1. To compute this associated edge, consider the reflex vertices of P from right to left sorted by their x-coordinate. Then, the associated edge e_{i_2} described in Lemma 5.2.1 for a vertical edge e_i is the edge $e \in V_P$ such that the reflex vertex v incident to e is the leftmost

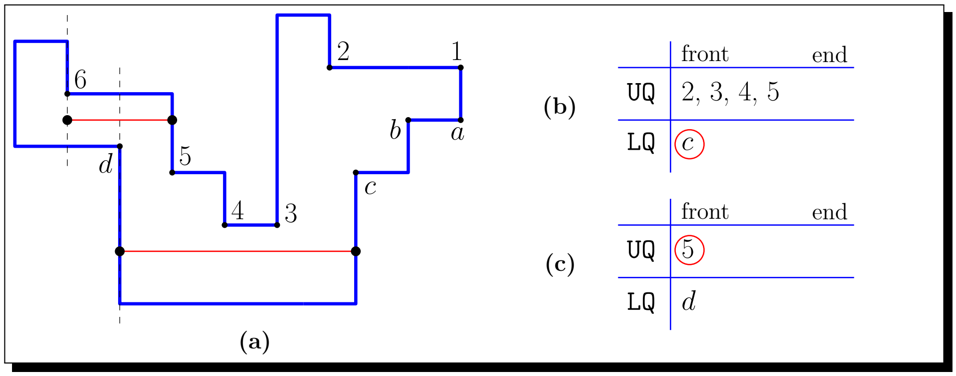

Figure 5.3: An illustration of the sweep line algorithm. (a) An orthogonal x-monotone polygon P with its reflex vertices on the upper and lower chains labeled from right to left. (b) The status of queues UQ and LQ when the sweep line halts at vertex d and the invariant is maintained: the associated edge e_{i_1} for the vertical edge incident to d is set to the vertical edge incident to vertex c and vertex d is then added to LQ. (c) The status of queues UQ and LQ when the sweep line halts at vertex 6 and the invariant is maintained: the associated edge e_{i_1} for the vertical edge incident to 6 is set to the vertical edge incident to vertex 5 and vertex 6 is then added to UQ.

left reflex vertex of P such that $x(v) > x(v_i)$; such vertex v and, therefore, its incident vertical edge e can be computed in linear time for all the vertical edges in V_P.

Step 3: the associated edge e_{i_1} described in Lemma 5.2.2. To compute this associated edge for an edge $e_i \in V_P$, we first need to compute the vertical edge $e_{i'}$ of P. The edge $e_{i'}$ for e_i is the edge $e \in V_P$ such that the reflex vertex v incident to e is the leftmost right reflex vertex of P such that $x(v) > x(v_i)$. Then, the edge e_{i_1} for e_i is exactly the associated edge that we have already computed for $e_{i'}$ in Step 1. Both vertical edges $e_{i'}$ and e_{i_1} can be computed in linear time for all the vertical edges in V_P.

Step 4: the associated edge e_{i_2} described in Lemma 5.2.2. To find this associated edge for a vertical edge e_i, we first find the leftmost right reflex vertex v_j such that $x(v_j) > x(v_i)$; observe that every point of P that lies between L_i and L_j (i.e., the maximal vertical line segments in P aligned with e_i and e_j, respectively) is guarded by leftEdge(P_i^+). Therefore, the associated edge e_{i_2} for e_i is in fact the vertical edge that is furthest to the right from L_j such that every point between L_j and L_{i_2} is guarded by L_{i_2}. But, L_{i_2} is aligned with exactly the associated edge that we have already computed for e_j in Step 2. Therefore, to compute the associated edge e_{i_2} for a vertical edge e_i, we first find the leftmost right reflex vertex v_j to the right of e_i and then return the associated edge computed in Step 2 for e_j. Both vertical edges v_j and e_{i_2} can be computed in $O(n)$ time for all the vertical edges in V_P.

Therefore, we can compute all the four associated edges in $O(n)$ time for all the vertical edges in V_P. This completes the proof of the lemma. □

By Lemma 5.2.3, we first compute the associated edges of all the vertical edges of P in $O(n)$ time. Then, we consider the vertical edges of P in order from right to left and compute the optimal solution for guarding P_i^+ in $O(1)$ time by computing $A[i]$ and $B[i]$ as described, respectively, in Lemma 5.2.1 and Lemma 5.2.2. Finally, $A[1]$ is returned as the optimal solution for the MCSC problem on P. Therefore, we have the main result of this section:

Theorem 5.2.4. *There exists an algorithm that solves the MCSC problem on any simple orthogonal and x-monotone polygon with n vertices in $O(n)$ time.*

5.3 Orthogonal Path Polygons

In this section, we show that the dynamic programming algorithm given in Section 5.2 can be used to solve the MCSC problem on any orthogonal path polygon P with n vertices in $O(n)$ time; that is, we show that the MCSC problem can be solved in $O(n)$ time on any simple orthogonal polygon P for which the dual graph $G(P)$ is a path. To this end, we first describe the structure of P and then will show that P can be converted into an x-monotone polygon by *unfolding*.

Let P be an orthogonal path polygon with n vertices. If P is x-monotone, then we solve the MCSC problem on P in linear time by Theorem 5.2.4. If polygon P is not x-monotone, then we first partition P into x-monotone subpolygons as follows. Since polygon P is not x-monotone, it must have a vertical edge e whose both endpoints are reflex vertices of P. Partition P into three subregions by the maximal vertical line segment L that is aligned with e. The subregions induced by L are a rectangle R and two subregions P_L and P_U that are connected to lower and upper parts of one of the sides of R, respectively. Partition P_L and P_R recursively until the subregions induced by the partitions become x-monotone; see Figure 5.4 for an illustration. Let $P_1, P_2, \ldots, P_k$ be the set of x-monotone subpoly-

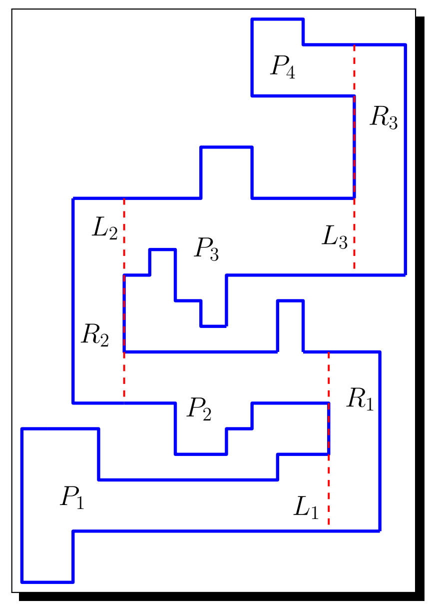

Figure 5.4: An example of an orthogonal path polygon P that is not x-monotone along with an illustration of partitioning P into x-monotone subpolygons.

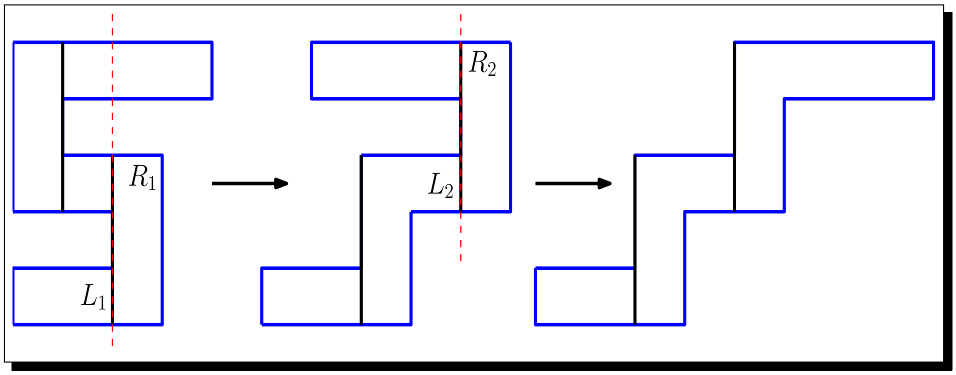

Figure 5.5: An illustration of transforming a non-x-monotone polygon into an x-monotone polygon by *unfolding* the polygon.

gons of P from bottom to top. Moreover, let L_i, for all $1 \leq i < k$, be the maximal line segment by which we perform the partition and let R_i, for all $1 \leq i < k$, be the corresponding rectangle. Now, for each rectangle R_i in order, we unfold P by flipping the subregion $P_{i+1} \cup P_{i+2} \cup \cdots \cup P_k$ across the line through L_i such that R_{i+1} lies to the same side of L_i as R_i lies. The i-th flip ensures that the subregion $P_1 \cup P_2 \cup \cdots \cup P_{i+1}$ of P is an x-monotone polygon. Therefore, polygon P is converted to an x-monotone polygon after the last flip. See Figure 5.5 for an illustration.

To summarize, we first convert P into an x-monotone polygon using at most $k < n$ flip operations as described above and then solve the MCSC problem on the resulting x-monotone polygon using the dynamic programming algorithm given in Section 5.2. We can compute the set of line segments L_i, for all $1 \leq i < k$, in $O(n)$ time by detecting each vertical edge of P whose both endpoints are reflex vertices of P. Next, by keeping track of the lower and upper chains of P starting from L_1, we can compute the flipped polygon in $O(n)$ time. Therefore, we have the following theorem:

Theorem 5.3.1. *There exists an algorithm that solves the MCSC problem on any orthogonal path polygon with n vertices in $O(n)$ time.*

Part II

Optimal Partitions of Orthogonal Polygons

Chapter 6

Background

A *rectangular partition* of an orthogonal polygon P is a decomposition of P into rectangles whose interiors are disjoint and whose union is P. Given a rectangular partition R of P and a line segment ℓ inside P, we say that ℓ *stabs* a rectangle of R if ℓ passes through the interior of the rectangle. The (*orthogonal*) *stabbing number* of R is the maximum number of rectangles of R stabbed by any axis-parallel line segment inside P. Moreover, the *vertical* (resp., *horizontal*) stabbing number of R is defined as the maximum number of rectangles stabbed by any vertical (resp., horizontal) line segment inside P. We say an edge of a rectangle in a rectangular partition of P is *fully anchored* if both of its endpoints are on the boundary of P. Consequently, a rectangular partition of P is called *conforming*, if all edges of its rectangles are fully anchored.

In this part, we study the Minimum Stabbing Number (MSN) problem on orthogonal polygons. Recall that the objective of the MSN problem is to compute a conforming rectangular (CR) partition of P whose stabbing number is minimum over that of all such partitions of P. Related work and preliminaries are given in this chapter, and we present our results in the rest of the chapters of this part.

6.1 Related Work

De Berg and van Kreveld [11] proved that every n-vertex orthogonal polygon has a rectangular partition with stabbing number $O(\log n)$. They showed that this bound is asymptotically tight, as the stabbing number of any rectangular partition of a staircase polygon with n vertices is $\Omega(\log n)$ (see Figure 7.2 as an example of a staircase). De Berg and van Kreveld [11] gave a polynomial-time algorithm that computes a partition with stabbing number $O(\log n)$. Hershberger and Suri [22] showed that any simple polygon P can be partitioned into $O(n)$ triangles such that any line segment inside P can intersect only $O(n)$ triangles. Recently, Abam et al. [1] considered the problem of computing an optimal rectangular partition of a simple orthogonal polygon P, that is, a rectangular partition whose stabbing number is minimum over all such partitions of P. By finding an optimal partition for histogram polygons in polynomial time (see Section 8.1), they obtained an $O(n^7 \log n \log\log n)$-time 3-approximation algorithm for this problem. As Abam et al. [1] noted, however, the computational complexity of the general problem is unknown.

De Berg et al. [10] studied a related problem in which the objective is to partition a given set of n points in $\mathbb{R}^d$ into sets of cardinality between $n/2r$ and $2n/r$ for a given r, where each set is represented by its bounding box, such that the stabbing number, defined as the maximum number of bounding boxes intersected by any axis-parallel hyperplane, is minimized. They showed that the problem is NP-hard in $\mathbb{R}^2$. Moreover, they gave an exact $O(n^{4dr+3/2}\log^2 n)$-time algorithm in $\mathbb{R}^d$ as well as an $O(n^{3/2}\log^2 n)$-time 2-approximation algorithm in $\mathbb{R}^2$, when r is constant. Fekete et al. [15] proved that the problem of finding a perfect matching with minimum stabbing number for a given point set is NP-hard, where the (orthogonal) stabbing number of a matching is the maximum number of edges of the matching intersected by any (axis-parallel) line. They also showed that, for a given point set, the problems of finding a spanning tree or a triangulation with minimum stabbing number are NP-hard. Shewchuk [46] showed that in d dimensions, a line can stab the interiors of $\Theta(n^{\lceil n/2 \rceil})$ Delaunay d-simplices. This means that a Delaunay triangulation in the plane may have linear stabbing number. Tóth [47] proved that for any subdivision of d-dimensional Euclidean space, $d \geq 2$, by n axis-aligned boxes, there is an axisparallel line that stabs at least $\Omega(\log^{1/(d-1)} n)$ boxes, which is the best possible lower bound.

6.2 Preliminaries

Given an orthogonal polygon P and a CR partition R of P, we refer to any maximal line segment whose interior lies in the interior of P and contains an edge of some rectangle in R as a *partition edge*. That is, the partition edges of R correspond to the cuts that divide P into rectangles. We denote the set of reflex vertices of P by $V_R(P)$. For each reflex vertex $u \in V_R(P)$, we denote the maximal horizontal (resp., vertical) line segment contained in the interior of P with one endpoint at u by H_u (resp., V_u) and refer to it as the *horizontal line segment* (resp., *vertical line* *segment*) of u. Observe that for every reflex vertex u of P, at least one of H_u and V_u must be present in R. The following observation allows us to consider only a discrete subset of the set of all possible rectangular partitions of P to find an optimal partition:

Observation 6.2.1. *Any orthogonal polygon P has an optimal rectangular partition in which every partition edge has at least one reflex vertex of P as an endpoint.*

Consequently, every partition edge is either H_u or V_u for some $u \in V_R(P)$. Given an integer $k \geq 1$, a k-Sum Linear Program (KLP)[1] [42] consists of an $m \times n$ matrix A, an m-vector b, and an n-vector $X = (x_1, x_2, \ldots, x_n)$ for which the objective is to

$$\begin{aligned} \text{minimize} \quad & \max_{S \subseteq N : |S| = k} \sum_{j \in S} c_j x_j \\ \text{subject to} \quad & AX \geq b \\ & X \geq 0, \end{aligned} \tag{6.1}$$

where $N = \{1, 2, \ldots, n\}$. Observe that when $k = n$, the KLP is equivalent to a classical linear program (LP). Our 2-approximation algorithm for the MSN problem is based on a KLP formulation of the problem (see Chapter 8).

[1] Throughout Part II, we use KLP to abbreviate either k-Sum Linear Program or k-Sum Linear Programming. Similarly, we use LP to denote either Linear Program or Linear Programming.

Chapter 7

Hardness Result

In this chapter, we show that computing a conforming rectangular (CR) partition of an orthogonal polygon P (possibly with holes) with minimum stabbing number is NP-hard. That is, we show that the MSN problem is NP-hard on orthogonal polygons with holes:

MSN WITH HOLES

INPUT: A orthogonal polygon P possibly with holes

OUTPUT: An optimal CR partition of P

To show the hardness of the MSN WITH HOLES problem, we describe a reduction from PLANAR VARIABLE RESTRICTED 3SAT (PLANAR VR3SAT). The PLANAR VR3SAT problem is a constrained version of 3SAT in which each variable can appear in at most three clauses and the corresponding *variable-clause graph* must be planar. Efrat et al. [12] show that PLANAR VR3SAT is NP-hard.

7.1 Reduction Overview

Let $I = \{C_1, C_2, \ldots, C_k\}$ be an instance of PLANAR VR3SAT with k clauses and n variables, $X_1, X_2, \ldots, X_n$. We construct a polygon P with holes such that P has a CR partition with stabbing number at most $5c$ if and only if I is satisfiable, where c is a constant that does not depend on I. [1] Given I, we first construct the variable-clause graph of I in the non-crossing comb-shape form of Knuth and Raghunathan [31]. Without loss of generality, we assume that the variable vertices lie on a vertical line and the clause vertices are connected from left or right of that line. Then, we replace each variable vertex X_i with a polygonal variable gadget to which three connecting corridors are attached from its left. The corridors are then connected to the clause gadgets whose associated clauses contain that variable. Figure 7.1 shows an example of a variable gadget; note the vertex v. Due to the structure of the variable gadget, any CR partition must contain exactly one of the edges V_v or H_v; including V_v (resp., H_v) in the partition corresponds to a truth assignment of *true* (resp., *false*) for the variable x. Moreover, choosing V_v or H_v imposes constraints on how the rest of the variable gadget and its associated clause gadgets can be partitioned. The overall construction implies the following lemma:

Lemma 7.1.1. *P has a CR partition with stabbing number at most $5c$, for some constant c, if and only if I is satisfiable.*

Before proving Lemma 7.1.1 we first give some definitions and describe the details of the gadgets used in the reduction from the PLANAR VR3SAT problem.

[1] The precise value of c is specified after giving the detailed description of the gadgets used in the reduction; see end of Section 7.3 for more details. Note that the value of c can be specified in polynomial time.

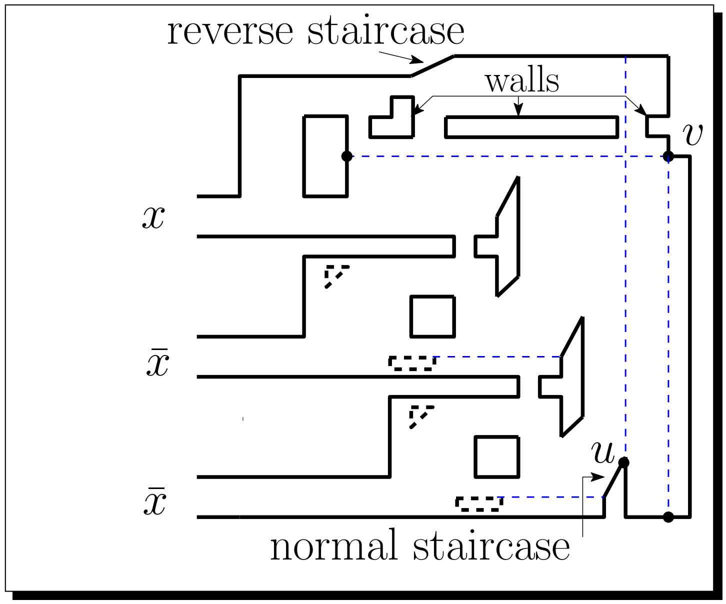

Figure 7.1: An example of a variable gadget X linked by three respective corridors to its occurrences (x, $\overline{x}$ and $\overline{x}$) in clauses. Each pair of dashed triangular and rectangular holes form a negation gadget that negates the truth value of x in the associated clause linked by the adjacent corridor. Each staircase consists of c steps.

An instance of the PLANAR 3SAT problem consists of a planar bipartite graph $G_I = (V, E)$, called a *variable-clause graph*, corresponding to a Boolean formula I in conjunctive normal form (CNF), where each clause contains three variables. The vertices in one partition of G_I correspond to the variables in I while the vertices in the other partition of G_I correspond to the clauses of I. Each clause vertex is connected by an edge to the variable vertices it contains. Knuth and Raghunathan [31] show that such a graph can be drawn on a grid with all variable vertices on a horizontal line and the clause vertices connecting vertices in a comb shape from above or below that line without any edge crossings. We assume that the variable vertices lie on a vertical line and the clause vertices are connected from left or right of that line. Next, we describe the gadgets used in our reduction.

7.2 Variable Gadgets

In this section, we describe the variable gadgets. Figure 7.1 shows an example of a variable gadget. We denote the variable gadget corresponds to variable X_i by $\texttt{VG}(X_i)$. Each variable gadget has three open corridors, namely the *top*, *middle* and *bottom* corridors. Each open corridor of $\texttt{VG}(X_i)$ is connected to one of the clauses that contains X_i. Let C_j be a clause that contains X_i. We denote the corridor connecting $\texttt{VG}(X_i)$ to C_j by $\texttt{corr}(X_i, C_j)$. $\texttt{corr}(X_i, C_j)$ indicates the presence of a literal of X_i (i.e., x_i or $\bar{x}_i$) in the clause C_i. There are two holes in the begin of $\texttt{corr}(X_i, C_j)$: a rectangular hole, called *variable hole*, and a *t-shaped hole* that has two staircases on its boundary, each consisting of c steps (each staircase is represented by a single diagonal edge in Figure 7.1), where c is a constant whose value we will determine later. To avoid confusion, we call the upper staircase of each t-shaped hole a *normal staircase* and the lower staircase of each t-shaped hole a *reverse staircase*. As Figure 7.1 shows, each variable gadget has also a normal staircase and a reverse staircase on its boundary. See Figure 7.2(a) for an illustration of a normal staircase. The details of a reverse staircase are similar, but in reverse.

We separate the upper part of each variable gadget from the rest with two holes and a part of the boundary of $\texttt{VG}(X_i)$, called *walls*. There is a gap between the two right walls. We observe in Figure 7.1 that V_u passes through this gap and enters into the upper part of P. Note that the vertical lines through all remaining vertices on this staircase intersect one of the walls.

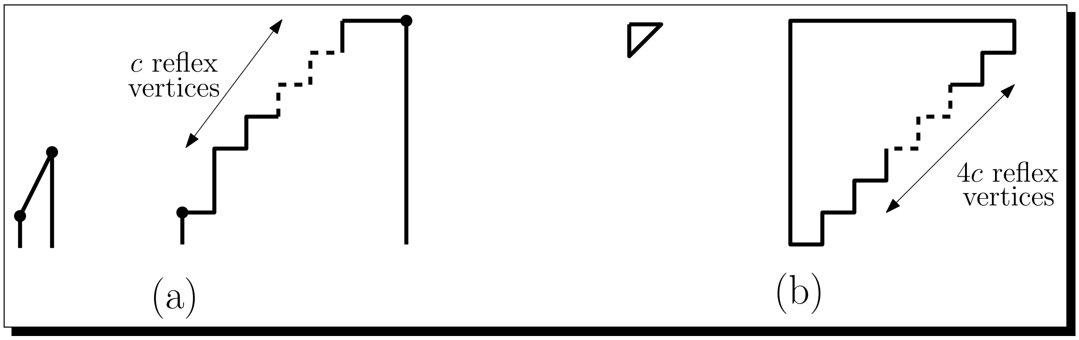

Figure 7.2: (a) **Left:** a high-level view of a normal staircase. **Right:** a detailed description of a normal staircase. (b) **Left:** a high-level view of a triangle-negation gadget. **Right:** the detailed description of a triangle-negation gadget.

If $\bar{x}_i \in C_j$, then we locate a pair of holes inside $\texttt{corr}(X_i, C_j)$ that serve as a negation gadget. The dashed rectangles and the triangles within the middle and bottom corridors of the variable gadget shown in Figure 7.1 are examples of negation gadgets (i.e., each dashed triangle and rectangle visible to that triangle constitute a negation gadget). The dashed rectangle, called *rectangle-negation hole*, is located below the variable hole inside $\texttt{corr}(X_i, C_j)$. The triangle, called *triangle-negation hole*, is located on the left and above the variable hole such that no vertical or horizontal line segment inside $\texttt{VG}(X_i)$ can intersect both the triangle-negation hole and the variable hole or both the triangle-negation hole and the rectangle-negation hole. Note that the two upper vertices of the rectangle-negation hole have the same y-coordinate as the lowest reflex vertex of the normal staircase inside $\texttt{corr}(X_i, C_j)$. Moreover, the x-coordinate of the left-upper vertex of the rectangle-negation hole is less than the x-coordinate of the lower-left vertex of the variable hole inside $\texttt{corr}(X_i, C_j)$.

Each triangle-negation gadget is a reverse staircase consisting of $4c$ steps. Figure 7.2(b) shows the details of a triangle-negation gadget. Finally, recall vertex v, the right-most reflex vertex of $\texttt{VG}(X_i)$ (see Figure 7.1). We call this vertex, the *decision vertex* of $\texttt{VG}(X_i)$.

7.3 Clause Gadgets

We now describe the clause gadgets. Note that in the variable-clause graph, each variable vertex has degree at most three. Moreover, in the comb-shaped drawing of the variable-clause graph, edges might be incident to a variable vertex from both left and right of that vertex. Therefore, we have two possibilities for $\texttt{corr}(X_i, C_j)$. Figure 7.3 shows a part of a clause gadget whose corresponding clause vertex is on the left of the variable vertices. This means that, the clause gadget lies to the left of the variable gadgets. We call such clause gadget a *left clause gadget*. We call the clause gadget that lies to the right of the variable gadgets a *right clause gadget*. When it does not matter, we omit the left or right prefix when referring to a clause gadget.

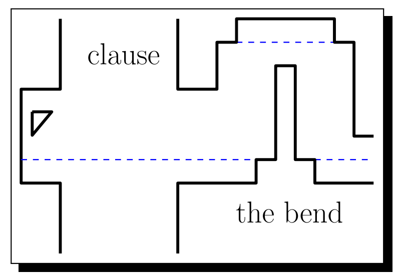

Figure 7.3: A part of a left clause gadget.

Figure 7.3 shows only a part of a left clause gadget. To describe the complete gadget, we extend the open vertical half-segments upwards and downwards until we connect the three corridors that come from the variables contained in this clause. Then, we connect the two half-segments by a horizontal line segment in the top and bottom parts of the clause gadget. According to the structure of a comb-shaped drawing of the variable-clause graph, the three corridors connecting variables to a clause must all be connected to only from left or right of the clause gadget.

In the opposite side of a corridor connected to a clause gadget, we locate a reverse staircase inside the clause gadget headed towards the corridor (see the triangle in Figure 7.3). The reverse staircases inside a clause gadget have $2c$ steps. We create a bend in the middle of the corridor connecting a variable gadget to a left clause gadget as shown in Figure 7.3. There are four separate steps on the corners of the bend. The steps are created such that no vertical line segment inside the corridor can intersect two of them at the same time.

The structure of a right clause gadget is similar to that of a left clause gadget. Figure 7.4 shows an example of a right clause gadget. Since we have to bend the corridor connecting a variable gadget to a right clause gadget we do not create any additional bend inside the corridor. There are two separate steps on the corners of the bend of a right clause gadget. See Figure 7.4. Recall that P is the polygon (with holes) given by the union of the variable and clause gadgets. Finally, we set c to a constant greater than the number of reflex vertices of P that are neither on a staircase nor on a hole of P. In other words, c is a constant greater than the number of reflex vertices of P'; a simple polygon obtained from P by removing the all holes

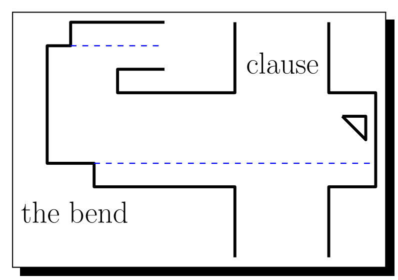

Figure 7.4: A part of a right clause gadget.

and the staircases of P. We are now ready to prove Lemma 7.1.1.

7.4 Proof of Lemma 7.1.1

We first assume that I is satisfiable. We give a CR partition of P that has stabbing number at most $5c$. For each variable X_i; if X_i is true, then we add V_v to the partition, where v is the decision vertex of $\texttt{VG}(X_i)$. Similarly, if X_i is false, then we add H_v to the partition, where v is the decision vertex of $\texttt{VG}(X_i)$.

1. If X_i is true, then V_v forces all reverse staircases of $\texttt{VG}(X_i)$ to be partitioned vertically except for the reverse staircase on the boundary of $\texttt{VG}(X_i)$ (i.e., the upper-most staircase of $\texttt{VG}(X_i)$). Thus, the normal staircases that are headed towards these reverse staircases are forced to be partitioned vertically. Therefore, the vertical edge that passes through exactly one of the vertices of the staircase located on the boundary of $\texttt{VG}(X_i)$ (i.e., V_u in Figure 7.1) passes through the two right walls of $\texttt{VG}(X_i)$. This forces the upper-most staircase of $\texttt{VG}(X_i)$ to be partitioned vertically, which implies that all staircases of $\texttt{VG}(X_i)$ must be partitioned vertically. It is easy to see that no vertical or horizontal line segment inside $\texttt{VG}(X_i)$ can stab the rectangles generated by partitioning more than four staircases at the same time. Now, let `cord` be a corridor of $\texttt{VG}(X_i)$.

    - If there is no negation gadget inside `cord`, then we add additional vertical partition edges to partition `cord`. The steps inside the bend of `cord` force the bend and, consequently, the reverse staircase of the clause gadget, which is headed towards `cord`, to be partitioned vertically. Therefore, any horizontal line segment through the corridor stabs at most $3c$ rectangles.

- If there are the negation gadgets inside `cord`, then we add H_u for every reflex vertex of the triangle-negation and the rectangle-negation holes. This forces the steps inside the bend of `cord` and, therefore, the reverse staircase of the clause gadget headed towards `cord` to be partitioned horizontally. Note that since I is satisfiable it is not possible for all three reverse staircases inside the clause gadget to be partitioned horizontally. Moreover, we observe that no horizontal or vertical line segment inside the $\texttt{VG}(X_i)$ can stab all rectangles generated by partitioning a triangle-negation and a staircase simultaneously.

Therefore, we conclude that the stabbing number of the CR partition of $\texttt{VG}(X_i)$ is at most $5c$.

2. If X_i is false, then we can use an analogous argument as when X_i is true to show that all staircases of $\texttt{VG}(X_i)$ must be partitioned horizontally. Then, it is easy to see that no vertical or horizontal line segment inside $\texttt{VG}(X_i)$ can stab the rectangles generated by partitioning more than four staircases at the same time. Now, let `cord` be a corridor of $\texttt{VG}(X_i)$.

   - If there is no negation gadget inside the corridor, then we add additional horizontal partition edges to partition the corridor. By an analogous argument as in the first part of Case 1, we can show that the entire corridor and the reverse staircase of the clause gadget headed towards the corridor must be partitioned horizontally. Again, since I is satisfiable it is not possible that all three reverse staircases inside the clause gadget to be partitioned horizontally. Therefore, any vertical line segment through the clause gadget stabs at most $5c$ rectangles.

- If there are the negation gadgets inside `cord`, then we add V_u for every reflex vertex on the triangle-negation and the rectangle-negation holes. By an analogous argument as in the second part of Case 1, we can show that the entire corridor and, consequently, the reverse staircase inside the clause gadget connected to `cord` must be partitioned vertically.

Therefore, the stabbing number of the CR partition of $\texttt{VG}(X_i)$ is at most $5c$. This completes the first part of the proof.

Now, we assume that we are given a CR partition of P that has stabbing number at most $5c$. We give a truth assignment for I as follows. For each variable X_i, we set X_i to *true* (resp., to $false$) if and only if the partition has V_v (resp., has H_v), where v is the decision vertex of $\texttt{VG}(X_i)$.

Let $C(X_i) \in \{x_i, \bar{x}_i\}$ denote the literal of X_i that is appeared in the clause C. We denote the value of a literal x_i by $\texttt{val}(x_i)$. Suppose by a contradiction that this assignment does not result in a truth value for I. Thus, there exists a clause $C = (X_i, X_j, X_k)$ such that $\texttt{val}(C(X_i)) = \texttt{val}(C(X_j)) = \texttt{val}(C(X_k)) = false$. In the following, we decide on $C(X_i)$. The procedure for $C(X_j)$ and $C(X_k)$ is analogous.

1. If $C(X_i) = x_i$, then H_v is present in $\texttt{VG}(X_i)$ because $\texttt{val}(C(X_i)) = false$.

Therefore, all staircases inside $\texttt{VG}(X_i)$ must have been partitioned horizontally. Since $C(X_i) = x_i$ there is no negation gadget in $\texttt{corr}(X_i, C)$. Thus, the lowest reflex vertex of the normal staircase, which belongs to the t-shaped hole in the begin of $\texttt{corr}(X_i, C)$, is forced to be an endpoint of a horizontal partition edge of the partition. This horizontal partition edge will go through $\texttt{corr}(X_i, C)$ and force the steps inside the bend of $\texttt{corr}(X_i, C)$ to remain partitioned horizontally. Therefore, we must have the partition edge of $\texttt{corr}(X_i, C)$ that goes through the interior of C and passes below the reverse staircase of C, which is headed towards $\texttt{corr}(X_i, C)$. It follows that, in this case, this reverse staircase must been partitioned horizontally.

2. If $C(X_i) = \bar{x}_i$, then V_v is present in $\texttt{VG}(X_i)$ because $\texttt{val}(C(X_i)) = false$. Since V_v is present in $\texttt{VG}(X_i)$ all staircases of $\texttt{VG}(X_i)$ have been partitioned vertically. Since $C(X_i) = \bar{x}_i$ there are negation gadgets (i.e., the triangle-negation and the rectangle-negation holes) inside $\texttt{corr}(X_i, C)$. The triangle-negation hole inside $\texttt{corr}(X_i, C)$ must have been partitioned horizontally. Otherwise, there is a horizontal line segment inside the corridor (consider the horizontal line segment that passes through the gap between the variable hole and the rectangle-negation hole of $\texttt{corr}(X_i, C)$) that stabs the all rectangles generated by the triangle-negation hole and the reverse staircase on the t-shaped hole just above $\texttt{corr}(X_i, C)$. It follows that the stabbing number of the given rectangular partition is greater than $5c$, which is a contradiction. The horizontal rectangles generated by the triangle-negation hole block the upper-left vertex of the rectangle-negation hole to be an endpoint of a vertical partition

edge of the partition. Therefore, the partition edge through this vertex must be horizontal. This horizontal partition edge has forced the steps inside the bend of $\texttt{corr}(X_i, C)$ to be partitioned horizontally. It follows that, in this case, the reverse staircase must be partitioned horizontally, as well.

We conclude that if $\texttt{val}(C(X_i)) = false$, then the reverse staircase inside the clause C, which is headed towards $\texttt{corr}(X_i, C)$, must be partitioned horizontally. Since $\texttt{val}(C(X_i)) = \texttt{val}(C(X_j)) = \texttt{val}(C(X_k)) = false$, we conclude that all the reverse staircases inside C must been partitioned horizontally. Since each reverse staircase inside a clause gadget consists of $2c$ stairs, there exists a vertical line segment inside C that stabs more than $5c$ rectangles, which is a contradiction. This completes the second part of the proof. □

By Lemma 7.1.1, we obtain the main result of this chapter:

Theorem 7.4.1. MSN WITH HOLES *is* NP-*hard.*

Chapter 8

Algorithmic Results

In this chapter, we give two algorithmic results on the MSN problem. In Section 8.1, we give an $O(n \log n)$-time algorithm that solves the MSN problem on histograms exactly, where n is the number of the vertices of the histogram. We then give a polynomial-time 2-approximation algorithm for the MSN problem on orthogonal polygons (see Section 8.2) using a linear programming formulation of the problem.

8.1 An Optimal Algorithm for Histograms

In this section, we present an algorithm for computing an optimal CR partition of a histogram. Recall that a *histogram* (polygon) H is a simple orthogonal polygon that has one edge e that can see every point in P. Equivalently, as defined by Katz and Morgenstern [27], a simple orthogonal polygon P is a vertical (resp., horizontal) histogram if it is monotone with respect to some horizontal (resp., vertical)

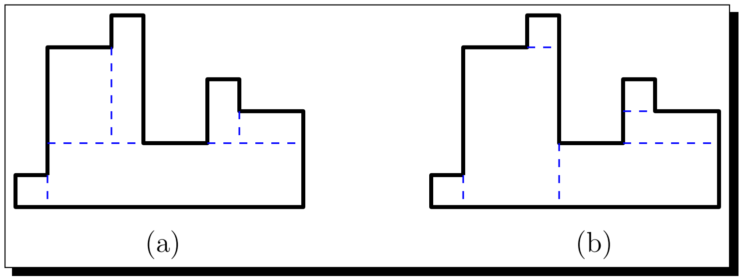

Figure 8.1: A vertical histogram H. (a) An optimal rectangular partition of H with stabbing number 2. (b) Any CR partition of H has stabbing number at least 3.

edge e that spans P; we call e the *base* of H.

Abam et al. [1] gave a polynomial-time algorithm for computing an optimal rectangular partition of a histogram. A rectangular partition of a histogram is not necessarily a CR partition. Figure 8.1(a) shows a histogram whose optimal rectangular partition has stabbing number 2. However, any CR partition of this histogram has stabbing number at least 3; see Figure 8.1(b). Without loss of generality, we assume for the rest of this section that every histogram is a vertical histogram.

Let H be a histogram with n vertices and let H^- denote the set of horizontal edges of H. Recall that every CR partition of H must include at least one of the edges H_u or V_u for every reflex vertex u in H. The algorithm begins with an initial partition of H, consisting exclusively of horizontal partition edges, that will be modified to produce an optimal CR partition of H by greedily replacing horizontal edges with vertical edges. The initial partition of H is obtained by adding the edge H_u for each reflex vertex u.

Observation 8.1.1. *For any CR partition of any vertical histogram H and any reflex vertex u in H, the vertical partition edge V_u may be included at u if and only if no horizontal* *partition edge is included directly below u (otherwise it would intersect V_u).*

Observation 8.1.1 suggests a hierarchical tree structure that determines a partial order in which each horizontal partition edge can be removed and replaced by a vertical partition edge, provided it does not intersect any horizontal partition edge below it. Thus, we construct a forest (initially a single tree denoted T_0) associated with the partition; the algorithm proceeds to update the forest and, in doing so, modifies the associated partition as horizontal partition edges are replaced by vertical ones. Define a tree node for each edge in $H^- \cup S$, where $S = \{H_u \mid u \in V_R(H)\}$. Add an edge between two vertices u and v if some vertical line segment intersects both edges associated with u and v, but no other edge of $H^- \cup S$. When the polygon H is a histogram, the resulting graph, T_0, is a tree. See the example in Figure 8.2(a). We now describe how to construct T_0 in $O(n \log n)$ time. Note that the set S need not be known before construction.

Each edge in H^- is adjacent to two vertical edges on the boundary of H, which we call its left and right neighbours, respectively. Sort the edges of H^- lexicographically, first by y-coordinates and then by x-coordinates. The algorithm sweeps a horizontal line ℓ across H from bottom to top. Initially, ℓ coincides with the base of H; root the tree T_0 at a node u that corresponds to the base of H. The construction refers to a separate balanced search tree [9] that archives the set of vertical edges of H on or below the sweepline, indexed by x-coordinates. Initially, only the leftmost and rightmost vertical edges of H are in the search tree, i.e., the base's neighbours. The construction of the tree T_0 proceeds recursively on u as follows.

Suppose the next edges of H^- encountered by the sweepline ℓ are $e_1, \ldots, e_k$,

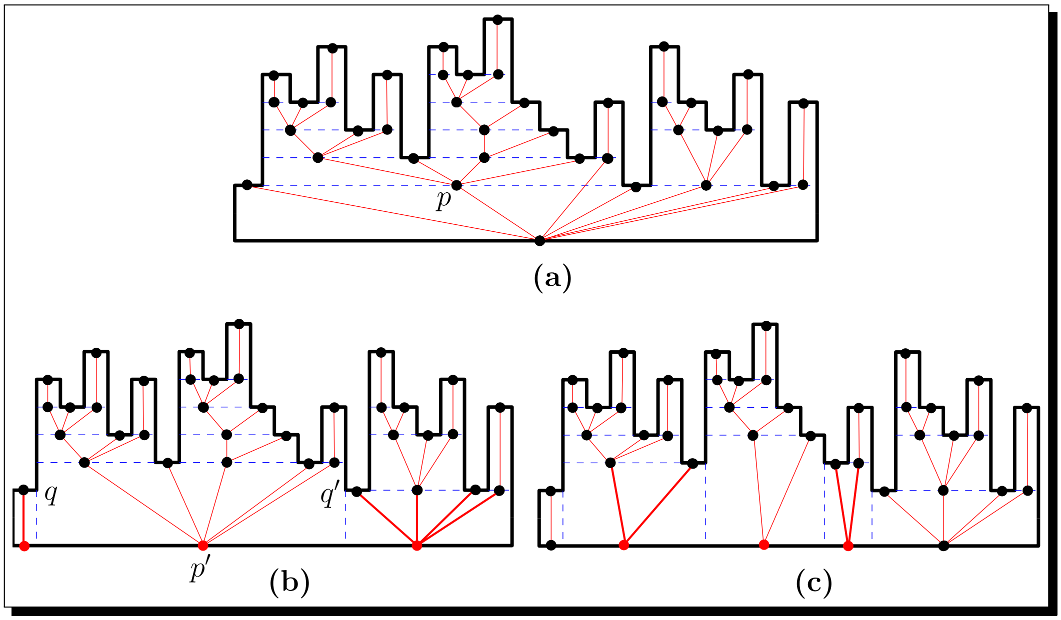

Figure 8.2: **(a)** A histogram H and the tree T_0 that corresponds to the initial partition of H. **(b)** The edge associated with node p is removed from the partition and is replaced by two vertical edges anchored at the reflex vertices q and q'. The red vertices denote the roots of the three new resulting trees. **(c)** The algorithm terminates after one more iteration, giving an optimal CR partition of H (with stabbing number 5) along with the corresponding forest.

each of which has equal y-coordinate. Add the respective left and right neighbours of $e_1, \ldots, e_k$ to the search tree. Let l_1 and r_1 denote the x-coordinates of the respective left and right endpoints of edge e_1. Add a node representing e_1 to T_0 as a child of u. Check whether the left neighbour of e_1 (indexed by l_1) lies below ℓ. If not, then find the predecessor of l_1 in the search tree and let y denote its x-coordinate. Let u' denote the line segment on line ℓ with respective endpoints at the x-coordinates y and l_1. Add a node representing u' to T_0 as a child of u. Recursively construct the subtree of u'. Apply an analogous procedure to the right neighbour of e_1 (indexed by r_1). Repeat for each edge $e_i \in \{e_2, \ldots, e_k\}$. Upon completion, the tree T_0 is constructed storing a representation of the initial horizontal partition (see Figure 8.2(a)). Finally, each tree node stores its height and links to its children in order of x-coordinates; the tree can be updated accordingly after construction. The running time for constructing T_0 is bounded by sorting $O(n)$ edges and a sequence of $O(n)$ searches and insertion on the search tree, resulting in $O(n \log n)$ time to construct T_0.

We now describe a greedy algorithm to construct an optimal CR partition of H using T_0. Observe that the horizontal stabbing number of the initial partition is initially one, whereas its vertical stabbing number corresponds to the height of T_0. The algorithm stores the forest's trees in a priority queue indexed by height. While the vertical stabbing number of H remains greater than its horizontal stabbing number, split the tree of maximum height, say T. To do this, remove the horizontal partition edge stored in a tree node p, where p is a child of the root of T on a longest root-to-leaf path in T. The choice of T and p is not necessarily unique; it suffices to select any tallest tree T and any longest path in T. Observe that p has at least one and possibly two reflex vertices as endpoints, denoted a and b. Remove the horizontal partition edge associated with p and add a vertical partition edge (V_a or V_b) for each neighbour of p that lies above p on the boundary of H. The tree T is then divided into up to three new trees: a) the subtrees of the root of T to the left of p, b) the subtree rooted at p, and c) the subtrees of the root of T to the right of p. The root of each new tree corresponds to the base edge of H. See Figure 8.2(b).

The following observation is straightforward:

Observation 8.1.2. *The horizontal stabbing number of the partition associated with the forest corresponds to the number of trees in the forest, whereas its vertical stabbing number corresponds to the height of the tallest tree in the forest.*

Once the height of the tallest tree becomes less than or equal to the number of trees in the forest, we return either the current partition or the previous partition, whichever has lower stabbing number. The number of steps is $O(n)$, where each step requires $O(\log n)$ time to determine the tree with maximum height using the priority queue [9].

The algorithm's correctness follows from Observations 8.1.1 and 8.1.2, and the fact that reducing the vertical stabbing number requires reducing the height of the tallest tree, which is exactly how the algorithm proceeds, decreasing the height of a tallest tree by one on each step. Therefore, we have the following theorem:

Theorem 8.1.3. *Given a histogram H, an optimal* CR *partition of H can be found in $O(n \log n)$ time, where n is the number of vertices of H.*

8.2 A 2-Approximation Algorithm

In this section, we present a Linear Programming (LP) formulation for the problem of finding an optimal CR partition of a orthogonal polygon, possibly with holes. We show that a simple rounding of the LP relaxation leads to a 2-approximation algorithm for this problem. Our algorithm works even when the input polygon has holes.

For an orthogonal polygon P, we define two binary variables u_h and u_v for every reflex vertex $u \in V_R(P)$ that correspond to H_u and V_u, respectively. Each variable's value ($1 =$ present, $0 =$ absent) determines whether its associated partition edge is included in the partition. If two reflex vertices align, then they share a common variable. For each reflex vertex u in $V_R(P)$, let ℓ_u^- and $\ell_u^|$ be respective maximal horizontal and vertical line segments that pass through $f_\epsilon(u)$ and are completely contained in P, where $f_\epsilon(u)$ denotes an ϵ translation of the point u along the bisector of the interior angle determined by the boundary of P locally at u, for some ϵ less than the minimum distance between any two vertices of P. This perturbation ensures that ℓ_u^- and $\ell_u^|$ lie in the interior of P, as in the definition of stabbing number. See Figure 8.3. Let S_u^- (resp., $S_u^|$) be the set of reflex vertices in $V_R(P)$, like v, such that V_v (resp., H_v) intersects ℓ_u^- (resp., $\ell_u^|$). For each reflex vertex $u \in V_R(P)$, let

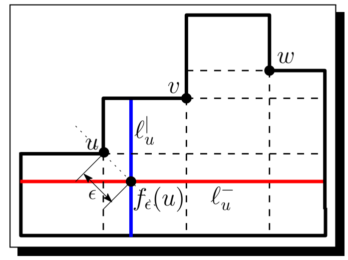

Figure 8.3: The maximal line segments ℓ_u^- and $\ell_u^|$ that pass through the point $f_\epsilon(u)$ are shown in red and blue, respectively. In this example, $u_{\Sigma^-} = 1 + u_v + v_v + w_v$ and $u_{\Sigma^|} = 1 + u_h$.

$$u_{\Sigma^-} = 1 + \sum_{p \in S_u^-} p_v, \quad \text{and} \quad u_{\Sigma^|} = 1 + \sum_{p \in S_u^|} p_h.$$

Thus, u_{Σ^-} and $u_{\Sigma^|}$ denote the number of rectangles stabbed by ℓ_u^- and $\ell_u^|$, respectively, and their maximum values among all reflex vertices u in P correspond to one less than the respective horizontal and vertical stabbing numbers of P. Consequently, the stabbing number of the partition of P determined by the binary variables is

$$1 + \max_{u \in V_R(P)} \{\max\{u_{\Sigma-}, u_{\Sigma|}\}\}. \tag{8.1}$$

A partition divides the polygon into convex regions (more specifically, rectangles) if and only if at least one partition edge is rooted at every reflex vertex. Thus, a CR partition of P corresponds to an assignment of truth values to the set of binary variables such that (i) no two edges of the partition cross, and (ii) for every reflex vertex u, at least one of V_u and H_u is present in the partition.

Therefore, the problem of finding an optimal CR partition can be formulated as a *k-sum integer linear program* as follows:

$$\begin{aligned}
&\text{minimize } (8.1) && && (8.2)\\
&\text{subject to } u_h + u_v \geq 1, && \forall u \in V_R(P),\\
&\quad v_h + u_v \leq 1, && \text{if } H_v \text{ intersects } V_u,\\
&\quad u_h, u_v \in \{0,1\}, && \forall u \in V_R(P).
\end{aligned}$$

To obtain an integer linear program, we introduce an additional variable y. The following integer linear program is equivalent to the above KLP (see Section 6.2):

$$\begin{aligned}
&\text{minimize } y && && (8.3)\\
&\text{subject to } y - u_{\Sigma-} \geq 0 && \forall u \in V_R(P),\\
&\quad y - u_{\Sigma|} \geq 0 && \forall u \in V_R(P),\\
&\quad u_h + u_v \geq 1, && \forall u \in V_R(P),\\
&\quad -v_h - u_v \geq -1, && \text{if } H_v \text{ intersects } V_u,\\
&\quad u_h, u_v \in \{0,1\}, && \forall u \in V_R(P). && (8.4)
\end{aligned}$$

Since the number of sums in (8.1) is $O(n^2)$, the size of the integer linear program above is polynomial in n. Next, we relax the above program by replacing (8.4) with $u_h, u_v \in [0,1], \forall u \in V_R(P)$ and obtain the following LP:

$$\begin{aligned} &\text{minimize } y && && (8.5)\\ &\text{subject to } y - u_{\Sigma^-} \geq 0 && \forall u \in V_R(P), &&\\ &\qquad y - u_{\Sigma^|} \geq 0 && \forall u \in V_R(P), &&\\ &\qquad u_h + u_v \geq 1, && \forall u \in V_R(P), &&\\ &\qquad -v_h - u_v \geq -1, && \text{if } H_v \text{ intersects } V_u, &&\\ &\qquad u_h, u_v \geq 0, && \forall u \in V_R(P). && \end{aligned}$$

We observe that the constraints $u_h, u_v \leq 1$ are redundant since we can reduce any $u_h > 1$ (resp., $u_v > 1$) to u_h=1 (resp., u_v=1) without increasing the value of the objective function for any feasible solution. Let s^* be a solution to the above LP. We round s^* to a feasible solution for our problem as follows. For each vertex $u \in V_R(P)$, let

$$u_h = \begin{cases} 0, & \text{if } s^*(u_h) \leq 1/2, \\ 1, & \text{if } s^*(u_h) > 1/2, \end{cases} \quad \text{and} \quad u_v = \begin{cases} 0, & \text{if } s^*(u_v) < 1/2, \\ 1, & \text{if } s^*(u_v) \geq 1/2. \end{cases} \qquad (8.6)$$

We first show that, for every reflex vertex u, at least one of V_u and H_u is present in the partition.

Lemma 8.2.1. *For each vertex $u \in V_R(P)$, at least one of u_h and u_v is equal to 1 after rounding a solution to (8.5).*

Proof. We give a proof by contradiction. Suppose that after rounding a solution to (8.5), $u_h = u_v = 0$ for some $u \in V_R(P)$. Since $u_h = 0$ by (8.6) we have

$s^*(u_h) \leq 1/2$ and, similarly, since $u_v = 0$ we have $s^*(u_v) < 1/2$. Therefore, $s^*(u_h) + s^*(u_v) < 1$, which contradicts the constraint $u_h + u_v \geq 1$ of (8.5). □

The next lemma proves that no two edges of the partition obtained by the LP cross each other.

Lemma 8.2.2. *If H_v intersects V_u, for two vertices $u, v \in V_R(P)$, then at most one of the variables v_h and u_v is 1 after rounding a solution to the LP.*

Proof. We give a proof by contradiction. Suppose that for two vertices $u, v \in V_R(P)$: (i) H_v intersects V_u, and, (ii) both v_h and u_v are 1 after rounding. Since after rounding v_h=1 by (8.6) we have $s^*(v_h) > 1/2$. Similarly, since after rounding u_v=1 we have $s^*(u_v) \geq 1/2$. Therefore, $s^*(v_h) + s^*(u_v) > 1$, which contradicts the constraint $v_h + u_v \leq 1$ (or equivalently $-v_h - u_v \geq -1$) of the LP. □

By combining Lemmas 8.2.1 and 8.2.2, we get the following result:

Corollary 8.2.3. *The partition determined by a feasible solution to the LP after rounding is a* CR *partition.*

By (8.6), the value of each variable after rounding is at most twice the value of the corresponding variable in the LP solution. Moreover, it is easy to see that the number of constraints in (8.5) is polynomial in $V_R(P)$, allowing a 2-approximate solution to be found in polynomial time [8, 50]. Therefore, we have the following theorem:

Theorem 8.2.4. *There exists a polynomial-time algorithm that constructs a* CR *partition of any given orthogonal polygon P with stabbing number at most twice that of any* CR *partition of P.*

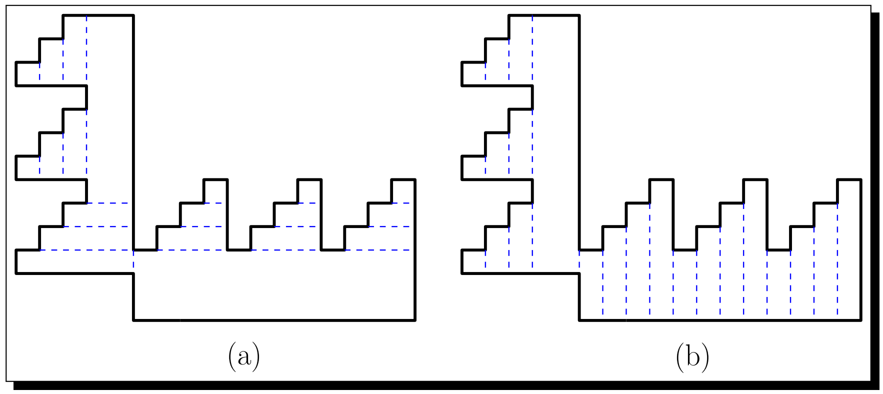

Figure 8.4: A simple orthogonal polygon P for which (a) the optimal partition has stabbing number 4 while (b) assigning V_u (or H_u) to every reflex vertex u of P results in a partition with stabbing number at least 10.

Remark. A preliminary attempt at obtaining a 2-approximation might be to assign to each reflex vertex u its vertical partition edge, V_u (or, equivalently, assigning the horizontal partition edge H_u to each u). Unfortunately, this is not the case: Figure 8.4 shows an orthogonal polygon for which the optimal CR partition has stabbing number 4. However, the partition obtained by assigning V_u (or H_u) consistently to every vertex $u \in V_R(P)$ has stabbing number at least 10. In fact, the polygon in this example can be extended to show that this heuristic does not provide any constant-factor approximation.

Part III

Guarding Orthogonal Terrains

Chapter 9

Background

A *1.5-dimensional terrain* T is an x-monotone polygonal chain in the plane, where $V(T) = \{v_1, \ldots, v_n\}$ is the set of vertices of T ordered from left to right, and $E(T) = \{e_1 = (v_1, v_2), \ldots, e_{n-1} = (v_{n-1}, v_n)\}$ is the set of edges of T induced by the vertex set $V(T)$. Terrain T is called an *orthogonal terrain* if each edge $e \in E(T)$ is either horizontal or vertical. Let p be a point guard on T; p is called a *vertex* guard if $p \in V(T)$. A point q on T is seen/guarded by p (or, p sees/guards q) if and only if every point of the line segment $\overline{pq}$ lies either on or above T.

Given a (not necessarily orthogonal) terrain T, two common types of guarding problems are defined on T. In the *continuous* terrain guarding problem, the objective is to find a minimum-cardinality set S of points on T that *guards* T; that is, for every point $p \in T$, either p is in S or p is guarded by at least one point in S. In the *discrete* terrain guarding problem, on the other hand, two sets P and G of points on T are given along the terrain T as input and the objective is to find a subset $G' \subseteq G$ of minimum cardinality such that G' guards the points in P.

9.1 Problem Definition and Our Result

In this part, we study the Directed Terrain Guarding (DTG) problem. The DTG problem is a variant of the discrete terrain guarding problem on an orthogonal terrain T under *directed visibility* such that $P = G = V(T)$; let $n = |T|$. The directed visibility is defined as follows.

Definition 2. ***(Directed Visibility).*** *Let u be a vertex of T. If u is a reflex vertex, then u sees a vertex v of T if and only if every point in the interior of the line segment $\overline{uv}$ lies* strictly *above T. If u is a convex vertex, then u sees a vertex v of T if and only if $\overline{uv}$ is a non-horizontal line segment that lies on or above T.*

It is possible, under directed visibility, that a vertex u of T sees a vertex v, but vertex v cannot see u; see Figure 9.1(a) for an example. Therefore, we consider the following problem:

Definition 3. ***(The Directed Terrain Guarding (DTG) Problem on Orthogonal Terrains).*** *Given an orthogonal terrain T, compute a subset $S \subseteq V(T)$ of minimum cardinality that guards the vertices of T under directed visibility. That is, for every vertex $u \in V(T)$, either $u \in S$ or u is guarded by at least one vertex in S under directed visibility.*

In this chapter, we present related work on the terrain guarding problem and provide preliminary results on the DTG problem. In Chapter 10, we give an $O(n)$-time algorithm for the DTG problem on orthogonal terrains under directed visibility. To this end, we first reduce the DTG problem to two subproblems such that an exact solution for the DTG problem reduces to the union of exact solutions of the

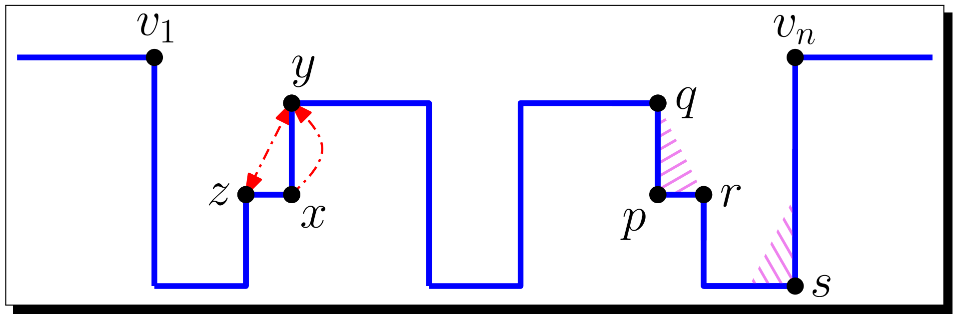

Figure 9.1: An orthogonal terrain T; throughout this part, we assume that the leftmost and rightmost edges of T are two horizontal rays starting from v_1 and v_n, respectively. (a) An illustration of directed visibility: neither vertex y nor z can see vertex x under directed visibility, but they can see each other. The vertex x can see vertex y, but it cannot see vertex z because the line segment $\overline{xz}$ is horizontal. (b) The vertices q and r are reflex while the vertices p and s are convex. Moreover, q and r are both left reflex, p is left convex and s is right convex; vertex v_n is a right reflex vertex.

two subproblems. We then give an $O(n)$-time greedy algorithm for solving each of the subproblems. To the best of our knowledge, this is the first exact algorithm for a nontrivial instance of the terrain guarding problem and partially answers a question posed by Ben-Moshe et al. [4] for orthogonal terrains.

9.2 Related Work

The terrain guarding problem belongs to the well-known family of art gallery problems. See Chapter 2 for an overview of related work on the art gallery problem. Regarding the terrain guarding problem, Ben-Moshe et al. [4] gave the first constant-factor approximation algorithm for this problem and left the complexity of the problem open. King and Krohn [30] showed that both continuous and discrete versions of the terrain guarding problem are NP-hard on arbitrary terrains. A 4-approximation algorithm for the terrain guarding problem was given by Elbassioni et al. [14], and Katz and Roisman [28] gave a 2-approximation algorithm for the OTG problem[1] Gibson et al. [20] gave a polynomial-time approximation scheme (PTAS) for the discrete version of the terrain guarding problem, and a PTAS for the continuous version of the problem was recently given by Friedrichs et al. [18]. To the best of our knowledge, however, the complexity of the DTG problem on orthogonal terrains remains open. We note that the hardness result of King and Krohn [30] does not apply to the DTG problem on orthogonal terrains due to a number of essential differences between arbitrary and orthogonal terrains (e.g., see Lemma 9.3.3).

9.3 Definitions and Preliminary Results

We denote the x- and y-coordinates of a point p on an orthogonal terrain T by $x(p)$ and $y(p)$, respectively. For the rest of this part, we use terms "terrain" and "guard" to refer to an orthogonal terrain and a vertex guard, respectively, unless otherwise stated. Moreover, we simply use "guarding" to mean "guarding under directed visibility" unless otherwise stated.

A vertex u of T is *convex* (resp. *reflex*), if the angle formed by the edges incident

[1]Recall from Chapter 1 that the OTG problem is in fact the DTG problem under standard visibility.

to u above T is $\pi/2$ (resp., $3\pi/2$). We denote the set of convex vertices and reflex vertices of T by $V_c(T)$ and $V_r(T)$, respectively. A convex vertex $v_c \in V_c(T)$ is *left convex* (resp., *right convex*), if the area above the horizontal edge incident to v_c lies to the right (resp., to the left) of the vertical edge incident to v_c. Similarly, a reflex vertex $v_r \in V_r(T)$ is *left reflex* (resp., *right reflex*), if the area below the horizontal edge incident to v_r lies to the left (resp., to the right) of the vertical edge incident to v_r. See Figure 9.1(b) for an example of these definitions. We denote the set of left convex vertices and the set of right convex vertices of T by $V_{lc}(T)$ and $V_{rc}(T)$, respectively. Similarly, the set of left reflex vertices and the set of right reflex vertices of T are denoted by $V_{lr}(T)$ and $V_{rr}(T)$, respectively.

For consistency, we assume that the leftmost and rightmost edges of T are two horizontal rays starting from v_1 and v_n, respectively; see Figure 9.1 for an illustration. For a reflex vertex u of T, we denote the convex vertex directly below u by $B(u)$. We say that a subset M of vertices of T guards a subset M' of vertices of T, where $M \cap M' = \emptyset$, if every vertex in M' is guarded by at least one vertex in M. We first describe some properties of orthogonal terrains.

Observation 1. *Let u and v be two reflex vertices of T. If vertex u sees $B(v)$, then u must also see v; see Figure 9.2 for an illustration.*

Observation 2. *Let u be a reflex vertex of a terrain T. If u is right reflex and sees a left convex vertex v of T, then $x(u) > x(v)$ and $y(u) > y(v)$. Similarly, if u is left reflex and sees a right convex vertex v of T, then $x(u) < x(v)$ and $y(u) > y(v)$.*

We first show that the following property, called *the order claim*, still holds under directed visibility:

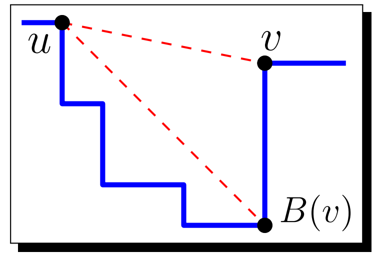

Figure 9.2: If a reflex vertex u sees $B(v)$, for some reflex vertex v, then u must also see vertex v itself.

Lemma 9.3.1 (Ben-Moshe et al. [4]). *Let p, q, r and s be four vertices of a terrain T such that $x(p) < x(q) < x(r) < x(s)$. If p sees r and q sees s, then p sees s.*

Proof. Note that there are three lines of visibilities involved in the order claim: the visibility lines between p and r, between q and s, and between p and s. Since $x(p) < x(q) < x(r) < x(s)$, none of these three lines of visibilities can occur between two adjacent vertices of T and, therefore, directed visibility does not affect the lines of visibilities involved in order claim. This completes the proof of the lemma. □

Lemma 9.3.2. *Let u be a reflex vertex of a terrain T. If u is right reflex (resp., left reflex), then u cannot see any right convex (resp., left convex) vertex of T.*

Proof. We prove the lemma for when u is right reflex; the other case is proved by a symmetric argument. Let v be a right convex vertex of T. If $x(v) = x(u)$, then $v = B(u)$ and, therefore, u cannot see v under directed visibility. If $x(v) \neq x(u)$, then there are two cases: (i) if $y(v) = y(u)$, then v is the adjacent vertex to the right of u and so u cannot see v under directed visibility, and (ii) if $y(v) \neq y(u)$, then it is clearly not possible for u to see v. This completes the proof of the lemma. □

In an arbitrary terrain, it is possible that a reflex vertex can guard both a left and a right convex vertex. For orthogonal terrains, however, this is not the case. This property is stated in the following lemma.

Lemma 9.3.3. *Let u be a right convex vertex and v be a left convex vertex of a terrain T. Then, there is no reflex vertex of T that sees both u and v under directed visibility.*

Proof. By Lemma 9.3.2, (i) no right reflex vertex of T can see u, and (ii) no left reflex vertex of T can see v. Therefore, no reflex vertex of T can see both u and v. This completes the proof of the lemma. □

Chapter 10

An Exact Algorithm for the Directed Terrain Guarding Problem

In this chapter we present our exact $O(n)$-time algorithm for the DTG problem on orthogonal terrains. To this end, in Section 10.1, we show that the DTG problem on T can be reduced to two subproblems such that an exact solution for the DTG problem is equivalent to the union of the exact solutions for the two subproblems. In Section 10.2, we show that each subproblem can be solved exactly in $O(n)$-time using a greedy algorithm.

10.1 Defining Subproblems

In this section, we first define the two subproblems and then will show that an exact solution for the DTG problem is equivalent to the union of the exact solutions for the two subproblems.

Definition 4. ***(The Left-Convex Guarding (LCG(M)) Problem).*** *Given a set $M \subseteq V_{lc}(T)$, the objective of the LCG(M) problem is to compute a minimum-cardinality set $M' \subseteq V(T)$ such that for every vertex $u \in M$, either $u \in M'$ or u is guarded by at least one vertex in M'.*

Definition 5. ***(The Right-Convex Guarding (RCG(M)) Problem).*** *Given a set $M \subseteq V_{rc}(T)$, the objective of the RCG(M) is to compute a minimum-cardinality set $M' \subseteq V(T)$ such that for every vertex $u \in M$, either $u \in M'$ or u is guarded by at least one vertex in M'.*

To compute an exact solution for the DTG problem on T, we first show that we can restrict our attention to solutions that are in standard form. A feasible solution S to the DTG problem on T is in *standard form* if and only if every reflex vertex in S sees at least one convex vertex of T.

Lemma 10.1.1. *For any orthogonal terrain T, there exists an exact solution S for the DTG problem on T that is in standard form.*

Proof. Take any exact solution S_0 for the DTG problem on T. We construct a feasible solution S from S_0 such that $|S| \leq |S_0|$ and S is in standard form. To this end, for each reflex vertex $u \in S_0$ that does not see any convex vertex of T, replace u with $B(u)$ (i.e., the convex vertex directly below u). Let S be the resulting set. Clearly, $|S| \leq |S_0|$ and every reflex vertex in S sees at least one convex vertex of T. We now show that S is a feasible solution for the DTG problem on T. Consider a reflex vertex $u \in S_0$ that was replaced by $B(u)$ in S and let $\texttt{Vis}(u)$ be the set of vertices of T that are seen by u. We next prove that every vertex in $\texttt{Vis}(u)$ is still guarded by at least one vertex in S. First, note that every vertex in $\texttt{Vis}(u)$ is a reflex vertex. Let $v \in \texttt{Vis}(u)$ and consider $B(v)$. If $B(v) \in S$, then v is guarded by at least one vertex in S (i.e., the vertex $B(v)$). If $B(v) \notin S$, then there must be a reflex vertex $w \in S_0$ that guards $B(v)$. We note that $w \in S$ because w sees at least one convex vertex of T and so we have not replaced it with $B(w)$ in S. By Observation 1, vertex $w \in S$ guards v and, therefore, S is a feasible solution. Since $|S| \leq |S_0|$, the set S is an exact solution for the DTG problem on T that is in standard form. This completes the proof of the lemma. □

The following lemma states a necessary and sufficient condition for solving the DTG problem on T.

Lemma 10.1.2. *Let S be a feasible solution for the DTG problem on T. The set S is an exact solution if and only if there exists a partition $\{S_L, S_R\}$ of S such that (i) the set S_L is an exact solution for the LCG($V_{lc}(T)$) problem on T, and (ii) the set S_R is an exact solution for the RCG($V_{rc}(T)$) problem on T.*

Proof. (⇒) Let S be an exact solution for the DTG problem on T; by Lemma 10.1.1, we assume that S is in standard form. Let $S_L \subseteq S$ such that $u \in S_L$ if and only if u is either a left convex vertex or it is a right reflex vertex of T. Similarly, let $S_R \subseteq S$ such that $v \in S_R$ if and only if v either is a right convex vertex or it is a left reflex vertex of T that sees at least one right convex vertex. Since S is in standard form, $\{S_L, S_R\}$ is a partition of S.

We first prove that S_L is a feasible solution for the LCG($V_{lc}(T)$) problem on T. Let a be a left convex vertex of T. If $a \in S$, then $a \in S_L$. If $a \notin S$, then by Lemma 9.3.2 and the fact that no convex vertex can see another convex vertex we conclude that there must be a right reflex vertex $b \in S$ that guards a and, therefore, $b \in S_L$. This means that for every left convex vertex a of T, we have either $a \in S_L$ or a is guarded by at least one vertex in S_L. Therefore, S_L is a feasible solution for the LCG($V_{lc}(T)$) problem on T. By an analogous argument, we can show that S_R is a feasible solution for the RCG($V_{rc}(T)$) problem on T.

We next prove that S_L is an exact solution for the LCG($V_{lc}(T)$) problem on T. Suppose for a contradiction that there exists a feasible solution S'_L for the LCG($V_{lc}(T)$) problem on T such that $|S'_L| < |S_L|$. In the following, we prove that the set $\{S'_L \cup S_R\}$ is a feasible solution for the DTG problem on T, which is a contradiction to the fact that S is an exact solution for the DTG problem on T because $|S'_L \cup S_R| \leq |S'_L| + |S_R| < |S_L| + |S_R| = |S|$ (the last equality follows from the fact that $\{S_L, S_R\}$ is a partition of S). Let u be a vertex of T. If u is left convex, then u is either in S'_L or it is guarded by a right reflex vertex in S'_L because S'_L is feasible solution for the LCG($V_{lc}(T)$) problem on T. Similarly, if u is a right convex vertex, then u is either in S_R or it is guarded by a left reflex vertex in S_R because S_R is feasible solution for the RCG($V_{rc}(T)$) problem on T. Now, suppose that u is a reflex vertex that is not in $S'_L \cup S_R$. Then, consider the vertex $B(u)$. If $B(u) \in \{S'_L \cup S_R\}$, then u is guarded by at least one vertex in $S'_L \cup S_R$ (i.e., the vertex $B(u)$). If $B(u) \notin \{S'_L \cup S_R\}$, then it must be guarded by a reflex vertex $w \in \{S'_L \cup S_R\}$. By Observation 1, vertex w must also guard the vertex u. We have proved that every vertex of T that is not in $S'_L \cup S_R$ is guarded by at least one vertex in $S'_L \cup S_R$ and, therefore, $S'_L \cup S_R$ is a feasible solution for the DTG problem on T. By an analogous argument, we can show that S_R is an exact solution for the RCG($V_{rc}(T)$) problem on T.

($\Leftarrow$) Suppose that there exists a partition $\{S_L, S_R\}$ of S such that S_L is an exact solution for the LCG($V_{lc}(T)$) problem on T and S_R is an exact solution for the RCG($V_{rc}(T)$) problem on T. We now prove that $S = \{S_L \cup S_R\}$ is an exact solution for the DTG problem on T. Suppose for a contradiction that there exists a feasible solution S' for the DTG problem on T such that $|S'| < |S|$; by Lemma 10.1.1, we assume that S' is in standard form. Let X be a subset of S' such that $u \in X$ if and only if u is either a left convex vertex or it is a right reflex vertex of T. Similarly, let Y be a subset of S' such that $v \in S_R$ if and only if v either is a right convex vertex or it is a left reflex vertex of T. Since S' is in standard form, $\{X, Y\}$ is a partition of S'. Since $|S'| < |S|$, we must have $|X| < |S_L|$ or $|Y| < |S_R|$. Without loss of generality, assume that $|X| < |S_L|$. In the following, we show that X is a feasible solution for the LCG($V_{lc}(T)$) problem on T, which is a contradiction to the fact that S_L is an exact solution for the LCG($V_{lc}(T)$) problem on T. To show the feasibility of X, let x be a left convex vertex of T. If $x \in S'$, then $x \in X$. If $x \notin S'$, then we conclude by Lemma 9.3.2 that there must be a right reflex vertex $y \in S'$ that guards x and so we have $y \in X$. This means that every left convex vertex of T is either in X or it is guarded by at least one right reflex vertex in X. Therefore, the set X is a feasible solution for the LCG($V_{lc}(T)$) problem on T.

We have proved that the set S', where $|S'| < |S|$, does not exist and, therefore, the set S is an exact solution for the DTG problem on T. This completes the proof of the lemma. □

By Lemma 10.1.2, we have the following theorem.

Theorem 10.1.3. *To solve the DTG problem on T, it is sufficient to solve the LCG($V_{lc}(T)$)*

and the RCG($V_{rc}(T)$) problems on T.

10.2 Solving the LCG($V_{lc}(T)$) Problem

In this section, we present an $O(n)$-time exact algorithm for the LCG($V_{lc}(T)$) problem on T; an exact algorithm for the RCG($V_{rc}(T)$) problem can be derived analogously. Note that no convex vertex of T can see one other convex vertex of T and, by Lemma 9.3.2, no left reflex vertex of T can see a left convex vertex of T. Therefore, we have the following result.

Lemma 10.2.1. *If M is a feasible solution for the LCG($V_{lc}(T)$) problem on T, then $M \subseteq \{V_{lc}(T) \cup V_{rr}(T)\}$.*

We first show that we can restrict our attention to solutions that are in a standard form. A feasible solution M for the LCG($V_{lc}(T)$) problem on T is in *standard form* if and only if a left convex vertex u is in M if and only if no reflex vertex of T can see u.

Lemma 10.2.2. *For any orthogonal terrain T, there exists an exact solution M for the LCG($V_{lc}(T)$) problem on T that is in standard form.*

Proof. Take any exact solution M_0 for the LCG($V_{lc}(T)$) problem on T. We construct a feasible solution M from M_0 such that $|M| \leq |M_0|$ and M is in standard form. For every left convex vertex $u \in M_0$ that is seen by at least one right reflex vertex v of T, replace u with v; let M be the resulting set. Clearly, $|M| \leq |M_0|$. Moreover, M is a feasible solution for the LCG($V_{lc}(T)$) problem on T because (i) the vertex u is now guarded by v, and (ii) the vertex u, which is left convex, cannot see any other left convex vertex of T. Therefore, every left convex vertex of T is still guarded by at least one vertex in M. Since $|M| \leq |M_0|$ and no left convex vertex of T that is in M is seen by a right reflex vertex of T, we conclude that M is an exact solution for the LCG($V_{lc}(T)$) problem on T that is in standard form. □

To solve the LCG($V_{lc}(T)$) problem on T, we next give a characterization for an exact solution of the LCG($V_{lc}(T)$) problem on T. The following lemma is similar to the one given in Lemma 10.1.2 for the DTG problem.

Lemma 10.2.3. *Let M be a feasible solution for the LCG($V_{lc}(T)$) problem on T. The set M is an exact solution if and only if there exists a partition $\{A, B\}$ of M such that (i) $u \in A$ if and only if u is a left convex vertex and no reflex vertex of T can see u, and (ii) $B = M \setminus A$ is a minimum-cardinality subset of $V_{rr}(T)$ that guards $V_{lc}(T) \setminus A$.*

Proof. ($\Rightarrow$) Suppose that M is an exact solution for the LCG($V_{lc}(T)$) problem on T; by Lemma 10.2.2, we assume that M is in standard form. Let A be the subset of M such that $u \in A$ if and only if u is a left convex vertex of T, and let $B = M \setminus A$. Clearly, $\{A, B\}$ is a partition of M. Also, no reflex vertex of T can see a vertex in A because M is in standard form and, by Lemma 10.2.1, we have that $B \subseteq V_{rr}(T)$. Moreover, since M is a feasible solution for the LCG($V_{lc}(T)$) problem, every left convex vertex of T that is not in A is guarded by at least one right reflex vertex in B. Therefore, it only remains to show that B has minimum cardinality among all subsets of $V_{rr}(T)$ that guard $V_{lc}(T) \setminus A$. Suppose for a contradiction that $B' \subseteq V_{rr}(T)$ guards $V_{lc}(T) \setminus A$ such that $|B'| < |B|$. Then, $\{A \cup B'\}$ is a feasible solution for the LCG($V_{lc}(T)$) problem on T, but $|A \cup B'| \leq |A| + |B'| < |A| + |B| = |M|$ (the last equality is due to the fact that $\{A, B\}$ is a partition of M); this is a contradiction to the fact that M is an exact solution for the LCG($V_{lc}(T)$) problem on T.

($\Leftarrow$) Suppose that there exists a partition $\{A, B\}$ of M such that (i) $u \in A$ if and only if u is a left convex vertex and no reflex vertex of T can see u, and (ii) $B = M \setminus A$ is a minimum-cardinality subset of $V_{rr}(T)$ that guards $V_{lc}(T) \setminus A$. We now show that $M = \{A \cup B\}$ is an exact solution for the LCG($V_{lc}(T)$) problem on T. Suppose for a contradiction that there exists a feasible solution M' for the LCG($V_{lc}(T)$) problem on T such that $|M'| < |M|$. By Lemma 10.2.1, we have that $M' \subseteq \{V_{lc}(T) \cup V_{rr}(T)\}$. Partition M' into two sets X and Y such that $x \in X$ if and only if x is a left convex vertex that is not seen by any right reflex vertex of T, and let $Y = M' \setminus X$. We can assume that $Y \subseteq V_{rr}(T)$ because otherwise we can replace every left convex vertex y in Y with a right reflex vertex of T that sees y.[1] Recall that if $x \in X$, then no right reflex vertex of T can see x and, by Lemma 9.3.2, no left reflex vertex of T can see x; therefore, $x \in A$ because no reflex vertex of T can see x and M is a feasible solution for the LCG($V_{lc}(T)$) problem on T. By an analogous argument, we can show that if $x \in A$, then $x \in X$. Therefore, $X = A$. This means that Y is a subset of $V_{rr}(T)$ that guards $V_{lc}(T) \setminus X = V_{lc}(T) \setminus A$. Since $X = A$ and $|M'| < |M|$, we must have that $|Y| < |B|$, which is a contradiction to the fact that B is a minimum-cardinality subset of $V_{rr}(T)$ that guards $V_{lc}(T) \setminus A$. This completes the proof of the lemma. □

A similar result can be derived for an exact solution of the RCG($V_{rc}(T)$) problem analogously.

[1] Note that at least one such right reflex vertex of T exists because otherwise we would have added y into X.

Lemma 10.2.4. *Let M be a feasible solution for the RCG($V_{rc}(T)$) problem on T. The set M is an exact solution if and only if there exists a partition $\{P, Q\}$ of M such that (i) $u \in P$ if and only if u is a right convex vertex and no reflex vertex of T can see u, and (ii) $Q = M \setminus P$ is a minimum-cardinality subset of $V_{lr}(T)$ that guards $V_{rc}(T) \setminus P$.*

By Lemma 10.2.3 and Lemma 10.2.4, we have the following theorem.

Theorem 10.2.5. *To solve the LCG($V_{lc}(T)$) problem on T, it is sufficient to first find the subset A of $V_{lc}(T)$, where $u \in A$ if and only if no reflex vertex of T can see u, and then compute a minimum-cardinality subset B of $V_{rr}(T)$ that guards $V_{lc}(T) \setminus A$. Similarly, to solve the RCG($V_{rc}(T)$) problem on T, it is sufficient to first find the subset P of $V_{rc}(T)$, where $u \in P$ if and only if no reflex vertex of T can see u, and then compute a minimum-cardinality subset Q of $V_{lr}(T)$ that guards $V_{rc}(T) \setminus P$.*

For the rest of this chapter, we show how to compute an exact solution for the LCG($V_{lc}(T)$) problem on T; an exact solution for the RCG($V_{rc}(T)$) problem on T can be computed analogously. By Theorem 10.2.5, we first compute the set A, where $u \in A$ if and only u is a left convex vertex and it is not seen by any reflex vertex of T, and then let $C = V_{lc}(T) \setminus A$. We now give an $O(n)$-time greedy algorithm for the problem of guarding C with the minimum-cardinality subset B of $V_{rr}(T)$.

For each left convex vertex $u \in C$, let $R(u)$ be the righmost right reflex vertex of T (i.e., the rightmost vertex in $V_{rr}(T)$) that sees u. Consider the left convex vertices of C from right to left: for each left convex vertex u in order, if u is not yet guarded by a reflex vertex in B, then we add $R(u)$ into B. Clearly, B is a feasible solution for guarding the vertices in C. Let B' be the set of convex vertices that force the algorithm to add a new guard into B. Clearly, $|B'| = |B|$. We now show that no

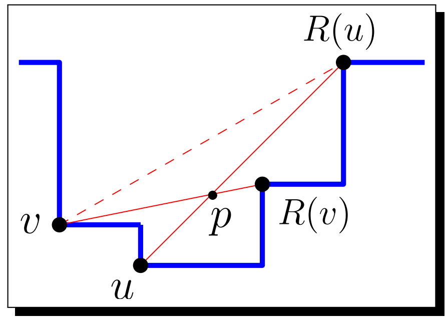

Figure 10.1: An illustration in support for the proof of Lemma 10.3.1.

right reflex vertex of T can see two vertices in B', which proves that the set B is an exact solution. Suppose for a contradiction that there exists a right reflex vertex v that sees two vertices w_i and w_j in B'. Without loss of generality, assume that $x(w_i) > x(w_j)$; that is, vertex w_i is guarded before vertex w_j in the ordering. Since v sees w_i, we must have that $x(R(w_i)) \geq x(v)$. Note that $x(R(w_i)) \neq x(v)$ because otherwise we would have not added a new guard for w_j. Therefore, we have the ordering $x(w_j) < x(w_i) < x(v) < x(R(w_i))$ such that w_j sees v and w_i sees $R(w_i)$. But, by Lemma 9.3.1, this means that w_j is seen by $R(w_i)$ which is a contradiction. We have proved that no right reflex vertex of T can see two convex vertices in B' and so the set B is an exact solution for guarding the vertices in C.

10.3 Algorithmic Details

In this section, we show how to implement the algorithm in time linear in n, the number of vertices of T. Our implementation of the algorithm uses the following result.

Lemma 10.3.1. *Let u and v be two left convex vertices of T such that $x(v) < x(u)$. Then, the line segments $\overline{uR(u)}$ and $\overline{vR(v)}$ do not intersect at an interior point.*

Proof. Suppose for a contradiction that the line segments $\overline{uR(u)}$ and $\overline{vR(v)}$ intersect at an interior point p. Since $x(v) < x(u)$, we must have that $x(R(v)) < x(R(u))$. Therefore, we have the ordering $x(v) < x(u) < x(R(v)) < x(R(u))$; see Figure 10.1 for an example. By Lemma 9.3.1, the vertex v must see vertex $R(u)$, which is a contradiction to the fact that $R(v)$ is the righmost right reflex vertex of T that sees v. This completes the proof of the lemma. □

Consider the left convex vertices of T from right to left and let u and v be two left convex vertices such that $x(v) < x(u)$. By Lemma 10.3.1, vertex $R(v)$ cannot lie between the vertices u and $R(u)$; that is, vertex $R(v)$ is either $R(u)$ or a vertex to the right of $R(u)$, or it is a vertex to the left of vertex u. This property leads us to a linear-time algorithm for computing $R(u)$ for all the left convex vertices u in C as follows. Consider the vertices in $\{C \cup V_{rr}(T)\}$ from right to left in order. Note that the first vertex must be a right reflex vertex r. Moreover, we assume that the second vertex is also right reflex; otherwise, we set $R(u)$ to r for every visited left convex vertex until we reach to a right reflex vertex s; we push r and s into a stack $\texttt{S}$ in the order they have been visited. In the following, let s and r be the vertices on top of the stack $\texttt{S}$. Moreover, let t be the next visited vertex and let α be the angle formed by the line segments $\overline{ts}$ and $\overline{sr}$ that faces above T:

- if t is right reflex, then we pop the two vertices s and r from $\texttt{S}$. If $\alpha > \pi$, then we push the three vertices r, s and t into the stack $\texttt{S}$; otherwise, we ignore vertex s and push only vertex r into $\texttt{S}$. Now, we repeat the same procedure with the current two top vertices s' and r' of $\texttt{S}$ until α becomes greater than π in which case we push the three vertices r', s' and t into $\texttt{S}$.

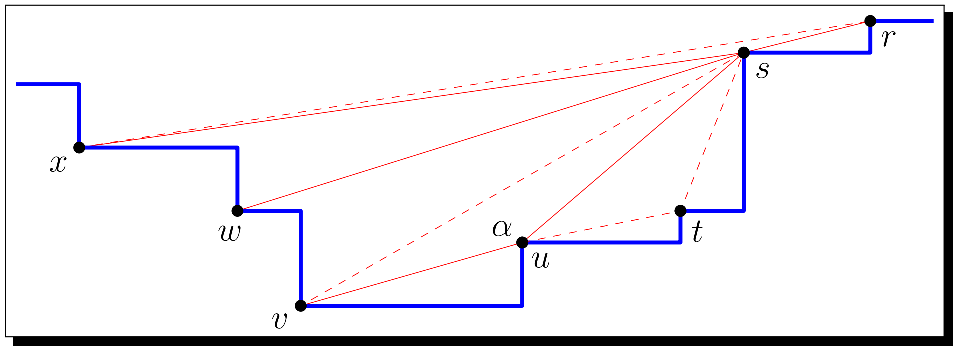

Figure 10.2: An example illustrating the computation of $R(v)$, $R(w)$ and $R(x)$. After processing vertex u, the status of the stack `S` from top to bottom is: $[u, s, r]$. When processing vertex v, vertex u is removed from `S` since $\alpha < \pi$ for the line segments $\overline{vu}$ and $\overline{us}$; then $R(v)$ is set to s. Vertex $R(w)$ is also set to s because $\alpha > \pi$ for the line segments $\overline{ws}$ and $\overline{sr}$. Finally, vertex s is removed from `S` and $R(x)$ is set to r. The final status of `S` is: $[r]$.

- if t is left convex, then we pop the two vertices s and r from `S`. If $\alpha > \pi$, then we set $R(t)$ to s and push vertices r and s back into the stack `S`; otherwise, we ignore vertex s and push only vertex r into `S`. Now, we repeat the same procedure with the current two top vertices s' and r' of `S` until α becomes greater than π in which case we set $R(t)$ to s' and push r' and s' into the stack `S`.

See Figure 10.2 for an example of the algorithm. Let u be a right reflex vertex of T. If $\alpha > \pi$, then we process u in $O(1)$ time and move to the next vertex. If $\alpha \leq \pi$, then one vertex is removed from the stack `S` and we then repeat the same procedure which may consist of removing further vertices from `S`. Therefore, at each right reflex vertex u, either we perform an $O(1)$-time operation or we remove a set S_u of vertices from $\mathtt{S}$ permanently. Note that by Lemma 10.3.1, the vertices in S_u will not be pushed back into $\mathtt{S}$ in the future. We can show using an analogous argument that at each left convex vertex, either we perform an $O(1)$-time operation or we remove a set of vertices from $\mathtt{S}$ permanently. Therefore, the overall procedure of computing $R(u)$ for all the left convex vertices u in C can be completed in $O(n)$ time. Therefore, we have the following theorem.

Theorem 10.3.2. *The LCG($V_{lc}(T)$) problem on T can be solved in $O(n)$ time, where $n = |V(T)|$.*

We note that the RCG($V_{rc}(T)$) problem on T can be solved analogously in $O(n)$ time. Let S_1 and S_2 be the exact solutions for the LCG($V_{lc}(T)$) and the RCG($V_{rc}(T)$) problems on T, respectively. By Theorem 10.1.3, the set $S = \{S_1 \cup S_2\}$ is an exact solution for the DTG problem on T and, therefore, we have the main result of this chapter.

Theorem 10.3.3. *There exists an $O(n)$-time exact algorithm for the DTG problem on any orthogonal terrain T with n vertices.*

Part IV

Conclusion and Future Work

Chapter 11

Conclusion

In this thesis, we studied the computational complexity of three geometric optimization problems on guarding and partitioning of orthogonal polygons and, moreover, gave exact and approximation algorithms for these problems. We first studied two variants of the well-known art gallery problem (i.e., the MCSC and MLSC problems) in which sliding cameras are used to guard the gallery. We then considered a partitioning problem on orthogonal polygons in which the objective is to minimize the stabbing number of the partition (i.e., the MSN problem). Finally, we studied a discrete version of the terrain guarding problem on orthogonal terrains (i.e., the DTG problem). In the following, we provide a summary of our results and conclude the thesis by discussion on open problems and future work on these problems.

11.1 Minimum Sliding Cameras Problem

Summary. In Part I, we studied two variants of the orthogonal art gallery problem; namely, the MCSC and MLSC problems. These problems in which sliding cameras are used to guard the gallery were introduced by Katz and Morgenstern [27]. In this thesis, we gave an $O(n^{2.3727})$-time exact algorithm for the MLSC problem on any orthogonal polygon P with n vertices; our algorithm works even on orthogonal polygons with holes. In Chapter 3, we showed that the MCSC problem is NP-hard on orthogonal polygons with holes. Finally, we gave an exact $O(n)$-time dynamic programming algorithm for the MCSC problem on monotone polygons, where n is the number of vertices of the polygon (see Chapter 5), improving the 2-approximation algorithm of Katz and Morgenstern [27]. We also showed that our algorithm can be used to solve the MCSC problem on any orthogonal polygon for which the dual graph induced by the vertical decomposition of the polygon is a path.

Open Problems. Our work provides a number of open problems on the orthogonal art gallery problem using sliding cameras. The main question is what is the complexity of the MCSC problem on simple orthogonal polygons? Note that the problem is NP-hard if the polygon is allowed to have holes (see Chapter 3). Although our hardness does not apply to simple orthogonal polygons, we conjecture that the problem remains NP-hard even on simple orthogonal polygons.

Conjecture 11.1.1. *The MCSC problem is* NP-*hard on simple orthogonal polygons.*

Moreover, designing approximation algorithms or showing hardness of approximation is another direction for future work on the MCSC problem.

11.2 Minimum Stabbing Number Problem

Summary. In Part II, we studied the MSN problem; a partitioning problem on orthogonal polygons in which the objective is to compute a conforming partition of an orthogonal polygon P such that the stabbing number of the resulting partition is minimum over that of all such partitions of P. In Chapter 7, we showed that the MSN problem is NP-hard on orthogonal polygons with holes, providing the first complexity result for this problem. In Chapter 8, we gave a polynomial-time 2-approximation algorithm for the MSN problem on any orthogonal polygon and showed that the MSN problem can be solved on a histogram H in $O(n \log n)$-time, where n is the number of the vertices of H; our algorithm used a tree data structure in such a way that the stabbing number of a partition of H corresponds to some properties of the tree.

Open Problems. We showed in Chapter 7 that the MSN problem is NP-hard on orthogonal polygons with holes, however, the complexity of the problem on simple orthogonal polygons remains open as the main question on the MSN problem. We note that our reduction from a variant of the 3SAT problem does not apply to simple orthogonal polygons since holes are the most essential component of our gadgets. However, we believe that the MSN problem remains NP-hard even on simple orthogonal polygons and a reduction from a variant of the 3SAT problem requires a substantial modification to our gadgets; in particular, the variable gadget.

Conjecture 11.2.1. *The MSN problem is* NP*-hard on simple orthogonal polygons.*

In Chapter 8, we gave a polynomial-time 2-approximation algorithm for the MSN problem, however, we have no lower bound on the best approximation factor possible. Therefore, designing better approximation algorithms or showing a hardness of approximation is another direction for future work.

11.3 Directed Terrain Guarding Problem

Summary. In Part III, we studied a discrete variant of the terrain guarding problem on orthogonal terrains under directed visibility (i.e., the DTG problem). In our directed visibility model, the visibility is directed at adjacent vertices meaning that the visibility may not be mutual between two adjacent vertices. In Chapter 10, we considered several instances of the DTG problem, based on the type of the visibility constraints between adjacent vertices, and showed that they can be solved in linear time. Our algorithms are based on reducing the DTG problem to two subproblems and then solving each subproblem using a linear-time greedy algorithm.

Open Problems. Our results on the DTG problem provide the first exact algorithms for non-trivial instances of the terrain guarding problem. To achieve these results, we imposed the directed visibility constraint that made our techniques applicable only under that constraint. Therefore, the main open problem is to resolve the complexity of the DTG problem without directed visibility constraint (i.e., the

OTG problem). We note that the MTG problem (i.e., the terrain guarding problem on arbitrary terrains) is proved to be NP-hard by King and Krohn [30], but their reduction does not apply to orthogonal terrains.

In our directed visibility model, we assumed that a reflex vertex v cannot see its adjacent convex vertices. Our reduction to two subproblems does not apply to the case in which a reflex vertex can see its adjacent convex vertices. Therefore, resolving the complexity of the DTG problem under the assumption that a reflex vertex can see its adjacent convex vertices is another direction for future work.

Bibliography

[1] Mohammad A. Abam, Boris Aronov, Mark de Berg, and Amirali Khosravi. Approximation algorithms for computing partitions with minimum stabbing number of rectilinear and simple polygons. In *Proc. ACM SoCG*, pages 407–416, 2011. 4, 62, 78

[2] David Avis and Godfried T. Toussaint. An optimal algorithm for determining the visibility of a polygon from an edge. *IEEE Trans. Computers*, 30(12):910–914, 1981. 13

[3] Palrice Belleville. Computing two-covers of simple polygons. Master's thesis, McGill University, 2014. 15

[4] Boaz Ben-Moshe, Matthew J. Katz, and Joseph S. B. Mitchell. A constant-factor approximation algorithm for optimal terrain guarding. In *SODA*, pages 515–524, 2005. 93, 96

[5] Iliana Bjorling-Sachs and Diane L. Souvaine. A tight bound for edge guards in monotone polygons DIMACS. Technical Report 92-52, 1992. 14

[6] Danny Z. Chen and Haitao Wang. Visibility and ray shooting queries in

polygonal domains. In *Algorithms and Data Structures - 13th International Symposium, WADS London, ON, Canada, August 12-14, 2013. Proceedings*, pages 244–255, 2013. 38

[7] Vasek Chvatal. A combinatorial theorem in plane geometry. *Journal of Combinatorial Theory, Series B*, 18:39–41, 1975. v, 12, 13

[8] Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein. *Introduction to Algorithms (3rd ed.)*. MIT Press, 2009. 86

[9] Mark de Berg, Otfried Cheong, Marc van Kreveld, and Mark Overmars. *Computational Geometry: Algorithms and Applications*. Springer-Verlag, Heidelberg, 3rd edition, 2008. 79, 82

[10] Mark de Berg, Amirali Khosravi, Sander Verdonschot, and Vincent van der Weele. On rectilinear partitions with minimum stabbing number. In *Proc. WADS*, volume 6844 of *LNCS*, pages 302–313. Springer, 2011. 62

[11] Mark de Berg and Mark van Kreveld. Rectilinear decompositions with low stabbing number. *Inf. Proc. Let.*, 52(4):215–221, 1994. 62

[12] Alon Efrat, Cesim Erten, and Stephen Kobourov. Fixed-location circular arc drawing of planar graphs. *J. Graph Alg. & Applications*, 11(1):165–193, 2007. 65

[13] Stephan Eidenbenz, Christoph Stamm, and Peter Widmayer. Inapproximability results for guarding polygons and terrains. *Algorithmica*, 31(1):79–113, 2001. 14

[14] Khaled M. Elbassioni, Erik Krohn, Domagoj Matijevic, Julian Mestre, and Domagoj Severdija. Improved approximations for guarding 1.5-dimensional terrains. *Algorithmica*, 60(2):451–463, 2011. 94

[15] Sandor Fekete, Marco Lübbecke, and Henk Meijer. Minimizing the stabbing number of matchings, trees, and triangulations. *Disc. Comp. Geom.*, 40:595–621, 2008. 63

[16] Sándor P. Fekete and Joseph S. B. Mitchell. Terrain decomposition and layered manufacturing. *International Journal of Computational Geometry and Applications*, 11(6):647–668, 2001. 17

[17] Steve Fisk. A short proof of chvátal's watchman theorem. *J. Comb. Theory, Ser. B*, 24(3):374, 1978. 12

[18] Stephan Friedrichs, Michael Hemmer, and Christiane Schmidt. A PTAS for the continuous 1.5D terrain guarding problem. In *CCCG*, 2014. 6, 94

[19] Subir K. Ghosh. Approximation algorithms for art gallery problems in polygons. *Disc. App. Math.*, 158(6):718–722, 2010. 14

[20] Matt Gibson, Gaurav Kanade, Erik Krohn, and Kasturi R. Varadarajan. An approximation scheme for terrain guarding. In *APPROX-RANDOM*, pages 140–148, 2009. 6, 94

[21] Refael Hassin and Nimrod Megiddo. Approximation algorithms for hitting objects with straight lines. *Discrete Applied Mathematics*, 30(1):29–42, 1991. 19, 20

[22] John Hershberger and Subhash Suri. A pedestrian approach to ray shooting: shoot a ray, take a walk. *J. Alg.*, 18(3):403–431, 1995. 62

[23] Frank Hoffmann. On the rectilinear art gallery problem. In *17th International Colloquium on Automata, Languages and Programming, (ICALP)*, pages 717–728, 1990. 13

[24] Frank Hoffmann, Michael Kaufmann, and Klaus Kriegel. The art gallery theorem for polygons with holes. In *32nd Annual Symposium on Foundations of Computer Science, San Juan, Puerto Rico, 1-4 October 1991*, pages 39–48, 1991. 12

[25] Jeff Kahn, Maria M. Klawe, and Daniel J. Kleitman. Traditional galleries require fewer watchmen. *SIAM Journal on Algebraic Discrete Methods*, 4(2):194–206, 1983. v, 12, 13

[26] Richard M. Karp. Reducibility among combinatorial problems. In *Complexity of Computer Computations*, pages 85–103, 1972. 32

[27] Matthew J. Katz and Gila Morgenstern. Guarding orthogonal art galleries with sliding cameras. *International Journal of Computational Geometry and Applications*, 21(2):241–250, 2011. 2, 3, 16, 32, 40, 77, 114

[28] Matthew J. Katz and Gabriel S. Roisman. On guarding the vertices of rectilinear domains. *Comput. Geom.*, 39(3):219–228, 2008. 94

[29] James King. Fast vertex guarding for polygons with and without holes. *Comput. Geom.*, 46(3):219–231, 2013. 14, 15

[30] James King and Eric Krohn. Terrain guarding is NP-hard. *SIAM J. Comput.*, 40(5):1316–1339, 2011. 6, 94, 117

[31] Donald Knuth and Arvind Raghunathan. The problem of compatible representatives. *SIAM J. Disc. Math.*, 5(3):422–427, 1992. 66, 67

[32] Denes König. Gráfok és mátrixok. *Matematikai és Fizikai Lapok*, 38:116–119, 1931. 32, 38

[33] Erik Krohn and Bengt J. Nilsson. Approximate guarding of monotone and rectilinear polygons. *Algorithmica*, 66(3):564–594, 2013. 14, 15

[34] D. T. Lee and Arthur K. Lin. Computational complexity of art gallery problems. *IEEE Transactions on Information Theory*, 32(2):276–282, 1986. 2, 14

[35] Andrzej Lingas, Agnieszka Wasylewicz, and Pawel Zylinski. Linear-time 3-approximation algorithm for the r-star covering problem. In *Worshop on Algorithms and Computation*, pages 157–168, 2008. 15

[36] Rajeev Motwani, Arvind Raghunathan, and Huzur Saran. Covering orthogonal polygons with star polygons: the perfect graph approach. In *Symposium on Computational Geometry (SoCG)*, pages 211–223, 1988. 15, 16

[37] Marcin Mucha and Piotr Sankowski. Maximum matchings via gaussian elimination. In *45th Symposium on Foundations of Computer Science (FOCS 2004), 17-19 October 2004, Rome, Italy, Proceedings*, pages 248–255, 2004. 39

[38] Ernst P. Mücke, Isaac Saias, and Binhai Zhu. Fast randomized point location without preprocessing in two- and three-dimensional delaunay triangu-

lations. In *Proceedings of the Twelfth Annual Symposium on Computational Geometry, Philadelphia, PA, USA, May 24-26, 1996*, pages 274–283, 1996. 15

[39] Joseph O'Rourke. The complexity of computing minimum convex covers for polygons. In *20th Allerton Conf. Commun. Control Comput.*, pages 75–84, 1982. 12

[40] Joseph O'Rourke. Galleries need fewer mobile guards: a variation to Chvátal's theorem. *Geometriae Dedicata*, 14:273–283, 1983. 14

[41] Joseph O'Rourke. *Art Gallery Theorems and Algorithms*. The International Series of Monographs on Computer Science. Oxford University Press, New York, NY, 1987. 2, 11, 13, 14

[42] Abraham Punnen. *K*-sum linear programming. *J. Oper. Res. Soc.*, 43(4):359–363, 1992. 64

[43] Dietmar Schuchardt and Hans-Dietrich Hecker. Two NP-hard art-gallery problems for ortho-polygons. *Mathematical Logic Quarterly*, 41(2):261–267, 1995. 2, 14

[44] Saeed Seddighin. Guarding polygons with sliding cameras. Master's thesis, Sharif University of Technology, 2014. 17

[45] Thomas Shermer. Polygon guarding II: Efficient reduction of triangulation fragments. Technical Report 3D-16, NYIT Computer Graphics, New York Institute of Technology, 1985. 13

[46] Jonathan Richard Shewchuk. Stabbing delaunay tetrahedralizations. *Discrete & Computational Geometry*, 32(3):339–343, 2004. 63

[47] Csaba D. Tóth. Axis-aligned subdivisions with low stabbing numbers. *SIAM J. Discrete Math.*, 22(3):1187–1204, 2008. 63

[48] Jorge Urrutia. Art gallery and illumination problems. In *Handbook of Computational Geometry*, pages 973–1027. North-Holland, 2000. 11

[49] Virginia Vassilevska Williams. Multiplying matrices faster than coppersmith-winograd. In *STOC*, pages 887–898, 2012. 39

[50] David P. Williamson and David B. Shmoys. *The Design of Approximation Algorithms*. Cambridge University Press, 2011. 1, 86

[51] Chris Worman and J. Mark Keil. Polygon decomposition and the orthogonal art gallery problem. *International Journal of Computational Geometry and Applications*, 17(2):105–138, 2007. 15

[52] Pawel Zylinski. Orthogonal art galleries with holes: A coloring proof of Aggarwal's theorem. *Electr. J. Comb.*, 13(1), 2006. 13