Extremal solutions to some art gallery and terminal-pairability problems

Tamás Róbert Mezei

Supervisor: Prof. Ervin Győri

Department of Mathematics and its Applications
Central European University

In partial fulfillment of the requirements for the degree of
Doctor of Philosophy

Budapest, September 2017

I would like to dedicate this thesis to my family, without whose support this would never have been written.

Declaration

I hereby declare that except where specific reference is made to the work of others, the contents of this dissertation are original and have not been submitted in whole or in part for consideration for any other degree or qualification in this, or any other university. This dissertation is my own work and contains nothing which is the outcome of work done in collaboration with others, except as specified in the text.

Tamás Róbert Mezei
Budapest, September 2017

Acknowledgments

I would like to thank Ervin Győri and Gábor Mészáros for suggesting the topics discussed in this thesis. I would also like to thank them for a fruitful collaboration. I am ever so grateful to Ervin for all he has taught me (and the great stories he told me!) during the many hours of supervision over the years.

I am thankful to the examiners for their helpful comments.

Abstract

The thesis consists of two parts. In both parts, the problems studied are of significant interest, but are either *NP*-hard or unknown to be polynomially decidable. Realistically, this forces us to relax the objective of optimality or restrict the problem. As projected by the title, the chosen tool of this thesis is an *extremal type approach*. The lesson drawn by the theorems proved in the thesis is that surprisingly small compromise is necessary on the efficacy of the solutions to make the approach work. The problems studied have several connections to other subjects (e.g., geometric algorithms, graph immersions, multi-commodity flow problem) and practical applications (e.g., VLSI design, image processing, routing traffic in networks). Therefore, even slightly improving constants in existing results is beneficial.

The first part of the thesis is concerned with orthogonal art galleries. A sharp extremal bound is proved on partitioning orthogonal polygons into at most 8-vertex polygons using established techniques in the field of art gallery problems. This fills in the gap between already known results for partitioning into at most 6- and 10-vertex orthogonal polygons.

Next, these techniques are further developed to prove a new type of extremal art gallery result. The novelty provided by this approach is that it establishes a connection between mobile and stationary guards. This theorem has strong computational consequences, in fact, it provides the basis for an $\frac{8}{3}$-approximation algorithm for guarding orthogonal polygons with rectangular vision.

In the second part, the graph theoretical concept of terminal-pairability is studied in complete and complete grid graphs. Once again, the extremal approach is conductive to discovering efficient methods to solve the problem.

In the case of a complete base graph, the new demonstrated lower bound on the maximum degree of realizable demand graphs is 4 times higher than previous best results. The techniques developed are then used to solve the classical extremal edge number problem for the terminal-pairability problem in complete base graphs.

The complete grid base graph lies on the other end of the spectrum in terms density amongst path-pairable graphs. It is shown that complete grid graphs are relatively efficient in routing edge-disjoint paths. In fact, as a corollary, the minimum maximum degree a path-pairable graph may have is lowered to $O(\log n)$ (prior studies show a lower bound of $\Omega(\log n/\log\log n)$).

Table of contents

List of figures

List of tables

Preface

By merely reading the title of this work, the reader might wonder (and justifiably so) how the two main problems discussed in this thesis relate to each other. Well, I believe one of the main connection between them is my and Ervin's taste in mathematics. Let me explain.

Both the art gallery and the terminal-pairability problems encumber a vast family of natural questions. This is a direct consequence of the intuitiveness of these problems: they are very abstract models of challenges that appear in the real world, therefore they lend themselves to innumerable variations. Unfortunately, the generality of these problems — their computational complexity is either *NP*-hard or unknown to be polynomial — prevents us from finding an optimal solution (a lazy excuse, I know).

However, this is no reason to give up. The logical next step (at least to us) is to relax the goal of seeking an optimal solution to finding a bound, that which a solution achieving is guaranteed to exist. Hence, the purpose of this thesis is to find such bounds that are either sharp (the orthogonal art gallery theorems in Part I), or up to a small constant sharp (the terminal-pairability theorems in Part II). This *extremal* approach is a main theme of this thesis.

A pleasant phenomenon accompanying this approach is that we are also able to find efficient algorithms that construct the above described solutions. Moreover, our theorems guarantee that these solutions are constant approximations of the optimal solution, and thus are even relevant *in practice*.

I am hoping this preface provides a satisfying explanation of the apparent dichotomy present in the title. Now, I invite you, dear reader, to join me in my 3-year journey into discrete geometry, graph theory, algorithms, complexity, and combinatorics in general.

Part I

Orthogonal art galleries

Chapter 1

Introduction to orthogonal art galleries

1.1 Origins and summary of the new results

The original art gallery problem was stated by Victor Klee in 1973 [Hon76]. He posed the following question: given a simple polygon of n vertices, how many stationary guards are required to cover the interior of the polygon? To clarify, a point in the gallery is visible to the guard if the line segment spanned by the point and the guard lies in the closed gallery (line of sight vision).

The problem was solved by Vašek Chvátal in 1975:

Theorem 1.1 (Chvátal [Chv75]). *$\lfloor \frac{n}{3} \rfloor$ guards are sufficient and sometimes necessary to cover a domain bounded by a simple closed polygon.*

It is easy to see that at least $\lfloor \frac{n}{3} \rfloor$ guards are required, even if only the vertices of the polygon must be covered:

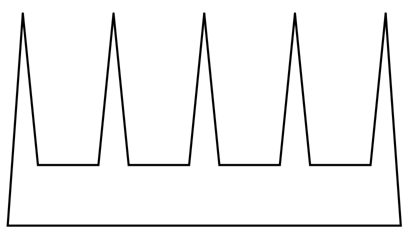


Fig. 1.1 The tips of the tooth are only visible from pairwise disjoint regions; therefore, a guard has to be placed in each tooth

The original proof by Chvátal used an inductive partitioning argument with 3 main cases and a couple of small subcases. Not much later, in 1978, Steve Fisk found such a beautiful proof of this theorem, that it is said to be "from the book" (see [AZ14]).

Proof of Theorem 1.1 [Fis78]. First, we prove the well-known fact that simple closed polygons can be triangulated, i.e., we can select $n-3$ pairs of vertices of the polygon, such that the line segments spanned by the pairs are in the polygon and these segments may only intersect in their endpoints. The proof is by induction. For $n = 3$, the statement is trivial. By sweeping the plane with a line whose slope is different from the slope of every side of the polygon, we can find a convex vertex v_2. Let v_1 and v_3 be its two neighbors.

- If the line segment $\overline{v_1v_3}$ is intersects the polygon in two points, we found a diagonal. Proceed by induction on the polygon obtained by deleting v_2 and adding $\overline{v_1v_3}$ as a new side.
- Otherwise, $L_0 = \{v_2\}$ and $L_1 = \overline{v_1v_3}$. For $t \in (0,1)$, let

$$L_t = \{(1-t)\cdot v_2 + t\cdot x \ : \ x \in L_1\},$$

  and take the minimum t for which L_t intersects the polygon in more than 2 points. One of these points must be a vertex y of the polygon, which is not contained in $\overline{v_1v_2} \cup \overline{v_2v_3}$. Clearly, $\overline{v_2y}$ is a diagonal of the polygon, which cuts it into two pieces, say, of n_1 and n_2 vertices. As $n_1 + n_2 = n + 2$, and $n_1, n_2 \geq 3$, we may proceed by induction to obtain $1 + (n_1 - 3) + (n_2 - 3) = n - 3$ diagonals.

To any triangulation of the (interior) of the polygon there is a corresponding planar graph G, whose outer face has exactly n points, but every other face of G is a triangle. Thus, the dual of this graph without the node corresponding to the outer face is a 3-regular tree. If G has 3 vertices, it is trivially 3-colorable. If G has more than 3 vertices, remove a degree 2 vertex (and its edges) of a face which is a leaf in the dual of G. By induction, the obtained graph is 3-colorable, and we can easily extend the 3-coloring to the removed degree 2-vertex.

The smallest color class A has size at most $\lfloor \frac{n}{3} \rfloor$. Clearly, the vertices in A cover the interior of the polygon, as any triangle face has a vertex in A. □

Observe, that the proof produces an interesting partition of the domain bounded by the polygon: triangles sharing the same vertex from A form a fan (imagine a handheld one without gaps), which can trivially be covered by one guard.

With the original problem of Chvátal solved, interest turned to different variations of the art gallery problem. One such version is when instead of a general polygon, the gallery is assumed to be bounded by an orthogonal polygon. In 1980, Kahn, Klawe, and Kleitman proved that

Theorem 1.2 (Kahn, Klawe, and Kleitman [KKK83]). $\lfloor \frac{n}{4} \rfloor$ *guards are sufficient and sometimes necessary to cover a domain bounded by an orthogonal polygon (even on a Riemann surface whose singularities lie outside the polygon).*

The sharp example is the orthogonal comb:

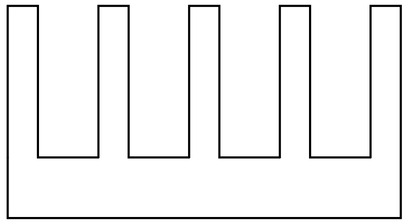


Fig. 1.2 An orthogonal comb; a guard has to be placed for each tooth

In fact, the trio proved the following deep geometric lemma.

Lemma 1.3 ([KKK83]). *Any closed region bounded by a finite number of straight lines, each parallel to one of two orthogonal axes, has a convex quadrilateralization (even on a Riemann surface whose singularities lie outside the closed region).*

To prove Theorem 1.2 using this lemma, they follow Fisk's argument. We need to assume that the gallery is bounded by an orthogonal polygon (i.e., holes are prohibited), so that that the dual graph of its quadrilateralization (without the outer face) is a tree. Add the two diagonals to each quadrilateral face. Notice, that this graph is 4-colorable, as the degree 2 (3 with the diagonals) vertices of a quadrilateral face which is a leaf in the dual can always be properly colored. The smallest color class covers the gallery.

As we noted earlier, the original proof of Lemma 1.3 uses deep geometrical insight, and its proof is about 10 pages long. However, Lubiw [Lub85] gives a sophisticated and much shorter proof by induction. Moreover, her proof is more general, as it includes certain polygons with holes, and it even leads to an efficient algorithm.

Even though Lemma 1.3 applies to even orthogonal polygons with holes, we need simply connectedness to construct the 4-coloring whose existence proves Theorem 1.2.

In the first half of the 1980's, Győri and O'Rourke independently gave a simple and short proof of Theorem 1.2.

Theorem 1.4 (Győri [Győ86] and O'Rourke [ORo87, Thm. 2.5])**.** *Every orthogonal polygon of n vertices can be partitioned into $\lfloor \frac{n}{4} \rfloor$ orthogonal polygons of at most 6 vertices.*

Theorem 1.4 is in some aspects a deeper result than that of Kahn, Klawe, and Kleitman, as any simple orthogonal polygon of 6 vertices can be covered by a stationary guard.

Each proof so far shines light on an interesting phenomenon, which we will refer to as the "metatheorem":

Metatheorem of art galleries. *Each (orthogonal) art gallery theorem has an underlying partition theorem (into simple parts).*

Although both Theorem 1.2 and Theorem 1.4 only apply to simply connected art galleries, Hoffman showed that the same bound holds for any closed region bounded by axis parallel line segments.

Theorem 1.5 (Hoffmann [Hof90])**.** *Any orthogonal polygon with holes of a total of n vertices can be partitioned into $\lfloor \frac{n}{4} \rfloor$ rectangular stars of at most 16 vertices.*

Hoffmann's theorem also verifies the metatheorem. Soon after this result, Hoffmann and Kaufmann [HK91] provided an efficient algorithm to construct such a partition.

In Chapter 2, we present further evidence that the metatheorem holds, namely we prove the following partition theorem:

Theorem 1.6 (Győri and Mezei [GM16])**.** *Any simple orthogonal polygon of n vertices can be partitioned into at most $\lfloor \frac{3n+4}{16} \rfloor$ orthogonal polygons of at most 8 vertices.*

A mobile guard is one who can patrol a line segment in the gallery, and it covers a point x of the gallery if there is a point y on its patrol such that the line segment $[x, y]$ is contained in the gallery. The upper bound of the mobile guard art gallery theorem for orthogonal polygons follows immediately from Theorem 1.6, as an orthogonal polygon of at most 8 vertices can be covered by a mobile guard.

Theorem 1.7 (Aggarwal [Agg84], [also in ORo87, Thm. 3.3])**.** $\lfloor \frac{3n+4}{16} \rfloor$ *mobile guards are sufficient for covering an n-vertex simple orthogonal polygon.*

The lower bound for the previous two theorems is given by stringing together a series of swastikas (Fig. 1.3). Observe, that a mobile guard may cover some points of the end of an arm of a swastika for at most one arm. Therefore a mobile guard has to be put in each arm. For $n \not\equiv 0 \pmod{16}$, a spiral has to attached to one of the arms.

	Point guards	Mobile guards
Simple polygons	$\left\lfloor \frac{n}{3} \right\rfloor$	$\left\lfloor \frac{n}{4} \right\rfloor$
Simple orthogonal polygons	$\left\lfloor \frac{n}{4} \right\rfloor$	$\left\lfloor \frac{3n+4}{16} \right\rfloor$

Table 1.1 The extremal number of guards required to cover an n-vertex gallery

Theorem 1.6 is a stronger result than Theorem 1.7 and it is interesting on its own. It fits into the series of results in [Győ86; HK91; ORo87, Thm. 2.5; GHKS96] showing that orthogonal art gallery theorems are based on theorems on partitions into smaller ("one guardable") pieces.

Moreover, Theorem 1.6 directly implies the following corollary which strengthens the previous theorem and answers two questions raised by O'Rourke [ORo87, Section 3.4].

Corollary 1.8 (Győri and Mezei [GM16]). $\lfloor \frac{3n+4}{16} \rfloor$ *mobile guards are sufficient for covering an n-vertex simple orthogonal polygon such that the patrols of two guards do not pass through one another and visibility is only required at the endpoints of the patrols.*

The results on guarding simple polygons and orthogonal polygons are summarized in Table 1.1. The proof of the sharp bound on mobile guards in simple polygons due to O'Rourke [ORo87, Thm. 3.1] also confirms the metatheorem. A combinatorial proof that does not use complex geometric reasoning has already existed for three of the four bounds listed in Table 1.1. The until recently missing fourth such proof is that of Theorem 1.6.

Joseph O'Rourke pointed out in his 1987 book titled "Art gallery theorems and algorithms" [ORo87] that there is a mysterious 4 : 3 ratio between the extremal number of point and mobile guards for art galleries given by both simple polygons and simple orthogonal polygons, as can be seen on Table 1.1.

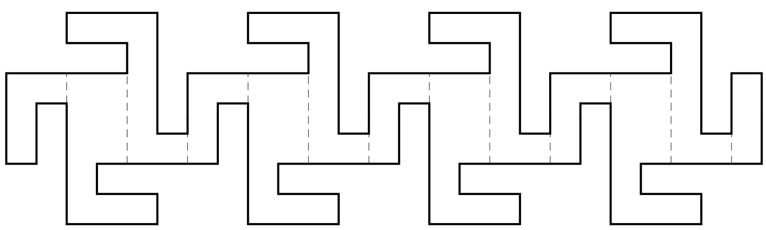


Fig. 1.3 The dashed lines show a minimum cardinality partition into at most 8-vertex pieces

1.2 Outline of Part I

After precisely defining the subjects of our study in Chapter 2, we introduce the concept of R-trees, which is a well-known tool in the literature. The proof of Theorem 1.6 follows.

In Chapter 3 we show that this ratio between the efficacy of point and mobile guards is not only an extremal phenomenon in simple orthogonal polygons appearing for a fixed number of vertices. The magical ratio appears in an upper bound for the ratio of the minimum number of stationary guards covering the gallery and the minimum size of a special, restricted mobile guard cover. The results of Chapter 2 and 3 have been discovered in collaboration with my supervisor, Ervin Győri.

In the last chapter of Part I of this thesis (Chapter 4) we discuss algorithmic versions of our proofs and the computational complexity of orthogonal art gallery problems in general.

1.3 Definitions and preliminaries

Our universe for the study of art galleries is the plane $\mathbb{R}^2$. A **polygon** is defined by a cyclically ordered list of pairwise distinct vertices in the plane. It is drawn by joining each successive pair of vertices on the list by line segments, that only intersect in vertices of the polygon. The last requirement ensures that the closed domain bounded by the polygon is simply connected (to emphasize this, such polygons are often referred to as simple polygons in the literature). An **orthogonal polygon** is a polygon such that its line segments are alternatingly parallel to one of the axes of $\mathbb{R}^2$. Consequently, its angles are $\frac{1}{2}\pi$ (convex) or $\frac{3}{2}\pi$ (reflex).

A **rectilinear domain** is a closed region of the plane ($\mathbb{R}^2$) whose boundary is an orthogonal polygon, i.e., a closed polygon without self-intersection, so that each segment is parallel to one of the two axes. A **rectilinear domain with holes** is a rectilinear domain with pairwise disjoint simple rectilinear domain holes. Its boundary is referred to as an **orthogonal polygon with holes**.

The definitions imply that number of vertices of an orthogonal polygon (even with holes) is even. We denote the number of vertices of the polygon by $n(P)$, and define $n(D) = n(P)$, where D is the domain bounded by P. Conversely, we write $P = \partial D$. We want to emphasize that in our problems not just the walls, but also the interior of the gallery must be covered. In the proofs of the theorems, therefore, we are working on rectilinear domains, not orthogonal polygons, even though one defines the other uniquely, and vice versa.

Whenever results about objects that are allowed to have holes are mentioned, it is explicitly stated.

To avoid confusion, we state that throughout this part, **vertices** and **sides** refer to subsets of an orthogonal polygon or a rectilinear domain; whereas any **graph** will be defined on a set of **nodes**, of which some pairs are joined by some **edges**. Given a graph G, the edge set $E(G)$ is a subset of the 2-element subsets of the vertices $V(G)$.

Unless otherwise noted, we adhere to the same terminology in the subject of art galleries as O'Rourke [ORo87]. However, for technical reasons, sometimes we need to assume extra conditions over what is traditionally assumed. In Lemma 1.9, we prove that we may, without restricting the problem, require the assumptions typeset in *italics* in the following definitions.

Two points x, y in a domain D have **line of sight vision**, **unrestricted vision**, or simply just **vision** of each other if the line segment induced by x and y is contained in D.

A **point guard** in an art gallery D is a point $y \in D$. It has vision of a point $x \in D$ if the line segment $\overline{xy}$ is a subset of D. The term "stationary guard" refers to the same meaning, and is used mostly in contrast with "mobile guards".

A **mobile guard** is a line segment $L \subset D$. A point $x \in D$ is seen by the guard if there is a point $y \in L$ which has vision of y. Intuitively, a mobile guard is a point guard patrolling the line segment L.

The points **covered by a guard** is just another name for the set of points of D that are seen by the guard. A **system of guards** is a set of guards in D which cover D, i.e., for any point $x \in D$, there is a guard in the system covering x.

Two points x, y in a rectilinear domain D have ***r*-vision** of each other (alternatively, x is r-visible from y) if there exists an axis-aligned *non-degenerate* rectangle in D which contains both x and y. This vision is natural to use in orthogonal art galleries instead of the more powerful line of sight vision. For example, r-vision is invariant on the transformation depicted on Fig. 1.4.

A **point *r*-guard** is a point $y \in D$, *such that the two maximal axis-parallel line segments in D containing y do not intersect vertices of D*. A set of point guards ***r*-cover** D if any point $x \in D$ is r-visible from a member of the set. Such a set is called a **point *r*-guard system**.

A **vertical mobile *r*-guard** is a vertical line segment in D, *such that the maximal line segment in D containing it does not intersect vertices of D*. **Horizontal** mobile guards are defined analogously. A **mobile *r*-guard** is either a vertical or a horizontal mobile r-guard. A mobile

r-guard ***r*-covers** any point $x \in D$ for which there exists a point y on its line segment such that x is r-visible from y.

Lemma 1.9. *Any rectilinear domain D can be transformed into another rectilinear domain D' so that the point guard r-cover, and the vertical/horizontal mobile guard r-cover problems in D, without the restrictions typeset in italics, are equivalent to the respective problems, as per our definitions (i.e., with the restrictions), in D'.*

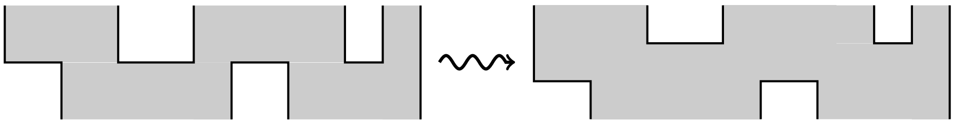


Fig. 1.4 After this transformation, those mobile guards whose maximal containing line segment do not intersect vertices of the rectilinear domain, are just as powerful as mobile guards that are not restricted in such a way.

Proof. Let ε be the minimal distance between any two horizontal line segments of ∂D. The transformation depicted in Fig. 1.4 in D takes a maximal horizontal line segment L in D which is touched from both above and below by the exterior of D, and maps D to

$$D' = D \bigcup \left(L + \overline{(0, -\varepsilon/4)(0, \varepsilon/4)} \right),$$

where addition is taken in the Minkowski sense. There is a trivial correspondence between the point and mobile guards of D and D' such that taking this correspondence guard-wise transforms a guarding system of D (guards without the restrictions) into a guarding system of D' (guards with the restrictions), and vice versa.

After performing this operation at every vertical and horizontal occurrence, we get a rectilinear domain D'', in which any vertical or horizontal line segment is contained in a non-degenerate rectangle in D''. Therefore, degenerate vision between any two points implies non-degenerate vision between the pair. Furthermore, the line segment of any mobile guard can be translated slightly along its normal (at least in one direction) while staying inside D'', and this clearly does not change the set of points r-covered by the guard. Similarly, we can perturb the position of a point guard without changing the set of points of D'' it r-covers. $\square$

Chapter 2

Partitioning orthogonal polygons

2.1 Introduction

For the sake of completeness, we mention that any orthogonal polygon of n vertices can be partitioned into at most $\lfloor \frac{n}{2} \rfloor - 1$ rectangles, and this bound is sharp. In this case, the interesting question is the minimum number of covering rectangles, see Chapter 4.

Theorem 1.6 fills in a gap between two already established (sharp) results: in [Győ86] it is proved that orthogonal polygons can be partitioned into at most $\lfloor \frac{n}{4} \rfloor$ orthogonal polygons of at most 6 vertices, and in [GHKS96] it is proved that any orthogonal polygon in general position (an orthogonal polygon without 2-cuts) can be partitioned into $\lfloor \frac{n}{6} \rfloor$ orthogonal polygons of at most 10 vertices. However, we do not know of a sharp theorem about partitioning orthogonal polygons into orthogonal polygons of at most 12 vertices.

Furthermore, for $k \geq 4$, not much is known about partitioning orthogonal polygons with holes into orthogonal polygons of at most $2k$ vertices. Per the "metatheorem," the first step in this direction would be proving that an orthogonal polygon of n vertices with h holes can be partitioned into $\lfloor \frac{3n+4h+4}{16} \rfloor$ orthogonal polygons of at most 8 vertices. This would generalize the corresponding art gallery result in [GHKS96, Thm. 5.].

The proof of Theorem 1.6 is similar to the proofs of Theorem 1.4 in that it finds a suitable cut and then uses induction on the parts created by the cut. However, a cut along a line segment connecting two reflex vertices is no longer automatically good. We also rely heavily on a tree structure of the orthogonal polygon (Section 2.3). However, while O'Rourke's proof of Theorem 1.4 only uses straight cuts, in our case this is not sufficient: Fig. 2.1 shows an orthogonal polygon of 14 vertices which cannot be cut into 2 orthogonal polygons of at most 8 vertices using cuts along straight lines. Therefore, we must consider L-shaped cuts too.

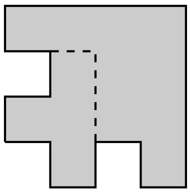


Fig. 2.1 An L-shaped cut creating a partition into 8-vertex orthogonal polygons

2.2 Definitions and preliminaries

Let D, D_1, D_2 be rectilinear domains of n, n_1, n_2 vertices, respectively. If $D = D_1 \cup D_2$, $\mathrm{int}(D_1) \cap \mathrm{int}(D_2) = \emptyset$, $0 < n_1, n_2$ and $n_1 + n_2 \leq n + 2$ are satisfied, we say that D_1, D_2 form an **admissible partition** of D, which we denote by $D = D_1 \biguplus D_2$. Also, we call $L = D_1 \cap D_2 = \partial D_1 \cap \partial D_2$ a **cut** in this case. We may describe this relationship concisely by $L(D_1, D_2)$. If, say, we have a number of cuts L_1, L_2, etc., then we usually write $L_i(D_1^i, D_2^i)$. Generally, if a rectilinear domain is denoted by D_x^y, then y refers to a cut and $x \in \{1, 2\}$ is the label of the piece in the partition created by said cut. Furthermore, if

$$\left\lfloor \frac{3n_1 + 4}{16} \right\rfloor + \left\lfloor \frac{3n_2 + 4}{16} \right\rfloor \leq \left\lfloor \frac{3n + 4}{16} \right\rfloor . \tag{2.1}$$

is also satisfied, we say that D_1, D_2 form an **induction-good partition** of D, and we call L a **good cut**.

Lemma 2.1. *An admissible partition $D = D_1 \biguplus D_2$ is also and induction-good partition if*

(a) $n_1 + n_2 = n + 2$ *and* $n_1 \equiv 2, 8,$ *or* $14 \pmod{16}$*,*

or

(b) $n_1 + n_2 = n$ *and* $n_1 \equiv 0, 2, 6, 8, 12,$ *or* $14 \pmod{16}$*,*

or

(c) $n \not\equiv 14 \mod 16$ *and* ***either***

$n_1 + n_2 = n$ *and* $n_1 \equiv 10 \pmod{16}$, ***or***

$n_1 + n_2 = n + 2$ *and* $n_1 \equiv 12 \pmod{16}$.

Proof. Using the fact that the floor function satisfies the triangle inequality, the proof reduces to an easy case-by-case analysis, which we leave to the reader. □

Any cut L falls into one of the following 3 categories (see Fig. 2.2):

(a) 1-cuts: L is a line segment, and exactly one of its endpoints is a (reflex) vertex of D.

(b) 2-cuts: L is a line segment, and both of its endpoints are (reflex) vertices of D.

(c) L-cuts: L consists of two connected line segments, and both endpoints of L are (reflex) vertices of D.

Note that for 1-cuts and L-cuts the size of the parts satisfy $n_1 + n_2 = n + 2$, while for 2-cuts we have $n_1 + n_2 = n$.

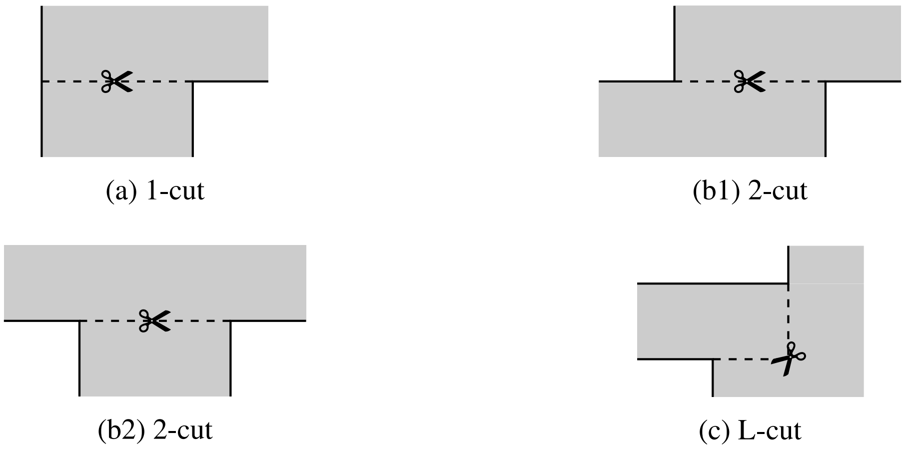


Fig. 2.2 Examples for all types of cuts. Light gray areas are subsets of int(D).

In the proof of Theorem 1.6 we are searching for an induction-good partition of D. As a good cut defines an induction-good partition, it is sufficient to find a good cut. We could hope that a good cut of a rectilinear piece of D is extendable to a good cut of D, but unfortunately a good cut of a rectilinear piece of D may only be an admissible cut with respect to D (if it is a cut of D at all). Lemma 2.1, however, allows us to look for cut-systems containing a good cut. Fortunately, it is sufficient to consider non-crossing, nested cut-systems of at most 3 cuts, defined as follows.

Definition 2.2 (Good cut-system)**.** The cuts $L_1(D_1^1, D_2^1)$, $L_2(D_1^2, D_2^2)$ and $L_3(D_1^3, D_2^3)$ (possibly $L_2 = L_3$) constitute a good cut-system if $D_1^1 \subset D_1^2 \subseteq D_1^3$, and the set

$$\left\{n(D_1^i) \mid i \in \{1,2,3\}\right\} \cup \left\{n(D_1^i) + 2 \mid i \in \{1,2,3\} \text{ and } L_i \text{ is a 2-cut}\right\}$$

contains three consecutive even elements modulo 16 (i.e., the union of their residue classes contains a subset of the form $\{a, a+2, a+4\} + 16\mathbb{Z}$). If this is the case we also define their kernel as $\ker\{L_1, L_2, L_3\} = (D_1^1 - L_1) \cup (D_2^3 - L_3)$, which will be used in Lemma 2.10.

Lemma 2.1(a) and 2.1(b) immediately yield that any good cut-system contains a good cut.

Remark 2.3. It is easy to see that if a set of cuts satisfies this definition, then they obviously satisfy it in the reverse order too (the order of the generated parts is also switched). Actually, these are exactly the two orders in which they do so. Thus, the kernel is well-defined, and when speaking about a good cut-system it is often enough to specify the set of participating cuts.

2.3 Tree structure

Any reflex vertex of a rectilinear domain D defines a (1- or 2-) cut along a horizontal line segment whose interior is contained in $\mathrm{int}(D)$ and whose endpoints are the reflex vertex and another point on the boundary of D. Next, we define a graph structure derived from D, which is a standard tool in the literature, for example it is called the R-graph of an orthogonal polygon in [GHKS96]. A similar structure is used by O'Rourke to prove Theorem 1.4, see [ORo87, p. 76].

Definition 2.4 (Horizontal (vertical) R-tree)**.** The horizontal (vertical) R-tree T of a rectilinear domain D (or the orthogonal polygon ∂D bounding it) is obtained as follows. First, partition D into a set of rectangles by cutting along all the horizontal (vertical) cuts of D. Let $V(T)$, the vertex set of T be the set of resulting (internally disjoint) rectangles. Two rectangles of T are connected by an edge in $E(T)$ iff their boundaries intersect.

The graph T is indeed a tree as its connectedness is trivial and since any cut creates two internally disjoint rectilinear domains, T is also cycle-free. We can think of T as a sort of dual of the planar graph determined by the union of ∂D and its horizontal cuts. The nodes of T represent rectangles of D and edges of T represent horizontal 1- and 2-cuts. For this reason, we may refer to nodes of T as rectangles. This nomenclature also helps in distinguishing between vertices of D (points) and nodes of T. Moreover, for an edge $e \in E(T)$, we may denote the cut represented by e by simply e, as the context should make it clear whether we are working in the graph T or in the plane.

Note that the vertical sides of rectangles are also edges of the orthogonal polygon bounding D.

Definition 2.5. Let T be the horizontal R-tree of D. Define $t : E(T) \to \mathbb{Z}$ as follows: given any edge $\{R_1, R_2\} \in E(T)$, let

$$t(\{R_1, R_2\}) = n(R_1 \cup R_2) - 8.$$

Observe that

$$t(e) = \begin{cases} 0, & \text{if } e \text{ represents a 2-cut;} \\ -2, & \text{if } e \text{ represents a 1-cut.} \end{cases}$$

The following claim is used throughout the chapter to count the number of vertices of a rectilinear piece of D.

Claim 2.6. *Let T be the horizontal R-tree of D. Then*

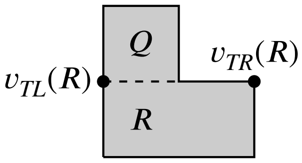
$$n(D) = 4|V(T)| + \sum_{e \in E(T)} t(e).$$

Proof. The proof is straightforward. □

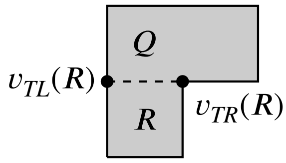
Remark 2.7. Equality in the previous claim holds even if some of the rectangles of T are cut into several rows (and the corresponding edges, for which the function t takes -4, are added to T).

2.4 Extending cuts and cut-systems

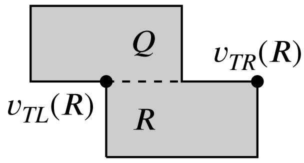
The following two technical lemmas considerably simplify our analysis in Section 2.5, where many cases distinguished by the relative positions of reflex vertices of D on the boundary of a rectangle need to be handled. For a rectangle R let us denote its top left, top right, bottom left, and bottom right vertices with $v_{TL}(R)$, $v_{TR}(R)$, $v_{BL}(R)$, and $v_{BR}(R)$, respectively.



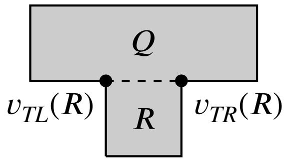
(a) $Q \subseteq R_{TL}$, and $e_{TL}(R)$ is a 1-cut



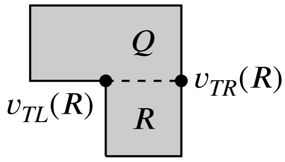
(b) $Q \subseteq R_{TL}$, $e_{TL}(R)$ is a 1-cut, and $R_{TR} = \emptyset$



(c) $Q \subseteq R_{TL}$, and $e_{TL}(R)$ is a 2-cut



(d) $R_{TL} = R_{TR} = \emptyset$, R is either a corridor or a pocket (see Section 2.5)



(e) $R_{TL} = \emptyset$, but $Q \subseteq R_{TR}$

Fig. 2.3 $R \cup Q$ in all essentially different relative positions of $R, Q \in V(T)$ (up to dilation and contraction of the segments of $R \cup Q$ such that its angles are preserved eventually), where $\{R, Q\} \in E(T)$ and $v_{TL}(R) \in Q$

Definition 2.8. Let $R, Q \in V(T)$ be arbitrary. We say that Q is **adjacent** to R at $v_{TL}(R)$, if $v_{TL}(R) \in Q$ and $v_{TL}(R)$ is not a vertex of the rectilinear domain $R \cup Q$, or $v_{TR}(R) \notin Q$. Such situations are depicted on Figures 2.3a, 2.3b, and 2.3c. However, in the case of Fig. 2.3d and 2.3e we have $v_{TR}(R) \in Q \nsubseteq R_{TL}$ $(= \emptyset)$.

If Q is adjacent to R at $v_{TL}(R)$, let $e_{TL}(R) = \{R, Q\}$; by cutting D along the dual of $e_{TL}(R)$, i.e., $R \cap Q$, we get two rectilinear domains, and we denote the part containing Q by R_{TL}. If there is no such Q, let $e_{TL}(R) = \emptyset$ and $R_{TL} = \emptyset$. These relations can be defined analogously for top right (R_{TR}, $e_{TR}(R)$), bottom left (R_{BL}, $e_{BL}(R)$), and bottom right (R_{BR}, $e_{BR}(R)$).

Lemma 2.9. *Let R be an arbitrary rectangle such that $R_{BL} \neq \emptyset$. Let U be the remaining portion of the rectilinear domain, i.e., $D = R_{BL} \uplus U$ is a partition into rectilinear domains. Take an admissible partition $U = U_1 \uplus U_2$ such that $v_{BL}(R) \in U_1$. We can extend this to an admissible partition of D where the two parts are $U_1 \cup R_{BL}$ and U_2.*

Proof. Let $Q_1 = R \cap U_1$ and let $Q_2 \in V(T)$ be the rectangle which is a subset of R_{BL} and adjacent to R.

Observe that U_1 and R_{BL} only intersect on R's bottom side, therefore their intersection is a line segment L and so $U_1 \cup R_{BL}$ is a rectilinear domain. Trivially, $D = (U_1 \cup R_{BL}) \uplus U_2$ is partition into rectilinear domains, so only admissibility remains to be checked.

Let the horizontal R-tree of U_1 and R_{BL} be T_{U_1} and $T_{R_{BL}}$, respectively. The horizontal R-tree of $U_1 \cup R_{BL}$ is $T_{U_1} + T_{R_{BL}} + \{Q_1, Q_2\}$, except if $t(\{Q_1, Q_2\}) = -4$. Either way, by referring to Remark 2.7 we can use Claim 2.6 to write that

$$\begin{aligned} n(U_1 \cup R_{BL}) + n(U_2) - n(D) &= n(U_1) + n(R_{BL}) + t(\{Q_1, Q_2\}) + n(U_2) - n(D) = \\ &= \Big(n(U_1) + n(U_2) - n(U)\Big) + \Big(n(U) + n(R_{BL}) - n(D)\Big) + t(\{Q_1, Q_2\}) = \\ &= \Big(n(U_1) + n(U_2) - n(U)\Big) - t(\{R, Q_2\}) + t(\{Q_1, Q_2\}). \end{aligned} \tag{2.2}$$

Now it is enough to prove that $t(\{Q_1, Q_2\}) \le t(\{R, Q_2\})$. If $t(\{R, Q_2\}) = 0$ this is trivial. The remaining case is when $t(\{R, Q_2\}) = -2$. This means that $v_{BL}(R)$ is not a vertex of $R \cup Q_2$, therefore it is not a vertex of $Q_1 \cup Q_2$ either, implying that $n(Q_1 \cup Q_2) < 8$. □

Lemma 2.10. *Let $R \in V(T)$ be such that $R_{BL} \neq \emptyset$. Let U be the other half of the rectilinear domain, i.e., $D = R_{BL} \uplus U$. If U has a good cut-system $\mathscr{L}$ such that $v_{BL}(R) \in \ker \mathscr{L}$, then D also has a good cut-system.*

Proof. Let us enumerate the elements of $\mathscr{L}$ as L_i where $i \in I$. Take $L_i(U_1^i, U_2^i)$ such that $v_{BL}(R) \in U_1^i$. Using Lemma 2.9 extend L_i to a cut $L_i'(D_1^i, D_2^i)$ of D such that $U_2^i = D_2^i$. Equation (2.2) and the statement following it implies that

$$n(D_1^i) + n(D_2^i) = n(D) + 2 \implies n(U_1^i) + n(U_2^i) = n(U) + 2.$$

In other words, if L_i is a 2-cut then so is L_i'. Therefore

$$\begin{aligned} &\left\{n(U_2^i) \mid i \in I\right\} \cup \left\{n(U_2^i) + 2 \mid i \in I \text{ and } L_i \text{ is a 2-cut}\right\} \subseteq \\ &\qquad \subseteq \left\{n(D_2^i) \mid i \in I\right\} \cup \left\{n(D_2^i) + 2 \mid i \in I \text{ and } L_i' \text{ is a 2-cut}\right\}, \end{aligned}$$

and by referring to Remark 2.3, we get that $\{L_i' \mid i \in I\}$ is a good cut-system of D. □

2.5 Proof of Theorem 1.6

Let us recall the theorem to be proved.

Theorem 1.6 (Győri and Mezei [GM16])**.** *Any simple orthogonal polygon of n vertices can be partitioned into at most $\lfloor \frac{3n+4}{16} \rfloor$ orthogonal polygons of at most 8 vertices.*

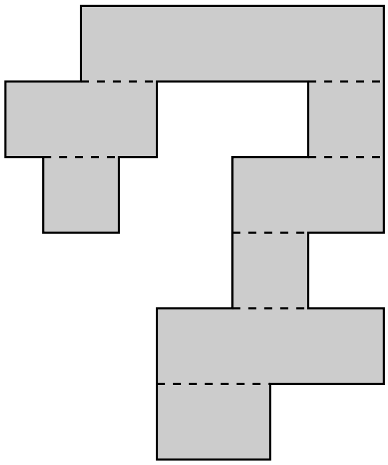
We will prove Theorem 1.6 by induction on the number of vertices. For $n \le 8$ the theorem is trivial.

For $n > 8$, let D be the rectilinear domain bounded by the orthogonal polygon wall of the gallery. We want to partition D into smaller rectilinear domains. It is enough to prove that D has a good cut. The rest of this proof is an extensive case study. Let T be the horizontal R-tree of D. We need two more definitions.

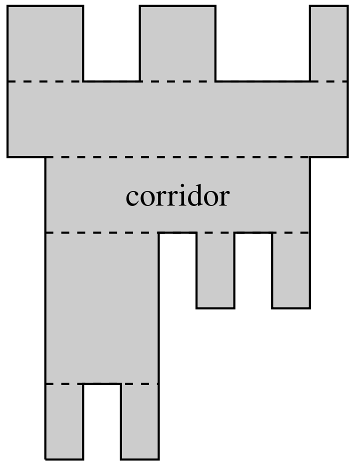
- A **pocket** in T is a degree-1 rectangle R, whose only incident edge in T is a 2-cut of D, and this cut covers the entire top or bottom side of R.
- A **corridor** in T is a rectangle R of degree ≥ 2 in T, which has an incident edge in T which is a 2-cut of D, and this cut covers the entire top or bottom side of R.

We distinguish 4 cases.

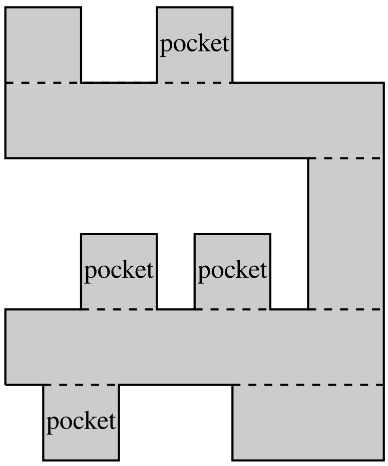
Case 1. T is a path, Fig. 2.4(a);
Case 2. T has a corridor, Fig. 2.4(b);
Case 3. T does not have a corridor, but it has a pocket, Fig. 2.4(c);
Case 4. None of the previous cases apply, Fig. 2.4(d).



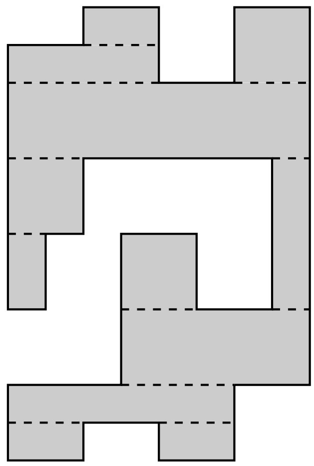
(a) T is a path.



(b) T has a corridor.



(c) T does not have a corridor, but it has a pocket.



(d) T does not have a corridor or a pocket, and it is not a path.

Fig. 2.4 The 4 cases of the proof.

Case 1 $\quad T$ is a path

Claim 2.11. *If an edge incident to a degree-2 vertex R of T is a 2-cut of D, then the incident edges of R form a good cut-system.*

Proof. Let the two incident edges of R be e_1 and e_2. Let their generated partitions be $e_1(D_1^1, D_2^1)$ and $e_2(D_1^2, D_2^2)$, such that $R \subseteq D_2^1 \cap D_1^2$. Then $D_1^2 = D_1^1 \cup R$, so

$$n(D_1^2) = n(D_1^1) + n(R) + t(e_1) = n(D_1^1) + 4.$$

Definition 2.2 is satisfied by $\{e_1, e_2\}$, as $\{n(D_1^1), n(D_1^2)\} \cup \{n(D_1^1) + 2\}$ is a set of three consecutive even elements. □

Claim 2.12. *If there are two rectangles R_1 and R_2 which are adjacent degree-2 vertices of T, then the union of the set of incident edges of R_1 and R_2 form a good cut-system.*

Proof. Let the two components of $T - R_1 - R_2$ be T_1 and T_2, so that $e_1, e_2, f \in E(T)$ joins T_1 and R_1, R_1 and R_2, R_2 and T_2, respectively. Obviously, $\cup V(T_1) \subset (\cup V(T_1)) \cup R_1 \subset (\cup V(T_1)) \cup R_1 \cup R_2$. If one of $\{e_1, e_2, f\}$ is a 2-cut, we are done by the previous claim. Otherwise

$$n((\cup V(T_1)) \cup R_1) = n(\cup V(T_1)) + n(R_1) + t(e_1) = n(\cup V(T_1)) + 2,$$
$$n((\cup V(T_1)) \cup R_1 \cup R_2) = n(\cup V(T_1)) + n(R_1) + n(R_2) + t(e_1) + t(f) = n(\cup V(T_1)) + 4,$$

and so $\{n(\cup V(T_1)), n((\cup V(T_1)) \cup R_1), n((\cup V(T_1)) \cup R_1 \cup R_2)\}$ are three consecutive even elements. This concludes the proof that $\{e_1, e_2, f\}$ is a good cut-system of D. □

Suppose T is a path. If T is a path of length ≤ 3, such that each edge of it is a 1-cut, then $n(D) \leq 8$. Also, if T is path of length 2 and its only edge represents a 2-cut, then $n(D) = 8$. Otherwise, either Claim 2.11, or Claim 2.12 can be applied to provide a good cut-system of D.

Case 2 T has a corridor

Let $e = \{R', R\} \in E(T)$ be a horizontal 2-cut such that R' is a wider rectangle than R, and $\deg(R) \geq 2$. Let the generated partition be $e(D_1^e, D_2^e)$ such that $R' \subseteq D_1^e$. We can handle all possible cases as follows.

1. If $n(D_1^e) \not\equiv 4, 10 \bmod 16$ or $n(D_2^e) \not\equiv 4, 10 \bmod 16$, then e is a good cut by Lemma 2.1(b).
2. If $\deg(R) = 2$, we find a good cut using Claim 2.11.
3. If $R_{BL} = \emptyset$, then $L(D_1^L, D_2^L)$ such that $R' \subseteq D_1^L$ in Fig. 2.5 is a good cut, since $n(D_1^L) = n(D_1^e) + 4 - 0 \equiv 8, 14 \bmod 16$.

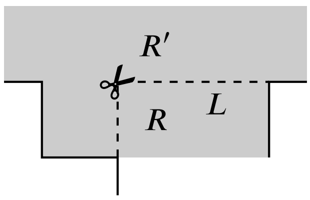


Fig. 2.5 L is a good cut

4. If $R_{BL} \neq \emptyset$ and $\deg(R) \geq 3$, then let us consider the following five cuts of D (Fig. 2.6): $L_1(R_{BL}, R \cup D_1^e)$, $L_2(R_{BL} \cup Q_1, Q_2 \cup Q_3 \cup D_1^e)$, $L_3(R_{BL} \cup Q_1 \cup Q_2, Q_3 \cup D_1^e)$, $L_4(Q_3, R_{BL} \cup Q_1 \cup Q_2 \cup D_1^e)$, and $L_5(Q_3 \cup Q_2, R_{BL} \cup Q_1 \cup D_1^e)$.

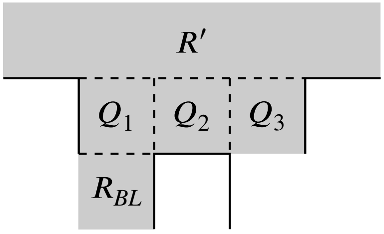


Fig. 2.6 $\deg(R) \geq 3$ and $R_{BL} \neq \emptyset$

The first pieces of these partitions have the following number of vertices (respectively).

(a) $n(R_{BL})$
(b) $n(R_{BL} \cup Q_1) = n(R_{BL}) + n(Q_1) + (t(e_{BL}(R)) - 2) = n(R_{BL}) + t(e_{BL}(R)) + 2$
(c) $n(R_{BL} \cup Q_1 \cup Q_2) = n(R_{BL}) + n(Q_1 \cup Q_2) + t(e_{BL}(R)) = n(R_{BL}) + t(e_{BL}(R)) + 4$
(d) $n(Q_3)$
(e) $n(Q_3 \cup Q_2) = n(Q_3) + n(Q_2) - 2 = n(Q_3) + 2$

- If $t(e_{BL}(R)) = 0$, then $\{L_1, L_2, L_3\}$ is a good cut-system, so one of them is a good cut.
- If $t(e_{BL}(R)) = -2$, and none of the 5 cuts above are good cuts, then using Lemma 2.1(b) on L_2 and L_3 gives $n(R_{BL}) \equiv 4, 10 \bmod 16$. The same argument can be used on L_4 and L_5 to conclude that $n(Q_3) \equiv 4, 10 \bmod 16$. However, previously we derived that

$$n(D_2^e) \equiv 4, 10 \bmod 16,$$
$$n(R_{BL} \cup Q_1 \cup Q_2 \cup Q_3) = n(R_{BL} \cup Q_1 \cup Q_2) + n(Q_3) - 2 =$$
$$= n(R_{BL}) + n(Q_3) \equiv 4, 10 \bmod 16.$$

  This is only possible if $n(R_{BL}) \equiv n(Q_3) \equiv 10 \bmod 16$. Let $e_{BL}(R) = \{R, S\}$.

  - If $\deg(S) = 2$, then let $E(T) \ni e' \neq e_{BL}(R)$ be the other edge of S. Let the partition generated by it be $e'(D_1^{e'}, D_2^{e'})$ such that $R' \subseteq D_1^{e'}$. We have

$$n(R_{BL}) = n(D_2^{e'}) + n(S) + t(e')$$
$$n(D_2^{e'}) = n(R_{BL}) - 4 - t(e') \equiv 6 - t(e') \bmod 16$$

Either e' is a 1-cut, in which case $n(D_2^{e'}) \equiv 8 \mod 16$, or e' is a 2-cut, giving $n(D_2^{e'}) \equiv 6 \mod 16$. In any case, Lemma 2.1 says that e' is a good cut.

– If $\deg(S) = 3$, then we can partition D as in Fig. 2.7. Since $n(Q_5 \cup Q_6) = 4 + n(Q_6) - 2$, by Lemma 2.1(a) the only case when neither

$$L_6(Q_5 \cup Q_6, Q_4 \cup R \cup D_1^e), \text{ nor}$$
$$L_7(Q_6, Q_4 \cup Q_5 \cup R \cup D_1^e)$$

is a good cut of D is when $n(Q_6) \equiv 4, 10 \mod 16$. Also,

$$10 \equiv n(Q_4 \cup Q_5 \cup Q_6) = n(Q_4) + n(Q_5) + n(Q_6) - 2 - 2 \equiv$$
$$\equiv n(Q_4) + n(Q_6) \mod 16.$$

∗ If $n(Q_6) \equiv 10 \mod 16$, then $n(Q_4) \equiv 0 \mod 16$, hence

$$n(Q_4 \cup Q_5 \cup Q_{11} \cup Q_{12}) =$$
$$= n(Q_4 \cup Q_5) + 4 - 4 = n(Q_4) + n(Q_5) - 2 \equiv 2 \mod 16,$$

showing that $L_8(Q_4 \cup Q_5 \cup Q_{11} \cup Q_{12}, Q_6 \cup Q_{13} \cup Q_2 \cup Q_3 \cup D_1^e)$ is a good cut.

∗ If $n(Q_6) \equiv 4 \mod 16$,

$$n(Q_6 \cup Q_{13} \cup Q_2 \cup Q_3) = n(Q_6) + n(Q_{13} \cup Q_2) + n(Q_3) - 2 - 2 \equiv$$
$$\equiv n(Q_6) + 10 \equiv 14 \quad \mod 16,$$

therefore $L_9(Q_6 \cup Q_{13} \cup Q_2 \cup Q_3, Q_4 \cup Q_5 \cup Q_{11} \cup Q_{12} \cup D_1^e)$ is a good cut.

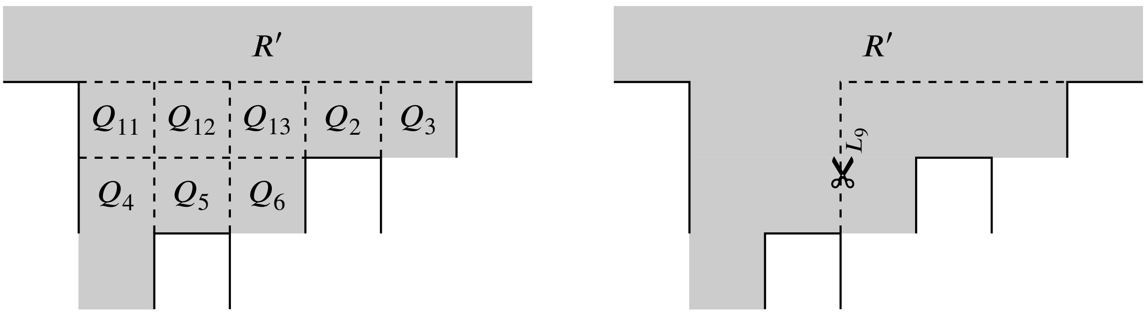


Fig. 2.7 $\deg(Q) \geq 3$ and $Q_{BL} \neq \varnothing$

In each of the above subcases we found a good cut.

Case 3 There are no corridors in T, but there is a pocket

Let S be a (horizontal) pocket. Also, let R be the neighbor of S in T. If $\deg(R) = 2$, then Claim 2.11 provides a good cut-system of D. However, if $\deg(R) \geq 3$, we have two cases.

Case 3.1 If R is adjacent to at least two pockets

Let U be the union of R and its adjacent pockets, and let T_U be its **vertical** R-tree. It contains at least 4 reflex vertices, therefore $|V(T_U)| \geq 3$.

- If $V(T_U) = 3$, then $|E(T_U)| = 2$. Thus $t(e) = 0$ for any $e \in E(T_U)$, and Claim 2.11 gives a good cut-system $\mathscr{L}$ of U such that all 4 vertices of R are contained in $\ker \mathscr{L}$.
- If $V(T_U) \geq 4$, then Claim 2.12 gives a good cut-system $\mathscr{L}$ of U such that all 4 vertices of R are contained in $\ker \mathscr{L}$.

Since there are no corridors in D, we have

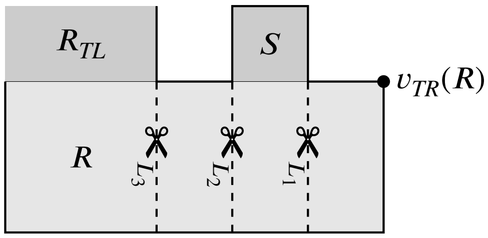
$$D = \Big(\big((U \cup R_{BL}) \cup R_{TL} \big) \cup R_{BR} \Big) \cup R_{TR}.$$

By applying Lemma 2.10 repeatedly, the good cut-system $\mathscr{L}$ can be extended to a good cut-system of D.

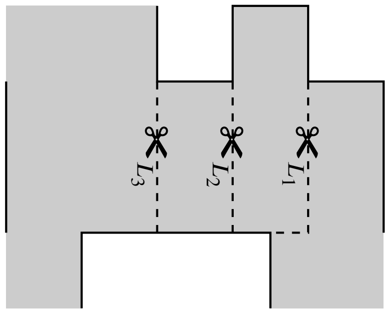
Case 3.2 If S is the only pocket adjacent to R

We may assume without loss of generality that S intersects the top side of R. Again, define U as the union of R and its adjacent pockets.

- If $R_{TL} \neq \emptyset$, let $V = U \uplus R_{TL}$. The cut-system $\{L_1, L_2, L_3\}$ in Fig. 2.8a is a good cut-system of V, and all 4 vertices of R are contained in $\ker\{L_1, L_2, L_3\}$. By applying Lemma 2.10 repeatedly, we get a good cut-system of D, for example, see Fig. 2.8b.



(a) $\{L_1, L_2, L_3\}$ is a good cut-system of $V = R \cup R_{TL} \cup S$



(b) The extended cuts

Fig. 2.8 R has one pocket

- If $R_{TR} \neq \varnothing$, the case can be solved analogously to the previous case.
- Otherwise $R_{BL} \neq \varnothing$ and $R_{BR} \neq \varnothing$. Let $L_1(U_1^1, U_2^1)$ and $L_2(U_1^2, U_2^2)$ be the vertical cuts (from right to left) defined by the two reflex vertices of U, such that $v_{BR}(R) \in U_1^1 \subset U_1^2$. Let $V = R_{BL} \uplus U$. As before, L_1 and L_2 can be extended to cuts of V, say $L_1'(U_1^1, V_2^1)$, $L_2'(U_1^2, V_2^2)$. We claim that together with $e_{BL}(R)(U, V_2^3)$, they form a good cut-system $\mathscr{L}$ of V. This is obvious, as $\{n(U_1^1), n(U_1^2), n(U)\} = \{4, 6, 8\}$. Since $v_{BR}(R) \in \ker \mathscr{L}$, D also has a good cut-system by Lemma 2.10.

Case 4 T is not a path and it does not contain either corridors or pockets

By the assumptions of this case, any two adjacent rectangles are adjacent at one of their vertices, so the maximum degree in T is 3 or 4. We distinguish between several subcases.

Case 4.1. There exists a rectangle of degree ≥ 3 such that its top or bottom side is entirely contained in one of its neighboring rectangles;

Case 4.2. Every rectangle of degree ≥ 3 is such that its top and bottom sides are not entirely contained in any of their neighboring rectangles;

Case 4.2.1. There exist at least two rectangles of degree ≥ 3;

Case 4.2.2. There is exactly one rectangle of degree ≥ 3.

Case 4.1 There exists a rectangle of degree $\geq$ 3 such that its top or bottom side is entirely contained in one of its neighboring rectangles

Let R be a rectangle and R' its neighbor, such that the top or bottom side of R is a subset of $\partial R'$. Moreover, choose R such that if we partition D by cutting $e = \{R, R'\}$, the part containing R is minimal (in the set theoretic sense).

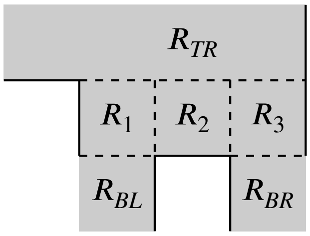


Fig. 2.9 The top side of $R = R_1 \cup R_2 \cup R_3$ is contained entirely by a neighboring rectangle.

Without loss of generality, the top side of R is contained entirely by a neighboring rectangle R', and $R_{TL} = \varnothing$. This is pictured in Fig. 2.9, where $R = R_1 \cup R_2 \cup R_3$. We can cut off R_{BL}, $R_{BL} \cup R_1$, and $R_{BL} \cup R_1 \cup R_2$, whose number of vertices are respectively

1. $n(R_{BL})$,
2. $n(R_{BL} \cup R_1) = n(R_{BL}) + n(R_1) + (t(e_{BL}(R)) - 2) = n(R_{BL}) + t(e_{BL}(R)) + 2$,
3. $n(R_{BL} \cup R_1 \cup R_2) = n(R_{BL}) + n(R_1 \cup R_2) + t(e_{BL}(R)) = n(R_{BL}) + t(e_{BL}(R)) + 4$.

If $t(e_{BL}(R)) = 0$, then one of the 3 cuts is a good cut by Lemma 2.1(a).

Otherwise $t(e_{BL}(R)) = -2$, thus, one of the 3 cuts is a good cut, or $n(R_{BL}) \equiv 4, 10 \mod 16$. Let S be the rectangle for which $e_{BL}(R) = \{R, S\}$. Since $e_{BL}(R)$ is a 1-cut containing the top side of S, we cannot have $\deg(S) = 3$, as it contradicts the choice of R. We distinguish between two cases.

Case 4.1.1 $\deg(S) = 1$

Let $U = R' \cup R \cup R_{BL} \cup R_{BR}$, which is depicted on Fig. 2.10a. It is easy to see that $L_1(Q_1, U_2^1)$, $L_2(Q_1 \cup Q_2, U_2^2)$, and $L_3(Q_1 \cup Q_2 \cup Q_3, U_2^3)$ in Fig. 2.10b is a good cut-system of U.

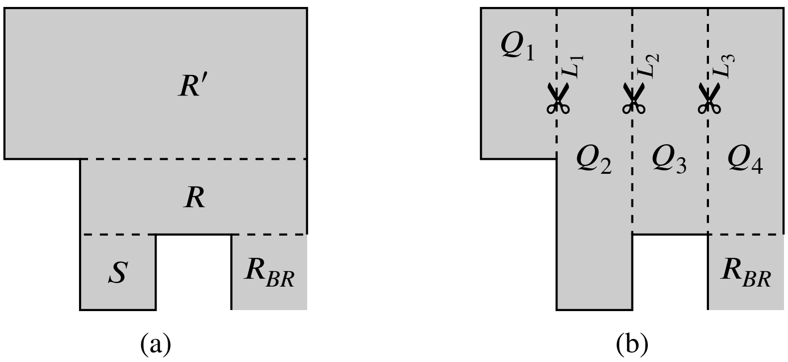


Fig. 2.10 The rectilinear domain U is shown in (a). The cuts L_1, L_2, L_3, shown in (b), form a good-cut system of U.

As all 4 vertices of S are contained in $\ker\{L_1, L_2, L_3\}$, we can extend this good cut-system to D by reattaching S_{TL}, S_{BL}, S_{TR} (if non-empty) via Lemma 2.10. Therefore, D has a good cut.

Case 4.1.2 deg(S) = 2

Let f be the edge of S which is different from $e_{BL}(R) = e_{TL}(S)$. Let the partition generated by it be $f(D_1^f, D_2^f)$, where $S \subseteq D_2^f$. We have $n(D_1^f) = n(R_{BL}) - n(S) - t(f)$.

- If $t(f) = -2$, then $n(D_1^f) \equiv 2, 8 \mod 16$, so f is a good cut by Lemma 2.1(a).
- If $t(f) = 0$, then $n(D_1^f) \equiv 0, 6 \mod 16$, so f is a good cut by Lemma 2.1(b).

Case 4.2 Every rectangle of degree $\geq$ 3 is such that its top and bottom sides are not entirely contained in any of their neighboring rectangles

Let R be a rectangle of degree ≥ 3 and $e = \{R, S\}$ be one of its edges. Let the partition generated by e be $e(D_1^e, D_2^e)$, where $R \subset D_1^e$ and $S \subseteq D_2^e$. If e is a 1-cut, then by the assumptions of this case $\deg(S) \leq 2$.

- If $\deg(S) = 1$ and $t(e) = -2$, then $n(D_2^e) + t(e) = 2$.
- If $\deg(S) = 1$ and $t(e) = 0$, then $n(D_2^e) + t(e) = 4$.
- If $\deg(S) = 2$ and one of the edges of S is a 0-cut, then D has a good cut by Claim 2.11.
- If $\deg(S) = 2$ and both edges of S, e and (say) f are 1-cuts: Let the partition generated by f be $D = D_1^f \uplus D_2^f$, such that $S \in D_1^f$. Then $n(D_2^e) = n(D_2^f) + n(S) + t(f) = n(D_2^f) + 2$. Either one of e and f is a good cut, or by Lemma 2.1(a) we have $n(D_2^f) \equiv 4, 10 \mod 16$. In other words, $n(D_2^e) + t(e) \equiv 4, 10 \mod 16$. Similarly, $n(D_1^f) = n(D_1^e) + 4 - 2 = n(D_1^e) + 2$, so $n(D_1^e) \equiv 4, 10 \mod 16$.
- If $\deg(S) \geq 3$, then $t(e) = 0$. Either e is a good cut, or by Lemma 2.1(b) we have $n(D_2^e) + t(e) \equiv 4, 10 \mod 16$. Lemma 2.1(b) also implies $n(D_1^e) \equiv 4, 10 \mod 16$.

From now on, we assume that none of the edges of the neighbors of a degree $\geq$ 3 rectangle represent a good cut, so in particular, we have

$$n(D_2^e) + t(e) \equiv 2, 4, \text{ or } 10 \mod 16.$$

In addition to the simple analysis we have just conducted, we deduce an easy claim to be used in the following subcases.

Claim 2.13. *Let $R \in V(T)$ be of degree ≥ 3 and suppose both $R_{BR} \neq \emptyset$ and $R_{TR} \neq \emptyset$. Then D has two admissible cuts L_1 and L_2 such that they form a good cut-system or*

(i) *one of the parts generated by L_1 has size* $\left(n(R_{BR}) + t(e_{BR}(R))\right) + \left(n(R_{TR}) + t(e_{TR}(R))\right) + 2,$

and

(ii) *one of the parts generated by L_2 has size* $\left(n(R_{BR}) + t(e_{BR}(R))\right) + \left(n(R_{TR}) + t(e_{TR}(R))\right) + 4.$

Proof. Let $U = R \cup R_{BL} \cup R_{BR}$. Let $L_1(U_1^1, U_2^1)$ and $L_2(U_1^2, U_2^2)$ be the vertical cuts of U defined by the two reflex vertices of U that are on the boundary of R, such that $v_{BR}(R) \in U_1^1 \subset U_1^2$. By Lemma 2.9, L_1 and L_2 can be extended to cuts of $V = R \cup R_{BL} \cup R_{BR} \cup R_{TR}$, say $L_1'(V_1^1, U_2^1)$, $L_2'(V_1^2, U_2^2)$. If one of L_1' or L_2' is a 2-cut, then similarly to Claim 2.11, one can verify they form a good cut-system of V, which we can extend to D. Otherwise

$$
\begin{aligned}
n(V_1^1) &= n(R \cap U_1^1) + n(R_{BR}) + n(R_{TR}) + (t(e_{BR}(R)) - 2) + t(e_{TR}(R)) = \\
&= \left(n(R_{BR}) + t(e_{BR}(R))\right) + \left(n(R_{TR}) + t(e_{TR}(R))\right) + 2, \\
n(V_1^2) &= n(R \cap U_1^2) + n(R_{BR}) + n(R_{TR}) + t(e_{BR}(R)) + t(e_{TR}(R)) = \\
&= \left(n(R_{BR}) + t(e_{BR}(R))\right) + \left(n(R_{TR}) + t(e_{TR}(R))\right) + 4.
\end{aligned}
$$

Lastly, we extend L_1' and L_2' to D by reattaching R_{TL} using Lemma 2.9. This step does not affect the parts V_1^1 and V_1^2, so we are done. □

Case 4.2.1 There exist at least two rectangles of degree ≥ 3

In the subgraph T' of T which is the union of all paths of T which connect two degree ≥ 3 rectangles, let R be a leaf and $e = \{R, S\}$ its edge in the subgraph. As defined in the beginning of Case 4.2, the set of incident edges of R (in T) is $\{e_i \mid 1 \leq i \leq \deg(R)\}$, and without loss of generality we may suppose that $e = e_{\deg(R)}$. The analysis also implies that for all $1 \leq i \leq \deg(R) - 1$, we have $n(D_2^{e_i}) + t(e_i) = 2, 4$.

By the assumptions of this case $\deg(S) \geq 2$, therefore $n(D_1^e) \equiv 4$ or $10 \mod 16$. If $\deg(S) \geq 3$, let $Q = S$. Otherwise $\deg(S) = 2$, and let Q be the second neighbor of R in T'. The degree of Q cannot be 1 by its choice. If $\deg(Q) = 2$, then we find a good cut using Claim 2.12. In any case, we may suppose from now on that $\deg(Q) \geq 3$.

Let $\{f_i \mid 1 \leq i \leq \deg(Q)\}$ be the set of incident edges of Q such that they generate the partitions $D = D_1^{f_i} \uplus D_2^{f_i}$ where $R \subset D_1^{f_i}$ and $Q \subset D_2^{f_1}$. We have

$$n(D_1^e) = n(R) + \sum_{i=1}^{\deg(R)-1} \left(n(D_2^{e_i}) + t(e_i) \right) \in 4 + \{2,4\} + \{2,4\} + \{0,2,4\} =$$
$$= \{8, 10, 12, 14, 16\},$$

so the only possibility is $n(D_1^e) = 10$.

- If $\deg(S) \geq 3$, e is a 2-cut (by the assumption of Case 4.2), so by Lemma 2.1(c), either e is a good cut or $n(D) \equiv 14 \bmod 16$. Since $Q = S$ and $e = f_1$, we have

$$n(D_1^{f_1}) + t(f_1) = n(D_1^e) + t(e) = 10,$$
$$n(D_2^{f_1}) = n(D) - n(D_1^{f_1}) - t(f_1) \equiv 14 - 10 \equiv 4 \bmod 16.$$

- If $\deg(S) = 2$, either e is a 1-cut or we find a good cut using Claim 2.11. Also, $f_1 = \{S, Q\}$ is a 1-cut too (otherwise apply Claim 2.11), so

$$n(D_1^{f_1}) + t(f_1) = n(D_1^e) + n(S) + t(e) + t(f_1) = 10.$$

  By Lemma 2.1(c), either f_1 is a good cut (as $n(D_1^{f_1}) = 12$) or $n(D) \equiv 14 \bmod 16$. Thus

$$n(D_2^{f_1}) = n(D) - n(D_1^{f_1}) - t(f_1) \equiv 14 - 12 + 2 \equiv 4 \bmod 16.$$

We have

$$n(D) = n(Q) + \left(n(D_1^{f_1}) + t(f_1) \right) + \sum_{i=2}^{\deg(Q)} \left(n(D_2^{f_i}) + t(f_i) \right) \in$$
$$\in 14 + \{2,4,10\} + \{2,4,10\} + \{0,2,4,10\} \quad \bmod 16.$$

The only way we can get 14 mod 16 on the right-hand side is when $\deg(Q) = 4$ and out of

$$n(Q_{BL}), n(Q_{BR}), n(Q_{TL}), n(Q_{TR}) \quad \bmod 16,$$

one is 2, another is 4, and two are 10 mod 16.

The last step in this case is to apply Claim 2.13 to Q. If it does not give a good cut-system, then it gives an admissible cut where one of the parts has size congruent to $2+10+2=14$ or $2+4+2=8$ modulo 16, therefore we find a good cut anyway.

Case 4.2.2 There is exactly one rectangle of degree ≥ 3

Let R be the rectangle of degree ≥ 3 in T, and let $\{e_i \mid 1 \leq i \leq \deg(R)\}$ be the edges of R, which generate the partitions $D = D_1^{e_i} \uplus D_2^{e_i}$ where $R \subset D_1^{e_i}$. Then $D_2^{e_i}$ is path for all i. If either Claim 2.11 or Claim 2.12 can be applied, D has a good cut. The remaining possibilities can be categorized into 3 types:

$$\begin{array}{llll}
\textbf{Type 1:} & t(e_i) = -2, & n(D_2^{e_i}) = 4, & \text{and} \quad n(D_2^{e_i}) + t(e_i) = 2; \\
\textbf{Type 2:} & t(e_i) = -2, & n(D_2^{e_i}) = 4+4-2 = 6, & \text{and} \quad n(D_2^{e_i}) + t(e_i) = 4; \\
\textbf{Type 3:} & t(e_i) = 0, & n(D_2^{e_i}) = 4, & \text{and} \quad n(D_2^{e_i}) + t(e_i) = 4.
\end{array}$$

Without loss of generality $R_{BR} \neq \varnothing$ and $R_{TR} \neq \varnothing$. We will now use Claim 2.13. If it gives a good cut-system, we are done. Otherwise

- If exactly one of $e_{BR}(R)$ and $e_{TR}(R)$ is of **type 1**, apply Claim 13(i): it gives an admissible cut which cuts off a rectilinear domain of size $2+4+2=8$, so D has a good cut.
- If both $e_{BR}(R)$ and $e_{TR}(R)$ are of **type 1**, apply Claim 13(ii): it gives an admissible cut which cuts off a rectilinear domain of size $2+2+4=8$, so D has a good cut.
- If none of $e_{BR}(R)$ and $e_{TR}(R)$ are of **type 1** and $n(D) \not\equiv 14 \pmod{16}$, apply Claim 13(i): it gives an admissible cut which cuts off a rectilinear domain of size $4+4+4=12$, which is a good cut by Lemma 2.1(c).

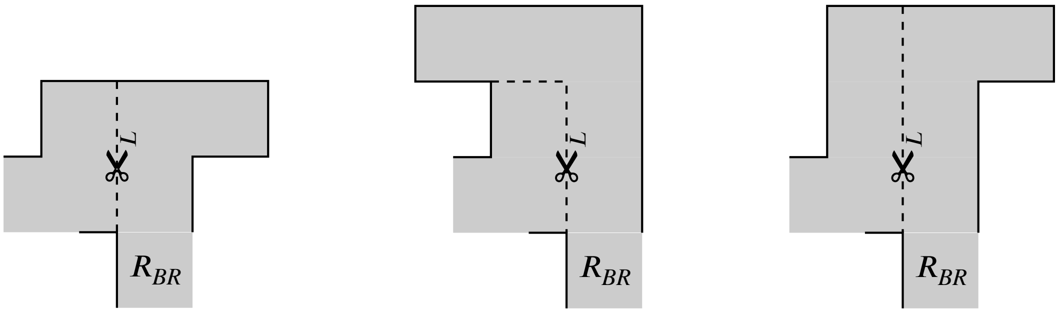
Now we only need to deal with the case where $n(D) = 14$ and neither $e_{BR}(R)$ nor $e_{TR}(R)$ is of **type 1**.

If R still has two edges of **type 1**, again Claim 13(i) gives a good cut of D. If R has at most one edge of **type 1**, we have

$$14 = n(D) = n(R) + \sum_{i=1}^{\deg(R)} \left(n(D_2^{e_i}) + t(e_i) \right) \in 4 + \{2,4\} + \{4\} + \{4\} + \{0,4\} = \{14,16,18,20\},$$

and the only way we can get 14 on the right hand is when $\deg(R) = 3$ and both $e_{BR}(R)$ and $e_{TR}(R)$ are of **type 2 or 3** while the third incident edge of R is of **type 1**. We may assume without loss of generality that the cut represented by $e_{TR}(R)$ is longer than the cut represented by $e_{BR}(R)$.

- If D is vertically convex, its vertical R-tree is a path, so it has a good cut as deduced in Case 1.
- If D is not vertically convex, but $e_{TR}(R)$ is a **type 2** edge of R, such that the only horizontal cut of R_{TR} is shorter than the cut represented by $e_{TR}(R)$, then $D' = D - R_{BR}$ is vertically convex, and has $(10 - 4)/2 = 3$ reflex vertices. By Claim 2.11 or Claim 2.12, D' has a good cut-system such that its kernel contains $v_{BR}(R)$, since its x-coordinate is maximal in D'. Lemma 2.10 states that D also has a good cut-system.
- Otherwise we find that the top right part of D looks like to one of the cases in Fig. 2.11. It is easy to see that in all three pictures L is an admissible cut which generates two rectilinear domains of 8 vertices.



(a) (b) (c)

Fig. 2.11 The last 3 cases of the proof. Since $e_{BR}(R)$ is a **type 2 or 3** edge, cutting the rectilinear domain at L creates two rectilinear domains of 8-vertices.

The proof of Theorem 1.6 is complete. To complement the formal proof, we now demonstrate the algorithm on Fig. 2.12.

First, we resolve a corridor via the L-cut ①, which creates two pieces of 20 and 34 ($\equiv 2 \bmod 16$) vertices. Because of this cut, a new corridor emerges in the 20-vertex piece, so we cut the rectilinear domain at ②, cutting off a piece of 8 vertices. The other piece of 14 vertices containing two pockets is further divided by ③ into two pieces of 6 and 8 vertices. Another pocket is dealt with by cut ④, which divides the rectilinear domain into 8- and 28-vertex pieces. To the larger piece, Case 4.2 applies, and we find cut ⑤, which produces 16- and 14-vertex pieces. The 16-vertex piece is cut into two 8-vertex pieces by ⑥. Lastly, Fig. 2.11c of Case 4.2.2 applies to the 14-vertex piece, so cut ⑦ divides it into two 8-vertex pieces.

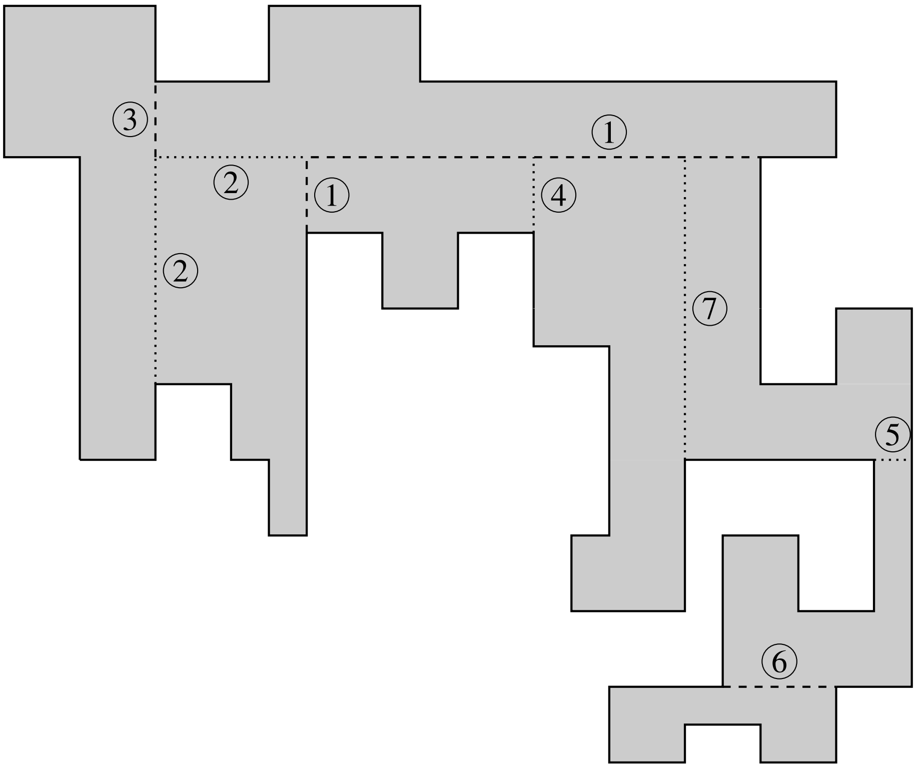


Fig. 2.12 The output of the algorithm on a rectilinear domain of 52 vertices.

We got lucky with cuts ② and ⑥ in the sense that they both satisfy the inequality in (2.1) strictly. Hence, only 8 pieces are needed to partition the rectilinear domain of Fig. 2.12 into rectilinear domains of $\leq$ 8-vertex pieces, instead of the extremal upper bound of $(3 \cdot 52+4)/16 = 10$.

Chapter 3

Mobile vs. point guards

3.1 Introduction

The main goal of this chapter is to explore the ratio between the numbers of mobile guards and points guards required to control an orthogonal polygon without holes. At first, this appears to be hopeless, as Fig. 3.1 shows a comb, which can be guarded by one mobile guard (whose patrol is shown by a dotted horizontal line). However, to cover the comb using point guards, one has to be placed for each tooth, so ten point guards are needed (marked by solid disks). Combs with arbitrarily high number of teeth clearly demonstrate that the minimum number of points guards required to control an orthogonal polygon cannot be bounded by the minimum size of a mobile guard system covering the comb.

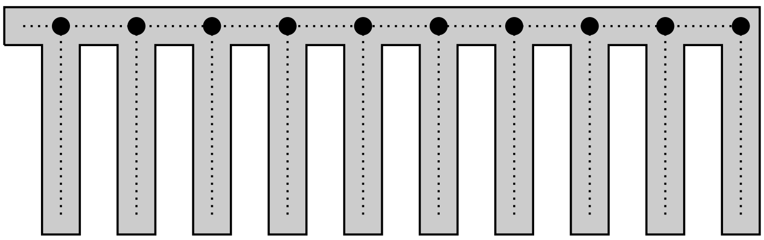


Fig. 3.1 A comb with 10 teeth

Katz and Morgenstern [KM11] defined and studied the notion of “horizontal sliding cameras”. This notion is identical to what we call horizontal mobile r-guard (a horizontal mobile guard with rectangular vision). The main result of this chapter, Theorem 3.1, shows that a constant factor times the sum of the minimum sizes of a horizontal and a vertical mobile r-guard system can be used to estimate the minimum size of a point r-guard system. It is surprising to have

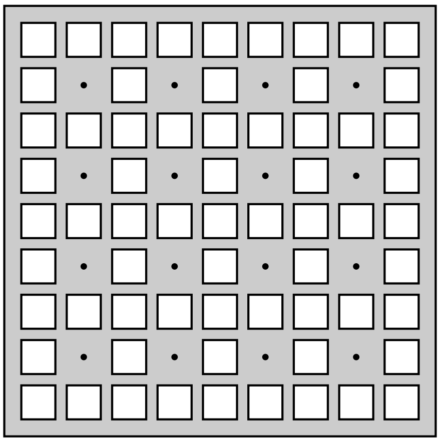


Fig. 3.2 A polygon with holes — unlimited ratio.

such a result after encountering the comb, but it is similarly unexpected that even this ratio cannot be bounded if the region may contain holes.

Take, for example, Fig. 3.2, which generally contains $3k^2 + 4k + 1$ square holes (in the figure $k = 4$). The regions covered by line of sight vision by the black dots are pairwise disjoint, because the distance between adjacent square holes is less than half of the length of a square hole's side. Therefore, no two of the black dots can be covered by one point guard, so at least k^2 point guards are necessary to control gallery. However, $2k + 2$ horizontal mobile guards can easily cover the polygon, and the same holds for vertical mobile guards.

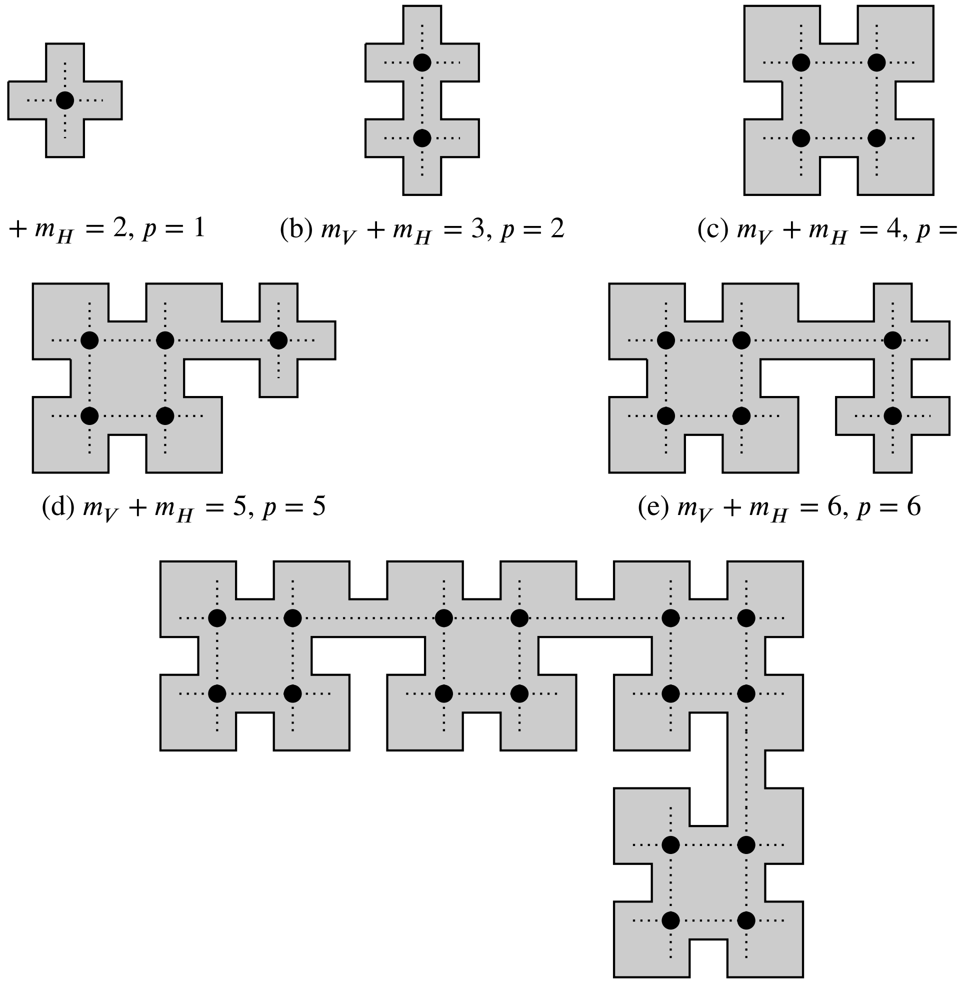
In the next chapter, we show that a minimum size horizontal mobile r-guard system can be found in linear time (Theorem 4.4). This improves the result in [KM11], where it is shown that this problem can be solved in polynomial time.

Theorem 3.1 (Győri and Mezei [GM17])**.** *Given a rectilinear domain D let m_V be the minimum size of a vertical mobile r-guard system of D, let m_H be defined analogously for horizontal mobile r-guard systems, and finally let p be the minimum size of a point r-guard system of D. Then*

$$\left\lfloor \frac{4(m_V + m_H - 1)}{3} \right\rfloor \geq p.$$

In case it is not confusing, the prefix "r-" is omitted from now on. Before moving onto the proof of Theorem 3.1, we discuss the aspects of its sharpness.

For $m_V + m_H \leq 6$, sharpness of the theorem is shown by the examples in Fig. 3.3. The polygon in Fig. 3.3f can be easily generalized to one satisfying $m_V + m_H = 3k + 1$ and $p = 4k$. For $m_V + m_H = 3k + 2$ and $m_V + m_H = 3k + 3$, we can attach 1 or 2 plus signs to the previously



(a) $m_V + m_H = 2$, $p = 1$ (b) $m_V + m_H = 3$, $p = 2$ (c) $m_V + m_H = 4$, $p = 4$

(d) $m_V + m_H = 5$, $p = 5$ (e) $m_V + m_H = 6$, $p = 6$

(f) $m_V + m_H = 13$, $p = 16$

Fig. 3.3 Vertical dotted lines: a minimum size vertical mobile guard system;
Horizontal dotted lines: a minimum size horizontal mobile guard system;
Solid disks: a minimum size point guard system.

constructed polygons, as shown in Fig. 3.3d and 3.3e. Thus Theorem 3.1 is sharp for any fixed value of $m_V + m_H$.

By stringing together a number of copies of the polygons in Fig. 3.3a and 3.3c in an L-shape (Fig. 3.3f is a special case of this), we can construct rectilinear domains for any (m_H, m_V) pair satisfying $m_V \leq 2(m_H - 1)$ and $m_H \leq 2(m_V - 1)$, such that the polygon satisfies Theorem 3.1 sharply. The analysis in Section 3.2 immediately yields that if $m_V = 1$ or $m_H = 1$, then $m_V + m_H - 1$ is an upper bound for the minimum size of a point guard system (see Proposition 3.7), whose sharpness is shown by combs (Fig. 3.1).

3.2 Translating the problem into the language of graphs

For graph theoretical notation and theorems used in this chapter (say, the block decomposition of graphs), the reader is referred to [Die10].

Definition 3.2 (Chordal bipartite or bichordal graph, [GG78])**.** A graph G is chordal bipartite iff any cycle C of ≥ 6 vertices of G has a chord (that is $E(G[C]) \supsetneqq E(C)$).

Let S_V be the set of internally disjoint rectangles we obtain by cutting vertically at each reflex vertex of a rectilinear domain D. Similarly, let S_H be defined analogously for horizontal cuts of D. We may refer to the elements of these sets as **vertical and horizontal slices**, respectively. Let G be the intersection graph of S_H and S_V, i.e.,

$$G = \left(S_H \cup S_V, \left\{\{h, v\} \ : \ h \in S_H,\ v \in S_V,\ \mathrm{int}(h) \cap \mathrm{int}(v) \neq \varnothing\right\}\right).$$

In other words, a horizontal and a vertical slice are joined by an edge iff their interiors intersect; see Fig. 3.4. We may also refer to G as the **pixelation graph** of D. Clearly, the **set of pixels** $\{\cap e \mid e \in E(G)\}$ is a cover of D. Let us define $c(e)$ as the center of gravity of $\cap e$ (the pixel determined by e).

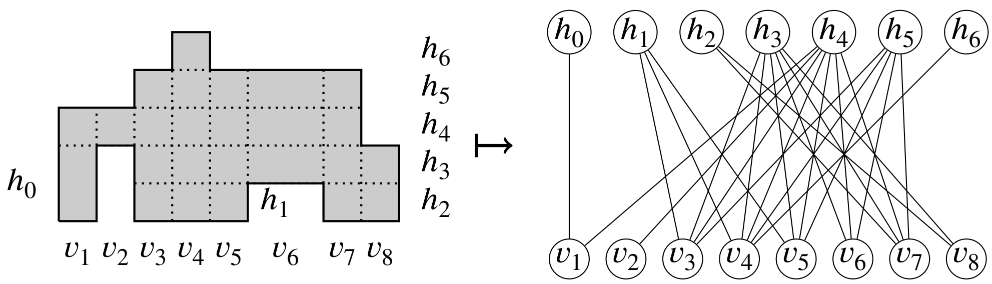


Fig. 3.4 A rectilinear domain and its associated pixelation graph

The horizontal R-tree T_H of D defined in Section 2.3 is equal to

$$T_H = \left(S_H, \left\{\{h_1, h_2\} \subseteq S_H \ : \ h_1 \neq h_2,\ h_1 \cap h_2 \neq \varnothing\right\}\right),$$

i.e., T_H is the intersection graph of the horizontal slices of D. Similarly, T_V is the intersection graph of the vertical slices of D.

Lemma 3.3. *G is a connected chordal bipartite graph.*

Proof. Connectedness of D immediately yields that G is connected too. Suppose C is a cycle of ≥ 6 vertices in G. For each node of the cycle C, connect the centers of gravity of its two incident edges with a line segment. This way we get an orthogonal polygon P in D.

If P is self-intersecting, then the vertices which are represented by the two intersecting line segments are intersecting. This clearly corresponds to a chord of C in G.

If P is simple, then the number of its vertices is $|V(C)|$, thus one of them is a reflex vertex, say $c(v_1 \cap h_1)$ is one. As P lives in D, its interior is a subset of D as well (here we use that D is simply connected). The simpleness of P also implies that the vertical line segment intersecting $c(v_1 \cap h_1)$ after entering the interior of P at $c(v_1 \cap h_1)$, intersects P at least once more when it emerges, say at $c(v_1 \cap h_2)$. As this is not an intersection of the line segments corresponding to two vertices of D, the edge $\{v_1, h_2\}$ is a chord of C. □

It is worth mentioning that even if D is a rectilinear domain with rectilinear hole(s), G may still be chordal bipartite. Take, for example, $[0,3]^2 \setminus (1,2)^2$; the graph associated to it has only one cycle, which is of length 4.

We will use the following technical claim to translate r-vision of points of D into relations in G.

Claim 3.4. *Let $e_1, e_2 \in E(G)$, where $e_1 = \{v_1, h_1\}$, $e_2 = \{v_2, h_2\}$, $v_1, v_2 \in S_V$, and $h_1, h_2 \in S_H$. The points $p_1 \in \operatorname{int}(\cap e_1)$ and $p_2 \in \operatorname{int}(\cap e_2)$ have r-vision of each other in D iff $e_1 \cap e_2 \neq \emptyset$ or $e_1 \cup e_2$ induces a 4-cycle in G.*

Proof. If $v_1 \in e_1 \cap e_2$, then $p_1, p_2 \in v_1$, therefore p_1 and p_2 have r-vision of each other. If $h_1 \in e_1 \cap e_2$, the same holds. If $\{v_1, h_1, v_2, h_2\}$ induces a 4-cycle, then

$$\operatorname{Conv}((v_1 \cap h_1) \cup (v_1 \cap h_2)) \subseteq v_1 \subseteq D$$

by v_1's convexity. Moreover,

$$\begin{aligned} B =&\operatorname{Conv}((v_1 \cap h_1) \cup (v_1 \cap h_2)) \cup \operatorname{Conv}((v_1 \cap h_2) \cup (v_2 \cap h_2)) \cup \\ &\cup \operatorname{Conv}((v_2 \cap h_2) \cup (v_2 \cap h_1)) \cup \operatorname{Conv}((v_2 \cap h_1) \cup (v_1 \cap h_1)) \end{aligned}$$

is contained in D. Since D is simply connected, we have $\operatorname{Conv}(B) \subseteq D$, which is a rectangle containing both p_1 and p_2.

In the other direction, suppose $e_1 \cap e_2 = \emptyset$. If R is an axis-aligned rectangle which contains both p_1 and p_2, then R clearly intersects the interiors of each element of $e_1 \cup e_2$, which implies that $\operatorname{int}(v_2) \cap \operatorname{int}(h_1) \neq \emptyset$ and $\operatorname{int}(v_1) \cap \operatorname{int}(h_2) \neq \emptyset$. Thus $e_1 \cup e_2$ induces a cycle in G. □

This easily implies the following claim.

Claim 3.5. *Two points $p_1, p_2 \in D$ have r-vision of each other iff $\exists e_1, e_2 \in E(G)$ such that $p_1 \in \cap e_1$, $p_2 \in \cap e_2$, and either $e_1 \cap e_2 \neq \emptyset$ or $e_1 \cup e_2$ induces a 4-cycle in G.*

These claims motivate the following definition.

Definition 3.6 (r-vision of edges)**.** For any $e_1, e_2 \in E(G)$ we say that e_1 and e_2 have r-vision of each other iff $e_1 \cap e_2 \neq \emptyset$ or there exists a C_4 in G which contains both e_1 and e_2.

Let $Z \subseteq E(G)$ be such that for any $e_0 \in E(G)$ there exists an $e_1 \in Z$ so that e_1 has r-vision of e_0. According Claim 3.5, if we choose a point from $\text{int}(\cap e_1)$ for each $e_1 \in Z$, then we get a point r-guard system of D.

Observe that any vertical mobile r-guard is contained in $\text{int}(v)$ for some $v \in S_V$ (except $\leq$ 2 points of the patrol). Extending the line segment the mobile guard patrols increases the area that it covers, therefore we may assume that this line segment intersects each element of $\{\text{int}(\cap e) \mid v \in e \in E(G)\}$, which only depends on some $v \in S_V$. Using Claim 3.5, we conclude that the set which such a mobile guard covers with r-vision is exactly $\cup\{h \in S_H \mid \{h, v\} \in E(G)\}$. The analogous statement holds for horizontal mobile guards as well.

Thus, a set of mobile guards of D can be represented by a set $M_V \subseteq S_V$. Clearly, M_V covers D if and only if

$$D = \bigcup_{v \in M_V} \left(\bigcup N_G(v) \right), \text{ which holds iff } S_H = \bigcup_{v \in M_V} N_G(v),$$

or in other words, M_V dominates each element of S_H in G. Similarly, a horizontal mobile guard system has a representative set $M_H \subseteq S_H$, which dominates S_V in G. Equivalently, $M_H \cup M_V$ is a totally dominating set of G, i.e., a subset of $V(G)$ that dominates every node of G (even the nodes of $M_H \cup M_V$).

The same arguments imply that a mixed set of vertical and horizontal mobile r-guards is represented by a set of vertices of $S \subseteq V(G)$. The set of guards is a covering system of guards of D if and only if every node $V(G) \setminus S$ has neighbor in S, i.e., S is a dominating set in G. Table 3.1 is the dictionary that lists the main notions of the original problem and their corresponding phrasing in the pixelation graph.

As promised, the following claim has a very short proof using the definitions and claims of this section.

Proposition 3.7. *If $m_V = 1$ or $m_H = 1$, then $p \leq m_V + m_H - 1$.*

Orthogonal polygon	**Pixelation graph**
Mobile guard	Vertex
Point guard	Edge
Simply connected	Chordal bipartite ($\Rightarrow$, but $\not\Leftarrow$)
r-vision of two points	$e_1 \cap e_2 \neq \varnothing$ or $G[e_1 \cup e_2] \cong C_4$
Horiz. mobile guard cover	$M_H \subseteq S_H$ dominating S_V
Covering system of mobile guards	Dominating set

Table 3.1 Translating the orthogonal art gallery problem to the pixelation graph

Proof. Let Z be the set of edges of G induced by $M_H \cup M_V$. Clearly, $G[M_H \cup M_V]$ is a star, thus $|Z| = |M_H| + |M_V| - 1$.

We claim that Z covers $E(G)$. There exist two slices, $h_1 \in M_H$ and $v_1 \in M_V$, which are joined by an edge to v_0 and h_0, respectively. Since $G[M_H \cup M_V]$ is a star, $\{v_1, h_1\} \in Z$. This edge has r-vision of e_0, as either $\{v_1, h_1\}$ intersects e_0, or $\{v_0, h_0, v_1, h_1\}$ induces a C_4 in Z. □

Finally, we can state Theorem 3.1 in a stronger form, conveniently via graph theoretic concepts.

Theorem 3.1′. *Let A_V be a set of internally disjoint axis-parallel rectangles of a rectilinear domain D (we call them vertical slices). Similarly, let A_H be another set with the same property, whose elements we call the horizontal slices. Also, suppose that for any $v \in A_V$, its top and bottom sides are a subset of ∂D, and for any $h \in A_H$, its left and right sides are a subset of ∂D. Furthermore, suppose that their intersection graph*

$$G = \left(A_H \cup A_V, \left\{\{h, v\} \subseteq A_V \cup A_H \ :\ \mathrm{int}(v) \cap \mathrm{int}(h) \neq \varnothing\right\}\right)$$

is connected.

If $M_V \subseteq A_V$ dominates A_H in G, and $M_H \subseteq A_H$ dominates A_V in G, then there exists a set of edges $Z \subseteq E(G)$ such that any element of $E(G)$ is r-visible from some element of Z, and

$$|Z| \leq \frac{4}{3} \cdot (|M_V| + |M_H| - 1).$$

Now we are ready to prove the main theorem of this chapter.

3.3 Proof of Theorem 3.1′

Both A_H and A_V can be extended to a partition of D (while preserving the assumptions of the theorem on them), so G is a subgraph induced by $A_H \cup A_V$ in a chordal bipartite graph (see Lemma 3.3), thus G is chordal bipartite as well. Let $M = G[M_V \cup M_H]$ be the subgraph induced by the dominating sets. Notice, that the bichordality of G is inherited by M.

Claim 3.9. *If M is connected, then any edge $e_0 = \{h_0, v_0\} \in E(G)$ is r-visible from some edge of M.*

Proof. As $N_G(M_V \cup M_H) = V(G)$, there exists two vertices, $v_1 \in M_V$ and $h_1 \in M_H$, such that $\{v_1, h_0\}, \{v_0, h_1\} \in E(G)$.

If $v_0 \in M_V$ or $h_0 \in M_H$, then $\{v_0, h_1\}$ or $\{v_1, h_0\}$ is in $E(M)$.

Otherwise, there exists a path in M, whose endpoints are v_1 and h_1, and this path and the edges $\{v_1, h_0\}$,$\{h_0, v_0\}$,$\{v_0, h_1\}$ form a cycle in G. By the bichordality of G, there exists a C_4 in G which contains an edge of M and e_0. □

We distinguish 3 cases based on the level connectivity of M.

3.3.1 *M* is 2-connected

The $\frac{4}{3}$ constant in the statement of Theorem 3.1′ is determined by this case. Knowing this, it is not surprising that this is the longest and most complex case of the proof.

If $E(M)$ consists of a single edge e, then $Z = \{e\}$ is clearly a point guard system of G by Claim 3.9.

Suppose now, that M has more than two vertices. Any edge of M is contained in a cycle of M, and by the bichordality property, there is such a cycle of length 4. It is easy to see that the convex hull of the pixels determined by the edges of a C_4 is a rectangle. Define

$$D_M = \bigcup_{\{e_1,e_2,e_3,e_4\} \text{ is a } C_4 \text{ in } M} \text{Conv}\left(\bigcup_{i=1}^{4} \cap e_i \right).$$

The simply connectedness of D implies that $D_M \subseteq D$.

Claim 3.10. *For any slice $s \in V(M)$ the intersection of s and D_M is connected.*

Proof. Suppose that $e_1, e_2 \in E(M)$ are such that $\cap e_1$ and $\cap e_2$ are in two different components of $s \cap D_M$. Since M is 2-connected, there is a path connecting $e_1 \setminus \{s\}$ and $e_2 \setminus \{s\}$ in $M - s$.

Take the shortest cycle in M containing e_1 and e_2. If this cycle contains 4 edges, then the convex hull of their pixels is in D_M, which is a contradiction. Similarly, if the cycle contains more than 4 edges, the bichordality of M implies that s is joined to every second node of the cycle, which contradicts our assumption that $s \cap D_M$ is disconnected. □

Claim 3.11. *For any slice $s \in V(G)$, the intersection of* $\mathrm{int}(s)$ *and D_M is connected.*

Proof. If $s \in V(M)$, we are done by Claim 3.10. If $s \in V(G) \setminus V(M)$, let $e_1, e_2 \in E(G)$ be the two edges such that $e_1 \cap e_2 = \{s\}$, $\partial(\cap e_1) \bigcap \partial D_M \neq \emptyset$, $\partial(\cap e_2) \bigcap \partial D_M \neq \emptyset$. Then, we must have $\left(e_1 \bigcup e_2\right) \setminus \{s\} \subseteq V(M)$. Take the shortest path in M joining $e_1 \setminus \{s\}$ to $e_2 \setminus \{s\}$. The proof can be finished as that of the previous claim. □

Let $B_H \subset M_H$ be the set of those slices whose top and bottom sides both intersect ∂D_M in an uncountable number of points of $\mathbb{R}^2$.

For technical reasons, we split each element of $h \in B_H$ horizontally through $c(h)$ to get two isometric rectangles in $\mathbb{R}^2$; let the set of the resulting refined horizontal slices be B'_H. Replace the elements of A_H and M_H contained in B_H with their corresponding two halves in B'_H to get

$$A'_H = B'_H \bigcup A_H \setminus B_H \quad \text{and} \quad M'_H = B'_H \bigcup M_H \setminus B_H,$$

respectively. Let $A'_V = A_V$, $M'_V = M_V$. Let τ be the function which maps $h \in B'_H$ to the $\tau(h) \in A_H$ for which $h \subseteq \tau(h)$ holds, and let τ be the identity function on $A'_V \cup A'_H \setminus B'_H$.

Let G' be the intersection graph of A'_H and A'_V (as in the statement of Theorem 3.1′). Also, let $M' = G'[M'_H \cup M'_V] = \tau^{-1}(M)$. Observe that τ naturally defines a graph homomorphism $\tau : G' \to G$ (edges are mapped vertex-wise).

Claim 3.12. *In G', the set M'_H dominates A'_V, and M'_V dominates A'_H. Furthermore, if $Z' \subseteq E(M')$ is a point guard system of G', then $Z = \tau(Z') \subseteq E(M)$ is a point guard system of G.*

Proof. The first statement of this claim holds, since τ maps non-edges to non-edges, and both $M'_H = \tau^{-1}(M_H)$ and $M'_V = \tau^{-1}(M_V)$ by definition. As τ is a graph homomorphism, it preserves r-visibility, which implies the second statement of this claim. □

Notice, that M' is 2-connected and $D_M = D_{M'}$. An edge $e \in E(M')$ falls into one of the following 4 categories:

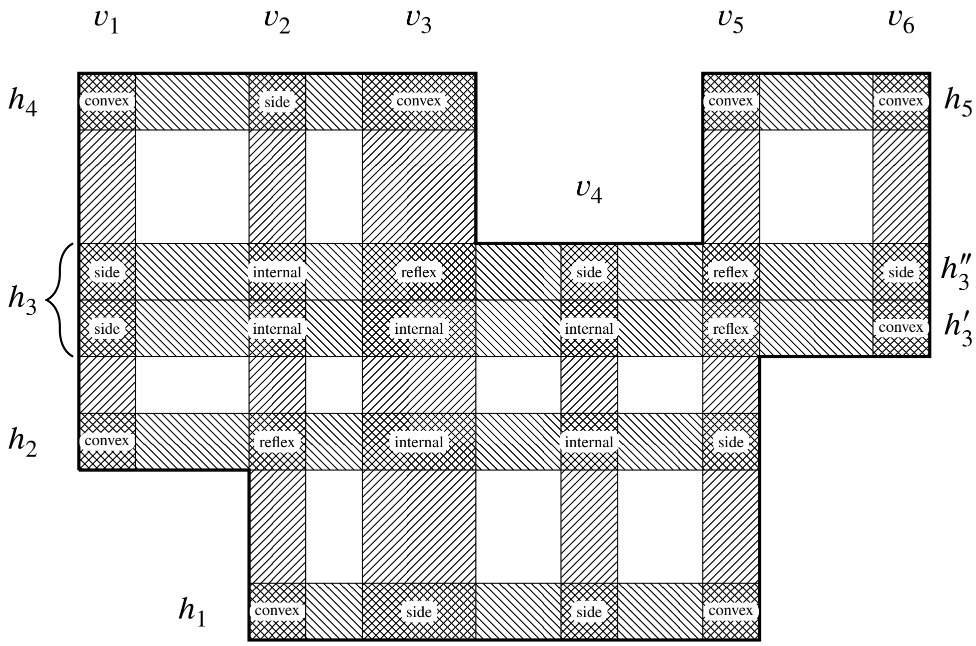


Fig. 3.5 We have $M_H = \{h_1, h_2, h_3, h_4, h_5\}$, $M'_H = M_H - h_3 + h'_3 + h''_3$, and $M_V = M'_V = \{v_1, v_2, v_3, v_4, v_5, v_6\}$. The thick line is the boundary of D_M. Each rectangle pixel is labeled according to the type of its corresponding edge of M'.

Convex edge: 3 vertices of $\cap e$ fall on ∂D_M, e.g., the edge $\{h_2, v_1\}$ on Fig. 3.5;

Reflex edge: exactly 1 vertex of $\cap e$ falls on ∂D_M, e.g., $\{h''_3, v_3\}$ on Fig. 3.5;

Side edge: two neighboring vertices of $\cap e$ fall on ∂D_M, e.g., $\{h_1, v_4\}$ on Fig. 3.5;

Internal edge: zero vertices of $\cap e$ fall on D_M, e.g., $\{h_2, v_3\}$ on Fig. 3.5.

Notice that on Fig. 3.5, the edge $\{h_3, v_5\}$ falls into neither of the previous categories, as two non-neighboring (diagonally opposite) vertices of pixel $h_3 \cap v_5$ fall on D_M. This clearly cannot happen with edges of G', but G may contain edges of this type.

Observe that τ maps convex edges to convex edges, and side edges to side edges. Conversely, the preimages of a convex edge are a convex edge and a side edge (M' is 2-connected), the preimages of a side edge are two side edges, and the preimages of a reflex edge are a reflex edge and an internal edge.

The following definition allow us to break our proof into smaller, transparent parts, which ultimately boils down to presenting a precise proof. It captures a condition which in certain circumstances allows us to conclude that a guard e_1 can be replaced by e_2 such that we still have complete coverage of G'.

Definition 3.13. For any two edges $e_1, e_2 \in E(M')$, where $e_1 = \{v_1, h_1\}$ and $e_2 = \{v_2, h_2\}$, we write $e_2 \to e_1$ (e_2 dominates e_1) iff either

- $e_1 \cap e_2 \subset A'_H$, and $\exists h_3, h_4 \in M'_H$ such that $\{v_1, v_2, h_3, h_4\}$ induces a C_4 in M', and $h_1 = h_2$ is between h_3 and h_4; or
- $e_1 \cap e_2 \subset A'_V$, and $\exists v_3, v_4 \in M'_V$ such that $\{v_3, v_4, h_1, h_2\}$ induces a C_4 in M', and $v_1 = v_2$ is between v_3 and v_4; or
- $e_1 \cap e_2 = \emptyset$, and $\exists v_3 \in M'_V$ and $h_3 \in M'_H$ such that both $\{v_1, h_2, v_2, h_3\}$ and $\{h_1, v_3, h_2, v_2\}$ induces a C_4 in M'; furthermore, v_1 is between v_2 and v_3, and h_1 is between h_2 and h_3.

We write $e_2 \leftrightarrow e_1$ iff both $e_2 \to e_1$ and $e_1 \to e_2$ hold. Note that $\leftrightarrow$ is a symmetric, but generally intransitive relation.

For example, on Fig. 3.5, $\{h_1, v_3\} \leftrightarrow \{h''_3, v_3\}$, and $\{h_1, v_2\} \to \{h''_3, v_3\}$. Also, $\{h''_3, v_3\} \leftrightarrow \{h''_3, v_1\}$, but $\{h''_3, v_3\} \nleftrightarrow \{h'_3, v_1\}$. This is a technicality which makes the proofs easier, but does not cause any issues in the end, as $\tau(\{h''_3, v_1\}) = \tau(\{h'_3, v_1\})$.

We will search for a point guard system of M' with very specific properties, which are described by the following definition.

Definition 3.14. Suppose $Z' \subseteq E(M')$ is such, that

1. Z' contains every convex edge of M,
2. for any non-internal edge $e_1 \in E(M) \setminus Z'$, there exists some $e_2 \in Z'$ for which $e_2 \to e_1$, and
3. for each $h_0 \in A'_H$ for which $\mathrm{int}(h_0) \cap D_M \neq \emptyset$ holds, $\exists \{h_2, v_2\} \in Z'$ such that $\{h_0, v_2\} \in E(M')$ and $N_{M'}(h_2) \supseteq N_{G'}(h_0) \bigcap M'_V$.

If these three properties hold, we call Z' a **hyperguard** of M'.

Lemma 3.15. *Any hyperguard Z' of M' is a point guard system of G', i.e., any edge of G' is r-visible from some element of Z'.*

Proof. Let $e_0 = \{v_0, h_0\} \in E(G')$ be an arbitrary edge. By Claim 3.9, there exists an edge $e_1 \in E(M')$ which has r-vision of e_0, and we also suppose that e_1 is chosen so that Euclidean distance $\text{dist}(\cap e_0, \cap e_1)$ is minimal.

Trivially, if $e_1 \in Z'$ (for example, if e_1 is a **convex edge** of M'), then e_0 is r-visible from e_1. Assume now, that $e_1 \notin Z'$.

- If e_1 is a **reflex or side edge** of M', then $\exists e_2 \in Z'$ so that $e_2 \to e_1$. We claim that e_2 has r-vision of e_0 in G' (this is the main motivation for Definition 3.13).

  1. If $e_1 \cap e_2 \subset A'_H$: by the choice of e_1 and e_2, v_1 is joined to h_0, h_3, h_4 in G'. The choice of e_1 guarantees that $v_1 \cap h_0$ is between $v_1 \cap h_3$ and $v_1 \cap h_4$. Therefore $\text{int}(v_2 \cap h_0) \neq \emptyset$, so $\{v_0, h_0, v_2, h_1(= h_2)\}$ induces a C_4 in G'.
  2. If $e_1 \cap e_2 \subset A'_V$: the proof proceeds analogously to the previous case.
  3. If $e_1 \cap e_2 = \emptyset$: by the choice of e_1 and e_2, v_1 is joined to h_0, h_3, h_2 in G', and v_1 is joined to v_0, v_3, v_2 in G'. The choice of e_1 guarantees that $v_1 \cap h_0$ is between $v_1 \cap h_3$ and $v_1 \cap h_2$, and that $v_0 \cap h_1$ is between $v_3 \cap h_1$ and $v_2 \cap h_1$. Therefore $\text{int}(v_2 \cap h_0) \neq \emptyset$ and $\text{int}(v_0 \cap h_2) \neq \emptyset$, so $\{v_0, h_0, v_2, h_2\}$ induces a C_4 in G'.

  In any of the three cases, e_0 is r-visible from e_2 in G'.

- If e_1 is an **internal edge** of M', then $\cap e_0 \subset D_M$. By the 3^{rd} property of hyperguards, there $\exists\{h_2, v_2\} \in Z'$ such that $\{h_0, v_2\} \in E(M')$ and $N_{M'}(h_2) \supseteq N_{G'}(h_0) \bigcap M'_V$. An easy argument (use that $D_M \subset D$ are both simply connected) gives that $\{v_0, h_2\} \in E(G')$. Thus $\{v_0, h_2, v_2, h_0\}$ induces a C_4 in G', so e_0 is r-visible from $\{v_2, h_2\} \in Z'$.

We have verified the statement in every case, so the proof of this lemma is complete. □

Notice, that the set of all convex, reflex, and side edges of $E(M')$ form a hyperguard of M'. By Lemma 3.15, this set is a point guard system of G', and Claim 3.12 implies that its τ-image is a point guard system of G. The cardinality of the τ-image of this hyperguard is bounded by $2|V(M)| - 4$ (we will see this shortly), which is already a magnitude lower than what the trivial choice of $E(M)$ would give (generally, $|E(M)|$ can be equal to $\Omega(|V(M)|^2)$).

Let the number of convex, side, and reflex edges in M' be c', s', and r', respectively. Claim 3.10 and 3.11 allow us to count these objects.

1. The number of reflex vertices of D_M is equal to r': any reflex vertex is a vertex of a reflex edge, and the way M' and D_M is constructed guarantees that exactly one vertex of the pixel of a reflex edge is a reflex vertex of D_M.

2. The number of convex vertices of D_M is equal to c': any convex vertex is a vertex of the pixel of a convex edge, and the way D_M is constructed guarantees that exactly one vertex of the pixel of a convex edge is a convex vertex.

3. The cardinality of $V(M')$ is $c' + \frac{1}{2}s'$: the first and last edge incident to any element of $V(M')$ ordered from left-to-right (for elements of M'_H) or from top-to-bottom (for elements of M'_V) is a convex or a side edge. Conversely, any convex edge is the first or last incident edge of exactly one element of M'_H and one element of M'_V. A side edge is the first or last incident edge of exactly one element of $V(M')$.

4. For any reflex edge $e_1 = \{v_1, h_1\} \in E(M')$, there is exactly one reflex or side edge in $E(M')$ which contains v_1 and is in the $\leftrightarrow$ relation with e_1, and the same can be said about h_1.

5. Any side edge $e_1 \in E(M')$ is in $\leftrightarrow$ relation with exactly one reflex or side edge which it intersects. The intersection is the slice in $V(M')$ on which e_1 is a boundary edge.

We can now compute the size of the set of all convex, reflex, and side edges of $E(M')$:

$$c' + r' + s' = 2c' - 4 + s' = 2|V(M')| - 4.$$

Furthermore, it is clear that taking the τ-image of this set decreases its by cardinality by $2|B_H|$ (new reflex and side edges are created at both ends of slices in B_H when splitting them), and $2|V(M')| - 4 - 2|B_H| = 2|V(M)| - 4$, proving the claim from the previous page. Readers who are only interested in a result which is sharp up to a constant factor, may skip to Case 3.3.2. Further analysis of M' allows us to lower the coefficient 2 to $\frac{4}{3}$.

Define the **auxiliary graph X** as follows: let $V(X)$ be the set of reflex and side edges of M', and let

$$E(X) = \Big\{ \{e, f\} \; : \; e \neq f, \; e \cap f \neq \emptyset, \; e \leftrightarrow f \Big\}.$$

By our observations, X is the disjoint union of some cycles and $\frac{1}{2}s'$ paths. This structure allows us to select a hyperguard which contains a subset of the reflex and side edges of M', instead of the whole set.

3.3.1.1 Constructing a hyperguard Z' of M'.

We will define $(Z'_j)_{j=0}^{\infty}$, a sequence of (set theoretically) increasing sequence of subsets of $E(M')$, and $(X_j)_{j=0}^{\infty}$, a decreasing sequence of induced subgraphs of X.

Additionally, we will define a function $w_j : V(X) \to \{0, 1, 2\}$, and extend its domain to any subgraph $H \subseteq X$ by defining $w_j(H) = \sum_{e \in V(H)} w_j(e)$. The purpose of w_j, very vaguely, is that as Z' will contain every third node of X, we need to keep count of the modulo 3 remainders. Furthermore, w_j serves as buffer in a(n implicitly defined) weight function (see inequality (3.2)).

For a set $E_0 \subseteq E(X)$, let the indicator function of E_0 be

$$\mathbb{1}_{E_0}(e) = \begin{cases} 1, & \text{if } e \in E_0, \\ 0, & \text{if } e \in E(X) \setminus E_0. \end{cases}$$

Let $Z'_0 = \emptyset$ and $X_0 = X$. By our previous observations, X does not contain isolated nodes. Define $w_0 : V(X) \to \{0, 1, 2\}$ such that

$$w_0(e) = \begin{cases} 1, & \text{if } d_{X_0}(e) = 1, \\ 0, & \text{if } d_{X_0}(e) = 0 \text{ or } 2. \end{cases}$$

In each step, we will define Z'_j, X_j, and w_j so that

- $Z'_{j-1} \subseteq Z'_j$, $X_j \subseteq X_{j-1}$,
- $\{e \in V(X_j) \mid d_{X_j}(e) = 1\} \subseteq w_j^{-1}(1)$,
- $\{e \in V(X_j) \mid d_{X_j}(e) = 0\} = w_j^{-1}(2)$, and
- $\forall e_0 \in V(X) \setminus V(X_j)$, either $e_0 \in Z'_j$, or $\exists e_1 \in Z'_j$ so that $e_1 \to e_0$.

If these hold, then for any path component P_j in X_j, we have $w_j(P_j) \geq 2$.

Phase 1 Let the set of convex edges of M' be C'. Let

$$
\begin{aligned}
S' &= \left\{e \in V(X) : \ \tau(e) \text{ is a side edge}, \exists f \in V(X)\ e \leftrightarrow f,\ e \cap f \subseteq B'_H\right\},\\
T' &= \left\{f \in V(X) : \ \exists e \in S'\ f \leftrightarrow e,\ \tau^{-1}(\tau(f)) \setminus \{f\} \to N_X(\tau^{-1}(\tau(e))) \setminus \{f\}\right\},\\
U' &= \tau^{-1}(\tau(C')) \setminus C',\\
Q' &= \bigcup_{\substack{e_1,e_4 \in S' \cup U' \\ e_2,e_3 \in V(X) \\ e_1 \leftrightarrow e_2, e_2 \leftrightarrow e_3, e_3 \leftrightarrow e_4}} \{e_1, e_2, e_3, e_4\}.
\end{aligned}
$$

Take

$$
\begin{aligned}
Z'_1 &= \tau^{-1}\left(\tau(C') \bigcup \tau(T')\right),\\
X_1 &= X - T' - N_X(T') - U' - N_X(U'),\\
w_1 &= w_0 - \mathbb{1}_{S'} - \mathbb{1}_{U'} + \sum_{f \in T'} \mathbb{1}_{N_X(N_X(f)) \setminus \{f\} \setminus Q'} + \sum_{e \in U'} \mathbb{1}_{N_X(N_X(e)) \setminus \{e\} \setminus Q'}.
\end{aligned}
$$

Phase 2 Take a cycle $e_1, e_2, \ldots, e_{2k_j}$ in X_j ($k_j \geq 2$, $j \geq 1$). This set of nodes of X_j is the edge set of a cycle of length $2k_j$ in M'.

- If $2k_j = 4$, observe that $e_1 \leftrightarrow e_2$, $e_1 \leftrightarrow e_4$, $e_2 \leftrightarrow e_3$, $e_4 \leftrightarrow e_3$ together imply that $e_1 \leftrightarrow e_3$. Take

$$
\begin{aligned}
Z'_{j+1} &= \{e_1\} \bigcup Z'_j,\\
X_{j+1} &= X_j - \{e_1, e_2, e_3, e_4\},\\
w_{j+1} &= w_j.
\end{aligned}
$$

- If $2k_j \geq 6$, the chordal bipartiteness of M' implies that without loss of generality there is a chord $f \in E(M')$ which forms a cycle with e_1, e_2, e_3 in M'. Take

$$
\begin{aligned}
Z'_{j+1} &= \{f\} \bigcup Z'_j,\\
X_{j+1} &= X_j - \{e_{2k_j}, e_1, e_2, e_3, e_4\},\\
w_{j+1} &= w_j + \mathbb{1}_{e_5} + \mathbb{1}_{e_{2k_j-1}}.
\end{aligned}
$$

Iterate this step until X_{j_1} is cycle-free.

Phase 3 Take a path $e_1, e_2, \dots, e_k$ in X_j (for $j \geq j_1$), such that

$$E\left(M'\left[\bigcup_{i=2}^{k-1} e_i\right]\right) \setminus \{e_2, \dots e_{k-1}\} \neq \emptyset.$$

Using the bichordality of M', there exists a chord $f \in E(M')$ which forms a C_4 with $\{e_{l-1}, e_l, e_{l+1}\}$, where $3 \leq l \leq k-2$. It is easy to see that $e_{l-2} \leftrightarrow e_{l-1}$ implies $f \to e_{l-2}$ and $f \to e_{l-1}$. Similarly, we have that $f \to e_{l+1}$ and $f \to e_{l+2}$. Also, $f \to e_{l-1}$ and $f \to e_{l+1}$ together imply $f \to e_l$. Therefore, we take

$$\begin{aligned}
Z'_{j+1} &= \{f\} \bigcup Z'_j, \\
X_{j+1} &= X_j - \{e_{l-2}, e_{l-1}, e_l, e_{l+1}, e_{l+2}\}, \\
w_{j+1} &= w_j + \mathbb{1}_{\{\text{dist}_X(\bullet, e_l) = 3\}}.
\end{aligned}$$

Iterate this step until X_{j_2} is free of the above defined paths.

Phase 4 The set A'_H is the subset of the nodes of a horizontal R-tree of D. Let $h_{\text{root}} \in A'_H$ be an arbitrarily chosen node serving as the root of the horizontal R-tree. Process the elements of A'_H in decreasing distance (measured in the horizontal R-tree) from h_{root}.

Suppose $h_0 \in A'_H$ is the next horizontal slice to be processed. If $\text{int}(h_0) \cap D_M = \emptyset$ or $h_0 \in M'_H$, then move on to the next slice of A'_H, as the 3rd property of hyperguards for Z' is satisfied by any edge of M' incident to h_0.

Suppose now, that $h_0 \notin M'_H$. It is easy to see that there exists a C_4 in M' whose edge set $\{e_1, e_2, e_3, e_4\}$ satisfies

$$h_0 \cap D_M \subset \text{Conv}\left(\bigcup_{i=1}^{4} \cap e_i\right).$$

Without loss of generality, we may suppose that we choose the C_4 so that the convex hull of the pixels of its edges is minimal. Then e_i (for $i = 1, 2, 3, 4$) is not an internal-edge of M', as this would contradict the choice of the C_4.

If e_i is a convex edge of M', then it is already contained in $Z'_1 \subset Z'$, so it satisfies the 3rd property of hyperguards for Z' and h_0, and we may skip to processing the next slice. If e_i

is a side edge of M', then for any edge $f \in Z'$ which satisfies $f \to e_i$, we have $\cap f \subset \mathrm{Conv}\left(\bigcup_{i=1}^{4} \cap e_i\right)$, so f satisfies the 3rd property of hyperguards for Z' and h_0, and again, we may skip to processing the next slice.

Suppose now, that each e_i (for $i = 1, 2, 3, 4$) is a reflex edge of M'. Let

$$\{h_1, h_2\} = M'_H \bigcap \bigcup_{i=1}^{4} e_i \quad \text{and} \quad \{v_1, v_2\} = M'_V \bigcap \bigcup_{i=1}^{4} e_i.$$

The minimality of the chosen C_4 implies that $\{h_1, v_1\} \leftrightarrow \{h_1, v_2\}$ and $\{h_2, v_1\} \leftrightarrow \{h_2, v_2\}$.

If $\{h_1, v_1\}, \{h_1, v_2\}$ were removed in Phase 2 or Phase 3 in one step, then the edge by which Z' is extended in the same step satisfies the 3rd property of hyperguards for Z' and h_0. The same holds for $\{h_2, v_1\}, \{h_2, v_2\}$. In both cases, we may skip to the next slice to be processed.

Without loss of generality, we may suppose that h_1 is farther away from the root of the horizontal R-tree than h_2.

If $\{\{h_1, v_1\}, \{h_1, v_2\}\} \cap V(X_j)$ is non-empty, take the path component P_j of X_j containing this set; otherwise let P_j be the empty graph. Observe, that Claim 3.11 implies that as a result of Phase 3, for any node $e \in V(P)$, its horizontal slice $e \cap M_H$ is at least as far away from the root as h_1.

Split the path P_j into two components $P_{j,1}$ and $P_{j,2}$ by deleting $\{h_1, v_1\}$ and $\{h_1, v_2\}$ (if one of them is not in $E(X_j)$, then one of the components is empty), so that $\{h_1, v_1\} \notin V(P_{j,2})$ and $\{h_1, v_2\} \notin V(P_{j,1})$.

- If $|V(P_{j,1})| \not\equiv 0 \pmod 3$ or $|V(P_{j,2})| \not\equiv 0 \pmod 3$, then let Y_j be a dominating set of P_j which contains $\{h_1, v_1\}$ or $\{h_1, v_2\}$, and is minimal with respect to these conditions. Set

$$\begin{aligned} Z'_{j+1} &= Y_j \bigcup Z'_j, \\ X_{j+1} &= X_j - P_j, \\ w_{j+1}(e) &= \begin{cases} 0, & \text{if } e \in V(P_j), \\ w_j(e) & \text{if } e \notin V(P_j). \end{cases} \end{aligned}$$

  Clearly, one of $\{h_1, v_1\}$ and $\{h_1, v_2\}$ is contained in $Y_j \subset Z'_{j+1} \subseteq Z'$, and it satisfies the 3rd property of hyperguards for Z' and h_0.

- If $|V(P_{j,1})| \equiv |V(P_{j,2})| \equiv 0 \pmod 3$, then let Y_j be a minimal dominating set of P_j. Moreover, if $\{\{h_2, v_1\}, \{h_2, v_2\}\} \bigcap (V(X_j) \bigcup Z'_j)$ is non-empty, let f_j be an element of it, otherwise set $f_j = \{h_2, v_1\}$. Take

$$\begin{aligned} Z'_{j+1} &= Y_j \bigcup \{f_j\} \bigcup Z'_j, \\ X_{j+1} &= X_j - P_j - \{f_j\} - N_{X_j}(\{f_j\}), \\ w_{j+1}(e) &= \begin{cases} 0, & \text{if } e \in V(P_j) \bigcup \{\{h_2, v_1\}, \{h_2, v_2\}\}, \\ w_j(e) + 1, & \text{if } \operatorname{dist}_X(e, f_j) = 2, \\ w_j(e) & \text{otherwise.} \end{cases} \end{aligned}$$

Observe, that f_j satisfies the 3$^{\text{rd}}$ property of hyperguards for Z' and h_0.

In any case, some element of $Z'_{j+1} \subseteq Z'$ satisfies the 3$^{\text{rd}}$ property of hyperguards for Z' and h_0. Furthermore, this holds for any slice of A'_H between h_1 and h_2, so we skip processing these elements.

Phase 5 Lastly, we get X_{j_3} which is the disjoint union of paths and isolated nodes (or it is an empty graph). Take a component P_j of X_j (for some $j \geq j_3$). Let Y_j be a dominating set of P_j (if $|V(P_j)| = 1$, then $Y_j = V(P_j)$). Take

$$\begin{aligned} Z'_{j+1} &= Y_j \bigcup Z'_j, \\ X_{j+1} &= X_j - P_j, \\ w_{j+1}(e) &= \begin{cases} 0, & \text{if } e \in V(P_j), \\ w_j(e) & \text{if } e \notin V(P_j). \end{cases} \end{aligned}$$

By repeating this procedure, eventually X_{j_4} is the empty graph for some $j_4 \geq j_3$.

Let $Z' = Z'_{j_4}$. This procedure is orchestrated in a way to guarantee that Z' is a hyperguard of M', so only an upper estimate on the cardinality of $\tau(Z')$ needs to be calculated to complete the proof of Case 3.3.1.

3.3.1.2 Estimating the size of $Z = \tau(Z')$.

We have

$$|V(X_0)| = r' + s', \; w_0(X) = s', \; |B'_H| = |T'| + |U'|.$$

By definition, $|Z_1'| = c' + |U'| + 2|T'|$ and $|\tau(Z_1')| = |Z_1'| - |B_H'|$. It is easy to check that

$$|V(X_1)| + w_1(X) + 2|U'| + 5|T'| \leq |V(X_0)| + w_0(X).$$

Therefore, we have

$$\begin{aligned} |Z_1'| + \frac{|V(X_1)| + w_1(X)}{3} &\leq c' + |U'| + 2|T'| + \frac{|V(X_1)| + w_1(X)}{3} \leq \\ &\leq c' + |B_H'| + \frac{|V(X_0)| + w_0(X) - 2|U'| - 2|T'|}{3} \leq \\ &\leq c' + |B_H'| + \frac{r' + 2s' - 2|B_H'|}{3}. \end{aligned} \tag{3.1}$$

We now show that

$$|Z_{j+1}'| + \frac{|V(X_{j+1})| + w_{j+1}(X)}{3} \leq |Z_j'| + \frac{|V(X_j)| + w_j(X)}{3}. \tag{3.2}$$

holds for any $j \geq 1$.

In Phase 2, we choose a node from each cycle of X_1. Inequality (3.2) is preserved, since

$$\begin{aligned} |Z_{j+1}'| &= |Z_j'| + 1, \\ |V(X_{j+1})| &= |V(X_j)| - 5 + \mathbb{1}_{\{4\}}(k_j), \\ w_{j+1}(X) &\leq w_j(X) + 2 - 2 \cdot \mathbb{1}_{\{4\}}(k_j). \end{aligned}$$

In Phase 3, for every $j_2 > j \geq j_1$, we have

$$\begin{aligned} |Z_{j+1}'| &= |Z_j'| + 1, \\ |V(X_{j+1})| &= |V(X_{j_1})| - 5, \\ w_{j+1}(X) &\leq w_j(X) + 2. \end{aligned}$$

Let $j_3 > j \geq j_2$. If $|V(P_{j,1})| \not\equiv 0 \pmod 3$ and $|V(P_{j,2})| \not\equiv 2 \pmod 3$, then take a dominating set of P_j containing $\{v_1, h_1\}$. We have

$$\begin{aligned} |Y_j| &\leq 1 + \left\lceil \frac{|V(P_{j,1})| - 2}{3} \right\rceil + \left\lceil \frac{|V(P_{j,2})| - 1}{3} \right\rceil \leq \\ &\leq 1 + \frac{|V(P_{j,1})| - 1}{3} + \frac{|V(P_{j,2})|}{3} = \frac{|V(P_j)| + 2}{3}. \end{aligned}$$

Similarly, if $|V(P_{j,1})| \not\equiv 2 \pmod 3$ and $|V(P_{j,2})| \not\equiv 0 \pmod 3$, then there is a small dominating set of P_j containing $\{h_1, v_2\}$. Also, if both $|V(P_{j,1})| \equiv 2 \pmod 3$ and $|V(P_{j,2})| \equiv 2 \pmod 3$, then there is a small dominating set of P_j containing $\{h_1, v_2\}$. Thus, if $|V(P_{j,1})| \not\equiv 0 \pmod 3$ or $|V(P_{j,2})| \not\equiv 0 \pmod 3$, then

$$\begin{aligned} |Z'_{j+1}| &= |Z'_j| + |Y_j| \le |Z'_j| + \frac{|V(P_j)|+2}{3}, \\ |V(X_{j+1})| &= |V(X_{j_1})| - |V(P_j)|, \\ w_{j+1}(X) &\le w_j(X) - 2. \end{aligned}$$

If both $|V(P_{j,1})| \equiv 0 \pmod 3$ and $|V(P_{j,2})| \equiv 0 \pmod 3$, then $|Y_j| = \frac{|V(P_j)|}{3}$. Observe, that

$$\{h_1, v_1\}, \{h_1, v_2\}, \{h_2, v_1\}, \{h_2, v_2\} \notin V(P_k) \text{ for any } k < j.$$

If both $\{h_1, v_1\} \notin Z'_j$ and $\{h_1, v_2\} \notin Z'_j$, but were removed in different steps, then when $\{h_1, v_1\}$ is removed in step k we must have set $w_k(\{h_1, v_2\}) = 1$, which is the consequence of the previous observation. Thus, $w_j(\{h_1, v_2\}) = 1$. Similarly, we must have $w_j(\{h_1, v_1\}) = 1$. This reasoning holds for $\{h_2, v_1\}$ and $\{h_2, v_2\}$, as well.

If P_j is not the empty graph or $f_j \in Z(X_j)$, then inequality (3.2) trivially holds. If P_j is the empty graph, then $w_j(\{h_1, v_1\}) = w_j(\{h_1, v_2\}) = 1$. If $f_j \in V(X_j)$, these 2 extra weights can be used to compensate for the new degree 1 vertices of X_{j+1}. If $f_j \notin Z(X_j) \bigcup V(X_j)$, then even $w_j(\{h_2, v_1\}) = w_j(\{h_2, v_2\}) = 1$, and in total the 4 extra weights compensate for adding f_j to Z'_{j+1}.

In any case, inequality (3.2) holds for $j_3 > j \ge j_2$.

For any $j_4 > j \ge j_3$, we have

$$|Y_j| \le \left\lceil \frac{|V(P_j)|}{3} \right\rceil \le \frac{|V(P_j)|+2}{3}$$

and $w_j(P_j) = 2$, so inequality (3.2) holds for j.

3.3.1.3 Summing it all up.

By definition, we have

$$|Z'| = |Z'_{j_4}|,\ X_{j_4} = \emptyset,\ 0 \le w_{j_4}(X).$$

Inequality (3.2) is preserved from Phase 2 up to Phase 5, therefore

$$|Z'| \le |Z'_{j_4}| + \frac{|V(X_{j_4})| + w_{j_4}(X)}{3} \le |Z'_1| + \frac{|V(X_1)| + w_1(X)}{3}.$$

Lastly, using inequality (3.1), we get

$$\begin{aligned}
&|Z| = |\tau(Z')| = |\tau(Z' \setminus Z'_1)| + |\tau(Z'_1)| \le |Z' \setminus Z'_1| + |Z'_1| - |B'_H| = \\
&= |Z'| - |B'_H| \le c' + \frac{r' + 2s' - 2|B'_H|}{3} = c' + \frac{(c' - 4) + 2s' - 2|B'_H|}{3} = \\
&= \frac{4\left(c' + \frac{1}{2}s'\right) - 4 - 2|B'_H|}{3} = \frac{4|V(M')| - 4 - 2|B'_H|}{3} = \\
&= \frac{4|M'_H| + 4|M'_V| - 4 - 2|B'_H|}{3} = \frac{4|M_H| + 4|B_H| + 4|M_V| - 4 - 2|B'_H|}{3} = \\
&= \frac{4(|M_H| + |M_V|) - 4}{3}, \text{ as desired.}
\end{aligned}$$

3.3.2 *M* is connected, but not 2-connected

Let the 2-connected components (or blocks) of M be M_i for $i = 1, \dots, q$. Since induced graphs of G inherit the chordal bipartite property, by Case 3.3.1, there exists a subset $Z_i \subseteq E(M_i)$, such that for any edge $e_0 \in E(G[N_G(M_i)])$, there exists an edge $e_1 \in Z_i$ which has r-vision of e_0 in $G[N(M_i)]$, and $|Z_i| \le \frac{4}{3}(|V(M_i)| - 1)$. Let $Z = \cup_{i=1}^{q} Z_i$.

Since the intersection graph of the vertex sets of the 2-connected components is a tree (and any two components intersect in zero or one elements), we have

$$|Z| \le \frac{4}{3}\left(-q + \sum_{i=1}^{q} |V(M_i)|\right) = \frac{4\left(-q + |V(M)| + (q-1)\right)}{3} = \frac{4(|V(M)| - 1)}{3}.$$

Furthermore, given an arbitrary $e_0 = \{v_0, h_0\} \in E(G)$, there exists a $v_1 \in M_V$ and an $h_1 \in M_H$ such that $\{v_1, h_0\}, \{v_0, h_1\} \in E(G)$.

- If $v_0 \in M_V$ or $h_0 \in M_H$, then $\{v_0, h_1\}$ or $\{v_1, h_0\}$ is in $E(M)$.
- Otherwise, there exists a path in M whose endpoints are v_1 and h_1, and this path and the edges $\{v_1, h_0\}$,$\{h_0, v_0\}$,$\{v_0, h_1\}$ form a cycle in G. By the bichordality of G, there exists a C_4 in G which contains an edge of M and e_0.

In any case, e_0 is r-visible from some $e_1 \in E(M)$. As e_1 is an edge of one of the 2-connected components M_i, we have $e_0 \subset N_G(M_i)$, therefore $e_0 \in E(G[N_G(M_i)])$. Thus, some $e_2 \in Z_i$ has r-vision of e_0.

3.3.3 *M* has more than one connected component.

Let us take a decomposition of M into connected components M_i for $i = 1, \dots, t$.

Let $N_i = N(M_i)$, so we have $M_i \subseteq N_i$ and $\cup_{i=1}^{t} N_i = V(G)$.

For all $i > 1$ let q_i be the number of components of $G[\cup_{k=1}^{i-1} N_k \setminus \cup_{k=i}^{t} N_k]$ to which $N_i \setminus \cup_{k=i+1}^{t} N_k$ is joined in $G[\cup_{k=1}^{i} N_k \setminus \cup_{k=i+1}^{t} N_k]$. Let $F_{i,j}$ be the set of edges joining $N_i \setminus \cup_{k=i+1}^{t} N_k$ to the j^{th} component of $G[\cup_{k=1}^{i-1} N_k \setminus \cup_{k=i}^{t} N_k]$. Furthermore, let $F_{i,j}^V = \{f \in F_{i,j} \mid f \cap A_V \cap N_i \neq \emptyset\}$ and $F_{i,j}^H = \{f \in F_{i,j} \mid f \cap A_H \cap N_i \neq \emptyset\}$.

Claim 3.16. *For any two edges $f_1, f_2 \in F_{i,j}^V$ either $f_1 \cap f_2 \neq \emptyset$ or $\exists f_3 \in F_{i,j}^V$ such that f_3 intersects both f_1 and f_2. The analogous statement holds for $F_{i,j}^H$.*

Proof. Suppose f_1 and f_2 are disjoint. Since M_i is connected, there is a path in G whose endpoints are $f_1 \cap N_i$ and $f_2 \cap N_i$, while its internal points are in $V(M_i)$; let the shortest such path be Q_1. There is also a path in the j^{th} component of $G[\cup_{k=1}^{i-1} N_k \setminus \cup_{k=i}^{t} N_k]$ whose endpoints are $f_1 \setminus N_1$ and $f_2 \setminus N_i$, let the shortest one be Q_2.

Now Q_1, f_1, Q_2, f_2 form a cycle in $G[\cup_{k=1}^{i} N_k \setminus \cup_{k=i+1}^{t} N_k]$, which is bipartite chordal. Since $V(Q_2) \cap N_i = \emptyset$, there cannot be a chord between $V(M_i) \cap V(Q_1)$ and $V(Q_2)$. This implies that $|V(Q_1)| = 3$ by its choice, and that either $(f_1 \cap N_i) \cup (f_2 \setminus N_i)$ or $(f_2 \cap N_i) \cup (f_1 \setminus N_i)$ is a chord. □

Claim 3.17. *For any two edges $f^V \in F_{i,j}^V$ and $f^H \in F_{i,j}^H$, the two-element set*

$$(f^V \cap N_i) \cup (f^H \cap N_i)$$

is an edge of $G[N_i]$.

Proof. Similar to the proof of Claim 3.16. □

Let $f_{i,j}^V \in F_{i,j}^V$ be the element which intersects the maximum number of edges from $F_{i,j}$, and choose $f_{i,j}^H \in F_{i,j}^H$ in the same way. If only one of these exist, let $w_{i,j}$ be the existing one, otherwise let $w_{i,j} = (f_{i,j}^V \cap N_i) \cup (f_{i,j}^H \cap N_i)$ (as in Claim 3.17). Let us finally define

$$W = \{w_{i,j} \mid i = 2, \dots, t \text{ and } j = 1, \dots, q_i\}.$$

Claim 3.18. $|W| = t - 1$.

Proof. Observe that for every $i = 1, \dots, t$, the subgraph $G[N_i \setminus \cup_{k=i+1}^t N_k]$ is connected, since $M_i \subseteq N_i \setminus \cup_{k=i+1}^t N_k \subseteq N_i = N(M_i)$. Moreover, $G[\cup_{k=1}^t N_k] = G$ is connected, therefore $t - 1 = \sum_{i=2}^t q_i = |W|$. □

By Case 3.3.2, there exists a subset $Z_i \subseteq E(M_i)$, such that for any edge $e_0 \in E(G[N_i])$ there exists an edge $e_1 \in Z_i$ which has r-vision of e_0 in $G[N_i]$, and $|Z_i| \leq \frac{4}{3}(|V(M_i)| - 1)$.
Let $Z = W \cup \left(\cup_{i=1}^t Z_i\right)$. An easy calculation gives that

$$|Z| \leq (t-1) + \sum_{i=1}^{t} \frac{4|V(M_i)| - 4}{3} \leq \frac{4|V(M)| - 4t + 3(t-1)}{3} \leq$$
$$\leq \frac{4(|M_H| + |M_V| - 1)}{3}.$$

Take an arbitrary edge $e_0 = \{v_0, h_0\} \in E(G)$. We have three cases.

1. If $e_0 \in F_{i,j}^V$ for some i, j, then we claim that $f_{i,j}^V \cap e_0 \neq \emptyset$. Suppose not; by Claim 3.16 there exists $f \in F_{i,j}^V$ which intersects both e_0 and $f_{i,j}^V$. For any edge $e \in F_{i,j}^V$ intersecting $f_{i,j}^V$ it either intersects f too, or there is an edge intersecting both e and f. Thus, f intersects at least as many edges as $f_{i,j}^V$, plus it intersects e_0 too, which contradicts the choice of $f_{i,j}^V$.

   If $w_i = f_{i,j}^V$, then w_i trivially has r-vision of e_0. If both $f_{i,j}^V$ and $f_{i,j}^H$ exist, we have two cases.

   - If $v_0 \in f_{i,j}^V$, then $v_0 \in w_i$ too, so w_i has r-vision of e_0.
   - If $h_0 \in f_{i,j}^V$, then Claim 3.17 yields that $\{v_0\} \cup (f_{i,j}^H \cap N_i) \in E(G)$. Thus, $\left\{\{v_0, h_0\}, f_{i,j}^V, w_i, \{v_0\} \cup (f_{i,j}^H \cap N_i)\right\}$ is the edge set of a C_4 in G, so w_i has r-vision of e_0.

2. If $e_0 \in F_{i,j}^H$ for some i, j, the same argument as above gives that $w_{i,j}$ has r-vision of e_0.
3. If neither of the previous two cases holds, then $e_0 \in E(G[N_i])$ for some i, so some element of Z_i has r-vision of it.

Thus, Z satisfies Theorem 3.1′, and the proof is complete.

Chapter 4

Algorithms and complexity

In this chapter, we develop algorithms based on the main theorems of the previous chapters. Also, the computational complexity of art gallery problems is discussed. It turns out that our algorithms are efficient, although a considerable amount of preparation and review of literature is necessary. Given their efficiency, our algorithms have the potential to be applicable in practice, too.

To achieve linear running times, we will rely on Chazelle's triangulation algorithm. Although the algorithm of Kirkpatrick, Klawe, and Tarjan [KKT92] runs only in $O(n \log \log n)$, it is not nearly as complex, which may be preferable in real world applications. Nonetheless, our algorithms run in linear time, given a triangulated input.

As data structures, we represent polygons by the (cyclically) ordered, doubly linked lists of their vertices.

Theorem 4.1 (Chazelle [Cha91])**.** *An n-vertex polygon can be triangulated in $O(n)$ time.*

A common subroutine in our art gallery algorithms is the construction of R-trees (Section 2.3).

Proposition 4.2 (Győri, Hoffmann, Kriegel, and Shermer [GHKS96, Section 5])**.** *The horizontal (vertical) R-tree of an orthogonal polygon P can be constructed in linear time.*

Proof. Algorithm B.3 produces the R-tree. The main part of this, Algorithm B.1, is an adaptation of the (in my opinion incomplete) algorithm of Fournier and Montuno [FM84, Algorithm 4b]. Suppose we are at the beginning of a loop (Step 18 in Algorithm B.2), and let D be the set of sides of P that are contained in triangles already deleted from T. For each diagonal d drawn by the triangulation, in $S[d]$ we store a y-coordinate decreasing list of (the whole of or a segment of) the vertical sides of P in D seen (via horizontal vision) by d.

Almost every step of the algorithm in the main loop (starting at Step 18) can be executed in $O(1)$. The following steps require further consideration.

- Step 27 requires $O(|S[s_v]|+|S[s_w]|)$ time. After this step, their elements are discarded.
- Steps 29-31 can be computed in $O(|R| + |S[s_w]|)$. The elements in R and $S[s_w]$ are discarded.
- A similar result holds for Steps 33-35.
- Lastly, the for-loop starting following Step 37 completes $|S[s_u]|$ cycles, after which the contents of $S[s_u]$ are discarded.

At any point in the algorithm, let W be the set of sides of $\cup T$. Observe, that

$$\bigcup_{d \in W} S[d]$$

contains every original side of the polygon at most twice. Every element is processed at most once by each of the previous highlighted steps, therefore Algorithm B.1 runs in $O(n)$.

The rest of Algorithm B.3 is straightforward. Step 15 runs in $O(1)$ as we store the orthogonal polygon pieces in doubly linked lists. Therefore Algorithm B.3 runs in linear time. □

4.1 Partitioning orthogonal polygons

Liou, Tan, and Lee [LTL89] proved that an n-vertex rectilinear domains can be partitioned into the minimum number of rectangles in $O(n)$ (assuming the preprocessing step uses linear time triangulation Theorem 4.1). An $O(n \log n)$ algorithm by Lopez and Mehta [LM96] produces an optimal partition by horizontal cuts of an n-vertex rectilinear domain into at most 6-vertex rectilinear domains. Both algorithms have important applications to computer graphics, image processing, and automated VLSI layout designs.

However, beyond these cases, not much is known about the complexity of determining a minimum size partition of a rectilinear domain into at most $2k$-vertex rectilinear domains, when $k \geq 4$. Instead of insisting on finding a minimum size partition, we present an efficient algorithm to find a partition whose cardinality is not greater than the extremal optimum.

Theorem 4.3. *An n-vertex orthogonal polygon can be partitioned into at most $\lfloor \frac{3n+4}{16} \rfloor$ orthogonal polygons of at most 8 vertices in linear time.*

Proof. The proof of Theorem 1.6 describes a recursive algorithm. Use Algorithm B.3 to construct the horizontal R-tree of P, such that the edge list of a vertex is ordered by the x coordinates of the corresponding cuts. Furthermore, compute the list of pockets, corridors, and special rectangles of Case 4.1. These structures can be maintained in $O(1)$ for the partitions after finding and performing a cut in $O(1)$. □

The art gallery problem corresponding to partitioning into at most 8-vertex pieces is the *MSC* problem (see Table 4.1). The *NP*-hardness of finding a minimum cardinality partition of an orthogonal polygon into at most 8-vertex pieces is an open problem, similarly to the *MSC* problem discussed in the next section.

4.2 Finding guard systems

Minimum cardinality Sliding Cameras problem (*MSC*)	
Input	An orthogonal polygon P
Feasible solution	A set M of vertical and horizontal mobile r-guards covering the domain enclosed by P
Objective	Minimize $\|M\|$
Decision version	Given (P,k), decide whether min $\|M\| \leq k$.

Minimum cardinality Horizontal Sliding Cameras problem (*MHSC*)	
Input	An orthogonal polygon P
Feasible solution	A set M_H of **horizontal** mobile r-guards covering the domain enclosed by P
Objective	Minimize $\|M_H\|$
Decision version	Given (P,k), decide whether min $\|M_H\| \leq k$.

Table 4.1 Definitions of the sliding camera problems (*MHSC* and *MSC*)

Dominating Set problem	
Input	A simple graph G
Feasible solution	A set $S \subseteq V(G)$ which satisfies $\forall v \in V(G) \setminus S \quad \exists u \in S \quad \text{such that} \quad \{u, v\} \in E(G)$
Objective	Minimize $\|S\|$
Decision version	Given (G, k), decide whether $\min \|S\| \leq k$.

Total Dominating Set problem	
Input	A simple graph G
Feasible solution	A set $S \subseteq V(G)$ which satisfies $\forall v \in V(G) \quad \exists u \in S \quad \text{such that} \quad \{u, v\} \in E(G)$
Objective	Minimize $\|S\|$
Decision version	Given (G, k), decide whether $\min \|S\| \leq k$.

Table 4.2 Definitions of the *Total Dominating Set* and *Dominating Set* problem

Finding a minimum cardinality horizontal mobile r-guard system, which is also known as the *Minimum cardinality Horizontal Sliding Cameras* or *MHSC* problem (Table 4.1), is known to be polynomial [KM11] in orthogonal polygons without holes. In orthogonal polygons with holes, the problem is *NP*-hard as shown by Biedl, Chan, Lee, Mehrabi, Montecchiani, and Vosoughpour [BCLMMV16]. In their paper, a polynomial time constant factor approximation algorithm for the *MHSC* problem is described, too. As explained in Section 3.2, the *MHSC* problem translates to the *Total Dominating Set* problem (Table 4.2) in the pixelation graph (Section 3.2), which can be solved in polynomial time for chordal bipartite graphs [DMK90].

Finding a minimum cardinality mixed vertical and horizontal mobile r-guard system (also known as the *Minimum cardinality Sliding Cameras* or *MSC* problem) has been shown by Durocher and Mehrabi [DM13] to be *NP*-hard for orthogonal polygons with holes. For orthogonal polygons without holes, the problem translates to the *Dominating Set* problem in the pixelation graph. This reduction in itself has little use, as Müller and Brandstädt [MB87] have shown that *Dominating Set* is *NP*-complete even in chordal bipartite graphs. To our knowledge, the complexity of *MSC* is still an open question. There is, however, a polynomial time 3-approximation algorithm by Katz and Morgenstern [KM11] for the *MSC* problem for orthogonal polygons without holes. In case holes are allowed, [BCLMMV16] give a polynomial time constant factor approximation algorithm.

The algorithm for the *MHSC* problem in [KM11] relies on a polynomial algorithm solving the *Clique Cover* problem in chordal graphs. Our analysis of the R-tree structures and the pixelation graph allows us to reduce the polynomial running time to linear.

Theorem 4.4 (Győri and Mezei [GM17])**.** *Algorithm B.4 finds a solution to the MHSC problem in linear time.*

Proof. By Proposition 4.2, both the horizontal R-tree T_H and the vertical R-tree T_V of D can be constructed in linear time.

The main idea of the algorithm is to only sparsely construct the pixelation graph G of D. Observe, that the neighborhood of a vertical slice in G is a path in T_H, and vice versa. Label each horizontal edge of D by the horizontal slice that contains it. Furthermore, label each vertical edge of each horizontal slice by the edge of D containing it; do this for the horizontal edges of vertical slices as well. This step also takes linear time. The endpoints of a path induced by the neighborhood of any node in G can be identified via these labels in $O(1)$ time.

In Section 3.2, we showed that a horizontal guard system is a subset of $V(T_H)$ which intersects (covers) each element of $\mathscr{F}_H = \{N_G(v) \mid v \in V(T_V)\}$. Dirac's theorem [Fra13, p. 10] states that ν, the maximum number of disjoint subtrees of the family, is equal to τ, the minimum number of nodes covering each subtree of the family. Obviously, $\nu \le \tau$. The other direction is proved using a greedy algorithm:

1. Choose an arbitrary node r of T_H to serve as its root. The distance of a vertical slice $v \in V(T_V)$ from r is $\mathrm{dist}_r(v) = \min_{h \in N_G(v)} \mathrm{dist}(h, r)$, and let $h_r(v) = \arg\min_{h \in N_G(v)} \mathrm{dist}(h, r)$.
2. Enumerate the elements of $V(T_V)$ in decreasing order of their distance from r, let $v_1, v_2, \dots, v_{|V(T_V)|}$ be such an indexing. Let $S_0 = \emptyset$.
3. If $N_G(v_i)$ is disjoint from the elements of $\{N_G(v) \mid v \in S_{i-1}\}$, let $S_i = S_{i-1} \cup \{v_i\}$; otherwise let $S_i = S_{i-1}$.

We claim that $\{h_r(v) \mid v \in S_{|V(T_V)|}\}$ is a cover of $\mathscr{F}_H$. Suppose there exists $v_j \in V(T_V)$ such that $N_G(v_j)$ is not covered. Let i be the smallest index such that $v_i \in S_i$ and $N_G(v_j) \cap N_G(v_i) \neq \emptyset$. Clearly, $i < j$, therefore $\text{dist}_r(v_i) \geq \text{dist}_r(v_j)$. However, this means that $h_r(v_i) \in N_G(v_j)$.

Now $\{h_r(v) \mid v \in S_{|V(T_V)|}\}$ is a cover of the same cardinality as the disjoint set system $\{N_G(v) \mid v \in S_{|V(T_V)|}\}$, proving that $\nu = \tau$.

Each neighborhood $N_G(v)$ for $v \in V(T_V)$ induces a path in T_H. Therefore, the first part of the algorithm, including calculating $\text{dist}_r(v)$ and $h_r(v)$ for each v, can be performed in $O(n)$ time, using the off-line lowest common ancestors algorithm of Gabow and Tarjan [GT85].

Calculating the distance decreasing order takes linear time via breadth-first search started from the root. In the i^{th} step of the third part of the algorithm, we maintain for each node in $V(T_H)$ whether it is under an element of $\{h_r(v) \mid v \in S_i\}$. Summed up for the $|V(T_H)|$ steps, this takes only linear time. $N_G(v_{i+1})$ is disjoint from the elements of $\{N_G(v) \mid v \in S_i\}$ if and only if one of the ends of the path induced by $N_G(v_{i+1})$ is under one of the elements of $\{h_r(v) \mid v \in S_i\}$, which now can be checked in constant time. Thus, the algorithm takes in total some constant factor times the size of the input time to run. □

The computational complexity of the *Point guard* problem (see Table 4.3) in orthogonal polygons with or without holes has attracted significant interest since the inception of the problem. Schuchardt and Hecker [SH95] showed that even for orthogonal polygons (without holes), *Point guard* is *NP*-hard. However, a minimum cardinality *Point r-guard* system of an orthogonal polygon can be computed in $\tilde{O}(n^{17})$ time [WK07]. To our knowledge, the exponent of the running time is still in the double digits, which makes its use impractical. Therefore, approximate solutions to the problem are still relevant. A linear-time 3-approximation algorithm is described in [LWŻ12].

Corollary 4.5. *An $\frac{8}{3}$-approximation of the minimum size of a point guard system of an orthogonal polygon can be computed in linear time.*

Proof. Compute m_V and m_H using the previous algorithm. By Theorem 3.1 and the trivial statement that both $m_H \leq p$ and $m_V \leq p$, we get that $\frac{4}{3} \cdot (m_H + m_V)$ is an $\frac{8}{3}$-approximation for p. □

Unfortunately, we can only compute the corresponding solution (guard system) in $O(n^2)$, because the pixelation graph may have $\Omega(n^2)$ edges. I consider it an interesting open problem to reduce this running time to linear as well, so that it matches the algorithm of [LWŻ12].

Point guard problem	
Input	An orthogonal polygon P (which bounds the domain D)
Feasible solution	A subset of points $X \subset D$ satisfying $\forall y \in D$ there exists $x \in X$ such that $\overline{xy} \subset D$.
Objective	Minimize $\lvert X \rvert$
Decision version	Given (P, k), decide whether $\min \lvert X \rvert \leq k$.

Point r-guard problem	
Input	An orthogonal polygon P (which bounds the domain D)
Feasible solution	A subset of points $X \subset D$ satisfying $\forall y \in D$ there exists $x \in X$ such that x has r-vision of y in D.
Objective	Minimize $\lvert X \rvert$
Decision version	Given (P, k), decide whether $\min \lvert X \rvert \leq k$.

Table 4.3 Definitions of the *Point (r-)guard* problem

Part II

Terminal-pairability (edge-disjoint path problem)

Chapter 5

The terminal-pairability problem

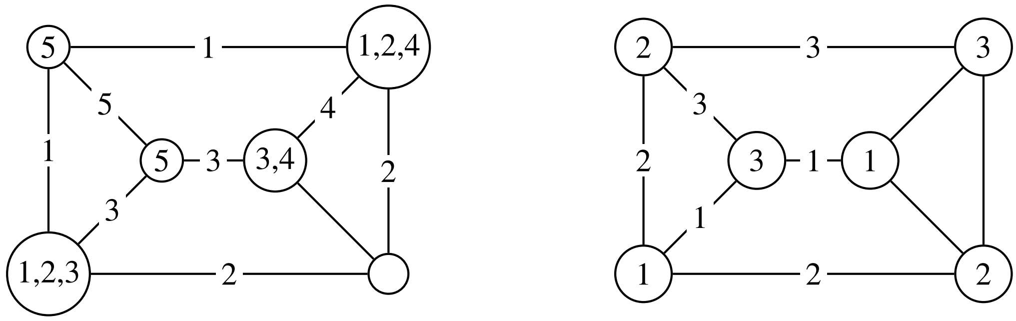
5.1 Problem statement and origins

We discuss the graph theoretic concept of **terminal-pairability** emerging from a practical networking problem introduced by Csaba, Faudree, Gyárfás, Lehel, and Schelp [CFGLS92], further studied by Faudree, Gyárfás, and Lehel [FGL92; Fau92; FGL99] and Kubicka, Kubicki, and Lehel [KKL99]. Given a simple undirected graph $G = (V(G), E(G))$ and an undirected multigraph $D = (V(D), E(D))$ on the same vertex set ($V(D) = V(G)$), we say that D can be **realized** in G iff there exist edge-disjoint paths $P_1, \ldots, P_{|E(D)|}$ in G such that P_i joins that endpoints of $e_i \in E(D)$ for any $i = 1, 2, \ldots, |E(D)|$. We call D and its edges the **demand graph** and the **demand edges** of G, respectively. Given G and a family $\mathscr{F}$ of (demand)graphs defined on $V(G)$ we call G **terminal-pairable** with respect to $\mathscr{F}$ if every demand graph in $\mathscr{F}$ can be realized in G.

In particular, let $|V(G)|$ be even and let $\mathscr{M}$ consist of all perfect matchings of the complete graph on $|V(G)|$ vertices; we call G a **path-pairable** graph if it is terminal-pairable with respect to $\mathscr{M}$. See Fig. 5.1 for an example of a terminal- and a path-pairability problem and their corresponding solutions.

The way the terminal-pairability problem was originally formulated considered an instance of the problem to consist of G (the graph of the internal vertices) and a prescribed number of pairwise distinct terminal vertices attached to each vertex of G. This graph extended with stars is called terminal-pairable if any matching of the terminal vertices has a realization. In our language, the family of demand graphs is of the form

$$\left\{ D \ : \ V(D) = V(G), \ \forall v \in V(G) \ d_D(v) \leq c(v) \right\},$$



(a) A terminal-pairability problem and its solution (b) A path-pairability problem and its solution

Fig. 5.1 Examples for terminal-pairability and path-pairability problems. Each vertex is labeled with the demand edges to which it is incident to.

where $c : V(G) \to \mathbb{N}$ describes the number of terminals incident to a vertex of G. We call the process of substituting the demand edges by disjoint paths in G a **realization** of the demand graph.

Given a simple graph G, one central question in the topic of terminal-pairability is the maximum value of q for which any demand graph in the set

$$\left\{D \ : \ V(D) = V(G), \ \forall v \in V(G) \ d_D(v) \leq q\right\}$$

is realizable in G. As at a given vertex $v \in V(G)$ at most $d_G(v)$ edge-disjoint paths can start, the minimum degree $\delta(G)$ of the base graph provides an obvious upper bound on q. Often, a better upper bound on q is obtained by choosing a q-regular multigraph D, such that except e_0 elements, edges of D can only be resolved into paths of length at least ℓ in G. By the pigeonhole principle, we must have

$$\left(\frac{1}{2} \cdot q \cdot |V(G)| - e_0\right) \cdot \ell \leq e(G) - e_0,$$

or equivalently,

$$q \leq \frac{\overline{d}(G)}{\ell} + \frac{2e_0\,(\ell - 1)}{|V(G)|}. \tag{5.1}$$

An improvement on this method is described by Girão and Mészáros [GiMé17] for $G = K_n$.

5.2 Outline of Part II

In Chapter 6, the case $G = K_n$ is studied in detail, where the best known lower bound on q is determined (Theorem 6.4). Using the techniques of this proof, the exact extremal edge number of a demand graph realizable in K_n is obtained (Theorem 6.7).

On the other end of the spectrum, a relatively sparse graph which is still path-pairable is sought after in Chapter 7. At the end of the chapter, some open problems are discussed. The results of Chapter 6 and 7 are a result of a fruitful collaboration with my supervisor, Ervin Győri, and Gábor Mészáros.

Lastly, in Chapter 8, subjects related to terminal-pairability and the problem's complexity are surveyed.

Chapter 6

Complete base graphs

Csaba, Faudree, Gyárfás, Lehel, and Schelp [CFGLS92] studied the extremal value of the maximum degree of the demand graph for the complete graph K_n as the base graph and investigated the following question:

Problem 6.1 ([CFGLS92])**.** *What is the highest number q for which any demand graph on n vertices and maximum degree q is realizable in K_n?*

One can easily verify that the parameter q cannot exceed $\frac{n}{2}$. Indeed, take a demand graph D obtained by replacing every edge in a one-factor on n vertices by q parallel edges. In order to create edge-disjoint paths, most paths need to use at least two edges in K_n, thus inequality (5.1) implies the indicated upper bound.

The authors of [CFGLS92] conjectured that if $n \equiv 2 \pmod 4$, then $q = \frac{n}{2}$. However, Girão and Mészáros showed, that asymptotically, q/n is less than $\frac{1}{2}$.

Proposition 6.2 (Girão and Mészáros [GiMé17])**.** *If every n-vertex demand graph D with $\Delta(D) \leq q$ is realizable in K_n, then $q \leq \frac{13}{27}n + O(1)$.*

Csaba, Faudree, Gyárfás, Lehel, and Shelp showed the following lower bound.

Theorem 6.3 (Csaba, Faudree, Gyárfás, Lehel, and Schelp [CFGLS92])**.** *Any demand graph D on n-vertices with $\Delta(D) \leq \frac{n}{4+2\sqrt{3}}$ is realizable in K_n.*

We improve their result by proving the following theorem:

Theorem 6.4 (Győri, Mezei, and Mészáros [GMM16])**.** *Any demand graph D on n-vertices with $\Delta(D) \leq 2\lfloor \frac{n}{6} \rfloor - 4$ is realizable in K_n.*

Kubicka, Kubicki, and Lehel [KKL99] investigated terminal-pairability properties of the Cartesian product of complete graphs. In their paper, the following "Clique-Lemma" was proved and frequently used:

Lemma 6.5 (Kubicka, Kubicki, and Lehel [KKL99]). *Let D be an n-vertex demand graph, where $n \geq 5$. If $\Delta(D) \leq n-1$ and $e(D) = n$, then D is realizable in K_n.*

In the same paper, the following related problem was raised about the possible strengthening of Lemma 6.5:

Problem 6.6 ([KKL99]). *Let D be an n-vertex demand graph such that $\Delta(D) \leq n-1$. What is the largest value of α such that $e(D) \leq \alpha \cdot n$ implies that D is realizable in K_n?*

Obviously, $1 \leq \alpha$ due to Lemma 6.5. It is also easy to see that $\alpha < 2$. Let D be a demand graph on $n \geq 4$ vertices, in which two pairs of vertices, u, v and x, y are both joined by $(n-2)$ parallel edges ($d_D(w) = 0$ for $w \notin \{x, y, u, v\}$). Observe that to realize the demand graph, any disjoint path system must contain a path from x to y passing through u or v. However, there are also $n-2$ disjoint paths connecting u and v, meaning that u or v is incident to at least $2+(n-2) = n$ disjoint edges, which is clearly a contradiction. This implies that the number of edges in D cannot exceed $2n-5$. We show that this bound is sharp by proving the following theorem:

Theorem 6.7 (Győri, Mezei, and Mészáros [GMM16]). *Let D be a demand graph on n vertices with at most $2n-5$ edges, such that no vertex is incident to more than $n-1$ edges. Then D has a realization in K_n.*

Before the proofs, we fix further notation and terminology. For convenience, we call a pair of edges joining the same two vertices a C_2. For $k > 2$, C_k denotes the cycle on k vertices. For a subset $S \subset V(G)$ of vertices let $e(S, V(G) - S)$ denote the number of edges with exactly one endpoint in S. Let $G[S]$ denote the subgraph induced in G by the subset of vertices S. We call a pair of vertices joined by k parallel edges a ***k*-bundle**.

For a vertex v we denote the set of neighbors by $N(v)$ and use $\gamma(v) = |N(v)|$. We define the **multiplicity** $m(v)$ of a vertex v as follows: $m(v) = d(v) - \gamma(v)$. Observe that $m(v)$ is the minimal number of edges incident to v that need to be replaced by longer paths in a realization to guarantee an edge-disjoint path-system for the terminals of v.

We define an operation that is repeatedly used in our proofs: given an edge $uv \in E$, we say that we **lift** uv to a vertex w when the edge uv is substituted by uw and wv (forming a path of length 2). Note that this operation increases the degree of w by 2, but does not affect the degree of any other vertex (including u and v). Also, as a by-product of the operation, if w is

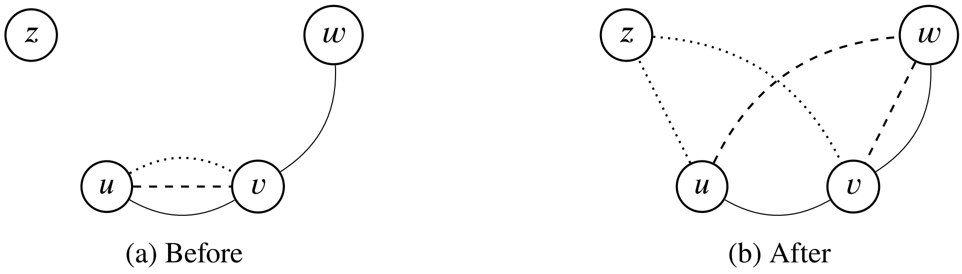


Fig. 6.1 Lifting 2 edges of uv to z and w

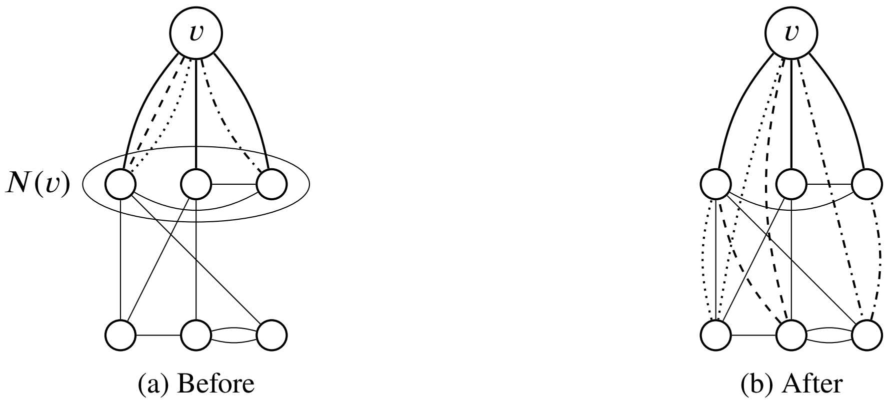


Fig. 6.2 Resolving the multiplicities at v

already joined by an edge to u or v, the multiplicity of the appropriate pairs increase by one (see Fig. 6.1).

Finally, note that if a graph G has n vertices and $d(v) \leq n-1$, all multiplicities of v can be easily resolved by subsequent liftings. Indeed, v has $n-1-\gamma(v)$ non-neighbors and $m(v) = d(v)-\gamma(v) \leq n-1-\gamma(v)$ multiplicities, thus we can assign every edge of v causing a multiplicity to a non-neighbor to which that particular edge can be lifted without creating new multiplicities at v. As a result, $d(v)$ does not change but $\gamma(v)$ becomes equal to $d(v)$. We call this the **resolution of the multiplicities** of v (see Fig. 6.2).

6.1 Proof of Theorem 6.4

We show that if $D = (V, E)$ is a demand multigraph on n vertices and $\Delta(G) \leq 2\lfloor \frac{n}{6} \rfloor - 4$, then D can be transformed into a simple graph by replacing parallel edges by paths of D. We prove the statement by induction on n. Observe first that the statement is obvious for $n < 18$. For $18 \leq n < 24$, note that the demand graph D is the disjoint union of 2-bundles, circles, paths, and isolated vertices (i.e., a 2-matching). It is easy to realize 2-matchings in K_n; the verification of the statement is left to the reader.

From now on assume $n \geq 24$. We may assume without loss of generality that D is an $\left(2\lfloor\frac{n}{6}\rfloor - 4\right)$-regular multigraph; if necessary, additional parallel edges may be added to D. Should a single vertex v fail to meet the degree requirement, we bump up its degree by further lifting operations as follows: as the deficit $\left(2\lfloor\frac{n}{6}\rfloor - 4\right) - d(v)$ must be even, we can lift an arbitrary edge $e \in E([V(D) - v])$ to v. We remind the reader that lifting e to v increases $d(v)$ by two while it does not affect the degree of the rest of the vertices.

We will use the well-known 2-Factor-Theorem of Petersen [Pet91]. Be aware that a 2-factor of a multigraph may contain several C_2's (however, this is the only way parallel edges may appear in it).

Theorem 6.8 (Petersen [Pet91])**.** *Let G be a $2k$-regular multigraph. Then $E(G)$ can be decomposed into the union of k edge-disjoint 2-factors.*

Some operations, which are performed later in the proof, are featured in the following definition, claim, and lemma.

Definition 6.9 (Lifting coloring)**.** Let F be a multigraph, and $c : E(F) \cup V(F) \to \{1, 2, 3\}$ be a coloring of the edges and vertices of F. We call c a *lifting coloring* of F if and only if

1. for any edge $e = uv \in E(F)$, $c(u) \neq c(e)$ and $c(v) \neq c(e)$, and
2. for any two edges $e_1, e_2 \in E(F)$ incident to a common vertex we have $c(e_1) \neq c(e_2)$.

Moreover, if the number of vertices in different color classes differ by either 0, 1, or 2, then we call c a *balanced lifting coloring* of F.

Claim 6.10. *Let F be a multigraph such that $\forall v \in V(F)$ we have $d_F(v) \leq 2$. If $w_1, w_2, w_3 \in V(F)$ are three pairwise non-adjacent different vertices, then F has a balanced lifting coloring where w_i gets color i.*

Proof. The proof is easy but its complete presentation requires a rather lengthy (but straightforward) casework. We leave the verification of the statement to the reader. Fig. 6.3 shows an example output of this lemma. □

Lemma 6.11. *Let D be a demand graph on n vertices, such that $\Delta(D) \leq \lfloor\frac{n}{3}\rfloor - 4$. Furthermore, let $X = \{x_1, x_2, x_3\}$ be a subset of $V(D)$ of cardinality 3, such that $|E(D[X])| = 0$. Let B be an at most 3-element subset of $V(D) \setminus X$. Let F be a 2-matching of D, i.e., there are at most*

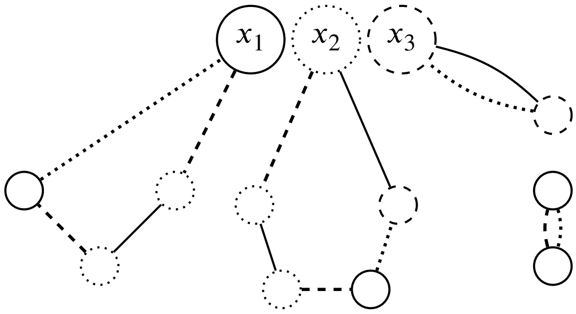


Fig. 6.3 A balanced lifting coloring, where x_1, x_2, x_3 get pairwise different colors.

2 edges of F incident to any vertex of D. Moreover, either $d_F(x_i) = 2$ or $d_D(x_i) \le \lfloor \frac{n}{3} \rfloor - 5$ for $i = 1, 2, 3$. Then there exists a demand graph H which satisfies

- $V(H) = V(D) \setminus X$,
- $E(H) \supset E(D[V(H)]) \setminus F$,
- $\{e \in E(H) :\ e$ *is incident to at least one of* $B\} \subset E(D)$, *and*
- *for any* $v \in V(H)$ *we have* $d_H(v) \le d_D(v) - d_F(v) + \mathbb{1}(v \notin B)$.

Moreover, if H has a realization in K_{n-3}, then D is realizable in K_n.

Proof. We will perform a series of liftings in D in two phases, obtaining D' and D''. At the end of the second phase, we will achieve that there are no incident parallel edges to X in D''. Therefore, setting $H = D'' - X$ will satisfy the second claim of the lemma.

First, we determine the series of liftings to be executed in the first phase. Notice that Claim 6.10 implies the existence of a balanced lifting coloring c of F such that $c(x_i) \equiv i + 1 \pmod 3$. Lift each edge $f \in F$ to $x_{c(f)}$, except if f is incident to $x_{c(f)}$, then leave f where it is. Let F' be the set of lifted edges, that is

$$F' = \biguplus_{\substack{f \in F,\\ x_{c(f)} \notin f}} \left\{ \text{the two edges joining } x_{c(f)} \text{ to the two vertices of } f \right\},$$

where $\biguplus$ denotes the disjoint union. Let the multigraph D' be defined on the same vertex set as D, and let its edge set be

$$E(D') = \{e \in E(D) :\ e \notin F \text{ or } x_{c(e)} \in e\} \biguplus F'.$$

In other words, D' is the demand graph into which D is transformed by lifting the elements of F. Let $Y = V(D) \setminus X$. Observe that $d_{D'}(y) = d_D(y)$ for $y \in Y$. Let

$$Y_i = \{y \in Y \setminus B \mid c(y) = i\}$$

be the color i vertices in $Y \setminus B$. The balancedness of c guarantees that

$$|Y_i| = |c^{-1}(i) \setminus X \setminus B| \geq |c^{-1}(i)| - 1 - |B| \geq \left\lfloor \frac{n}{3} \right\rfloor - 5.$$

In the second phase, our task is to resolve all multiplicities of x_i in D'. Observe that as edges of F of the same color formed a matching, out of every two parallel edges that are incident to x_i in D', at least one is an initial edge in $E(D') \setminus F'$. The vertex x_i is incident to $d_{D'}(x_i) - d_{F'}(x_i)$ edges of $E(D') \setminus F'$; we plan to lift these edges to elements of Y_i by using every vertex in Y_i for lifting at most once. If $d_F(x_i) = 2$, then one of the two edges of F incident to x has color $i - 1$, and this edge is lifted to x_{i-1} in D'. Thus

$$d_{D'}(x_i) - d_{F'}(x_i) \leq \left\{ \begin{array}{ll} d_D(x_i) - 1, & \text{if } d_F(x_i) = 2; \\ \left\lfloor \frac{n}{3} \right\rfloor - 5, & \text{if } d_F(x_i) < 2; \end{array} \right\} \leq |Y_i|.$$

As elements of $c^{-1}(i)$ are not incident to edges of color i, the set $Y_i \setminus B$ offers enough space to carry out the liftings. That being said, note that neighbors of x_i in Y_i cannot be used for lifting as they would create additional multiplicities. On the other hand, if $v \in Y_i$ and $e = vx_i \in E(D)$, then e is an initial edge of x_i that either generates no multiplicity at all or is part of a bundle of parallel edges, one of which we do not lift. In other words, for every vertex of Y_i that is excluded from the lifting we mark an initial edge of x_i that we do not need to lift. Because of this, resolution of the remaining multiplicities at x_i can be performed in $Y_i - N(x_i)$. Let D'' denote the demand graph obtained after resolving every multiplicity of x_1, x_2, and x_3.

At most 1 element of $E(D') \setminus E(F')$ has been lifted to each $y \in Y$, therefore there are no multiple edges between the sets X and Y in the demand graph D''. Moreover, $D''[X] = D'[X]$ is a subgraph of a triangle, which emerges as we lift the at most one edge of color $i + 2$ of x_i to x_{i+1} (take the indices cyclically), for $i = 1, 2, 3$.

Any vertex $y \in Y$ of color i has at most two incident edges in F', joining y to a subset of $\{x_{i+1}, x_{i+2}\}$.

- If an edge has been lifted to $y \in Y$ of color i, then y is adjacent to x_i and $d_{D''}(y) = d_{D'}(y) + 2$. Thus, y is joined to at least $d_{F'}(y) + 1$ elements of X in D''. As no edge of color i can be incident to y, we have $d_{F'}(y) = d_F(y)$. Therefore

$$d_{D''[Y]}(y) \leq d_{D''}(y) - d_{F'}(y) - 1 = d_{D'}(y) - d_F(y) + 1 = d_D(y) - d_F(y) + 1.$$

- If no edges have been lifted to $y \in Y$, then $d_{D''}(y) = d_{D'}(y)$ and y is adjacent to at least $d_F(y)$ elements of X in D''. Therefore

$$d_{D''[Y]}(y) = d_{D''}(y) - d_F(y) \leq d_{D'}(y) - d_F(y) = d_D(y) - d_F(y).$$

As elements of B are excluded from Y_i, 0 edges are lifted to them, and so we proved the statement of the lemma. □

Let $X_1 = \{x_1, x_2, x_3\}$ be a subset of 3 elements of $V(D)$, such that $D[X_1]$ has 0 edges. Such a set trivially exists, as any two non-adjacent vertices have $(n-2) - 2\Delta(D) \geq \frac{n}{3} + 2$ common non-neighbors. Since the degree in D is at least $2 \cdot (24/6) - 4 = 4$, Theorem 6.8 implies the existence of two disjoint 2-factors, A_1 and A_2 of D. Notice that $A_2 - X_1$ has 3 path components (as a special case, an isolated vertex is a path on one vertex). Extend $A_2 - X_1$ to a maximal 2-matching F_2 of $D - X_1 - A_1$. It is easy to see that there exists a 3-element subset B_1 of $V(D) \setminus X_1$ such that

- B_1 induces 0 edges in $D - A_1$,
- $\{v \in V(D) \setminus X_1 \,:\, d_{F_2}(v) = 0\} \subset B_1$, and
- $B_2 = \{v \in V(D) \setminus X_1 \,:\, d_{F_2}(v) = 1\} \setminus B_1$ has cardinality at most 3.

We are ready to use Lemma 6.11. First, apply it to D, where we lift $F = A_1$ to elements of $X = X_1$, while not creating new edges incident to $B = B_1$. Let the obtained graph be H_1. We have $\Delta(H_1) \leq \Delta(D) - \delta(A_1) + 1 = \Delta(D) - 1$. Furthermore, $E(H_1[B_1]) \subseteq E(D[B_1]) = \emptyset$, and for all $v \in B_1$ we have $d_{H_1}(v) \leq \Delta(D) - \delta(A_1) \leq \Delta(D) - 2$.

We apply Lemma 6.11 once again. Now H_1 is our base demand graph, F_2 is the 2-matching to be lifted to elements of B_1, and we avoid lifting to elements of B_2. Let the resulting demand graph be H_2, whose vertex set is $V(D) \setminus X_1 \setminus B_1$ of cardinality $n-6$. We have

$$d_{H_2}(v) \le \left\{ \begin{array}{ll} d_{H_1}(v) - d_{F_2}(v) + 1 & \text{if } v \notin B_2, \\ d_{H_1}(v) - d_{F_2}(v) & \text{if } v \in B_2. \end{array} \right\} \le$$
$$\le \left\{ \begin{array}{ll} (\Delta(D) - 1) - 2 + 1 & \text{if } v \notin B_2, \\ (\Delta(D) - 1) - 1 & \text{if } v \in B_2. \end{array} \right\} \le \Delta(D) - 2 = 2\left\lfloor \frac{n-6}{6} \right\rfloor - 4.$$

By induction on n, we know that H_2 is realizable in K_{n-6}, implying that H_1 is realizable in K_{n-3}, which in turn implies that D has a realization in K_n.

6.2 Proof of Theorem 6.7

We prove our statement by induction on n. For $n \le 4$ the statement is straightforward, the cases $n = 5, 6$ require a somewhat cumbersome casework. Note that if $n \ge 4$, we may assume that D has exactly $2n-5$ edges, otherwise we join two non-neighbors whose degree is smaller than $n-1$.

For the inductive step, we choose a vertex x, resolve each of its multiplicities, and delete it from the demand graph. There are two additional conditions to assert as the number of vertices decreases from n to $n-1$:

i) The number of edges of D must decrease by at least two.

ii) Every vertex of degree $n-1$ must lose at least one edge. Decreasing the degree $d(v)$ of a vertex v can be achieved by lifting an edge incident to v to x. Note that this operation might create additional multiplicities that need to be resolved before the deletion of x.

In addition, observe that we can lift at least one edge to a vertex v without its degree exceeding the degree bound for $n' = n-1$ if and only if $d(v) < n-2$. Let

$$B = \{z_1, \dots, z_{|B|}\} = \{v \in V(D) : d(v) \ge n-2\}.$$

As $\sum_{v \in V(D)} d(v) = 4n - 10$, it follows that $|B| \le 3$. We perform a casework on $|B|$.

$|B| = 0$: If B is empty, then the only condition we need to guarantee is the deletion of at least two edges in D. We have two cases.

- *If* $\forall x \in V(D)$ *we have* $\gamma(x) \leq 1$*,* then D is the disjoint union of bundles and isolated vertices. Let $\{x, y\}$ be the edge with the highest multiplicity. If $d(x) = d(y) \leq n-2$, and every other degree is at most $n-4$, then lift copies of $\{x, y\}$ to the other $n-2$ vertices and delete both x and y; we can use induction on the remaining graph. If D is composed of an $n-2$ and an $n-3$ bundle, it is easy to find a realization of D in K_n directly.
- *If there is an* $x \in V(D)$ *with* $\gamma(x) \geq 2$*:* we have $n-1-\gamma(x)$ vertices as a lifting target to resolve the $d(x)-\gamma(x)$ multiplicities of x. Obviously, $d(x)-\gamma(x) \leq n-3-\gamma(x)$, thus we have enough space to resolve all multiplicities of x. After the deletion of x, the graph has $\gamma(x) \geq 2$ fewer edges, and the maximum degree is still two less than the number of vertices.

$|B| = 1$: We perform the same operation as in the previous case with the choice $x = z_1$. Observe that our inequality becomes $d(z_1)-\gamma(z_1) \leq n-1-\gamma(z_1)$, thus we have enough vertices in the multigraph to perform all the necessary liftings.

$|B| = 2$: Observe first that z_1 and z_2 are joined by an edge e or else

$$2n-5 \geq e(B, V(D)-B) = d(z_1) + d(z_2) \geq 2n-4,$$

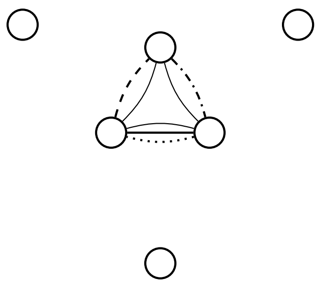
a contradiction. Let us first assume that z_1 or z_2 (say, z_1) has more than one neighbor (i.e., $e(B, V(D)-B) > 0$). Observe that in this case

$$m(z_1) = d(z_1) - \gamma(z_1) \leq (n-1) - \gamma(z_1),$$

thus, each multiplicity of z_1 can be resolved by lifting the appropriate edges to $V(D) - \{z_1\} - N(z_1)$.

In the remaining case z_1 and z_2 form a bundle of at most $n-1$ edges. We can lift $n-2$ of these edges to $V(D)-B$ without difficulties, delete one of the vertices in B, and proceed by induction.

$|B| = 3$**:** Observe that any two vertices of $\{z_1, z_2, z_3\}$ must be joined by an edge, else the same reasoning as above leads to a contradiction. Note also, that a simple average degree calculation guarantees the existence of an isolated vertex x. We distinguish two cases:


(a) Demand graph

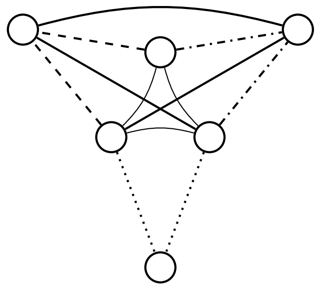

(b) A solution

Fig. 6.4 A demand graph and a possible realization in K_6

i) If $e(B, V(D) - B) = 0$, we may assume that $V(D) - B$ induces an edge, otherwise $3(n-3) \geq 4n - 10 \Rightarrow n \leq 7$ and all edges are contained in B. For $n = 5, 6, 7$, that leads to 4 possible demand graphs whose realization can easily be completed; a case for $n = 6$ is shown in Fig. 6.4.

   Let f denote an arbitrary edge in $V(D) - B$. We lift two non-parallel edges of B as well as f to x; observe that the degrees of all vertices in B dropped by at least 1. As $n \geq 7$, the multiple edge created at vertex x can be lifted to a vertex of $V(D) - B$ that was not incident to f.

ii) If $e(B, V(D) - B) > 0$, let f be an edge between B and $V(D) - B$. Without loss of generality, we may assume f is incident to z_3. We lift f as well as an edge e between z_1 and z_2 to x; as e and f are disjoint, no new multiplicity is created, thus we can delete x and proceed by induction.

Chapter 7

Complete grid base graphs

7.1 Introduction

A long-standing open question concerning path-pairability of graphs is the minimal possible value of the maximum degree $\Delta(G)$ of a path-pairable graph G. Faudree, Gyárfás, and Lehel [FGL99] proved that the maximum degree must grow together with the number of vertices in path-pairable graphs. They in fact showed that a path-pairable simple graph with maximum degree Δ has at most $2\Delta^{\Delta}$ vertices. The result yields that for a path-pairable simple graph G on n vertices we have

$$c \cdot \frac{\log n}{\log \log n} \leq \Delta(G).$$

This bound is conjectured to be asymptotically sharp, although to date only constructions of much higher order of magnitude have been found. Until recently, the best-known construction was due to Kubicka, Kubicki, and Lehel [KKL99] who showed that two dimensional complete grids on an even number of vertices (of at least 6) are path-pairable.

A two-dimensional complete grid is the **Cartesian product** $K_s \square K_t$ of two complete graphs K_s and K_t and it can be constructed by taking the Cartesian product of the sets $\{1, 2, \ldots s\}$ and $\{1, 2, \ldots t\}$ and joining two distinct vertices if they share a coordinate. Higher dimensional complete grids can be defined similarly: let $d, t_1, \ldots, t_d$ be positive integers and let V denote the set of d-dimensional vectors of positive integer coordinates not exceeding t_i in the i^{th} coordinate, that is,

$$V_{(t_1,\ldots,t_d)} = \{(a_1, \ldots, a_d) : 1 \leq a_i \leq t_i, i = 1, 2, \ldots, d\}.$$

The d-dimensional grid graph $K_{(t_1,\dots,t_d)}$ is constructed by taking $V_{(t_1,\dots,t_d)}$ as its vertex set, and two vertices are joined by an edge if the corresponding vectors differ at exactly one coordinate. Note that this graph is isomorphic to the Cartesian product $K_{t_1}\Box K_{t_2}\Box\dots\Box K_{t_d}$. For $t_1 = t_2 = \dots = t_d = t$ we use the notation

$$K_t^d = \overbrace{K_t\Box\dots\Box K_t}^{d}.$$

For a more detailed introduction to the Cartesian product of graphs the reader is referred to [IK00].

With $s = t$ the construction of Kubicka, Kubicki, and Lehel gives examples of path-pairable graphs on $n = s \cdot t$ vertices with maximum degree $2 \cdot \sqrt{n}$. This bound was recently improved to $\sqrt{n}$ by Mészáros [Més16]. It was also conjectured in [KKL99] that $K_t\Box K_t\Box K_t$ is path-pairable for sufficiently large even values of t.

In this chapter, we significantly improve the upper bound on the minimal value of the maximum degree of path-pairable graphs: we prove that high dimensional complete grids are path-pairable. We eventually study the more general terminal-pairability variant of the above path-pairability problem and prove the following theorem:

Theorem 7.1 (Győri, Mezei, and Mészáros [GMM17])**.** *Let $G = K_t^d$ and let $D = (V(D), E(D))$ be a demand graph with $V(D) = V(K_t^d)$ and $\Delta(D) \le 2\lfloor \frac{t}{12} \rfloor - 2$. Then D can be realized in G.*

Theorem 7.1 immediately implies the following corollary:

Corollary 7.2. *If $t \ge 24$, K_t^d is path-pairable.*

The above construction provides examples of path-pairable graphs on $n = t^d$ vertices with maximum degree

$$\Delta(K_t^d) = (t-1)\cdot d = \log n \cdot \frac{t-1}{\log t}.$$

Observe that t can be chosen to be a constant ($t = 24$) thus we have obtained path-pairable graphs on n vertices with $\Delta \approx 7.24 \log n$.

7.2 Proof of Theorem 7.1

For $i = 1,\dots,t$, let L_i be the subgraph of K_t^d induced by

$$\left\{(a_1,\dots,a_{d-1},i) : 1 \le a_j \le t_j, j = 1,2,\dots,d-1\right\}.$$

We call $L_1, \dots, L_t$ the **layers** of K_t^d. Similarly, by fixing the first $d-1$ coordinates we get t^{d-1} copies of K_t; we denote these complete subgraphs by $l_1, \dots, l_{t^{d-1}}$ and refer to them as **columns**.

Given an edge uv of the demand graph with $u, v \in K_t^d$ we replace uv by a path of three edges uu', $u'v'$, and $v'v$, where $u', v' \in L_i$ for some $i \in \{1, 2, \dots, t\}$ and u, u' and v, v' lie in the same columns. Having done that, we consider the new demand edges defined within the t layers and t^{d-1} columns and break the initial problem into $t^{d-1} + t$ subproblems that we solve inductively. The upcoming paragraphs discuss the details of the drafted solution.

For the discussion of the base case $d = 1$ as well as for the inductive step we use Theorem 6.4. We mention that instead of using this theorem we could use a weaker version of it with $\Delta(D) \leq \frac{t}{4+2\sqrt{3}}$ proved by Csaba, Faudree, Gyárfás, Lehel, and Schelp [CFGLS92]. With every further step of our proof unchanged, a result corresponding to the weaker theorem can be proved with a smaller bound on $\Delta(D)$.

Let q be an even number with $2 \leq q \leq \lfloor \frac{t}{6} \rfloor - 1$ and let $D = (V(D), E(D))$ be a demand multigraph with $V(D) = K_t^d$ and $\Delta(D) \leq q$. Let $E'(D)$ denote the set of demand edges whose vertices lie in different l_i, l_j columns. We construct an auxiliary graph H with $V(H) = V(K_t^{d-1})$ and project every edge of $E'(D)$ into H by deleting the last coordinates of the vertices. It is easy to see that $\Delta(H) \leq t \cdot q$.

We may assume without loss of generality that D is $(t \cdot q)$-regular by joining additional pairs of vertices or replacing edges by paths of length two if necessary. Again, we use the 2-factor decomposition theorem of Petersen (Theorem 6.8) to distribute the original demand edges among the layers $L_1, \dots, L_t$ and define new subproblems on them.

Obviously, the graph H satisfies the conditions of Theorem 6.8, thus $E(H)$ can be partitioned into $\frac{q}{2} \cdot t$ edge-disjoint 2-factors. By arbitrarily grouping the above 2-factors into $\frac{q}{2}$-tuples, we can partition $E(H)$ into t edge-disjoint subgraphs $H_1, \dots, H_t$ with $\Delta(H_i) \leq q$.

Assume now that the vertices $u = (\underline{a}, i)$ and $v = (\underline{b}, j)$ ($a, b \in [t]^{d-1}$) are joined by a demand edge belonging to $E'(D)$ (thus $\underline{a} \neq \underline{b}$) and the corresponding edge in H is contained by H_k. We then replace the demand edge uv by the following triple of newly established demand edges:

$$\left\{(\underline{a}, i), (\underline{a}, k)\right\}, \left\{(\underline{a}, k), (\underline{b}, k)\right\}, \left\{(\underline{b}, k), (\underline{b}, j)\right\}.$$

We claim the following:

(i) For every layer L_j the condition $\Delta(L_j) \leq q$ holds.

(ii) For every column l_j the condition $\Delta(l_j) \leq 2q$ holds.

The first statement obviously follows from the partition of $E'(D)$. For the second one, observe that a vertex v in l_j is initially incident to q demand edges and at most q additional demand edges are joined to it (otherwise *(i)* is violated). Notice now that every layer L_j contains a $(d-1)$-dimensional subproblem that can be solved (within the layer) by the inductive hypothesis. Also, every column l_j contains a subproblem (note that the original demand edges in $E(D) \setminus E'(D)$ are incorporated into these subproblems) that can be solved by Theorem 6.4. This completes our proof.

7.3 Remarks

By using Theorem 6.4 and the described inductive approach we proved that K_t^d is path-pairable for $t \geq 24$, $d \in \mathbb{Z}^+$. Even if the bound in Theorem 6.4 is improved to the point of being sharp, it only improves the constant 5.2 in Theorem 7.1 and decreases the lower bound on t in Corollary 7.2, however, it does not affect the $\Omega(\log n)$ order of magnitude of the maximum degree.

Let us measure the sharpness of our theorem. Assume $d \geq 2$ and that t is even. Take the following matching M of the vertices of $V(K_t^d)$: pair vertex $(x_1, \dots, x_d)$ with $(t+1-x_1, t+1-x_2, \dots, t+1-x_d)$. Let $D = (V(K_t^d), q \cdot M)$, i.e., every edge in M is taken with multiplicity q. Since the vertices of each edge are different in each coordinate, inequality (5.1) becomes

$$\frac{1}{2} \cdot t^d \cdot q \cdot d \leq \frac{1}{2} \cdot d \cdot (t-1) \cdot t^d,$$

implying that $q \leq (t-1)$. This means that there is at most a factor of $(6+\varepsilon_t)$ between the extremal bound and the result of Theorem 7.1.

We mention that one particularly interesting and promising path-pairable candidate (with the same order of magnitude of vertices but with a better constant for Δ) is the d-dimensional hypercube Q_d on 2^d vertices ($\Delta(Q_d) = d$). Observe that hypercubes are special members of the above studied complete grid family as $Q_d = K_2^d$. Although it is known that Q_d is not path-pairable for even values of d (see [Fau92]), the question is open for odd dimensional hypercubes for $d \geq 5$ (Q_1 and Q_3 are both path-pairable).

Conjecture 7.3 ([CFGLS92])**.** *The $(2k+1)$-dimensional hypercube Q_{2k+1} is path-pairable for all $k \in \mathbb{N}$.*

Chapter 8

Related subjects, algorithms and complexity

8.1 Immersions

Recently, the study of graph immersions has become increasingly popular. As the following definition shows, it is very closely related to the concepts of terminal-pairability.

Definition 8.1. Let H and G be (multi)graphs. There is an immersion of H in G (or H is immersed in G, or G contains an immersion of H) iff there is map $\phi : V(H) \cup E(H) \to V(G) \cup E(G)$ such that

- ϕ maps vertices of H into distinct vertices of G,
- a loop on a vertex u is mapped to a cycle of G which traverses $\phi(u)$,
- an edge uv of H is mapped to a path connecting $\phi(u)$ to $\phi(v)$ in G, and
- for two distinct edges e_1 and e_2 in H, their images $\phi(e_1)$ and $\phi(e_2)$ are edge-disjoint.

This relation is denoted as $H \leq_{\mathrm{i}} G$. If in addition, for any vertex $v \in V(H)$ and edge $e \in E(H)$ we have $v \notin e \implies \phi(v) \notin V(\phi(e))$, then H has a strong immersion in G, denoted $H \leq_{\mathrm{si}} G$.

Remark 8.2. In the study of graph immersions, the lifting operation is defined as the inverse of our lifting operation.

In the terminal-pairability problem, the function mapping vertices of H to vertices of G is fixed. However, if $G = K_n$ and H is loopless, the notion of realizable and immersible coincide.

The transitive property of the immersion relations means that both $\leq_{\mathrm{i}}$ and $\leq_{\mathrm{si}}$ define partial orders on the set of finite graphs. The following fundamental result about graph immersions was conjectured by Nash-Williams.

Theorem 8.3 (Robertson and Seymour [RS10])**.** *In any infinite sequence of graphs* $(G_i)_{i=1}^{\infty}$*, there exists a pair* $i < j$ *such that* $G_i \leq_{\mathrm{i}} G_j$*.*

This property of the finite graphs with the order $\leq_{\mathrm{i}}$ is called **well-partial-ordered**. An immediate consequence of the theorem is that any subset of graphs that is upward-closed w.r.t. $\leq_{\mathrm{i}}$ has finitely many minimal elements. Similarly, any property which is downward-closed on immersion order can be described by finitely many graphs. This is a useful property, since for every fixed graph H there is a polynomial time algorithm to check whether there exists an immersion of H in the input graph G, see [FL92].

Abu-Khzam and Langston have explored connections between the immersion order and graph colorings, and made the following conjecture.

Conjecture 8.4 (Abu-Khzam and Langston [AL03])**.** *If* $\chi(G) \geq n$*, then* $K_n \leq_{\mathrm{i}} G$*.*

This is the immersion analogue of Hajós's refuted conjecture (where immersion order is replaced with topological minor order), and Hadwiger's unsolved conjecture (where immersion order is replaced with minor order). Observe, that

$$\mathscr{F}_n = \left\{ G \text{ finite graph } : \chi(G) < n \text{ and } \forall H \leq_{\mathrm{i}} G \text{ satisfies } \chi(H) < n \right\}$$

is a downward-closed set w.r.t. $\leq_{\mathrm{i}}$. Thus, its complement is upward-closed, and therefore has finitely many minimal elements; these graphs are called ***n*-immersion-critical**. It is easy to see that K_n is n-immersion-critical (any graph properly immersed in K_n has two vertices that are not joined by an edge). If true, Conjecture 8.4 would imply that K_n is immersed in any graph that is not in $\mathscr{F}_n$, i.e., K_n is the only n-immersion-critical graph. They also proved that

Theorem 8.5 (Abu-Khzam and Langston [AL03])**.** *If* G *is* n*-immersion-critical and* $G \not\cong K_n$*, then* G *is* n*-edge-connected.*

A trivial consequence of n-edge-connectivity is that $\delta(G) \geq n$. The above ideas motivate the following problem, which is the dual of the terminal-pairability problem in complete graphs (Problem 6.1).

Problem 8.6 (DeVos, Kawarabayashi, Mohar, and Okamura [DKMO10])**.** *Determine the minimum value of* $f(n)$ *such that any simple graph with minimum degree* $f(n)$ *contains an immersion of* K_n*.*

Maximum Edge-Disjoint Paths problem (MaxEDP)	
Input	Two loopless multigraphs, D and G, on the same vertex set
Feasible solution	A subgraph $D^* \subseteq D$ and its realization in G
Objective	Maximize $e(D^*)$
Decision version	Given (D, G, k), decide whether $\max e(D^*) \geq k$.

Table 8.1 Definition of the Maximum Edge-Disjoint Paths problem (*MaxEDP*)

Clearly, $f(n) \geq n - 1$. For small values of $n \leq 7$, it has been verified in [DKMO10] that $f(n) = n - 1$. However, a class of counterexamples to this equality have been constructed for $n \geq 8$ in [CH14]. The first bound proved on $f(n)$ is the following theorem.

Theorem 8.7 (Devos, Dvořák, Fox, McDonald, Mohar, and Scheide [DDFMMS14])**.** *If H is a simple graph with $\delta(H) \geq 200n$, then $K_n \leq_{\mathrm{si}} H$.*

Dvořák and Yepremyan [DY15] claim to have improved the lower bound on the minimum degree to $11n + 7$.

Theorem 6.4 has the following alternative statement in the language of immersions.

Corollary 8.8. *If H is a loopless multigraph on at most n vertices with $\Delta(H) \leq 2\lfloor \frac{n}{6} \rfloor - 4$, then $H \leq_{\mathrm{i}} K_n$.*

Combining our result with that of Dvořák and Yepremyan [DY15], we get a sufficient condition (with fairly strong asymptotic consequences) on when a loopless multigraph immerses in a simple graph.

Theorem 8.9. *Let H be a loopless multigraph, and let G be a simple graph. If*

$$\max\{33\Delta(H) + 172, 11|V(H)| + 7\} \leq \delta(G),$$

then $H \leq_{\mathrm{i}} G$.

8.2 Algorithms and complexity

The maximum edge-disjoint paths problem (see Table 8.1) is among the early problems shown to be *NP*-complete by Richard Karp [Kar75], although he referred to it as the “disjoint paths problem". For a fixed number of paths the problem is solvable in polynomial time (see [RS10]). However, if the number of required paths is part of the input, then the (decision version of the) problem is *NP*-complete even for series-parallel [NVZ01] and complete graphs [EV06]. This has been one of the reasons that forced us to consider an extremal approach to the terminal-pairability problem.

Surprisingly, and inadvertently, Theorem 6.4 gives the to date tightest approximation to the maximum edge-disjoint paths problem in complete graphs (see Theorem 8.14). Although these results are by-products of our study, we believe that this efficiency is not a coincidence, even though our approach has been an extremal one from the beginning.

We store graphs concurrently as edge lists and adjacency lists; an edge contains pointers to its copies in both lists. For a note on models of computation and graph representations, the reader is advised to consult Appendix A.

8.2.1 Algorithmic versions of Theorem 6.4 and 7.1

We will use the following results to find 2-factors.

Theorem 8.10 (Cole, Ost, and Schirra [COS01, Thm. 2])**.** *Given a regular bipartite multigraph on m edges, there is a deterministic algorithm that finds a complete matching in $O(m)$.*

Theorem 8.11 (Cole, Ost, and Schirra [COS01, Thm. 1])**.** *Given a regular bipartite multigraph on m edges with maximum degree Δ, there is a deterministic $O(m \log \Delta)$ time algorithm that finds a proper edge coloring using Δ colors.*

Using randomization, there are even more efficient algorithms to find perfect matchings (see Goel, Kapralov, and Khanna [GKK13]), however, using them would not improve the order of magnitude of the running time of the following theorem.

Theorem 8.12. *Given a loopless multigraph D on n vertices with $\Delta(D) \leq 2\lfloor \frac{n}{6} \rfloor - 4$, there is a deterministic $O(\Delta(D)^2 n)$ time algorithm which finds a realization of D in K_n.*

Proof. **Preprocessing.** We label each edge of D with a unique label. Whenever we lift a labeled edge, the two new edges inherit their ancestors label. Via these labels, we can recover the edge-disjoint paths in the solution.

Let the vertex set of the demand graph be $V(D) = \{v_1, \dots, v_n\}$. By lifting existing edges or joining non-maximal degree vertices (via an edge with a yet unused label), we can make the input graph $2\lceil \frac{\Delta(D)}{2} \rceil$-regular in $O(\Delta(D)n)$ time.

Iterative step (see Theorem 6.4). If D is 2-regular after preprocessing, a realization can be found in $O(n)$ time.

We can find an Eulerian orientation $\overrightarrow{D}$ in $O(\Delta(D)n)$ time. Construct a bipartite graph G by taking two copies, $\{v'_1, \dots, v'_n\}$ and $\{v''_1, \dots, v''_n\}$, of the vertex set of D, and join v'_i to v''_j iff there is an edge from v_i to v_j in $\overrightarrow{D}$. Observe, that perfect matchings of G correspond to 2-factors of D.

Via Theorem 8.10, two perfect matchings of G can be found in $O(\Delta(D)n)$ time, which correspond to two edge-disjoint 2-factors, A_1 and A_2, of D.

A lifting coloring of a 2-matching can be constructed in $O(n)$ time. A run of Lemma 6.11 can be computed in $O(n)$ time, as a lifting operation can be performed in $O(1)$. A set X_1 can be chosen in $O(n)$, and $A_2 - X_1$ can be extended to a maximal (not maximum!) 2-matching in $O(n)$, as well. Similarly, B_1 and B_2 can be determined in $O(n)$ time. The edges incident to the 6 vertices removed by two iterations of Lemma 6.11 are saved to a separate solution graph.

By lifting or adding a constant number of edges, we can make the remaining graph H_2 regular. Recurse on H_2.

Running time. The degree of the demand graph decreases by two in every iteration, whose running time is dominated by finding 2-factors. Theoretically, it could be profitable to compute a 2-factor decomposition of D during preprocessing (for example, via Theorem 8.11) and maintain this structure for subsequent iterations. However, it is unclear how an Eulerian orientation could be maintained in $o(\Delta(D)n)$ time, let alone a 2-factor decomposition. □

In the following we show using an argument which is similar to Kosowski's [see Kos08, Thm. 6], that there is a polynomial time $(3+\varepsilon_n)$-approximation scheme for the *MaxEDP* problem in K_n. We will need the following theorem.

Theorem 8.13 (Gabow [Gab83, Thm. 4.1])**.** *Let H be a multigraph on the vertex set $\{v_1, v_2, \dots, v_n\}$ with m edges. Let $u_i \in \mathbb{N}$. The spanning subgraph of H with the maximum number of edges in which $d_H(v_i) \le u_i$ (for all $i = 1, \dots, n$) can be found in $O(mn \log n)$ time and $O(m)$ space.*

We are ready to prove our approximation result of *MaxEDP* in complete graphs.

Theorem 8.14. *Let D be a demand graph on the vertex set of K_n. There is an $O(mn \log n + n^3)$ time algorithm which gives a $(3 + O(1/n))$-approximation solution to the MaxEDP problem in K_n.*

Proof. Let D_{opt} be a subgraph of D which is realizable in K_n, such that it has the maximum possible number of edges. Obviously, $\Delta(D_{\text{opt}}) \le n-1$. Run the algorithm of Theorem 8.13 on D with $u_i = 2\lfloor \frac{n}{6} \rfloor - 4$ (for $i = 1, \dots, n$) to obtain D^*.

According to Theorem 6.8, we can partition $E(D_{\text{opt}})$ into $\lceil \frac{n-1}{2} \rceil$ edge-disjoint 2-matchings. Order the 2-matchings in decreasing order of their cardinality, and choose the first $\lfloor \frac{n}{6} \rfloor - 2$. Let the spanning subgraph of D_{opt} formed by the union of the chosen 2-matchings be D'. Since $\Delta(D') \le 2\lfloor \frac{n}{6} \rfloor - 4$, we have $e(D') \le e(D^*)$. Furthermore, because we take the largest 2-matchings,

$$e(D^*) \ge e(D') \ge \frac{\lfloor \frac{n}{6} \rfloor - 2}{\lfloor \frac{n}{2} \rfloor} \cdot e(D^{\text{opt}}) \ge \left(\frac{1}{3} - \frac{5 + \frac{1}{3}}{n-1} \right) \cdot e(D^{\text{opt}}).$$

By Theorem 8.12, we can compute a realization of D^* in K_n in $O(n^3)$ time. □

Theorem 8.12 and Theorem 7.1 have the following consequence.

Corollary 8.15. *Let D be a loopless multigraph as a demand graph in K_t^d (which has $n = t^d$ vertices). If $\Delta(D) \le 2\lfloor \frac{t}{12} \rfloor - 2$, then there is a deterministic $O(d \cdot n \cdot \Delta(D)^2)$ time algorithm which finds a realization of D in K_t^d.*

Proof. Without loss generality, we may make D regular with an even degree. We can use Theorem 8.11 to find a 2-factor decomposition of D in $O(t^d \cdot \Delta(D) \cdot \log \Delta(D))$. Furthermore, this decomposition can be inherited by the layers L_i, so it does not have to be recomputed when the algorithms invokes recursion on the layers. Let $T_t^{\Delta(D)}(d)$ be a bound on the running time of the rest of the algorithm. We have

$$T_t^{\Delta(D)}(1) = O(\Delta(D)^2 t)$$
$$T_t^{\Delta(D)}(d) = O(\Delta(D) \cdot t^d) + t^{d-1} \cdot T_t^{2\Delta(D)}(1) + t \cdot T_t^{\Delta(D)}(d-1)$$

Solving the recursion, we get

$$T_t^{\Delta(D)}(d) = O\left(d \cdot t^d \cdot \Delta(D)^2\right) = O(d \cdot n \cdot \Delta(D)^2),$$

which clearly dominates the time spent preprocessing the graph. □

8.2.2 Comparing our results to the state of the art

According to a result of Carmi, Erlebach, and Okamoto [CEO03], the solution produced by a shortest-path-first or a bounded-length greedy algorithm is not better than a 3-approximation result for every input graph. Theorem 8.14 almost achieves this bound by producing a $(3+\varepsilon_n)$-approximation for any instance.

The champion before Theorem 8.14 was the 3.75-approximation algorithm of Kosowski [Kos08]. On demand graphs where $\overline{d}(D) = o(\Delta(D))$, our algorithm is up to a factor of n slower than that of Kosowski. However, if $\Delta(D) \geq 2\lfloor \frac{n}{6} \rfloor - 4$, we may replace Theorem 8.13 with a 2-matching decomposition to gain a $\log n$ on the running time. The solution produced by this modified algorithm is a $\left(\frac{3\Delta(D)}{n} + o(1)\right)$-approximation of the optimum.

8.3 Further base graphs

I am hopeful that our new results demonstrated in this part will increase interest in the terminal-pairability problem. Continuing this line of research, complete bipartite base graphs have been studied by Colucci, Erdős, Győri, and Mezei in two settings: the case when the demand graph is bipartite with respect to the classes of the base graph has been studied in [CEGM17a], and when no structural restrictions (other than maximum degree) are made on the demand graph is explored in [CEGM17b].

A risky and undertaking research direction is to study the terminal-pairability problem very generally, and to try to discover sufficient conditions for a demand graph to be realizable in a base graph, without a priori specifying too much information about any of them.

The degree conditions in our theorems are special cut conditions. Can we prove stronger theorems if we consider more than only single vertex cuts? In other words, does interpreting the problem as an integer multi-commodity flow task help?

Problem 8.16. *Suppose D and G are loopless multigraphs on the vertex set $V = \{1, \dots, n\}$. What is the minimum value of $f(n)$ so that*

$$f(n) \leq \min_{\emptyset \neq A \subset V} \frac{e_G(A, V - X)}{e_D(A, V - X)} \implies D \text{ is realizable in } G?$$

By choosing G as a regular expander graph, one can prove that $f(n) \geq \Omega(\log n)$. If G is required to be simple, a similar construction implies $f(n) \geq \Omega(\log n / \log \log n)$.

References

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

[AL03] Faisal N. Abu-Khzam and Michael A. Langston, *Graph coloring and the immersion order*, Computing and combinatorics, vol. 2697, Lecture Notes in Comput. Sci. Springer, Berlin, 2003, pp. 394–403, DOI: 10.1007/3-540-45071-8_40, MR2063516.

[AZ14] Martin Aigner and Günter M. Ziegler, *Proofs from The Book*, Fifth Edition, Including illustrations by Karl H. Hofmann, Springer-Verlag, Berlin, 2014, pp. viii+308, ISBN: 978-3-662-44205-0, DOI: 10.1007/978-3-662-44205-0, MR3288091.

[BCLMMV16] Therese C. Biedl, Timothy M. Chan, Stephanie Lee, Saeed Mehrabi, Fabrizio Montecchiani, and Hamideh Vosoughpour, *On Guarding Orthogonal Polygons with Sliding Cameras*, arXiv preprint (2016), arXiv: 1604.07099v1.

[CEGM17a] Lucas Colucci, Péter L Erdős, Ervin Győri, and Tamás Róbert Mezei, *Terminal-Pairability in Complete Bipartite Graphs*, submitted to Discrete Appl. Math. (2017), arXiv: 1702.04313.

[CEGM17b] Lucas Colucci, Péter L Erdős, Ervin Győri, and Tamás Róbert Mezei, *Terminal-Pairability in Complete Bipartite Graphs with Non-Bipartite Demands*, submitted to Theoret. Comput. Sci. (2017), arXiv: 1705.02124.

[CEO03] Paz Carmi, Thomas Erlebach, and Yoshio Okamoto, *Greedy edge-disjoint paths in complete graphs*, Graph-theoretic concepts in computer science, vol. 2880, Lecture Notes in Comput. Sci. Springer, Berlin, 2003, pp. 143–155, DOI: 10.1007/978-3-540-39890-5_13, MR2080076.

[CFGLS92] László Csaba, Ralph J. Faudree, András Gyárfás, Jenő Lehel, and Richard H. Schelp, *Networks communicating for each pairing of terminals*, Networks 22.**7** (1992), pp. 615–626, ISSN: 0028-3045, DOI: 10.1002/net.3230220702, MR1189290.

[CH14] Karen L. Collins and Megan E. Heenehan, *Constructing graphs with no immersion of large complete graphs*, J. Graph Theory 77.**1** (2014), pp. 1–18, ISSN: 0364-9024, DOI: 10.1002/jgt.21770, MR3231086.

[Cha91] Bernard Chazelle, *Triangulating a simple polygon in linear time*, Discrete Comput. Geom. 6.**5** (1991), pp. 485–524, ISSN: 0179-5376, DOI: 10.1007/BF02574703, MR1115104.

[Chv75] Vašek Chvátal, *A combinatorial theorem in plane geometry*, J. Combinatorial Theory Ser. B 18 (1975), pp. 39–41, MR0372750.

[COS01] Richard Cole, Kirstin Ost, and Stefan Schirra, *Edge-coloring bipartite multigraphs in $O(E \log D)$ time*, Combinatorica 21.**1** (2001), pp. 5–12, ISSN: 0209-9683, DOI: 10.1007/s004930170002, MR1805711.

[DDFMMS14] Matt Devos, Zdeněk Dvořák, Jacob Fox, Jessica McDonald, Bojan Mohar, and Diego Scheide, *A minimum degree condition forcing complete graph immersion*, Combinatorica 34.**3** (2014), pp. 279–298, ISSN: 0209-9683, DOI: 10.1007/s00493-014-2806-z, MR3223965.

[Die10] Reinhard Diestel, *Graph theory*, Fourth Edition, vol. 173, Graduate Texts in Mathematics, Springer, Heidelberg, 2010, pp. xviii+437, ISBN: 978-3-642-14278-9, DOI: 10.1007/978-3-642-14279-6, MR2744811.

[DKMO10] Matt DeVos, Ken-ichi Kawarabayashi, Bojan Mohar, and Haruko Okamura, *Immersing small complete graphs*, Ars Math. Contemp. 3.**2** (2010), pp. 139–146, ISSN: 1855-3966, MR2729363.

[DM13] Stephane Durocher and Saeed Mehrabi, *Guarding orthogonal art galleries using sliding cameras: algorithmic and hardness results*, Mathematical foundations of computer science 2013, vol. 8087, Lecture Notes in Comput. Sci. Springer, Heidelberg, 2013, pp. 314–324, DOI: 10.1007/978-3-642-40313-2_29, MR3126217.

[DMK90] Peter Damaschke, Haiko Müller, and Dieter Kratsch, *Domination in convex and chordal bipartite graphs*, Inform. Process. Lett. 36.**5** (1990), pp. 231–236, ISSN: 0020-0190, DOI: 10.1016/0020-0190(90)90147-P, MR1081460.

[DY15] Zdeněk Dvořák and Liana Yepremyan, *Complete graph immersions and minimum degree*, arXiv preprint (2015), arXiv: 1512.00513v1.

[EV06] Thomas Erlebach and Danica Vukadinović, *Path problems in generalized stars, complete graphs, and brick wall graphs*, Discrete Appl. Math. 154.**4** (2006), pp. 673–683, ISSN: 0166-218X, DOI: 10.1016/j.dam.2005.05.017, MR2202473.

[Fau92] Ralph J. Faudree, *Properties in pairable graphs*, New Zealand J. Math. 21 (1992), New Zealand Mathematics Colloquium (Dunedin, 1991), pp. 91–106, ISSN: 1171-6096, MR1167462.

[FGL92] Ralph J. Faudree, András Gyárfás, and Jenő Lehel, *Minimal path pairable graphs*, Proceedings of the Twenty-third Southeastern International Conference on Combinatorics, Graph Theory, and Computing (Boca Raton, FL, 1992), vol. 88, 1992, pp. 111–128, MR1208923.

[FGL99] Ralph J. Faudree, András Gyárfás, and Jenő Lehel, *Path-pairable graphs*, J. Combin. Math. Combin. Comput. 29 (1999), pp. 145–157, ISSN: 0835-3026, MR1677781.

[Fis78] Steve Fisk, *A short proof of Chvátal's watchman theorem*, J. Combin. Theory Ser. B 24.**3** (1978), p. 374, ISSN: 0095-8956, DOI: 10.1016/0095-8956(78)90059-X, MR496460.

[FL92] Michael R. Fellows and Michael A. Langston, *On well-partial-order theory and its application to combinatorial problems of VLSI design*, SIAM J. Discrete Math. 5.**1** (1992), pp. 117–126, ISSN: 0895-4801, DOI: 10.1137/0405010, MR1146902.

[FM84] Alain Fournier and Delfin Y Montuno, *Triangulating simple polygons and equivalent problems*, ACM Transactions on Graphics (TOG) 3.**2** (1984), pp. 153–174.

[Fra13] András Frank, *Diszkrét Optimalizálás jegyzet*, 2013, eprint: http://www.cs.elte.hu/~frank/jegyzet/disopt/dopt13.pdf.

[Gab83] Harold N Gabow, *An efficient reduction technique for degree-constrained subgraph and bidirected network flow problems*, Proceedings of the fifteenth annual ACM symposium on Theory of computing, ACM, 1983, pp. 448–456.

[GG78] Martin Charles Golumbic and Clinton F. Goss, *Perfect elimination and chordal bipartite graphs*, J. Graph Theory 2.**2** (1978), pp. 155–163, ISSN: 0364-9024, DOI: 10.1002/jgt.3190020209, MR493395.

[GHKS96] Ervin Győri, Frank Hoffmann, Klaus Kriegel, and Thomas Shermer, *Generalized guarding and partitioning for rectilinear polygons*, Comput. Geom. 6.**1** (1996), pp. 21–44, ISSN: 0925-7721, DOI: 10.1016/0925-7721(96)00014-4, MR1387672.

[GiMé17] António Girão and Gábor Mészáros, *An improved upper bound on the maximum degree of terminal-pairable complete graphs*, arXiv preprint (Aug. 24, 2017), arXiv: 1708.06837.

[GKK13] Ashish Goel, Michael Kapralov, and Sanjeev Khanna, *Perfect matchings in $O(n \log n)$ time in regular bipartite graphs*, SIAM J. Comput. 42.**3** (2013), pp. 1392–1404, ISSN: 0097-5397, DOI: 10.1137/100812513, MR3504633.

[GM16] Ervin Győri and Tamás Róbert Mezei, *Partitioning orthogonal polygons into ≤ 8-vertex pieces, with application to an art gallery theorem*, Comput. Geom. 59 (2016), pp. 13–25, ISSN: 0925-7721, DOI: 10.1016/j.comgeo.2016.07.003, MR3545993.

[GM17] Ervin Győri and Tamás Róbert Mezei, *Mobile vs. point guards*, submitted to Discrete Comput. Geom. (2017), arXiv: 1706.02619.

[GMM16] Ervin Győri, Tamás Róbert Mezei, and Gábor Mészáros, *Terminal-Pairability in Complete Graphs*, J. Combin. Math. Combin. Comput. (2016), accepted for publication (October 2016), arXiv: 1605.05857.

[GMM17] Ervin Győri, Tamás Róbert Mezei, and Gábor Mészáros, *Note on terminal-pairability in complete grid graphs*, Discrete Math. 340.**5** (2017), pp. 988–990, ISSN: 0012-365X, DOI: 10.1016/j.disc.2017.01.014, MR3612429.

[GT85] Harold N. Gabow and Robert Endre Tarjan, *A linear-time algorithm for a special case of disjoint set union*, J. Comput. System Sci. 30.**2** (1985), pp. 209–221, ISSN: 0022-0000, DOI: 10.1016/0022-0000(85)90014-5, MR801823.

[Győ86] Ervin Győri, *A short proof of the rectilinear art gallery theorem*, SIAM J. Algebraic Discrete Methods 7.**3** (1986), pp. 452–454, ISSN: 0196-5212, DOI: 10.1137/0607051, MR844048.

[HK91] Frank Hoffmann and Michael Kaufmann, *On the rectilinear art gallery problem—algorithmic aspects*, Graph-theoretic concepts in computer science (Berlin, 1990), vol. 484, Lecture Notes in Comput. Sci. Springer, Berlin, 1991, pp. 239–250, DOI: 10.1007/3-540-53832-1_46, MR1114588.

[Hof90] Frank Hoffmann, *On the rectilinear art gallery problem*, Automata, Languages and Programming: 17th International Colloquium Warwick University, England, July 16–20, 1990 Proceedings, ed. by Michael S. Paterson, Berlin, Heidelberg: Springer Berlin Heidelberg, 1990, pp. 717–728, ISBN: 978-3-540-47159-2, DOI: 10.1007/BFb0032069.

[Hon76] Ross Honsberger, *Mathematical gems. II*, Dolciani Mathematical Expositions, No. 2, The Mathematical Association of America, Washington, D.C., 1976, pp. ix+182, MR0472273.

[IK00] W. Imrich and S. Klavžar, *Product Graphs: Structure and Recognition*, J. Wiley& Sons, New York, 2000.

[Kar75] R. Karp, *On the computational complexity of combinatorial problems*, Networks 5 (1975), pp. 45–68.

[KKK83] J. Kahn, M. Klawe, and D. Kleitman, *Traditional galleries require fewer watchmen*, SIAM J. Algebraic Discrete Methods 4.**2** (1983), pp. 194–206, ISSN: 0196-5212, DOI: 10.1137/0604020, MR699771.

[KKL99] Ewa Kubicka, Grzegorz Kubicki, and Jenő Lehel, *Path-pairable property for complete grids*, Combinatorics, graph theory, and algorithms, Vol. I, II (Kalamazoo, MI, 1996), New Issues Press, Kalamazoo, MI, 1999, pp. 577–586, MR1985088.

[KKT92] David G. Kirkpatrick, Maria M. Klawe, and Robert E. Tarjan, *Polygon triangulation in $O(n \log\log n)$ time with simple data structures*, Discrete Comput. Geom. 7.**4** (1992), pp. 329–346, ISSN: 0179-5376, DOI: 10.1007/BF02187846, MR1148949.

[KM11] Matthew J. Katz and Gila Morgenstern, *Guarding orthogonal art galleries with sliding cameras*, Internat. J. Comput. Geom. Appl. 21.**2** (2011), pp. 241–250, ISSN: 0218-1959, DOI: 10.1142/S0218195911003639, MR2786379.

[Kos08] Adrian Kosowski, *The maximum edge-disjoint paths problem in complete graphs*, Theoret. Comput. Sci. 399.**1-2** (2008), pp. 128–140, ISSN: 0304-3975, DOI: 10.1016/j.tcs.2008.02.017, MR2416955.

[LM96] Mario A. Lopez and Dinesh P. Mehta, *Efficient Decomposition of Polygons into L-shapes with Application to VLSI Layouts*, ACM Trans. Des. Autom. Electron. Syst. 1.**3** (July 1996), pp. 371–395, ISSN: 1084-4309, DOI: 10.1145/234860.234865.

[LTL89] W. T. Liou, J. J. Tan, and R. C. Lee, *Minimum Partitioning Simple Rectilinear Polygons in O(N Log Log N) - Time*, Proceedings of the Fifth Annual Symposium on Computational Geometry, SCG '89, Saarbruchen, West Germany: ACM, 1989, pp. 344–353, ISBN: 0-89791-318-3, DOI: 10.1145/73833.73871.

[Lub85] Anna Lubiw, *Decomposing Polygonal Regions into Convex Quadrilaterals*, Proceedings of the First Annual Symposium on Computational Geometry, SCG '85, Baltimore, Maryland, USA: ACM, 1985, pp. 97–106, ISBN: 0-89791-163-6, DOI: 10.1145/323233.323247.

[LWŻ12] Andrzej Lingas, Agnieszka Wasylewicz, and Paweł Żyliński, *Linear-time 3-approximation algorithm for the r-star covering problem*, Internat. J. Comput. Geom. Appl. 22.**2** (2012), pp. 103–141, ISSN: 0218-1959, DOI: 10.1142/S021819591250001X, MR2981125.

[MB87] Haiko Müller and Andreas Brandstädt, *The NP-completeness of* STEINER TREE *and* DOMINATING SET *for chordal bipartite graphs*, Theoret. Comput. Sci. 53.**2-3** (1987), pp. 257–265, ISSN: 0304-3975, DOI: 10.1016/0304-3975(87)90067-3, MR918093.

[Més16] Gábor Mészáros, *On path-pairability in the Cartesian product of graphs*, Discuss. Math. Graph Theory 36.**3** (2016), pp. 743–758, ISSN: 1234-3099, DOI: 10.7151/dmgt.1888, MR3518137.

[NVZ01] Takao Nishizeki, Jens Vygen, and Xiao Zhou, *The edge-disjoint paths problem is NP-complete for series-parallel graphs*, Discrete Appl. Math. 115.**1-3** (2001), 1st Japanese-Hungarian Symposium for Discrete Mathematics and its Applications (Kyoto, 1999), pp. 177–186, ISSN: 0166-218X, DOI: 10.1016/S0166-218X(01)00223-2, MR1869356.

[ORo87] Joseph O'Rourke, *Art gallery theorems and algorithms*, International Series of Monographs on Computer Science, The Clarendon Press, Oxford University Press, New York, 1987, pp. xvi+282, ISBN: 0-19-503965-3, MR921437.

[Pet91] Julius Petersen, *Die Theorie der regulären graphs*, Acta Math. 15.**1** (1891), pp. 193–220, ISSN: 0001-5962, DOI: 10.1007/BF02392606, MR1554815.

[RS10] Neil Robertson and Paul Seymour, *Graph minors XXIII. Nash-Williams' immersion conjecture*, J. Combin. Theory Ser. B 100.**2** (2010), pp. 181–205, ISSN: 0095-8956, DOI: 10.1016/j.jctb.2009.07.003, MR2595703.

[Sch88] Arnold Schönhage, *A nonlinear lower bound for random-access machines under logarithmic cost*, J. Assoc. Comput. Mach. 35.**3** (1988), pp. 748–754, ISSN: 0004-5411, DOI: 10.1145/44483.44492, MR963171.

[SH95] Dietmar Schuchardt and Hans-Dietrich Hecker, *Two NP-hard art-gallery problems for ortho-polygons*, Math. Logic Quart. 41.**2** (1995), pp. 261–267, ISSN: 0942-5616, DOI: 10.1002/malq.19950410212, MR1330860.

[WK07] Chris Worman and J. Mark Keil, *Polygon decomposition and the orthogonal art gallery problem*, Internat. J. Comput. Geom. Appl. 17.**2** (2007), pp. 105–138, ISSN: 0218-1959, DOI: 10.1142/S0218195907002264, MR2309901.

Appendix A

A note on models of computation and representations of graphs

When multigraphs are part of the input, one should exercise great care when determining running times. The problem has its roots in the details of the graph representation used when describing the input and output graphs.

Let us choose the very common edge list representation. Consider the following problem: given an input multigraph with $m \leq \binom{n}{2}$ edges on the vertex set $\{1, \ldots, n\}$, output any simple graph with m edges on the vertex set of the input graph. Clearly, the output needs $\Omega(m \log n)$ space. However, if the edges of the multigraph are incident to only a subset of the vertex set, say, $\{1, 2, 3, 4\}$, then the input may need only $O(m + \log n)$ bits of space. Thus, an algorithm solving this problem cannot have a running time which is at most a linear function of the size of the input.

The running time of breadth first and depth first search algorithms is usually regarded as $O(m+n)$, but a factor of $\log n$ is clearly missing. However, these algorithms are correctly regarded as having a linear running time as a function of the size of the input. It is easy to see that describing a simple graph with $\Omega(n)$ edges requires $\Omega(n \log n)$ space, and thus the extra $\log n$ factor is usually not a problem.

One could argue that the previous problem is only a question of whether one chooses the unit cost or the logarithmic cost RAM machine model. However, we may exacerbate the problem (of describing the running time of an algorithm in terms of the size of its input) further by describing a multiedge by the vertices it joins and its multiplicity. Then the size of the input may be as low as $O(\log m + \log n)$.

However, the logarithmic cost RAM machine has surprising limitations. Schönhage [Sch88] showed, that storing n arbitrary bits takes $\Omega(n \log^* n)$ time in this model.

For these reasons, our choice for the model of computation is the unit cost RAM machine for both Chapter 4 and Chapter 8. Alternatively, one may multiply the running time of our algorithms in Chapter 8 by a factor of $O(\log n)$ (where n is the number of vertices of the output graph) to get the logarithmic cost running times of our algorithms. However, the $O(n)$ algorithms outlined in Chapter 4 remain linear even in the logarithmic cost model.

Appendix B

Algorithms on orthogonal polygons

Algorithm B.1 Finding the horizontal cuts of an orthogonal polygon, part I

Require: P orthogonal polygon

Ensure: $pair[v]$ will contain the vertical side of P which the other end of the horizontal cut starting at the reflex vertex v intersects

```
function FIND HORIZONTAL CUTS(P)
    n ← number of vertices of P
    T ← TRIANGULATE(P)                                        ▷ List of triangles
    Initialize L[v] = ∅, doubly linked lists for each vertex v of P
    for all t ∈ T do
        for i = 1, 2, 3 do
            L[t.v_i] ← append a link to t
            t.l_i ← a link to t's location in L[t.v_i]
        end for
    end for

    for all side or diagonal s of P do
        Initialize S[s] to empty doubly linked list
        if s is a vertical side of P then
            S[s] ← insert s
        end if
    end for
```

Algorithm B.2 Finding the horizontal cuts of an orthogonal polygon, part II

$Q \leftarrow$ a queue of vertices with $|L[v]| = 1$
while $Q \neq \emptyset$ **do**
$u \leftarrow$ pop the first element of Q
$t \leftarrow L[u]$
$v, w \leftarrow$ other two vertices of t so that $w.y \leq v.y$
$s_u, s_v, s_w \leftarrow$ sides of t opposite the vertex in the index
Delete t from T and the 3 links to it in L, update Q
if $w.y \leq u.y \leq v.y$ **then**
$S[s_u] \leftarrow S[s_w]$ append $S[s_v]$
else if s_u is horizontal **then**
Process $S[s_v]$ and $S[s_w]$ in y-order and update *pair*[] for the vertices in them
else if $v.y < u.y$ **then**
$S[s_u] \leftarrow$ segment of $S[s_v]$ visible along the x-axis from s_u
$R \leftarrow$ segment of $S[s_v]$ **not** visible along the x-axis from s_u
Process R and $S[s_w]$ in y-order and update *pair*[] for the vertices in them
else if $w.y > u.y$ **then**
$S[s_u] \leftarrow$ segment of $S[s_w]$ visible along the x-axis from s_u
$R \leftarrow$ segment of $S[s_w]$ **not** visible along the x-axis from s_u
Process R and $S[s_v]$ in y-order and update *pair*[] for the vertices in them
end if

if s_u is a vertical side of P **then**
for all reflex vertex z in $S[s_u]$ **do**
if the horizontal cut of z starts towards s_u **then**
pair$[z] \leftarrow s_u$
end if
end for
if v is a reflex vertex **then**
pair$[v] \leftarrow$ the first or last element of $S[s_u]$ that contains a point with the same y-coordinate as v
end if
if w is a reflex vertex **then**
pair$[w] \leftarrow$ the first or last element of $S[s_u]$ that contains a point with the same y-coordinate as w
end if
end if
end while
return the list *pair*[]
end function

Algorithm B.3 Constructing the horizontal R-tree of an orthogonal polygon

```
function HORIZONTAL R-TREE(P)
    opposite_side[] ← FIND HORIZONTAL CUTS(P)                ▷ see Algorithm B.1
    for all reflex vertex v of P do
        w ← new vertex at the height of v.y in opposite_side[v]
        if not present, insert w into P
        cut_pair[v] ← w
        cut_pair[w] ← v
    end for

    G ← ({P}, ∅)                                ▷ G is a graph with a single node P
    Initialize cut_to_edge[] empty
    CurrentNode ← P
    for all vertex v of P in clockwise order do
        if v is a reflex or a new vertex then
            if cut_to_edge[v] = ∅ then
                Split CurrentNode along {v, cut_pair[v]} in G
                e ← {N_1, N_2} the two new pieces whose union is CurrentNode
                G ← G + e
                cut_to_edge[v] ← e
                cut_to_edge[cut_pair[v]] ← e
                CurrentNode ← the piece containing clockwise_next_P(v)
            else
                CurrentNode ← pair of CurrentNode in cut_to_edge[v]
            end if
        end if
    end for
    return G
end function
```

Algorithm B.4 Finding a minimum cardinality horizontal r-guard system

```
function SOLVE MHSC(P)
    T_H ← HORIZONTAL R-TREE(P)                                   ▷ Algorithm B.3
    T_V ← VERTICAL R-TREE(P)
    for all vertical slice t ∈ T_V do
        a, b ← vertical sides of P bounding t
        h_a ← horizontal slice in V(T_H) containing a
        h_b ← horizontal slice in V(T_H) containing b
        N[t] ← {h_a, h_b}
    end for

    r ← arbitrary node of T_H to serve as root
    dist[] ← BREADTH FIRST SEARCH(T_H, r)                         ▷ distance from r

    LCA[] ← LOWEST COMMON ANCESTORS(T_H, r, N[])                 ▷ Algorithm of [GT85]
                       ▷ LCA[t] contains the lowest common ancestors of the elements of N[t]

    S ← ∅
    Set every node of T_H unmarked
    for all t ∈ V(T_V) so that dist[LCA[t]] is not increasing do   ▷ reverse BFS-order
        if both elements of N[t] are unmarked then
            S ← S ∪ {LCA[t]}
            SET MARK(LCA[t])
        end if
    end for
    return S
end function

function SET MARK(u)
    for all neighbor w of u in T_H do
        if dist[w] > dist[u] and w is unmarked then
            SET MARK(w)
        end if
    end for
end function
```